Toward a Human-Centric Automatic Piano Music Transcription System

by

Andrea Cogliati

Submitted in Partial Fulfillment of the

Requirements for the Degree

Doctor of Philosophy

Supervised by Professor Zhiyao Duan

Department of Electrical and Computer Engineering

Arts, Sciences and Engineering

Edmund A. Hajim School of Engineering and Applied Sciences

University of Rochester

Rochester, New York

2018

To my dad.

Table of Contents

Biographical Sketch

The author was born in Milano, Italy. He attended University of Pisa and Scuola Normale Superiore, and graduated with a Bachelor of Science degree (*Laurea*) and a Master of Science degree (*Diploma*) in Mathematics, with a dissertation on Global Optimization in R^n. He spent almost 20 years working in the IT industry as a consultant and trainer, focusing mainly on software security. He began doctoral studies in Electrical and Computer Engineering at the University of Rochester in 2012. He received a Master of Science degree in Electrical and Computer Engineering from the University of Rochester in 2014. He pursued his research in Music Information Retrieval under the direction of Professor Zhiyao Duan.

The following publications were a result of work conducted during doctoral study:

Journal Papers

- Andrea Cogliati, Zhiyao Duan, and Brendt Wohlberg. Piano Transcription with Convolutional Sparse Lateral Inhibition. *IEEE Signal Processing Letters*, 24(4): 392–396, 2017. [1].

- Andrea Cogliati, Zhiyao Duan, and Brendt Wohlberg. Context-dependent Piano Music Transcription with Convolutional Sparse Coding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2016. [2].

- Andrea Cogliati, Cristina Canavesi, Adam Hayes, Patrice Tankam, Virgil-Florin Duma, Anand Santhanam, Kevin P Thompson, Jannick P Rolland. MEMS-based handheld scanning probe with pre-shaped input signals for distortion-free images in Gabor-domain optical coherence microscopy. *Optics express*, 24(12): 13365-13374, 2016. [3].

Peer-Reviewed Conference Papers

- Andrea Cogliati and Zhiyao Duan. A metric for Music Notation Transcription Accuracy. *Proc. of International Society for Music Information Retrieval (ISMIR)*, 2017. [4].

- Andrea Cogliati, David Temperley, and Zhiyao Duan. Transcribing human piano performances into music notation. *Proc. of International Society for Music Information Retrieval (ISMIR)*, 2016. [5]

- Andrea Cogliati, Zhiyao Duan, and Brendt Wohlberg. Piano music transcription with fast convolutional sparse coding. In *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*, pages 1–6. IEEE, 9 2015. [6].

- Andrea Cogliati and Zhiyao Duan. Piano Music Transcription Modeling Note Temporal Evolution. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 429–433, Brisbane, Australia, 4 2015. [7].

Patents

Andrea Cogliati, Zhiyao Duan, Brendt Wohlberg. "Context-dependent Piano Music Transcription with Convolutional Sparse Coding." U.S. Patent 9,779,706 B2, issued October 3, 2017.

Acknowledgments

Writing acknowledgments is always an arduous task, as one has to navigate through the multiple risks of repeating platitudes, sounding insincere, or forgetting someone along the journey.

I will start by thanking my advisor, Professor Zhiyao Duan, for his unwavering support during the past five years. It must not have been easy to supervise such an atypical student as myself as his first doctoral student. I cannot thank him enough for his patience, understanding, and knowledge. He introduced me to the wonderful community of Music Information Retrieval, and always had great words for me. It has been a long journey, with many unexpected twists, but I hope it was as rewarding for him as it was for me.

I also wish to thank my collaborators Professor David Temperley and Doctor Brendt Wohlberg, for guiding important pieces of my research. Their insights were instrumental for the results that we obtained. Our collaboration was always enjoyable and productive, and I hope they agree.

I want to thank Professor Mark Bocko for supervising the beginning of my adventure as a graduate student at the University of Rochester. Our discussions were always engaging and insightful. He taught me to always think and dream big, and his advice will guide me for the rest of my professional journey.

I thank all my colleagues at the AIR Lab. Thanks for sharing your research with me, thanks for your advice and encouragement. I am proud to be the first to graduate from our lab, and I look forward to seeing your future successes as well.

I cannot end my acknowledgements for the academic part of my life without thanking Professor Franco Flandoli, who supervised my undergraduate thesis, which I wrote exactly twenty years ago. Professor Flandoli was a great mentor and an inspiration for his knowledge, humility, and willingness to always help his students. He is one of my examples of great teachers.

A special thank you to Michele Foster, who was always helpful and responsive. She made my life as a graduate student so much easier.

Among the many people, supervisors, and colleagues, who helped and supported me in my professional life, I want to especially thank Adalberto Fontana, Stefano Maruzzi, Maurizio Liverani, Carla Anghileri, Shelley Weiner, Nancy Curtis, Brian Wilson, Professor Jannick Rolland, Professor Kevin Rolland-Thompson, and Professor Ted Pawlicki. They all believed in me at some point, and I will always be grateful for their trust.

I cannot thank my parents enough for all they gave to me, both materially and spiritually, so I will only thank them for always working so hard to provide me with the best education possible. Being far away is always hard, and these past few months have been especially testing for all of us. Just another reminder that lost moments will never come back but to haunt you.

I want to thank my brother Guido, my sister-in-law Sabina, and their kids Irene, Sara, and Pietro. They have always been there when I needed them, and they have been there for my mom and dad in the past few months when I could not. They are a wonderful family, and they keep reminding me what family really means.

I would not be here without the help and support of my in-laws, Luisa and Giuliano. They welcomed me into their family, and always made me feel proud of it.

Finally I want to thank my wife, Cristina, my better half, my angel, my down-to-earth companion, and my crazy partner in crime. She showed me how to overcome all difficulties and achieve our goals with hard work, focus, passion, compassion, and love. She is the one who, above anyone else, truly personifies our *alma mater* motto, *Meliora*!

Abstract

Music Transcription is the process of notating a music piece solely by hearing. It is the highest skill acquired by musicians during the ear training process, as it involves the identification of pitches, intervals, melodies, chords, and rhythms, along with a mature understanding of music theory. Some of the skills involved in transcription, such as recognizing melodies and rhythms, are natural for most people, but extremely difficult for machines. The goal of my research is to design a computer system that automatically transcribes a piece of music into standard music notation, which is the most natural notation for musicians.

I call this system *human-centric* as its purpose is mainly to assist musicians in being more precise and effective and to speak to them in the language they are most familiar with, i.e., common music notation. Amateur musicians might use the system to improve their skills, while professional musicians, such as composers, might appreciate the ability to quickly notate improvisation sessions on a piano.

The core of the proposed method is a high-accuracy music transcription system for piano music in a context-specific setting. That is, the system has to be trained on the same instrument and in the same acoustic context, i.e., room and microphone position, that will be used for the transcription. This constraint allows to increase the accuracy of the transcription above the accuracy of state-of-the-art transcription systems, which cannot be reliably used for practical applications.

The system displays the transcription in music notation in an interactive editor, so that the transcription can be edited and corrected. Music notation output is not common for automatic music transcription systems, which are mostly focused on the *parametric transcription*, i.e., determining the played pitches, their onsets and offsets. As a consequence, there is no way to objectively evaluate the accuracy of a transcription in music notation. For my research, I designed an algorithm to convert a parametric transcription into music notation. I also propose an objective evaluation metric that can predict human evaluation of a music notation transcription.

Finally, I outline a possible generalization of the transcription system, in order to relax the context-specific constraint and transcribe music played on different instruments.

Contributors and Funding Sources

This work was supported by a dissertation committee consisting of Professor Zhiyao Duan (advisor), and Professor Mark Bocko of the Department of Electrical and Computer Engineering, Professor David Temperley of the Department of Music Theory of the Eastman School of Music, and Doctor Brendt Wohlberg of the Los Alamos National Laboratory. All other work conducted for the dissertation was completed by the student independently.

Graduate study was initially funded by the Department of Electrical and Computer Engineering. Starting the second year, it was funded by the startup funds of Professor Zhiyao Duan. The last year of the graduate study was also partially funded by the Goergen Institute for Data Science Collaborative Pilot Award Program in Health Analytics on "Predicting Adverse Events From Cardiac Signals using Deep Neural Networks".

List of Tables

List of Figures

Chapter 1

Introduction

This thesis presents a complete piano music transcription system, with high accuracy and robustness to common environmental conditions. This chapter explains the motivations of the research, in Section 1.1 and the structure of the thesis, in Section 1.2. Finally, it lists the main contributions of this work, in Section 1.3.

1.1 Motivations and Aims

Automatic Music Transcription (AMT) is one of the most fundamental problems in Music Information Retrieval, the domain of retrieving information about music, and Computer Audition, the science of audio understanding by computers [11]. AMT involves inferring a symbolic music representation, such as a music score or a MIDI piano-roll, purely from an audio signal, typically a recorded performance of a music piece or song. Trained musicians learn the relevant aural skills during ear training classes, and are expected to be able to accurately transcribe a four-part piece after listening to it a few times. Even untrained people are generally able to listen to a song, and recognize and repeat its melody, thus showing basic aural skills. AMT has been studied for several decades since the first published papers on the topic [12, 13],

however computers can still not match the performance of trained musicians in terms of transcription accuracy and robustness in the general case [11].

AMT has several applications in music education (e.g., providing feedback to piano learners who have not yet developed accurate self-diagnostic skills), content-based music search (e.g., searching songs with similar bassline, melody or harmony), musicological analysis of non-notated music (e.g., Jazz improvisations and most non-Western music), and music enjoyment (e.g., visualizing the music content).

While AMT was initially formulated as a method to convert musical sounds into common music notation [13], most AMT systems so far have opted for lower-level representations [11], such as representations expressed in absolute time and frequency; these representations are called *parametric representations* [14]. Very few systems have attempted to estimate higher-level musical information, such as beats or pattern repetitions, directly from audio [15, 16]. Higher level musical information can also be estimated from an intermediate representation [17, 18]. The two main issues limiting the effectiveness of existing methods in this regard are the low transcription accuracy and the difficulty of incorporating musical knowledge in the systems. My goal is to design an end-to-end music transcription system for piano performances, i.e., a system that analyzes a recording of a piano performance and outputs a transcription of the performance in music notation. In my preliminary research, I have worked on two separate sub-problems: a parametric transcription system, from audio to a MIDI piano roll, and a music notation system, from a MIDI piano roll to music notation. For my research I combine the two sub-problems into a complete, end-to-end system.

A core problem of music transcription, and parametric transcription in particular, is figuring out *which* notes are played and *when* they are played in a piece of music. This is also called *note-level transcription* [19]. A note produced by a pitched musical instrument has five basic attributes: pitch, onset, offset, timbre and dynamic. Pitch

is a perceptual attribute but it can be reliably related to the fundamental frequency (F0) of a harmonic or quasi-harmonic sound [20]. Onset refers to the beginning time of a note, in which the amplitude of that note instance increases from zero to an audible level. This increase is very sharp for percussive pitched instruments such as piano. Offset refers to the ending time of a note, i.e., when the waveform of the note vanishes. Compared to pitch and onset, offset is often ambiguous [19]. Timbre is the quality of a sound that allows listeners to distinguish two sounds of the same pitch and loudness [20]. Dynamic refers to the player's control over the loudness of the sound; e.g., a piano player can strike a key with different forces, causing notes to be soft or loud. The dynamic can also change the timbre of a note; e.g., on a piano, notes played *forte* have a richer spectral content than notes played *piano* [21].

Pitch estimation in monophonic music is considered a solved problem [22]. In contrast, polyphonic pitch estimation is much more challenging because of the complex interaction (e.g., the overlapping harmonics) of multiple simultaneous notes. To properly identify all the concurrent pitches, the partials of the mixture must be separated and grouped into clusters belonging to different notes. Most multi-pitch analysis methods operate in the frequency domain with a time-frequency magnitude representation [11]. This approach has two fundamental limitations: it introduces the time-frequency resolution trade-off due to the Gabor limit [23], and it discards the phase, which contains useful cues for the harmonic fusing of partials [20]. Current state-of-the-art results are slightly above 70% in F-measure [19], which is too low for practical purposes, as evaluated by MIREX 2016 on orchestral pieces with up to 5 instruments and piano pieces [24]. While there is no definite standard on the acceptable reliability for commercial applications, it is generally believed that a commercial application should work satisfactorily at least 95% of the times, possibly over 99% of the times. Speech recognition, for example, is now widely used by consumers world-

wide thanks to robust implementations such as Apple Siri, Microsoft Cortana, and Amazon Alexa. Current state-of-the-art speech recognition systems have an error rate below 7% [25]. While this value cannot be directly compared to F-measure, it is clear that speech recognition is a much more mature and robust technology, which is now widely accepted among the general population.

The aim of this work is to propose and develop a complete automatic music transcription system with music notation output, with high accuracy and robustness to common household acoustical conditions, such as moderate background noise and reverberation. The main focus of the thesis will be on transcribing music played on a piano in a specific acoustical context, that is, a specific instrument, recorded with a fixed microphone in a specific room. While this may appear as a severe limitation of the system, these constraints allow the system to achieve greater accuracy than state-of-the-art transcription systems. Also, there are several use cases in which the proposed system can be useful, for instance in a domestic setting, where the position of an instrument and the acoustical environment is relatively static. Moreover, the initial training of the system only requires less than 3 minutes, so it is not a practical limitations in many circumstances.

1.2 Thesis Structure

Chapter 2 presents an introduction to music and audio analysis. Afterwards the problem of automatic music transcription is defined, along with its main challenges.

Chapter 3 presents the proposed method for pitch and onset detection. It begins with a presentation of related work. It then illustrates the Convolutional Sparse Coding framework, which is used to perform the transcription task. It presents the

proposed algorithm for automatic music transcription. Finally, it presents the evaluation results.

Chapter 4 extends the method presented in Chapter 3 in order to estimate the length of the notes. It begins with an illustration of the concept of Structured Sparsity. It then presents the Lateral Inhibition constraint used to enforce the Structured Sparsity. It illustrates how to incorporate Lateral Inhibition in the method presented in Chapter 3. Finally, it presents the evaluation results.

Chapter 5 introduces possible methods to relax the context-specific constraint of the method proposed in Chapter 3 and Chapter 4. It introduces the concepts of Dictionary Learning and Structured Dictionary Learning. It then shows preliminary results on the application of Structured Dictionary Learning to the problem of Automatic Music Transcription.

Chapter 6 presents a method to output the transcription into music notation and an objective metric to evaluate the results. It starts with a review of the related work in music notation output. It illustrates the proposed method to automatically convert the result of parametric transcription into music notation. It then presents the evaluation of the proposed method by human evaluators.

Chapter 7 presents an objective metric to predict human evaluations of transcriptions based on a novel error count measure of twelve different musical features.

Chapter 8 concludes the dissertation, summarizing the contributions and providing future perspectives on the work.

1.3 Summary of Contributions

Current AMT systems generally suffer from poor accuracy and the lack of music notation output. My research aims at overcoming those two main limitations.

The principal contributions of this thesis are:

- Chapter 3: a Convolutional Sparse Coding model for automatic music transcription, which achieves high accuracy in estimating the pitch and the timing of notes.
- Chapter 3: a sparse peak-picking algorithm to minimize false positives in the transcription.
- Chapter 5: a Lateral Inhibition constraint to promote Structured Sparsity in order to estimating note length during automatic music transcription.
- Chapter 6: a method to automatically output music notation from a parametric transcription. This is the first method of this kind based on a probabilistic model of music analysis.
- Chapter 7: an objective metric to calculate the differences between a transcription and the ground truth score. This is the first proposed metric of this kind.
- Chapter 7: a method to predict human evaluations of a transcription based on error counts.

Chapter 2

Computational Music Analysis and Music Information Retrieval

In the history of mankind, music has always been pervasive among all known cultures. We have traces of musical instruments dated from the Paleolithic, about 40,000 years ago. Ethnomusicologists and anthropologists have found musical practices among all the aboriginal cultures still surviving today around the world. Music is also a profitable industry in many countries, so it is not surprising that a considerable amount of research has been devoted to the analysis of audio and musical signals.

Music can be experienced in different ways: by listening, by performing, or by analyzing it. Music analysis can also be performed from different vantage points. For example, a music performance can be analyzed by listening to it live or as a recording, in order to judge the quality of the music, the performance or both. A music score can be perused manually in order to understand its structure, style, and uncover interesting aspects of the composition. Computational Music Analysis (CMA) is the field researching ways to analyze music through algorithms and computer programs, either without human intervention or with some human interaction. CMA focuses on the same tasks as Music Theory, such as analysis of musical forms and composer classification. Music Information Retrieval is a closely related field and aims at retrieving

information from music. The focus of MIR is broader than CMA, as it includes tasks such as source separation, which can extract the music played by individual players from a song, and music fingerprinting, which allows computers to retrieve the title of a song from an audio excerpt.

This chapter provides an introduction to the background concepts in audio, music, mathematics, and signal processing necessary to understand the remainder of the dissertation.

2.1 Background and Terminology

This section introduces the basic terminology related to audio and music.

2.1.1 Audio and Music Signals

Sound can be analyzed both from a physical and a perceptual perspective. Physically, sound is a vibration propagated as pressure waves through air or another elastic transmission medium, gaseous, solid or liquid. Perceptually, sound is the auditory sensation caused by the reception of such waves, in humans and animals, through the vibration of the ear drums and transmitted to the primary auditory cortex through the auditory system [26].

An audio signal is a representation of sound, typically used for recording, transmitting, and reproducing sound. Audio signals can be processed with analog devices, digital devices or a combination of the two.

Music can be a very contentious word to define precisely. Listeners, performers, and scholars can have very different definitions of music, and often they might have strong opinions on what is music and what is just sound or noise. In the English language, and, in general, in most contemporary Western cultures, music is a

metaphor for beautiful and desirable sounds with some recognizable melodic and/or harmonic structure [27]. In this sense, music is produced by one or more instruments (or singers) playing (or singing) together. We will call an instrument or a singer a *source*. Each source can produce musical notes or sounds. Certain sources, such as the human voice or the clarinet, can normally produce a single note at a time. We call these sources *monophonic*. Other sources, such as the piano or the guitar, can produce multiple notes at the same time. We call these sources *polyphonic*. A single monophonic source or a polyphonic source playing a single note at a time can produce monophonic music, also called a *melody*. When multiple notes are played at the same time, by a combination of monophonic and polyphonic sources, we have polyphonic music. This dissertation will deal primarily with polyphonic music from the common practice period, even though the proposed approach can be applied equally successfully to early music, to some modern and contemporary music, and to popular music, including jazz.

2.1.2 Harmonic Sounds

The first systematic study of pleasant sounds is credited to Greek philosopher and mathematician Pythagoras. Pythagoras discovered that the periodic vibrations of strings produced pleasant sounds. He also discovered that multiple strings with equal tension but different lengths produced pleasant combinations if the relative lengths of the strings were simple integer ratios, e.g., 2:1, 3:2, 4:3, ...

The theory of vibrating strings shows that a string can produce sinusoidal vibrations at different frequencies, depending on the length, the density, and the tension

of the string, according to the formula:

$$f_n = \frac{n}{2L}\sqrt{\frac{T}{\mu}}, \tag{2.1}$$

where T is the tension, μ is the linear density, L is the length, and n is an integer number [28]. From the formula, we can see that a string with given length, tension, and density can vibrate only at frequencies that are integer multiples of the lowest frequency $f_1 = \frac{1}{2L}\sqrt{\frac{T}{\mu}}$, which is also called the *fundamental frequency* or *F0*. A wave produced by the sinusoidal vibration at a certain frequency f_n is called *partial*, *mode*, or *simple tone*. In practice, depending on how the vibration is induced (typically by plucking or striking the string), multiple modes can be excited, and the resulting vibration is the linear superposition of multiple partials, called a *complex tone*. However, the different partials are fused together in the auditory system into a single discernible *note* [20]. This fusing mechanism is still not fully understood, and it works similarly for quasi-harmonic sounds, that is, sounds that are composed of partials whose frequencies are only approximately integer multiples of the F0, such as notes produced by pianos, and non-harmonic sounds, such as the sound produced by bells [20].

2.1.3 Pitch and Perceptual Attributes

Pitch is the perceptual attribute of sound that makes it possible to distinguish notes with different fundamental frequencies [20]. In other words, the pitch of a complex tone is perceived as the same as a simple tone with the same fundamental frequency.

The other main perceptual attributes of musical notes are *duration*, *loudness*, and *timbre*. Duration is the length in time of the note. Loudness is its volume. Timbre is

the quality that makes it possible to distinguish notes with the same pitch, loudness, and duration played by different sources or with different articulations.

In music theory, pitch refers to the location of a sound in the tonal scale [27]. In the past, different tunings were used, however the development of the equal temperament tuning system and the standardization on 440 Hz established a widely accepted correspondence between pitches and fundamental frequencies of sounds [27].

2.1.4 Tonality

Tonality, in general, is the systematic arrangements of pitches and relations between them [27]. In common practice music, tonality refers to the orientation of melodies and harmonies around a particular pitch, called the *tonic*, which also gives the name to the *key* of a piece. In monophonic music, melodies often start and end with the tonic, or with the related pitches in the tonic chord, such as the third and the fifth. In polyphonic music, pieces will often end on the tonic chord. The main key of a piece is expressed in the *key signature*.

2.1.5 Rhythm

Rhythm refers to the temporal aspect of music [27], that is the evolution of music in time. Rhythm does not refer just to the length of each individual note, but to a rich and complex hierarchical structure that informs how music is played and understood [8]. Western music (particularly, popular music and classical music of the common practice period) is characterized by a strong hierarchy of accents which repeat according to the *meter* of a piece. At the foundation of the meter is the *tactus*, also called pulse or beat. The tactus is what people typically entrain to when they listen to a piece of music, either unconsciously or when tapping or clapping along.

