Technische Universität München
Lehrstuhl für Mensch-Maschine-Kommunikation

Diplomarbeit

Polyphonic Music Transcription based on Real Audio

Verfasser:	Peter Matthias Grosche Zasingerstr. 8 81547 München
Matrikelnummer:	2555252
Betreuer:	Dr.-Ing. Björn Schuller
Abgabetermin:	30.04.2008

Abstract

Automatic music transcription is a challenging task that has been the topic of many research groups since 1977. The problem has been addressed in several manners and suitable results were obtained for the monophonic case, where a single note is played at a time. Still, the transcription of real-world polyphonic music with arbitrary instruments, genre and tempi suffers from many unsolved problems.
The two key sub-problems in automatic music transcription necessary to obtain a note-level parameter representation of a musical piece are polyphonic pitch estimation and notes' onset detection. Both tasks are addressed in this thesis for real-world music.
Effective signal representations are proposed that adapt the properties of music signals. The spectral characteristics are described with a semitone spectrogram that receives enhanced time/frequency resolution through the use of a multiresolution DFT and instantaneous frequency estimation. Non-harmonic signal components are suppressed using RASTA processing.
The temporal characteristics are modeled in complex domain, taking envelope and frequency continuity conditions into account. An iterative approach is proposed for the polyphonic pitch estimation that detects the predominant F0 in a signal and uses harmonic spectrum estimation to extract it, before the estimation is continued with the residual. The onset and duration detection of each note is resolved through temporal properties of the signal.
In addition, Hidden Markov Models are used to model each note as a distinct event with both, spectral and temporal characteristics simultaneously to determine its pitch, onset and offset.
To further enhance the results achievable with the methods based on signal properties, musical knowledge is used, namely the musical key to find probable notes in a song, and musical meter to define probable on- and offset positions. The evaluation of the proposed methods is carried out on a database of acoustic recordings, whose reference transcription are obtained through force alignment with midi files.
Note precision and recall rates above 60% are achieved, together with a temporal accuracy of 40%.

Contents

Chapter 1

Introduction

1.1 Music Transcription

Transcribing musical recordings is the antipode of making music: *Listening to a piece of music and writing down its musical notation.*

This musical notation, or score, represents the parameters of each note in a composition: the pitch, onset, duration, loudness and instrument playing the note. Additionally to this note-level parameters, song-level parameters that provide further information about the structure of the whole musical piece are defined by the notation. These are the tempo, meter and musical key, but are not necessary to reflect the composition. For this, the note-level parameters are sufficient and all overall structure defining information is for support of the musician. An example of the traditional musical notation is shown in Figure 1.1, defining each note with it distinct parameters *pitch*, *onset* and *duration*, as well as the *musical key* and the *meter* of the whole composition.

The musical score of a composition is not meant to be a strict rule for the musician how to play a song, but rather a recommendation or guideline to support the musician and let room for individual influences. Especially in classical music this is observable, where often many different recordings of the same musical piece exist that are performed by different conductors using different instruments and variations.

So *Automatic* Music Transcription (AMT) is the process of converting the acoustical waveform of a musical piece into a musical notation by the use of signal processing methods. AMT can be seen as *reverse-engineering the source code of a music signal* [40], to estimate both, note-level and song-level parameters.

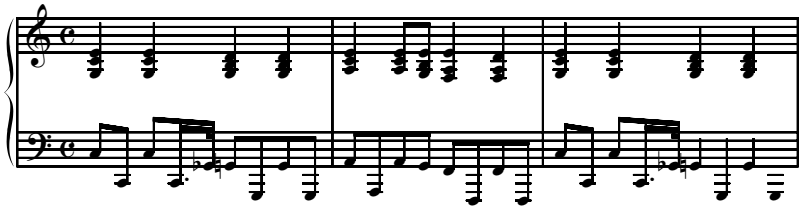


Figure 1.1: Typical score of a musical piece: Extract of *The Beatles - Let it be*.

1.2 Motivation

Transcription of polyphonic music is a nearly impossible task for the average human without any musical education. However, skilled musicians are able to transcribe music with high accuracy, but still, the transcription remains a very time consuming task.

Music is not only transcribed by the hobby musician, who wants to be able to play his favourite song from the radio, but also by experts, when it comes to the recovery of lost musical sheets, or recordings of unknown composers.

Automatic transcription systems would allow to cut down the time needed to regenerate the transcription of an existing recording considerably.

Real-time transcription of live music would allow to create systems for the assistance in musical education that are able to react and correct the performance of the pupil, by detecting wrong notes and timings, giving adequate advices and react like a real music teacher.

Aside from the transcription itself, the musical notation or parameter representation of music obtained through automatic transcription is a critical step for some very interesting applications, including content-based music retrieval, automatic music summarization, interactive music systems, low-bitrate compression coding for music signals and so on.

Multimedia music applications are nowadays rapidly moving from simple metadata related scenarios to more complex domains with content based descriptions and annotations for song identification. The creation of huge databases of digital content requires reliable and fast methods for content based analysis and description, to be used for database annotation and the connection of music in different representations: audio, sheet and midi files.

In the case of content-based music retrieval, the musical notation is crucial. Measures of similarity between musical excerpts, for instance, can easily be

defined, when harmonic and rhythm information is known. Higher level music analysis tasks such as melody extraction, rhythm tracking, harmony analysis, audio thumbnailing or query by example are a lot easier to perform, when the general parameter representation of the recording is known.

For the first time, the musical recording allows an intuitive display of the content of the recording that is readable by a majority of the people, and thus, allows a rapid *insight* on the content of the recording.

Another possible application of the generation of a parameter representation of a musical recording is to assist for the very-low-bitrate coding of music signal. The MPEG-4 Structured Audio coding standard [28] provides new methods for very-low-bitrate storage using parametric compression, in a similar way as the compression used for speech signals through parametric voice-coders.

In general, the high-level parameters obtained through automatic transcription of musical recordings would lead to the possibility to describe all attributes of a musical piece, apart from the individual influences of the musician and the instruments used.

1.3 Decomposition of the Transcription Task

The complex structure of automatic music transcription requires to split the task into smaller, less complex subtasks to achieve the overall goal.

A musical note is defined in the musical notation by five essential parameters that have to be estimated for a note-level representation:

1. Pitch: The perception defined by the physical fundamental frequency (F0), the lowest frequency in a harmonic series that is the inverse of the pitch period length in time domain.

2. Onset: The beginning of a note, i.e. the time where the vibration of the instrument is induced.

3. Offset: The end of a note, i.e. the time where the incitement of the vibrating system is stopped. But the vibration may still be present beyond this time and fading away. Onset and offset time together define the duration of a note.

4. Loudness: How *loud* the note is sounding, affected by the strength of the inducement of the vibrating system.

5. The instrument playing this note.

Additionally, the structure of the whole composition is described by some global (song-level) properties in the musical score:

1. Musical key: Defined by the chords used in the piece.
2. Rhythm: Defining the tempo and meter, and thus, the beat position that define the temporal structure of the notes in the song, with temporal continuity, repetition and pattern.
3. Harmony: The structure of the notes played at the same time, i.e. *Chords*, defined by the vertical intervals.
4. Melody: The linear event of notes played in a row. This includes the change of notes' pitch and duration over time, i.e. the linear intervals.

1.4 State of the Art

Music transcription has been the topic of many research projects, starting in the mid 1970 and is still not solved satisfyingly, although many research groups focus on this tasks and a wide variety of different approaches was proposed.

Due to this complex character of the transcription problem, many restrictions have been made and the overall task has been split into smaller subtasks or attempts to obtain only a partial transcription.

1.4.1 Database Restrictions

Most of the transcription systems proposed so far constrain the input material in some way to reduce the complexity of AMT.

Musical genres are limited in [30], [46] and [11]. Some systems focus on specific instruments ([6] or [20]), especially piano music attracted a lot of attention due to its rather simple and straight-forward structure, ([44], [49], [45] or [56]). Others restrict their approach to only partial transcription of complex musical signals, [24]. Synthesized music is used in favor over real musical recordings, due to the easily obtainable reference transcription through midi files, although their synthetic character is meant to simplify the transcription task.

Drums and percussion transcription systems were developed, too, but most of the voiced music transcription systems exclude drums from input material, while others do allow but do not transcribe them.

Only recently were systems proposed that work on any types of music and do not make any restrictions on input material [62].

1.4.2 Time Frequency Representations

The nonstationary character of musical signals leads to the necessity of adequate methods of Time-Frequency Representations (TFR). The Fourier uncertainty principle [8] states that it is impossible for a TFR to have both, the best time and frequency resolution at the same time. Different methods of TFR have been proposed in terms of music signal representations, some of them with the ability to bypass this constraint [48].

A multiresolution fourier transform is used in [13], to allow higher frequency resolution in lower frequency ranges, and at the same time keep a sufficient time resolution in higher frequency ranges.

The Discrete Wavelet Transform (DWT) is another method of TFR that has a constant-Q frequency resolution that is similar to the inner ears frequency resolution and is very appropriate to music representation, because it allows different time-frequency resolutions in each frequency range.

An Auditory Filter Bank transformation is commonly used whenever a system should adapt the human perception of sound [39] and follows the assumption that the basilar membrane *filters* parts of a sound into the auditory-nerve fibres, so that the output of the cochlea is like the output of a bank of filters. Each of these *auditory* filters is centred at a different frequency, and responds to only a narrowed range of frequency and thus, a filter bank is created.

Using the temporal derivative of the phase, the frequency *resolution* can not be enhanced, but the actual frequency of each component can be estimated more precisely. This method is called *instantaneous frequency* estimation and is used in [13].

1.4.3 Estimation of Pitch

Determining the fundamental frequency of a musical sound is the most challenging task in music transcription and many affords have been made to solve this problem.

Monophonic Pitch Estimation

Early works focus on the fundamental frequency determination in speech signals for the use in speech recognition systems. In musical signals with wider bandwidth, the need for higher resolution and multiple fundamental frequencies at the same time, most of these algorithms are not suitable.

Rabiner *et al.* presented an early comparison of different monophonic fundamental frequency estimation methods in 1976 [59], examining frequency domain, time domain and hybrid algorithms focusing on speech signals.

Piszczalski and Galler [53] proposed the first monophonic music transcription system in 1977 limited to certain types of instruments with strong fundamental frequencies that - in frequency domain - simply chose the most dominant partial as the fundamental. The system was enhanced to instruments with arbitrary harmonic structure in [54], but the basic algorithm and thus, the limitations remained the same.

Algorithms in Time Domain: A common approach to fundamental frequency estimation in time domain is the autocorrelation method [7] to find the periodicity of the signal, because it is simple, fast and reliable [3]. Time domain peak and valley, zero-crossing and cepstrum measurements are other methods used. In addition to being computationally more efficient than methods in frequency domain they can even be used to construct real-time transcription systems.

More complex, psychoacoustically motivated methods measure the periodicity of the amplitude envelope in different subbands obtained through an auditory filterbank, as in [69] or [57].

Frequency domain: Pitch estimation methods that work in frequency domain analyse the spectral patterns of the harmonic structure, i.e. the relation of harmonic frequencies, to find fundamental frequencies whose harmonics best fit the overall structure of the harmonic signal [26], or the periodicity in frequency domain, i.e. the interval of the partials. The spectrum autocorrelation method is used in [43].

Polyphonic Pitch Estimation

Monophonic pitch estimation is not yet completely solved [23] and rather useless for real world polyphonic music with an unknown number of multiple notes sounding at the same time. Most of the proposed polyphonic

transcription systems made restrictions in some way on the input material to reduce the complexity of the task. Only recently systems were developed that do work on all types of musical signals, without limiting or predefining the polyphony, number and types of instruments, the presence of drums or *tempi* and *meter*.

At the same time as Piszczalski and Galler released their work on monophonic transcription [53], Moorer [50] proposed the first polyphonic music transcription system based on finding the root, or *greatest common harmonic* of the harmonic sounds. The system focused on duets and thus, was limited to the use of any two monophonic instruments and no overlap between the notes played or any of their harmonics was allowed. Beat tracking and note length quantization needed some kind of user interaction.

Recent methods with the focus on general music signals try to decompose the signal into smaller elements to separate the multiple notes sounding at a time. Sinusoidal component tracking and grouping into sound sources according to their individual attributes [24] or comb filtering in the time domain can achieve this. Mostly used are auditory models that decompose the signal into subbands and periodicity analysis is performed within these bands [57].

Iterative Algorithms: Many systems process the polyphonic signal iteratively using monophonic pitch estimation algorithms to find multiple notes at a time. The spectrum of a detected note, or predominant F0, is estimated and subtracted from the signal and the F0 estimation process continued on the residual, as in [12], [37] or [38].

Classification based Pitch Estimation: Classification based music transcription has been applied using different classifiers. Support Vector Machines (SVM) are used in [55] or [63] for frame level pitch classification. In [72], [62] and [67] Hidden Markov Models (HMM) are used to consider the temporal structure of a note and classify sequences of features, or a combination of both, SVM and HMM is proposed in [56] for the classification of sequences of framewise classification results.

1.4.4 Onset Detection

The most audible and most import time information of a note played is the onset. Thus, estimating the time structure of a note mainly focused on onset detection. While the onset is clearly defined and detectable for each

instrument, the offset is not, because no abrupt transition of signal properties exists between the steady and the release part of a note.

The task of estimating the evolution of a note over time is often referred to onset detection, while both, onset and offset detection are analysed. Again, onset detection in the polyphonic case is harder than in monophonic case, due to the overlapping notes, and smooth, slowly evolving onsets are more difficult to detect, than sharp, explosive onsets.

The methods proposed can be divided in two main groups, energy or amplitude distribution based onset detection and pitch based onset detection. The former can be expressed by the change of the spectral energy distribution (*Spectral Flux*) [35] and the latter by the change of signal phase [5], or using the outcome of a pitch estimator [10]. While the former is suitable for the detection of strong onsets, the latter has advantages in the case of weak onsets.

The combination of these two approaches leads to onset detection in complex domain [16], using magnitude and phase continuity conditions.

1.4.5 Top-down Processing

Instead of the bottom-up, or data-driven approaches described so far, where all information flows from the bottom (the acoustical waveform) through higher level processes to top (the musical notation), top-down, or prediction driven approaches use prior knowledge of the properties of musical signals, similar to the human auditory system.

Of course, top-down processing still needs the information obtained through the bottom-up processing techniques, but can use dependencies to solve otherwise ambiguous situations, and is meant to be necessary to achieve good transcription results [65].

Chafe *et al.* [9] 1982 developed the first system making use of high-level information, such as key, beat and metric structure extracted from the low level musical signal. In 1985 Chafe *et al.* proposed the first blackboard system that repeatedly processes music with modules for signal transformation, note modelling, event detection and meter estimation.

Blackboard systems utilizing *expert* systems with certain knowledge are widely used to combine the bottom-up features in a prediction driven way, creating additional information and enhancing the performance of AMT, e.g. [33], [4] or [36]. These experts may carry knowledge of tones, auditory models, chords, intervals, musical key or rhythm.

A speech recognition motivated *musicological* model defining note transition

probabilities is proposed in [62] using Hidden Markov Models.

Complete top-down processing is recently applied using genetic algorithms [61], generating rough pitch estimations, rendering these estimates into audio signals through a synthesizer and iteratively optimizing the difference between the audio signal and the target signal.

1.5 Human Performance on Music Transcription

Music transcription is a task that can currently only be done by musical experts: people with a strong musical background and knowledge. While this remains a very time consuming task for these people, it is nearly impossible to be done by people without such a background.

The challenge for automatic transcription systems is to keep up with human transcription performance in terms of speed and accuracy. While the former can be achieved quite easily with current systems, the latter is still far from solved satisfactorily. Thus, the few commercially available transcription system tend only to support the musician in his time consuming work, and not to provide a 100% accurate transcription.

In [41], Klapuri *et al.* compare their proposed transcription system with the performance of trained musicians on signals with polyphony ranging from two to five.

Only intervals were asked to transcribe, no absolute pitch, and the number of co-occurring sounds was given beforehand. So this was only a listening test that does not necessarily adapt the way a musician would transcribe a song, using his instrument and musical knowledge to transcribe parts of a song step by step and repeatedly.

The two best skilled of the ten musicians were able to outperform the AMT system, especially in higher frequency regions, while the system had a better performance in very low frequency regions.

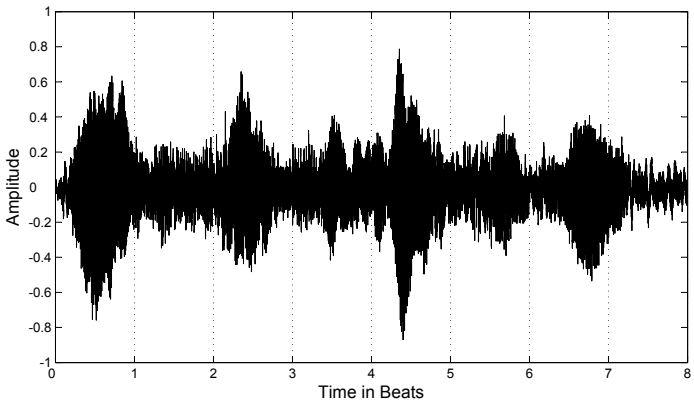
1.6 Defining Work and Thesis Organization

In contrast to other systems, this thesis focuses on real-world music without any restrictions on the input material. Thus, music of all genres is processed, containing arbitrary instruments including drums, unknown polyphony, tempo and meter.

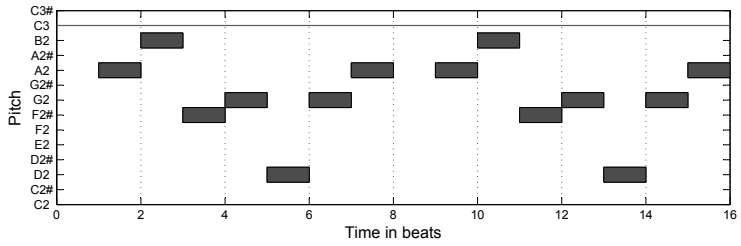
Instead of creating a traditional score shown in Figure 1.1 out of the music recording, this thesis is constrained on the creation of a so called pianoroll notation, seen in Figure 1.2(b).

To achieve this, each note has to be defined with the distinct parameters: pitch, onset and duration.

Only note-level parameters are estimated, no song-level, but the latter may be used to enhance the accuracy of the former.



(a) Waveform



(b) Pianoroll notation

Figure 1.2: Exemplary waveform and pianoroll notation of a musical recording.

In Chapter 2, the temporal and spectral representations suitable for musical signals are presented. The actual methods of music transcription, using these representations, are described in Chapter 3, namely multiple F0 estimation, onset and duration detection, as well as the use of musical knowledge in the form of key and meter information. Finally, in Chapter 4 the results of the proposed transcription methods are given together with a discussion. Chapter 5 closes with a summary and an outlook on possible future work and enhancements to the transcription system.

Chapter 2

Proposed Music Signal Representations

Prior to the estimation of the fundamental parameters of each note in a musical recording, representations of the spectral and temporal properties of the signal have to be found, that do enhance the musical features.

Therefore, some time-frequency representation of the signal must be found and the harmonic parts of the signal, caused by the voiced instruments must be separated from all nonharmonic, unvoiced parts, which are created by all nonharmonic instruments, like drums, and are in the following refered to as *noise*.

In addition, some temporal representation is needed, that is able to describe the evolution of the signal over time.

2.1 Properties of Music Signals

The most common way of instrument tuning in modern western music is the twelve-tone equal temperament, which divides the octave into 12 (logarithmically) equal parts - semitones - with equal frequency ratios.

With an octave step being defined as doubling the frequency, or a frequency ratio of two, the frequency ratios r of adjacent semitones are

$$r = \sqrt[12]{2} \approx 1.059 \tag{2.1}$$

or roughly 6 %. This is sometimes defined as 100 *cent* with *cent* being the hundredth of a semitone.

In the following, midi note numbers [1] are used to clearly define each semi-

piano key	midi	Helmholtz	Scientific	F0 [Hz]	notes
88	108	c''''	$C8$	4186.01	highest note
87	107	b'''	$B7$	3951.07	
⋮	⋮	⋮	⋮	⋮	
49	69	a'	$A4$	440.00	reference tone
⋮	⋮	⋮	⋮	⋮	
2	22	$B♭,,$	$B♭0$	29.14	
1	21	$A,,$	$A0$	27.50	lowest note

Table 2.1: Naming schemes and frequencies of some notes playable on a 88 key piano.

tone. The MIDI specification defines the *middle C* as midi note number 60, and all other notes are relative. Thus, the reference pitch A4 with 440 Hz is midi note number 69. Out of this relation, midi note numbers for all other notes with fundamental frequency F_0 are computable through

$$\text{MIDI note number} = 69 + 12 \cdot \log_2(\frac{F_0}{440\ Hz}) \tag{2.2}$$

Given the midi note number n, the fundamental frequency is found through

$$F_0 = 440\ Hz \cdot 2^{\frac{n-69}{12}} \tag{2.3}$$

While not limiting the frequency range of the notes transcribable by the system, the main focus is layed on the note range playable by an 88 key standard piano ranging from $A0$ to $C8$ in scientific notation. There are some wind instruments, that exceed this note range, but notes outside of this range do occur rarely and are not considered.

The main problem resulting from this logarithmic frequency scale is obvious in table 2.1. The frequency ratio of roughly 1.06 between adjacent semitones leads to very small absolute frequency differences of notes on the lower end of the piano keyboard, namely $\Delta f = 1.64\ Hz$ between middle frequencies of $A0$ and $B♭0$.

Using some linear time-frequency representation, this would lead to an unacceptable low time resolution. Assuming some common moderate tempo of 120 beats per minute (bpm) in duple meter $\frac{4}{4}$ the period time Δt results for quarter notes in

$$\Delta t = \frac{60\ sec/min}{120\ 1/min} = 0.5\ sec\ . \tag{2.4}$$

Assuming sixteenth notes, this leads to a necessary time resolution $\Delta t = \frac{0.5\ sec}{4} = 125\ ms$. Still, 120 *bpm* is a moderate tempo, with tempi of 180 *bpm* being used in classical concert music and tempi above 200 *bpm* being commonly used in modern pop and dance music.

The challenge of the time frequency method used for the representation of musical signals, is to allow both, necessary time and frequency resolution at the same time.

2.2 Spectral Representation

The designated time and frequency resolutions found in Section 2.1 are not achievable by methods of time-frequency representation working on a linear frequency scale. Thus, alternative transformations have to be used, which match the spectral properties of the musical signal.

2.2.1 Frequency Transformation

Short Time Fourier Transformation

The commonly used time frequency representation (TFR) is the Short-Time Fourier Transform (STFT), which cuts the signal into time-frames and calculates the Discrete Fourier Transform (DFT) on each time-frame to obtain a local stationary time-frequency analysis of nonstationary signals. The length of the analysis window specifies the resulting time and frequency resolution: Long windows lead to high frequency but low temporal resolution, and vice versa. The time-frequency resolution uncertainty principle [48] limits time resolution Δt and frequency resolution $\Delta\omega = 2\pi\Delta f$ to

$$\Delta\omega \cdot \Delta t \geq \frac{1}{2}\ . \tag{2.5}$$

The Short Time Fourier Transform $X_l[k]$ of a discrete sequence of data samples $x[n]$ for each frame l is defined as:

$$X_l[k] = \frac{1}{M}\sum_{n=0}^{M-1} x[n + l \cdot L] \cdot w[n] \cdot e^{-j2\pi k \frac{n}{M}}\ , \tag{2.6}$$

with

$$l = 0, 1, ... \ \text{ and } \ k = 0, 1, ..., N-1\ , \tag{2.7}$$

and the STFT length N, the hop-size L, the time-window size M.

