Automatic transcription of recorded music

Peter Grosche, Björn Schuller, Meinard Müller, Gerhard Rigoll

Angaben zur Veröffentlichung / Publication details:

Grosche, Peter, Björn Schuller, Meinard Müller, and Gerhard Rigoll. 2012. "Automatic transcription of recorded music." Acta Acustica united with Acustica 98 (2): 199–215. https://doi.org/10.3813/aaa.918505.

ACTA ACUSTICA UNITED WITH ACUSTICA
Vol. 98 (2012) 199 – 215

Review Article
DOI 10.3813/AAA.918505

Automatic Transcription of Recorded Music

Peter Grosche[1)], Björn Schuller[2)], Meinard Müller[1)], Gerhard Rigoll[2)]
[1)] Saarland University and MPI Informatik, 66123 Saarbrücken, Germany. pgrosche@mpi-inf.mpg.de
[2)] Institute for Human-Machine Communication, Technische Universität München, D-80333 München, Germany. schuller@tum.de

Summary
The automatic transcription of music recordings with the objective to derive a score-like representation from a given audio representation is a fundamental and challenging task. In particular for polyphonic music recordings with overlapping sound sources, current transcription systems still have problems to accurately extract the parameters of individual notes specified by pitch, onset, and duration. In this article, we present a music transcription system that is carefully designed to cope with various facets of music. One main idea of our approach is to consistently employ a mid-level representation that is based on a musically meaningful pitch scale. To achieve the necessary spectral and temporal resolution, we use a multi-resolution Fourier transform enhanced by an instantaneous frequency estimation. Subsequently, having extracted pitch and note onset information from this representation, we employ Hidden Markov Models (HMM) for determining the note events in a context-sensitive fashion. As another contribution, we evaluate our transcription system on an extensive dataset containing audio recordings of various genre. Here, opposed to many previous approaches, we do not only rely on synthetic audio material, but evaluate our system on real audio recordings using MIDI-audio synchronization techniques to automatically generate reference annotations.
PACS no. 43.75.Xz, 43.75.zz

1. Introduction

Transcribing music recordings is the antipode of making music: *Listening to a piece of music and writing down its musical notation.* Various kinds of musical notation and ways of representing music in a symbolic form exist. For Western music, the *musical score*, or *sheet music*, is the traditional form of notating music on a piece of paper. This notation encodes a musical work using a universal and well known language. Being able to "read" this notation allows a musician to create a musical performance of this piece by following the given instructions. In particular, a score contains information on each note of the piece, such as onset time, pitch, and durations. However, the information given by the score is not a strict set of rules, but rather a recommendation or guideline admitting individual influences and articulations. Due to this ambiguous nature of this representation different recordings of the same piece of music may exhibit very different characteristics.

The goal of *automatic* music transcription (AMT) is to transform a music performance, given in the form of an audio recording, into a symbolic representation by the use of signal processing methods. In this context, the transcription process can be seen as *reverse-engineering the source code of a music signal* [1]. The waveform of an audio signal encodes changes of the air pressure which are caused by some vibrating object such as the vocal chords of a singer, the strings of a violin, or the standing wave vibration in the air column of a trumpet [2]. In this physical representation, no note-level parameters such as onset times, durations, or pitches are given explicitly. Playing even a single note on an instrument results in a complex mixture of sounds comprising components that correspond to the fundamental frequency (or pitch) as well as harmonics, i. e., integer multiple frequencies of the pitch [3]. For music, one generally has several notes played at the same time and even several instruments playing together. In this scenario, the determination of note parameters is a very challenging task.

Automatic music transcription systems typically proceed in three stages. In the first stage, pitch candidates are estimated for each time position. Here, one generally reverts to frequency domain approaches that analyze the harmonic structure of the spectrum to determine the fundamental frequency. In the second stage, note onset positions within the music signal are estimated. Here, a typical approach is to capture changes of the signal's energy or spectrum deriving a so-called *novelty curve*. The peaks of such a curve yield good indicators for note onset candidates. In the last stage, the information on note onsets and fundamental frequencies is fused to determine notes with the specific parameters onset time, pitch, and duration. Here, expert systems incorporating models of sound characteristics or musical properties are employed. These experts allow for solving otherwise ambiguous situations and obtaining meaningful transcription results.

Received 18 October 2010,
accepted 26 October 2011.

Note onset detection and polyphonic pitch estimation are well-studied problems and a variety of approaches have been proposed that work well under specific conditions. On the one side, pitch estimation algorithms exist that perform well in the case of monophonic music (only a single note is played at a time) and for instruments with a pronounced fundamental and only slight temporal and spectral variation, e. g., a piano. In the case of polyphonic mixtures, however, pitch estimation becomes a very difficult problem. Even a single note results in a complex sound with several harmonics, noise components, and vibrations. In this scenario, pitch estimation is a challenging task, especially when the number of simultaneously sounding notes is unknown and different notes played by different instruments are superimposed upon each other. On the other side, note onset detection is a manageable task when dealing with percussive instruments. In the case of classical music, however, one typically has to deal with instruments that feature weak onset information and smooth note transitions, which is often the case for string instruments. Here, the detection of relevant note onsets is problematic.

In this article, we describe a framework for automatic music transcription of audio recordings. Given an audio recording, our goal is to derive a symbolic notation. More precisely, we determine the properties of all notes contained in this recording on the basis of the waveform representation. We only focus on pitched instruments, leaving the transcription of percussive instruments aside, but percussive instruments may be present in the signal. Generally speaking, the system is carefully designed to cope with any piece of Western music. In our framework, we build on existing approaches to pitch estimation and note onset detection. However, we modify these state-of-the-art approaches leading to advantages under practical conditions. More precisely, we introduce a spectrogram computed using a combination of a multi-resolution Fourier transform and an estimate of the instantaneous frequency, which allows for obtaining a suitable time and frequency resolution. From this, we derive a semitone spectrogram, which accounts for the logarithmic properties of the equal-tempered scale commonly used in Western music. This representation is subsequently used as a basis for the transcription system. We then use a state-of-the-art pitch estimator [4] that we adapt to work on the semitone spectrogram. As a result, we obtain for each time position a number of pitch estimates as well as the respective saliences. Furthermore, we use a state-of-the-art onset detection method [5], which we adapt in such a way that it delivers a novelty curve for each of the 88 semitone bands. The peaks of this curves indicate likely positions for note onsets as well as offsets. As one main contribution of this article, we then sketch methods for combining the band-wise onset and pitch information. In particular, we introduce an approach based on Hidden Markov Models (HMM). Here, each note is modeled as one event with distinct temporal and spectral properties. The basic idea of this approach is to segment each of the semitone bands into regions where a note is active and regions where no note is active.

Our algorithm delivers the parameter representation in the form of a MIDI file which encodes the physical positions and properties of each note in the recording. As another major contribution, we intensively evaluate the performance of the transcription system on several large-scale datasets. To cover a wide range of musical styles and variabilities, we use 50 recordings of modern pop/rock music and 50 recordings of classical music. As ground truth transcriptions are not available for such extensive datasets, we first evaluate our algorithm on the basis of synthesized MIDI files. Then, we create a second dataset consisting of real audio recordings of the pieces. In this case, we obtain the reference transcription through force-aligning the MIDI files to the audio recordings, which allows for evaluating the performance of our transcription system under real-world conditions. In total, our evaluation datasets contain over 10 hours of audio with more than 700,000 notes.

The remainder of this article is organized as follows. In section 2, we give a detailed background on automatic music transcription while discussing relevant work. Then, we give a short overview of the proposed transcription system in section 3. In section 4, we describe the computation of semitone spectrograms on the basis of the multi-resolution Fourier transform, which is subsequently used as the basis for the pitch estimation algorithm described in section 5 as well as for computing the novelty curves explained in section 6. section 7 constitutes one main contribution of this article. Here, we introduce a method for deriving the note events by combining pitch and onset information. In particular, we introduce our approach to note event modeling using Hidden Markov Models. In section 8, we report on our experiments and discuss the transcription results of our system. Finally, we conclude in section 9 with a summary and an outlook on future work and possible enhancements to the transcription quality.

2. Background

In this section, we give a detailed background on automatic music transcription as well as the steps involved in deriving a symbolic representation from audio recordings.

Automatic transcription of music recordings is an active research topic, with early work starting in the mid 1970s [6]. Many different approaches have been proposed since then. However, the transcription of real-world music with an unknown number of coinciding notes, arbitrary instrumentation, various musical genre and tempi, or percussive accompaniment suffers from many unsolved problems. In restricted scenarios, however, the complexity of the transcription problem is reduced and state-of-the-art approaches deliver an acceptable performance. For example, restricting the audio to certain musical genre or even to only one specific class of instruments [7, 8] significantly simplifies the transcription problem. In particular piano music attracted a lot of attention due to its comparatively limited spectral and temporal variations [9, 10, 11, 12].

Other approaches are limited to obtain only a partial transcription of complex musical signals, e. g., the dominant melody or bass lines [13, 14]. Another common approach to reducing the difficulty of the transcription problem is to use synthesized MIDI files instead of real recorded performances. However, the synthetic character of this data simplifies the transcription task neglecting many variations of real sounds. Drums and percussion transcription systems were developed, too, (e. g., see [15]) but most of the harmonic music transcription systems exclude drums from input material, while others do allow the presence of drums but do not transcribe them.

A music transcription system typically includes three major stages: pitch estimation, note onset detection, and a fusion stage where actual notes are derived from this information. In the first stage, fundamental frequencies of musical sounds are determined. This is a very challenging task and many efforts have been made to solve this problem. In 1976, Rabiner *et al.* [16] compared various monophonic fundamental frequency estimation methods, examining frequency domain, time domain, and hybrid algorithms for analyzing speech signals. For music signals, exhibiting a wider bandwidth of fundamental frequencies, most of these approaches are not applicable. In 1977, Piszczalski and Galler [6] proposed the first monophonic music transcription system which is limited to certain types of instruments with strong fundamental frequencies. This frequency domain approach simply chooses the most pronounced spectral peak as the fundamental frequency. A common approach to fundamental frequency estimation in time domain is the autocorrelation method [17], because it is simple, fast, and reliable [18]. Here, the signal is compared with time shifted copies to reveal the underlying periodicities. Time domain zero-crossings [19] and cepstrum measurements [20] are further methods proposed for monophonic fundamental frequency estimation. Pitch estimation methods that work in frequency domain analyze spectral patterns to detect fundamental frequencies that best explain the harmonic structure [4, 21].

Monophonic pitch estimation is still challenging for signals with arbitrary temporal or spectral characteristics. For music signals, the assumption of monophony does not hold for the majority of musical pieces. Here, one generally has multiple notes that sound at the same time. In particular for Western music, the concept of chords plays an important role. Typically, a combination of three or more notes sound simultaneously and affect the harmony of a piece. In general, the number of simultaneously sounding notes is unknown. Even more challenging becomes the detection of fundamental frequencies in the case of complex mixtures of various different instruments exhibiting heterogeneous temporal and spectral characteristic. As a result, pitch estimation algorithms were proposed that are able to estimate the pitches present in polyphonic mixtures. In 1977, Moorer [22] proposed the first polyphonic music transcription system. This system focuses on duets and is limited to two monophonic instruments playing at the same time. Furthermore, no overlap between notes or harmonics is allowed. Recent methods with the focus on general music signals try to decompose the signal into smaller elements to separate simultaneous notes. Here, sinusoidal component tracking and grouping of sound sources according to specific attributes [14] are used. Furthermore, a sub-band decomposition allows for analyzing periodicities in certain frequency regions [23, 24, 25]. Another typical approach to polyphonic pitch estimation is to process the signal iteratively using a monophonic pitch estimation algorithm. Here, the spectrum of a detected note, or predominant fundamental frequency, is first subtracted from the signal and the pitch estimation is iteratively repeated on the residual, see [26, 27, 4, 28]. Classification based pitch estimation has been proposed using different classifiers. Support Vector Machines are used in [12, 29] for frame level pitch classification. In [30, 31, 32, 12], hidden Markov models are used to account for the temporal properties of a note and to classify sequences of features representing specific states of a note.