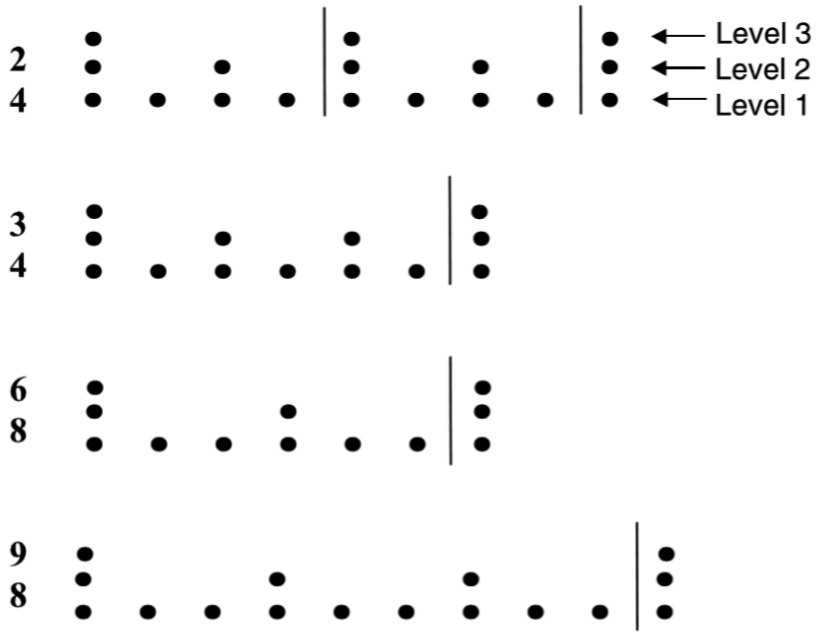


Figure 2.1: Metrical grids for four common time signatures. Level 2 represents the tactus. Level 3 represents the downbeat. Level 1 is the first subdivision of the tactus. From Temperley [8], reproduced with permission from the author.

The meter establishes the relative importance of consecutive beats in a piece of music. Every second or third beat is more important and generally perceived as a stronger accent. The *downbeat* is the strongest and most important beat in a *measure*, which is the smallest group of beats in the meter. The beats can be further subdivided into smaller metrical units, thus creating a hierarchical *metrical grid*. The meter is expressed in the *time signature*. Most common meters in Western music are: 2/4, 4/4, 3/4, 6/8 and 9/8. The metrical grids of four common time signatures in Western music are shown in Fig. 2.1.

2.1.6 Music Representations

Music can be represented in different ways, which can be classified according to their level of abstractions. At the highest level of abstraction, we have the *musical score*, also called *sheet music*, which can be either printed on paper or stored in computer images. This is what composers use to share their compositions with performers. The scores are written in *music notation*, which represents the musical notes to be played, their durations, and the instruments that should play them. Musical scores normally indicate the key and the meter of a piece explicitly. Musical scores usually also contain performance indications, such as dynamics, articulations, and tempo changes. We call the music score a *semantic representation* of music, as it represents music in the form commonly understood by musicians. Music notation is quite complex, and requires a special form of literacy, which trained musicians acquire over a relatively long period of time. Optical Music Recognition (OMR), which studies technologies to allow computers to recognize and interpret music notation, is an active research field that has achieved a certain degree of success but it still faces several open problems.

A *symbolic representation* of music is a semantic representation that can encode all the information contained in a music score and that can be readily accessed by computers. MusicXML [29] is an example of such representations, which is used by open source and commercial programs. The Musical Instrument Digital Interface (MIDI) protocol is another symbolic representation commonly used to represent music in digital format, even though it is not as expressive as MusicXML, that is, it cannot encode all the information that can be represented in MusicXML.

A *parametric representation* is a non-semantic representation in which musical notes are encoded in physical terms, such as seconds for note onset and duration, and Hertz or MIDI numbers for pitch [5]. It can faithfully represent the musical

performance, but normally it does not explicitly encode high-level musical structures, such as key, meter and voicing [18]. A commonly used representation of this kind is an unquantized MIDI pianoroll [30], which is widely used in MIR for different tasks, including automatic music transcription.

An *audio representation* is an audio signal of a music performance. The most common representation of this kind in Music Information Retrieval is the *waveform* of the sound, i.e., the pressure level of sound varying in time, captured through one or more microphones, and digitized through an Analog to Digital Converter (ADC). A waveform can be played back with a Digital to Analog Converter (DAC) connected to an amplifier and loudspeakers, in order to reproduce the original performance. Modern recording and reproduction systems can recreate a musical performance with remarkable precision. However, audio representations do not explicitly encode any semantic music information. A great deal of research in Music Information Retrieval is devoted to extracting semantic information from audio representations.

2.2 Audio Analysis

The most common and natural way to consume music is by listening. It is therefore understandable that a considerable amount of research has been focused on analyzing musical audio signals.

2.2.1 Audio Signal Processing

The analysis and alteration of audio signals through various automatic systems is called audio signal processing. Such processing applies mathematical operations to either analog or digital signals. In the latter case, which is the most common for computer based systems, it is called more specifically Digital Signal Processing (DSP).

2.2.2 Time Domain

Audio signals can be analyzed in the time domain, that is, where audio signals are represented as functions of time, typically in the form of waveforms. This representation is very easy to obtain, as it is the common way digital music is distributed. Most computers, tablets, and smartphones today are also capable of recording audio in digital form.

2.2.3 Frequency Domain

Audio signals can also be analyzed in the frequency domain or, more precisely, in a transform domain with respect to a certain transform.

French mathematician Joseph Fourier is credited for making substantial contributions to the study of trigonometric series, which he used to solve the partial differential equation describing the distribution of heat in a metal plate. Fourier showed that certain functions can be represented as a sum of trigonometric functions, i.e., sinusoids [31]. He also showed that such representation greatly simplifies the solution of certain mathematical problems. His work provided the foundation of modern Fourier analysis, which is central to the study of Linear Time-Invariant (LTI) systems, for which sinusoids are *eigenfunctions* [32].

The frequency domain has been used extensively in DSP as it simplifies operations such as equalization and filtering. The use of the frequency domain in MIR is also physiologically motivated, as the *basilar membrane*, the membrane in the inner ear responsible for converting sound waves into electrical signals to be transmitted to the brain, responds to frequency components in sounds, i.e., different sections of the cochlea vibrate at specific frequencies.

The Discrete Fourier Transform (DFT) [33] is the most common transform used in DSP. Other common transforms are the Constant-Q Transform (CQT), the Mel Frequency Spectrum, and the Wavelet Transform.

2.3 Symbolic Analysis

Audio signals represent music performances very faithfully, but the music semantic is not immediately accessible to be analyzed. For example, in order to determine the key of a particular piece, the pitches played therein may have to be identified first. On the other side, the same information is readily available from the music score of the piece. Symbolic analysis is the analysis of music performed on a symbolic representation or a parametric representation.

Analysis of symbolic representations is what music theorists and musicologists are concerned about. Several tasks can be performed starting from a music score or an equivalent similar symbolic representation, such as identification of the musical form, composer recognition, music structure analysis, and so on.

The analysis of parametric representations of music is relative recent compared to the analysis of audio representations, but it has the potential of substantially improving the accuracy of several MIR tasks, similarly to the breakthroughs achieved by introducing language models in the field of speech recognition. Among the several analysis that can be performed on a parametric representation we can find rhythm and meter determination, key and harmony estimation, and stream separation [8, 18].

2.4 Automatic Music Transcription

Automatic Music Transcription (AMT) is the process of automatically inferring a high-level symbolic representation, such as music notation or piano-roll, from a music performance [11]. It has several applications in music education (e.g., providing feedback to a piano learner), content-based music search (e.g., searching songs with a similar bassline), musicological analysis of non-notated music (e.g., Jazz improvisations and most non-Western music), and music enjoyment (e.g., visualizing the music content).

Music transcription of polyphonic music is a challenging task even for humans. It is related to *ear training*, a required course for professional musicians on identifying pitches, intervals, chords, melodies, rhythms, and instruments of music solely by hearing. AMT for polyphonic music was first proposed in 1977 by Moorer [12], and Piszczalski and Galler [13]. Despite four decades of active research, it is still an open problem and current AMT systems cannot match human performance in either accuracy or robustness [11].

A core problem of music transcription is figuring out *which* notes are played and *when* they are played in a piece of music. This is also called *note-level transcription* [19]. In the literature, the problems of *pitch estimation* and *onset detection* are often addressed separately and then combined to achieve note-level transcription (see Section 3.1). For onset detection, commonly used methods are based on spectral energy changes in successive frames [34]. They do not model the harmonic relation of frequencies that exhibit this change, nor the temporal evolution of partial energy of notes. Therefore, they tend to miss onsets of soft notes in polyphonic pieces and to detect false positives due to local partial amplitude fluctuations caused by overlapping harmonics, reverberation or beats [35].

Pitch estimation in monophonic music is considered a solved problem [22]. In contrast, polyphonic pitch estimation is much more challenging because of the complex interaction (e.g., the overlapping harmonics) of multiple simultaneous notes. To properly identify all the concurrent pitches, the partials of the mixture must be separated and grouped into clusters belonging to different notes. Most multi-pitch analysis methods operate in the frequency domain with a time-frequency magnitude representation [11]. This approach has two fundamental limitations: it introduces the time-frequency resolution trade-off due to the Gabor limit [23], and it discards the phase, which contains useful cues for the harmonic fusing of partials [20]. Current state-of-the-art results are below 70% in F-measure, which is too low for practical purposes, as evaluated by MIREX 2015 on orchestral pieces with up to 5 instruments and piano pieces [36].

My approach to parametric transcription, described in Chapters 3 and 4, overcomes these two limitations by working in the time domain and exploiting phase information.

In the closely related field of speech recognition, in which the input is spoken audio and the output is the transcription of the audio into text, the recent breakthroughs that allowed the creation of widely used applications, such as Apple Siri or Amazon Alexa, are largely due to the introduction of *language models* [37]. A language model is a probability distribution over sequences of words and allows a speech recognition system to select the most probable transcription among similarly sounding possibilities. A similar approach has been attempted for automatic music transcription, even though the results have not been as successful as in the case of speech recognition, with average improvements of a few percentage points. The complexity of the musical language and the sparsity of the data are two major challenges in creating musical

language models [8]. Recent attempts in this direction include Bayesian models [38], neural networks [39, 40], and Linear Dynamic Systems [41].

Chapter 3

Parametric Transcription with Convolutional Sparse Coding

AMT is the process of automatically inferring a high-level symbolic representation, such as music notation or piano-roll, from a music performance [11]. A core problem of music transcription is figuring out *which* notes are played and *when* they are played in a piece of music. This is also called *note-level transcription* [19]. This chapter introduces the core model for parametric transcription used in the proposed end-to-end transcription system. This parametric transcription module will be the foundation for the music notation module, described Chapter 6.

3.1 Related Work

There are in general three approaches to note-level music transcription. *Frame-based* approaches estimate pitches in each individual time frame and then form notes in a post-processing stage. *Onset-based* approaches first detect onsets and then estimate pitches within each inter-onset interval. *Note-based* approaches directly estimate notes including pitches and onsets. The proposed method uses the third approach. In the following, I will review these approaches and discuss their advantages and limitations.

3.1.1 Frame-Based Approach

Frame-level multi-pitch estimation (MPE) is the key component of this approach. The majority of recently proposed MPE methods operate in the frequency domain. One group of methods analyze or classify features extracted from the time-frequency representation of the audio input [11]. Raphael [42] used a Hidden Markov Model (HMM) in which the states represent pitch combinations and the observations are spectral features, such as energy, spectral flux, and mean and variance of each frequency band. Klapuri [43] used an iterative spectral subtraction approach to estimate a predominant pitch and subtract its harmonics from the mixture in each iteration. Yeh et al. [44] jointly estimated pitches based on three physical principles – harmonicity, spectral smoothness and synchronous amplitude evolution. More recently, Dressler [45] used a multi-resolution Short Time Fourier Transform (STFT) in which the magnitude of each bin is weighted by the bin's instantaneous frequency. The pitch estimation is done by detecting peaks in the weighted spectrum and scoring them by harmonicity, spectral smoothness, presence of intermediate peaks and harmonic number. Poliner and Ellis [46] used Support Vector Machines (SVM) to classify the presence of pitches from the audio spectrum. Pertusa and Iñesta [47] identified pitch candidates from spectral analysis of each frame, then selected the best combinations by applying a set of rules based on harmonic amplitudes and spectral smoothness. Saito et al. [48] applied a specmurt analysis by assuming a common harmonic structure of all the pitches in each frame. Finally, methods based on deep neural networks are beginning to appear [49, 50, 51, 39].

Another group of MPE methods are based on statistical frameworks. Goto [52] viewed the mixture spectrum as a probability distribution and modeled it with a mixture of tied-Gaussian mixture models. Duan et al. [53] and Emiya et al. [9]

proposed Maximum-Likelihood (ML) approaches to model spectral peaks and non-peak regions of the spectrum. Peeling and Godsill [54] used non-homogenous Poisson processes to model the number of partials in the spectrum.

A popular group of MPE methods in recent years are based on *spectrogram factorization* techniques, such as Non-negative Matrix Factorization (NMF) [55] or Probabilistic Latent Component Analysis (PLCA) [56]; the two methods are mathematically equivalent when the approximation is measured by Kullback-Leibler (KL) divergence. The first application of spectrogram factorization techniques to AMT was performed by Smaragdis and Brown [57]. Since then, many extensions and improvements have been proposed. Grindlay et al. [58] used the notion of *eigeninstruments* to model spectral templates as a linear combination of basic instrument models. Benetos et al. [59] extended PLCA by incorporating shifting across log-frequency to account for vibrato, i.e., frequency modulation. Abdallah et al. [60] imposed sparsity on the activation weights. O'Hanlon et al. [61, 62] used structured sparsity, also called group sparsity, to enforce harmonicity of the spectral bases. A recent paper by Ewert and Sandler [63] introduces several additional regularizations on the activation weights besides the well established sparsity constraint to achieve higher accuracy in a fixed context setting.

Time domain methods are far less common than frequency domain methods for multi-pitch estimation. Early AMT methods operating in the time domain attempted to simulate the human auditory system with bandpass filters and autocorrelations [64, 65]. More recently, other researchers proposed time-domain probabilistic approaches based on Bayesian models [66, 67, 68]. Bello et al. [69] proposed a hybrid approach exploiting both frequency and time-domain information. More recently, Su and Yang [70] also combined information from spectral (harmonic series) and temporal (subharmonic series) representations.

The closest work in the literature to my approach was proposed by Plumbley et al. [71]. In that paper, the authors proposed and compared two approaches for sparse decomposition of polyphonic music, one in the time domain and the other in the frequency domain. The time domain approach adopted a similar shift-invariant (i.e., convolutional) sparse coding formulation to ours. However, they used an unsupervised approach and a complete transcription system was not demonstrated due to the necessity of manual annotation of atoms. The correct number of individual pitches in the piece was also required in their approach. In addition, the sparse coding was performed in 256-ms long windows using 128-ms long atoms, thus not modeling the temporal evolution of notes. As I will show in Section 3.4.1, this length is not sufficient to achieve good accuracy in transcription. Furthermore, the system was only evaluated on very short music excerpts, possibly because of the high computational requirements at the time the method was designed.

To obtain a note-level transcription from frame-level pitch estimates, a post-processing step, such as a median filter [70] or an HMM [72], is often employed to connect pitch estimates across frames into notes and remove isolated spurious pitches. These operations are performed on each note independently. To consider interactions of simultaneous notes, Duan and Temperley [73] proposed a maximum likelihood sampling approach to refine note-level transcription results.

3.1.2 Onset-Based Approach

In onset-based approaches, a separate onset detection stage is used during the transcription process. This approach is often adopted for transcribing piano music, given the relative prominence of onsets compared to other types of instruments. SONIC, a piano music transcription by Marolt et al., used an onset detection stage to refine the

results of neural network classifiers [74]. Costantini et al. [75] proposed a piano music transcription method with an initial onset detection stage to detect note onsets; a single CQT window of the 64 ms following the note attack is used to estimate the pitches with a multi-class SVM classification. Cogliati and Duan [7] proposed a piano music transcription method with an initial onset detection stage followed by a greedy search algorithm to estimate the pitches between two successive onsets. This method models the entire temporal evolution of piano notes.

3.1.3 Note-Based Approach

Note-based approaches combine the estimation of pitches and onsets (and possibly offsets) into a single framework. While this increases the complexity of the model, it has the benefit of integrating the pitch information and the onset information for both tasks. As an extension to Goto's statistical method [52], Kameoka et al. [76] used so-called harmonic temporal structured clustering to jointly estimate pitches, onsets, offsets and dynamics. Berg-Kirkpatrick et al. [77] combined an NMF-like approach in which each note is modeled by a spectral profile and an activation envelope with a two-state HMM to estimate play and rest states. Ewert et al. [78] modeled each note as a series of states, each state being a log-magnitude frame, and used a greedy algorithm to estimate the activations of the states. In this paper, I propose a note-based approach to simultaneously estimate pitches and onsets within a convolutional sparse coding framework. A preliminary version of this work was published in [6].

3.2 Background

In this section, I present the background material for convolutional sparse coding and its recently proposed efficient algorithm to prepare the reader for its application to automatic music transcription in Section 3.3.

3.2.1 Convolutional Sparse Coding

Sparse coding – the inverse problem of sparse representation of a particular signal – has been approached in several ways. One of the most widely used is Basis Pursuit DeNoising (BPDN) [79]:

$$\arg\min_x \frac{1}{2}\|D\boldsymbol{x} - \boldsymbol{s}\|_2^2 + \lambda\|\boldsymbol{x}\|_1, \tag{3.1}$$

where $\boldsymbol{s}$ is a signal to approximate, D is a dictionary matrix, $\boldsymbol{x}$ is the vector of activations of dictionary elements, and λ is a regularization parameter controlling the sparsity of $\boldsymbol{x}$.

Convolutional Sparse Coding (CSC), also called shift-invariant sparse coding, extends the idea of sparse representation by using convolution instead of multiplication. Replacing the multiplication operator with convolution in Eq. (3.1) we obtain Convolutional Basis Pursuit DeNoising (CBPDN) [80]:

$$\arg\min_{\{\boldsymbol{x}_m\}} \frac{1}{2}\left\|\sum_m \boldsymbol{d}_m * \boldsymbol{x}_m - \boldsymbol{s}\right\|_2^2 + \lambda \sum_m \|\boldsymbol{x}_m\|_1, \tag{3.2}$$

where $\{\boldsymbol{d}_m\}$ is a set of dictionary elements, also called filters; $\{\boldsymbol{x}_m\}$ is a set of activations, also called coefficient maps; and λ controls the sparsity penalty on the

coefficient maps $\boldsymbol{x}_m$. Higher values of λ lead to sparser coefficient maps and lower fidelity approximation to the signal $\mathbf{s}$.

CSC has been widely applied to various image processing problems, including classification, reconstruction, denoising and coding [81]. In the audio domain, $\boldsymbol{s}$ represents the audio waveform for analysis, $\{\boldsymbol{d}_m\}$ represents a set of audio atoms, and $\{\boldsymbol{x}_m\}$ represents their activations. Its applications to audio signals include music representations [82, 71] and audio classification [83]. However, its adoption has been limited by its computational complexity in favor of faster factorization techniques, such as NMF or PLCA.

CSC is computationally very expensive, due to the presence of the convolution operator. A straightforward implementation in the time-domain [80] has a complexity of $\mathcal{O}(M^2N^2L)$, where M is the number of atoms in the dictionary, N is the size of the signal and L is the length of the atoms.

3.2.2 Efficient Convolutional Sparse Coding

An efficient algorithm for CSC has recently been proposed [84, 81]. This algorithm is based on the Alternating Direction Method of Multipliers (ADMM) for convex optimization [85]. The algorithm iterates over updates on three sets of variables. One of these updates is trivial, and the other can be computed in closed form with low computational cost. The additional update consists of a computationally expensive optimization due to the presence of the convolution operator. A natural way to reduce the computational complexity of convolution is to use the Fast Fourier Transform (FFT), as proposed by Bristow et al. [86] with a computational complexity of $\mathcal{O}(M^3N)$. The computational cost of this subproblem has been further reduced to $\mathcal{O}(MN)$ by exploiting the particular structure of the linear systems resulting from

the transformation into the spectral domain [84, 81]. The overall complexity of the resulting algorithm is $\mathcal{O}(MN \log N)$ since it is dominated by the cost of FFTs. The complexity does not depend on the length of the atoms L as the atoms are zero-padded to the length of the signal N.

3.3 Proposed Method

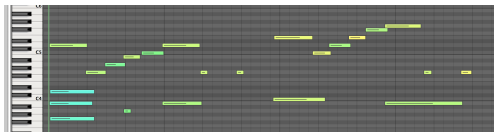
In this section, I describe how I model the piano transcription problem as a convolutional sparse coding problem in the time domain, and how I apply the efficient CSC algorithm [84, 81] to solve the problem.

3.3.1 Transcription Process

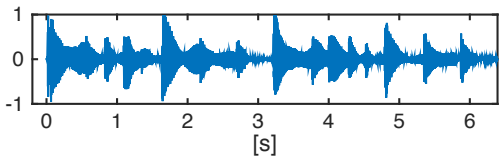
The whole transcription process is illustrated with an example in Fig. 3.1. Taking a monaural, polyphonic piano audio recording $\boldsymbol{s}(t)$ as input (Fig. 3.1(b)), I approximate it with a sum of dictionary elements $\boldsymbol{d}_m(t)$, representing a typical, amplitude-normalized waveform of each individual pitch of the piano, convolved with their activation vectors $\boldsymbol{x}_m(t)$:

$$\boldsymbol{s}(t) \simeq \sum_m \boldsymbol{d}_m(t) * \boldsymbol{x}_m(t). \tag{3.3}$$

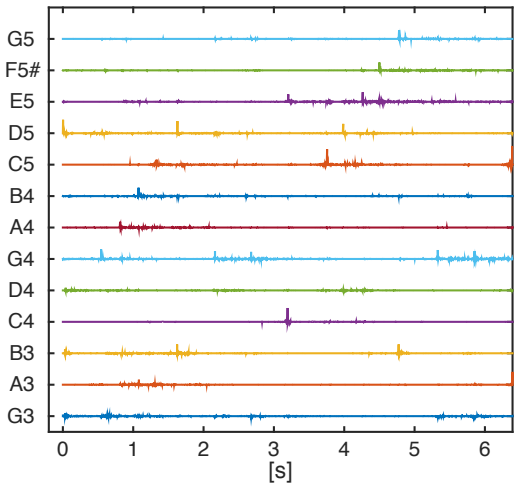
The dictionary elements $\boldsymbol{d}_m(t)$ are pre-set by sampling all the individual notes of a piano – see Section 3.3.1 (Training) – and are fixed during transcription. The activations $\boldsymbol{x}_m(t)$ are estimated using the efficient convolutional sparse coding algorithm [84, 81]. Note that the model is based on an assumption that the waveforms of the same pitch do not vary much with dynamic and duration. This assumption seems to be over-simplified, yet I will show that it is effective in the experiments. I will also discuss its limitations and how to improve the model in Section 3.3.2. Ideally, these



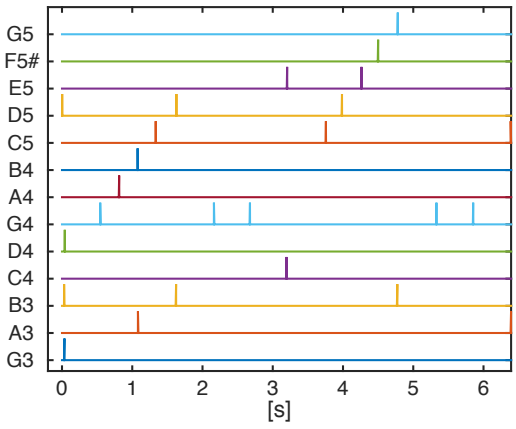
(a) Ground truth piano roll



(b) Waveform



(c) Raw activations



(d) Binarized Activations

Figure 3.1: Piano roll (a), waveform (b), raw activation vectors (c) and the final detected note onsets (d) of Bach's Minuet in G major, BWV Anh 114, from the Notebook for Anna Magdalena Bach.

activation vectors are impulse trains, with each impulse indicating the onset of the corresponding note at a certain time. In practice, the estimated activations contain some noise (Fig. 3.1(c)). After post-processing, however, they look like impulse trains (Fig. 3.1(d)), and recover the underlying ground-truth note-level transcription of the piece (Fig. 3.1(a)). Details of these steps are explained below.