$w[n]$ is a Hamming window to reduce the leaking effect:

$$w[n] = 0.54 + 0.46 \cdot \cos\left(\frac{2\pi n}{M}\right),\ n = -\frac{M}{2}, ..., \frac{M}{2} \tag{2.8}$$

The frequency and time resolution are controlled by the window parameters and the sampling frequency F_s of the signal:

$$\Delta f = \frac{F_s}{M}\ , \tag{2.9}$$

$$\Delta t = \frac{M}{F_s} \tag{2.10}$$

The DFT can be efficiently computed through the Fast Fourier Transform (FFT) algorithm.

Multi-Resolution Fourier Transform

To bypass the time-frequency resolution constraints especially in lower frequency regions, Multi-Resolution Fourier Transformations (MRFT) can be used. Using different lengths of transformation windows, the frequency resolution can be matched to the frequency region, allowing both, a sufficient frequency resolution and the best possible time resolution in each frequency range.

According to Equation 2.6 there are two ways to achieve this composition of different resolutions: Either through altering the windows size M and the DFT length N together, or just changing N alone, keeping the window size unchanged and using zero-padding to align window size and DFT length.

While the former leads to accurate results, it is computationally less efficient. The latter uses zero-padding that was found to not enhance spectral resolution, because the window length is shorter than a time period of the signal, but introduces an interpolating (smoothing) effect [71] and allows fast computation.

Here, different window sizes are used, to avoid zero-padding and to obtain enhanced frequency resolution, resulting in different time resolutions in different parts of the spectrum.

The lower time resolution in lower frequency regions does not necessarily harm, because of the more stationary character of the signal, in comparison to the rapid changing signal in high frequency regions.

Three different resolutions are used for signals with a sampling frequency $F_s = 44.1\ kHz$: All frequency regions above $220Hz$ of $A3$ are transformed

using a window length $M = N = 4096$ samples and a hop size $L = 2048$ samples, or a overlap between adjacent windows of 50%. This leads to a frequency resolution that allows to discriminate adjacent semitones and at the same time maintains a time resolution that fulfills the precondition found in Section 2.1.

For the lower frequency regions windows with length $M_1 = 2 \cdot M$ and $M_2 = 4 \cdot M$ are used, together with a hop size of $L_1 = \frac{L}{2}$ and $L_2 = \frac{L}{4}$ to assure the same frame rate in each frequency region.

The boundary between these two is $220/2\ Hz = 110\ Hz \cong A3$. While the maximum window size of 16384 can still not resolve the lowest semitones, this was chosen due to computational reasons and the rather low occurence of these notes. The harmonics of fundamental frequencies in this frequency range are clearly distinguishable using this resolution.

The multiresolution fourier transform allows adaptive time frequencies resolutions in different frequency regions, but can not bypass the problems arising from the limitations of the linear fourier transform. In contrast to a constant Q wavelet transform, the multiresolution approach proposed here allows to adjust the time-frequency resolutions as essential for each frequency region. The window lengths used, favour the time resolution over the frequency resolution and the higher frequency resolution is only used were absolutely necessary.

In the following chapter, a method is described that allows more accurate estimation of the actual frequency in each frequency bin, and is able to weaken the effect of low frequency resolution.

Instantaneous Frequency through Phase Derivation

The Instantaneous Frequency (IF) refers to the actual effective frequency of a sinusoidal component and makes use of the derivative of the phase to define it. While the methods of TFR describe the amount of signal energy in a frequency *range*, the instantaneous frequency estimates the most prominent frequency component within this range.

Estimating the instantaneous frequency of the signal does not lead to a nominal higher frequency *resolution*, but allows to estimate the actual frequency of each bin more precisely. The overall grid size that defines the discrimination of adjacent frequency components is still defined by the resolution of the underlying transformation.

Different methods exists to estimate the instantaneous frequency (IF) and magnitude from fourier coefficients. Some are compared in [32] and those,

which base on the phase information of the FFT were found to give the best results regarding frequency resolution.

Here, the phase vocoder method [19] is used to estimate the instantaneous frequency $f_i[k]$ using the phase difference $\Delta\phi[k]$ between adjacent spectra obtained through the FFT:

$$f_i[k] = (k + \kappa[k]) \cdot \frac{F_s}{N} \ . \tag{2.11}$$

Where $\kappa[k]$ is the frequency offset within each bin and is related to the phase difference through

$$\kappa[k] = \frac{N}{2\pi L} \cdot \text{princarg}\left(\phi_l[k] - \phi_{l-1}[k] - \frac{2\pi L}{N}k\right) \ . \tag{2.12}$$

princarg is the principal argument function, that maps the phase to the $\pm\pi$ range. With the maximum bin offset being $\pm\frac{N}{2L}$, the instantaneous frequency of each bin is detected in the range of the two adjacent frequency bins for an overlap $L = 0.5 \cdot N$. This method does only work if overlapping STFT windows are used, to confine $|\Delta\phi| < 2\pi$.

The frequency of each bin is thus, shifted away from the middle frequency of this bin. As the windowed time domain signal results in windowed magnitude in the frequency range represented by each bin, shifting the middle frequency of the bins necessitates a correction of the magnitude of the peak.

The instantaneous magnitude $X_i[k]$ is computed from the magnitude of its bin $|X[k]|$ through

$$X_i[k] = \frac{1}{2}\frac{|X[k]|}{w\left(\frac{N}{M}\kappa[k]\right)}, \tag{2.13}$$

where w is again the Hamming window function defined in Equation 2.8 and used in the STFT.

The magnitude of each bin is corrected and the bin assigned to the respective semitone according to the estimated IF, instead of the middle frequency of the bin.

This method does only work for sinusoidal peaks with stationary frequency and amplitude and not for noise components of the signal [13]. As music transcription only relies on the harmonic signal parts in spectral peaks, this limitation is acceptable.

Although the instantaneous frequency allows the estimation of the frequency of each bin more precisely, it can not enhance the actual frequency resolution of the TFR, because only one frequency is estimated for each bin.

It has to be guarantied that there is only one sinusoidal, or spectral peak from a harmonic partial in each bin at one time. Multiple partials in one bin can not be separated and would lead to incorrect results of the estimated IF. So the frequency resolution is still given by the underlying time-frequency transformation.

Together with the actual multiresolution frequency estimation in each bin, the rather low frequency resolution of the STFT in the lowest octaves can by bypassed, as long as no direct neighbouring notes are sounding together.

2.2.2 Spectral Preprocessing

The task of the preprocessing step is to separate the complex music signal composition into harmonic (voiced) and nonharmonic (unvoiced) or noise parts and match the spectral distribution of energy to the sensitivity of the human ear.

The input signals are converted to *mono* signals, with a sampling frequency $F_s = 44.1\ kHz$, if necessary.

dB(A) Filtering

To model the frequency characteristics of the human ear, the spectrum of the signal is weighted using the A filter $W_A(f)$ [29] that takes the spectral sensitivity of the human ear into account. Especially low frequencies are enhanced during the recording and mastering process of a musical recording to overcome this low sensitivity of the ear, as shown in Figure 2.1. Without this correction, these frequency ranges would be relatively too loud.

$$W_A(f) = \frac{12200^2 \cdot f^4}{(f^2 + 20.6^2) \cdot (f^2 + 12200^2) \cdot \sqrt{f^2 + 107.7^2} \cdot \sqrt{f^2 + 737.9^2}} \tag{2.14}$$

RASTA Processing

Noise suppression is a task, widely studied in terms of speech processing. While noise in speech signals is normally a more or less stationary background noise, that can be estimated by statistics of the signals over a period of time [47], the term "noise" in musical signal refers to nonstationary signal parts that are unvoiced, nonharmonic and thus, do have a wide bandwidth.

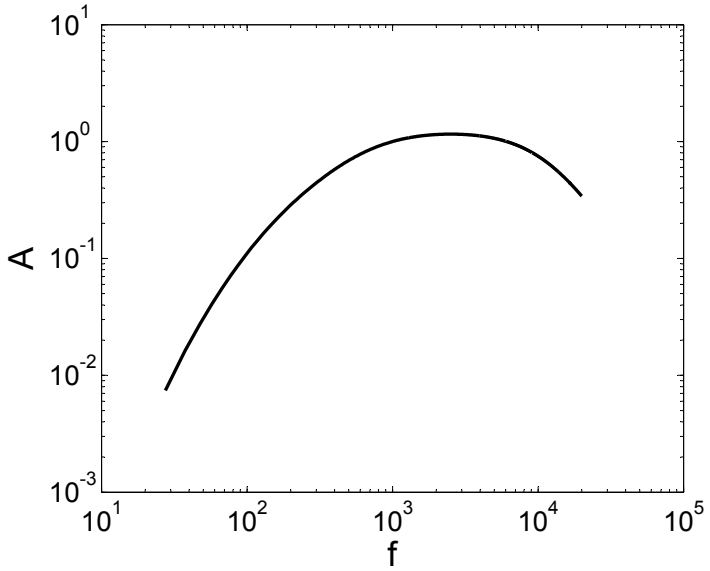


Figure 2.1: dB(A) weighting of the frequency components f in Hz, according to IEC 61672-1 Ed. 1.0 [29] to model the sensitivity characteristics of the human ear.

These transient signal parts are mostly caused by percussion or drums, but even slower onsets from voiced instruments - like piano - or the noise caused by the air in wind instruments produce some nonharmonic signal parts.

While these nonharmonic signal parts are important for meter and onset detection, they hinder the estimation of the harmonic structure of the signal and thus, the estimation of the fundamental frequency, where only the harmonic components, or spectral peaks are meaningful.

Relative spectra (RASTA) processing has been proposed by Hermansky and Morgan [25] for noise suppression in speech signals and has been successfully applied to music signals by Klapuri *et al.* [42].

The environmental influences on a signal are modeled by a set of transformations: convolution with the environmental impulse response and the addition of additive noise.

The signal of the vibrating system $S[k]$ in spectral domain is altered by convolutive noise through the impulse response $H[k]$, which is defined by the frequency response of the environment and the body of the musical instrument, and additive noise $N[k]$ resulting in the observed magnitude spectrum $X[k]$:

$$X[k] = S[k] \cdot H[k] + N[k] \ . \tag{2.15}$$

The convolutive noise does alter the spectral distribution of the sound and its suppression converts it back to the linear spectrum of the vibrating system. This is also called as *spectral whitening* and similar approaches have been used in [34], [66] and [38] for preprocessing of musical signals.

The nonstationary additive noise is caused by unvoiced instruments and is defined by all nonharmonic signal components and has to be estimated and removed for each single time-frame.

RASTA processing removes both, convolutive and additive noise simultaneously by transforming $X[k]$ into a logarithmic scale magnitude spectrum $Y[k]$

$$Y[k] = \ln(1 + J \cdot X[k]) \ , \tag{2.16}$$

where J is a signal dependent positive constant that scales additive noise to be $N[k] < 1$ and the harmonic spectral peaks to be $H[k] \cdot S[k] > 1$. The transform is linear-like for $J < 1$ and logarithmic like for $J >> 1$. Thus, J should depend on both, the additive noise and the spectral peaks, i.e. the harmonics of the signal.

J is chosen to be

$$J = \frac{k_{max} - k_{min} + 1}{\sum_{k=k_{min}}^{k_{max}} X[k]} \ , \tag{2.17}$$

with k_{max} and k_{min} denoting the frequency bins belonging to the frequencies of lowest and highest note on 88 key piano keyboard, $A0 = 27.5\ Hz$ and $C8 = 4186\ Hz$, respectively. Thus, J denotes the inverse of the average spectrum in frequency regions important for musical signals and scales noise like values of $X[k]$ to $J \cdot X[k] < 1$ and spectral peaks with $X[k]$ greater than the mean value to $J \cdot X[k] >> 1$. By taking the logarithm, spectral peaks are enhanced while low, noise-like values do remain nearly constant.

The additive noise component $N'[k]$ of $Y[k]$ is then subtracted in each frequency bin k by calculating a moving average in octave wide windows $W(k_n, k)$ on the frequency scale. Octave wide bands are chosen instead of ERB, Bark critical-band or linear frequency scale to follow the logarithmic scale of musical notes [74].

$$N'[k] = \frac{\sum_{k_n=2/3k}^{3/2k} Y[k_n] \cdot W[k_n]}{3/2 \cdot k - 2/3 \cdot k} \ , \tag{2.18}$$

with a simple rectangular window $W[k_n]$

$$W[k_n] = \begin{cases} 1 & \text{for} \quad \frac{2}{3} \cdot k < k_n < \frac{3}{2} \cdot k \\ 0 & \text{otherwise} \ . \end{cases} \tag{2.19}$$

The logarithmic scale of the octave wide averaging windows ensures that an equal amount of notes and thus, spectral fine structure of harmonic peaks are taken into account. In lower frequency ranges, the length of the averaging window was fixed to be at least 250 Hz wide.

The estimated noise spectrum $N'[k]$ of each frequency bin is then linearly subtracted from the logarithmic magnitude spectrum $Y[k]$

$$Z[k] = Y[k] - N'[k] \tag{2.20}$$

The last step of RASTA processing is to convert the logarithmic magnitude spectrum with enhanced spectral peaks and subtracted additive noise $Z[k]$ back into a noise suppressed linear magnitude spectrum $\tilde{X}[k]$

$$\tilde{X}[k] = \frac{e^{Z[k]} - 1}{J} \ . \tag{2.21}$$

Resulting values of $\tilde{X}[k] < 0$ are set to

$$\tilde{X}[k] = \max[0, \tilde{X}[k]] \ . \tag{2.22}$$

This noise suppressed linear magnitude spectrum is used to create the spectral representation of the music transcription system.

2.2.3 Prominent Frequency Detection

Prior to mapping the frequency components to each semitone, a prominent frequency detection is used to correct detuned musical recordings.

A single FFT is applied to get the spectrum of the whole signal. The prominent frequency is assumed to be the frequency bin with maximum amount of signal magnitude, in the note range $A0 - C8$ and is assigned to the center frequency of the nearest semitone.

A *tuning factor* T is calculated using the original middle frequency of the nearest note F_0 and the detected prominent frequency f_{prom}

$$T = \frac{f_{prom}}{F_0} \ . \tag{2.23}$$

This tuning factor is used to retune each notes middle frequency F_0 to the corrected middle frequency $F_0' = F_0 \cdot T$.

2.2.4 Semitone Spectrogram

The short-time multiresolution fourier transform together with the estimation of the instantaneous frequency of each bin and the suppression of wideband noise caused by nonharmonic signal parts through RASTA preprocessing leads to an effective low-level spectral signal representation of the harmonic parts of the signal.

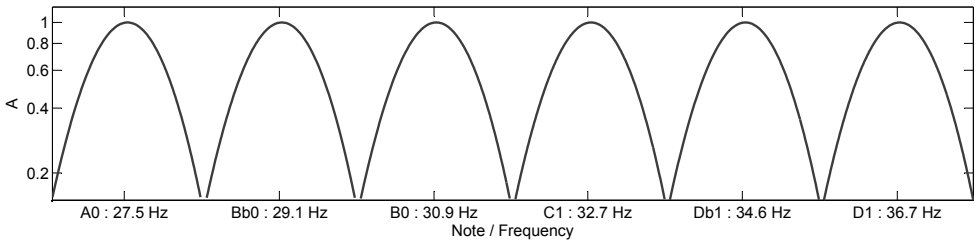


Figure 2.2: Mapping of spectral components to the corresponding semitone, using narrowed gaussian windows.

To further adapt this representation for the recognition of musical notes within this signal, a representation is used which applies the logarithmic properties of the semitone based equal temperament western musical scale to create a semitone band spectrogram, which models the human perception of pitch [6].

Following the musical note structure, a subband spectrogram is created with a subband for each semitone. The middle frequency F_0 of each subband equals one musical note (see Table 2.1).

The bandwidth $f_{max} - f_{min}$ of each subband equals the corresponding semitone limits:

$$f_{max} = 1.03 \cdot F_0 \tag{2.24}$$

$$f_{min} = \frac{1.03}{1.06} \cdot F_0 \tag{2.25}$$

The range of bins k with $k_{min} < k < k_{max}$ is found using the estimated instantaneous frequency of each bin through $f_{min} < f_i[k] < f_{max}$.

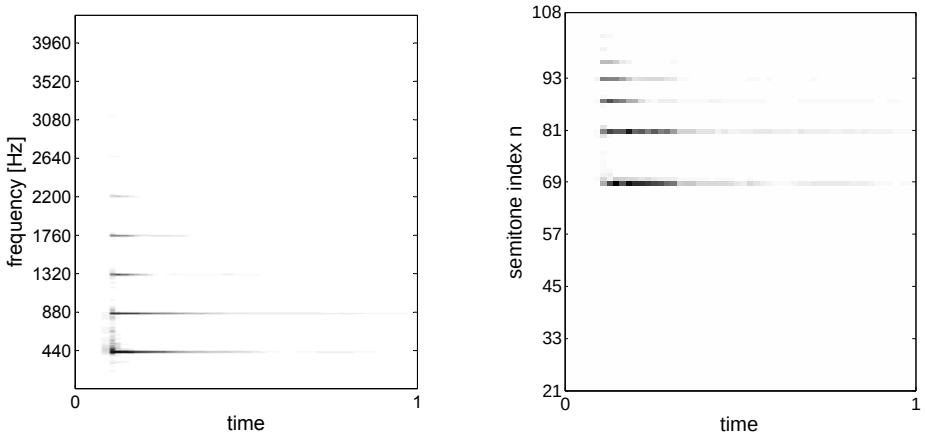
For each midi note $n = 21, ..., 108$ playable on the standard piano keyboard $A0 - C8$, the subband magnitude spectrum is computed:

$$Z[n] = \sum_{k_{min}[n]}^{k_{max}[n]} X_i[k] \cdot W_{Gauss,n}[k],\ n = 21, ..., 108\ , \tag{2.26}$$

with $W_{Gauss,n}[k]$ being a Gaussian Window function

$$W_{Gauss,n}[k] = e^{-\frac{1}{2}\left(\alpha \frac{k}{K/2}\right)^2} , \tag{2.27}$$

with the overall number K of frequency bins assigned to the semitone n and α the reciprocal of the standard deviation. With $\alpha = 3$, a narrowed window is used to map the linear frequency contents to each semitone as shown in Figure 2.2.



(a) Linear Spectrogram (b) Semitone Spectrogram

Figure 2.3: Comparison of a linear and semitone spectrogram of a piano A4 onset. Note, that both representations cover the same frequency range: roughly 27.5 to 4185 Hz, i.e. the range of semitone indexes 21 to 108.

The advantage of the logarithmic semitone representation of the spectral components of a musical signal are shown in Figure 2.3. Note, that there are four octaves with 48 notes below the A4 with 440 Hz shown. Most of the frequency information in the linear spectrogram is assigned to frequency ranges that are less important for the representation of musical signals, while especially the low frequency regions are underrepresented.

The semitone spectrum assigns each note the same ratio and thus, leads to a equally spaced representation adjusted to the logarithmic properties of the music signal.

2.3 Temporal Representation

The temporal structure of each note being played is, after the pitch, the second parameter needed to fully determine the parameter representation of each note. Thus, it is necessary to find a method to represent the progress of the signal over time. Two actual moments are to be defined: notes on- and offset.

Although each instrument has its own, specific temporal properties, the common structure is assumed to be the same for any note played by any instrument. The Attack, Decay, Sustain, Release (ADSR) Model describes this general structure of a note shown in Figure 2.4, by defining specific states.

While the attack state, right after the onset of a note, is characterized by increasing signal amplitude and thus, leads to transients, and the release-state, right after the offset of the note, can be found in each instrument's temporal slope, the decay and sustain state may be less pronounced or even not present at all.

Using this representation, the signal can be divided in different parts: transients (attack, decay and release) with changing amplitude and spectral instability, and steady-state (sustain) regions, with constant amplitude and a stable harmonic spectrum with clear discriminable partials.

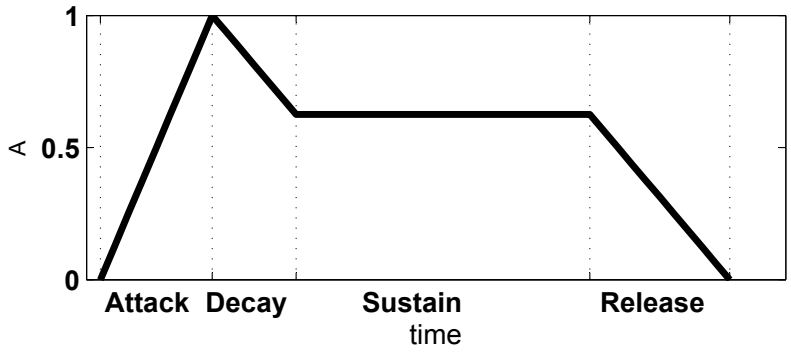


Figure 2.4: Amplitude based *Attack, Decay, Sustain and Release Model* of the temporal structure of a note.

Simple note onset detection methods use energy based approaches to detect notes' on- and offsets. While this leads to good results with hard onsets of notes, it is limited in soft onsets, i.e. caused by strings.

Phase based approaches to onset detection [5] showed to be superior for soft, less percussive onsets but still, the energy based methods show better results

for hard onsets.

Combinations of these two methods can eliminate the drawbacks of both [15]. This is carried a step further in [16], where onset detection based on phase and energy information is applied using a signal representation in complex domain.

2.3.1 Frequency Transformation

To model the temporal structure of the signal, a high time resolution is more important, than a high frequency resolution. Thus, a single resolution DFT, see Section 2.2.1, with a window length of $N = 4096$ with 50% overlap between adjacent frames is used, resulting in the complex spectrum $X_k[l]$ for each frame l.

Again, the instantaneous frequency described in Section 2.2.1 is used, to enhance the accuracy of the frequency estimation of each bin k.

2.3.2 Envelope based Signal Modeling

The onset of a note leads always to an increase in signal energy. Strong percussive note attacks show abrupt energy accent, while the energy does remain roughly constant in locally steady-state regions.

Using the first order derivative of the signal envelope, this positive change in signal energy will lead to a peak. Using the time-frequency representation of the STFT in Equation 2.6, the derivative can be bandwise calculated for the magnitude of each frequency bin k:

$$\Delta|X_k[l]| = |X_k[l]| - |X_k[l-1]| \ , \tag{2.28}$$

to define a stationarity criterion within this actual frequency region.

2.3.3 Phase-based Signal Modeling

The Fourier transform states that a signal is consisting of a group of sinusoidal oscillators with time-varying magnitudes and phases.

During the sustain- or steady-state of a note the signal is assumed to have constant amplitude and frequency, while in transient parts, both of them do change.