In the second stage, physical starting times of notes and other musical events in a music recording are determined. The general idea of this *onset detection* is to capture sudden changes in the music signal, which are typically caused by newly occurring events. As a result, one obtains a so-called *novelty curve*, the peaks of this curve yield good indicators for locating note onsets. Many different methods for computing novelty curves have been proposed, see [33] for an overview. For example, playing a note on a percussive instrument typically results in a sudden increase of the signal's energy, which allows for determining note onsets [33].

Much more challenging, however, is the detection of onsets in the case of non-percussive music, where one often has to deal with soft onsets or blurred note transitions. This is often the case for vocal music or classical music dominated by string instruments. As a consequence, more refined methods have to be used for computing the novelty curves, e. g., by analyzing the signal's spectral content [33, 34], pitch [34, 35], harmony [36, 37], or phase [33, 38]. Furthermore, in complex polyphonic mixtures of music, simultaneously occurring events may result in masking effects, which makes it hard to detect individual onsets. As a consequence, detection functions were proposed that analyze the signal in a band-wise fashion to extract transients occurring in certain frequency regions of the signal [39, 40]. A widely used approach to onset detection in the frequency domain is the *spectral flux* [33], where changes of pitch and timbre are detected by analyzing the signal's short-time spectra. Combining approaches to spectral change detection and phase analysis results leads to onset detection in the complex domain [5], which shows good onset detection quality for a wide range of signals.

In the third stage, information from the pitch estimation and onset detection stages is combined to create note events which best explain the signal's content. Blackboard systems utilizing *expert* systems with certain knowledge are widely used for combining the features in a prediction

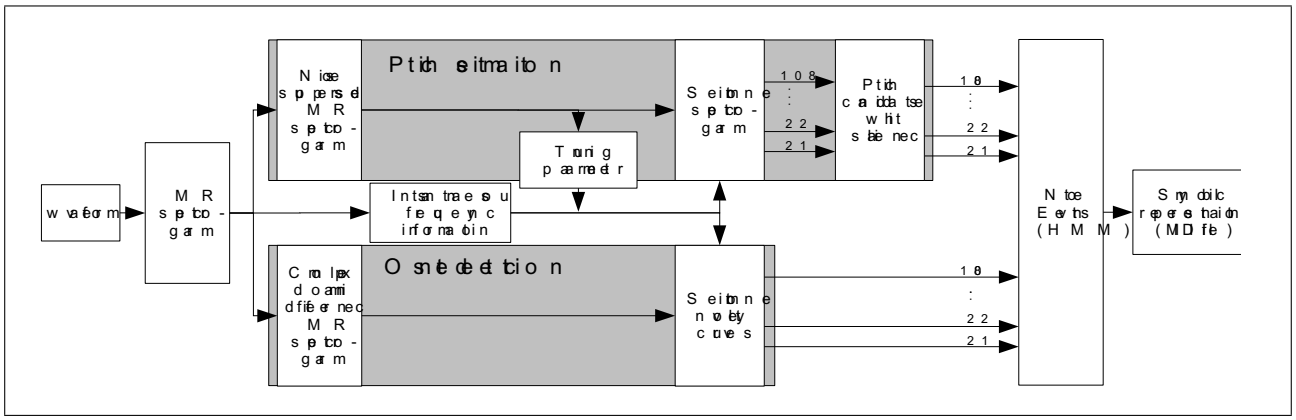

Figure 1. Flowchart of the proposed music transcription system.

driven way [41, 42, 43]. These experts carry knowledge of physical and acoustical properties of tones or auditory models [44]. In [45], the first system making use of musically high-level information, such as key, beat, and metric structure extracted from the low level musical signal is presented. Information on chords, musical intervals, key, and rhythm can be used for enhancing the transcription result and resolve otherwise ambiguous situations [46]. A speech recognition motivated *musicological* model defining note transition probabilities is proposed in [31] using hidden Markov models. A prediction-driven analysis by synthesis system employing genetic algorithms is proposed in [47]. Probabilistic matrix factorization [48] or harmonic-temporal structured clustering [49] together with complex probabilistic models of notes and higher-level musical knowledge allow for further improving the transcription results. In this context, Markov Chain Monte Carlo methods [50], or Poisson point processes [51] were proposed. See [52] for a general overview of statistical methods in the field of music processing.

3. Overview of the Transcription System

In this section, we give an overview of the processing steps of our transcription system. As mentioned in the introduction, our goal is to derive a symbolic representation of all notes contained in a recording, given as an audio waveform. Here, we only regard pitched instruments leaving percussive instruments aside. Furthermore, our aim is not to create a traditional Western music score, but a symbolic representation where all physical note parameters are given explicitly.

Figure 1 shows the flowchart of the system. Starting with the waveform of a music recording, we first compute a *multi-resolution (MR) spectrogram* employing a combination of three Fourier transforms. This spectrogram accounts for the non-stationary character of music signals and the need for a high frequency resolution at the same time. In the following, this representation is employed for the pitch estimation and onset detection.

For the pitch estimation, we only consider notes played by pitched instruments. Therefore, we suppress broadband, noise-like components typically caused by percussive instruments and enhance harmonic parts of the pitched instruments. To this end, we apply a RASTA preprocessing on the magnitude MR-spectrogram and obtain a *noise suppressed MR spectrogram*. In the next step, the MR spectrogram is mapped to a musically meaningful frequency scale. Here, the Fourier coefficients are grouped to form a new spectrogram, whose frequency axis reflects the logarithmically spaced pitches (each corresponding to a semitone) of the equal-tempered scale. Because of this property, the resulting representation is referred to a *semitone spectrogram* and each of its bands corresponds to one of the 88 musical pitches A0 to C8. In this step, a *tuning parameter* is estimated and used to compensate for a possible detuning of the recording. Furthermore, an estimate of the *instantaneous frequency (IF)* is used to enhance the precision of the assignment of the frequency components to the respective semitones. Then, we apply a pitch estimator on this semitone representation. More precisely, we adapt the state-of-the-art pitch estimator proposed in [4]. The general idea of this method is to iteratively determine predominant pitch candidates by summing up amplitudes of harmonics. As a result, we obtain for each time position a number of *pitch candidates* together with a respective *salience* value.

The goal of onset detection is to detect changes of the signal evoked by note onsets. We compute a *complex domain difference MR spectrogram*, which encodes changes of the signal's phase and magnitude for each of the Fourier coefficients [33]. Again, we map this representation to a semitone frequency scale which results in 88 *semitone novelty curves*, one for each of the pitches. The peaks of these curves indicate likely positions for note onsets.

The approaches to pitch estimation and onset detection used here are state-of-the-art yet basic variants. In our framework, we focus on the post-processing step combining the (possibly noisy) pitch and onset information to obtain distinct note events. Such note events are defined by a pitch, onset, and duration. For determining the note events that best explain the band-wise pitch saliences and novelty curves, we use Hidden Markov Models that segment each semitone band into regions where a note is active and regions where no note is active. In particular, we model

each note as a temporal event ranging from the onset to the release. In the final step, the note events are converted into a symbolic representation and provided in the form of a MIDI file. This MIDI file can be regarded as an annotation of the audio recording explicitly revealing all note parameters of the specific performance.

4. Multi-Resolution Spectrogram

In this section, we introduce the multi-resolution spectrogram, which constitutes the first step in the transcription system and provides an efficient time/frequency representation that fulfills certain requirements on time and frequency resolutions in the different frequency regions.

The most common musical temperament in Western music is the twelve-tone equal temperament. On this scale, an octave represents a doubling of frequency and is split into 12 (logarithmically) equally spaced semitones. The frequency of each semitone $p \in [21 : 108] := \{21, 22, \ldots, 108\}$ is defined by

$$f_p = 440 \cdot 2^{\frac{p-69}{12}} \text{ Hz} \tag{1}$$

where p is also referred to as the MIDI note number. explicitly identifies the pitch of a note. As a consequence, the relation of adjacent semitones is always $f_p / f_{p-1} = \sqrt[12]{2} \approx 1.059$ or roughly 6%. This logarithmic frequency scale is problematic for processing music signals, as it results in very small absolute frequency differences between notes on the lower end of the scale. For example, the difference of the center frequencies of pitches $p = 21$ (A0) and $p = 22$ ($B^{\flat}0$) is only 1.64 Hz. Using a short-time Fourier transform requires a long analysis window to obtain such a fine frequency resolution [53]. However, music is highly non-stationary, where quasi-stationarity can only be assumed in very short durations of time. As a consequence, an adequate time/frequency representation for analyzing music signals needs to provide both, necessary time and frequency resolution at the same time [54]. Constant-Q transforms [55], wavelet transforms [56], or multirate filter banks [3] are used in the musical context to account for this issue. Here, a long analysis window is used for low pitches to provide the desired frequency resolution and a short analysis window is used for high pitches to retain a high time resolution.

Similar to [57, 53], we construct a multi-resolution spectrogram which consists of three spectrograms of different time and frequency resolutions. Let x be a discrete music signal. Then, we fix a window of N samples, use a step size of M samples and apply a short-time Fourier transform to obtain a spectrogram $X = (X(t, k))_{t,k}$ with $t \in [1 : T]$ and $k \in [1 : K]$. Here, T determines the number of frames, $K = N/2$ denotes the number of Fourier coefficients, and $X(t, k)$ refers to the k^{th} Fourier coefficient for time frame t. In our system, we use signals with a sampling rate $Fs = 44100$ Hz and compute Fourier transforms for three Hann windows of different sizes, $N_1 = 4096$, $N_2 = 8192$, and $N_3 = 16384$ with the step sizes $M_1 = N_1/2$, $M_2 = N_2/4$, and $M_3 = N_3/8$. This results in three spectrograms X_1, X_2, and X_3 with different time and frequency resolutions but corresponding frames t. Using these settings, each time parameter t reflects $r = 46$ ms of the audio recording. Furthermore, zero padding at the boundaries of the signal ensures that we obtain the same number of frames T for each of the spectrograms. We then create a multi-resolution spectrogram $\mathcal{X} = (\mathcal{X}(t, q))_{t,q}$ choosing the spectrogram which constitutes the best compromise for the desired time/frequency resolution for each frequency range and copying the respective Fourier coefficients to $\mathcal{X}$. X_3 provides the best frequency resolution and is used for the low frequency region from 0 to 110 Hz (A3). Similarly, X_2 is used from 110 to 220 Hz (A4) and the coefficients of X_1 are used for frequencies above 220 Hz. The multi-resolution spectrogram provides adaptive time frequencies resolutions in different frequency regions and constitutes an effective tool for representing music signals.

5. Pitch Estimation

In this section, we explain the polyphonic pitch estimation algorithm used in our transcription system. The multi-resolution spectrogram as described in section 4 is first processed to obtain a noise suppressed multi-resolution spectrogram. Then, a semitone spectrogram is derived which reflects the logarithmic characteristics of pitch. Finally, a multiple pitch estimation algorithm is applied to this representation which estimates the pitches which are active at a certain time position.