Training

The dictionary elements are pre-learned in a supervised manner by sampling each individual note of a piano at a certain dynamic level, e.g., *forte*, for 1 s. I used a sampling frequency of 11,025 Hz to reduce the computational workload during the experiments. The length was selected by a parameter search (see Section 3.4.1). The choice of the dynamic level is not critical, even though I observed that louder dynamics produce better results than softer dynamics.

Convolutional sparse coding

The activation vectors are estimated from the audio signal using an open source implementation [87] of the efficient convolutional sparse coding algorithm described in Section 3.2.2. The sampling frequency of the audio mixture to be transcribed must match the sampling frequency used for the training stage, so I downsampled the audio mixtures to 11,025 Hz. As described in Section 3.4.1, I investigated the dependency of the performance on the parameter λ on an acoustic piano dataset and selected the best value, $\lambda = 0.005$. I then used the same value for all experiments covering synthetic, anechoic, noisy and reverberant scenarios. I used 500 iterations in my experiments, even though I observed that the algorithm usually converges after approximately 200 iterations.

The result of this step is a set of raw activation vectors, which can be noisy due to the mismatch between the atoms in the dictionary and notes in the audio mixture (see Fig. 3.1 (c)). Note that no non-negativity constraints are applied in the formulation, so the activations can contain negative values. Negative activations can appear in order to correct mismatches in loudness and duration between the dictionary element and the actual note in the sound mixture. However, because the waveform of each note is quite consistent across different instances (see Section 3.3.2), the strongest activations are generally positive.

Post-processing

I perform peak picking by detecting local maxima from the raw activation vectors to infer note onsets. However, because the activations are noisy, multiple closely located peaks are often detected from the activation of one note. To deal with this problem, I only keep the earliest peak within a 50 ms window and discard the others. This enforces local sparsity of each activation vector. I choose 50 ms because it represents a realistic limit on how fast a performer can play the same note repeatedly. In fact, Fig. 3.2 shows the distribution of the time intervals between two consecutive activations of the same note in the ENSTDkCl collection of the MAPS dataset [9]. No interval is shorter than 50 ms.

Binarization

The resulting peaks are also binarized to keep only peaks that are higher than 10% of the highest peak in the entire activation matrix. This step is necessary to reduce ghost notes, i.e., false positives, and to increase the precision of the transcription. The value was chosen by comparing the RMS of each note played *forte* with the RMS of the corresponding note played *piano* in the isolated note collection of MAPS

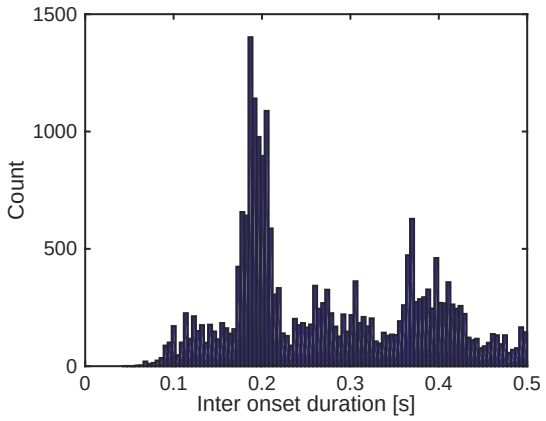


Figure 3.2: Distribution of the time intervals between two consecutive activations of the same note in the ENSTDkCl collection of the MAPS dataset [9]. The distribution has been truncated to 0.5 s for visualization.

(ENSTDkCl set). The average ratio is 6.96, with most of the ratios below 10. This threshold is not tuned and is kept fixed throughout the experiments.

3.3.2 Discussion

The proposed model is based on the assumption that the waveform of a note of the piano is consistent when the note is played at different times at the same dynamic. This assumption is valid, thanks to the mechanism of piano note production [21]. Each piano key is associated with a hammer, one to three strings, and a damper that touches the string(s) by default. When the key is pressed, the hammer strikes the string(s) while the damper is raised from the string(s). The string(s) vibrate freely to produce the note waveform until the damper returns to the string(s), when the key is released. The frequency of the note is determined by the string(s); it is stable and cannot be changed by the performer (e.g., vibrato is impossible). The loudness of the note is determined by the velocity of the hammer strike, which is affected by how hard the key is pressed. The force applied to the key is the only control that

the player has over the onset articulation. Modern pianos generally have three foot pedals: sustain, sostenuto, and soft pedals; some models omit the sostenuto pedal. The sustain pedal is commonly used. When it is pressed, all dampers of all notes are released from all strings, regardless whether a key is pressed or released. Therefore, its usage only affects the offset of a note, if we ignore the sympathetic vibration of strings across notes.

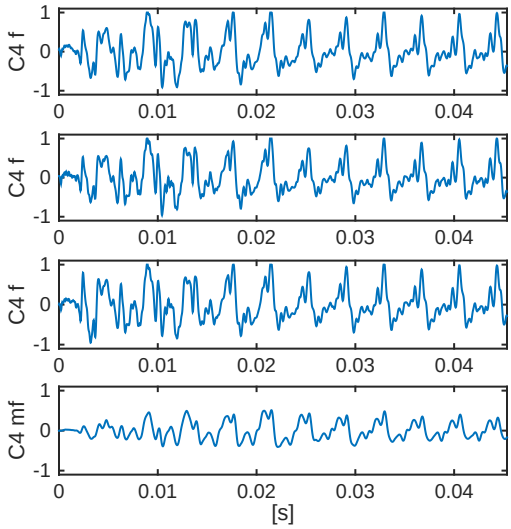


Figure 3.3: Waveforms of four different instances of note C4 played manually on an acoustic piano, three at *forte (f)* and one at *mezzo forte (mf)*. Their waveforms are very similar, after appropriate scaling.

Fig. 3.3 shows the waveforms of four different instances of the C4 note played on an acoustic piano at two dynamic levels. We can see that the three f notes are very similar, even in the transient region of the initial 20 ms. The waveform of the *mf* note is slightly different, but still resembles the other waveforms after applying a global scaling factor. My assumption is that softer dynamics excite fewer modes in the vibration of the strings, resulting in less rich spectral content compared to louder dynamics. However, because the spectral envelope of piano notes is monotonically decreasing, higher partials have less energy compared to lower partials, so softer notes

Table 3.1: Pearson correlation coefficients of a single C4 note played *forte* with the same pitch played at different dynamic levels and with different pitches. Values shown are the maxima in absolute value over all the possible alignments.

Note	Correlation Coefficient
C4 *f* #1	0.989
C4 *f* #2	0.969
C4 *f* #3	0.977
C4 *mf* #1	0.835
C4 *mf* #2	0.851
C4 *mf* #3	0.837
C4 *p* #1	0.608
C4 *p* #2	0.602
C4 *p* #3	0.606
C5 *f* #1	-0.144
C5 *f* #2	-0.146
C5 *f* #3	-0.143
G4 *f* #1	-0.016
G4 *f* #2	-0.019
D4 *f* #1	0.042
D4 *f* #2	-0.042

can still be approximated with notes played at louder dynamics. To prove the last assertion, I compared an instance of a C4 note played *forte* with different instances of the same pitch played at different dynamics and also with different pitches. As we can see from Table 3.1, different instances of the same pitch are highly correlated, regardless of the dynamic, while the correlation between different pitches is low.

As discussed in Section 3.1, Plumbley et al. [71] suggested a model similar to the one proposed here. The efficient CSC algorithm has also been applied to a score-informed source separation problem by Jao et al. in [88]. This method used very short atoms (100 ms), which might be a limiting factor as I prove in Section 3.4, however this limitation may be mitigated, especially for sustaining instruments, by including 4 templates per pitch.

The proposed method can operate online by segmenting the audio input into 2 s windows, and retaining the activations for the first second. The additional second of audio is necessary to avoid the border effects of the circular convolution. Initial experiments show that the performance of the algorithm is unaffected by online processing, with the exception of silent frames. As the binarization step is performed in each window, silent frames introduce spurious activations in the final transcription, so an additional step to detect silent frames, either with a global thresholding or an adaptive filter, is required. Since the computation time of the algorithm is linear in the length of the signal, a shorter signal does not make the algorithm run in real-time in the current CPU-based implementation, which runs in about 5.9 times the length of the signal, but initial experiments with a GPU-based implementation of the CSC algorithm suggest that real-time processing is achievable.

3.4 Experiments

I conduct experiments to answer two questions: (1) How sensitive is the proposed method to key parameters such as the sparsity parameter λ, and the length and loudness of the dictionary elements? (2) How does the proposed method compare with state-of-the-art piano transcription methods in different settings such as anechoic, noisy, and reverberant environments?

For the experiments I used three different datasets: the ENSTDkCl (close-mic acoustic recordings) and the SptkBGCl (synthetic recordings) collections from the MAPS dataset [9], and another synthetic dataset I created specially for this paper, using MIDI files in the ENSTDkCl collection. I will call this dataset ENSTGaSt.

The ENSTDkCl dataset is used to validate the proposed method in a realistic scenario. This collection contains 30 pieces of different styles and genres generated from

high quality MIDI files that were manually edited to achieve realistic and expressive performances. The MIDI files will be used as the ground-truth for the transcription. The pieces were played on a Disklavier, which is an acoustic piano with mechanical actuators that can be controlled via MIDI input, and recorded in a close microphone setting to minimize the effects of reverb. The SptkBGCl dataset uses a virtual piano, the Steinway D from The Black Grand by Sampletekk. For both datasets, MAPS also provides the 88 isolated notes, each 1 s long, played at three different dynamics: *piano* (MIDI velocity 29), *mezzo-forte* (MIDI velocity 57) and *forte* (MIDI velocity 104). I always use the *forte* templates for all the experiments, except for the experiment investigating the effect of the dynamic level of the dictionary atoms. The synthetic dataset is also useful to set a baseline of the performance in an ideal scenario, i.e., absence of noise and reverb.

The ENSTGaSt dataset was created to investigate the dependency of the proposed method on the length of the dictionary elements, as note templates provided in MAPS are only 1 s long. The dataset was also used to verify some alignment issues that I discovered in the ground truth transcriptions of the ENSTDkCl and SptkBGCl collections of MAPS. The ENSTGaSt dataset was created from the same 30 pieces in the ENSTDkCl dataset and re-rendered from the MIDI files using a digital audio workstation (Logic Pro 9) with a sampled virtual piano plug-in (Steinway Concert Grand Piano from the Garritan Personal Orchestra); no reverb was used at any stage. The details of the synthesis model, i.e., the number of different samples per pitch and the scaling of the samples with respect to the MIDI velocity, are not publicly available. To gain some insight on the synthesis model I generated 127 different instances of the same pitch, i.e., C4, one for each value of the valid MIDI velocities, each 1 s long. I then compared the instances with cross correlation and determined that the virtual

instrument uses 4 different samples per pitch, and that the amplitude of each sample is exponentially scaled based on the MIDI velocity.

I use F-measure to evaluate the note-level transcription [19]. It is defined as the harmonic mean of precision and recall, where precision is defined as the percentage of correctly transcribed notes among all transcribed notes, and recall is defined as the percentage of correctly transcribed notes among all ground-truth notes. A note is considered correctly transcribed if its estimated discretized pitch is the same as a reference note in the ground-truth and the estimated onset is within a given tolerance value (e.g., $\pm$ 50 ms) of the reference note. I do not consider offsets in deciding the correctness.

3.4.1 Parameter Dependency

To investigate the dependency of the performance on the parameter λ, I performed a grid search with values of λ logarithmically spaced from 0.4 to 0.0004 on the ENSTDkCl collection in the MAPS dataset [9]. The dictionary elements were 1 s long. The results are shown in Fig. 3.4. As we can observe from Fig. 3.4, the method is not very sensitive to the value of λ. For a wide range of values, from 0.0004 to about 0.03, the average F-measure is always above 80%.

I also investigated the performance of the method with respect to the length of the dictionary elements, using the ENSTGaSt dataset. The average F-measure versus the length over all the pieces is shown in Fig. 3.5. The sparsity parameter λ is fixed at 0.005. The highest F-measure is achieved when the dictionary elements are 1 second long. The MAPS dataset contains pieces of very different styles, from slow pieces with long chords, to virtuoso pieces with fast runs of short notes. My intuition suggested that longer dictionary elements would provide better results for the former,

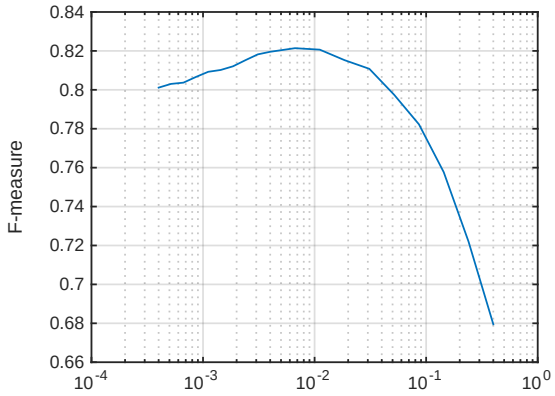


Figure 3.4: Average F-measure on the 30 pieces in the ENSTDkCl collection (close-mic acoustic recordings) of the MAPS dataset for different values of λ, using 1 s long atoms.

and shorter elements would be more appropriate for the latter, but I discovered that longer dictionary elements generally give better results for all the pieces.

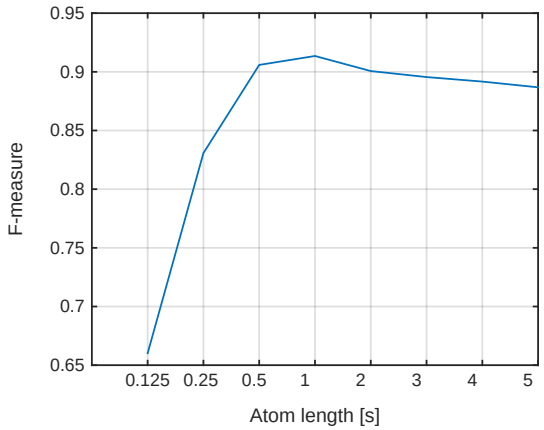


Figure 3.5: Average F-measure on the 30 pieces in the ENSTGaSt dataset versus dictionary atom length, with λ fixed at 0.005.

Finally, I investigated the effect of the dynamic level of the dictionary atoms, using the ENSTDkCl collection. In general I found the proposed method to be very robust to differences in dynamic levels, but I obtained better results when louder dynamics were used during training. A possible explanation can be seen in Fig. 3.6 and Fig. 3.7. In Fig. 3.6 I transcribed a signal consisting of a single C4 note played *piano* with a

dictionary of *forte* notes. The second most active note shows strong negative activations, which do not influence the transcription, as I only consider positive peaks. The negative activations might be due to the partials with greater amplitude contained in the *forte* dictionary element but not present in the *piano* note; i.e., CSC tries to achieve a better reconstruction by subtracting some frequency content. On the other side, in Fig. 3.7 I tested the opposite scenario, a single C4 note reconstructed *forte* with a dictionary of *piano* notes. The second most active note shows both positive and negative activations; positive activations might potentially lead to false positives. In this case, the *forte* note contains some spectral content not present in the *piano* template, so CSC improves the signal reconstruction by adding other note templates. Negative activations also appear when there is a mismatch between the length of a note in the audio signal and the length of the dictionary element. Using multiple templates per pitch, with different dynamics and different lengths, might reduce the occurrence of negative activations at the expense of increased computational time.

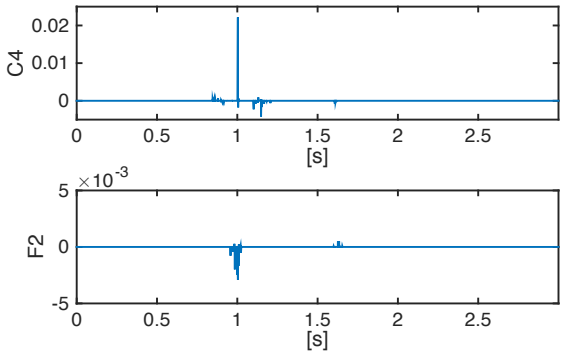


Figure 3.6: Raw activations of the two most active note templates when transcribing a *piano* C4 note with 88 *forte* note templates. Note that the activation of the wrong note template is mostly negative.

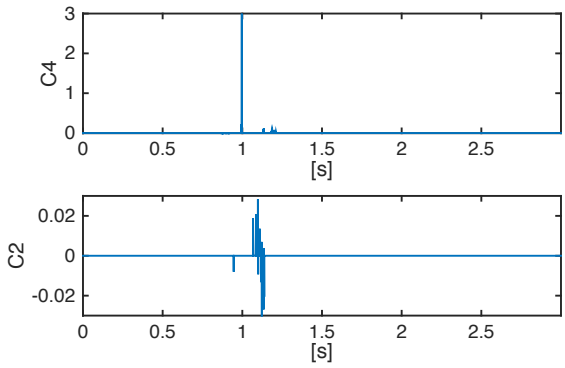


Figure 3.7: Raw activations of the two most active note templates when transcribing a *forte* C4 note with 88 *piano* note templates. Note that the activation of the wrong note template contains a strong positive portion, which may lead to false positives in the final transcription.

3.4.2 Comparison to State of the Art

I compared my method with a state-of-the-art AMT method proposed by Benetos and Dixon [59], which was submitted for evaluation to MIREX 2013 as BW3 [89]. The method will be referred to as BW3-MIREX13. This method is based on probabilistic latent component analysis of a log-spectrogram energy and uses pre-extracted note templates from isolated notes. The templates are also pre-shifted along the log-frequency in order to support vibrato and frequency deviations, which are not an issue for piano music in the considered scenario. The method is frame-based and does not model the temporal evolution of notes. To make a fair comparison, dictionary templates of both BW3-MIREX13 and the proposed method were learned on individual notes of the piano that was used for the test pieces. I used the implementation provided by the author along with the provided parameters, with the only exception of the hop size, which was reduced to 5 ms to test the onset detection accuracy.

Anechoic Settings

For this set of experiments I tested multiple onset tolerance values to show the highest onset precision achieved by the proposed method. The dictionary elements were 1 s long. I used the *forte* templates. The sparsity parameter λ was fixed at 0.005. The results are shown in Figs. 3.8-3.10. From the figures, we can notice that the proposed method outperforms BW3-MIREX13 by at least 20% in median F-measure for onset tolerance of 50 ms and 25 ms (50 ms is the standard onset tolerance used in MIREX [19]). When using dictionary elements played at *piano* dynamic, the median F-measure on the ENSTDkCl collection of the MAPS dataset drops to 70% (onset tolerance set at 50 ms). In the experiment with the ENSTGaSt dataset, shown in Fig. 3.8, the proposed method exhibits consistent accuracy of over 90% regardless of the onset tolerance, while the performance of BW3-MIREX13 degrades quickly as the tolerance decreases under 50 ms. The proposed method maintains a median F-measure of 90% even with an onset tolerance of 5 ms. In the experiment on acoustic piano, both the proposed method and BW3-MIREX13 show a degradation of the performances with small tolerance values of 10 ms and 5 ms.

The degradation of performance on ENSTDkCl and SptkBgCl with small tolerance values, especially the increased support in the distribution of F-measure at 10 ms and 5 ms, drove me to further inspect the algorithm and the ground truth. I noticed that the audio and the ground truth transcription in the MAPS database are in fact not consistently lined up, i.e., different pieces show a different delay between the activation of the note in the MIDI file and the corresponding onset in the audio file. Fig. 3.11 shows two files from the ENSTDkCl collection of MAPS. Fig. 3.11(b) shows a good alignment between the audio and MIDI onsets, but in Fig. 3.11(a) the MIDI

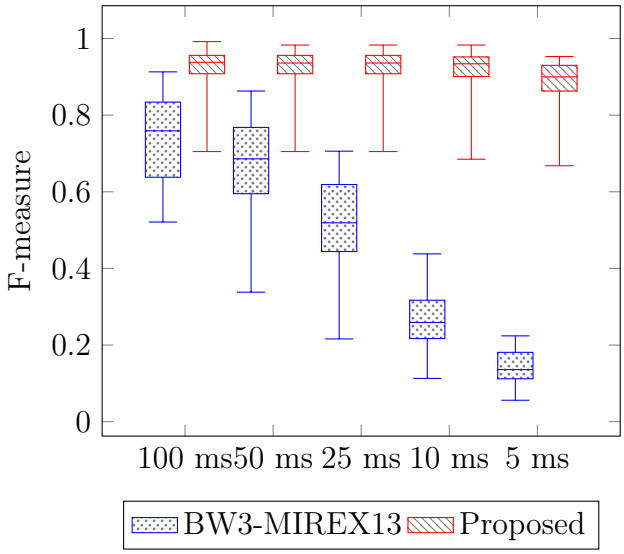


Figure 3.8: F-measure for 30 pieces in the ENSTGaSt dataset (synthetic recordings). Each box contains 30 data points.

onsets occur 15 ms earlier than audio onsets. This inconsistency may be responsible for the poor results with small tolerance values.