As constant frequency leads to a constant slope of the unwrapped phase $\tilde{\phi}$, assuming stationarity of the signal, the expected phase $\tilde{\phi}_k(l)$ at a given time

frame l in a frequency bin k can be predicted through extrapolating the phase of previous time frames:

$$\tilde{\phi}_k(l) = \tilde{\phi}_k(l-1) + \left[\tilde{\phi}_k(l-1) - \tilde{\phi}_k(l-2)\right]. \tag{2.29}$$

This results in a phase difference $\Delta\phi$ between the predicted target phase and the actual phase at frame l of

$$\Delta\phi = \text{princarg}\left[\tilde{\phi}_k(l) - 2 \cdot \tilde{\phi}_k(l-1) + \tilde{\phi}_k(l-2)\right] . \tag{2.30}$$

Again, *princarg* is the principal phase argument, that maps the phase difference to the $\pm\pi$ range.

$\Delta\phi$ will tend to be small for stationary signals, where the phase is accurately predicted using the preceding values and will derivate from zero for transient parts of the signal, resulting in a pronounced peak.

2.3.4 Complex Domain Signal Modeling

Combining the constant properties of frequency and amplitude in locally steady-state regions of the signal, a robust method is created that is able to model both, fast percussive changes, as well as smooth onsets.

This can be achieved by simultaneously predicting values for the magnitude and phase in the complex signal domain.

In polar notation, the complex value of the actual signal $X_k(l)$ is defined by the magnitude $|X_k(l)|$ and phase $\phi_k(l)$ in each bin k for each frame l through

$$X_k(l) = |X_k(l)| \cdot e^{j\phi_k(l)} . \tag{2.31}$$

The target signal $\hat{X}_k(l)$ is estimated through the predicted magnitude $|\hat{X}_k(l)|$ and phase $\hat{\phi}_k(l)$ according to Equations 2.28 and 2.29 using preceding frames:

$$\hat{X}_k(l) = |\hat{X}_k(l)| \cdot e^{j\hat{\phi}_k(l)} . \tag{2.32}$$

The distance between the current and predicted signal can then be expressed by measuring the Euclidean Distance $\Gamma_k(l)$ for each bin k in each time frame l in the complex domain

$$\Gamma_k(l) = \left(\left[\Re(\hat{X}_k(l)) - \Re(X_k(l))\right]^2 + \left[\Im(\hat{X}_k(l)) - \Im(X_k(l))\right]^2\right)^{\frac{1}{2}} , \tag{2.33}$$

with $\Re$ and $\Im$ denoting the real and imaginary part of the complex signals as shown in Figure 2.6.

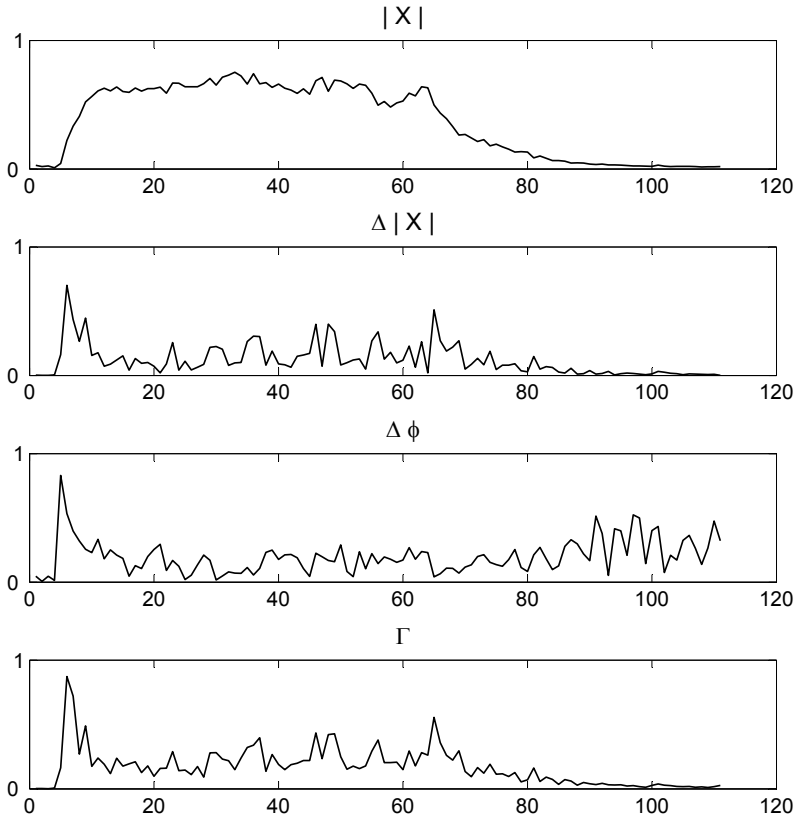


Figure 2.5: Comparison of the different modelling methods on a Cello B2 signal, whose envelope is shown in the first graph.

The Euclidean Distance $\Gamma_k(l)$ does quantify the stationarity of the signal for each frame l and bin k. Small distances between the predicted and actual signal do indicate small temporal changes of the signals frequency and amplitude and thus, stationary signal parts. Large distances are an indication for large changes of the signal properties and thus, for transient signal parts.

Figure 2.5 shows the slope of the difference between the predicted and actual magnitude, phase and the resulting difference in complex domain. All of them show a distinct peak at the beginning of the onset, while it is most pronounced in the complex distance. Of course, for this rather strong onset, the phase method is can not offer any advantage over the amplitude based method.

Instead of summing all distance values in each frame to form a global detection function $\eta(l)$

$$\eta(l) = \sum_{k=1}^{K} \Gamma_k(l) \ , \tag{2.34}$$

bandwise detection functions $\eta_l[n]$ are created for each midi note n similar to the creation of the semitone spectrogram defined in Equation 2.26 by summing

$$\eta_l[n] = \sum_{k=k_{min}[n]}^{k_{max}[n]} \Gamma_l[k] \cdot W_{Gauss,n}[k], \ n = 21, ..., 108 \tag{2.35}$$

over all bins k in the frequency range of each note defined by f_{min} and f_{max} in Equation 2.25 and determined by the instantaneous frequency of each bin.

Due to favouring the time resolution for accurate modeling of the temporal properties, the IF is very important here to allow accurate alignments despite the low frequency resolution.

As $\eta_l[n]$ is a bandwise measure of the overall *transiency* of the signal in band n at a time frame l, the local stationarity, or *steadiness* $\varsigma_l[n]$ of the signal can be described through the inverse of $\eta_l[n]$:

$$\varsigma_l[n] = (\eta_l[n])^{-1} \ . \tag{2.36}$$

To further separate the transiency description of a signal into signal parts with rising or descending magnitude the calculation of $\Gamma_k(l)$ can be enhanced to

$$\Gamma_k(l)^+ = \begin{cases} \Gamma_k(l) & \text{for} \quad |X_k(l)| > |X_k(l-1)| \\ 0 & \text{otherwise} \end{cases} \tag{2.37}$$

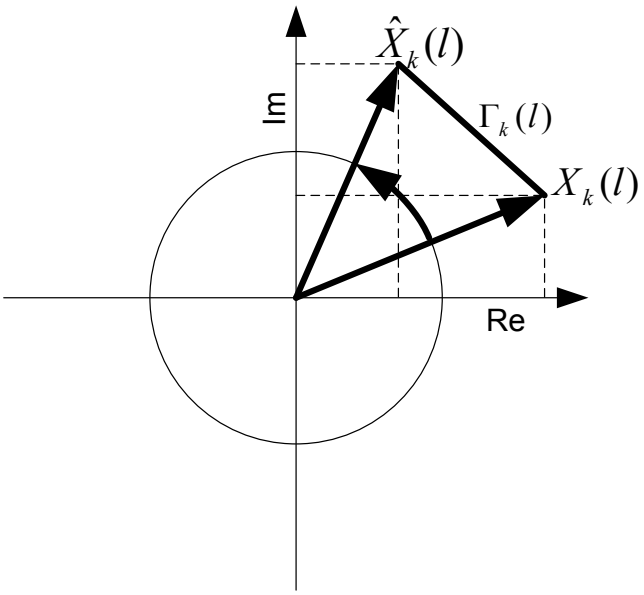


Figure 2.6: Euclidean Distance between real and imaginary parts of actual $X_k(l)$ and predicted $\hat{X}_k(l)$ signal in complex phase diagram.

$$\Gamma_k(l)^- = \begin{cases} \Gamma_k(l) & \text{for} \quad |X_k(l)| < |X_k(l-1)| \\ 0 & \text{otherwise} \end{cases} \tag{2.38}$$

with $\Gamma_k(l)^+$ and $\Gamma_k(l)^-$ denoting the euclidean distances as defined by Equation 2.33 of signal parts with rising and descending magnitude, respectively. Using Equation 2.35, bandwise rising $\eta_l[n]^+$ and descending $\eta_l[n]^-$ transient signal parts can be described for each semitone band n, while $\eta_l[n]^+ + \eta_l[n]^- = \eta_l[n]$.

Chapter 3

Proposed Methods of Music Transcription

The transcription of music signals includes at least two major steps: The estimation of the pitch, i.e. the fundamental frequency (F0) of each note being played, and the determination of the beginning (onset) and end (offset) of each note. The latter two are often combined and referred to as the *onset detection* task, because the beginning of the note is considered to be more important for human perception of music.

These two steps are necessary to define the note-level parameters of a musical piece.

3.1 Multiple F0 Estimation

The most important, but still not solved satisfyingly step in transcription of polyphonic real-world music is the multiple fundamental frequency estimation. While there do exist some widely accepted reference methods for F0 estimation in monophonic signals like speech, most of them working in time-domain, the polyphonic case necessary for music signals is still a subject of research and reference methods do only exist for partly transcription under narrowed conditions.

The F0 estimation process consists of several subtasks that have to be dealt with, to achieve a reliable estimate:

1. Harmonic partials detection.

2. Polyphony estimation.

3. Multiple F0 estimation using the harmonic structure of the harmonic partials.

Several problems can occur in the harmonic structure of the signal that have to be addressed by the F0 estimation process in order to allow reliable F0 detections:

- Missing fundamental frequency.
- Missing harmonics.
- Shared, overlapping components between different notes.

The method used here is based on a conceptually simple and computationally efficient method proposed by Klapuri [38] that is based on the summation of the amplitudes of the harmonic components of each possible F0. The method showed good results, even when compared with much more complex and computationally less efficient methods based on auditory models.

Iterative estimation and cancellation approaches are commonly used ([39], [37]) to find multiple F0 estimations in each time frame, using a monophonic pitch estimator. Signal properties are used to estimate the spectrum of a found F0 candidate and subtract it from the signal. The F0 estimation process is then continued on the residual of the signal.

3.1.1 Silence Detection

The term silence detection does here not only refer to the detection of true silent time frames of the signal but to the detection of silent and unvoiced signal parts. Thus, only harmonic signal parts are used to calculate the frame energy and mark a frame l as silent using a signal to noise ratio (SNR) like representation with the local steadiness $\varsigma_l[n]$ defined in Section 2.3.4 of each component n of the signal in each time frame l of the semitone spectrogram $Z_l[n]$ defined in Section 2.2.4:

$$SNR_l = 10 \log \left(\frac{\sum_{n=21}^{127} (Z_l[n]^2 \cdot \varsigma_l[n]^\sigma)}{\max_l (\sum_{n=21}^{127} (Z_l[n]^2 \cdot \varsigma_l[n]^\sigma))} \right) , \tag{3.1}$$

with $\sigma > 1$ to enhance the influence of the steadiness. The semitone index n ranges to 127 to cover the harmonics of midi note number 108.

This ensures, that no F0 is estimated in silent time regions and additionally prevents noisy signal parts, without any harmonic structure to avoid the

silence detection. A minimum of signal energy in harmonic components is needed to activate the F0 estimator.

While expressed here as SNR like representation in dB, this is clearly not a measure of signal to noise ratio, but a ratio of the harmonic energy in the current frame to the frame with maximum energy in harmonic components.

Signal parts with high energy but low steadiness (e.g. drums) are suppressed, while silent but highly stationary signal parts are enhanced.

3.1.2 F0 Candidate Determination

The basic idea of the F0 estimation method is based on the Fourier principle that states, that a periodic signal excites spectral components at integer multiples of the inverse of the time domain period that is closely related to the perception of pitch and thus, the physical F0. This harmonic pattern can be used to find the fundamental frequency of a complex signal in spectral domain.

For each possible note n in a time-frame of a signal l, a salience $s[n]$ can be calculated by summing the amplitudes of all harmonic components through

$$s[n] = \sum_{m=1}^{M} g_e(n,m) \cdot Z[n + d(m)] \ , \ n = 21, ..., 108 \ , \tag{3.2}$$

with the overall number of harmonics to be considered M, the semitone spectrum $Z[n]$ defined in Equation 2.26 and a weighting function $g_e(n,m)$ that defines the weight of component m contributing to the overall salience $s[n]$.

The salience $s[n]$ expresses how prominent a fundamental frequency n is within the given signal, adapted from the magnitude of the harmonic components, as shown in Figure 3.1.

Due to the logarithmic 12-tone properties of the semitone spectrum, the integer frequency ratios $f_m = m \cdot F0$ of the first $m = 1, ..., 5$ components n_m become the specific semitone indexes $n_m = n + d(m)$, with $d = 0, 12, 19, 24, 28$ and $n_1 = F_0$.

The overall number of harmonics to be used is limited to $M = 10$, because higher harmonic can no longer be separated, due to the applied semitone spectrum. While the semitone approach limits the separation of the components, the limited resolution is of advantage for the assignment of the harmonics. Inharmonicities are very common in tuned percussion instruments, as the piano, where deviations of the frequencies of the harmonics from the

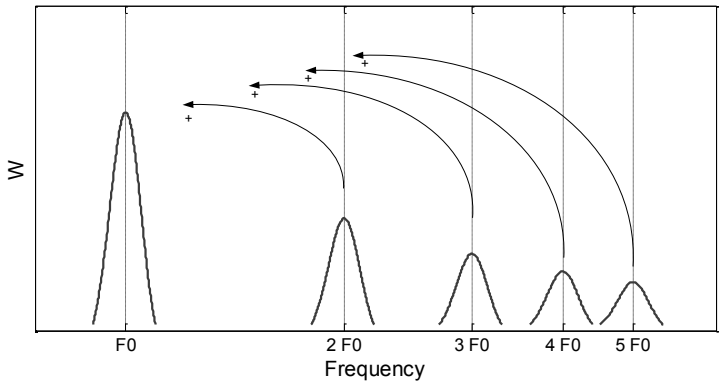


Figure 3.1: Summing magnitudes of harmonic frequencies to create a salience for a F0 candidate in the signal.

exact multiples of the fundamental frequencies occur. Using the whole semitone range instead of the exact frequency of the overtones overcomes these problems.

The weighting function $g_e(n, m)$ needs to be optimized for low F0 estimation errors and is crucial for the overall performance. Because each note leads to several peaks in the salience function and the problems that may occur, noted in the introduction of this Chapter, these weighting function is crucial to prevent especially octave errors, i.e. halving or doubling of the actual F0.

Thus, the overall weighting function $g_e(n, m)$ consists of several weighting components:

- $g_e(o(n))$: The overall weight of the salience according to the octave o of the fundamental frequency n.
- $g_e(m)$: The contribution of each harmonic partial m to the fundamental frequency n.
- $g_e(\varsigma)$: A weighting component according to the local steadiness of each component m, as defined in Equation 2.36.

The salience weighing according to the absolute frequency of the fundamental frequency of each F0 candidate $g_e(o(n))$ is needed to equalize the spectrum of the sound. Although a dB(A) weighting is used prior to the creation of the semitone spectrum, that already does spectrum correction adapting the

human perception, fine tuning is applied here using $g_e(o(n))$. In addition, applying different weights to each octave can reduce octave errors.

Especially the lowest octaves are over-suppressed and need to be enhanced to allow high recognition rates.

The overall note range $n = 21, ..., 108$ is divided in eight octaves by assigning the 12 semitones to the corresponding octave $o(n)$, starting with $n(A_0) = 21$ to $n(C_8) = 108$:

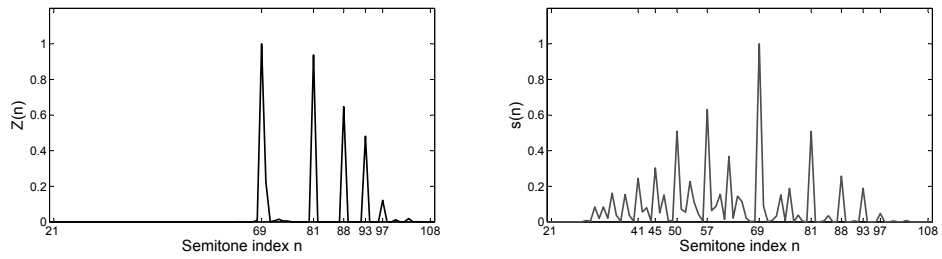
$$o(n) = \lfloor \frac{n-9}{12} \rfloor \ , \tag{3.3}$$

with $\lfloor \ \rfloor$ indicating rounding to the nearest integer towards zero. Each octave gets a distinct weight.

$g_e(m)$ describes the *importance* of each component of the harmonic series for the calculation of the salience function for each F0 and thus, takes the harmonic structure of the sound into account.

While all harmonic components do contribute to the salience, some of them are found to be more important, or meaningful, than others. This is especially necessary to prevent octave errors, i.e. F0 found one octave below or above of the true F0, that share most of the harmonic components of the signal and get a similar salience.

These additional peaks at harmonic and subharmonic indexes are clearly visible in the candidate spectrum shown in Figure 3.2(b).



(a) Semitone Spectrum (b) Candidate Spectrum

Figure 3.2: Semitone spectrum of a piano A4 with n = 69 and the corresponding candidate, or salience spectrum with multiple peaks generated at subharmonic indexes.

With the whole F0 estimation algorithm working on each time frame independently, this method does not take the temporal dependencies of adjacent frames into account to enhance the estimation results through temporal continuity conditions.

Here, the steadiness representation $\varsigma[n + d(m)]$ of the component m, defined in Equation 2.36, is used to model the temporal signal properties using the weighting function

$$g_e(\varsigma) = (\varsigma[m])^{\sigma} \ . \tag{3.4}$$

This ensures that stationary signal parts with stable harmonic structure are treated with a higher weight to create the salience function, while transient signal parts, with no stable harmonic components and thus, low local steadiness are suppressed, or not considered at all, and do not contribute to the salience. $\sigma > 1$ controls the overall influence of the steadiness over the magnitude of the component.

The overall weighting function $g_e(n, m)$ used to calculate the weighted sum of the harmonic amplitudes equals to

$$g_e(n, m) = g_e(o(n)) \cdot g_e(m) \cdot g_e(\varsigma) \tag{3.5}$$

Each of the components of the weighting function is optimized independently through a cyclically brute-force estimation process to minimize the F0 error rates.

The first F0 candidate $\hat{n}$, sometimes referred to as the predominant F0, is found by choosing the note $\hat{n}$ with the maximum salience within each time frame:

$$\hat{n} = \underset{n=21,\ldots,108}{\operatorname{argmax}} \ (s[n]) \tag{3.6}$$

In a monophonic or predominant F0 detection task we would chose this note and continue with the next time frame. To allow polyphonic F0 detections this method is applied in an iterative estimation and cancellation approach.

The iterative algorithm includes three steps, that are repeated until a termination condition is reached:

1. Estimation of the predominant F0 in the signal.
2. Cancellation of this predominant F0 and its harmonic components from the signal.
3. Continuation with the residual until a termination condition is fulfilled.

3.1.3 F0 Candidate Cancellation

Cancelling the components of a F0 found from the signal is necessary to find all notes in the signal iteratively. This step is crucial for all following

iterations, because each subtraction has to alter the original spectrum in a way, that all components belonging to the note are removed sufficiently, but at the same time the spectra of all other notes must remain unchanged, or at least, my not be corrupted in a way that makes it impossible to detect them.

The residual $R[n]$ at iteration i of the algorithm is calculated by subtracting the candidate spectrum $C[n]$ from the residual of the previous iteration R_{i-1}:

$$R_i[n] = \max\left(0, R_{i-1}[n] - C_i[n]\right) \ . \tag{3.7}$$

It is not possible to exactly determine the amount of a component belonging to a specific note that has to be cancelled. Thus, methods are used, that use some general assumptions on the spectral properties of a note, or some overall properties of the signal, to estimate the spectrum of the note and thus, the amount of subtraction applied to each component. All of them are only a approximation of the signal and tend to corrupt the spectral components of other notes.

Different methods are proposed here to estimate the candidate spectrum $C_i[n]$, with different complexity and amount of signal adaption.

The different subtraction methods lead to different remaining residual spectra and necessitate different component weighting functions in the estimation (salience calculation) step. For each of the methods, an estimation weight function is optimized independently.

3.1.3.1 Constant Spectrum Removal

The simplest method of candidate removal is based on the component weighting function $g_{sc}(m)$ used to create the salience function, together with the magnitude $Z[\hat{n} + d(m)]$ of each component and removes a constant fraction of each component, without taking any signal properties into account.

The candidate spectrum of the F0 candidate $\hat{n}$ is computed as the weighted magnitude of each component $\hat{n} + d(m)$

$$C[\hat{n} + d(m)] = s_f \cdot g_{sc}(m) \cdot Z[\hat{n} + d(m)] \tag{3.8}$$

The weighting of the partials of a detected sound by the function $g_{sc}(m)$ effectively insures that the partials are not entirely removed, but suppressed in a way that does not corrupt the harmonic structure to much. Especially higher harmonics are not entirely cancelled.

The overall subtraction is determined by the subtraction factor s_f and the weighting function $g_{sc}(m)$. This synthesis weighting function is again opti-

mized for low estimation errors.

3.1.3.2 General Spectrum Removal

The most obvious way to remove the components of a detected note is to assume some specific spectral properties that are common between all notes, played with any instrument. Of course, this generalization of the harmonic structure of a note does not necessarily represent the real structure of a specific note, but can suppress the note in a way that does not alter notes sharing components to much, but does suppress the candidate note sufficiently for the next iteration of the algorithm.

This method estimates the amount of substraction using the magnitude of the detected F0 $Z[\hat{n}]$. By chosing $\hat{n}$ as the anchor of the spectrum, all other components of the signal can be predicted by defining a general spectrum $g_{sg}(m)$ that is common between all notes:

$$C[\hat{n} + d(m)] = g_{sg}(m) \cdot Z[\hat{n}] \ . \tag{3.9}$$

While the estimation of all components out of the magnitude of the F0 of the note suffers from missing fundamental frequencies, it still proved to lead to good results for most of the signals and is able to give a good overall estimate of the harmonic structure of a note.

Again, the weighting function $g_{sg}(m)$ is optimized through a cyclically brute-force optimization process to minimize the F0 error rates.

3.1.3.3 Estimated Spectrum Removal

To overcome the general character of the general and constant spectrum subtraction, a method is proposed that again uses the saliences of each note in a frame to estimate the amount of each harmonic that does belong to the specific candidate and thus, how much of the magnitude has to be removed.

A component $\hat{n}_m = \hat{n} + d(m)$ of the candidate $\hat{n}$ is shared with another note n, i.e. $n_m = \hat{n}_{\hat{m}}$, when these two are in harmonic ratio. This is true for all n that are in subharmonic ratio $f = \frac{\hat{f}}{m}$ of $\hat{n}_{\hat{m}}$ and thus, for all N notes with $n = \hat{n}_m - d(m)$, with $d(m)$ again being the semitone distances of harmonic components.