5.1. Noise Suppression

As mentioned in section 3, our transcription system is restricted to transcribing pitched instruments, leaving percussive instruments unconsidered. To reduce the effect of noise-like signal components, we separate the signal into harmonic (voiced) parts of pitched instruments and non-harmonic (unvoiced) parts of percussive instruments.

Noise suppression has been widely studied in the speech domain. In this context, noise typically has a stationary character and can be estimated by statistics of the signals over a period of time [58]. When dealing with music, however, the term "noise" usually refers to parts of the signal that are unvoiced. Typically, these components result from percussive instruments, but even pitched instruments may produce noise components. For example, for instruments with a strong attack phase, as the piano, the onset goes along with a broadband noise event. While these parts are a strong indicator for onset positions, the notions of a pitch, or fundamental frequency only makes sense for harmonic signals. Therefore, the noisy parts are removed in the preprocessing step.

Here, we employ relative spectral (RASTA) processing which has been proposed for the suppression of noise in speech signals [59] and has been successfully applied to music [60]. The general idea is that noise-like components exhibit a high spectral bandwidth, while harmonic components result in a spectral peak [61]. For each time position,

the noise components are estimated by first converting the MR spectrogram $\mathcal{X}$ in a spectrogram with logarithmic intensities $\mathcal{Y}(t,q) = \ln(1+1/J \cdot |\mathcal{X}(t,q)|)$. Here, J is a signal dependent value which takes care of scaling noise components to the quasi linear range of the logarithm, while spectral peaks go through a logarithmic transform. Similar as in [60], we define J as the average of the magnitude in the frequency range of the pitches $p = [21 : 108]$. Then, the noise in $\mathcal{Y}(t,q)$ is estimated by computing a moving average in octave wide windows. This musically meaningful window is used instead of the of the ERB critical-bands proposed in [60] to better reflect the frequency characteristics of music signals by considering an equal amount of musical information in each window. The estimated noise spectrum is then subtracted from $\mathcal{Y}$, resulting negative values are set to zero, and $\mathcal{Y}$ is converted back into a linear intensities, see [60] for further details. In the following, the resulting spectrogram is referred to as noise suppressed multi-resolution magnitude spectrogram $\overline{\mathcal{X}}$.

5.2. Semitone Spectrogram

The noise suppressed multi-resolution spectrogram $\overline{\mathcal{X}}$ is an effective representation of the harmonic parts of the signal. In the next step, the frequency scale of this spectrogram is adapted to the logarithmically spaced equal tempered scale. To this end, we group the Fourier coefficients into semitone sub-bands, where each sub-band covers the frequency range of a pitch. The frequency range assigned to a pitch $p \in [21 : 108]$ with center frequency f_p is defined by the upper frequency bound $f_{p+0.5}$ and the lower bound $f_{p-0.5}$, see equation (1). We then assign all Fourier coefficients q of $\overline{\mathcal{X}}(t,q)$ with the frequency f_q in the range $f_q > f_{p-0.5}$ and $f_q \leq f_{p+0.5}$ to this pitch. For determining the frequency f_q we do not simply use the center frequency of the coefficient, but compute an estimate of the instantaneous frequency.

The instantaneous frequency (IF) refers to the actual effective frequency of a sinusoidal component in the spectrogram. The advantage of this methods is that it does not depend on the frequency grid defined by the parameters of the Fourier transform used for computing the spectrogram. While each Fourier coefficient describes an entire range of the spectrum the instantaneous frequency precisely determines the most prominent frequency component within this range. As a result, determining the IF allows for relocating the time/frequency information of the spectrogram, a task commonly referred to a time/frequency reassignment, see [62, 61, 57]. A variety of methods exists for estimating the IF from Fourier spectrograms, see [63] for an overview. We use the phase vocoder method [64] which relies on the phase difference between adjacent frames of the complex multi-resolution spectrogram $\mathcal{X}(t,q)$ to compute the instantaneous frequency $f_q(t)$.

Let $f_q(t)$ now denote the frequency of the Fourier coefficient q at time position t. Before mapping this coefficient to the pitch representation, we compensate for detuning effects. For example in classical music, solo instruments or even complete orchestras are often not strictly tuned to the concert pitch A4 with 440 Hz but to a sightly higher frequency. Computing a pitch representation of such a detuned signal would result in a very noisy representation. To determine such a detuning, we compute a single Fourier transform of the entire recording and specify the prominent frequency f_{prom} of the music recording, i. e., the frequency of the coefficient that exhibits the highest magnitude in range A0 − C8. Then, we assign the f_{prom} to the center frequency of the semitone p' which is the closest in terms of frequency. Finally, the global tuning factor T of the recording is defined as $T = f_{p'}/f_{prom}$ and used for refining all instantaneous frequencies $f_q(t)' = T \cdot f_q(t)$.

The tuned instantaneous frequency $f_q(t)'$ is then used for assigning the coefficient q of the MR spectrogram to the corresponding pitch of the equal tempered scale. For each MIDI pitch $p = [21 : 108]$ (A0 − C8) of the piano keyboard the sub-band magnitude is computed by summing all coefficients q of $\overline{\mathcal{X}}(t,q)$ in the range $f_q(t)' > f_{p-0.5}$ and $f_q(t)' \leq f_{p+0.5}$ of the pitch p. More precisely, the coefficients are weighted by a Gaussian window which is centered at f_p and gives higher weight to components which are closer to f_p. This results in a spectrogram representation $\mathcal{S}(t,p)$ exhibiting a frequency axis which follows the logarithmic semitone scale. Consequently, $\mathcal{S}$ is referred to as semitone spectrogram. Figure 2 shows a semitone spectrogram $\mathcal{S}(t,p)$ (Figure 2b) as well as a spectrogram $\mathcal{X}(t,q)$ with linear frequency axis (Figure 2a) for an A4 played by an piano. 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.

5.3. Pitch Estimation Algorithm

The noise suppressed magnitude semitone spectrogram is now used for detecting pitch candidates for each time position. candidate can be seen as a note with a certain pitch, in the signal at a specific time position. The method used in our framework is based on a conceptually simple and computationally efficient state-of-the-art approach [4]. However, we adapt this method to work on the semitone spectrogram in contrast to a linear spectrogram as proposed in [4]. This multi-pitch algorithm iteratively processes each time position to determine co-occurring notes with differing pitches.

Roughly speaking, the pitch estimator analyzes the harmonic structure of a composite signal to determine the pitch. Recall that a note played on a pitched instrument results in a mixture of many harmonics or partials. Each partial has a frequency that is an integer multiple of the fundamental frequency (F0), which defines the pitch of the note. Such a harmonic spectrum is shown in Figure 2a for an A4 (F0 = 440 Hz). Note that coefficients at multiples $m \in [1 : M]$ of 440 Hz, i. e., 880, 1320, 1760, 2200 Hz, for $m = 2, 3, 4, 5$ exhibit high intensity values, too. Because of the logarithmic properties of the semitone spectrum, the harmonics of a pitch p occur in the semitone bands $p+d(m)$ with $d(m) = \lfloor 12 \log_2(m) \rceil$, where $[\ \cdot\ ]$ denotes rounding to the closest integer, see Figure 2b.

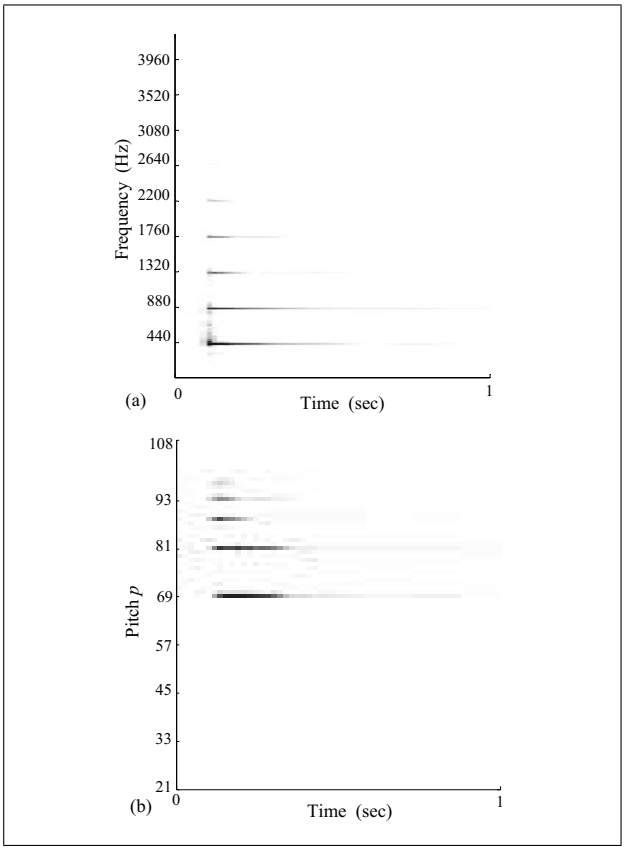

Figure 2. Comparison of a linear and semitone spectrogram of an A4 played by a piano. Dark colors indicate high intensities. (a) Linear spectrogram $\mathcal{X}$, (b) Semitone spectrogram $\mathcal{S}$.

Exploiting this harmonic relation of the overtones, a pitch salience spectrogram $\mathcal{S}^*(t, p)$ is computed, which defines a salience value for each semitone p. The salience expresses how dominant p is within the mixture. For each time position t and each p, a weighted sum of the amplitudes of all harmonics m in the semitone spectrogram $\mathcal{S}$ is computed:

$$\mathcal{S}^*(t, p) = \sum_{m=1}^{M} g_e(t, p, m) \cdot \mathcal{S}\big(t, p + d(m)\big). \tag{2}$$

Here, $g_e(t, p, m)$ is a weighting function that defines the weight of component m contributing to the salience of p.

Figure 3a shows a single time frame of a semitone spectrogram of an A4 played on a piano. Note the high intensities of the fundamental as well as the harmonics. In general, the fundamental does not necessarily correspond to the highest peak. Actually, for many instruments the fundamental may be very weak. In the pitch salience spectrum, however, the fundamental is supported by the harmonics showing a distinct peak, see Figure 3b.

From the pitch salience spectrogram $\mathcal{S}^*(t, p)$, the most probable, or *predominant*, pitch candidate $\hat{p}_t$ of a time frame t is determined as the one with maximum salience:

$$\hat{p}_t = \underset{p}{\operatorname{argmax}} \big(\mathcal{S}^*(t, p)\big). \tag{3}$$

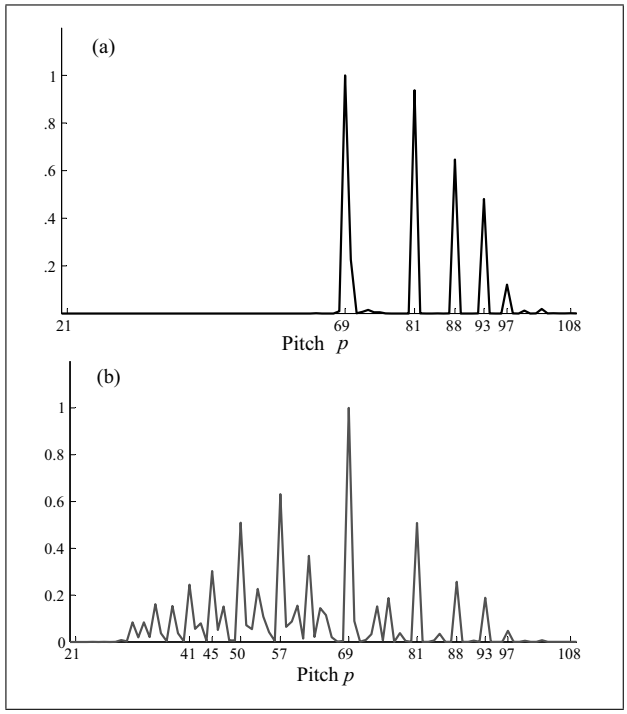

Figure 3. A single time frame of a semitone spectrogram $\mathcal{S}$ and the corresponding pitch salience spectrum $\mathcal{S}^*$ for a piano A4 ($p = 69$). (a) Semitone spectrum $\mathcal{S}$, (b) Salience spectrum $\mathcal{S}^*$.