To test this hypothesis I re-aligned the ground truth with the audio by picking the mode of the onset differences for the correctly identified notes by the proposed method per piece. With the aligned ground truth, the results on the SptkBgCl dataset for 10 ms of tolerance are similar to the ones on the ENSTGaSt dataset; for 5 ms, the minimum F-measure is increased to 52.7% and the median is increased to 80.2%. On the ENSTDkCl dataset, the proposed method increases the median F-measure by about 15% at 10 ms and 5 ms. It might be argued that the improvement might be due to a systematic timing bias in the proposed method. However, as shown in Fig. 3.8, the transcription performance of the proposed method on the ENSTGaSt dataset does not show clear degradation when the onset tolerance becomes smaller. This suggests that there are some alignment problems between the audio and ground-truth MIDI transcription in the SptkBGCl and ENSTDkCl collections of MAPS. This potential misalignment issue only becomes prominent when evaluating transcription

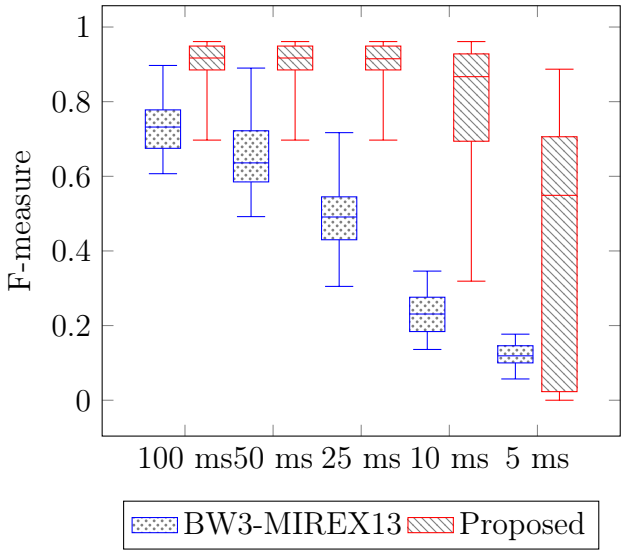


Figure 3.9: F-measure for 30 pieces in the SptkBGCl dataset (synthetic recordings). Each box contains 30 data points.

methods with small onset tolerance values, which are rarely used in the literature. Therefore, I believe that this issue requires additional investigations from the research community before the modified ground-truth can be accepted as the correct one. I thus made the modified ground-truth public, but still use the original non-modified ground truth in all the experiments.

Robustness to Pitch Range and Polyphony

Fig. 3.12 compares the average F-measure achieved by the two methods along the different octaves of a piano keyboard. The figure clearly shows that the results of BW3-MIREX13 depend on the fundamental frequencies of the notes; the results are very poor for the first two octaves, and increase monotonically for higher octaves, except for the highest octave, which is not statistically significant. The proposed method shows a more balanced distribution. This suggests the advantage of my time-domain approach in avoiding the time-frequency resolution trade-off. I do not claim that operating in the time domain automatically overcomes the time-frequency

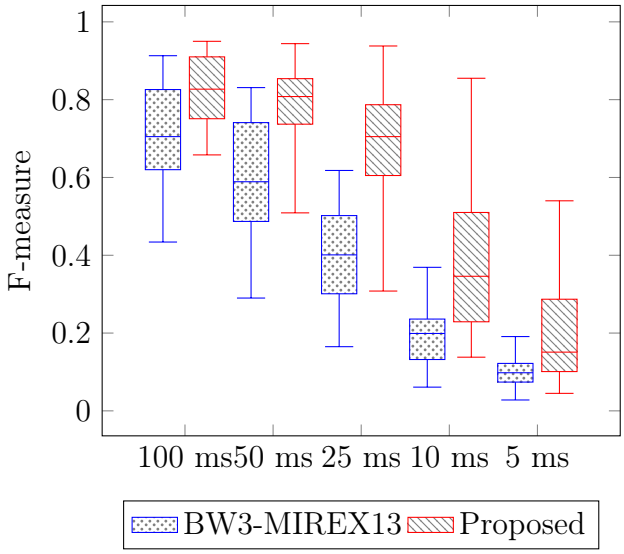
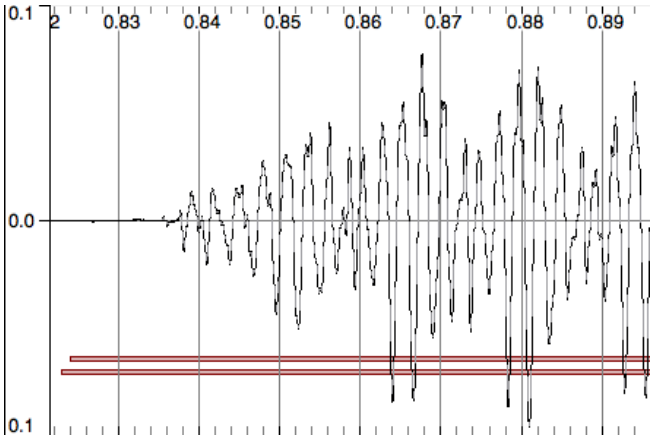


Figure 3.10: F-measure for the 30 pieces in the ENSTDkCl collection (close-mic acoustic recordings) of the MAPS dataset. Each box contains 30 data points.

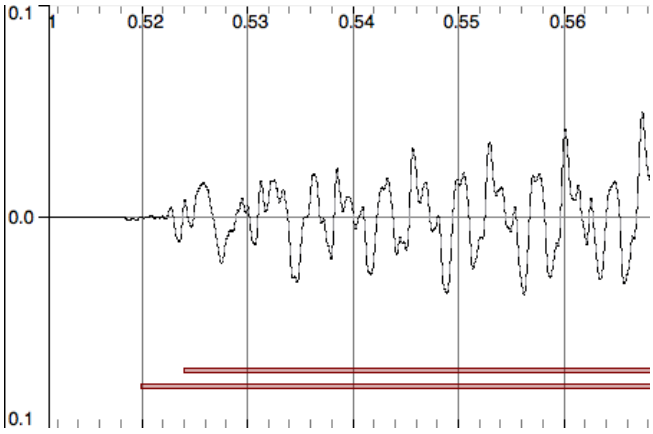
trade-off, and explain the high accuracy of the proposed method as follows. Each dictionary atom contains multiple partials spanning a wide spectral range, and the relative phase and magnitude of the partials for a given note have low variability across instances of that pitch. This, together with the sparsity penalty, which limits the model complexity, allows for good performance without violating the fundamental time-frequency resolution limitations.

The proposed algorithm is less sensitive to the polyphony of the pieces compared to BW3-MIREX13. For each piece in the ENSTDkCl collection of MAPS I calculated the average polyphony by sampling the number of concurrently sounding notes every 50 ms. The results are shown in Fig. 3.13. BW3-MIREX13 shows a pronounced degradation in performance for denser polyphony, while the proposed method only shows minimal degradation.

Fig. 3.14 shows the results on the individual pieces of the ENSTDkCl collection of MAPS. The proposed method outperforms BW13-MIREX13 for all pieces except for two, for which the two methods achieve the same F-measure – Mozart's *Sonata 333*,



(a) Debussy's Claire de Lune (deb_cla) from ENSTDkCl.



(b) Borodin's Piano Sonata 6 (bor_ps6) from ENSTDkCl.

Figure 3.11: Two pieces from the ENSTDkCl collection in MAPS showing different alignments between audio and ground truth MIDI notes (each red bar represents a note, as in a MIDI pianoroll). The figures show the beginning of the two pieces. The audio files are downmixed to mono for visualization. The time axis is in seconds.

second movement (mz_333_2) and Tchaikovsky's *May - Starlight Nights* (ty_mai) from *The Seasons*. The definite outlier is Schuman's *In Slumberland* (scn15_12), which is the piece with the worst accuracy for both the proposed method and BW13-MIREX13; it is a slow piece with the highest average polyphony in the dataset (see Fig. 3.13). The piece with the second worst score is Tchaikovsky's *May - Starlight Nights* (ty_mai); again a slow piece but with a lower average polyphony. A very

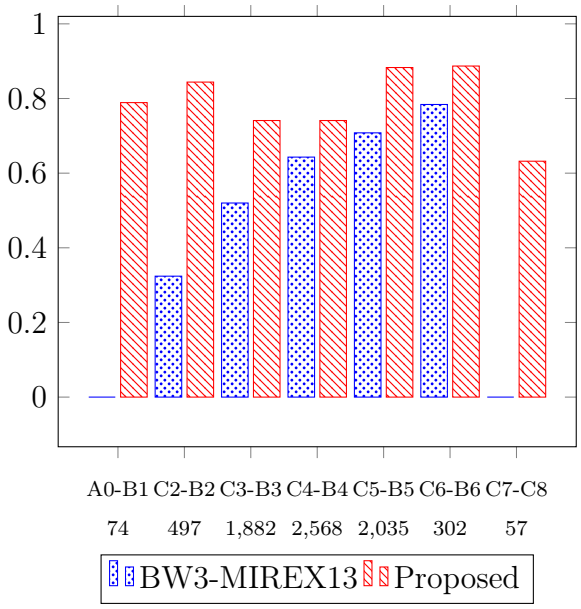


Figure 3.12: Average F-measure per octave for the 30 pieces in the ENSTDkCl collection (close-mic acoustic recordings) of the MAPS dataset. Onset tolerance 50 ms. λ set to 0.005. The letters on the horizontal axis indicate the pitch range, the numbers show the total number of notes in the ground truth for the corresponding octave.

different piece with an F-measure still under 70% is Listz's *Transcendental Étude no. 5* (liz_et5); it is a very fast piece with many short notes and high average polyphony. Further research is needed to investigate why a lower accuracy resulted from these pieces.

Robustness to Noise

In this section, I investigate the robustness of the proposed method to noise, and compare the results with BW3-MIREX13. I used the original noiseless dictionary elements with length of 1 second and tested both white and pink additive noisy versions of the ENSTDkCl collection of MAPS. White and pink noises can represent typical background noises (e.g., air conditioning) in houses or practice rooms. I used the same parameter settings: $\lambda = 0.005$ and 1 s long, *forte* templates. The results are shown in Fig. 3.15 and Fig. 3.16. As we can notice from the plots, the proposed

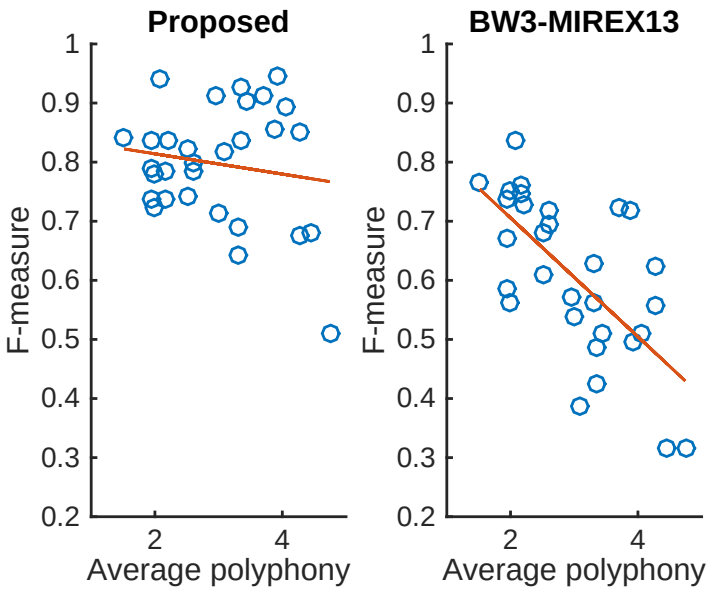


Figure 3.13: F-measure of the 30 pieces in the ENSTDkCl collection (close-mic acoustic recordings) of MAPS versus average instantaneous polyphony. The orange line shows the linear regression of the data points.

method shows great robustness to white noise, even at very low SNRs, always having a definite advantage over BW3-MIREX13. The proposed method consistently outperforms BW3-MIREX13 by about 20% in median F-measure, regardless of the level of noise. The proposed method is also very tolerant to pink noise and outperforms BW3-MIREX13 with low and medium levels of noise, up to an SNR of 5 dB.

Robustness to Reverberation

In the third set of experiments I tested the performance of the proposed method in the presence of reverberation. Reverberation exists in nearly all real-world performing and recording environments, however, few systems have been designed and evaluated in reverberant environments in the literature. Reverberation is not even mentioned in recent surveys [14, 11]. I used a real impulse response of an untreated recording space[1]

[1]WNIU Studio Untreated from the Open AIR Library `http://www.openairlib.net/auralizationdb/content/wniu-studio-untreated`

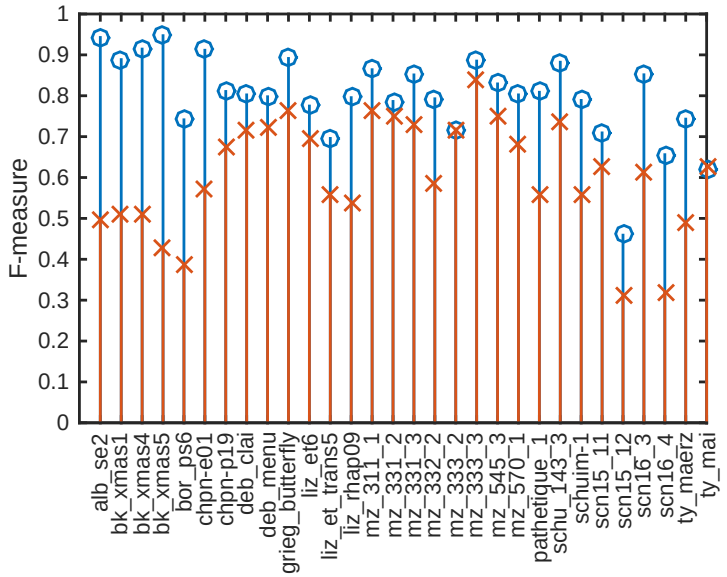


Figure 3.14: Individual F-measures of the 30 pieces in the ENSTDkCl collection (close-mic acoustic recordings) of MAPS. Proposed method in blue circles, BW-MIREX13 in orange crosses.

with an RT60 of about 2.5 s, and convolved it with both the dictionary elements and the audio files. The results are shown in Fig. 3.17. As we can notice, the median F-measure is reduced by about 3% for the proposed method in presence of reverb, showing a high robustness to reverb. The performance of BW3-MIREX13, however, degrades significantly, even though it was trained on the same reverberant piano notes. This further shows the advantage of the proposed method in real acoustic environments.

Sensitivity to Environment Mismatch

To illustrate the sensitivity of the method to the acoustic environment, I generated two synthetic impulse responses with RIR Generator [90], one with RT60 equal to 500 ms and the other with RT60 equal to 250 ms. These two values were picked to simulate an empty concert hall, and the same hall with an audience, whose presence reduces

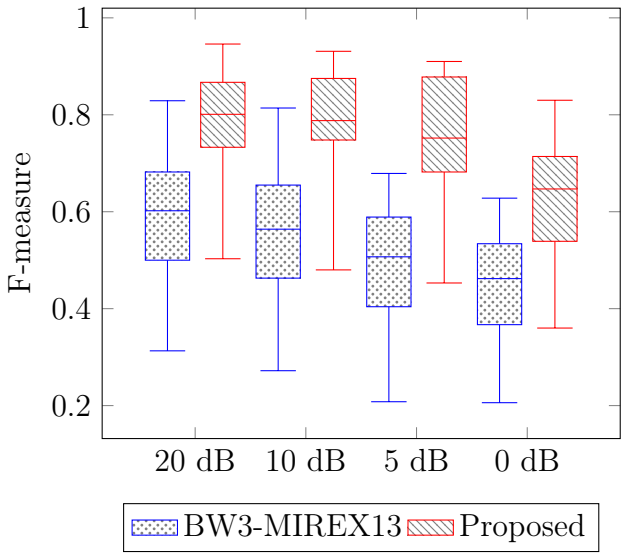


Figure 3.15: F-measure for the 30 pieces in the ENSTDkCl collection (close-mic acoustic recordings) of MAPS with white noise at different SNR levels. Each box contains 30 data points.

the reverberation time by adding absorption to the acoustic environment. I applied the longer impulse response to the dictionary and the shorter one to the 30 pieces in the ENSTDkCl collection of MAPS. The median F-measure for the experiment decreases from 82.7%, as in Fig. 3.10, to 75.2%. It should be noted that this is an extreme scenario, as a typical application would use a close mic setup, reducing the influence of the room acoustics.

Runtime

I ran all the experiments on an iMac equipped with a 3.2 GHz Intel Core i5 processor and 16 GB of memory. The code was implemented in MATLAB. For the 30 pieces in the ENSTDkCl collection of MAPS, the median runtime was 174 s, with a maximum of 186 s. Considering that I transcribed the first 30 s of each piece, the entire process takes about 5.9 times the length of the signal to be transcribed. Initial experiments

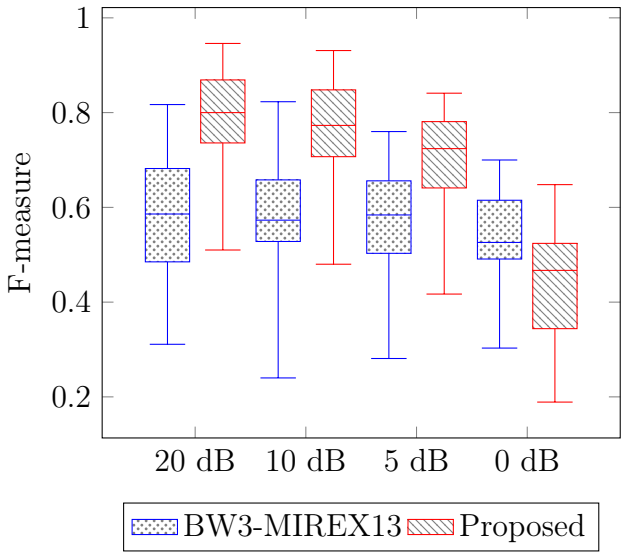


Figure 3.16: F-measure for the 30 pieces in the ENSTDkCl collection (close-mic acoustic recordings) of MAPS with pink noise at different SNR levels. Each box contains 30 data points.

with GPU implementation of the CSC algorithm show an average speedup of 10 times with respect to the CPU implementation.

3.5 Discussion

In this chapter I presented an automatic music transcription algorithm based on convolutional sparse coding in the time-domain. The proposed algorithm consistently outperforms a state-of-the-art algorithm trained in the same scenario in all synthetic, anechoic, noisy, and reverberant settings, except for the case of pink noise at 0 dB SNR. The proposed method achieves high transcription accuracy and time precision in a variety of different scenarios, and is highly robust to moderate amounts of noise. It is also highly insensitive to reverb, as long as the training session is performed in the same environment used for recording the audio to be transcribed. Furthermore,

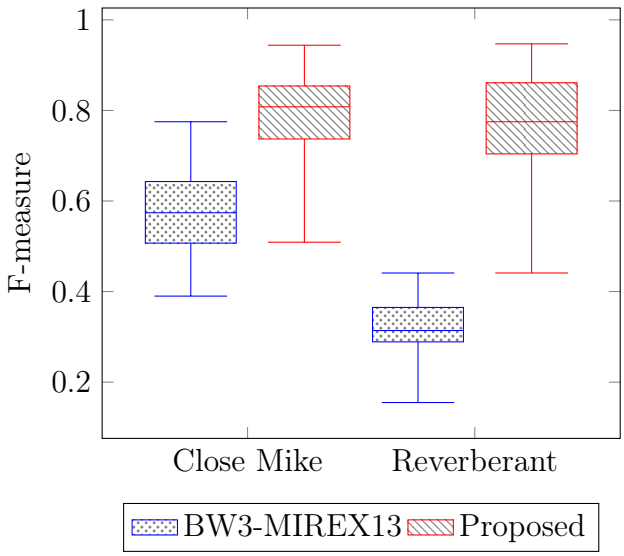


Figure 3.17: F-measure for the 30 pieces in the ENSTDkCl collection (close-mic acoustic recordings) of MAPS with reverb. Each box contains 30 data points.

a limited generalization to a different room acoustic has also been shown in the experiments.

While in this specific context the proposed method is clearly superior to the state-of-the-art algorithm used for comparison (BW3-MIREX13 [59]), it must be noted that this method cannot, at the moment, generalize to different contexts. In particular, it cannot transcribe performances played on different pianos not used for the training. Preliminary experiments with transcribing the ENSTDkCl dataset using the dictionary from the SptkBGCl dataset show a dramatic drop in precision resulting in an average F-measure of 16.9%; average recall remains relatively high at 64.7%. BW3-MIREX13 and, typically, other spectral domain-based methods are capable of being trained on multiple instruments and generalize to different instruments of the same kind. Nonetheless, the proposed context-dependent approach is useful in many realistic scenarios, considering that pianos are usually fixed in homes or studios. Moreover, the training procedure is simple and fast, in case the context changes. More research

is needed to adapt the dictionary to different pianos, and I will describe some preliminary results in this direction in Chapter 5.

Estimating note lengths and dynamics is essential to achieve music notation transcription, however, the proposed method cannot estimate note offsets or dynamics, even though the amplitude of the raw activations (before binarization) is proportional to the loudness of the estimated notes. A dictionary containing notes of different lengths and different dynamics can be used in order to estimate those two additional parameters, even though group sparsity penalties [91, 92] need to be introduced in order to avoid concurrent activations of multiple templates for the same pitch. This approach will be detailed in Chapter 4.

Chapter 4

Structured Sparsity for Estimating Note Lengths

Most existing research in Automatic Music Transcription has focused on pitch and onset detection, while considerably less attention has been devoted to offset detection [11]. However, for many applications, especially those requiring music notation transcription [5], relatively accurate note length estimation is essential, as we will see in Section 6.2.