To estimate the fraction of each of this shared components that belongs to the candidate $\hat{n}$, saliences $s[n]$ are computed for $\hat{n}$ and all n sharing this component according to Equation 3.2, but this time, leaving all shared com-

ponents out and taking only the unique, or unshared components of each note into account.

For each component $\hat{n}_{\hat{m}}$ of our candidate $\hat{n}$ repeat:

1. Find all notes n that share this component.
2. For each of the sharing notes, compute a salience relation.
3. Cancel the ratio assigned to the candidate $\hat{n}$ and continue with the next component.

The relation between two notes sharing a component is computed through

$$\frac{s[\hat{n}]}{s[n]+s[\hat{n}]} = \frac{\sum_{\hat{m}=1}^{M} Z[\hat{n}+d(\hat{m})] \cdot g_e(\hat{n},\hat{m})}{\sum_{\hat{m}=1}^{M} Z[\hat{n}+d(\hat{m})] \cdot g_e(\hat{n},\hat{m}) + \sum_{m=1}^{M} Z[n+d(m)] \cdot g_e(n,m)} \tag{3.10}$$

$$\text{for} \quad \hat{n}+d(\hat{m}) \notin n+d(m)$$

with the number of harmonics $M = 10$. $g_e(n,m)$ is again the estimation weight function described in Section 3.1.2.

The overall ratio of our note $\hat{n}$ on the component $\hat{n}_{\hat{m}}$ is then computed using the relations for all N notes n that do share the component:

$$r[\hat{m}] = \prod_{n=1}^{N} \frac{s[\hat{n}]}{s[n]+s[\hat{n}]} \ . \tag{3.11}$$

The candidate spectrum is then computed using the magnitude $Z[\hat{m}]$ of each harmonic of the candidate F0 and the relation $r[\hat{m}]$:

$$C[\hat{n}+d(\hat{m})] = r[\hat{m}] \cdot Z[\hat{n}+d(\hat{m})] \ , \tag{3.12}$$

with $\hat{m} = 1, ..., M$.

Using only components not shared between the notes, we compute a salience for each note in an isolated case, without taking the shared harmonics into account. While other notes still affect the relation, this factor is the same for all notes and thus, does not negatively influence the result.

Clearly, this method does again show the desired behavior of underestimating the fraction of each component that is removed. This underestimation together with the adaption of the spectral properties should allow an efficient

cancellation process, with a minimum of spectral corruption of simultaneously sounding notes.

This method is computationally less effective than the removal of a general or constant spectrum, but at same time uses signal properties to estimate the amount of each harmonic component that needs to be removed.

Using the saliences of notes with overlapping components, it is possible to remove the amount of the signal that does most likely belong to each note.

3.1.4 Termination Condition

The termination of the iterative algorithm needs to verify that no true F0 candidates are missed and at the same time, no false F0 candidates are inserted.

This task can be seen as estimation of the polyphony of the signal. Instead of determining the polyphony prior to the F0 estimation, a condition is checked after each iteration, wether the newly detected candidate F0 is still valid.

Here, a representation is used, that depends on the candidate signal $C_i[n]$ at iteration i of the algorithm and the candidate signals of all previous iterations $C_{1,\ldots,i-1}$. The algorithm is stopped, whenever a newly estimated candidate spectrum does no longer contribute to the overall candidate signal in a significant way, by computing the relations

$$t(i) = \frac{\sum_{j=1}^{i} \sum_n C_j[n] \cdot \varsigma[n]}{i^\gamma} \ , \tag{3.13}$$

$$t(i-1) = \frac{\sum_{j=1}^{i-1} \sum_n C_j[n] \cdot \varsigma[n]}{(i-1)^\gamma} \ . \tag{3.14}$$

With the steadiness representation $\varsigma[n]$, and

$$T(i) = \frac{t(i)}{t(i-1)} \ . \tag{3.15}$$

The iterative F0 estimation is cancelled if $T(i) < 1$ and the newly detected candidate not included in the overall signal. For $S(i) > 1$ the detected candidate is accepted, the candidate signal included, the residual computed and the algorithm continued from the beginning.

γ provides the possibility to adjust the termination. For $\gamma = 0$ the $S(i) > 1$ is always true and a global termination condition that stops the algorithm

after applying a maximum of 12 iterations is used. With increasing γ, the condition to accept detected candidates is getting stricter.

The multiple F0 estimation algorithm outputs a number of detected F0 and their appropriate salience. Instead of finding a continuous F0 and mapping it to the nearest note, the discrete semitone spectrum allows to find discrete F0 that are directly equivalent to a note.

No further steps are necessary to determine the pitch of a note in each time frame, but additional treatment is inevitable to convert the framewise estimated pitches into temporally continuous notes. This step is explained in the following Section.

3.2 Onset and Duration Detection

The framewise detected F0 candidates do indicate the presence of notes in each time frame. To describe notes with their distinct parameters onset and offset timings are necessary in addition to the pitch.

The tasks is typically referred to as onset detection because the beginning of a note is directly audible and thus, more important for the perception of music, but the offset detection of a note is covered by this step, too.

The challenge of the onset detection is here seen as combining the framewise estimated F0 and defining notes, with pitch, onset and offset times. Different methods are proposed, using different signal representations and features, as well as different detection functions. All of them use bandwise analyses within each subband of the semitone representation, as used in [52] or [75] and do thus have a direct relation to a distinct note.

The goal of the onset detection used here is not to detect each onset globally, but to enhance the time accuracy of the detected F0 partials and may be seen as a timeline correction of each note. Due to the weighting of each partial with the local steadiness in the F0 estimation step, the detected partials tend to cover only the steady parts of a note, not the transient parts of the attack and decay. The temporal accuracy must be enhanced to cover this parts of the note, too.

3.2.1 Decision Functions

Different decision functions are used, each with distinct advantages and drawbacks, and are applied to the different features explained below. Thus, $k_n(l)$ may be any of the below described features.

3.2.1.1 Simple Onset Correction

A simple way to correct the onset position l_{on} of a F0 partial found by the F0 estimator is to use some signal properties i.e. the signal energy in the given subband n and finding the maximum of the temporal derivative of the feature $\Delta k_n(l)$ in a window w_{on} of length W_{on} before the beginning of the partial l_p

$$l_{on} = \underset{l}{\operatorname{argmax}} \left[\Delta k_n(l) \cdot w_{on}(l)\right] \tag{3.16}$$

with

$$w(l) = \begin{cases} 1 & \text{for} \quad l_p - W_{on} < l < l_p \\ 0 & \text{otherwise} \end{cases} \tag{3.17}$$

and

$$\Delta k_n(l) = k_n(l) - k_n(l-1) \ . \tag{3.18}$$

The actual onset of the note detected by the F0 estimator is then assigned to this point with maximum change of signal properties.

3.2.1.2 Median Filter Onset Peak Picking

Finding a global threshold for onset detection functions is difficult, because the detection functions are noisy, and their magnitudes vary greatly, not only between different recordings but also between different instruments used in a single recording. An adaptive threshold must be applied locally instead of an global threshold, to ensure good detection rates [31].

Median filtering in a moving window is a common peak picking method to detect onsets or signal changes in music ([2], [14]).

The median filter is used to obtain an adaptive threshold $\rho_l[n]$ of the detection function $k_n(l)$ at each frame l by calculating the weighted median in a window of length W around the frame l:

$$\rho_n[l] = \rho + \gamma \cdot \text{median}\left[k_n(w_l)\right] \ , w_l \in [l - \frac{W}{2} \ , \ l + \frac{W}{2}] \ . \tag{3.19}$$

ρ and γ are constant values, that need to be adjusted for high detection rates.

An onset is detected, whenever the detection function $k_n(l)$ exceeds the adaptive threshold $k_n(l)$ at frame l.

While each detected onset is directly assigned to a semitone subband through the features used, a temporal confinement has to be made to assign each note to the nearest onset, without falsely appointing the onset of the note to early. This would be caused by a missed onset.

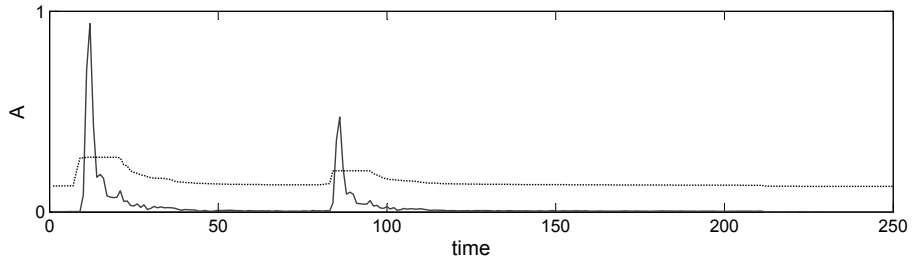


Figure 3.3: Adaptive threshold (dotted) creation using a median filter for peak picking on the transiency of two piano onsets.

Again, a window w_{on} is defined before the beginning l_p of a detected partial p, with length W_{on} and an onset is only assigned to this partial, when its found in this window. If no onset is detected within this window, the F0 candidate is discarded and no note announced.

The main advantage of the peak picking algorithm, compared with the simple correction of onsets described in Section 3.2.1.1 is the ability to not only *correct* the beginning of a note, but to define an additional requirement, that must be fulfilled for the announcement of a note. Instead of relying on the F0 estimation alone for the detection of a note, the onset is regarded, too, and both must be fulfilled to create a note.

3.2.1.3 Thresholding for Offset Detection

While the onset of a note is clearly defined by the signal properties, the offset is not for certain types of instruments. Many instruments do have a long release state that is not distinguishable from the sustain state of a note.

The temporal derivative of a signal property used to detect the onset is not able to indicate the offset of a note. Because the offset is considered to be less important for the perception of music, the temporal accuracy of the actual offset position is less important than the onset position.

The maximum of a detection function found in the range of the onset position l_{on} is used as reference $k_n(l_{on})$ and the actual offset value of the detection function $k_n(l_{off})$ is relatively defined:

$$k_n[l_{off}] = p \cdot k_n[l_{on}] \ , \tag{3.20}$$

with some relative factor p.

The offset position l_{off} is then found by following the detection function $k_n(l)$ from the onset position, until it falls below the threshold $k_n[l_{off}]$. $k_n[l]$ is smoothed using lowpass filtering prior to the offset estimation to prevent the detection of early offsets caused by local minima.

3.2.2 Detection Functions

The detection functions, the decision functions described in Section 3.2.1 are applied to, are based on signal properties that go through a distinct change at notes on- and offsets.

3.2.2.1 Pitch based Onset Detection

Pitch based onset detection uses only the information obtained from the multiple F0 estimator to detect notes onsets and offsets. This method is based on the assumption that the beginning of a new note results in a change of the detected pitch. The discrete semitone spectrum and the resulting discrete pitch estimations in each time frame makes this step quite straight forward.

A new onset is detected, whenever a $F0_n(l)$ is detected for semitone n in frame l that was not detected in the previous frame $l-1$.

The pitch difference is calculated between successive frames to detect onsets using the discrete pitch spectrogram $F0_n(l)$:

$$\text{onset}_n(l) \begin{cases} 1 & \text{for} \quad F0_n(l) - F0_n(l-1) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.21}$$

The drawbacks of this method are clearly visible, when examining the properties of the F0 estimator described in Section 3.1.2 and especially the computation of the salience function for each note in Equation 3.2 with the weighting function 3.5. The term $g_e(\varsigma)$ uses the steadiness representation of the signal defined in Equation 2.36 to prevent transient signal parts to contribute to the salience function and allows reliable F0 estimations taking only stationary signal parts with stable harmonic structure into account.

While this behavior is desired for the F0 estimation, the highly transient signal parts especially caused by strong onsets (e.g. from piano) are not included in the estimated F0. Thus, the pitch based method tends to result in delayed onsets.

The detection of offsets based on pitch information suffers from a similar problem. There is no change in the discrete F0 between the sustain state,

still belonging to the *on* state of the note and the release state, where the note is still sounding, but is no longer excited. While there are instruments whose offset of a note is nearly concurrent with the note falling silent (e.g. many wood instruments in dry room conditions), most instrument have a rather long release portion (e.g. plucked string instruments).

It is not possible to determine the real offset of a note by the discrete F0 spectrogram. It is only possible to detect the time when the note stops sounding, i.e. the end of the release of the note:

$$\text{offset}_n(l) \begin{cases} 1 & \text{for} \quad F0_n(l-1) - F0_n(l) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.22}$$

This may be quite accurate for instruments with a small resonating body, but its clearly not for instruments with a long, slowly evolving release.

This undesired behaviour leads to another problem, when two or more notes of the same pitch follow each other rapidly, with the new onset starting before the release of the previous note stopped sounding. This type of onset, or note transition, is not detectable using the pitch information alone and those two notes would be combined.

3.2.2.2 Pitch-strength based Onset Detection

To overcome the drawbacks of the pitch based onset detection, a pitch-strength based detection can be applied, using the salience $s[n]$ of each detected F0 instead of just the binary pitch information.

The simple onset correction in Section 3.2.1.1 and the peak picking decision function Section 3.2.1.2 is directly applied to the derivative $\Delta s_n[l] = s_n[l] - s_n[l-1]$. Still, the delayed onset detection remains, but the overestimated duration of the note caused by the still sounding release state of a note, can be avoided.

The onset is still detected according to the pitch based onset condition defined in Equation 3.21 and the salience of the onset position $s_n[l_{on}]$ is used as the reference, with maximum salience of the note. To define the offset of a note, a relative condition is created that, instead of following the note as long as the partial is detected, defines the end of a note when the salience falls under a given threshold.

This absolute salience threshold $s_n[l_{off}]$ depends on the maximum salience of the onset of the note $s_n[l_{on}]$ and some relative factor p, according to Equation 3.20.

While using the pitch-strength for offset detection allows better estimates

of the offset, it still suffers from the onset limitations explained in Section 3.2.2.1. Some additional problems may arise from inconsistent saliences caused by the iterative cancellation process. The cancellation of simultaneous notes may alter the saliences in a way that the lapse is corrupted and may lead to incorrect results.

3.2.2.3 Magnitude based Onset Detection

The magnitude based onset detection relies on the lapse of the amplitude, or envelope, in the distinct semitone subband defined by $Z_n(l)$ of a note to detect signal changes.

The decision functions from Section 3.2.1 are applied to the temporal derivative $\Delta Z_n(l) = Z_n(l) - Z_n(l-1)$ for onset detection and the thresholding method is used for the detection of the offset.

To further enhance this method, especially for sounds with missing fundamental frequency, and thus, no energy in Z_n but only in the semitones of the harmonics, these are taken into account to, by summing their amplitudes

$$Z_n^*(l) = \sum_{m=1}^{M} \frac{1}{m} Z_{n+d(m)}(l) \tag{3.23}$$

with M harmonics at the semitones indices $n + d(m)$.

3.2.2.4 Onset Detection in Complex Domain

The complex domain signal representation described in Section 2.3.4 enhances the energy based method to the use of the phase information and thus, should lead to better results for smooth, slowly evolving onsets.

For onset detection, the decision functions in Section 3.2.1 are directly applied to the rising transiency description $\Gamma_n(l)^+$, defined in Equation 2.37.

Because of no detectable change in signal propagation at the offset of a note, the transiency representation can not be used for offset detection. Thus, the pitch, pitch-strength or energy based detection function based on a relative threshold is used to define the notes' end.

3.2.3 Note Event Modeling

All former described methods detect onsets and offsets independently. Together with the detected F0 partial from the fundamental frequency estimation in each frame, these define a note. But this assignment is not consistent

when these two steps show conflictive results, i.e. a partial without an onset or an onset without a detected partial.

Instead of detecting notes on- and offset independently, a method is proposed now that takes the whole temporal structure, following the ADSR model decribed in Section 2.3, of a note into account. The note is seen as an event, with a distinct slope of features.

These note events are described using Hidden Markov Models (HMM). The models use specific features extracted from the signal that model both, the pitch as well as the temporal structure to calculate the overall likelihoods of each note with a distinct start and end point.

By combining the information from spectral and temporal dimension, a robust recognition system is created that is able to correct falsely detected F0 partials.

3.2.3.1 Theory of Hidden Markov Models

Hidden Markov Models are statistical models, which model a signal with unknown parameters as Markov process [58]. In contrast to Markov models, in HMMs the states are hidden and only the observations created by each state are visible.

Each state sequency does create an observation sequence and can thus be used in pattern recognition tasks to classify time-series, like speech or handwriting recognition.

First introduced for speech recognition tasks have HMMs lately been used for music transcription tasks, as in [62] or [72], and showed good results because of their ability to model the whole note as a stochastical process.

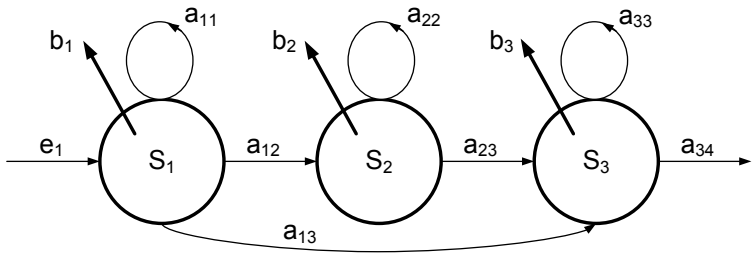


Figure 3.4: Exemplary left-right Hidden Markov Model with three states, state transition probabilities a_{ij}, observation probabilities b_j and entry probabilities e_j.

For each time frame l a feature or observation vector o_l is created, with

distinct features describing the signal.

Each distinct note event is represented by an observation sequence O of L feature vectors o_l [51]

$$O = o_1, o_2, ..., o_L \ , \tag{3.24}$$

and the underlying state-sequence $X = X_1, ..., X_L$ describes the temporal process of our signal [60]. The observation sequence O can be used to estimate X.

The recognition tasks is then to find the model λ_i that maximizes the observation probability $P(\lambda_i|O)$

$$\underset{i}{\text{argmax}} \ (P(\lambda_i|O)) \ . \tag{3.25}$$

This probability is not computable directly but using Bayes' rule gives

$$P(\lambda_i|O) = \frac{P(O|\lambda_i)P(\lambda_i)}{P(O)} \tag{3.26}$$

For a given set of prior probabilities $P(\lambda_i)$, the most probable class depends only on the likelihood $P(O|\lambda_i)$.

Assuming that the sequence of observed observation vectors is generated by a Markov model, a finite state machine changing states once every time unit and each time l a state j is entered an observation vector o_l is generated from the probability density $b_j(o_l)$. The transition from state i to state j is also probabilistic and is controlled by the discrete transition probabilities a_{ij}.

A process going through the state sequency $X = X_1, ..., X_L$ generates the observation sequence $O = o_1, ..., o_L$.

The joint probability $P(O, X|\lambda_i)$ that O is generated by the model λ_i through the state sequence X is calculated simply as the product of the transition probabilities a_{ij} and the output probabilities $b_j(o_l)$.

With the actual unknown (*hidden*) state sequence X the likelihood is calculated by summing over all possible state sequences $X = x(1), x(2), x(3), ..., x(L)$ that produce the observed feature sequence O

$$P(O|\lambda_i) = \sum_X a_{x(0)x(1)} \prod_{l=1}^{L} b_{x(l)}(o_l) a_{x(l)x(l+1)} \tag{3.27}$$

Gaussian Mixture Models (GMM) are used to model continuous densities of the observation distributions $b_j(o_l)$:

$$b_j(o_l) = \sum_{m=1}^{M} c_{jm} N(o_l; \mu_{jm}, \Sigma_{jm}) \tag{3.28}$$

with the number of mixture components M, c_{jm} the weight of the component m and $N(o_l; \mu_{jm}, \Sigma_{jm})$ a multivariate Gaussian distribution with mean vector μ and covariance matrix Σ:

$$N(o; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}(o-\mu)'\Sigma^{-1}(o-\mu)} \tag{3.29}$$

where n is the dimension of o.

The state transition probabilities a_{ij} together with the observation probability density $b_j(o_l)$ and the state entry probability $e_j = a_{0j}$ are all parameters needed to fully define the model.

Given a set of training examples for each model, these parameters can be determined by a reestimation procedure.

HMM Parameter Reestimation: The parameter estimation of each model is done using the Baum-Welch Reestimation algorithm [51].

The observation likelihood $b_j(o_l)$ in Equation 3.28 with the multivariate gaussian $N(o_l; \mu_{jm}, \Sigma_{jm})$ is defined by the mean μ_j and covariance Σ_j vector.

The full likelihood of each observation sequence is affected only by each possible state sequence. Each observation is assigned to every state in proportion to the probability of the model being in that state when the vector was observed [51].

The mixture components can be considered to be a special form of sub-state in which the transition probabilities are the mixture weights.

The parameters μ_j and Σ_j can then be estimated through the Baum-Welch reestimation formulae

$$\hat{\mu}_j = \frac{\sum_{l=1}^{L} L_j(l) o_l}{\sum_{l=1}^{L} L_j(l)} \tag{3.30}$$

and

$$\hat{\Sigma}_j = \frac{\sum_{l=1}^{L} L_j(l)(o_l - \mu_j)(o_l - \mu_j)'}{\sum_{l=1}^{L} L_j(l)} \tag{3.31}$$

with the probability $L_j(l)$ of being in state j and time l.

The state occupation probability $L_j(l)$ can be calculated using the *Forward-Backward* algorithm, with the forward probability $\alpha_j(l)$ of some model M

$$\alpha_j(l) = P(o_1, ..., o_l, x(l) = j | M) \tag{3.32}$$

that defines the joint probability of the first l observation vectors $o_1, ..., o_l$ and being in state j at time l.

The forward probability can be calculated recursively by summing the forward probabilities of all possible predecessors weighted with the transition probability a_{ij}:

$$\alpha_j(l) = \left[\sum_{i=2}^{N-1} \alpha_i(l-1)a_{ij}\right] b_j(o_l) \tag{3.33}$$

The backward probability $\beta_j(l)$ is defined as

$$\beta_j(l) = P(o_{l+1}, ..., o_L | x(l) = j, M) \tag{3.34}$$

and can be computed through the recursion

$$\beta_i(l) = \sum_{j=2}^{N-1} a_{ij} b_j(o_{l+1}) \beta_j(l+1) \ . \tag{3.35}$$

With

$$\alpha_j(l)\beta_j(l) = P(O, x(l) = j | M) \tag{3.36}$$

and

$$L_j(l) = P(x(l) = j | O, M) \tag{3.37}$$

the state occupation probability $L_j(l)$ can by calculated

$$L_j(l) = \frac{P(O, x(l) = j | M)}{P(O|M)} \tag{3.38}$$

$$= \frac{1}{P}\alpha_j(l)\beta_j(l) \ , \tag{3.39}$$

where $P = P(O|M) = \alpha_N(L)$, the total likelihood of the observation sequence O produced by model M.