We then obtain a pitch candidate spectrogram $\mathcal{C}$ that represents the determined pitch candidates along with the respective salience:

$$\mathcal{C}(t, p) = \begin{cases} \mathcal{S}^*(t, p) & \text{for} \quad p = \hat{p}_t, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

$\mathcal{C}$ now exhibits one entry for each time position, which would be the final result in a monophonic or predominant pitch estimation task. In a polyphonic context an iterative estimation and cancellation technique is applied. Here, the spectral shape $\mathcal{S}_D(p)$ of a detected pitch candidate is estimated and subsequently subtracted from the spectrogram $\mathcal{S}(t, p)$:

$$\mathcal{S}_R(t, p) = \max\big(0, \mathcal{S}(t, p) - \mathcal{S}_D(p)\big). \tag{5}$$

The pitch estimation process in equation (2) is then iteratively repeated using the residual $\mathcal{S}_R$ instead of $\mathcal{S}$. Finally, the algorithm is terminated when a certain number of pitches is detected or the energy of the residual $\mathcal{S}_R$ falls under a suitable threshold. The resulting pitch candidates spectrogram $\mathcal{C}(t, p)$ exhibits entries for all pitch candidates with the salience value, see Figure 4.

The algorithm in equation (2) relies on a weighting function g_e which defines the contribution of a partial to a pitch. As it turns out, the pitch estimation performance crucially depends on g_e, which needs to be optimized to account for the properties of the underlying signal. Similar to [4], we employ a weighting function that is a multiplication of three components $g_e(t, p, m) = g_{e1}(p) \cdot g_{e2}(m) \cdot g_{e3}(t, p + d(m))$. The first component g_{e1}

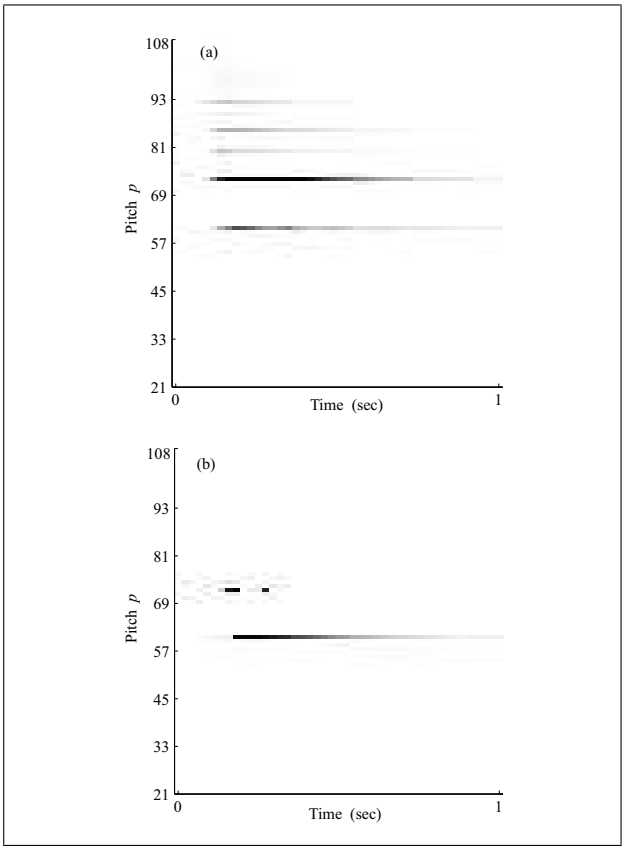

Figure 4. Semitone spectrogram S and the corresponding pitch candidate spectrogram C for a piano C4 ($p = 60$). (a) Semitone spectrogram S, (b) Pitch candidate spectrogram C.

only depends on the semitone index p and ensures that notes in each frequency range are detected with equal probability. This constitutes a form of spectral whitening [65, 4]. In our implementation, g_{e1} returns the same weight for the twelve pitches within the same octave. The second component, g_{e2} describes the *importance* of each component in the harmonic series. and can be regarded to model of the spectral shape of an average music instrument. As for the third component g_{e3}, note that the pitch estimation algorithm processes each time position independently. A note, however, is always a temporal event with certain time-dependent parameters. In particular, only stationary signal parts exhibit a stable harmonic structure whereas transients exhibit an irregular harmonic structure. Therefore, g_{e3} gives higher weight to stationary components of $S(t, p + d(m))$. At this point, we refer to section 6 for the definition of $g_{e3}(t, p + d(m))$.

The components of the weighting function are optimized independently through a cyclically brute-force estimation process to minimize the pitch estimation error rates, see [4] for further details. Figure 5 shows the resulting optimized weighting functions g_{e1} (top) and g_{e2} (middle). Most noticeable, g_{e1} gives a higher weight to notes of the lower two octaves and a lower weight to notes of the higher two octaves. g_{e2} roughly linearly declines with m ($m = 1$ corresponds to the fundamental), however, all

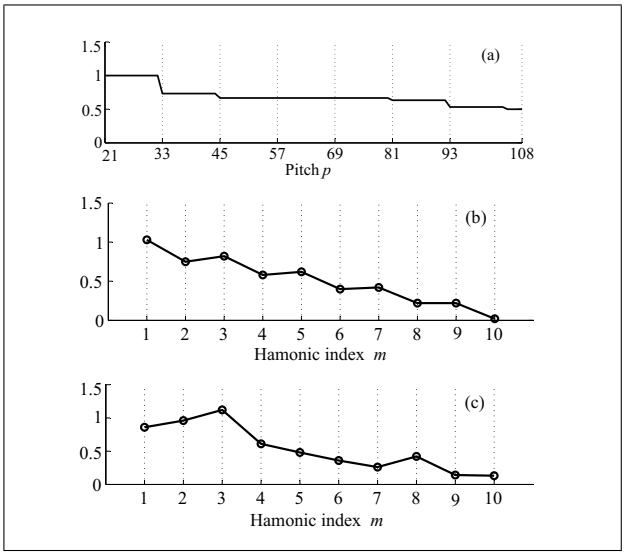

Figure 5. Optimized weighting functions $g_{e1}(p)$ (a) and $g_{e2}(m)$ (b) and the prototypical spectrum $g_s(m)$ (c).

odd numbered harmonics get a higher weight, than even numbered.

The subtraction of a pitch candidate from the spectrogram in equation (5) has to meet two contradicting requirements. On the one hand, all components belonging to the predominant pitch have to be removed. On the other hand, however, the spectra of the other notes must remain unchanged. Here, we follow [4] and exploit some general assumptions on the spectral shape of a note. We define $S_D(p)$ as a prototypical spectrum of an average instrument with specific spectral properties that are common between all notes, played by arbitrary instruments. To this end, we fix the magnitude $S(t, \hat{p}_t)$ of $\hat{p}_t$ as the anchor point and predict the magnitudes of the harmonics $\hat{p}_t + d(m)$ by employing a general harmonic shape g_s that is common between all notes,

$$S_D(p) = \begin{cases} g_s(m) \cdot S(t, \hat{p}_t) & \text{for} \quad p = \hat{p}_t + d(m), \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

The prototypical spectrum g_s is learned from training data consisting of notes played by different instruments, see Figure 5 (bottom) for the result.

The multi-pitch estimation algorithm presented here closely follows [4]. However, we apply the multi-pitch estimation algorithm to a pitch representation instead of to a spectrogram with a linear frequency scale. Here, we exploit the knowledge that musical notes occur on frequencies corresponding to pitches. Consequently, the computational complexity of the algorithm is drastically reduced. Furthermore, the reduced frequency resolution avoids the problem of inharmonicity, where harmonics are slightly detuned. On the other side, the overall number of harmonics that can be regarded for supporting the pitch estimation is limited to $M = 10$, because higher harmonics can no longer be captured by the pitch scale. However, as higher order harmonics only have a minor contribution, this effect is negligible.

6. Onset Detection

A musical note occurring in a recording is not only defined by the pitch, but also by the physical note onset time and the duration. In this section, we introduce the approach to onset detection which is employed in our transcription system. Recall from section 2 that a note onset typically goes along with a sudden change of the signal's energy, phase, or spectral content. The general idea is to capture such changes of the signal to derive a *novelty curve*. The peaks of this curve indicate likely candidates for note onset positions.

In our framework, we employ the complex domain onset detection method [5] which combines both, the detection of magnitude and phase changes. This technique has proven to yield a good detection performance for a wide variety of signals, see [33]. Intuitively, jointly analyzing the signal's magnitude and phase allows for capturing percussive note onsets (e. g. by drums) as well as smooth note transitions (e. g. by string instruments). Typically, percussive note onsets go along with a sudden change of the magnitude spectrum. Smooth note transitions, however, are hardly reflected in the magnitude spectrum. In this case, however, one can still observe an unexpected behavior of the phase resulting from changes of the pitch.

We again rely on the complex multi-resolution spectrogram $\mathcal{X}(t, q)$, see section 4. From this, we derive a complex domain difference spectrogram $\mathcal{X}(t, q)'$, by computing the Euclidean distance between the complex spectrum $\mathcal{X}(t, q)$ at time position t and a predicted spectrum $\hat{\mathcal{X}}(t, q)$

$$\mathcal{X}(t, q)' = \left| \mathcal{X}(t, q) - \hat{\mathcal{X}}(t, q) \right|^2. \tag{7}$$

The prediction $\hat{\mathcal{X}}(t, q)$ is created under the assumptions that the magnitude between adjacent frames is constant (no changes of the signal's spectral content) and the phase $\varphi(t, q)$ exhibits a constant slope (no pitch changes),

$$\hat{\mathcal{X}}(t, q) = \left| \mathcal{X}(t-1, q) \right| \cdot \mathrm{e}^{2\varphi(t-1,q) - \varphi(t-2,q)}. \tag{8}$$

Consequently, the complex domain difference spectrogram $\mathcal{X}(t, q)'$ quantifies the non-stationarity of the signal for each frame t and coefficient q, see [33] for details.