One of the limitations of the transcription system described in the previous chapter is the inability to estimate the length of the transcribed notes. In this chapter I will describe an extension to the proposed system to estimate the note length by using a dictionary containing multiple atoms with different lengths per pitch, thus creating *pitch groups* of atoms corresponding to the same pitch. When using multiple atoms per pitch, we need to avoid concurrent activations of multiple atoms in the same pitch group. In order to achieve this result I propose to use structured sparsity, more precisely to impose a lateral inhibition [92] regularization term on the activation coefficients of atoms in the same pitch group, in addition to the ℓ_1-regularization on all atoms. The lateral inhibition regularization prevents concurrent activation of multiple atoms in the same pitch group within a temporal neighborhood. We can call this property *within-group sparsity*.

4.1 Structured Sparsity

Standard sparsity assumes a representation that has only a few non-zero coefficients, but makes no additional assumptions on how these non-zero coefficients are distributed within the coefficient vector or matrix. Structured sparsity, in contrast, is based on the assumption that there is some sort of identifiable structure to the distribution of these coefficients. This structure can take many forms, the most common being group sparsity and joint sparsity [91]. The former requires the assignment of dictionary atoms to distinct groups, and assumes that only a few groups are active, but does not require sparse activations within each group. The latter is defined within a Multiple Measurement Vector context [93], and assumes that the representations of different signal vectors share the same or similar pattern of activations. Both of these types of structure can be promoted by the use of the $\ell_{2,1}$ norm [91].

Structured sparsity has previously been applied to AMT. For example, in an NMF framework, a dictionary with multiple atoms per pitch can be learned, in which each atom in the same group represents a different frame of a long note of a particular pitch. Group sparsity can be introduced to promote multiple atoms in the same group to be activated contiguously, i.e., one after the other. An example of such structured sparsity was introduced by O'Hanlon *et al.* [61], who used a modified Non-Negative Basis-Pursuit greedy approach. Another example of group sparsity in a NMF framework was proposed by O'Hanlon and Plumbley [62] to promote the co-activation of harmonically related narrowband atoms. In this case, each group still represents a single pitch, but each pitch is sliced harmonically, not temporally as in the previous method.

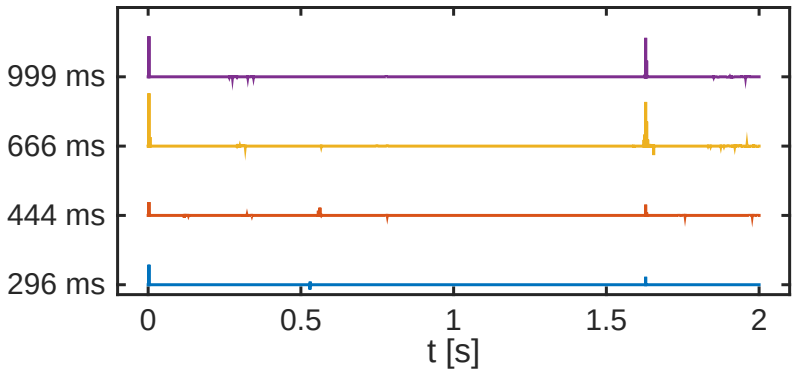
I am interested in limiting the number of concurrently active atoms inside each group, as each atom represents a full note. I call this property *within-group sparsity*.

However, this property alone is not sufficient to achieve a clean activation matrix and, thus, a good transcription. In order to obtain a good transcription, global sparsity on the activations must also be promoted.

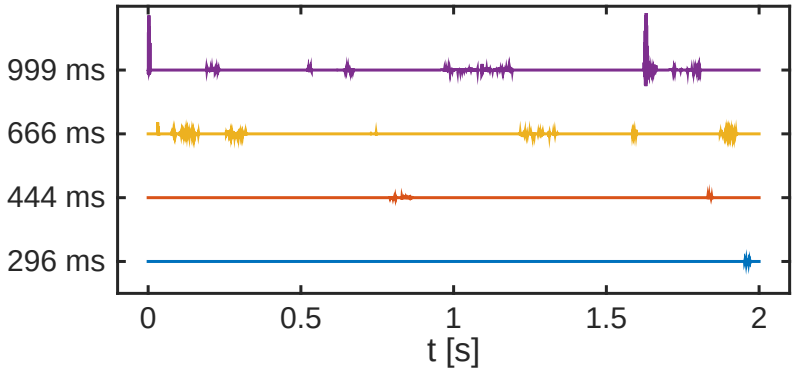
4.2 Proposed Method

The key idea of the proposed method is to jointly estimate pitch, onset and duration of notes by using a dictionary containing multiple atoms with different length for each pitch in the convolutional sparse coding (CSC) framework of (3.2). To create the dictionary I truncate the 1-second long template trained as in CDW-15 [2] to different lengths. This approach is easier and faster than sampling the same pitch played with different durations. However, expanding the dictionary does not lead to satisfying results, as multiple templates in the same pitch group are activated concurrently, as we can see in Fig. 4.1a at the beginning of the signal and slightly after $t = 1.5$ s. The reason is that the ℓ_1 regularization in (3.2) only promotes sparse activations of all templates across all times, but does not distinguish activations of templates in the same pitch group from activations of templates in different pitch groups; moreover, it does not distinguish activations that are temporally close from activations that are temporally apart. While it is possible for a player to play different notes in a rapid pace, it is unlikely to play the same note repeatedly too quickly [2]. Therefore, I need a regularization term that distinguishes these activations and penalizes close activations of templates in the same pitch group.

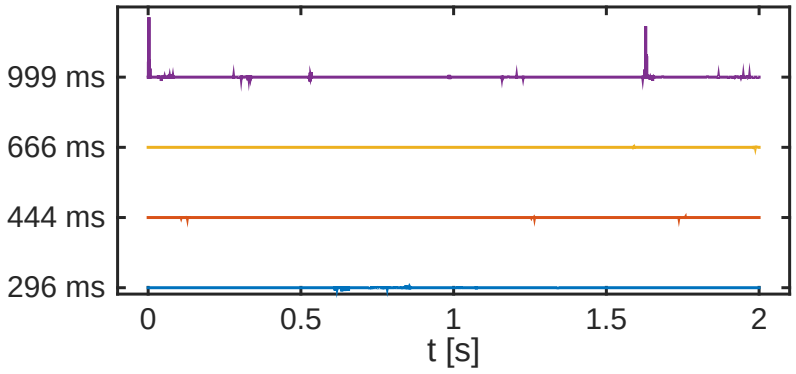
I propose to use a lateral inhibition [92] regularization term on the activations of templates in the same pitch group within a temporal window. The cost of activating



(a) ℓ_1 regularization only



(b) Lateral inhibition regularization only



(c) Combined lateral inhibition and ℓ_1 regularizations

Figure 4.1: Activations of the atoms for pitch D5 for the opening of Bach's Minuet in G. D5 should be activated twice, at $t = 0$ and $t \simeq 1.6$ s. The numbers on the vertical axis indicate the length of each template.

atom m at time t is given by

$$\Gamma(\{\mathbf{x}_m\}) = |\boldsymbol{x}_m(t)| \left[\left(\sum_{\substack{n \in G(m) \\ |t-\tau|<T}} |\boldsymbol{x}_n(\tau)| \right) - |\boldsymbol{x}_m(t)| \right], \tag{4.1}$$

where $G(m)$ is the pitch group to which atom m belongs, and T is the length of the temporal window of inhibition. The activation of atom m at time t will inhibit the activation of all the other atoms in the same pitch group within the temporal window around t. The term $|\boldsymbol{x}_m(t)|$ needs to be subtracted from the summation to avoid self-inhibition.

The full regularization term is the summation of all the costs over all atoms and all time instants, multiplied by a constant, μ. The objective function becomes

$$\arg\min_{\{\boldsymbol{x}_m\}} \frac{1}{2} \left\| \sum_m \boldsymbol{d}_m * \boldsymbol{x}_m - \boldsymbol{s} \right\|_2^2 + \mu \sum_m \Gamma(\{\mathbf{x}_m\}). \tag{4.2}$$

As we can see in Fig. 4.1b, this objective function minimizes the concurrent activations of atoms in the same pitch group and inside the inhibition time window (50 ms), but the activations are not globally sparse over time. Moreover, not shown in the figures, the activations of other groups are also non-sparse. Global sparsity is a key component of CDW-15, and has been successfully applied to AMT for a long time [11]. In order to promote global sparsity on all activations of all templates, I added a a global ℓ_1 norm to the basic lateral inhibition model in (4.9). The objective function with both the global ℓ_1-norm regularization and lateral inhibition regularization is

$$\arg\min_{\{\boldsymbol{x}_m\}} \frac{1}{2} \left\| \sum_m \boldsymbol{d}_m * \boldsymbol{x}_m - \boldsymbol{s} \right\|_2^2 + \lambda \sum_m \|\boldsymbol{x}_m\|_1 + \mu \sum_m \Gamma(\{\mathbf{x}_m\}). \tag{4.3}$$

Using this regularization, as we can see from Fig. 4.1c, the activation vectors are now sparser and less noisy, and also globally sparse, as I will show in the experimental section.

4.3 Algorithm

The simplest form of lateral inhibition structured sparse coding problem [92] is

$$\frac{1}{2}\|D\mathbf{x}-\mathbf{s}\|_2^2+|\mathbf{x}|^T\Omega|\mathbf{x}| \quad , \tag{4.4}$$

where D is a dictionary matrix, and Ω is a matrix encoding the pattern of desired mutual inhibitions. As was pointed out in [92], if the entries of Ω are non-negative, we can define $\mathbf{w}=|\mathbf{x}|^T\Omega$, and write (4.4) as a weighted Basis Pursuit DeNoising (BPDN) problem

$$\frac{1}{2}\|D\mathbf{x}-\mathbf{s}\|_2^2+\|\mathbf{w}\odot\mathbf{x}\|_1 \quad , \tag{4.5}$$

where $\odot$ is the Hadamard product, allowing the problem to be tackled by modifying a standard algorithm for the BPDN problem to include iteratively updating the weight vector $\mathbf{w}$, which depends on the solution variable $\mathbf{x}$. Szlam *et al.* reported [92] that good performance was obtained with a Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) algorithm. They also proposed a convolutional form of (4.4), but applied it to a sufficiently small $\mathbf{s}$ to make it feasible to retain an explicit weighting matrix Ω in the formulation.

Our innovation with respect to the algorithm is twofold. First, since I wish to apply the model to a signal $\mathbf{s}$ that is far too large for an explicit weighting matrix Ω to be practical, I have modified the regularization term so that the lateral inhibition is specified by the product of a convolution filter determining the inhibition in time, and

a small matrix that determines the inhibition within and between groups of dictionary atoms. Second, since ADMM (Alternating Direction Method of Multipliers) has been shown to be more effective than FISTA for the Convolutional BPDN (CBPDN) problem [81], I modify the ADMM algorithm proposed in [84] to include the necessary iterative reweighting. I found experimentally that good results were obtained by updating the new weight vector $\mathbf{w}$ from the primary variable $\mathbf{x}$ rather than from the auxiliary variable introduced in the variable splitting, and by smoothing this weight vector update by defining it as a convex linear combination of the previous and new values.

The lateral inhibition regularization terms in (4.2) and (4.3) are rewritten in terms of convolution as

$$\Gamma(\{\mathbf{x}_m\}) = \sum_m \sum_n c_{m,n} (|\boldsymbol{x}_n| * \boldsymbol{h})^T |\boldsymbol{x}_m| \;, \tag{4.6}$$

where $\boldsymbol{h}$ is the time inhibition window, which is equal to 1 around the origin within a radius of $T/2$, and $c_{m,n}$ is defined as

$$c_{m,n} = \begin{cases} 1 & \text{if } m \neq n \text{ and } G(m) = G(n), \\ 0 & \text{otherwise.} \end{cases} \tag{4.7}$$

If we define

$$\boldsymbol{\omega}_m^T = \sum_n c_{m,n} \left(|\boldsymbol{x}_n| * \boldsymbol{h}\right)^T \;, \tag{4.8}$$

then (4.2) can be rewritten as

$$\underset{\{\boldsymbol{x}_m\}}{\arg\min} \frac{1}{2} \left\| \sum_m \boldsymbol{d}_m * \boldsymbol{x}_m - \boldsymbol{s} \right\|_2^2 + \mu \sum_m \boldsymbol{\omega}_m^T |\boldsymbol{x}_m| \;, \tag{4.9}$$

which immediately shows that the regularization is a weighted ℓ_1-norm on $\boldsymbol{x}_m$. Similarly, (4.3) can be written as

$$\arg\min_{\{\boldsymbol{x}_m\}} \frac{1}{2} \left\| \sum_m \boldsymbol{d}_m * \boldsymbol{x}_m - \boldsymbol{s} \right\|_2^2 + \lambda \sum_m \|\boldsymbol{x}_m\|_1 + \mu \sum_m \boldsymbol{\omega}_m^T |\boldsymbol{x}_m| \; . \tag{4.10}$$

Finally, the two regularization terms can be combined into a single term as

$$\arg\min_{\{\boldsymbol{x}_m\}} \frac{1}{2} \left\| \sum_m \boldsymbol{d}_m * \boldsymbol{x}_m - \boldsymbol{s} \right\|_2^2 + \sum_m \left(\lambda \mathbf{1} + \mu \boldsymbol{\omega}_m^T \right) |\boldsymbol{x}_m| \; , \tag{4.11}$$

where $\mathbf{1}$ is a row vector comprised of all ones.

The resulting ADMM algorithm[1] is very similar to the efficient ADMM algorithm for the CBPDN problem [81], except for the use of a weighted ℓ_1 norm, which requires a minor modification to the soft-thresholding step [94], and in the need for recomputing the weight vector at every iteration, as described above.

The raw activation vectors thus obtained must be post-processed to detect peaks, which correspond to note onsets. This step is a refinement of the method described in [2], generalized to the extended dictionary. I start by setting all the activations below a global threshold, currently set at 10% of the maximum value across the activation matrix $\boldsymbol{X}$, to 0. Then I determine all the local peaks in each activation vector. Finally I iterate over all the peaks, in order of magnitude starting from the largest one, and I set to 0 all the activations in the same pitch group and inside the inhibition window, currently set at 50 ms.

The complexity of the algorithm is dominated by the calculation of the cost vectors $\boldsymbol{\omega}_m$ and is $\mathcal{O}(M^2 N \log N)$, where M is the number of atoms and N is the length of the signal $\boldsymbol{s}$.

[1] An implementation will be included in a future release of the SPORCO library [87].

4.4 Experiment

I applied the different models described in Section 4.2 to the first 10 s of the 30 pieces in the ENSTDkCl dataset of MAPS [9]. The limit of 10 s was determined by the amount of GPU memory required by the current Matlab implementation of the algorithm, however, a longer piece could be transcribed by segmenting it into 10 s long chunks, as described in my previous paper [2]. I used a value of $\lambda = 0.05$ and $\mu = 0.5$. These values were empirically tuned on a single piece and then fixed for the entire dataset. For each piece I calculated Precision, Recall, and F-measure with both onset-only and onset-offset criteria [19], with the standard MIREX parameters: onset tolerance of 50 ms and offset tolerance of 20% of the correct note length or 50 ms, whichever is longer. The lengths of the different atoms in the dictionary for each pitch were chosen to approximate the distribution of note lengths in MAPS, i.e., higher density for shorter notes around 100 ms and lower density for longer notes; I also spaced the durations exponentially in order to maximize the likelihood of estimating the correct length according to the onset-offset criterion. The durations were: 39 ms, 58 ms, 88 ms, 132 ms, 197 ms, 297 ms, 444 ms, 666 ms, 999 ms. I also calculated the Average Overlap Ratio (AOR) [19]. AOR gives a measure of how much a correctly returned note overlaps with the corresponding ground-truth note. I compared the proposed method with several baseline methods: CDW-15, with note lengths fixed at 100 ms; BW-13, a state-of-the-art frame-based method based on PLCA proposed by Benetos and Weyde [95]; SONIC, a piano music transcription system based on neural-networks [96]; DT-14, a generic music transcription system based on maximum likelihood by Duan and Temperley [73]; and VBB-10, an NMF-based transcription system by Vincent et al. [97]. For all the baseline methods I used the original authors' implementation. BW-13 was also trained in the same context

of the proposed method on the isolated notes in the ENSTDkCl dataset of MAPS. It must be noted that SONIC, VBB-10 and DT-14 cannot be trained in the same context, so the comparison is biased against these methods.

The average results for the entire dataset are shown in Table 4.1. We can observe that almost all variants of the CSC-based methods, except LI (Lateral Inhibition), outperform BW-13, VBB-10 and DT-14 on F-measure for the onset-only criterion; CDW-15 and ℓ_1+LI also outperform SONIC, showing the advantage of the time-domain approach over frequency-domain methods in this setting. Moreover, ℓ_1+LI significantly outperforms both LI and ℓ_1 on F-measure. This supports the analysis that both within-group and global sparsity are needed. From CDW-15 to ℓ_1 F-measure drops significantly for the onset-only criterion but increases slightly for the onset-offset criterion. The only difference between these two methods is that ℓ_1 uses 9 templates per pitch while CDW-15 uses only one template. As noted, multiple templates can be activated simultaneously in ℓ_1 resulting in a lower precision but higher recall, and when onset-offset criterion is used, the improvement on recall dominates the decrease on precision. Similarly, from CDW-15 to LI, precision drops significantly, while recall increases slightly under the onset-only criterion and significantly under the onset-offset criterion. However, the drop of precision is due to the false activation of wrong notes instead of the false activation of multiple templates of the correct note. Finally, when onset-only criterion is used, LI+ℓ_1 slightly under-performs CDW-15 on F-measure, but significantly outperforms CDW-15 on AOR; when onset-offset criterion is used, LI+ℓ_1 falls behind SONIC on F-measure but significantly outperforms CDW-15 on both F-measure and AOR. Overall, the proposed method with both lateral inhibition and global sparsity regularization brings the CSC-based approach to the highest level of performance.

Table 4.1: Average results on the first 10 s of the 30 pieces in the ENSTDkCl dataset of MAPS (higher values are better). Bold font indicates the best value in each column.

	Onset only				Onset-offset			
Method	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	AOR	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	AOR
BW-13	64.1	59.8	61.2	55.8	19.2	18.2	18.5	81.4
SONIC	78.0	72.0	74.5	**58.7**	**28.5**	25.7	**26.9**	83.4
DT-14	55.2	34.7	41.3	51.7	15.3	9.4	11.3	82.0
VBB-10	52.5	75.7	60.9	38.8	11.5	15.0	12.8	63.6
CDW-15	**79.7**	83.5	**80.8**	40.1	17.8	18.0	17.8	68.8
ℓ_1	55.4	**88.7**	65.4	54.6	16.8	25.8	19.5	84.4
LI	42.2	83.7	53.3	55.9	12.5	**27.3**	16.3	**84.8**
ℓ_1 + LI	77.7	79.6	77.5	54.6	22.3	23.0	22.3	84.5

4.5 Discussion

In this chapter I extended the work from the previous chapter on convolutional sparse coding for time-domain piano transcription in a context-dependent setting. The proposed method uses multiple templates with different lengths per pitch to achieve note length estimation. Lateral inhibition regularization is introduced to ensure that at most one template per pitch is activated within an inhibition window. Global sparsity is achieved through ℓ_1 regularization to reduce false activations of wrong notes. Experiments show that the proposed method significantly outperforms my prior work and another state-of-the-art frequency-domain method trained in the same context. Note length estimation is essential for music notation transcription, as we will see in Chapter 6, and this proposed method provides an accurate estimation of pitch, onset, and offset. However, the proposed method is still context-dependent. Chapter 5 will present some possible ways to overcome this limitation along with some preliminary results.

Chapter 5

Tentative Generalization to Different Contexts through Structured Dictionary Learning

In order to generalize the parametric transcription to different pianos, i.e., being able to transcribe a piece of music without necessarily having to train the system with the isolated notes, it is necessary either to adapt the dictionary of atoms to a different context or to learn the isolated notes directly from a mixture. This is similar to NMF and PLCA approaches, which are capable of learning or adapting their dictionary directly from a signal [57, 98, 99].

A possible approach is to apply the dictionary learning extension to CBPDN [81]:

$$\underset{\{\boldsymbol{d}_m\}\{\boldsymbol{x}_m\}}{\arg\min} \frac{1}{2} \left\| \sum_m \boldsymbol{d}_m * \boldsymbol{x}_m - \boldsymbol{s} \right\|_2^2 + \lambda \sum_m \|\boldsymbol{x}_m\|_1 \text{, such that } \|\boldsymbol{d}_m\|_2 = 1 \; \forall m, \tag{5.1}$$

where the constraint on the norms of filters $\boldsymbol{d}_m$ is necessary due to the scaling ambiguity between filters and coefficients.

The standard approach to solve this problem is via alternating minimization with respect to coefficients and dictionary. However, the atoms learned in this way are not necessarily piano notes, so the activation coefficients are not related to the transcription of the music piece in the signal $\boldsymbol{s}$. In order to be able to learn a relevant dictionary of atoms, I need to make sure that the atoms $\boldsymbol{d}_m$ are piano notes with the

correct pitch. One possible way to achieve that goal is to include a constraint on the atoms, so that they resemble piano notes.

This problem, which I call *structured dictionary learning*, is related to the task of dictionary design, which has been researched in the context of sparse coding. The next section will present an overview of the related techniques.

5.1 Related Work

Sparse representations have been applied to a wide range of signal, image and audio processing, including compression, denoising, image analysis, audio source separation, and music transcription [100, 101]. Depending on the application, the dictionary can be learned from several representative signals or designed for a specific purpose [100, 102]. Nguyen et al. [103] compare the representations of non-stationary radar returns obtained by a sinusoidal dictionary to the representations obtained with a chirp dictionary, and show that the chirp dictionary provides a more accurate time-frequency representation of the signals. When designing the dictionary, the atoms can often be expressed in analytical form depending on a number of parameters. However, due to practical limitations, only a finite number of atoms can be used so the parameters in the analytical form must be discretized.

O'Neill and Flandrin model signals as a sparse weighted sum of chirped Gabor functions. They propose a Maximum Likelihood Estimation to find the optimal parameters and a tractable sub-optimal estimator [104]. Yaghoobi et al. formalize the problem of parametric dictionary design as a sampling problem over the parameters of an analytical model and propose a method to determine an approximate solution in an alternating minimization framework in which the solutions to the dictionary update step are projected onto the admissible sets at each iteration [102]. Merlet et

al. [105] propose to update the dictionary by directly estimating the parameters that provide the sparsest representation at each iteration.