The Baum-Welch reestimation of the parameters of each HMM with Equation 3.30 and 3.31 using 3.39 according to [51], goes as follows:

1. Calculate the forward $\alpha_j(l)$ and backward $\beta_j(l)$ probabilities for each state j and time l.
2. Update $\hat{\mu}_j$ and $\hat{\Sigma}_j$ for each state j and time l using Equations 3.30 and 3.31 with $L_j(l)$ from Equation 3.39 and o_l.
3. Use the final value of $\hat{\mu}_j$ and $\hat{\Sigma}_j$ to calculate new parameters.
4. If $P = P(O|M)$ is not higher than the value of the previous iteration then stop, otherwise continue with the next iteration.

The reestimation of the transition probabilities is done in the same way, using the equation

$$\hat{a}_{ij} = \frac{\sum_{l=1}^{L-1} \alpha_i(l) a_{ij} b_j(o_{l+1}) \beta_j(l+1)}{\sum_{l=1}^{L} \alpha_i(l) \beta_i(t)} \tag{3.40}$$

All parameters of the set $\lambda = (e, A, B)$ needed to clearly define a model for each class are now estimated and the model can be used to classify observation sequences, by computing the probability a observation sequence is created by this model.

Viterbi Recognition: The trained models can then be used to classify some unknown observation sequence O. For each of the models, the observation probability has to be calculated and the model with maximum observation likelihood defines the class.

This likelihood is computed using the same algorithm as the forward probability calculation defined in Equation 3.33 except that the summation is replaced by a maximum operation.

For a given model M, $\phi_j(l)$ represents the maximum likelihood of observing the sequence O of feature vectors $o_1, ..., o_l$ and being in state j at time l. This partial likelihood can be computed using the following recursion:

$$\phi_j(l) = \max_i [\phi_i(l-1) a_{ij}] b_j(o_l) \tag{3.41}$$

where $\phi_1(1) = 1$ and $\phi_j(1) = a_{aj} b_j(o_1)$.

The maximum likelihood $\hat{P}(O|M)$ of the sequence of length L for a model M with N states can the be computed by

$$\phi_N(L) = \max_i [\phi_i(L) a_{iN}] \ . \tag{3.42}$$

This is called the Viterbi algorithm [51] that finds the best path through a matrix with the states on vertical dimension and the observation sequence on the horizontal dimension.

Recognition Networks: To be able to do a segmentation and classification of a observation sequence, i.e. detect the sequence of consecutive models within the observation sequence, recognition networks have to be created.

A complex HMM is created by connecting all simple class HMMs in a way that each state of the complex model is one of the simple models. Through the possible state transitions of the complex model, a *language* model can be created.

Still, the Viterbi algorithm is used for the recognition, to find the most likely state sequence of the complex model, indicating the class sequence. With the most probable state position at a distinct time, the segmentation can be achieved.

3.2.3.2 Observation Features

The observation sequence O_L produced by a note event consisting of feature vectors o_l at each time frame l needs to represent the properties of the signal in order to model the event.

Thus, features have to be used, that are able to describe both, the temporal slope of the signal and the pitch of the note. Modeling the whole event allows to estimate all properties of a note in a single step and takes the temporal dependencies of the features into account.

The features used are bandwise calculated according to the semitone representation described in Section 2.1. The features for the semitone band n in each frame l are:

1. The detected $F0_n(l)$ in this band n with the corresponding salience $s_n(l)$ and the temporal derivative $\Delta s_n(l)$.
2. The harmonic magnitude $Z_n^+(l) = \frac{1}{m} \cdot \sum_m Z_{n+d(m)(l)}$ and its temporal derivative $\Delta Z_n^+(l)$.
3. The subharmonic magnitude $Z_n^-(l) = \frac{1}{m} \cdot \sum_m Z_{n-d(m)(l)}$ and its temporal derivative $\Delta Z_n^-(l)$.
4. The complex domain transiency representation $\eta_l[n]$, including the separation in rising $\eta_l^+[n]$ and descending transiency $\eta_l^-[n]$ in the harmonic subbands and its temporal derivative.

While $F0_n(l)$ and $s_n(l)$ carry the pitch information of the given semitone band, $\eta_l[n]$ and its separation describe the temporal slope of each note event. Together, these feature model the whole event with distinct pitch and onset/offset times jointly.

To cope with missing fundamental frequencies in the slope related features, the semitone bands at harmonic indices can be used, too, by summing feature values of these bands. The subharmonic magnitude allows the detection of octave errors introduced by the F0 estimator.

Note, that because of the semitone properties of the spectral representation, the detected F0 are discrete values and thus, the $F0_n$ features used here

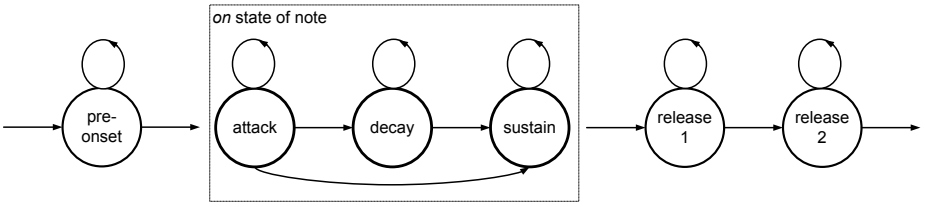
do not have an absolute value of pitch (this is defined by n) but just a binary pitch information, in the form of *F0 detected* or *F0 not detected*. The corresponding salience denotes the *pitch strength* of the particular F0 detection.

The observation sequence O_n of a note event consists of L observation vectors $o_n(l)$, each with all, or a subset, of the features described above.

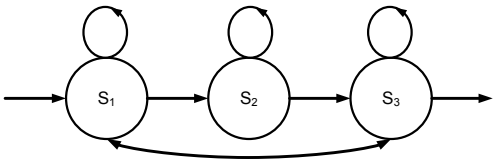
The features are standardized to have zero mean and standard deviation of one within each song.

3.2.3.3 Model Properties

The bandwise feature calculation described above results in feature sets, that are shared between the models for each note n. Thus, each of the models can share the same parameters and each of the semitone bands can be processed independently. The actual note index n does only depend of the semitone index of the band it is detected in.



(a) Note on model



(b) Note off model

Figure 3.5: Topology of the two models used for note on / off segmentation off each semitone band.

To detect notes in each semitone band and segment the note band in *note* and *no note* segments at the same time, two different models are used: a *note on* and a *note off* model. The recognition step does then simply perform a temporal segmentation of the note band in *note on* and *note off* regions.

The note *on* model is trained with bandwise feature sets of notes and the *off* model with sequences where no note is sounding.

A three state left-right model is used to model the *on* state of a note event, consisting of the attack, decay and sustain state of the note. While the signal properties before the onset and after the offset of the note do not belong to the *on* state, they are of course very characteristic for the note process. Especially the release state after the offset of a note is important for the event.

A *pre-onset* state is included before the three-state on part and two *release* states are included after the note offset to model these. In addition, a note transition state may be included, whenever two notes follow each other rapidly, and thus, the release of the first note and the onset of the second note are assumed to be greatly overlapped.

The topology of the recognition network is simple, allowing transitions from a note to the next or to silence.

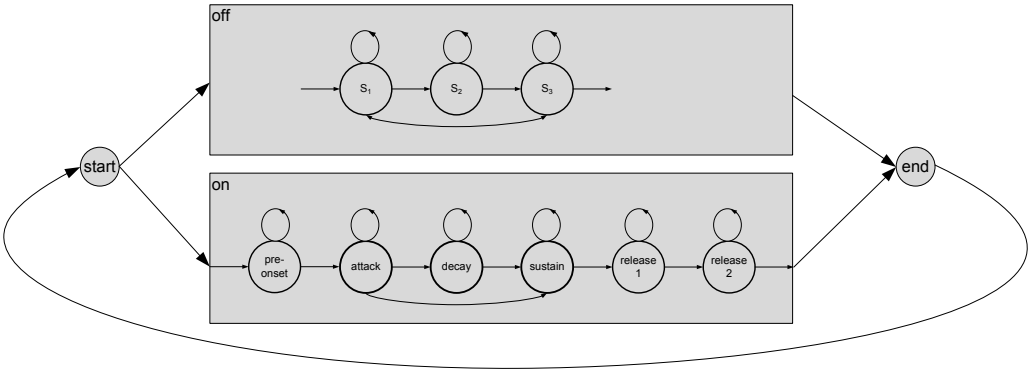


Figure 3.6: Recognition network used for note on / off segmentation.

Labeling: Labeling the data is necessary for training of the parameters. All signal frames occurring in the on state of a note a labeled *on*. No labeling of the underlying temporal structure (ADS) of the on state was used. So this can be seen as using word models in comparison to speech recognition.

While the on- and offset of the notes are clearly defined by the reference data, the release is not and depends greatly on the instrument type.

The detected F0 partial is followed and labeled *release*, only when no new note starts and a least a number of time frames. The *pre-onset* state is just a single time frame directly preceding the onset to model the signal changes starting with the onset of a note. Whenever the release of a note intersects the pre-onset of another note, the part is labeled *trans* to allow fast transitions between rapid note sequences.

Although the bandwise features are the same for each note n, and so, the same model could be used to segment each note band, there do of course differences exist in the distribution of the features in different bands. Note events in low frequency bands tend to be longer than those in higher frequency ranges, especially the release is longer for most instruments, and different distribution of the subband magnitude exist in different frequency regions.

Instead of creating global models covering all semitone bands, neighbouring bands are grouped to form models with similar note properties.

A single model is created combining the note bands over half a octave each, or six bands, while these ranges are larger in the lowest and highest octaves due to the lower occurence of these notes. This results in ten groups of notes and thus, ten note on and off models.

3.2.3.4 Event Modeling for Onset Detection

In a similar way compared to the onset and duration correction of the detected F0 partials, the note event models are used to detect the actual beginning and end of the notes, by modeling the whole event. Only notes are included in the parameter estimation process that were detected by the F0 estimator. This ensures consistent models, that rely on detected F0 and do only correct the onset and offset of a note, using the additional features of the signal within the subband of the note.

The used feature sets put greater authority on the features depending on the detected F0s, and a minimum of additional features is used to model the signal propagation.

3.2.3.5 Event Modeling for Note Detection

The note event modelling approach introduces the ability to discard F0 candidates detected by the F0 estimator, based on the information obtained from the signal slope.

Instead of using the outcome of the terminated F0 estimator optimized for both, high precision and recall rates, a F0 estimator optimized for high recall and lower precision rates is used to allow the note event modeling to discard falsely detected F0 candidates on the basis of the overall signal slope in the note range to enhance the precision rates.

To achieve this, the F0 estimator is not terminated using the termination condition described in Section 3.1.4, but an unterminated calculation is done, resulting in 12 detected F0 candidates in each frame. This defines the max-

imum polyphony detectable by the F0 estimator.

Larger feature sets are used with additional features, compared to the onset detection approach described in Section 3.2.3.4, to allow the detection of falsely inserted F0 candidates.

3.3 Making use of Musical Knowledge

All methods of music transcription described in the previous Section rely only on signal properties. Comparing the performance on music transcription of an average human, without any musical education with the performance of an highly skilled musician, this knowledgeless approach my not be sufficient.

In this Section are methods introduced that use high level musical information, namely the musical key of a song as well as the musical meter and beat position to enhance the results obtained by signal analysis in the preceding Sections.

3.3.1 Musical Key

The musical key of a musical piece is defined by the chords used in a song, and thus, by the note scale these are drawn from. A major key and a minor key are called a relative-key pair if they consist of scales with the same notes (e.g. the C major and the A minor).

In chromatic western music 24 different musical keys exist with 12 individual scales, each starting with one of the 12 semitones (the tonic) that defines the name of the scale.

The method used here was proposed in [22] for the estimation of musical keys in recordings. While ground truth key labeling is used, high recognition rates for the key estimation were achieved in [22] and could be used instead.

To enhance the accuracy of the fundamental frequency detection, the F0 estimation is extended to the knowledge of the key. Therefore, the weighting function $g_e(n, m)$ in Equation 3.5 is enhanced to carry another component g, defining the actual probability of a specific note in the scale of a given key.

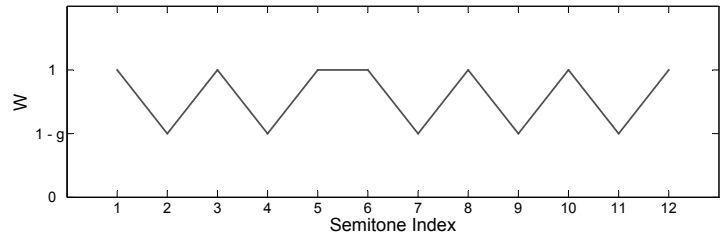
A root key pattern is defined that is shifted to the tonic for each of the keys $A, ..., G\sharp$, and does the scaling of the tones, that appear within the scale, or do not. The scaling is applied to each note n in the summation process of the F0 estimator and does not affect the summation of the harmonics. These are not scaled.

Chromatically altered notes may not be suppressed entirely. Although some

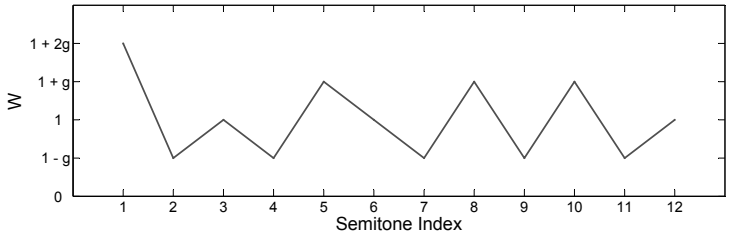
composers do stick with the notes of just one scale, normally some notes from others scales are used, too. Especially for modern Rock and Pop music these traditional guidelines are often not applied.

A scaling factor is used and the salience of chromatically altered notes are suppressed. Two different patterns are used: while the first assigns all notes on a scale the same weight, the second one uses the frequency of each note appearing in one of the chords of the scale to weight them differently. Not the complete pattern of the key but its main chords are used for accompaniment and the notes occurring in these chords are employed. The used chords are: tonic, subdominant, dominant and the relative minor chords. The two root patterns are shown in Figure 3.7 with the suppression factor g.

In Figure 3.7 the two resulting root patterns are shown.



(a) Scale based



(b) Chord based

Figure 3.7: Two different key patterns are used for scaling of chromatically altered notes.

3.3.2 Musical Meter

While the musical key of a song describes the pitch, or vertical structure of the notes, the musical meter of a song describes the temporal, or linear

structure of the notes.

This includes the song's quarter note tempo, meter and beat positions and is defined through both, the onset positions, as well as the duration of the notes. These information can be used to enhance the temporal precision of the notes, namely the on- and offset positions.

The method of beat tracking used here, was carried out in [18] is thus only briefly described. See [64] for a detailed explanation.

3.3.2.1 Beat Tracking

Preprocessing: A simplified psychoacoustic model is applied to the input signal using Mel-Frequency band filters and low pass filtering of the envelope to enhance onset peaks of the signal. Then a weighted differential d_w of the envelope is computed in each of the bands.

Meter and Tempo estimation: A delayed comb filter bank is used to compute the energy of the output of each filter over all bands. The delay of the filter with the maximum output energy is chosen to define the overall mean tempo of the song. The energy outcome of the filter indicates how well the signal resonates with the specific comb filter and is stored in the meter vector m.

A simple rule based method is used to discriminate between duple and triple meter, using the values of the meter vector m at distinct positions:

$$s_{duple} = \frac{1}{3}\left[m(4) + m(8) + m(16)\right] \tag{3.43}$$

$$s_{triple} = \frac{1}{3}\left[m(6) + m(9) + m(18)\right] \tag{3.44}$$

The greater of the two determines the base meter.

Beat tracking: To estimate the actual beat position p out of the tempo and meter estimation, a tracking envelope E_t is created by filtering the weighted signal differential d_w with a comb filter with the length of the detected tempo. Multiple pass beat tracking is performed on this tracking envelope, allowing the actual beat position to differ slightly from the expected position to adapt tempo changes within a song.

3.3.2.2 Using the Beat Information

To use the actual beat information for enhancement of the accuracy of the transcription system, the note event modeling approach described in Chapter 3.2.3 is enhanced to carry the actual beat position p as an additional feature. The discrete beat position vector p is converted into an continuous slope assigning a cosine modulation. The actual quarter beat positions p_1 and p_2 are on two successive maxima of the cosine and the values in between are the corresponding values of the cosine as shown in Figure 3.8.

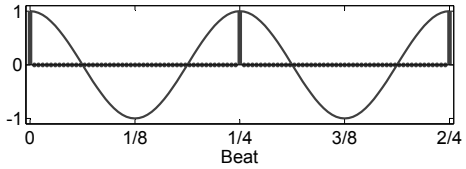


Figure 3.8: Cosine modulated beat position to obtain a continuous measure that follows the meter of a song.

Quantizing Note Positions: As a final step of the transcription system, the notes found after F0 estimation and onset and duration detection are quantized to the nearest fraction of a quarter beat position, i.e. 1/16 or 1/32.

Chapter 4

Results and Discussion

4.1 Evaluation Database

The collection of musical recordings used to evaluate the methods of music transcription described in the previous Chapter need a reference transcription to use as ground truth. While there are notations available for nearly all musical recordings, these are not sufficient here, because a reliable and accurate annotation of note onsets and offsets is necessary for a meaningful evaluation.

Two different solutions to this problems are introduced: first, a database is constructed of synthesized midi files, and second, an alignment is created between midi and real audio files to obtain the reference transcriptions of the recordings, from the midi files.

To cover a broad range of musical types, both classical and Rock/Pop music examples are used to construct the database. The Rock/Pop examples are taken from the *MTV Europe Most Wanted 1990 - 2000* and the classical pieces from *Hundert Meisterwerke der klassischen Musik*.

4.1.1 Midi Database

A freely available database with such accurate reference transcription does not exists, so most of the methods proposed up to now use midi files as source for both, the reference transcription and through synthesizing the acoustic data.

The midi files are taken from miscellaneous online databases and synthesized using *TiMidity++* [68], a software synthesizer that is able to convert midi to wave files, together with *EAWpats* [17], a set of high quality sound fonts.

While this way of obtaining the transcription data is perfect in terms of temporal accuracy, it lacks some acoustic variance present in real recordings. Songs played by musicians will sound different each time they are recorded, due to slightly different play of the musicians, different instruments, room conditions, noise, etc. All these influences are missing in midi file and in addition, the artificial instruments do have restricted and idealized properties, both temporal and spectral. The results of the transcription of synthesized midi files do not necessarily represent the results obtainable on real audio files.

4.1.2 Midi To Audio Alignment

To test the transcription system under real conditions, i.e. on real audio data, and to overcome the limitations of the midi database, the real audio database is created using the same songs, as in the midi database.

To obtain a accurate reference transcription for the real audio files using the note level parameters of the midi files, the synthesized midi files need to be force aligned to the real audio files. Each note with exact timings in the midi files is assigned to the corresponding region in the raw audio signal. This allows to automatically obtain the reference transcription of the database [70].

Chroma Representation

A chroma spectrogram is computed for both, real audio and synthesized midi signal, out of the discrete semitone spectrogram $Z_l[n]$ defined in Chapter 2.2.4, by creating a 12-dimensional pitch class vector summed over each octave o

$$C_l[p] = \sum_{o=0}^{8} Z[p + o \cdot 12] \ . \tag{4.1}$$

For each pitch class $p = 1, ...12$ or $A, ..., G\sharp$ one summed magnitude is created.

Chroma features are prefered over the full, 88 dimensional semitone representation, because they are less sensitive to the spectral shape, or timbre of the signal, but only on the presence of notes, or chords [27]. Because different instrumentations can be present in midi and audio files and the overall timbre of midi instruments is quite different from that of real instruments, this behavior is desired.

The chroma vectors are normalized to have zero mean and variance one, to eliminate loudness differences between the two signals and a silence frame

is appended at the beginning and end of both files to align start and end points.

Similarity Matrix

To compare these two sequences of chroma vectors, a distance measure using the Euclidean distance is used to express the difference of each vector pairs.

A similarity matrix (SM) or distance matrix $SM(i, j)$ is created by calculating the Euclidean distance for all pairs of real $i = 1, ..., I$ and midi $j = 1, ..., J$ chroma vectors:

$$SM(i,j) = \sqrt{\sum_{p=1}^{12} \left(C_{i,real}[p] - C_{j,midi}[p]\right)^2} \tag{4.2}$$

The distance matrix does indicate the pairwise distance of all chroma vectors I and J present in the real and midi file, respectively.

To align the two files, the best path, i.e. the path with minimum overall distance through the matrix has to be found.

Forced Alignment

Dynamic Time Warping (DTW) [21] is used to find the best path through the distance matrix $SM(i, j)$ starting in $(1, 1)$ and ending in (I, J).

DTW consists of a forward and and a backward step. In the forward step, a summed distance $D(i, j)$ for each matrix cell (i, j) is computed by summing all distances along the best path starting in the origin $(1, 1)$ to (i, j).

The calculation pattern shown in Figure 4.1 is used to calculate the distance for the cell (i, j). This pattern allows diagonal steps, i.e. going to the next vector in both, midi and real files, and horizontal or vertical steps, i.e. staying in the same vector in midi file while going to the next vector in real file, and vice versa. Each of these steps has the same weight assigned.

The best path up to location (i, j) in the matrix does then only depend on the neighbouring cells and the the actual distance between the vectors i and j:

$$D(i,j) = \min \begin{cases} D(i-1, j-1) & + SM(i,j) \\ D(i, j-1) & + SM(i,j) \\ D(i-1, j) & + SM(i,j) \end{cases} \tag{4.3}$$

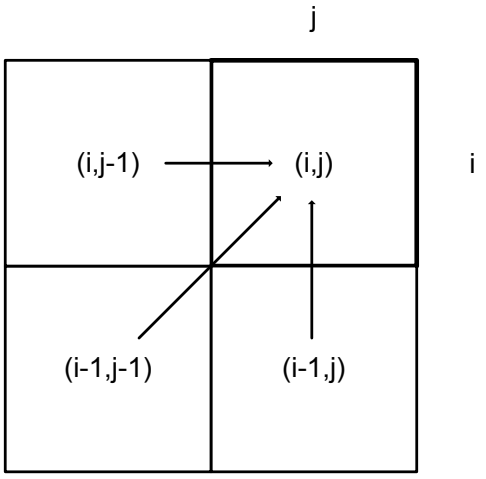


Figure 4.1: Calculation pattern of DTW to find best path through the similarity matrix.

In the backward step, backtracking is used, to find the optimal path of lowest overall distance through the matrix by recursively looking up the point providing the best antecedent for each point on the path.

Figure 4.2 shows the similarity matrix for a midi and audio recording of *Beethoven - Für Elise*. Regions of dark color indicate low distance, or high similarity. The best path (white) is shown, too.

Automated Alignment Evaluation

The DTW does always find a best path through the similarity matrix. Some criteria are introduced, to allow an automatic qualification of the obtained best path found, and thus, a measurement of the quality of the alignment.

Automatic Segmentation: Especially in complex classical recordings it can not be guaranteed that the midi and real recording contain an identical segment of the whole musical piece. Only parts of the composition may be present in the recording, repetitions are inserted or left out.

To detect these unmatchable parts automatically, regions with mainly diagonal steps are searched on the best path. It is assumed that identical and thus, well alignable parts in the signals contain diagonal steps in majority, while badly alignable, or missing parts result in mainly horizontal or vertical steps.