Small distances between the predicted and actual spectrum indicate small temporal changes, large distances indicate large changes of the signal properties caused by note onsets. However, $\mathcal{X}(t, q)'$ does not allow for discriminating note onsets and note offsets. To circumvent these limitations, we divide $\mathcal{X}(t, q)'$ into two components: the component $\mathcal{X}(t, q)^+$ that only represents coefficients of $\mathcal{X}(t, q)'$ where the magnitude is increasing, and the component $\mathcal{X}(t, q)^-$, where the magnitude is decreasing,

$$\mathcal{X}(t, q)^+ = \begin{cases} \mathcal{X}(t, q)' & \text{for } |\mathcal{X}(t, q)| > |\mathcal{X}(t-1, q)|, \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

$$\mathcal{X}(t, q)^- = \begin{cases} \mathcal{X}(t, q)' & \text{for } |\mathcal{X}(t, q)| \leq |\mathcal{X}(t-1, q)|, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

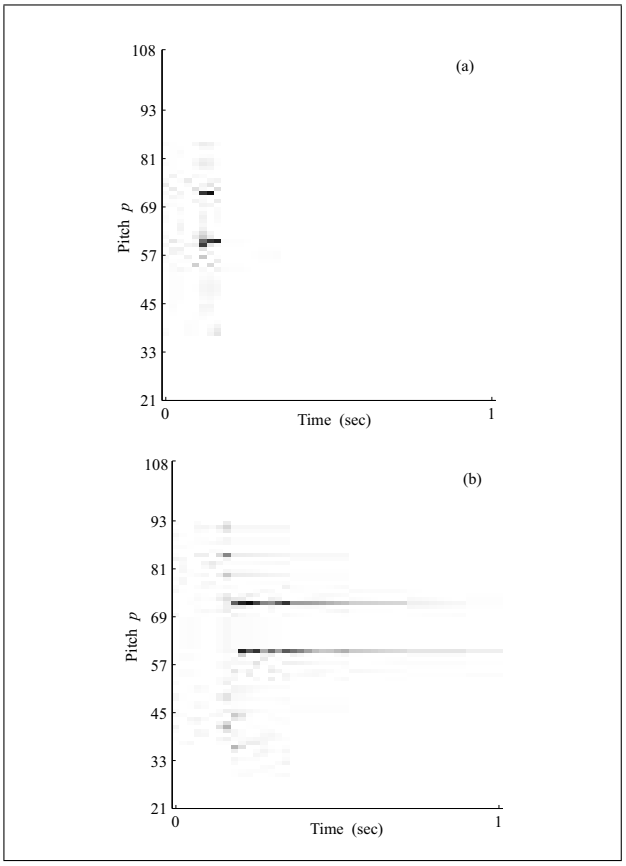

Figure 6. Semitone novelty curves for a piano C4 ($p = 60$). (a) $\Delta^+(t, p)$ exhibits high values (dark color) at the onset position, (b) $\Delta^-(t, p)$ exhibits high values in the release part of the note.

Usually, one then computes the sum of all values of $\mathcal{X}(t, q)'$, (or $\mathcal{X}(t, q)^+$ and $\mathcal{X}(t, q)^-$) over all q to obtain a global complex domain novelty curve. This curve exhibits peaks at likely note onset/offset positions. However, when computing just one global novelty curve, one loses the link between the onset and the pitch information. In our system, we rely on determining the onset and offset positions of specific notes with a specific pitch. Consequently, we do not compute a global novelty curve, but retain the spectral information of $\mathcal{X}(t, q)'$. Again, we employ a semitone based representation and compute a novelty curve for each of the semitones. Here, we are closely following the computation of the semitone spectrogram explained in section 5.2. Using the same instantaneous frequency and tuning factor information, we sum all entries q of $\mathcal{X}(t, q)'$, $\mathcal{X}(t, q)^+$, and $\mathcal{X}(t, q)^-$ which correspond to the same pitch p, see section 5.2. This results in 88 novelty curves $\Delta(t, p)$ ($\Delta^+(t, p)$ and $\Delta^-(t, p)$), each representing the temporal properties of a specific semitone band [66] and allows for determining onset and offset times in a polyphonic context, see Figure 6.

Furthermore, we exploit that Δ quantifies the non-stationarity of the signal for the weighting factor $g_{e3}(t, p + d(m))$ of the pitch estimation algorithm, see section 5.3. In order to give higher weight to stationarity signal components, we define $g_{e3}(t, p + d(m)) = 1/|\Delta(t, p + d(m))|$.

7. Note Event Modeling

The pitch candidate spectrogram $C(t, p)$ indicates the active pitches for each time position. Similarly, the novelty curves $\Delta^+(t, p)$ (and $\Delta^-(t, p)$) encode likely note onsets (and offsets) for each of the 88 semitones. In this section, we describe a method for combining this information to define the explicit set of notes with the parameters onset, duration, and pitch occurring in a recording.

In general, the number of simultaneously sounding notes in a complex mixture is unknown and needs to be estimated. Here, the iterative pitch estimation algorithm described in section 5.3 needs to be terminated when a newly detected pitch candidate does not contribute to the overall result in a significant way, see [4] for details. However, such a polyphony detection step is error prone, frequently introducing additional notes (false positives) or omitting notes (false negatives). Likewise, using a peak picking strategy on the basis of fixed or adaptive thresholding on the novelty curves introduced in section 6, one can determine note onset positions [33]. The novelty curves, however, tend to be rather noisy and exhibit many spurious peaks. As a result, the selection of relevant peaks corresponding to true note onset positions remains difficult, especially in the case of non-percussive music with soft note onsets.

The determination of note events from these two independent representations is further complicated in the case when the information is not consistent or even contradicting. For example, the pitch estimator may exhibit a pitch candidate with a high salience for a certain time position. The novelty curve for this semitone, however, may not indicate any likely note onset position in the proximity. On the other hand, there may be a high peak in the novelty curve indicating a likely note onset, but no pitch candidate that is related to this onset. In these cases, the determination of a note is problematic. For this reason, we avoid the error prone step of determining note parameters independently and propose a method which takes the temporal and spectral structure of a note into account. The idea is to model the note as a continuous event from the onset to the offset with distinct time-dependent properties. These note events are described using Hidden Markov Models (HMM).

An HMM is a probabilistic model and a powerful tool for modeling time-varying statistical processes, see [67] for a detailed introduction. First proposed for speech recognition tasks, HMMs are frequently used for music analysis tasks in general [68, 69, 70, 71, 72, 73, 74] and for music transcription in particular [75, 76, 77, 30, 31]. In the transcription context, the idea is to describe each note event as a sequence of states. This state sequence may, in an abstract from, reflect the onset, attack, decay, sustain, offset, or release parts of a note. The state sequence, however, is hidden and can only be observed through another stochastic process that produces a sequence of observation (feature) vectors. By combining spectral and temporal information, instead of determining pitch and onset/offset independently, a robust recognition system is obtained.

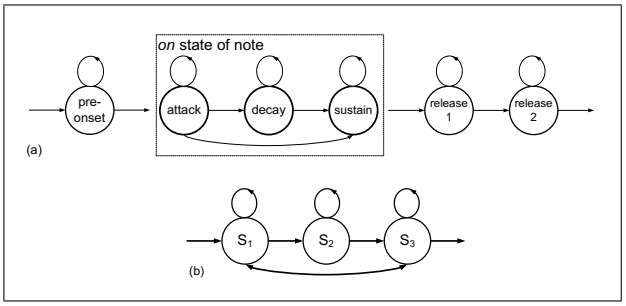

Figure 7. Schematic illustration of the topology of the two models used for note on/off segmentation of each semitone band. (a) Note *on* model, (b) Note *off* mode.

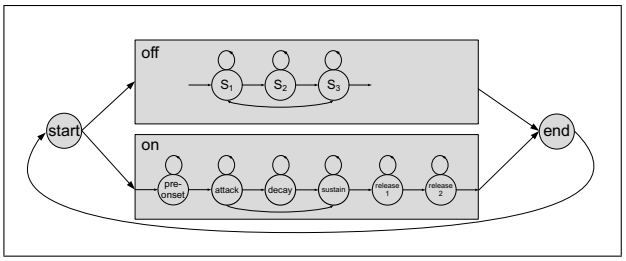

Figure 8. Schematic diagram of the recognition network used for note on/off segmentation of the semitone bands.

We propose a segmentation approach based on HMMs, which segments each semitone band into regions where a note is active and regions where no note is active. Each of the 88 semitones is then represented as a sequence of alternating note *on* and note *off* regions. In our case, we model the properties of notes using the pitch candidate spectrogram C as well as the semitone novelty curves Δ, Δ^+, and Δ^-. Both representations supply information on the semitone level. The feature space for all semitones is the same, which allows for sharing the models for all semitones. Furthermore, all semitone bands can be processed independently.

We use two models for segmenting the semitone bands. An *on* model that captures properties of sounding notes and an *off* model which captures regions where no note is active, as depicted in Figure 7. The *on* model exhibits three states with a left-right topology. This reflects the assumption that a note consists of an attack, decay, and sustain part. Furthermore, we add a *pre-onset* state as well as two *release* states that take signal properties before and after the *on* state into account. Similarly, the *off* model exhibits three states in left-right topology for modeling regions where no note is active. The three states adapt for silence periods of varying length. Then, a recognition network is generated that allows for detecting note events. Here, a simple two class segmentation into a sequence of alternating *on* and *off* regions can be obtained by a network that allows arbitrary transitions between the note on and note off model, see Figure 8.

The recognition network is trained by estimating optimal model parameters from training data using the Baum-

Welch algorithm. More precisely, we train the transition probabilities as well as emission probabilities. Here, four-component Gaussian mixture models turned out to be optimal for all states. All features are always normalized to have zero mean and standard deviation one. Furthermore, we add first and second order derivatives. In the recognition step, the Viterbi algorithm [67] is used for determining the most probable state sequence for each of the semitone bands independently.

As a result, one obtains a representation which – for each of the semitone bands – encodes the time positions with active notes. Here, the transition from the pre-onset state to the attack state defines the onset time. Likewise, the transition from the sustain to the first release state indicates the offset. This allows for deriving a transcription of the music recording, which encodes pitch and physical timings for each note. Furthermore, this transcription can be visualized in form of a piano-roll representation, see Figure 9.

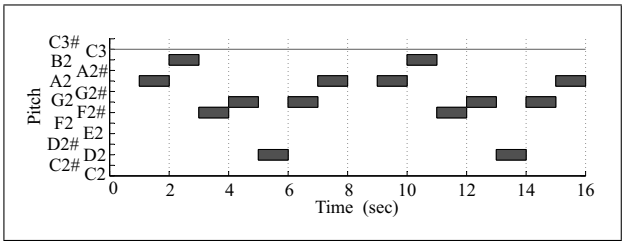

Figure 9. Example of the transcription of a recording visualized as a piano-roll representation. The rectangles encode time positions where a pitch is determined as active.

8. Experiments

In this section, we report on the experimental results of our music transcription system. We employ a large scale dataset of musical pieces of various genre. For evaluating the transcription quality one requires reference transcriptions to compare the computed results with. In particular, reliable and accurate annotations of all physical note parameters are absolutely essential. We use two different ways of obtaining reference data. On the one hand, we construct a database by synthesizing MIDI files to audio. On the other hand, we use the MIDI files to obtain reference data for real music recordings.

8.1. MIDI-Audio Synchronization

For obtaining reference transcriptions for real audio recordings, we employ a MIDI-audio synchronization technique [78, 3, 79]. The idea is to temporally align the MIDI note events with their corresponding physical occurrences in the audio recording [80]. The warped MIDI file can then be regarded as an annotation of the audio recording with the note events given by the MIDI file.

Most synchronization algorithms rely on some variant of dynamic time warping (DTW) and can be summarized as follows [3]. First, the MIDI file and audio recording are transformed into suitable feature sequences $V := (v_1, v_2, \ldots, v_N)$ and $W := (w_1, w_2, \ldots, w_N)$. In our implementation we employ chroma features [81]. Then, an $N \times M$ similarity matrix S is computed by evaluating a suitable similarity measure s: $S(n, m) = s(v_n, w_m)$ for $1 \leq n \leq N$ and $1 \leq m \leq M$, see Figure 10. Finally, the similarity maximizing alignment path is determined from S by dynamic programming. The alignment path assigns to each time frame in the MIDI file one or more time frames in the audio recording. This information is then used for temporally warping the MIDI files so that each MIDI note event corresponds to the physical position in the audio recording.