5.2 Experiment

For the experiment, I applied a method similar to Yaghoobi et al. [102]. The notes of a piano can be modeled as:

$$d_m(t) = \sum_{n=1}^{N} e_{m,n}(t) p_{m,n}(t), \tag{5.2}$$

where

$$p_{m,n}(t) = \sin\left(2\pi f_{m,n} t + \phi_{m,n}\right) \text{ with } f_{m,n} = n f_{m,0}\sqrt{1 + n^2 B_m}, \tag{5.3}$$

and

$$e_{m,n}(t) = A_{m,n}\tilde{e}_{m,n}(t) \text{ with } \max \tilde{e}_{m,n}(t) = 1, \tag{5.4}$$

where $\tilde{e}_{m,n}(t)$ represent the shape of the envelope of each partial, and $A_{m,n}$ their amplitude.

For a given pitch m, the frequencies of the partials $p_{m,n}$ are determined by the virtual fundamental frequency $f_{m,0}$ and the inharmonicity coefficient B_m [106]. From the theory of piano string coupling [107], the envelope $e_{m,n}$ of each partial can be modeled as linear decay (of log-energy), Fig. 5.1 (a), double decay, Fig. 5.1 (b), or curve decay (beats), Fig. 5.1 (c) [10].

Given the complexity of the model in eq. 5.2 with respect to the envelope of the individual partials, a simpler model was considered first:

$$d_m(t) = e_m(t) \sum_{n=1}^{N} A_{m,n} p_{m,n}(t) \tag{5.5}$$

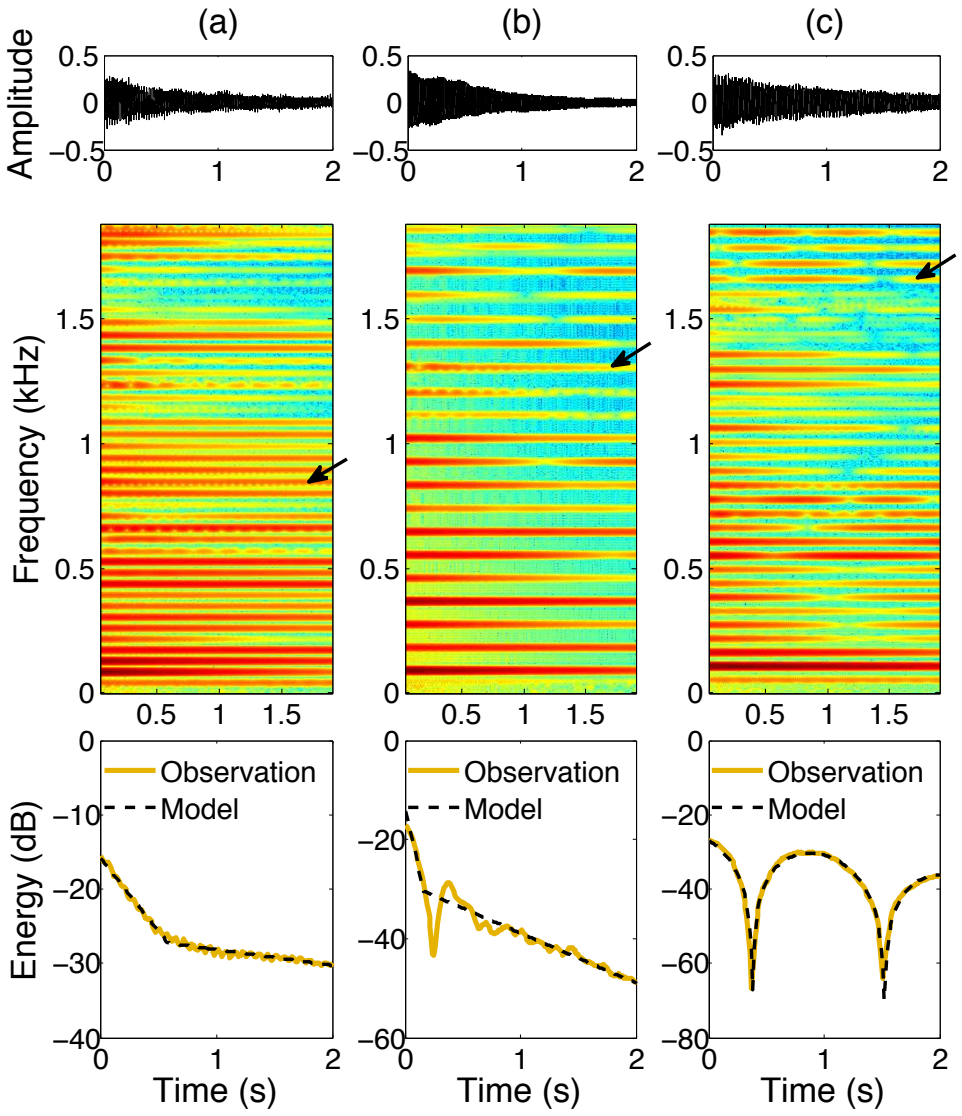


Figure 5.1: Different decay patterns of partials from notes (a) F1 (43.7Hz), (b) G♭2 (92.5Hz) and (c) A1 (55Hz). The top and middle panes show the waveforms and spectrograms, respectively. The bottom panes show the decay of selected partials, which are indicated by the arrows on the spectrograms. From Cheng et al. [10], reproduced with permission of the authors.

where $p_{m,n}$ are the same as in eq. (5.3).

In this simplified model, there is a single envelope governing the entire temporal evolution of a note of a given pitch. The parameters of this model can be learned from

Table 5.1: Results of the transcription of the first ten seconds of Bach's Minuet in G played on a real piano with a fixed dictionary learned from the ENSTDkCl dataset in MAPS and with the dictionary learning framework with the simple model in eq. (5.5).

Method	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
Fixed dictionary	5.3	55.3	9.7
Learned dictionary	6.5	73.7	11.9

the eight different pianos in the MAPS database. In the experiment I determined the acceptable ranges of the following parameters for a given pitch m: $f_{m,n}$, $A_{m,n}$, $\phi_{m,n}$, with $A_{m,n}$ normalized so that $A_{m,1} = 1$, $\forall m$. To model the envelopes e_m, I extracted the envelopes of the eight notes of the same pitch from the eight different pianos in MAPS, then I averaged them together to obtain an average envelope e_m. To test the model, I attempted to learn the atoms from the first ten seconds of Bach's Minuet in G played on a real piano. I applied the dictionary learning extension to CBPDN in eq. (5.1) by constraining the learned atoms to be of the form expressed in eq. (5.5). In order to test the potential of the model, I applied the constraint directly during the optimization process by projecting the learned atoms after each update step to an atom of the form in eq. (5.5). To do so I first applied the DFT to each learned atom, then determined the peaks in the magnitude spectrum closer to the expected harmonic peaks according to the frequency of the fundamental according to equal temperament tuning. From the spectrum I extracted the values of $f_{m,n}$, $A_{m,n}$ and $\phi_{m,n}$ of the frequency bins corresponding to the peaks. If any of the extracted values fell outside the acceptable range determined from MAPS, I pulled the corresponding value to the closest bound of the range. I then fitted the extracted $f_{m,n}$ to the formula for $f_{m,n}$ in eq. (5.3) to determine $f_{0,n}$ and B_m. Finally I reconstructed each atom with eq. (5.5) with the parameters determined at each step and the mean envelopes extracted from MAPS. The results are show in Table 5.1.

While there's a definite improvement in recall, the overall performance of the system is very poor. By listening to the learned atoms, it is clear that eq. (5.5) does not model piano notes accurately enough, so in order to achieve a good reconstruction of the signal, several spurious activations are required.

The model in eq. (5.2) is more promising as initial experiments in reconstructing piano notes from parameters learned from the instruments in MAPS sound more convincing. I also tested applying CBPDN to the original note using the reconstructed atom, and the activation is very close to a delta function. That suggests that eq. (5.2) might be suitable to model atoms for transcription. However, determining the parameters of the model and constraining the learned atoms to the model is non-trivial, and it is left to future work.

Chapter 6

From MIDI to Music Notation

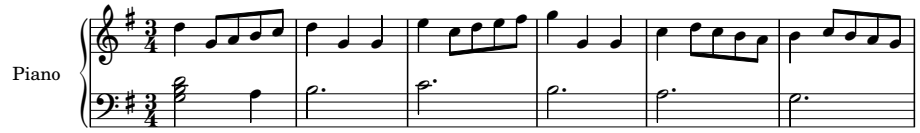
AMT systems can be broadly classified into two categories according to the chosen symbolic representation: parametric transcription and music notation transcription. Parametric transcription systems output a parametric representation of the musical performance, such as an unquantized MIDI pianoroll [30]. This representation is expressed in physical terms, such as seconds for note onset and duration, and hertz or MIDI numbers for pitch [5]. It can faithfully represent the musical performance, but normally it does not explicitly encode high-level musical structures, such as key, meter and voicing [18]. Music notation transcription systems, on the other hand, output a common music notation that human musicians read. This representation is expressed in musically meaningful terms, such as quantized meter for note onset and duration, and spelling distinctions (e.g., A♭ versus G♯) for pitch. Compared to parametric transcription, music notation transcription is generally more desirable for many applications connecting humans and machines, such as computational musicological analysis and music tutoring systems. The vast majority of existing AMT methods, however, are parametric transcription systems.

AMT was initially formulated as a method to convert musical sounds into common music notation [13]. However, most AMT systems so far have opted for lower level representations [11]; very few systems have attempted to estimate higher level musi-

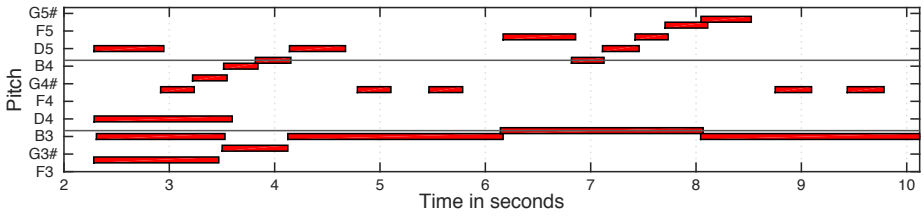
cal information, such as beats or pattern repetitions, directly from the audio [15, 16]. A notable recent contribution by Correa Carvalho and Smaragdis [108] attempts to produce music scores directly from audio signals through a neural network. Higher level musical information can also be estimated from an intermediate representation [17, 18]. For my research I opt for the latter approach; this allows the conversion of MIDI to notation, and eventually (in combination with an audio-to-MIDI conversion system, such as the one described in Chapters 3 and 4) could generate notation from audio as well.

A MIDI file can represent a piano performance very accurately; in fact, the only variables involved are note onset, offset, velocity and pedal activation. Moreover, MIDI representations of piano performances can be recorded from a MIDI keyboard, or from a piano with key sensors. The MIDI standard is capable of encoding high-level musical information, such as key and time signatures, into MIDI files, but this information is not typically included in recorded performances, unless the performer manually inserts it. Furthermore, recorded MIDI performances are typically unquantized, as performers continuously change the speed of playing to obtain a more expressive performance, and may play certain notes slightly earlier or later than they should be played to highlight certain musical lines.

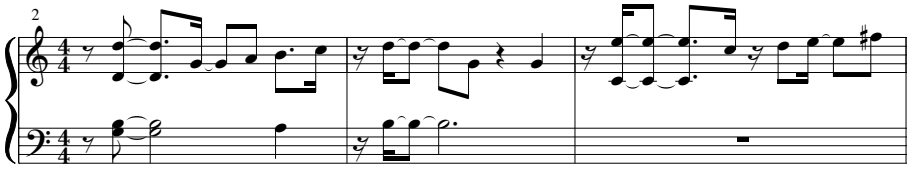
The process of producing a correct full music notation from an unquantized and un-annotated MIDI file is non-trivial and, to the best of my knowledge, no system capable of producing full music notation has been implemented and documented in academic research papers thus far. Without a proper estimation of the meter and the harmony, the results are very poor – see Fig. 6.1 (c). The task can be divided into two main sub-tasks: musical structure analysis and note placement on the score. For the first sub-task, the MIDI file must be analyzed to estimate the key signature and the correct note spelling, as well as the beats and the correct time signature. For



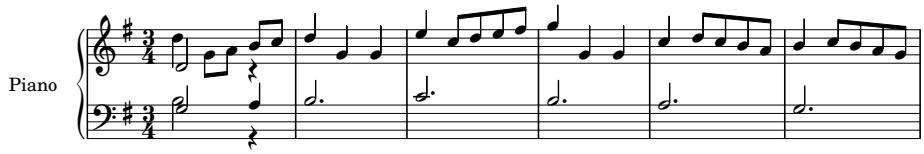
(a) Ground Truth



(b) Unquantized MIDI piano roll



(c) GarageBand



(d) Proposed method

Figure 6.1: Transcription of a performance of the Minuet in G from Bach's Notebook for Anna Magdalena Bach. (a) shows the original score (b) shows the unquantized pianoroll of a MIDI performance. (c) shows the output from GarageBand, which does not perform any analysis on the MIDI file. (d) shows the output of the proposed method after estimating the correct meter, key signature, beats and streams. The music excerpts are of different lengths for better formatting.

the second sub-task, once the notes have been correctly spelled and quantized to the correct meter, they must be properly positioned on the staff. Piano music is normally notated on two staves. The higher staff is usually notated in treble clef, and contains the notes generally played by the right hand. The lower staff is usually notated in bass clef, and contains the notes generally played by the left hand. Notes should be placed

on staves to simplify the reading of the score, e.g., notes should be well spaced and typographical elements should not clash with each other. The placement of the notes and other typographical elements also convey musical meanings, e.g., notes pertaining to the same voice should have the stems pointing in the same direction and beaming should follow the rhythm of the musical passage. Finally, concurrent notes played by a single hand as chords should share the same stem. Exceptions to these basic rules are not uncommon, typically to simplify the reading by a performer, e.g., if a passage requires both hands to play in the higher range of the piano keyboard, both staves may be notated in the treble clef to avoid too many ledger lines and too many notes on the same staff.

In this chapter I present a novel method to fully notate a piano performance recorded as an unquantized and un-annotated MIDI file, in which only the note pitches (MIDI number), onsets and offsets are considered. The initial analysis of the piece is done through a probabilistic model proposed by Temperley to jointly estimate meter, harmony and streams [18]. The engraving of the score is done through the free software LilyPond [1]. The evaluation dataset and the Python code are available on my web site[2].

6.1 Related Work

There are several free and commercial programs, such as Finale, Sibelius and MuseScore, that can import MIDI files and translate them into full music notation, but they typically require user intervention to inform the process to a certain degree. For instance, Finale requires the user to manually select the time signature, while it can infer the key signature from the file itself. Certain sequencers and Digital Audio

[1] `http://lilypond.org`
[2] `http://www.ece.rochester.edu/~acogliat/`

Workstations, such as GarageBand and Logic Pro, have various functions to facilitate the import of MIDI files; for example, Logic Pro has a function to align the time track to the beats in the MIDI files, but requires the user to input the time signature and estimate the initial tempo of the piece.

Among the programs used for the evaluation of the proposed method, MuseScore [3] has the most advanced MIDI file import feature. MuseScore has a specific option to import human performances, and is capable of estimating the meter and the key signature. During the experiment, MuseScore showed a sophisticated capability to position different voices on the piano staves, which resulted in high scores from the evaluators, especially in terms of overall voicing and staff placement. Unfortunately, details on how all these steps are performed are not documented in the website and have not been published in research papers.

The task of identifying musical structures from a MIDI performance has been extensively researched, especially in the past two decades. Cambouropoulos [17] describes the key components necessary to convert a MIDI performance into musical notation: identification of elementary musical objects (i.e., chords, arpeggiated chords, and trills), beat identification and tracking, time quantization and pitch spelling. However, the article does not describe how to render a musical score from the modules presented. Takeda et al. [109] describe a Hidden Markov Model for the automatic transcription of monophonic MIDI performances. In his PhD thesis, Cemgil [110] presents a Bayesian framework for music transcription, identifying some issues related to automatic music typesetting (i.e., the automatic rendering of a musical score from a symbolic representation), in particular, tempo quantization, and chord and melody identification. Karydis et al. [111] proposes a perceptually motivated model for voice separation capable of grouping polyphonic groups of notes, such as chords or

[3] `https://musescore.com`

other forms of accompaniment figures, into a perceptual stream. A more recent paper by Grohganz et al. [112] introduces the concepts of score-informed MIDI file (S-MIDI), in which musical tempo and beats are properly represented, and performed MIDI file (P-MIDI), which records a performance in absolute time. The paper also presents a procedure to approximate an S-MIDI file from a P-MIDI file – that is, to detect the beats and the meter implied in the P-MIDI file, starting from a tempogram then analyzing the beat inconsistency with a salience function based on autocorrelation.

Musical structures can also be inferred directly from audio. Ochiai et al. [15] propose a model for the joint estimation of note pitches, onsets, offsets and beats based on Non-negative Matrix Factorization constrained with a rhythmic structure modeled with a Gaussian mixture model. Collins et al. [16] propose a model for multiple F0 estimation, beat tracking, quantization, and pattern discovery. The pitches are estimated with a neural network. A Hidden Markov Model (HMM) is separately used for beat tracking. The results are then combined to quantize the notes. Note spelling is performed by estimating the key of the piece and assigning to MIDI notes the most probable pitch class given the key.

6.2 Proposed Method

The proposed method takes an unquantized and un-annotated MIDI file as input. The following subsections explain each step in the proposed method. The entire process is illustrated in Fig. 6.2. An example of the output is shown in Fig. 6.1 (d).

6.2.1 Fix Spurious Overlapping Notes

The first step is to fix spurious overlapping notes. Piano players do not play notes with the correct length all the time. As we can see from Fig. 6.1 (b), certain notes

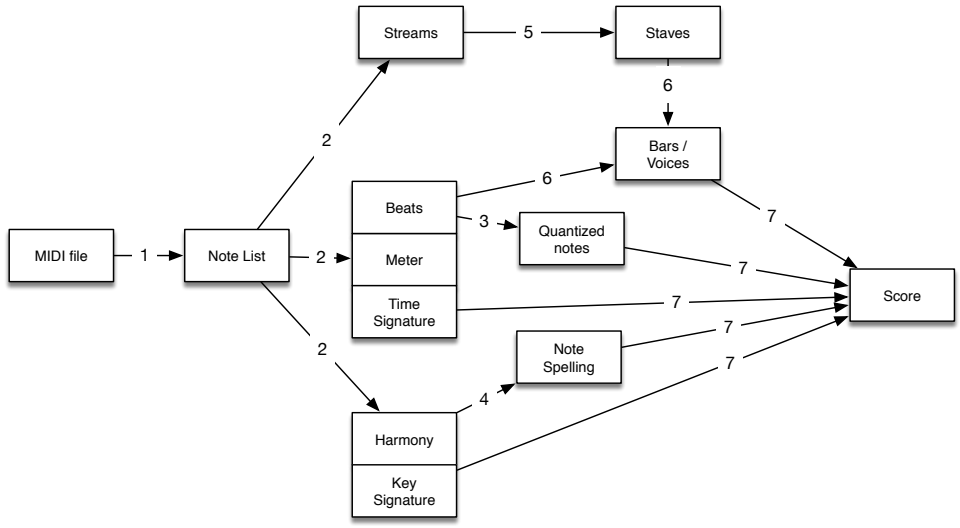
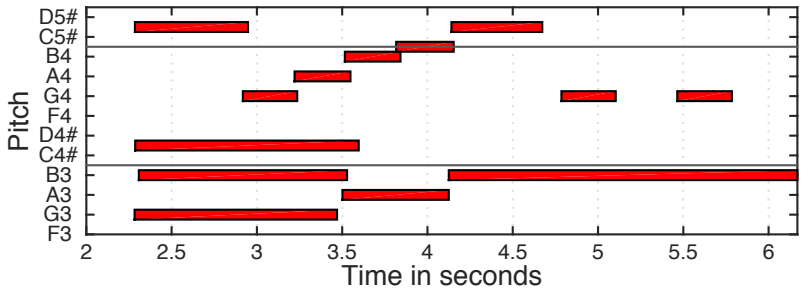


Figure 6.2: Illustration of the proposed method. The arrows indicate dependencies between entities. The numbers refer to the steps (subsection numbers) in Section 6.2.

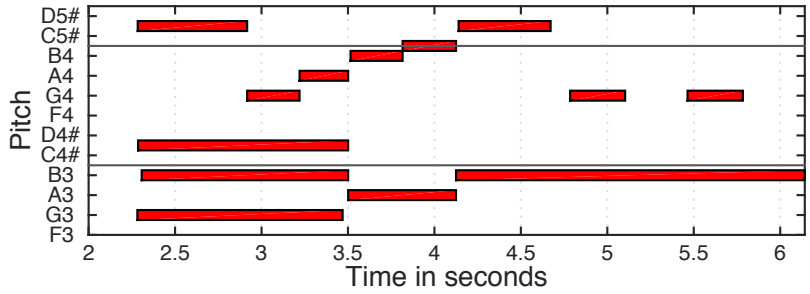
are played shorter than they should be, resulting in gaps between notes, while other notes are played longer than they should be, resulting in overlapping notes. Gaps between notes in the same melodic line might result in extra rests in the score, while overlapping notes might result in extra streams being created by the probabilistic model [18] used in the next step, resulting in extra voices in the final score. In particular, the probabilistic model used in this paper always assigns overlapping notes to different streams, so it is critical to remove erroneous overlaps.

To estimate whether the overlap is correct or wrong I consider pairs of overlapping notes separately. For each pair, I calculate one overlapping ratio for each note. The ratio is defined as the length of the overlapping region over the length of the note. The overlap is considered spurious if the sum of the two ratios is below a certain threshold. For the experiment I set a threshold of 30%. The output of the first step is a note list, i.e., a list of note events, each including an onset, a duration (both in milliseconds), and a MIDI note number. An example is shown in Fig. 6.3. Notice the small overlaps in the top figure between the three low notes in the initial chord and

the second bass note, as well as the short overlaps in the scale in the soprano line; these are removed in the second figure. Also notice that correct overlapping notes, such as a melody line moving over the same bass note, are preserved.