To detect unmatchable areas, a median filter is applied across the slope of

the best path. The median filter of length 1 *sec* computes the median of the slope and segment boundaries are found, when the median slope exceeds or falls under certain limits. This limits may not be to strict, otherwise songs of differing tempi can not be aligned. For the upper bound of the slope, 5 was chosen, and 0.2 for the lower bound.

In Figure 4.2 the different length of the two recordings is visible. While the first part of the signals aligns well, the acoustic recording contains an additional part that is not present in the midi recording. The segment detection finds the boundary (black) between these two segments and allows to recover the very good aligned first segment, while the second one, a repetition, is discarded.

Measures of the Quality of Alignment: To measure the overall quality of the alignment, the mean best path distance D_p is used, by computing the mean distance of all frame pairs on the best path. While this does indicate the quality of the alignment, this distance can not be directly compared with other alignments, because the overall distance between the two recordings is influenced by timbral differences caused by differing instrumentations. Even the best path distance will be large in case of different instruments used, although the actual alignment may be accurate.

The overall mean distance D_o of the whole SM is used to measure the timbral differences, and the ratio r_A of mean beast path and mean overall matrix distance is used to express the quality of the alignment:

$$D_p = \frac{1}{L_p} \sum_{i=1}^{I} \sum_{j=1}^{J} SM(i,j) \quad , \text{ for } (i,j) \in \text{best path} \tag{4.4}$$

$$D_o = \frac{1}{I \cdot J} \sum_{i=1}^{I} \sum_{j=1}^{J} SM(i,j) \tag{4.5}$$

$$r_A = \frac{D_p}{D_o} \ , \tag{4.6}$$

with the overall best path length L_p.

While these measures allow to preselect candidates with likely good alignment, the definite selection was carried out by subjective evaluation.

4.1.3 Obtaining the Transcription

The best path search through the similarity matrix results in an assignment of frame pairs in acoustic and midi files. Each time frame in the midi file is

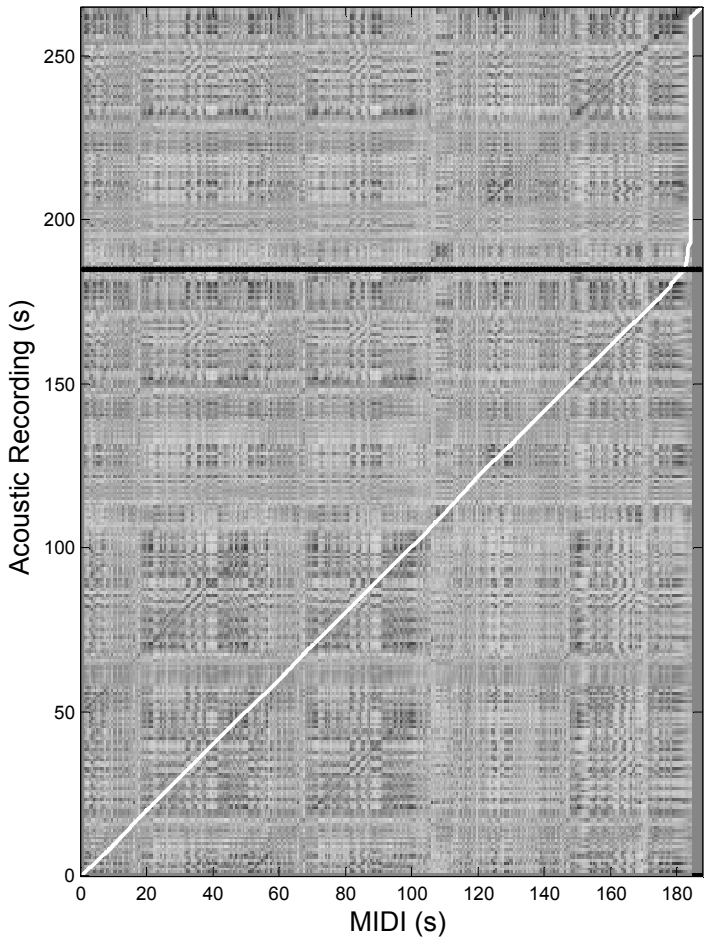


Figure 4.2: Similarity Matrix (SM) of midi and real audio recording of *Beethoven - Für Elise*, with best path (white) and detected segment boundary (black).

assigned to one or more time frames of the acoustic recording. The start and end point of the temporal range in the acoustic recording a midi note event is assigned to is used to correct the onset and duration information in the midi files to the corresponding positions in the acoustic recording.

The alignment of the reference transcription in midi files to the acoustic recording of the real audio files allows to obtain a reference transcription.

Even with manually evaluation the selected pairs will never be as accurate as the pairs of midi files and their synthesization. But still, this method allows the evaluation of the performance of the transcription system on real audio recordings.

4.1.4 Database Statistics

The overall database used to evaluate the performance of the music transcription methods consists of 4 parts: The *MTV* database extracted from the *MTV Europe Most Wanted 1990 - 2000* and *Classic* extracted from *Hundert Meisterwerke der klassischen Musik*. The *midi* database is constructed by synthesizing the midi files and the *real* database is created through aligning the midi transcriptions to the real audio recordings.

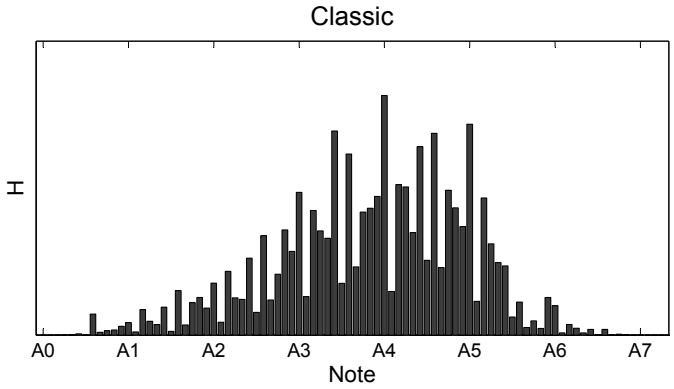
- *mMTV* : synthesized midi database of MTV music.
- *mClassic* : synthesized midi database of classical music.
- *aMTV* : real audio MTV database .
- *aClassic* : real audio classical music database.

50 songs are in *Classic* and *MTV*, with the same songs in *midi* and *real*. The real audio database may contain multiple well aligned segments of a single song that were recovered from a overall badly alignable song through the best path segmentation in the alignment process.

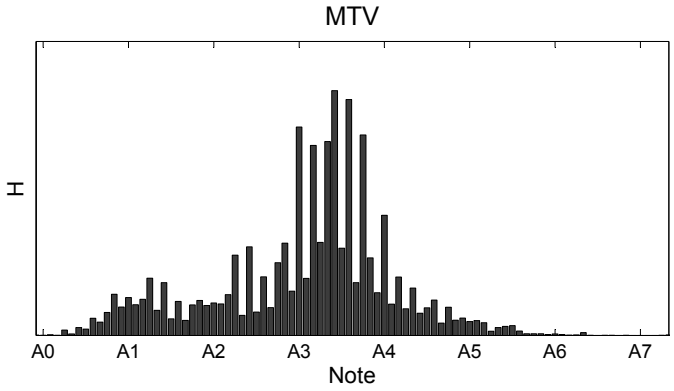
Statistics of the database are shown in Table 4.1 and the histogram of pitch distributions in the two databases is presented in Figure 4.3.

Some differences between the two datasets are clearly visible: The mean note duration of the *MTV* database is less than in *Classic*, indicating higher average tempo, and while the overall variance of the pitch distribution is smaller in *MTV*, there is a larger amount of notes with very low pitches caused by the prominent use of bass in modern Pop music.

These two properties of the *MTV* database should make the transcription task more challenging than on the *Classic* database.



(a) *Classical music* database



(b) *MTV* database

Figure 4.3: Histogram of pitch distributions in the two parts of the database.

	Corpus	Songs	Seg	dur [min]	Notes [10^3]	$\overline{note\ dur}$ [s]
midi	MTV	50	50	215.95	191.933	0.53
	Classic	50	50	201.88	234.208	0.59
audio	MTV	50	59	118.27	116.949	0.52
	Classic	50	87	151.43	183.294	0.60
	overall			687.53	726.384	

Table 4.1: Statistics on the database used to evaluate the performance of the transcription system. *midi* notates the synthesized midi files and *audio* the real audio files with aligned transcription. *Seg* notates the number of segments, and *dur* the duration.

4.2 Evaluation Criteria

To evaluate the performance of the transcription system on the datasets, evaluation criteria, or performance measures are introduced.

Different measures are used to be able to evaluate the performances of the F0 estimator and the onset detection independently.

Pitch-based Evaluation

To evaluate the performance of the F0 estimator measures are used, that do not rely on the onset and offset times of a note. A transcribed note is assigned to a reference note, when these two do at least partly overlap in time ($\min(\text{offset}_{ref}, \text{offset}_{trans}) > \max(\text{onset}_{ref}, \text{onset}_{trans})$) and have the same pitch. Only one transcribed note is assigned to each reference note, if multiple transcribed note do overlap with one reference note the one with minimum onset distance $|\,\text{onset}_{ref} - \text{onset}_{trans}\,|$ is assigned to the reference note, and all others are treated as errors.

Recall, percentage of reference notes correctly transcribed, and precision, percentage of correct transcribed notes of transcribed notes are computed, together with the F_1 measure:

$$\text{recall } (NR) = \frac{\text{correctly transcribed}}{\text{reference notes}} \tag{4.7}$$

$$\text{precision } (NP) = \frac{\text{correctly transcribed}}{\text{transcribed notes}} \tag{4.8}$$

$$\text{F1 measure } (NF_1) = \frac{2 \cdot NR \cdot NP}{NR + NP} \tag{4.9}$$

Time-based Evaluation

The actual temporal accuracy of onset and duration of each correctly detected note is estimated independently. Thus, only the correctly transcribed notes assigned to a reference note a regarded, all incorrectly described notes are discarded here.

The mean overlap ratio OR between transcribed note and the assigned reference note is calculated to express the temporal accuracy of each note:

$$OR = \frac{\min\left[\text{offset}_{ref}, \text{offset}_{trans}\right] - \max\left[\text{onset}_{ref}, \text{onset}_{trans}\right]}{\min\left[\text{onset}_{ref}, \text{onset}_{trans}\right] - \max\left[\text{offset}_{ref}, \text{offset}_{trans}\right]} \tag{4.10}$$

4.3 Results and Discussion

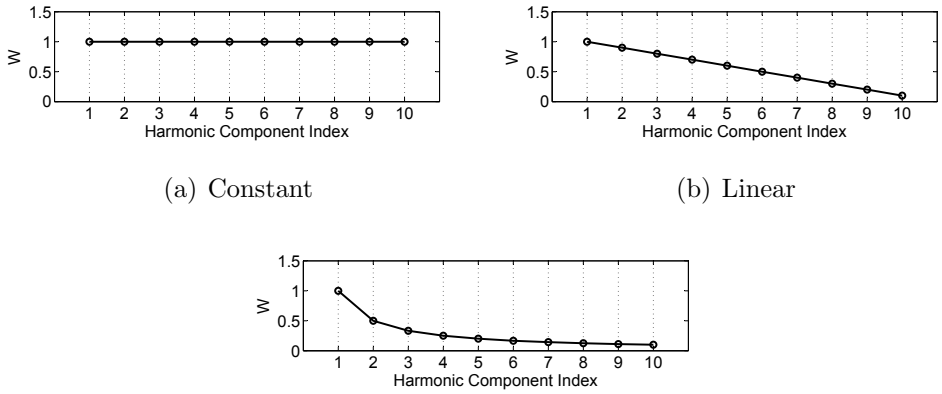
Using these evaluation criteria the performance of the multiple F0 estimator and the methods of onset and duration detection on the database are evaluated independently.

4.3.1 Multiple F0 estimation

The evaluation of the methods of F0 estimation proposed in Section 3.1 is carried out for each method independently. The main parameters are the weighting functions used to find the candidates and for the crucial step of candidate removal. No postprocessing in terms of on- and offset correction is applied to the detected notes to allow a clean evaluation of the performance of the F0 estimator and estimation of the parameters. Temporal continuously found F0 are simply combined to form a note. This is explained as pitch based on- and offset detection in Section 3.2.

Estimation weight functions optimization

The weighting function $g_e(n,m) = g_e(o(n)) \cdot g_e(m) \cdot g_e(\varsigma)$ of the estimation process is estimated for each of the subtraction methods independently, because the different kinds of subtraction lead to different kinds spectrum corruption and thus, enforce different weights for each component in the estimation step to compensate the spectral corruption.



(a) Constant

(b) Linear

(c) Quadratic

Figure 4.4: Different estimation weight functions $g_e(m)$ for weighting the components m of each candidate, with index 1 being the fundamental frequency.

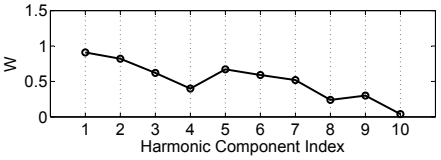
Component weight: To optimize the estimation weighting function of the harmonic components $g_e(m)$, the three initial functions shown in Figure 4.4 are used to find a ground truth, and the best of them is optimized for lowest F0 error rates by updating each component weight cyclically.

The optimized component weights are shown in Figure 4.5 for each of the three methods explained in Section 3.1.3. The optimization is especially important to prevent octave errors principally introduced by the F0 estimator.

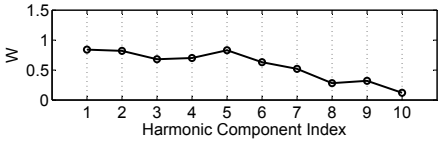
Despite the fact, that the estimation step is the same for all of the proposed methods, the different type of subtraction and thus, dissimilar alteration of the spectral structure leads to slightly different optimized estimation weights.

While all of them show a overall roughly linear slope, some components differ greatly from this characteristic.

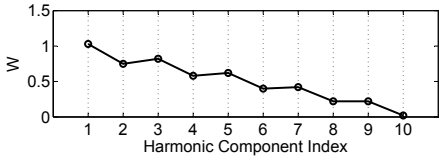
All of them give higher emphasise to the uneven numbered components - note, that component index $m = 1$ is the fundamental frequency and thus, these



(a) $g_{ec}(m)$ for *constant* spectrum removal



(b) $g_{eg}(m)$ for *general* spectrum removal



(c) $g_{ee}(m)$ for *estimated* spectrum removal

Figure 4.5: Optimized component weighting $g_e(m)$ for F0 estimation, for the three methods using constant, general and estimated spectrum removal.

are the event numbered harmonics - that seem to have higher significance to the actual fundamental frequency.

Table 4.2 displays the results obtained for the three methods, constant, general and estimated spectrum removal introduced in Section 3.1.3, together with the results of the weighting function shown in Figure 4.4 and the optimized $g_e(m)$ for each method.

The linear $g_e(m)$ is the best initial estimate in all of the methods, optimizing this leads to better results in each case. Constant and quadratic slope does note represent the impact of the harmonics to the actual fundamental frequency.

The results obtainable through the different weighting functions show big differences and thus, do indicate the importance of the optimization of $g_e(m)$. Especially the sensitivity to octave errors of the F0 estimator necessitates adapted weighting of each component.

The actual numbers of the weighting functions are mentioned in Table A.1.

F0 weight: Optimizing the component $g_e(o(n))$ of the overall estimation weight function $g_e(n, m)$ depending on the octave index $o(n)$ of the candidate is carried out on the spectrum estimation method.

One weighting function is optimized for the whole database and thus, all musical genres. Additional, for both, *MTV* and *Classic* databases a weighting function is optimized independently to be able to take the different pitch and

Method	$g_e(m)$	NF1 [%]	NP [%]	NR [%]
Constant	constant	52.15	51.76	52.56
	linear	61.89	60.11	63.77
	quadratic	56.00	52.55	59.94
	optimized	**62.02**	**60.90**	**63.19**
General	constant	55.98	61.94	51.08
	linear	60.33	56.65	64.53
	quadratic	44.24	33.55	64.93
	optimized	**61.56**	**62.00**	**61.13**
Estimation	constant	53.36	52.36	54.41
	linear	61.49	60.78	62.22
	quadratic	55.46	55.31	55.61
	optimized	**62.38**	**61.73**	**63.04**

Table 4.2: Results of the weighting function $g_e(m)$ optimization process for all three methods, with the initial weightings shown in Figure 4.4 and the optimized shown in Figure 4.5.

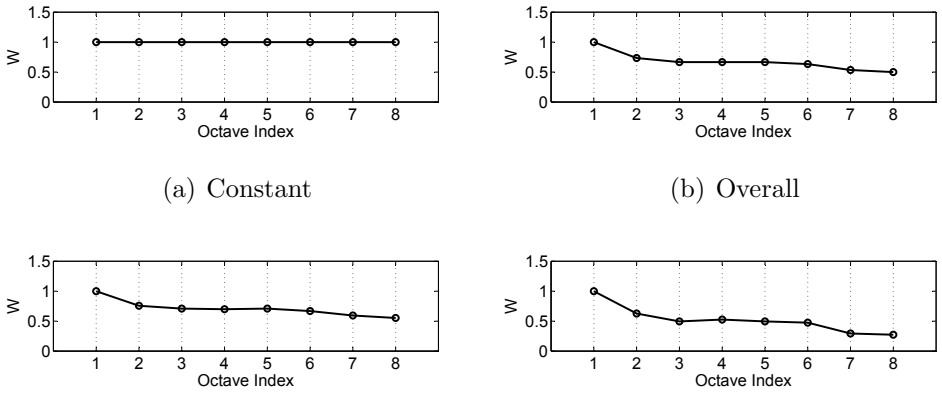
energy distributions, especially in the lower frequency regions, into account. Again, the overall goal of this weighting component is to prevent errors of frequency halving or doubling.

The overall influence of $g_e(o(n))$ shown in Table 4.3 is rather small compared to the influence of the component estimation weight function $g_e(m)$. The spectral energy distribution seems to be corrected sufficiently by the dB(A) filtering in the preprocessing step. Using an independent optimized weighting for Pop and classical music does further enhance the performance.

The lowest octave is suppressed to much by the dB(A) weighting and is thus,

$g(o(n))$	NF1 [%]	NP [%]	NR [%]
constant	62.38	61.73	63.04
overall	62.43	61.56	63.33
adapt	**62.68**	**62.02**	**63.36**

Table 4.3: Results of the F0 weighting function $g_e(o(n))$ optimization for the different functions shown in Figure 4.6. *adapt* indicates using the independently optimization for MTV and Classic.



(a) Constant (b) Overall

(c) Classic (d) MTV

Figure 4.6: Different octave index weighting functions $g_e(o(n))$. Starting point *Constant*, global optimized *Overall*, and optimized for *MTV* and *Classic* independently.

enhanced in each of the weighting function here. Especially in Rock/Pop music the low frequencies get a higher weight.

This weighting component is needed to allow the detection of notes in each frequency region and more importantly, to prevent octave errors introduced by the F0 estimator. The overall decrease of the weight from lower to higher frequencies is caused by the tendency to introduce errors doubling the actual frequency. The optimized weighting function try avoid this by applying a higher weight to lower frequencies.

The actual numbers of the weighting functions are mentioned in Table A.1.

Steadiness weight: The last component of the estimation weighting function $g_e(n, m)$ defined in Section 3.1.2 is the weight $g_e(\varsigma)$ according to the local steadiness $\varsigma_l[n]$ of the signal introduced in Section 2.3.4 at the position of each harmonic component. Thus, this is the only component influenced by the temporal structure in the framewise F0 estimation process.

Note that σ is an exponent in Equation 3.4, so $\sigma = 0$ results in no steadiness weighting at all and leaves only the magnitude of each component for candidate estimation.

Enhancing the estimation weighting process with the local steadiness representation of each component leads, as shown in Table 4.4, to better transcription results. Especially notes with low amplitude but clearly stationary signal slope are enhanced with this method, resulting in a higher recall rate.

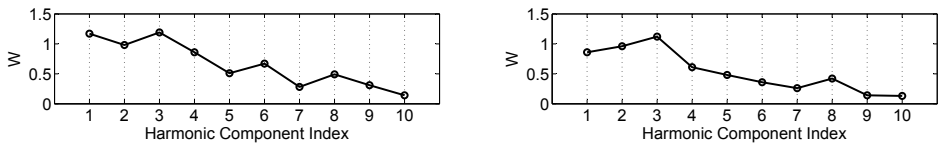
σ	NF1 [%]	NP [%]	NR [%]
0	61.87	63.38	60.42
1	62.09	63.74	60.52
2	62.36	63.41	61.34
5	**62.68**	**62.06**	**63.31**

Table 4.4: Selected results for the steadiness component weighting $g_e(\varsigma)$ of the estimation weight function $g_e(n, m)$.

Due to the RASTA preprocessing of the semitone spectrogram proposed in Section 2.2.2 that effectively suppresses transient signal parts and enhances the stationary partial within the signal, the overall gain in terms of recognition rate achievable with the steadiness weighting is rather low.

Spectrum subtraction optimization

The optimization of the synthesis function used to subtract the estimated spectrum of each found candidate is again carried out as described in Section 4.3.1. Due to the doubly estimation character of the spectrum estimation method explained in Section 3.1.3.3, no special synthesis weight function is created, instead, the optimized estimation weight function is used in the synthesis step, too.



(a) $g_{sc}(m)$ for *constant* spectrum removal (b) $g_{sg}(m)$ for *general* spectrum removal

Figure 4.7: Optimized synthesis weights of each component for the candidate subtraction $g_s(m)$, for the two methods using constant and general spectrum removal.

The overall influence of the subtraction weight functions on the recognition results is rather small, compared to the influence of the different functions in the estimation step.

Again, the linear function shows the best initial results, resulting in less subtraction of the higher harmonics, with the optimized function leading to slightly better results.

Method	$g_s(m)$	NF1 [%]	NP [%]	NR [%]
Constant	constant	60.19	63.22	57.43
	linear	61.93	60.15	63.82
	quadratic	61.60	62.63	60.61
	optimized	**61.96**	**60.93**	**63.02**
General	constant	61.30	61.57	61.04
	linear	61.44	61.40	61.49
	quadratic	61.46	63.14	59.86
	optimized	**61.56**	**62.00**	**61.13**

Table 4.5: Results of the synthesis weighting function $g_s(m)$ optimization process for the candidate spectrum removal of the two methods, with the weights shown in Figures 4.7 and 4.4.

The synthesise weight of the general spectrum removal shows the approximation of a general spectral shape that is common between each note played by any instrument. $g_{sg}(1)$ is the anchor point and all other components are relatives to this base.

The subtraction factor s_f of the constant spectrum removal method in Equation 3.8 was found to be $s_f = 0.29$, to enforce the desired behaviour of underestimation of the removed spectral components.

The actual numbers of the weighting functions are mentioned in Table A.1.

Termination Condition

The termination condition of the iterative F0 estimation algorithm does have a great influence on the recognition results. Less on the actual accuracy expressed by the NF1 measure, but huge impact on the note precision and recall rates.

A relaxed termination condition results in many note insertions and thus, in a high recall but low precision rate, and vice versa. It can be used to adjust the overall recognition rate to similar recall and precision rates, leading to a maximum NF1 measure. A different parameter γ is used for the three methods displayed in Table 4.6, because the dependency on the candidate signals of the termination condition expressed in Equation 3.15 necessitates different conditions for the different methods.