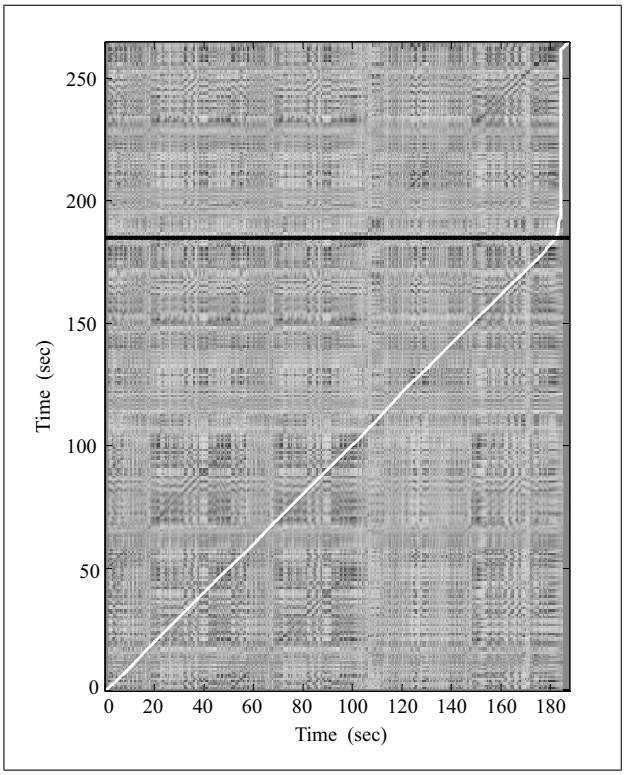

Figure 10. Similarity Matrix S of a MIDI (horizontal axis) and audio version (vertical axis) of Beethoven's *Für Elise*, with alignment path (white) and detected segment boundary (black). Dark colors indicate high similarity.

This method allows for obtaining reference transcription for the audio recordings. However, one drawback of this approach is that it force aligns any two feature sequences. Especially in complex classical music different representations often do not contain the same segment of the whole piece [82]. Only parts of the composition may be present in a recording, repetitions are inserted or left out. Here, we introduce criteria which allow for automatically expressing measures of the quality of the synchronization. First, regions with mainly diagonal steps are located on the best path. The underlying assumption is that identical parts in both sequences lead to diagonal steps in majority, while structural differences result in many horizontal or vertical steps. To discard such unstable areas, a sliding median of the slope of the alignment path is computed. Then, segments are determined where the median

slope stays within certain bounds. For the upper bound of the slope, 5 proved a good choice, and 0.2 for the lower bound. Figure 10 shows a similarity matrix S for a MIDI and an audio version of Beethoven's *Für Elise*. Here, the structural difference between the two representations is obvious. While the first part of the signals aligns well (diagonal steps of the alignment path), the acoustic recording contains an additional repetition that is not reflected in the MIDI file (vertical steps). However, the boundary (black) between these two segments is detected, which allows for recovering the consistent first segment.

Furthermore, we propose a measure of the overall quality of the alignment. The mean best path similarity is computed of all frame pairs on the best path. This value, however, is highly influenced by the timbral differences caused by different instrumentations. Therefore, we normalize the mean best path similarity with the mean overall similarity of the matrix S. This accounts for overall low similarity caused by timbral differences and quantifies the synchronization quality. Together with the segmentation, this measure allows for preselecting candidates of well aligned MIDI-audio pairs. The final selection of audio recordings for the dataset, however, was done manually to guarantee reference transcription with a reasonable accuracy. Although errors in the reference remain and the temporal accuracy is limited, this approach allows for evaluating the performance of the transcription system on real audio recordings. For enhancing the temporal accuracy of the reference transcription, high-resolution audio features that combine the high temporal accuracy of onset features with the robustness of chroma features [83] could be used.

8.2. Dataset Statistics

In our evaluation, we use a collection of 50 classical and 50 Rock/Pop music pieces. Covering such a broad range of musical types allows for evaluating the transcription quality under real world conditions. The Rock/Pop pieces are taken from the *MTV Europe Most Wanted 1990–2000*, the classical pieces from *100 Meisterwerke der klassischen Musik* (i. e. 100 Masterpieces of Classical Music). In the following, these datasets are referred to as `MTV` and `Classic`, respectively. For the 100 pieces, an audio recording and a MIDI file is available. The MIDI files explicitly encode a parameter representation of the piece and are now employed in two different ways. On the one hand, we use a software synthesizer[1] to obtain a synthesized audio version from the MIDI files. The ground truth of this audio is explicitly given by the MIDI files in high accuracy. The synthetic audio files are referred to as `syn` in the following. The synthesized audio files, however, lack some acoustic variance present in real recordings, such as different instruments, room conditions, or noise. Therefore, we create another dataset which contains the real audio recordings and is referred to as `real` in the following. As ground truth annotations are not available for these recordings, we employ the MIDI files for obtaining reference transcriptions through MIDI-audio synchronization, see section 8.1. Recall that structural differences between audio recordings and MIDI files are common. Here, only reliable segments are incorporated into the dataset. For some pieces, even more than one reliable segment is kept.

[1] http://timidity.sourceforge.net

Table I. Statistics on the datasets `MTV` and `Classic` used for evaluating the performance of the transcription system. `syn` notates the synthesized MIDI files and `real` the audio recordings with aligned transcriptions. Furthermore, the number of segments, the duration, number of notes, and mean note duration is given.

Corpus	#(Seg)	dur (min)	#(Notes)	∅(note dur) (sec)
$\texttt{MTV}_{\texttt{syn}}$	50	216	191933	0.53
$\texttt{Classic}_{\texttt{syn}}$	50	202	234208	0.59
$\texttt{MTV}_{\texttt{real}}$	59	118	116949	0.52
$\texttt{Classic}_{\texttt{real}}$	87	151	183294	0.60
overall		687	726384	

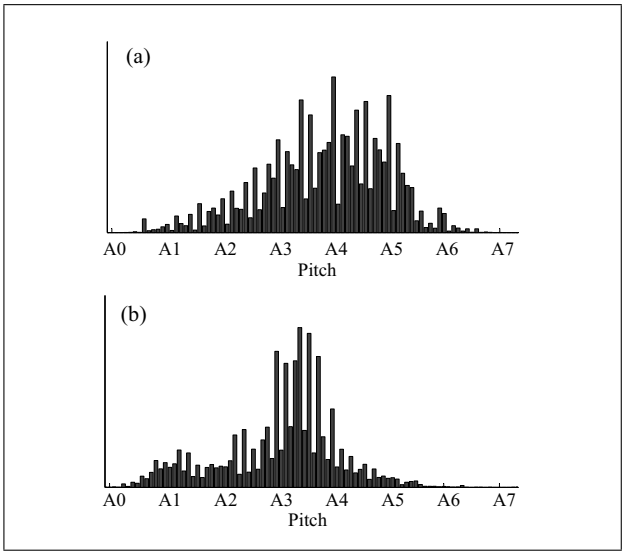

Figure 11. Histograms of pitch distributions (a) `MTV` and (b) `Classic`.

Altogether, our evaluation dataset consists of four parts. On the one hand we have $\texttt{MTV}_{\texttt{syn}}$ and $\texttt{Classic}_{\texttt{syn}}$, each consisting of 50 audio files synthesized from the MIDI files. On the other hand we have $\texttt{MTV}_{\texttt{real}}$ and $\texttt{Classic}_{\texttt{real}}$, which contain real audio recordings of the same musical pieces, as well as temporally warped MIDI files providing the reference transcriptions. Statistics of the datasets are shown in Table I and the histograms of pitch distributions for `MTV` and `Classic` are shown in Figure 11. Some differences between the two datasets are obvious: the mean note duration of `MTV` database is shorter than in `Classic` indicating higher average tempo. Furthermore, the overall variance of the pitch distribution is lower for `MTV` than for `Classic`. However, there is a larger amount of notes with very low pitches indicating the common use of bass in this dataset.

8.3. Evaluation Measures

To evaluate the performance of the transcription system, evaluation measures are employed which express the quality of a transcription result. Following [31] we evaluate the accuracy in terms of pitch and timing individually. First, a detected note is assigned to a reference note, when both have the same pitch and partly overlap in time. In the case that multiple detected notes overlap with one reference note, the detected note with minimal onset distance is assigned to the reference note. Each detected note can only be assigned to one reference note. A reference note is then considered a *correct detection* (CD) if there is a detected note assigned to it, otherwise a *false negative* (FN). All detected notes which are not assigned to any reference note are considered a *false positive* (FP). From this one obtains precision P and recall R,

$$\mathrm{P} = \frac{\#(\mathrm{CD})}{\#(\mathrm{CD}) + \#(\mathrm{FP})}, \qquad \mathrm{R} = \frac{\#(\mathrm{CD})}{\#(\mathrm{CD}) + \#(\mathrm{FN})}. \tag{11}$$

From these the F-measure $\mathrm{F} = 2 \cdot \mathrm{P} \cdot \mathrm{R}/(\mathrm{P} + \mathrm{R})$ is derived. All values are computed separately for each musical piece and the final values are obtained by averaging over all pieces of the respective dataset.

Furthermore, we define the *overlap ratio* OR which describes the ratio of temporal overlap between all reference notes considered a correct detection and the assigned detected notes. To this end, we determine the onset times $t_{\mathrm{ref}}^{\mathrm{on}}$ / $t_{\mathrm{det}}^{\mathrm{on}}$ and offset times $t_{\mathrm{ref}}^{\mathrm{off}}$ / $t_{\mathrm{det}}^{\mathrm{off}}$ for the reference / detected notes. The overlap ratio OR is then defined as

$$\mathrm{OR} = \frac{\min\left(t_{\mathrm{ref}}^{\mathrm{off}}, t_{\mathrm{det}}^{\mathrm{off}}\right) - \max\left(t_{\mathrm{ref}}^{\mathrm{on}}, t_{\mathrm{det}}^{\mathrm{on}}\right)}{\max\left(t_{\mathrm{ref}}^{\mathrm{off}}, t_{\mathrm{det}}^{\mathrm{off}}\right) - \min\left(t_{\mathrm{ref}}^{\mathrm{on}}, t_{\mathrm{det}}^{\mathrm{on}}\right)}. \tag{12}$$

for each reference note of the respective dataset and finally averaged. In other words, the overlap ratio expresses the temporal accuracy of the notes with a correct pitch estimate.

8.4. Results and Discussion

We now summarize and discuss our experimental result. Recall from section 7 that our transcription system exhibits components that require a training step. Therefore, we employ a three-fold cross validation in the evaluation. To this end, the overall dataset (consisting of 100 pieces) is separated into three parts with an equal number of pieces from `MTV` and `Classic`. The three parts contain both, the `syn` and `real` versions of the songs.

We start with discussing Table II. Here, we compare the results one can obtain from the pitch estimator described in section 5.3 and from the HMM based note event modeling described in section 7. For *Pitch estimation* we simply determine continuous notes from the pitch candidate spectrogram C introduced in section 5.3. Exclusively relying on a pitch estimator, without any further post-processing of the pitch estimates, we obtain an F-measure of $\mathrm{F} = 0.618$. This indicates that roughly 60% of all detected notes are correct. The mean overlap ratio of the correct notes is only $\mathrm{OR} = 0.210$. This indicates that the temporal accuracy of the transcription is rather low. The reason for this is that C typically contains errors, see Figure 4. In the case of the proposed *Note event modeling*, however, the overlap ratio significantly increases to $\mathrm{OR} = 0.425$. Obviously, a pitch estimation alone is not sufficient for obtaining a robust transcription. Incorporating temporal aspects as a post-processing, the HMMs introduce a kind of context-dependent smoothing that greatly enhances the transcription results and in particular the temporal accuracy of the note events. Note that the overall F-measure could be further improved using a more advanced pitch estimator. In our framework, however, we use a basic (and computationally efficient) variant, focusing on the proposed HHM-based approach to note event modeling.