(a) Original pianoroll of a MIDI performance



(b) Pianoroll after fixing spurious overlapping notes

Figure 6.3: An example of the step of fixing spurious overlapping notes.

6.2.2 Estimate Meter, Harmony and Streams

In the second step, I apply the probabilistic model [18] to the note list. The probabilistic model estimates the meter, the harmony, and the streams. The meter and harmony are estimated in a single joint process. This process is modeled as an HMM and is based on the concept of tactus-root combination (TRC), a combination of two adjacent tactus beats and a chord root. The probability of a TRC only depends on the previous TRC, and the probability of beats and notes within a TRC only depends on the TRC. The musical intuition behind this is that the "goodness" of a tactus interval

depends only on its relationship to the previous tactus interval (with a preference to minimize changes in length from one interval to the next), the goodness of a root depends only on the previous root (with a preference to maintain the same root if possible, or to move to another root that is a fifth away), and the goodness of a particular pattern of notes within a short time interval depends only on the current root and the placement of beats within that interval (with a preference for note onsets on tactus beats or at plausible points–e.g., roughly halfway–in between them, and a preference for notes that are chord-tones of the root). The process also considers different divisions of the tactus interval (representing simple or compound meter) and placements of strong beats (duple versus triple meter). In the current context, the metrical analysis is useful for the placement of barlines and for rhythmic notation; the harmonic analysis is useful for pitch spelling, and also influences the metrical analysis, since there is a preference for strong beats at changes of harmony (this is the reason for estimating the meter and harmony jointly). The stream segregation problem is solved with dynamic programming by grouping notes into streams such that the number of streams, the number and length of rests within streams, and pitch intervals within streams are all minimized [18].

The output of the probabilistic model is a list of beats, notes, and chord roots. Each beat includes an onset in milliseconds, and a level in a metrical hierarchy [8]. The probabilistic model considers the tactus and two subdivisions in the metrical structure; e.g., in a 3/4 meter, the tactus will be the quarter note, the first subdivision will be the 8th note, and the lowest subdivision the 16th note. The metrical structure also indicates the downbeats. Each note has an onset and a duration in milliseconds, a midi note number, and a stream number. The chord roots are quantized to the beats. An example of the output of this stage is shown in Fig. 6.4.

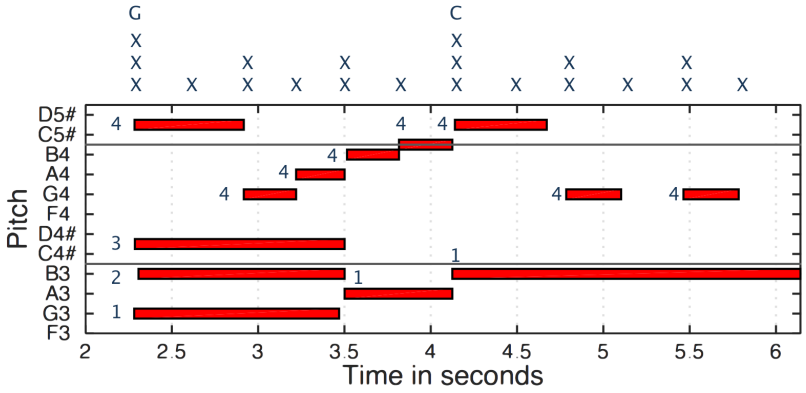


Figure 6.4: Sample output of the probabilistic model for estimating the metrical, harmonic, and stream structures. The Xs above the pianoroll illustrate the meter analysis (only 3 levels displayed). The letters above show the chord root (only roots on the downbeats are shown). The numbers next to the notes indicate the stream.

6.2.3 Quantize Notes

The third step quantizes the note onsets to the closest beat subdivision. The offset of each note is also set to coincide with the onset of the next note in a stream; i.e., gaps within each stream are discarded. This avoids extra rests in the final scores, which could stem from notes played shorter than they should be. See, for instance, the two quarter notes in stream 4 in the second bar of the pianoroll in Fig. 6.4; they were played slightly shorter than 8th notes.

6.2.4 Determine Note Spelling

The correct note spelling is determined from the harmony generated by the probabilistic model and is based on the proximity in the line of fifths (the circle of fifths stretched out into a line) to the chord root. For example, the MIDI note 66 (F♯/G♭) would be spelled F♯ on a root of D, but spelled as G♭ on a root of E♭.

6.2.5 Assign Streams to Staves

The staves of the final score are set to be notated in treble clef for the upper staff and bass clef for the lower staff. Streams are assigned to the staff that accommodates all the notes with the fewest number of ledger lines.

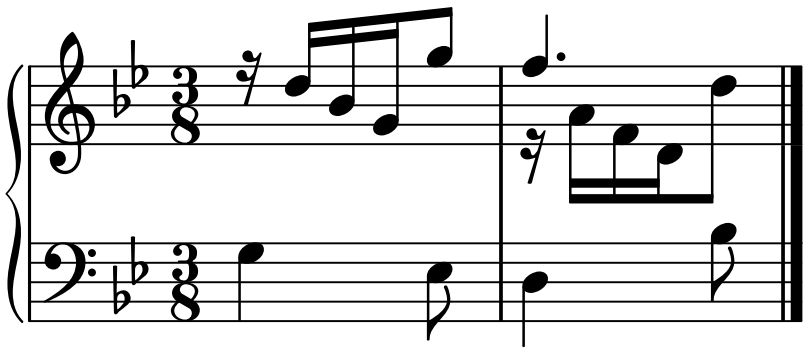


Figure 6.5: First two measures of Bach's Sinfonia in G minor, BWV 797. In the second bar, two streams are assigned to the same staff, so two separate monophonic voices must be created for proper rendering.

6.2.6 Detect Concurrent Voices

Once streams have been assigned to staves, I determine bars and voices. Bars are easily determined by the metrical structure, but note adjustments might be necessary if a note starts in one bar and continues to the next bar. In that case, the note has to be split into two or more tied notes. Concurrent notes in the same bar and staff must be detected and encoded appropriately for the next step. If a staff contains streams that overlap in time, I create monophonic voices consisting of sequences of notes. A sequence is defined as a gapless succession of notes and rests without overlaps. For example, as shown in Fig. 6.5, concurrent streams in measure 1 can be treated as monophonic inputs as they are assigned to separate staves, but in measure 2, two

concurrent streams are assigned to the same staff, so I have to create two monophonic sequences of notes as input for the next step, one containing the F dotted quarter, the other containing the 16th notes and the D 8th note.

6.2.7 Generate the Score

Finally, a Lilypond input file is generated. Lilypond is a free, command-line oriented music engraving program, which takes a text file as input and, thus, is suitable for the automatic generation of music notation. A possible alternative to Lilypond, which was considered during my research, is MusicXML [29]. Lilypond has the advantage of a simpler and more concise syntax. For instance, the music example from [29], which requires 130 lines of MusicXML, only requires 12 lines in Lilypond.

6.3 Evaluation and Discussion

To evaluate the proposed method, I asked five doctoral students in the Music Theory department of the Eastman School of Music, at various stages of advancement in their program, to blindly rate the output of the proposed method, two commercial programs (Finale 2015 [4] and GarageBand 10 [5]) and a free engraving program (MuseScore 2) applied to the Kostka-Payne dataset used to evaluate the probabilistic model [18]. The commercial programs have been chosen due to their popularity: GarageBand is freely available to all Mac users, Finale is one of the two major commercial music notation programs, the other being Sibelius. I also tested the import functionality of Sibelius but the results were very similar to the ones obtained by Finale, so I dropped this dataset to save time during the human evaluation. The dataset comprises 19

[4] `https://www.finalemusic.com`
[5] `https://www.apple.com/mac/garageband/`

music excerpts, all of them piano pieces by well-known composers, for a total of 76 music scores to evaluate. The pieces were performed on a MIDI keyboard by a semi-professional piano player. For each piece I provided the original score, i.e., the ground truth. All the scores had been anonymized, so that the source program name was unknown, and the order of the evaluation was randomized. The evaluators were asked the following questions: 1) Rate the pitch notation with regard to the key signature and the spelling of notes. 2) Rate the rhythmic notation with regard to the time signature, bar lines, and rhythmic values. 3) Rate the notation with regard to stems, voicing, and placement of notes on staves. These three questions summarize the most important features that determine the formatting and the readability of a musical score. The three features are also fairly independent of each other.

The ratings were on a scale from 1 to 10 – 10 being the best. I instructed the evaluators to rate the scores to reflect how close each output was to the ground truth. Finally, I told the evaluators that, since each rating may reflect multiple aspects of the notation, it was entirely up to their judgment to decide how to balance them (e.g., the relative importance of time signature, barline placement, and rhythmic values for the second question).

The results are shown in Figs. 6.6, 6.7 and 6.8. The ratings from each evaluator have been normalized (z-scores) by subtracting the mean and dividing by the standard deviation, and the results have been rescaled to the original range by setting their mean to 5 and their standard deviation to 2. The proposed method outperforms all the other methods in the first two ratings – pitch notation and rhythm notation – and ties for the top in median for the third rating – voicing and staff placement. Paired sign tests show that the ratings of the proposed method are significantly better than all the three baselines for the first two aspects, at a significance level of $p = 0.0001$. For the third aspect, the proposed method is superior to Finale and

equivalent to MuseScore at a significance level of $p = 0.0001$, while the comparison with GarageBand is statistically inconclusive.

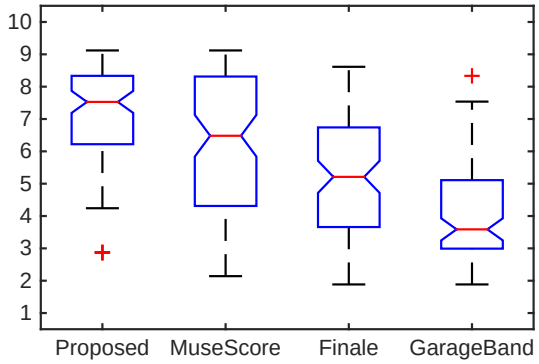


Figure 6.6: Normalized pitch notation ratings. Each box contains 76 scores from each of the 5 evaluators.

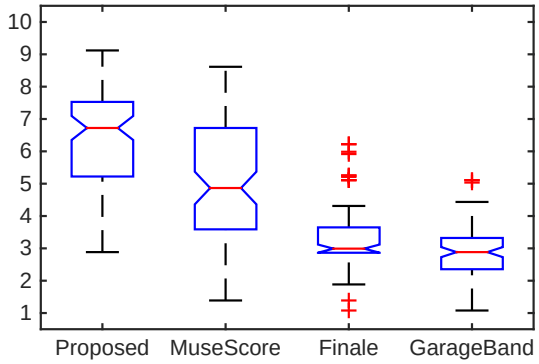


Figure 6.7: Normalized rhythm notation ratings. Each box contains 76 scores from each of the 5 evaluators.

More work is needed in the note placement. One common aspect of music notation that has not been addressed in the proposed method is how to group concurrent notes into chords; I can see how that affects the output in Fig. 6.1 (d). In the downbeat of the first bar, the lowest three notes are not grouped into a chord, as in the ground truth (Fig. 6.1 (a)). This makes the notation less readable, and also introduces an unnecessary rest in the upper staff. A possible solution to this problem consists in

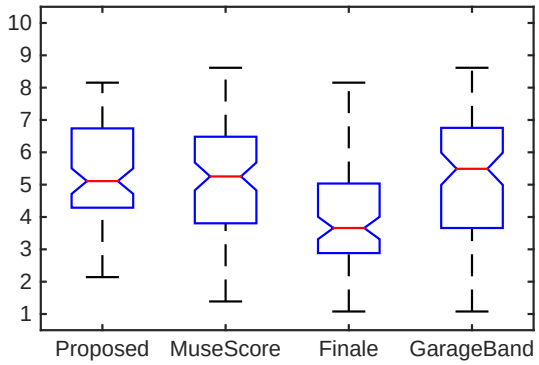


Figure 6.8: Normalized note positioning ratings. Each box contains 76 scores from each of the 5 evaluators.

grouping into chords notes that have the same onset and duration, and that are not too far apart, i.e., so that they could be played by one hand. A possible drawback of this approach is that it may group notes belonging to different voices.

Another limitation of the proposed method is the positioning of concurrent voices in polyphonic passages. Currently, the proposed method relies on the streams detected in step 2 to determine the order in which the voices are positioned in step 6. In polyphonic music, voices can cross so the relative positioning of voices might be appropriate for certain bars but not for others. A possible solution is to introduce another step between 6 and 7 to analyze each single measure and determine whether the relative positions of the voice is optimal or not. These two limitations affect the note positioning, reflected in the scores shown in Fig. 6.8. Finally, the probabilistic model does not always produce the correct results, especially with respect to beats and streams. A more sophisticated model may improve the rhythm notation and the note positioning, reflected in the scores shown in Fig. 6.7 and Fig. 6.8.

Chapter 7

Objective Evaluation of Music Notation Output

An immediate problem arising when building a music notation transcription system is to find an appropriate way to evaluate the transcription accuracy of the system. For the evaluation in Chapter 6, I asked music theorists to evaluate music notation transcriptions along three different musical aspects, i.e., the pitch notation, the rhythm notation, and the note positioning. However, subjective evaluation is time consuming and difficult to scale to provide enough feedback to further improve the transcription system. It would be very helpful to have an objective metric for music notation transcription, just like the standard metric F-measure for parametric transcription [19]. Considering the inherent complexity of music notation, such a metric would need to take into account all of the aspects of the high-level musical structures in the notation. To the best of my knowledge, there is no such metric, and the goal of this chapter is to propose such a metric.

Specifically, in this chapter I propose an edit distance, based on similar metrics used in bioinformatics and linguistics, to compare a music transcription with the ground-truth score. The design of the metric was guided by a data-driven approach, and by simplicity. The metric is calculated in two stages. In the first stage, the two

scores are aligned based on the pitch content; in the second stage, the differences between the two scores are accumulated, taking into account twelve different aspects of music notation: barlines, clefs, key signatures, time signatures, notes, note spelling, note durations, stem directions, groupings, rests, rest duration, and staff assignment. This will serve the same purpose as F-measure in evaluating parametric transcription. To validate the saliency and the usefulness of this metric I also apply a linear regression model to the errors measured by the metric to predict human evaluations of transcriptions.

7.1 Background

Approximate sequence comparison is a typical problem in bioinformatics [113], linguistics, information retrieval, and computational biology [114]. Its purpose is to find similarities and differences between two or more sequences of elements or characters. The sequences are assumed sufficiently similar but potentially corrupted by errors. Possible differences include the presence of different elements, missing elements or extra elements. Several metrics have been proposed to measure the distance between two sequences, including the family of edit metrics [114], and gap-penalizing alignment techniques [113].

A music score in traditional Western notation can be viewed as a sequence of musical characters, such as clefs, time and key signatures, notes and rests, possibly occurring concurrently, such as in simultaneous notes or chords. Transcription errors include alignment errors due to wrong meter estimation or quantization, extra or missing notes and rests, note and rest duration errors, wrong note spelling, wrong staff assignment, wrong note grouping and beaming, and wrong stem direction. All of these errors contribute to a various degree to the quality of the resulting transcription.

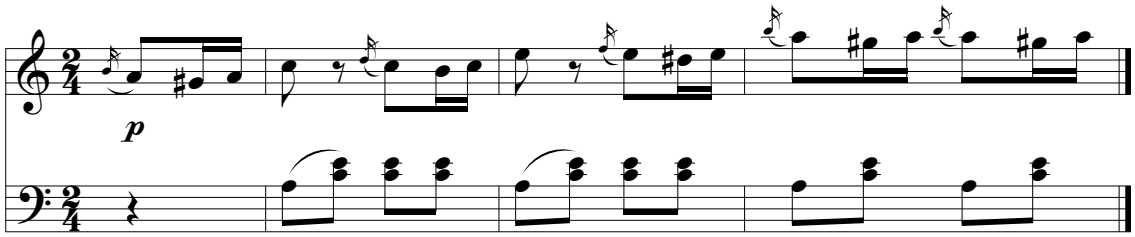
However, the impact of each error and error category has not, to the best of my knowledge, been researched.

As an example, Fig. 7.1 shows two transcriptions of the same piece. Both transcriptions contain similar errors, i.e., wrong meter detection, but the transcription in Fig. 7.1c is arguably worse than that in Fig. 7.1b. A similar problem can be observed with the standard F-measure typically used to evaluate parametric transcriptions [19]; while the metric is objective and widely used, the impact of different errors on the perceptual quality of a transcription has not been researched. Intuitively, certain errors, such as extra notes outside of the harmony, should be perceptually more objectionable than others, such as octave errors. This is the reason for both proposing an objective metric and correlating the metric with human evaluations of transcriptions.

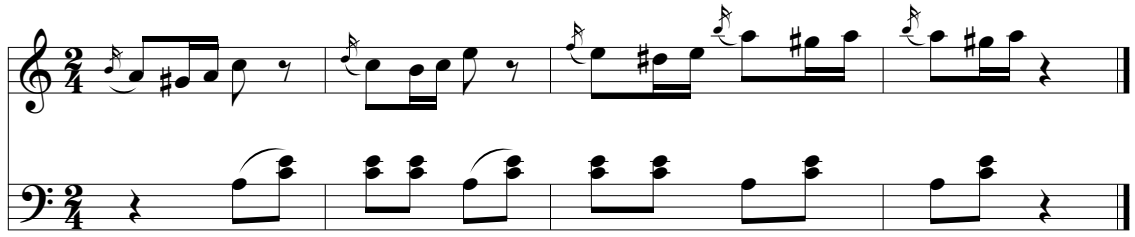
7.2 Proposed Method

The proposed metric is calculated in two stages: in the first stage, the transcription is aligned with the ground-truth music notation based on its pitch content only, i.e., all of the other objects, such as rests, barlines, and time and key signatures are ignored; in the second stage, all of the objects occurring at the aligned portions of the scores are grouped together and compared. The metric reports the differences in aligned portions in terms of twelve aspects: barlines, clefs, key signatures, time signatures, notes, note spelling, note durations, stem directions, groupings, rests, rest duration, and staff assignment.

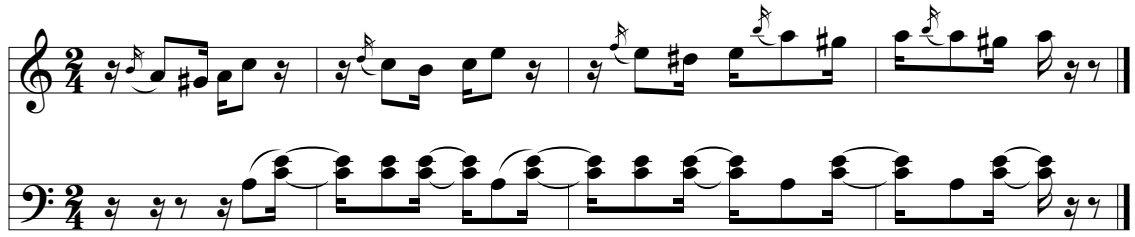
Some algorithms to efficiently calculate certain edit distances, e.g., the Wagner-Fischer algorithm to calculate the Levenshtein distance between two strings, are able to align two sequences and calculate the edit costs in a single stage. I initially tried to apply the same strategy to this problem, but I discovered that the algorithm was not



(a) Ground truth



(b) Transcription with a wrong pickup measure



(c) Transcription off by a 16th note

Figure 7.1: Comparison of two transcriptions of the same piece containing similar errors but with different readability.

sufficiently robust, especially with transcriptions highly corrupted by wrong meter estimation. Intuitively, notes are the most salient aspects of music, so it is arguable that the alignment of two transcriptions should be based primarily on that aspect, while the overall quality of the transcription should be judged on a variety of other aspects.

The ground truth and the transcription are both encoded in MusicXML, a standard format to share sheet music files between applications [29]. The two scores are

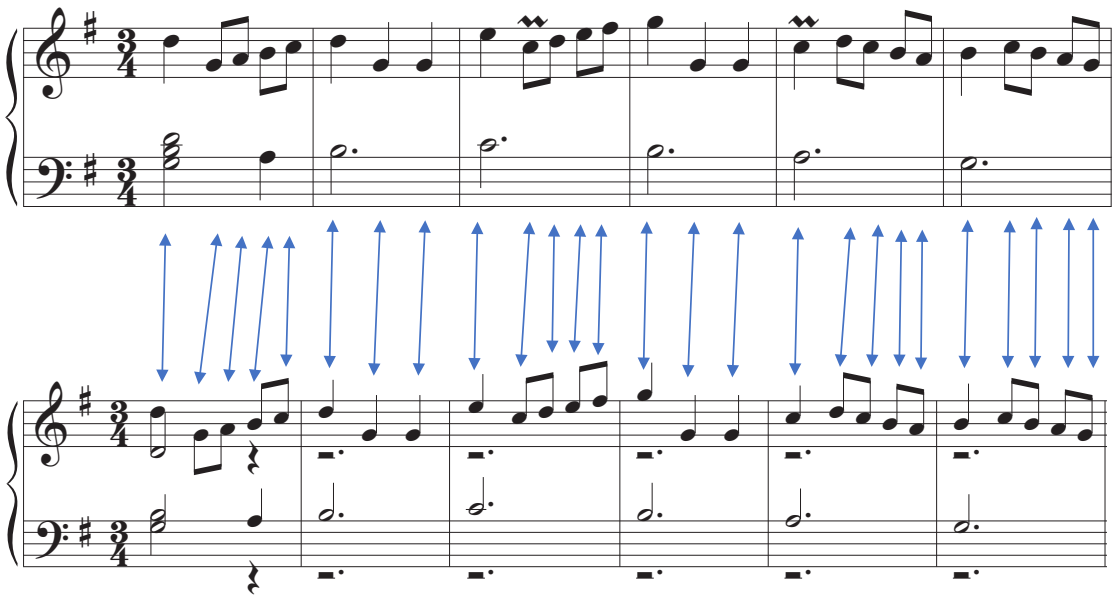


Figure 7.2: Alignment between the ground-truth (top) and a transcription (bottom) of Bach's Minuet in G. Arrows indicate aligned beats.

aligned using Dynamic Time Warping [115]. The local distance is simply the number of mismatching pitches, regardless of duration, spelling and staff positioning.