The overall influence of γ is shown in Figure 4.8, a precision / recall diagram for the spectrum estimation method.

Method	γ
Constant	0.8
General	0.47
Estimation	0.7

Table 4.6: Optimized termination condition parameter γ for the three different types of candidate cancellation.

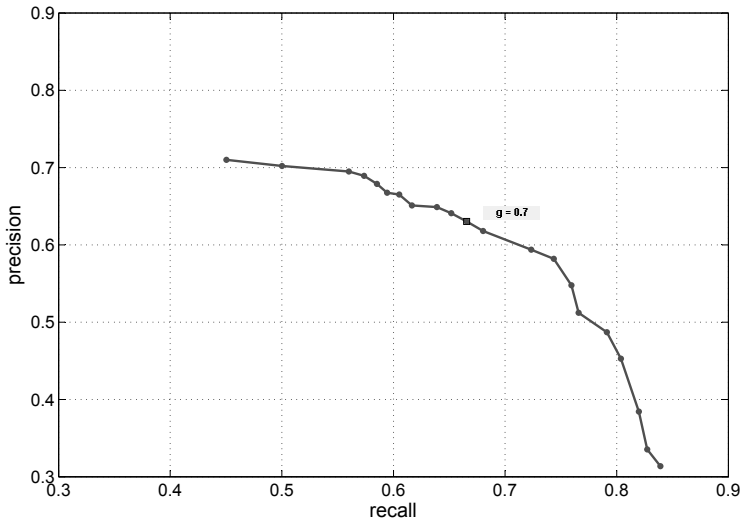


Figure 4.8: Precision/recall diagram for the spectrum estimation method, for γ in the range $0 < \gamma < 1$ in steps of 0.05.

Method	NF1 [%]	NP [%]	NR [%]	OR [%]	O(n)
Constant	62.02	60.90	63.19	21.99	2.09
General	61.75	63.98	59.68	20.93	1.00
Estimation	**62.68**	**62.02**	**63.36**	21.63	2.24

Table 4.7: Overall F0 estimation performance of the optimized methods, together with the overlap ratio OR and the computational complexity $O(n)$ relative to the general spectrum removal method.

A relaxed termination condition leads to high recall rates and low precision rates, through falsely accepted notes. The maximum recall rate is limited by the maximum polyphony of the F0 estimation method of $p = 12$.

This connection is used in Section 4.3.2 to adapt the F0 estimator to the different postprocessing methods of on- and offset correction, with influence on the precision and recall rates.

Method Comparison

While all of the methods are able to provide similar results, the spectrum estimation method is able to outperform the other two methods that do only rely on some general assumptions about the spectral components of the detected note, due to taking the overall spectral structure of the signal of all present notes and not only of the detected note into account.

The estimation method leads to lower corruption of simultaneously sounding notes. But at the same time, this method is computational the least effective, resulting in higher calculation times.

The overall recognition results are displayed in Table 4.7 together with the relative computational complexity $O(n)$.

The mean overlap ratio OR between correctly transcribed and reference notes is rather low, caused by the steadiness weighting of the estimation step that suppresses transient signal parts in the onset and tends to follow each note to the end of the release. The detected F0 partials tend to have a delayed onset and release. In addition, the F0 estimator is optimized for high note recognition rates (NF1), not for high temporal accuracy.

The task of the following onset and duration detection is to enhance the overlap ratio between transcribed and reference notes, by correcting the onset and especially offset of each of the notes detected by the F0 estimator. The spectrum estimation method is used in the following, due to the superior

Feature	NF1 [%]	NP [%]	NR [%]	OR [%]
Pitch	62.68	62.02	63.36	21.63
Magnitude	63.80	64.88	62.76	27.36
Transiency	**63.91**	**64.79**	**63.05**	**27.80**

Table 4.8: Comparison of the different features used for detecting the onset with the simple correction method. No offset correction is applied.

recognition performance.

4.3.2 Onset and Duration Detection

For the onset detection tasks, the detection functions or features described in Section 3.2.2 need to be evaluated for each of the decision functions.

Pitch defines the outcome of the F0 estimator, the onset is simply defined at the beginning of a detected partial, and the offset respectively. *Pitch strength* describes the salience $s[n]$ of the detected partial, *magnitude* is the summed magnitude $\Delta Z_n^*(l)$ in the harmonic subbands and *transiency* the complex domain rising transiency description $\Gamma_n(l)^+$.

No correction of the offset of the notes is applied here, i.e. pitchbased offset detection is used.

Simple Onset Correction

The simple correction of notes' onsets assigns the beginning of each detected F0 partial to the time with maximum signal change in a window, as described in Section 3.2.1.1.

The simple correction of the notes' onsets can only slightly enhance the overlap ratio of the notes, because the onset is already quite exactly determined by the F0 estimator, indicated by the *pitch* method in Table 4.8. The *pitch strength* is not shown here, because there is obviously no difference to the *pitch* method.

At the same time, the correction of the onset leads to slightly higher note precision and recall values, due to earlier detected onsets. Especially very short notes - without any clear steady state, but a present release portion - that would be detected to late and thus, falsely transcribed, are now treated as correct.

Using the complex domain transiency description of the signal, which com-

Feature	NF1 [%]	NP [%]	NR [%]	OR [%]
Pitch	62.68	62.02	63.36	21.63
Magnitude	62.78	67.03	59.04	27.45
Transiency	**63.16**	**67.04**	**59.23**	**27.86**

Table 4.9: Comparison of the different features used for detecting the onset with the peak picking algorithm. No offset correction is applied.

bines magnitude and phase information, enhances the accuracy of the onset detection, compared to the magnitude only method and especially to the outcome of the F0 estimator, both, in terms of note precision and notes' overlap ratio.

Peak Picking Onset Detection

The median filter based peak picking algorithm assigns a onset to a note, whenever a peak is detected within a window around the beginning of the detected F0 partial, as described in Section 3.2.1.2.

Using peak picking algorithms leads to different results compared with the simple correction in Table 4.8, because of the second condition introduced to note detection.

Each by the F0 estimator detected note is only accepted, when a onset peak is detected. This leads to a higher note precision rate, by discarding erroneous candidates, but at the same time to a lower recall rate, because true candidates are rejected.

Again, the complex domain transiency description enhances the recognition results, compared with the magnitude based detection as shown in Table 4.9.

Offset Detection

The results obtainable with the different features applied to the threshold based offset detector introduced in Section 3.2.1.3 are shown in Table 4.10.

Pitch again notes the direct outcome of the F0 estimator and is given for comparison. The *pitch strength* based offset detection can only detected earlier offset by cutting off the release part of the note. It is not able to lengthen a note, when the estimated F0 partial stops early. Hence, the enhancement in terms of the overlap ratio of the notes are rather small. The envelope based offset detection allows to lengthen and shorten detected notes

Feature	NF1 [%]	NP [%]	NR [%]	OR [%]
Pitch	63.91	64.79	63.05	27.80
Pitch strength	63.43	64.34	62.54	31.80
Magnitude	**60.97**	**66.46**	**56.32**	**40.61**

Table 4.10: Comparison of the different features used for detecting the offset of the note using thresholding. Simple correction is used for onset detection.

Method	NF1 [%]	NP [%]	NR [%]	OR [%]
optimized ($\gamma = 0.7$)	**62.68**	62.02	63.36	21.63
relaxed ($\gamma = 0.5$)	62.33	56.75	69.13	24.43
corrected optimized	**60.97**	66.46	56.32	40.61
corrected relaxed	58.24	59.12	57.37	41.16
peak optimized	57.63	66.99	50.57	39.79
peak relaxed	**57.87**	59.57	56.27	40.06

Table 4.11: Comparison of the results of the on- and offset detection methods, together with an adjustment of the optimized (*opt*) F0 estimator's termination condition to allow higher recall rates (*relaxed*). *corrected* denotes the simple correction method, *peak* the peak picking method. Magnitude based offset correction is applied.

and produces higher overlap ratios.

By combining notes' partials that are falsely separated and thus, are leading to two or more detected notes, the precision rate is enhanced by the offset detection. But at the same time, by falsely combining successive notes, the recall rate is reduced, too.

Method Comparison and Relaxation the Termination Condition

Table 4.11 shows the overall results of the different methods of onset detection. Due to the enhancing of the precision rate of the on- and offset correction, the note recall rate is reduced. With the F0 estimator being optimized for roughly equal NP and NR rates this leads to a discrepancy between these two. To again achieve NP and NR rates on par, the termination condition of the F0 estimator is relaxed.

Relaxing the termination condition and allowing the detection of *more* notes,

the outcome of the F0 estimator leads to a higher recall rate at the cost of a lower precision, by falsely inserting detected notes (see first two rows in Table 4.11). By simply correcting these notes in terms of on- and offset (middle two rows in Table 4.11), the overall note detection accuracy expressed through the NF1 measure, can not be enhanced. While the overall note recall rate is higher in case of the relaxed termination condition, the distinct decrease of the note precision rate leads to an overall reduced F1 measure in the simple correction case.

Using the median filter based peak picking algorithm for notes' onset detection the relaxed termination condition is beneficial, as shown in the bottom two rows of Table 4.11, due to the ability of this method to discard F0 estimates, when no onset is detected. This second notes' detection condition enhances the low precision rate at the cost of a lower recall rate. But still, the simple correction of notes' onsets together with the thresholding offset detection achieves better results, both, in terms of note detection accuracy and temporal overlap.

4.3.3 Note Event Modeling

The note event modeling approach uses hidden markov models for probabilistic modeling of notes with their distinct event, starting with the onset and lasting until the end of the release state.

Two different methods are proposed here, one rather simple with limited feature sets to detect the on- and offset of note detected by the F0 estimator, and a more complex one, which uses the outcome of the F0 estimator as harmonic structure description and finds the note event based on a larger feature set.

The evaluation is carried out using a 3-fold cross validation, separating the whole database in three song disjunct, roughly equal sized parts. For each of the three testfolds, the remaining two parts are used for training, taking care, that no song is present in different folds.

Event Modeling for On- and Offset Detection

The onset detection with HMM uses simple bandwise features to model the temporal slope of a note.

For training, only notes are used that are detected by the F0 estimator and additional features describing the signal slope are used to detect the beginning and the end of each note.

Feature set	NF1 [%]	NP [%]	NR [%]	OR [%]
SZ	55.43	71.35	45.32	37.68
ST	58.60	69.64	50.58	39.32
SZT	58.98	70.21	50.84	39.74
SZT'	**59.31**	**70.25**	**51.56**	**41.24**
SZTT'	59.16	69.82	51.32	41.21

Table 4.12: Comparison of the feature sets used for note on- off segmentation with 2 GMM components for each state.

Feature groups in the subband n:

- S : salience $s[n]$ and $\Delta s[n]$
- Z : harmonic magnitude $Z^*[n]$ and $\Delta Z^*[n]$ of harmonic bands
- T : transiency description $\Gamma[n]$ and $\Delta\Gamma[n]$ of harmonic bands
- T' : rising and descending transiency description $\Gamma^+[n]$, $\Gamma^-[n]$ and Δ of harmonic bands

These groups are used to form the distinct features sets in Table 4.12.

Estimating the on- and offset of a note by regarding the whole event of the note shows good results in term of overlapping, i.e. temporal accuracy of the notes.

Similar to the other onset detection methods, a high precision rate is achieved, but at the same time the recall rate is lowered, due to rejection of falsely detected F0, though, correct detected F0 candidates are discarded, too.

Compared with the results of the onset and offset detection in Table 4.11, the overall NF1 measure is slightly decreased and the mean overlap ratio is enhanced. The note event modeling is superior for notes with long duration, while very short note events with no distinct slope of the features are falsely discarded.

Relaxation of the Termination Condition To bypass the limitation on the overall NF1 measure caused by the low recall rate, the termination condition is again lowered as shown previously for the onset detection in Table 4.11. This results in a higher recall rate of the F0 estimator that can be corrected through the note rejection capability of the note event modeling approach.

Feature set	NF1 [%]	NP [%]	NR [%]	OR [%]
SZT' optimized	59.31	70.25	51.56	41.24
SZT' relaxed	60.13	65.43	55.62	40.96

Table 4.13: Results for onset detection focused note event modeling with relaxed termination condition of the F0 estimator and 2 GMM per state.

The relaxation of the termination condition allows a higher recall rate (Table 4.13), with at the same time high note precision. This leads to higher overall NF1 measures, compared to the outcome of the optimized F0 estimator in Table 4.12.

Compared with the on- and offset detection methods based on signal properties in Table 4.11, these results are slightly worse. The even modeling shows superior results for rather long notes, but has disadvantages in the detection of short note events.

Event Modeling for Note Detection

The note event modeling for onset detection still depends greatly on the F0 estimator. Notes are only detected, where a F0 candidate was found.

To carry on the process started with the relaxation of the termination condition to enhance the recall rate in favour of a lower precision, the termination condition is now completely discarded, resulting in 12 F0 estimates in each time frame.

This ensures that most of F0 partials are detected in the signal and the note event modeling gets the ability to detect the notes from this information, and no longer directly depends on the outcome of the F0 estimator. Therefore, additional features are necessary, because the bandwise features carry no information about the harmonic structure.

In addition to the features of each subband and the harmonic bands, now the subharmonic band amplitudes Z' are used, too. This allows the discrimination between an event caused by the fundamental frequency and an event caused by some harmonic component.

Finally, note precision, recall and F1 measure are on par and the precision enhancing character of the note event modeling approach is almost compensated by the very high recall rate of the unterminated F0 estimator. The overall detection rates are slightly better than those from the onset detection focused modeling approach shown in Figure 4.13.

Feature set	NF1 [%]	NP [%]	NR [%]	OR [%]
SZZ'T'	60.62	63.27	58.19	42.43

Table 4.14: Results of the note detection through note event modeling, with the number of GMM = 4.

Method	g	NF1 [%]	NP [%]	NR [%]
	0	61.64	62.23	61.06
scale	**0.2**	**61.82**	**62.74**	**60.93**
chord	0.05	61.79	62.52	61.07

Table 4.15: Comparison of the two methods used for suppression of chromatic altered notes with the unenhanced F0 estimator.

4.3.4 Musical Knowledge

In this Section, the results achieved in the preceding chapte, based on signal properties and probabilistic methods, are enhanced using further knowledge, or song-level parameters.

Musical Key

The musical key of a song is used as knowledge source for the notes which are most likely present in a song. The salience of all chromatically altered notes are suppressed in the estimation weight function of the F0 estimation step, as explained in Section 3.3.1.

The suppression factor is $1 - g$ for the scale and chord based approach, while in the chord based approach the notes without chromatic alteration are enhanced to $1 + g$ and $1 + 2 \cdot g$, depending on the frequency of occurence in the chords of the scale.

The results are shown in Table 4.15 in comparison to the unchanged F0 estimator ($g = 0$). While both methods achieve similar results, none of them is able to outperform the unenhanced F0 estimator significantly.

Looking at the functionality of the fundamental frequency estimation this is comprehensible, as the process of summing the harmonic amplitudes tends to introduce octave errors, i.e. F0s found an octave to high or to low. This behaviour is clearly visible in Figure 3.2, where an isolated note leads to multiple peaks in the candidate spectrum at harmonic and subharmonic fre-

Feature set	NF1 [%]	NP [%]	NR [%]	OR [%]
SZZ'T'	60.62	63.27	58.19	42.43
SZZ'T' + beat	60.78	66.73	55.82	42.53

Table 4.16: Comparison of the results of the note detection through note event modeling and the enhancement through the cosine modeled beat position.

quencies.

These octave errors can of course not be corrected by the musical key based suppression of chromatic altered notes, because the octave multiples of a note are assigned to the same scale.

Musical Meter

The beat position obtained from the musical meter estimation is used to enhance the temporal precision of the detected notes as described in Section 3.3.2. Two different methods are proposed: using the cosine modulated beat position as additional feature for the note event modeling approach, as well as a quantization of the onset and duration times to the nearest fraction of a quarter beat position.

Beat position for Note Event Modeling: The note detection focused note event modeling described in Section 3.2.3.5 is enhanced with the cosine modulated beat position.

The achievable results are shown in Table 4.16 in comparison with the optimum note event method from Table 4.14.

The results in terms of note precision and temporal accuracy are slightly enhanced through the use of the cosine modeled beat position as additional feature.

Note quantization: Table 4.17 shows the results achievable through the quantization of notes' onsets and durations to the nearest fraction of a quarter beat position for the simple onset correction method.

The quantization is only beneficial for slight quantization, compared with the results of unaltered simple correction method in the first row. Larger quantization leads to worse results, due to falsely shifting the temporal position of the detected note.

Fraction	NF1 [%]	NP [%]	NR [%]	OR [%]
0	60.97	66.46	56.32	40.61
1/8	58.07	63.57	53.45	38.42
1/16	61.04	65.35	57.26	40.23
1/32	61.25	66.68	56.64	41.42

Table 4.17: Quantization of notes' on- and offset to nearest fraction of estimated quarter beat position.

Chapter 5

Conclusion and Outlook

5.1 Conclusion

A complete music transcription system is presented in this work that is able to transcribe real audio musical recordings of different types of musical genres.

The database used for evaluation of the methods was created through force alignment of reference transcriptions obtained from midi files to the real audio recordings. This is achieved through chroma feature based dynamic time warping.

Effective signal representations are presented that are able to model both, spectral and temporal properties of musical signals, through the use of a semitone spectrogram, as well as a complex domain signal stationarity description. Additionally, RASTA preprocessing to suppress unharmonic components, especially drums, in the complex signal. The time/frequency resolution problem is bypassed through the use of a multiresolution fourier transform in combination with the estimation of the instantaneous frequency using the phase vocoder method.

The system proposed in this work covers each step that is necessary to obtain a note-level parameter representation of a musical recording: polyphonic pitch estimation as well as onset and duration detection of the notes. The multiple F0 estimator is based on an iterative approach that sums harmonic amplitudes to find F0 candidates and harmonic signal estimation to subtract candidates from the signal. The detection of notes' on- and offsets with high accuracy is based on the complex domain signal description.

Probabilistic models are utilized through Hidden Markov Models to model the distinct event created by each note following ideas from speech recognition, using pitch and temporal information. This approach has advantages

especially with note events of long durations, but performs slightly worse in case of short events.

High level musical knowledge in terms of musical key and meter estimation is used to enhance the transcription results, though, the overall gain in transcription accuracy is rather low.

Overall note F1 measures over 60% are achieved and leave room for enhancements of the F0 estimation process, and the mean overlap ratio between reference and transcribed notes of roughly 40% can still be enhanced.

5.2 Outlook

Although the system presented in this work includes each step necessary to transcribe musical recordings, each of them is still not solved satisfyingly.

Especially the polyphonic fundamental frequency estimation needs to be improved to allow better transcription results. Many different methods were proposed by many different research groups, but none of them can be regarded as a reference method.

Source separation that is able to extract each single instrument of a song could by beneficial for this step, even a simple drums separation would reduce transcription errors.

To further enhance the transcription results, information is needed, that is able to analyse the whole structure of a song. While the note event modelling approach presented in this thesis detects notes in its local context, no information of the structure, or melody of the song is used. Introducing a *musicological* model, following the language model idea in speech recognition, with knowledge of note transition probabilities could solve this drawback.

This is especially important, because of the tendency of the F0 estimator to introduce octave errors that are not detectable by using the musical key alone. Knowledge about probable note progression could bypass this problem.

The last step to a complete transcription system would be to create a traditional musical score out of the parameter representation. While this is mainly a cosmetic task, this would be necessary to make a *real* transcription system.

Most of the information needed for the generation of a musical sheet has already been generated: notes with pitch, onset and duration, musical meter, tempo and beat position, musical key and chords recognition, and would just needed to be combined.

To complete the set of note-level parameters to describe each note explicitly

an estimation of the dynamics, or *loudness* of the note has to be found, as well as an instrument recognition to determine the instrument playing the note.

The still not solved task and lack of a widely accepted reference method for automatic music transcription offers a wide area of research with many possibilities to further developments and ideas.

Appendix A

Detailed Parameter Evaluation

m	1	2	3	4	5	6	7	8	9	10
quad	1.00	0.50	0.33	0.25	0.20	0.17	0.14	0.13	0.11	0.10
lin	1.00	0.90	0.80	0.70	0.60	0.50	0.40	0.30	0.20	0.10
$g_{ec}(m)$	0.84	0.82	0.68	0.70	0.83	0.63	0.52	0.28	0.32	0.12
$s_{eg}(m)$	0.86	0.96	1.12	0.61	0.48	0.36	0.26	0.42	0.14	0.13
$g_{ee}(m)$	1.03	0.75	0.82	0.58	0.62	0.40	0.42	0.22	0.22	0.02
$g_{sg}(m)$	0.91	0.82	0.62	0.40	0.67	0.59	0.52	0.24	0.30	0.04
$s_{sc}(m)$	1.17	0.98	1.19	0.86	0.51	0.67	0.28	0.49	0.31	0.14
$g_e(o(n))$	1.50	1.10	1.00	1.00	1.00	0.95	0.80	0.75		
$g_e(o(n))_{clas}$	1.72	1.30	1.22	1.20	1.22	1.15	1.02	0.95		
$g_e(o(n))_{mtv}$	0.99	0.62	0.49	0.52	0.49	0.47	0.29	0.27		

Table A.1: Detailed values of the weighting functions used in the F0 estimator.

dB	NF1 [%]	NP [%]	NR [%]	OR [%]
-54	63.45	62.77	64.41	21.17
-49	63.46	62.80	64.13	21.18
-44	63.49	62.87	64.12	21.18
-39	63.52	62.95	64.10	21.20
-34	**63.57**	**63.11**	**64.04**	**21.20**

Table A.2: Selected results of the silence detection parameter.

σ	NF1 [%]	NP [%]	NR [%]
0	61.87	63.38	60.42
0.5	61.97	63.92	60.13
1	62.09	63.74	60.52
1.5	62.23	63.61	60.90
2	62.36	63.41	61.34
3	62.53	62.93	62.13
4	62.66	62.52	62.80
5	**62.68**	**62.06**	**63.31**

Table A.3: Selected results for the steadiness component weighting $g_e(\varsigma)$ in the estimation weight function.

s_f	NF1 [%]	NP [%]	NR [%]
0.13	61.44	63.70	61.93
0.21	62.10	62.26	64.62
0.22	62.13	62.11	64.84
0.27	62.29	61.45	65.89
0.29	**62.34**	**61.25**	**66.20**
0.30	62.30	61.12	66.27
0.34	62.29	60.78	66.64
0.36	62.27	60.69	66.74
0.39	62.28	60.56	66.92
0.40	62.27	60.51	66.94

Table A.4: Selected results for the subtraction factor s_f of the constant spectrum removal method.

g	NF1 [%]	NP [%]	NR [%]
0	62.68	62.02	63.36
0.05	62.78	62.20	63.38
0.1	62.86	62.38	63.36
0.125	62.90	62.43	63.35
0.15	62.92	62.52	63.32
0.175	62.92	62.55	63.30
0.2	**62.92**	**62.61**	**63.24**
0.225	62.89	62.62	63.16
0.25	62.88	62.65	63.11
0.3	62.78	62.64	62.91
0.35	62.62	62.62	62.84

Table A.5: Selected results of the scale based pattern suppression of chromatically altered notes.

g	NF1 [%]	NP [%]	NR [%]
0	61.64	62.23	61.06
0.05	**61.79**	**62.52**	**61.07**
0.1	61.76	62.70	60.85
0.15	61.51	62.67	60.40
0.2	61.10	62.45	59.81
0.25	60.48	62.00	59.02
0.3	59.63	61.36	58.00
0.35	58.68	60.54	56.92

Table A.6: Selected results of the chord based pattern suppression of chromatically altered notes.

w [sec]	NF1 [%]	NP [%]	NR [%]	OR [%]
0.05	55.75	68.71	46.90	40.13
0.1	56.94	68.05	48.94	39.89
0.15	57.50	67.61	50.02	39.68
0.2	57.68	67.26	50.49	39.73
0.25	**57.71**	**67.08**	**50.64**	**39.75**
0.3	57.62	66.91	50.60	39.78

Table A.7: Selected results of different window length for the simple onset correction.

p	NF1 [%]	NP [%]	NR [%]	OR [%]
0.1	50.06	68.29	39.52	35.34
0.15	51.89	67.86	42.01	37.61
0.2	53.40	67.52	44.17	38.97
0.25	54.72	67.29	46.10	39.75
0.3	55.86	67.13	47.83	40.10
0.4	**57.62**	**66.91**	**50.60**	**39.78**
0.5	56.43	64.41	51.40	38.68

Table A.8: Selected results of different thresholds for the offset correction.