Table II. Results for the pitch estimation algorithm and the note event modeling approach.

Method	F	P	R	OR
Pitch estimation	0.618	0.639	0.597	0.210
Note event modeling	0.613	0.633	0.592	0.425

Table III. Comparison of the results of the note event modeling using different combination of features: the pitch candidate spectrogram C, the semitone spectrogram S, and the semitone novelty curves Δ^+ and Δ^-.

Feature set	F	P	R	OR
C	0.506	0.583	0.480	0.231
$C\ S$	0.554	0.684	0.464	0.337
$C\ \Delta^+\ \Delta^-$	0.613	0.633	0.592	0.425

In a second experiment, we investigate the performance of different features sets for the note event modeling approach. More precisely, the features we use are the semitone spectrogram S (see section 5.2), the pitch candidate spectrogram C (see section 5.3), as well as the novelty curves Δ^+ and Δ^- (see section 6). Table III summarizes the results one obtains for different combinations of these features. Obviously, exclusively relying on C does not allow for obtaining an accurate transcription ($\mathrm{F} = 0.506$, $\mathrm{OR} = 0.231$). The reason for this is that the pitch candidate spectrogram C is only a sparse representation of the pitch candidates that typically exhibits errors. These errors cannot be corrected through note event modeling without further information. When combining C with subband magnitude information from the semitone spectrogram S, the F-measure already increases from $\mathrm{F} = 0.506$ to $\mathrm{F} = 0.554$. Likewise, the overlap ratio improves from $\mathrm{OR} = 0.231$ to $\mathrm{OR} = 0.337$. Intuitively, by combining pitch information supplied by C and information about the magnitude in the respective semitone band supplied by S, the robustness of the transcription is enhanced. One notices even further improvements when combining C with the complex domain novelty curves Δ^+ and Δ^- ($\mathrm{F} = 0.613$ and $\mathrm{OR} = 0.425$). This result shows that the combination of magnitude and phase information even better captures

Table IV. Detailed results of the four parts of the dataset using the note event modeling approach.

Corpus	F	P	R	OR
$\texttt{MTV}_{\texttt{syn}}$	0.629	0.643	0.610	0.361
$\texttt{Classic}_{\texttt{syn}}$	0.683	0.732	0.628	0.513
$\texttt{MTV}_{\texttt{real}}$	0.527	0.525	0.549	0.312
$\texttt{Classic}_{\texttt{real}}$	0.604	0.650	0.568	0.471

the temporal properties of notes than relying on the magnitude alone.

Finally, we investigate the transcription quality on different types of audio data. Table IV shows detailed results for the four different parts of our evaluation database. One observes a noticeable difference in the transcription quality between the synthetic files `syn` and the original recordings `real`. For example, for $\texttt{MTV}_{\texttt{syn}}$ one obtains F = 0.629 and OR = 0.361. For $\texttt{MTV}_{\texttt{real}}$, however, the F-measure decreases to F = 0.527 and the overlap ratio drops to OR = 0.312. The main reason for this effect is that the synthesized audio files lack many of the acoustic variations of the real audio recordings. For example, distorted electric guitars are a great challenge for a music transcription system because the resulting sounds contain many inharmonic components. Similarly, singing voices typically exhibit spectral/temporal variations such as tremolo or vibrato. As a result, the transcription of such recordings is problematic. In the case of synthetic audio files, however, these sounds are not reproduced with all the variations. Thus, the transcription of synthetic audio is less problematic. Similar effects lead to the decline from $\texttt{Classic}_{\texttt{syn}}$ (F = 0.683, OR = 0.513) to $\texttt{Classic}_{\texttt{real}}$ (F = 0.604, OR = 0.471).

At this point one has to address the problem of the accuracy of the reference annotations. In the case of `syn`, the annotations are perfectly aligned to the audio. For `real`, however, there remain uncertainties. Naturally, a lower accuracy of the annotations may have an influence on the evaluation. A manual inspection of the annotations revealed that the accuracy is sufficient for the pitch based evaluation (F-measure values), where the temporal accuracy is not crucial, see section 8.3. However, synchronization inaccuracies may have an impact on the OR values for `real`.

Continuing the discussion of Table IV, we notice a significantly lower quality for `MTV` music than for `Classic`. e. g., $F = 0.604$ for $\texttt{Classic}_{\texttt{real}}$ and $F = 0.527$ for $\texttt{MTV}_{\texttt{real}}$. In other words, the transcription of classical music seems to be "easier" than the transcription of Pop/Rock music. This is somehow surprising, as classical music tends to be more complex in terms of harmony and melody. In fact, this effect leads to a low transcription quality for complex orchestral music. On the other hand, however, our datasets contains a variety of solo instrument recordings. Especially piano pieces are of limited complexity, exhibiting only slight temporal and spectral variations. For these recordings, the transcription quality is significantly higher than for orchestral pieces. The situation is different with popular music. These recordings typically exhibit pronounced percussive instruments such as drums. Furthermore, the concepts of harmony and melody play a less important role in popular music. As a result, the signal contains more inharmonic and noise-like components that mask harmonic parts and make a transcription difficult. Also, OR is significantly lower for $\texttt{MTV}_{\texttt{real}}$ (OR = 0.312) than for $\texttt{Classic}_{\texttt{real}}$ (OR = 0.471). A further inspection showed a dependency between OR and the tempo of a piece. Here, for faster pieces with a shorter mean note duration, the exact determination of note onsets/durations is problematic. The mean tempo of $\texttt{MTV}_{\texttt{real}}$ is higher than the mean tempo of $\texttt{Classic}_{\texttt{real}}$. Consequently, OR is lower for this kind of music.

9. Conclusion

In this article, we presented an automatic music transcription system for deriving a symbolic representation from a given music recording. In particular, our system is designed to cope with a wide range of Western music based on the equal tempered scale. Throughout the framework, we employ an effective music signal representation that accounts for the logarithmic properties of the pitch scale. As one main advantage, this representation constitutes a musically meaningful data reduction.

Pitch estimation is the most important step in music transcription. However, as our evaluation reveals, this step is not sufficient for obtaining meaningful transcription results. Here, the introduced post-processing on the basis of HMMs incorporates a context-dependent smoothing and significantly improves the transcription accuracy. In this context, modeling the temporal properties of notes in the complex domain using magnitude and phase information further improves the quality of the transcription. As another important result, we showed that evaluating a transcription system on the basis of synthetic audio data is not a realistic assumption. Using real audio data, the quality one can expect from a transcription is significantly lower. Obtaining ground truth annotations for music recordings, however, is problematic. Here, our approach of using a MIDI-audio synchronization procedure for creating reference transcriptions turned out to be a valuable alternative to using synthetic audio material. Furthermore, our results showed that the transcription quality drastically differs between musical genre. For classical music of limited complexity, such as piano music, one can obtain a reasonable transcription quality. For popular music with percussive instruments, however, one gets unsatisfying results in many cases. Here, source separation techniques [84, 85] for separating harmonic and percussive instruments may be beneficial. Specialized transcription systems may account for various aspects of music. For example, one might first employ a drum transcription to handle percussive instruments [86, 87] and then a pitch-based transcription of the harmonic components.

As the quality of our transcriptions shows, there is still room for improvements. More refined approaches to pitch

estimation [24] and onset detection [34] may enhance the transcription quality. Another approach to further improve the results is to equip the system with high-level musical knowledge. example, certain pitches are more likely to occur, than others. To account for such information, one straight forward way is to give higher weights to these pitches in the computation of the weighted sum in equation (2). In our experiments, however, this modification only led to marginal improvements. One reason for this is that the majority of pitch estimation errors are actually octave confusions that do not benefit from the key information. musical piece is another valuable information. For example, the pitch distributions shown in Figure 11 indicate different characteristics for popular and classical music. This observation may be exploited by adaptively accentuating certain pitches that are frequently used in music of the respective genre. In our experiments, this approach lead to slight improvements of the transcription quality. Finally, knowing the tempo and beat positions is valuable information for the transcription. Typically, musical events do not occur randomly in time but are highly structured and aligned to an underlying rhythmic grid. Using an automatic beat tracker [39, 88, 61, 37] for extracting the beats one can emphasize note onset and offset positions in the note event modeling. In our experiments, such information lead to a reasonable improvement of temporal accuracy.

Although such musical knowledge is beneficial for music transcription, the automatic extraction of such high-level information is problematic itself. Beat tracking, for example, is robust for popular music exhibiting percussive instruments and a steady tempo. For classical music, however, without percussive instruments and with changing tempo, current beat tracker still have significant problems in accurately capturing the beats [89]. On the other hand, having a robust transcription of a recording, such musical knowledge can be easily extracted from the symbolic representation. For the future, one might consider a transcription system which jointly estimates note parameters and musical knowledge, where both parts support each other for enhanced robustness [90]. Similarly, one may exploit melodic or harmonic models defining likely note progressions [31]. Further adapted to rhythm [91] or genre [92], these models could lead to a significant improvement of the quality of automatic music transcription.

References

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

[2] H. L. von Helmholtz: On the sensations of tone. Dover, New York, 1954.

[3] M. Müller: Information retrieval for music and motion. Springer, 2007.

[4] A. Klapuri: Multiple fundamental frequency estimation by summing harmonic amplitudes. Proceedings of the International Conference on Music Information Retrieval (ISMIR), 2006, 216–221.

[5] C. Duxbury, J. P. Bello, M. Davies, M. B. Sandler: Complex domain onset detection for musical signals. Proceedings of the International Conference on Digital Audio Effects (DAFx), 2003.

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

[7] J. P. Bello, L. Daudet, M. B. Sandler: Automatic piano transcription using frequency and time-domain information. IEEE Transactions on Audio, Speech and Language Processing **14** (2006) 2242–2251.

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

[9] M. Marolt: Adaptive oscillator networks for partial tracking and piano music transcription. Proceedings of the International Computer Music Conference (ICMC), 2000.

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

[11] M. Marolt, S. Divjak: On detecting repeated notes in piano music. Proceedings of the International Conference on Music Information Retrieval (ISMIR), 2002.

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

[13] M. Ryynänen, A. Klapuri: Automatic transcription of melody, bass line, and chords in polyphonic music. Computer Music Journal **32** (2008) 72–86.

[14] M. Goto: A real-time music-scene-description system: predominant-F0 estimation for detecting melody and bass lines in real-world audio signals. Speech Communication **43** (2004) 311–329.

[15] J. Paulus, A. Klapuri: Drum sound detection in polyphonic music with hidden Markov models. EURASIP J. Audio Speech Music Process (2009).

[16] L. R. Rabiner, M. Cheng, A. Rosenberg, C. McGonegal: A comparative performance study of several pitch detection algorithms. IEEE Transactions on Acoustics, Speech and Signal Processing **24** (1976) 399–418.

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

[18] J. P. Bello, G. Monti, M. B. Sandler: Techniques for automatic music transcription. Proceedings of the International Conference on Music Information Retrieval (ISMIR), 2000.

[19] N. Geckinli, D. Yavuz: Algorithm for pitch extraction using zero-crossing interval sequence. IEEE Transactions on Acoustics, Speech and Signal Processing **25** (1977) 559–564.

[20] A. M. Noll: Cepstrum pitch detection. Journal of the Acoustical Society of America **41** (1967) 293–309.

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

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

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

[24] A. Klapuri: Multipitch analysis of polyphonic music and speech signals using an auditory model. IEEE Transactions on Audio, Speech, and Language Processing **16** (2008) 255–266.

[25] T. Tolonen, M. Karjalainen: A computationally efficient multipitch analysis model. IEEE Transactions on Speech and Audio Processing **8** (2000) 708–716.

[26] 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** (1993) 3271–3290.

[27] A. Klapuri: Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. IEEE Transactions on Speech and Audio Processing **11**.

[28] A. Klapuri: A perceptually motivated multiple-F0 estimation method. Proceedings International Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Paltz, NY, 2005.

[29] G. E. Poliner, D. P. W. Ellis: A classification approach to melody transcription. Proceedings of the International Conference on Music Information Retrieval (ISMIR), 2005, 161–166.

[30] E. Vincent, X. Rodet: Music transcription with ISA and HMM. Proceedings of Fifth International Conference in Independent Component Analysis and Blind Signal Separation, ICA, 2004, 1197–1204.

[31] M. Ryynänen, A. Klapuri: Polyphonic music transcription using note event modeling. Proceedings International Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Paltz, New York, USA, 2005, 319–322.

[32] H. Takeda, T. Otsuki, N. Saito, M. Nakai, H. Shimodaira, S. Sagayama: Hidden Markov Models for automatic transcription of MIDI signals. Proceedings of IEEE Workshop on Multimedia Signal Processing, 2002, 428–431.

[33] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, M. B. Sandler: A tutorial on onset detection in music signals. IEEE Transactions on Speech and Audio Processing **13** (2005) 1035–1047.

[34] R. Zhou, M. Mattavelli, G. Zoia: Music onset detection based on resonator time frequency image. IEEE Transactions on Audio, Speech, and Language Processing **16** (2008) 1685–1695.

[35] N. Collins: Using a pitch detector for onset detection in:. Proceedings of the International Conference on Music Information Retrieval (ISMIR), London, UK, 2005, 100–106.

[36] A. J. Eronen, A. P. Klapuri: Music tempo estimation with k-NN regression. IEEE Transactions on Audio, Speech, and Language Processing **18** (2010) 50–57.

[37] M. Goto: An audio-based real-time beat tracking system for music with or without drum-sounds. Journal of New Music Research **30** (2001) 159–171.

[38] A. Holzapfel, Y. Stylianou: Beat tracking using group delay based onset detection. Proceedings of the International Conference on Music Information Retrieval (ISMIR), 2008.

[39] E. D. Scheirer: Tempo and beat analysis of acoustical musical signals. Journal of the Acoustical Society of America **103** (1998) 588–601.

[40] P. Masri, A. Bateman: Improved modeling of attack transients in music analysis-resynthesis. Proceedings of the International Computer Music Conference (ICMC), Hong Kong, 1996, 100–103.

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

[42] J. P. Bello, M. B. Sandler: Blackboard systems and top-down processing for the transcription of simple polyphonic music. Proceedings of the International Conference on Digital Audio Effects (DAFx), 2000.

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

[44] H. Fastl, E. Zwicker: Psychoacoustics, facts and models, 3rd edition. Springer-Verlag, Berlin, Heidelberg, New York, 2007.

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

[46] M. Slaney: A critique of pure audition. Computational auditory scene analysis (1998) 27–41.

[47] G. Reis, F. Vega: A novel approach to automatic music transcription using electronic synthesis and genetic algorithms. GECCO '07: Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation, New York, NY, USA, 2007, 2915–2922.

[48] E. Benetos, S. Dixon: Multiple-instrument polyphonic music transcription using a convolutive probabilistic model. Proceedings of Sound and Music Computing Conference (SMC), Padova, Italy, 2011, 19–24.

[49] K. Miyamoto, H. Kameoka, H. Takeda, T. Nishimoto, S. Sagayama: Probabilistic approach to automatic music transcription from audio signals. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vol. 2, 2007, II–697–II–700.

[50] E. Kapanci, A. Pfeffer: Signal-to-score music transcription using graphical models. – In: Proceedings of the International Joint Conference on Artificial Intelligence. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005, 758–765.

[51] P. Peeling, C. Fai Li, S. Godsill: Poisson point process modeling for polyphonic music transcription. Journal of the Acoustic Society of America Express Letters (2007) EL168–EL175.

[52] A. T. Cemgil, S. J. Godsill, P. H. Peeling, N. Whiteley: Bayesian statistical methods for audio and music processing. – In: The Oxford Handbook of Applied Bayesian Analysis. Oxford University Press, 2010.

[53] P. Cancela, M. Rocamora, E. López: An efficient multi-resolution spectral transform for music analysis. Proceedings of the International Conference on Music Information Retrieval (ISMIR), Kobe, Japan, 2009.

[54] P. Masri, A. Bateman, N. Canagarajah: A review of time-frequency representations, with application to sound/music analysis/resynthesis. Organised Sound **2** (1997) 193–205.

[55] J. C. Brown: Calculation of a constant Q spectral transform. Journal of the Acoustical Society of America **89** (1991) 425–434.

[56] F. Kurth, M. Clausen: Filter bank tree and m-band wavelet packet algorithms in audio signal processing. IEEE Transactions on Signal Processing **47** (1999) 549–554.

[57] K. Dressler: Sinusoidal extraction using an efficient implementation of a multi-resolution FFT. Proceedings of the International Conference on Digital Audio Effects (DAFx), Montreal, Canada, 2006.

[58] R. Martin: Spectral subtraction based on minimum statistics. Proceedings Euro. Signal Processing Conference (EUSIPCO), 1994, 1182–1185.

[59] H. Hermansky, N. Morgan: RASTA processing of speech. IEEE Transactions on Speech and Acoustics **2** (1994) 587–589.

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

[61] D. P. W. Ellis, G. E. Poliner: Identifying 'cover songs' with chroma features and dynamic programming beat tracking. Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vol. 4, Honolulu, Hawaii, USA, 2007.

[62] S. Hainsworth, M. D. Macleod, P. J. Wolfe: Analysis of reassigned spectrograms for musical transcription. Proceedings International Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 2001.

[63] S. A. Fulop, K. Fitz: Algorithms for computing the time-corrected instantaneous frequency (reassigned) spectrogram, with applications. Journal of the Acoustical Society of America **119** (2006) 360–371.

[64] J. L. Flanagan, R. M. Golden: Phase vocoder. Bell Systems Technical Journal **45** (1966) 1493–1509.

[65] D. Stowell, M. D. Plumbley: Adaptive whitening for improved real-time audio onset detection. Proceedings of the International Computer Music Conference (ICMC), Copenhagen, Denmark, 2007, 312–319.

[66] J. Yin, A. Dhanik, D. Hsu, Y. Wang: The creation of a music-driven digital violinist. Proceedings of the 12th annual ACM international conference on Multimedia, New York, NY, USA, 2004, 476–479.

[67] L. R. Rabiner: A tutorial on Hidden Markov Models and selected applications in speech recognition. – In: Readings in Speech Recognition. A. Waibel, K.-F. Lee (eds.). Kaufmann, San Mateo, CA, 1990, 267–296.

[68] B. Schuller, G. Rigoll, M. Lang: Matching monophonic audio clips to polyphonic recordings. Proceedings of 31st DAGA, Munich, Germany, 2005, 299–300.

[69] B. Schuller, G. Rigoll, M. Lang: HMM-based music retrieval using stereophonic feature information and frame-length adaptation. Proceedings International Conference on Multimedia and Expo (ICME), Baltimore, MD, USA, 2003, 713–716.

[70] A. Sheh, D. Ellis: Chord segmentation and recognition using EM-trained Hidden Markov Models. Proceedings of the International Conference on Music Information Retrieval (ISMIR), Baltimore, Maryland, USA, 2003.

[71] K. Noland, M. Sandler: Key estimation using a Hidden Markov Model. Proceedings of the International Conference on Music Information Retrieval (ISMIR), Victoria, Canada, 2006, 121–126.

[72] K. Lee, M. Slaney: Acoustic chord transcription and key extraction from audio using key-dependent HMMs trained on synthesized audio. IEEE Transactions on Audio, Speech and Language Processing **16** (2008) 291–301.

[73] B. Schuller, F. Eyben, G. Rigoll: Beat-synchronous data-driven automatic chord labeling. Proceedings of the DAGA 2008, Dresden, Germany, 2008, 555–556.

[74] M. Levy, M. Sandler: Structural segmentation of musical audio by constrained clustering. IEEE Transactions on Audio, Speech and Language Processing **16** (2008) 318–326.

[75] F. J. Cañadas Quesada, N. Ruiz Reyes, P. Vera Candeas, J. J. Carabias, S. Maldonado: A multiple-F0 estimation approach based on gaussian spectral modelling for polyphonic music transcription. Journal of New Music Research **39** (2010) 93–107.

[76] S. A. Raczyński, E. Vincent, F. Bimbot, S. Sagayama: Multiple pitch transcription using DBN-based musicological models. Proceedings of the International Conference on Music Information Retrieval (ISMIR), Utrecht, NL, 2010.

[77] C. Raphael: Automatic transcription of piano music. Proceedings of the International Conference on Music Information Retrieval (ISMIR), Paris, France, 2002, 15–19.

[78] N. Hu, R. B. Dannenberg, G. Tzanetakis: Polyphonic audio matching and alignment for music retrieval. Proceedings International Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 2003.

[79] M. Müller, F. Kurth, T. Röder: Towards an efficient algorithm for automatic score-to-audio synchronization. Proceedings of the International Conference on Music Information Retrieval (ISMIR), Barcelona, Spain, 2004, 365–372.

[80] R. Turetsky, D. Ellis: Ground-truth transcriptions of real music from force-aligned midi syntheses. Proceedings of the International Conference on Music Information Retrieval (ISMIR), 2003.

[81] M. A. Bartsch, G. H. Wakefield: Audio thumbnailing of popular music using chroma-based representations. IEEE Transactions on Multimedia **7** (2005) 96–104.

[82] S. Ewert, M. Müller, R. B. Dannenberg: Towards reliable partial music alignments using multiple synchronization strategies. Proceedings of the International Workshop on Adaptive Multimedia Retrieval (AMR), Madrid, Spain, 2009.

[83] S. Ewert, M. Müller, P. Grosche: High resolution audio synchronization using chroma onset features. Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Taipei, Taiwan, 2009, 1869–1872.

[84] N. Ono, K. Miyamoto, H. Kameoka, S. Sagayama: A real-time equalizer of harmonic and percussive components in music signals. International Conference on Music Information Retrieval (ISMIR), Philadelphia, PA, USA, 2008, 139–144.

[85] B. Schuller, F. Eyben, G. Rigoll: Blind enhancement of the rhythmic and harmonic sections by NMF: Does it help? Proceedings International Conference on Acoustics (NAG/DAGA 2009), Rotterdam, The Netherlands, 2009.

[86] J. Paulus, A. Klapuri: Combining temporal and spectral features in HMM-based drum transcription. Proceedings of the International Conference on Music Information Retrieval (ISMIR), Vienna, Austria, 2007.

[87] O. Gillet, G. Richard: Transcription and separation of drum signals from polyphonic music,. IEEE Transactions on Audio, Speech, and Language Processing **16** (2008) 529–540.

[88] B. Schuller, F. Eyben, G. Rigoll: Tango or waltz? putting ballroom dance style into tempo detection. EURASIP Journal on Audio, Speech, and Music Processing, 2008, 12 pages.

[89] P. Grosche, M. Müller, C. S. Sapp: What makes beat tracking difficult? A case study on Chopin Mazurkas. Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR), Utrecht, Netherlands, 2010.

[90] C. Raphael: A graphical model for recognizing sung melodies. International Conference on Music Information Retrieval (ISMIR), London, UK, 2005, 658–663.

[91] A. P. Klapuri: Musical meter estimation and music transcription. Proceedings of the Cambridge Music Processing Colloquium, 2003, 40–45.

[92] G. Tzanetakis: Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing **10** (2002) 293–302.