γ	NF1 [%]	NP [%]	NR [%]	OR [%]
1.0	57.62	66.91	50.60	39.78
1.1	**57.63**	**66.99**	**50.57**	**39.79**
1.2	57.62	67.07	50.51	39.79
1.3	57.58	67.14	50.41	39.81
1.4	57.50	67.21	50.24	39.84
1.5	57.40	67.27	50.05	39.84
1.6	57.27	67.34	49.82	39.86
1.7	57.12	67.44	49.54	39.88
1.8	56.92	67.50	49.21	39.93

Table A.9: Selected results of the parameter γ of the peak picking onset detection.

ρ	NF1 [%]	NP [%]	NR [%]	OR [%]
0.000	57.66	65.97	51.21	39.52
0.001	57.67	66.05	51.18	39.56
0.002	57.70	66.15	51.16	39.59
0.003	**57.70**	**66.24**	**51.12**	**39.62**
0.005	57.71	66.44	51.00	39.69
0.007	57.69	66.66	50.85	39.74
0.008	57.67	66.77	50.75	39.77
0.009	57.64	66.88	50.64	39.81
0.010	57.63	66.99	50.57	39.79
0.011	57.56	67.08	50.41	39.87
0.012	57.53	67.20	50.30	39.89
0.013	57.48	67.31	50.16	39.91
0.015	57.39	67.55	49.89	39.97
0.017	57.26	67.77	49.57	40.03
0.019	57.12	67.99	49.25	40.12

Table A.10: Selected results of the parameter ρ of the peak picking onset detection.

GMMs	NF1 [%]	NP [%]	NR [%]	OR [%]
1	56.47	77.46	44.43	43.55
2	**59.31**	**70.25**	**51.56**	**41.24**
4	58.67	68.43	51.35	40.56

Table A.11: Selected results of different number of gaussian mixture components for the onset detection focused note event modeling with feature set SZT'.

GMMs	NF1 [%]	NP [%]	NR [%]	OR [%]
2	60.03	62.87	57.43	41.37
4	**60.62**	**63.27**	**58.19**	**42.43**
8	58.63	60.95	56.48	40.06

Table A.12: Selected results of different number of gaussian mixture components for the note detection focused note event modeling with feature set SZZ'T'.

Appendix B

Database

Name	Testfold
4 Non Blondes - What's Up	1
Ace Of Base - All That She Wants	1
Alanis Morrisette - Ironic	1
All 4 One - I Swear	1
Backstreet Boys - Everybody	1
Aerosmith - Cryin'	1
Backstreet Boys - We've Got It Goin' On	1
Bon Jovi - It's My Life	1
Cher - The Shoop Shoop Song (It's In His Kiss)	1
Depeche Mode - It's No Good	1
Die Aerzte - Schundersong	1
Bon Jovi - Always	1
Die Aerzte - Schrei nach Liebe	1
Eminem - Stan	1
Die Toten Hosen - Zehn Kleine Jaegermeister	1
Faith No More - I'm Easy	1
Cranberries - Zombie	1
Die Aerzte - Maenner sind Schweine	2
Guns N Roses - November Rain	2
HIM - Join me	2
Madonna - Music	2
Natalie Imbruglia - Torn	2
Crash Test Dummies - Mmm Mmm Mmm Mmm	2
Fugees - Killing Me Softly	2
Jennifer Lopez - If You Had My Love	2

Lenny Kravitz - Fly Away	2
Metallica - Nothing Else Matters	2
Oasis - Wonderwall	2
Pet Shop Boys - Go West	2
Puff Daddy - I'll Be Missing You	2
Queen - The Show Must Go On	2
Roxette - It Must Have Been Love	2
Green Day - Basket Case	2
Madonna - Frozen	2
No Doubt - Don't Speak	3
R Kelly - I Believe I Can Fly	3
REM - Losing My Religion	3
Red Hot Chili Peppers - Californication	3
Shaggy - Boombastic	3
Rammstein - Engel	3
Robbie Williams - Rock DJ	3
Roxette - Joyride	3
Take That - Back For Good	3
Take That - Relight My Fire	3
Red Hot Chili Peppers - Under the Bridge	3
Santana - Maria Maria	3
Spice Girls - Wannabe	3
TLC - Waterfalls	3
UB 40 - Kingston Town	3
Ugly Kid Joe - Cats In The Cradle	3

Table B.1: Filelist of the MTV database.

Name	Testfold
Grieg - Morgenstimmung (aus Peer Gynt)	1
Bizet - Les Toreadors (aus Carmen)	1
Mozart - Eine kleine Nachtmusik (1.Satz, Allegro)	1
Bach - Air (aus Ouvertuere Nr. 3)	1
Offenbach - Barcarole (Hoffmanns Erzaehlungen)	1
Beethoven - Fuer Elise	1
Tschaikowsky - Danse espagnole (aus Schwanensee)	1
Mozart - Tuerkischer Marsch	1
Schumann - Von fremden Laendern und Menschen	1
Bizet - Danse Boheme (aus Carmen)	1

Bach - Brandenburgerisches Konzert Nr. 2	1
Beethovens - Mondscheinsonate (1. Satz, Adagio sostenuto)	1
Wagner - Walkuerenritt (aus Die Walkuere)	1
Haendel - Alla Hornpipe (aus Wassermusik)	1
Vivaldi - Mandolinenkonzert (1. Satz, Allegro)	1
Beethoven - Symphonie Nr. 5 (1. Satz, Allegro con brio)	1
Pachelbel - Kanon D-dur	1
Mozart - Sonata facile (1. Satz, Allegro)	2
Schumann - Abschied (aus Waldszenen)	2
J. Strauss (Sohn) - Leichtes Blut	2
Tschaikowsky - Barcarole (aus Les saisons)	2
Liszt - Liebestraum Nr. 3 As-Dur	2
Mendelssohn - Hochzeitsmarsch (aus Ein Sommernachtstraum)	2
Mozart - Klavierkonzert Nr. 21 (2. Satz, Andante)	2
Tschaikowsky - Danse napolitaine (aus Schwanensee)	2
Beethoven - Klaviersonate Nr. 8 Pathetique (2. Satz, Adagio cantabile)	2
Dvorak - Slawischer Tanz Nr. 1 C-dur	2
Schumann - Traeumerei	2
Mendelssohn - Lied ohne Worte E-Dur op.19,1	2
Mozart - Klarinettenkonzert (2. Satz, Adagio)	2
Chopin - Mazurka h-moll op. 30,2	2
Schumann - Frhlicher Landmann	2
Mahler - Adagietto (Symphonie Nr. 5)	2
Mendelssohn - Symphonie Nr. 3 Schottische (Vivace non troppo)	2
Schumann - Fruehlings Symphonie (3. Satz, Scherzo)	3
Tschaikowski - Chanson triste	3
Grieg - An den Fruehling	3
Brahms - Intermezzo a-moll op. 76,7	3
Grieg - Anitras Tanz (aus Peer Gynt)	3
Tschaikowsky - Pas de quatre. Adagio (aus Dornroeschen)	3
Bizet - Farandole (aus L'Arlesienne)	3
Grieg - Arietta	3
Liszt - Klavierkonzert Nr. 2 (2. Satz, Larghetto)	3
Tschaikowsky - Mazurka (aus Schwanensee)	3
Grieg - Gavotte (aus Holbergs Zeit)	3
Gounod - Faust (Ballettmusik, Allegro vivo)	3
Mozart - Klavierkonzert Nr. 23 (2. Satz, Andante)	3
Puccini - Summ-Chor(aus Madame Butterfly)	3

Beethoven - Klavierkonzert Nr. 4 (3. Satz, Rondo Vivace)	3
J. Strauss (Sohn) - Rosen aus dem Sueden, Walzer op.388	3

Table B.2: Filelist of the classical music database.

List of Tables

List of Figures

List of Abbreviations

ADSR	Attack Decay Sustain Release	23
AMT	Automatic Music Transcription	1
bpm	Beats per Minute	12
DFT	Discrete Fourier Transform	13
DTW	Dynamic Time Warping	63
DWT	Discrete Wavelet Transform	5
F0	Fundamental Frequency	31
FFT	Fast Fourier Transform	14
GMM	Gaussian Mixture Model	48
HMM	Hidden Markov Model	7
IF	Instantaneous Frequency	15
MRFT	Multi-Resolution Fourier Transformations	14
RASTA	Relative Spectra Processing	18
SM	Similarity Matrix	63
SNR	Signal to Noise Ratio	32
STFT	Short-Time Fourier Transform	13
SVM	Support Vector Machine	7
TFR	Time-Frequency Representation	5

Bibliography

[1] MIDI Manufacturers Association. The complete midi 1.0 detailed specification. 1996.

[2] 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, 2005.

[3] J. P. Bello, G. Monti, and M. B. Sandler. Techniques for automatic music transcription. In *1st International Symposium on Music Information Retrieval (ISMIR)*, October 23-25, 2000.

[4] J. P. Bello and M. B. Sandler. Blackboard systems and top-down processing for the transcription of simple polyphonic music. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00)*, 2000.

[5] J. P. Bello and M. B. Sandler. Phase-based note onset detection for music signals. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2003.

[6] W. J. J. Boo, Y. Wang, and A. Loscos. A violin music transcriber for personalized learning. In *Proceedings of the 2006 IEEE International Conference on Multimedia and Expo (ICME)*, pages 2081–2084, 2006.

[7] J. C. Brown and B. Zhang. Musical frequency tracking using the methods of conventional and 'narrowed' autocorrelation. *Journal of the Acoustical Society of America*, 89:2346–2354, 1991.

[8] R. Carmona, W. Hwang, and B. Torrésani. *Practical time-frequency analysis*, volume 9 of *Wavelet Analysis and its Applications*. Academic Press Inc., San Diego, CA, 1998.

[9] C. Chafe, B. Mont-Reynaud, and L. Rush. Toward an intelligent editor of digital audio: Recognition of musical constructs. *Computer Music Journal*, 6(1):30–41, Spring 1982.

[10] N. Collins. Using a pitch detector for onset detection. In *Proceedings of ISMIR 2005, 6th International Conference on Music Information Retrieval*, pages 100–106, 11-15 September 2005.

[11] M. Davy and S. Godsill. Bayesian harmonic models for musical signal analysis. *J.M. Bernardo, Bayesian Statistics VII. Oxford University Press.*, 2002.

[12] A. de Cheveigné. Separation of concurrent harmonic sounds: Fundamental frequency estimation and a time-domain cancellation model of auditory processing. *Journal of the Acoustical Society of America*, 93:3271–3290, 1993.

[13] K. Dressler. Sinusoidal extraction using an efficient implementation of a multi-resolution FFT. In *Proceedings of Digital Audio Effects Conference 2006 (DAFX)*, Montreal, Canada, Sept. 18 - 20, 2006.

[14] C. Duxbury, J. P. Bello, M. B.Sandler, and M. Davies. A comparison between fixed and multiresolution analysis for onset detection in musical signals. In *Proceedings of the Digital Audio Effects Workshop (DAFx)*, 2004.

[15] C. Duxbury, J. P. Bello, M. Davies, and M. B. Sandler. A combined phase and amplitude based approach to onset detection for audio segmentation. In *Proceedings of the 4th European Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS-03)*, London, UK, 2003.

[16] C. Duxbury, J. P. Bello, M. Davies, and M. B. Sandler. Complex domain onset detection for musical signals. In *Proceedings of the Digital Audio Effects Workshop (DAFx)*, 2003.

[17] EAWpats. Eawpats 12 sound fonts. `http://www.freshports.org/audio/eawpats/`.

[18] F. Eyben. High level rhythmic audio features for robust music information retrieval, 2006. Bachelorthesis. Institute for Human-Machine Communication. TU München.

[19] J. L. Flanagan and R. M. Golden. Phase vocoder. In *Bell Systems Technical Journal*, volume 45, pages 1493–1509, 1966.

[20] M. Gainza, B. Lawlor, and E. Coyle. Onset detection and music transcription for the irish tin whistle. *Irish Signals and Systems Conference, Belfast, N. Ireland*, 2004.

[21] B. Gold and N. Morgan. *Speech and Audio Signal Processing: Processing and Perception of Speech and Music.* John Wiley & Sons, Inc., New York, NY, USA, 1999.

[22] B. Gollan. C or C#? musical features for data-driven recognition of key in real audio, 2007. Bachelorthesis. Institute for Human-Machine Communication. TU München. 2007.

[23] E. Gómez, A. Klapuri, and B. Meudic. Melody description and extraction in the context of music content processing. *Journal of New Music Research*, 32(1), 2003.

[24] M. Goto. A real-time music scene description system: Detecting melody and bass lines in audio signals. In *Working Notes of the IJCAI-99 Workshop on CASA*, pages 31–40, 1999.

[25] H. Hermansky and N. Morgan. RASTA processing of speech. In *IEEE Transactions on Speech and Acoustics*, volume 2, pages 587–589, October 1994.

[26] W. Hess. Algorithms and devices for pitch determination of speech-signals. *Springer-Verlag, Berlin*, 1983.

[27] Ning Hu, Roger B. Dannenberg, and George Tzanetakis. Polyphonic audio matching and alignment for music retrieval. In *Proceedings of the 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, October 19–22, 2003.

[28] International Electrotechnical Commission (IEC). Information technology - coding of audio-visual objects, part 3: Audio, section 5: Structured audio, ISO/IEC 14496-3:1999(e). 1999-3-10.

[29] International Electrotechnical Commission (IEC). Electroacoustics - sound level meters - part 1: Specifications, IEC 61672-1 ed. 1.0 b:2002. 5/29/2002.

[30] K. Kashino, K. Nakadai, T. Kinoshita, and H. Tanaka. Organization of hierarchical perceptual sounds: Music scene analysis with autonomous processing modules and a quantitative information integration mechanism. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 158–164, 1995.

[31] I. Kauppinen. Methods for detecting impulsive noise in speech and audio signals. In *14th International Conference on Digital Signal Processing, DSP*, volume 2, pages 967–970, 2002.

[32] F. Keiler and S. Marchand. Survey on extraction of sinusoids in stationary sounds. In *Proc. DAFx-02 Digital Audio Effects Conference*, pages 51–58, Hamburg, September 2002.

[33] M. Keith. A blackboard system for automatic transcription of simple polyphonic music. In *Perceptual Computing Technical Report #385, MIT Media Lab*, 1996.

[34] H. Kim, J. J. Burred, and T. Sikora. How efficient is mpeg-7 for general sound recognition? In *25th International AES Conference Metadata for Audio*, London, UK, June 2004.

[35] A. Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3089–3092, Washington, DC, USA, 1999.

[36] A. Klapuri. Means of integrating audio content analysis algorithms. In *Proceedings of the 110th Audio Engineering Society Convention*, 2001.

[37] A. Klapuri. Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. In *Proceedings IEEE Transactions on Speech and Audio Processing, 11(6)*, 2003.

[38] A. Klapuri. Multiple fundamental frequency estimation by summing harmonic amplitudes. In *Proceedings of 7th International Conference on Music Information Retrieval (ISMIR)*, pages 216–221, 2006.

[39] A. Klapuri. A perceptually motivated multiple-f0 estimation method. In *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, October 16-19, 2005.

[40] A. Klapuri. Automatic music transcription as we know it today. *Journal of New Music Research*, 33(3):269–282(14), September 2004.

[41] A. Klapuri, A. Eronen, J. Seppänen, and T. Virtanen. Automatic transcription of music. In *Symposium on Stochastic Modeling of Music, 14th Meeting of the FWO Research Society on Foundations of Music Research*, Ghent, Belgium, October 2001.

[42] A. Klapuri, T. Virtanen, A. Eronen, and J. Seppänen. Automatic transcription of musical recordings. In *Proceedings of the Consistent and Reliable Acoustic Cues Workshop*, Aalborg, Denmark, Sep. 2001.

[43] N. Kunieda, T. Shimamura, and J. Suzuki. Robust method of measurement of fundamental frequency by ACLOS: autocorrelation of log spectrum. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 232–235, Washington, DC, USA, 1996.

[44] M. Marolt. Adaptive oscillator networks for partial tracking and piano music transcription. In *Proceedings of the 2000 International Computer Music Conference*, 2000.

[45] M. Marolt and S. Divjak. On detecting repeated notes in piano music. In *Proceedings of 3rd International Conference on Music Information Retrieval (ISMIR)*, October 13–17, 2002.

[46] K. D. Martin. Automatic transcription of simple polyphonic music: Robust front end processing. Technical Report 399, Massachusetts Institute of Technology, The Media Laboratory, December 1996.

[47] R. Martin. Spectral subtraction based on minimum statistics. In *Proc. Euro. Signal Processing Conf. (EUSIPCO)*, pages 1182–1185, 1994.

[48] P. Masri, A. Bateman, and N. Canagarajah. A review of timefrequency representations, with application to sound/music analysisresynthesis. *Organised Sound*, 2(3):193–205, 1997.

[49] M. Matia. A comparison of feed forward neural network architectures for piano music transcription. In *Proceedings of the International Computer Music Conference (ICMC)*, 1999.

[50] J. A. Moorer. On the transcription of musical sound by computer. *Computer Music Journal*, 1(4):32–38, 1977.

[51] J. Odell, D. Ollason, P. Woodland, S. Young, and J. Jansen. *The HTK Book for HTK V2.0*. Cambridge University Press, Cambridge, UK, 1995.

[52] A. Pertusa, A. Klapuri, and J.M. Iñesta. Note onset detection using semitone bands, 2005. Algorithm presented for the MIREX competition at the 6th International Conference on Music Information Retrieval (ISMIR). London, 2005.

[53] M. Piszczalski and B. Galler. Automatic transcription. *Computer Music Journal*, 1(4):24–31, 1977.

[54] M. Piszczalski, B. Galler, R. Bossemeyer, M. Hatamian, and F. Looft. Performed music: Analysis, synthesis, and display by computer. *Audio Engineering Society*, 29(1/2):38–46, February 1981.

[55] G. E. Poliner and D. P. W. Ellis. A classification approach to melody transcription. In *Proceedings 6th International Conference on Music Information Retrieval (ISMIR)*, pages 161–166, 11–15 September 2005.

[56] G. E. Poliner and D. P. W. Ellis. A discriminative model for polyphonic piano transcription. *EURASIP Journal on Advances in Signal Processing*, 2007(1):154–154, 2007.

[57] Meddis R. and O'Mard L. A unitary model of pitch perception. *The Journal of the Acoustical Society of America*, 102(3):1811–1820, September 1997.

[58] L. R. Rabiner. A tutorial on hidden markov models and selected aplications in speech recognition. In A. Waibel and K.-F. Lee, editors, *Readings in Speech Recognition*, pages 267–296. Kaufmann, San Mateo, CA, 1990.

[59] L. R. Rabiner, M. Cheng, A. Rosenberg, and C. McGonegal. A comparative performance study of several pitch detection algorithms. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 24(5):399–418, 1976.

[60] C. Raphael. Automatic transcription of piano music. In *Proceedings of the Third International Conference on Music Information Retrieval: ISMIR 2002*, pages 15–19, Paris, France, October 13-17, 2002.

[61] G. Reis and F. Vega. A novel approach to automatic music transcription using electronic synthesis and genetic algorithms. In *GECCO '07: Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation*, pages 2915–2922, New York, NY, USA, 2007. ACM.

[62] M. Ryynänen and A. Klapuri. Polyphonic music transcription using note event modeling. In *Proc. 2005 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 319–322, New Paltz, New York, USA, October 2005.

[63] Y. Sakuraba and H. G. Okuno. Comparing features for forming music streams in automatic music transcription. *IPSJ SIG Notes*, 2003(82):35–42, 2003.

[64] B. Schuller, F. Eyben, and G. Rigoll. Fast and robust meter and tempo recognition for the automatic discrimination of ballroom dance styles. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007*, pages I–217 – I–220, Honolulu, HI, 15 – 20 April 2007.

[65] M. Slaney. A critique of pure audition. *Computational auditory scene analysis*, pages 27–41, 1998.

[66] D. Stowell and M. D. Plumbley. Adaptive whitening for improved real-time audio onset detection. In *Proceedings of the International Computer Music Conference (ICMC07)*, pages 312–319, Copenhagen, Denmark, August 2007.

[67] H. Takeda, T. Otsuki, N. Saito, M. Nakai, H. Shimodaira, and S. Sagayama. Hidden markov model for automatic transcription of midi signals. In *Proceedings of IEEE Workshop on Multimedia Signal Processing*, pages 428–431, 2002.

[68] TiMidity++. Timidity++ software synthesizer. `http://timidity.sourceforge.net/`.

[69] T. Tolonen and M. Karjalainen. A computationally efficient multipitch analysis model. In *IEEE Transactions on Speech and Audio Processing*, volume 8(6), pages 708–716, Nov 2000.

[70] R. Turetsky and D. Ellis. Ground-truth transcriptions of real music from force-aligned midi syntheses. In *Proc. Int. Conf. on Music Info. Retrieval, ISMIR-03*, 2003.

[71] S. V. Vaseghi. *Advanced Digital Signal Processing and Noise Reduction*. John Wiley & Sons, 2006.

[72] E. Vincent and X. Rodet. Music transcription with ISA and HMM. In *Proceedings of Fifth International Conference in Independent Component Analysis and Blind Signal Separation, ICA*, pages 1197–1204, 2004.

[73] H. L. von Helmholtz. *Die Lehre von den Tonempfindungen: als physiologische Grundlage für die Theorie der Musik*. Vieweg, Braunschweig, 6th edition, 1913.

[74] H. L. von Helmholtz. *On the Sensations of Tone as a Physiological Basis for the Theory of Music*. Dover, New York, 1954. Original title: [73].

[75] J. Yin, A. Dhanik, D. Hsu, and Y. Wang. The creation of a music-driven digital violinist. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 476–479, New York, NY, USA, 2004. ACM.