To illustrate the purpose of the initial alignment, I show two examples in Fig. 7.2 and Fig. 7.3. The alignment stage outputs a list of pairs of aligned beats. Fig. 7.2 shows the alignment of a fairly good transcription of Bach's Minuet in G from the Notebook for Anna Magdalena Bach, with the ground truth, which corresponds to the following sequence, expressed in beats, numbered as quarter notes starting from 0 (GT is ground truth, T is transcription):

GT	0.0	1.0	1.5	2.0	2.5	3.0	4.0
T	0.0	1.0	1.5	2.0	2.5	3.0	4.0
4.0	5.0	6.0	7.0	7.5	8.0	8.5	9.0
5.0	5.0	6.0	7.0	7.5	8.0	8.5	9.0
10.0	10.0	11.0	12.0	13.0	13.5	14.0	14.5
10.0	11.0	11.0	12.0	13.0	13.5	14.0	14.5
15.0	16.0	16.5	17.0	17.5			
15.0	16.0	16.5	17.0	17.5			

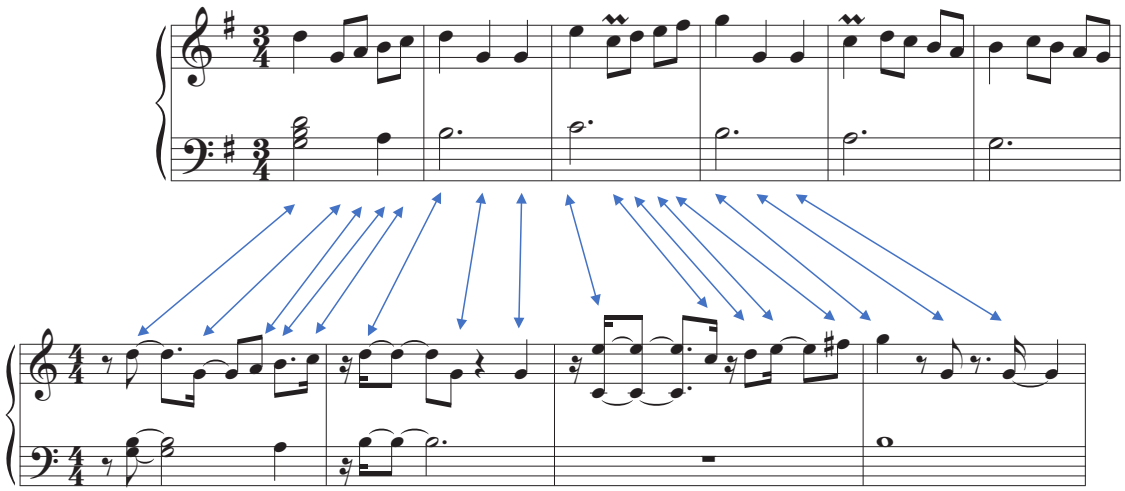


Figure 7.3: Alignment between the ground-truth (top) and another transcription (bottom) of Bach's Minuet in G. Arrows indicate aligned beats.

In this case, since the transcription is properly aligned with the ground truth, the sequence is just a list of all equal numbers, one for each onset of the notes in the score. However, beat 4.0 in the ground truth is matched with beats 4.0 and 5.0 in the transcription; the same happens for beats 10.0 and 11.0, so DTW cannot properly distinguish repeated pitches. Only one alignment is shown in the figure for clarity.

Fig. 7.3 shows an example of an alignment for a badly aligned transcription of the same piece. The corresponding sequence is the following:

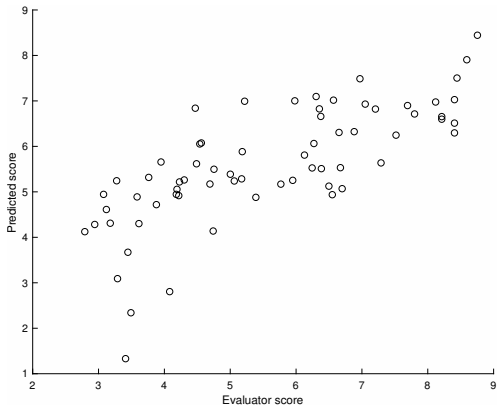
GT	0.0	0.0	0.0	1.0	1.0	1.5
T	0.0	0.5	1.0	1.75	2.0	2.5
2.0	2.5	3.0	3.0	3.0	4.0	4.0
3.0	3.75	4.25	4.5	5.0	5.5	7.0
5.0	6.0	6.0	6.0	7.0	7.5	8.0
7.0	8.25	8.5	9.0	9.75	10.25	10.75
8.0	8.5	9.0	10.0	10.0	10.0	11.0
11.0	11.5	12.0	13.5	14.75	15.0	15.0

In this case, multiple beats in the transcription correspond to the same beat in the ground truth, e.g., beat 1.0 in the ground truth corresponds to beats 1.75 and 2.0

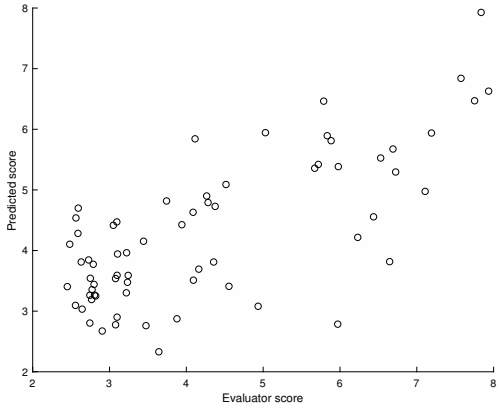
in the transcription, because a single note in the ground truth has been transcribed as two tied notes. Only one alignment is shown in the figure for clarity.

To calculate the distance between the two aligned scores, I proceed by first grouping all of the musical objects occurring inside aligned portions of the two scores into sets, thus losing the relative location of the objects within each set but preserving all of the other aspects, including staff assignment. Then the aligned sets are compared, and the differences between the two sets are reported separately. The following aspects only allow binary matching: barlines, clefs, key signatures, and time signatures. Rests are matched for duration and staff assignment, i.e., a rest with the correct duration but on the wrong staff will be considered a staff assignment error, a rest with the correct staff assignment but wrong duration will be considered a rest duration error. A missing or an extra rest will be considered a rest error. Notes are matched for spelling, duration, stem direction, staff assignment, and grouping into chords. For groupings, I only report the absolute value of the difference between the number of chords present in the two sets. The metric does not distinguish missing or extra elements. These choices were dictated by simplicity of design and implementation.

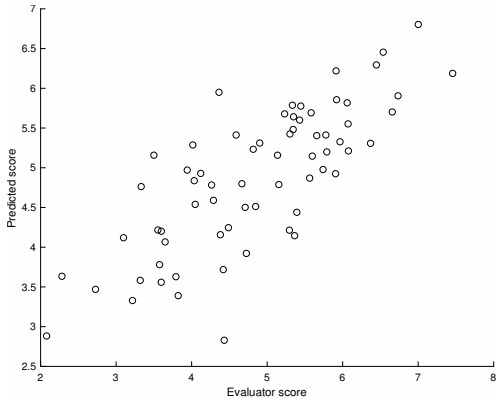
All of the errors are cumulated for all of the matching sets. The errors for barlines, notes, note spelling, note durations, stem directions, groupings, rests, rest duration, and staff assignment are then normalized by dividing the total number of errors for each aspect by the total number of musical objects taken into account in the score. This step is necessary to normalize the number of errors for pieces of different lengths. The errors for clefs, key signatures, and time signatures are not normalized, as they are typically global aspects of the scores, and not influenced by the length of the piece. This might be a limitation for pieces with frequent changes in key signature or time signature.



(a) Pitch Notation



(b) Rhythm Notation



(c) Note Positioning

Figure 7.4: Correlation between the predicted ratings and the average human evaluator ratings of all of the transcriptions in the dataset.

As an example, the set of objects at the first beat of the first measure of Fig. 7.2 include the initial barlines, clefs, time signature, key signature, and notes starting on the downbeat of the measure. Barlines, clefs, time signature, and key signature are all correctly matched. All of the notes are correct in pitch, spelling and duration, however there are two errors in stem direction, one error in grouping, and one error in staff assignment. All of the rests are considered rest errors at each respective onsets.

For the first beat of the first measure of Fig. 7.3, all of the elements of the transcription till the first transcribed notes (the three notes pointed by the first arrow) and the notes tied to them will be considered as part of the same set. The wrong key signature and time signature will be reported as errors. The two eight rests will be reported as rest errors. The three notes in the transcription are properly spelled, but their duration is wrong, so that will be counted as three note duration errors. The missing D from the chord will be reported as a note error. The extra tied notes will be reported as note errors as well.

In summary, the following twelve normalized error counts are calculated by the metric: barlines, clefs, key signatures, time signatures, notes, note spelling, note durations, stem directions, groupings, rests, rest duration, and staff assignment. In order to translate these error counts into a musically relevant evaluation, I propose to use linear regression of the twelve error counts to fit human ratings of three musical aspects of automatic transcriptions, i.e., the pitch notation, the rhythm notation, and the note positioning. For each aspect, the linear regression learns twelve weights, one for each of the normalized error counts, to fit the human ratings. These weights can then be used to predict the human ratings of other music notation transcriptions.

7.3 Experimental Results

To evaluate the proposed approach, I calculate the normalized error count and run linear regression to fit human ratings of 19 short music excerpts collected for the experiment in Chapter 6. These music excerpts were from the Kostka-Payne music theory book, all of them piano pieces by well-known composers, and were performed on a MIDI keyboard by a semi-professional piano player. These excerpts were then transcribed into music notation using four different methods: a novel method proposed in the paper (which will be referred to as CDT), MuseScore, GarageBand and Finale. For each transcription, the human evaluators were asked to assign a numerical rating between 1 and 10 for three musical aspects, i.e., the pitch notation, the rhythm notation, and the note positioning.

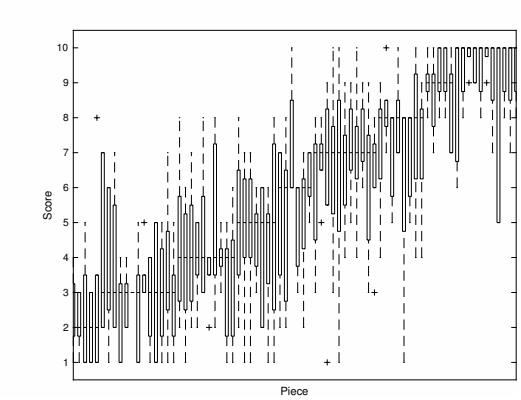
The proposed method of calculating the error counts uses MusicXML [29], the de facto standard for sharing sheet music files between applications, as the format of music notation. Two of the methods evaluated in the paper (Finale and MuseScore) can output the scores into MusicXML. For GarageBand, CDT and the ground truth, however, MusicXML was not available or was difficult to output automatically. I had to manually convert the scores into MusicXML. The transcribed scores are named with the initial of the transcription method and a number indicating the excerpt. So, `M-8.mxl` represents the eight excerpt transcribed with MuseScore. The letter K, for Kostka-Payne, indicates the ground truth scores. This dataset and a Python implementation of the proposed approach are available at `http://www.ece.rochester.edu/~acogliat/repository.html`. The implementation uses the music21 toolkit [116] for parsing the MusicXML files and processing the imported scores. The implementation has been tested with music21 V3.1.0.

In order to validate the quality of the prediction I calculated the coefficient of determination R^2, which is the square of the Pearson correlation coefficient. The R^2 was 0.558 for the pitch notation correlation, 0.534 for the rhythm notation, and 0.601 for note positioning. These results are reflected in Fig. 7.4; the proposed metric fits the data adequately, in general, even though the correlation is not perfect. It can also be noted that the prediction of the score for note positioning is the best, while the prediction of the score for rhythm notation is the worst.

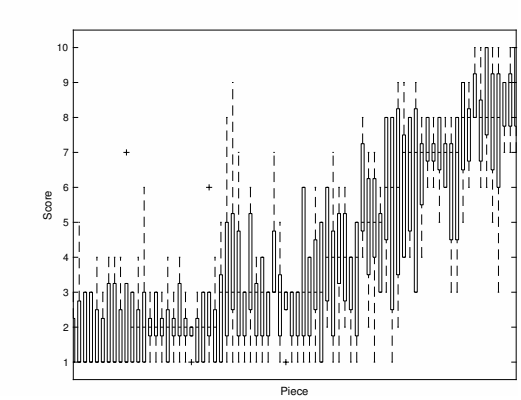
To understand the underlying causes of the covariance I firstly analyzed the ratings given by the human evaluators. As I can see from Fig. 7.5, the human evaluators were oftentimes in disagreement among themselves. It must also be noted that for the evaluation in Chapter 6, the human annotators were not given exact instructions on what features to consider for the evaluation, so a considerable amount of subjectivity and judgment calls were likely to be present in the ratings.

I also analyzed two transcriptions with the largest deviation from the predicted ratings, i.e., one transcription with a high predicted rating and a low human rating, and one transcription with a low predicted rating and a high human rating. The largest positive deviation occurred for the rhythm notation of transcription M-1, for which the proposed metric predicted a rating of 2.78, while the average human rating was 5.98. If I compare the transcription with the ground truth in Fig. 7.6 I can see that MuseScore misinterpreted the meter, causing the proposed metric to report a large number of note duration errors and barline errors, which resulted in a low rating. Human annotators, on the other side, likely penalized the meter error only once globally, but still considered the transcription acceptable overall.

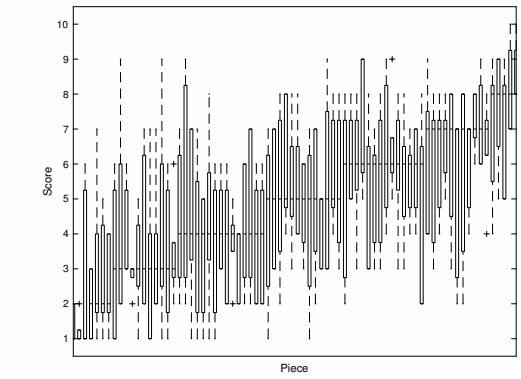
The largest negative deviation occurred for the pitch notation of transcription C-13, for which the proposed metric predicted a rating of 6.83, while the annotators assigned an average score of of 4.48. If we compare the transcription with the ground



(a) Pitch Notation

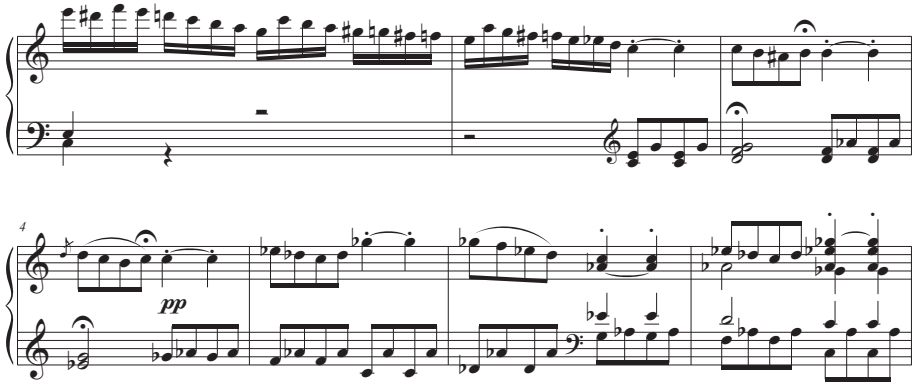


(b) Rhythm Notation

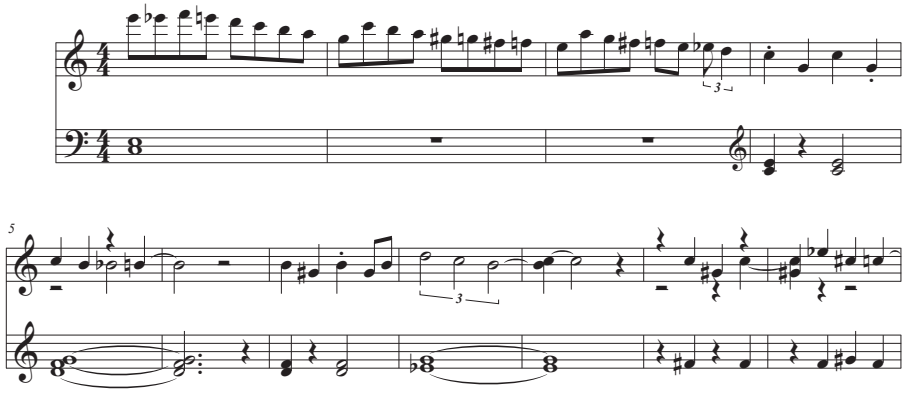


(c) Note Positioning

Figure 7.5: Distributions of the human ratings of the 76 transcriptions contained in the dataset. Each boxplot represents the ratings from 5 human evaluators.



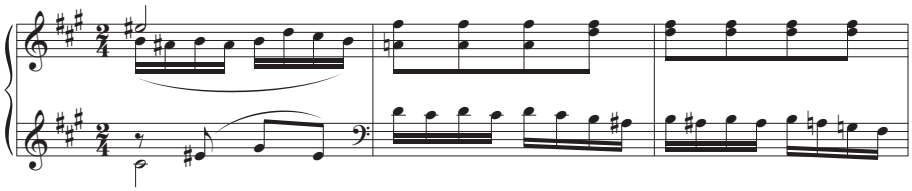
(a) Ground Truth



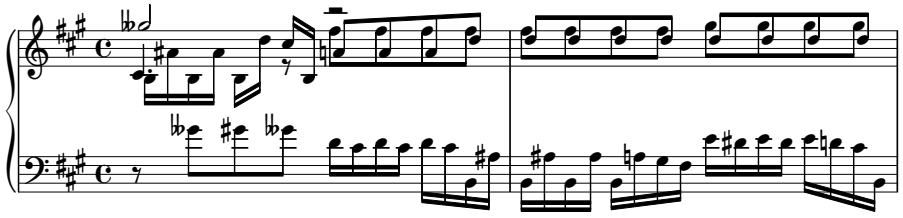
(b) M-1

Figure 7.6: Transcription of the first excerpt in the dataset by MuseScore, which shows the largest positive difference between the average human rating and the predicted rating, that is a high predicted rating and a low human rating. This evaluation difference occurs on the rhythm notation.

truth in Fig. 7.7, we can notice that CDT makes a single mistake in notating the pitches, i.e., G𝄫 instead of E♯. It also makes a systematic error notating all Bs one octave lower. Finally, not grouping the eight notes in the treble staff makes the transcription hard to read. Possibly, the human annotators penalized the transcription because of its poor readability.



(a) Ground Truth



(b) C-13

Figure 7.7: Transcription of the thirteenth excerpt in the dataset by CDT, which shows the largest negative deviation between the average human rating and the predicted rating on rhythm notation, that is a low predicted rating and a high human rating. This evaluation difference occurs on the pitch notation.

7.4 Conclusion

In this chapter I proposed an objective metric to measure the differences between music notation transcriptions and the ground truth score. The metric is calculated by first aligning the pitch content of the transcription and the ground-truth music notation, and then counting the differences in twelve key musical aspects: barlines, clefs, key signatures, time signatures, notes, note spelling, note durations, stem directions, groupings, rests, rest duration, and staff assignment. I then used linear regression to predict human evaluator ratings along three aspects of music notation, namely, pitch notation, rhythm notation, and note positioning, from the error counts. Experiments show a clear correlation between the predicted ratings and the average human ratings, even though the correlation is not perfect.

One issue with the prediction is the high variance of the evaluator ratings, which likely originates from the inherent subjectivity of the tasks. Another issue of the proposed metric is that it does not incorporate music theory knowledge, such as the method proposed by Temperley to evaluate metrical models [117].

The current experiments were conducted on music notation transcriptions of human performances recorded on a MIDI keyboard; as a consequence, the transcriptions do not contain the errors commonly observed in audio-to-MIDI conversion processes, such as octave errors and extra or missing notes [7, 1]. More research is necessary to evaluate the performance of the proposed method in the presence of such errors. In addition, the excerpts in the dataset were very short, compared to real piano pieces, so additional research is necessary to assess the robustness of the metric, and its computational complexity on longer pieces.

Chapter 8

Conclusions

In this dissertation, I proposed a full piano music transcription system, which I call *human-centric* as it presents the results to the users in the language they are most familiar with, i.e., music notation. The system is designed as two separate modules, one for parametric transcription, the other for the music notation output, so that advancements in either technology can be incorporated in the system without too much disruption.

High accuracy parametric transcription is achieved through convolutional sparse coding in the time-domain. The proposed method achieves high transcription accuracy and time precision in a variety of different scenarios, and is highly robust to moderate amounts of noise. It is also highly insensitive to reverb, as long as the training session is performed in the same environment used for recording the audio to be transcribed.

Note-length estimation is achieved with structured group sparsity. The proposed method uses multiple templates with different lengths per pitch to achieve note length estimation. Lateral inhibition regularization is introduced to ensure that at most one template per pitch is activated within an inhibition window. Global sparsity is achieved through ℓ_1 regularization to reduce false activations of wrong notes.

An initial experiment in structured dictionary learning shows the potential of parametric dictionary design based on a physical model of piano notes. More research is needed, especially in modeling the envelope of the partials.

For the generation of music notation output from a parametric transcription, I showed that the correct estimation of the meter, harmony and streams is fundamental in producing a properly formatted score. The proposed method can be combined with any note-level automatic music transcription method to complete the audio to music notation conversion process, but more experiments are needed to assess the performance on longer music pieces.

Finally, I proposed an objective metric to measure the differences between music notation transcriptions and the ground truth score. The metric is calculated by first aligning the pitch content of the transcription and the ground-truth music notation, and then counting the differences in twelve key musical aspects: barlines, clefs, key signatures, time signatures, notes, note spelling, note durations, stem directions, groupings, rests, rest duration, and staff assignment. I then used linear regression to predict human evaluator ratings along three aspects of music notation, namely, pitch notation, rhythm notation, and note positioning, from the error counts. Experiments show a clear correlation between the predicted ratings and the average human ratings.

8.1 Interactive Music Transcription

The ultimate goal of AMT is to obtain a very accurate transcription of any piano performance. A possible way to further increase the accuracy of AMT and to make it fully human-centric is to incorporate user input into the transcription process. An Interactive Music Transcription (IMT) system allows users to correct music that has

been automatically transcribed by a machine. The corrections can be used to improve the accuracy of future transcriptions (see Fig. 8.1).

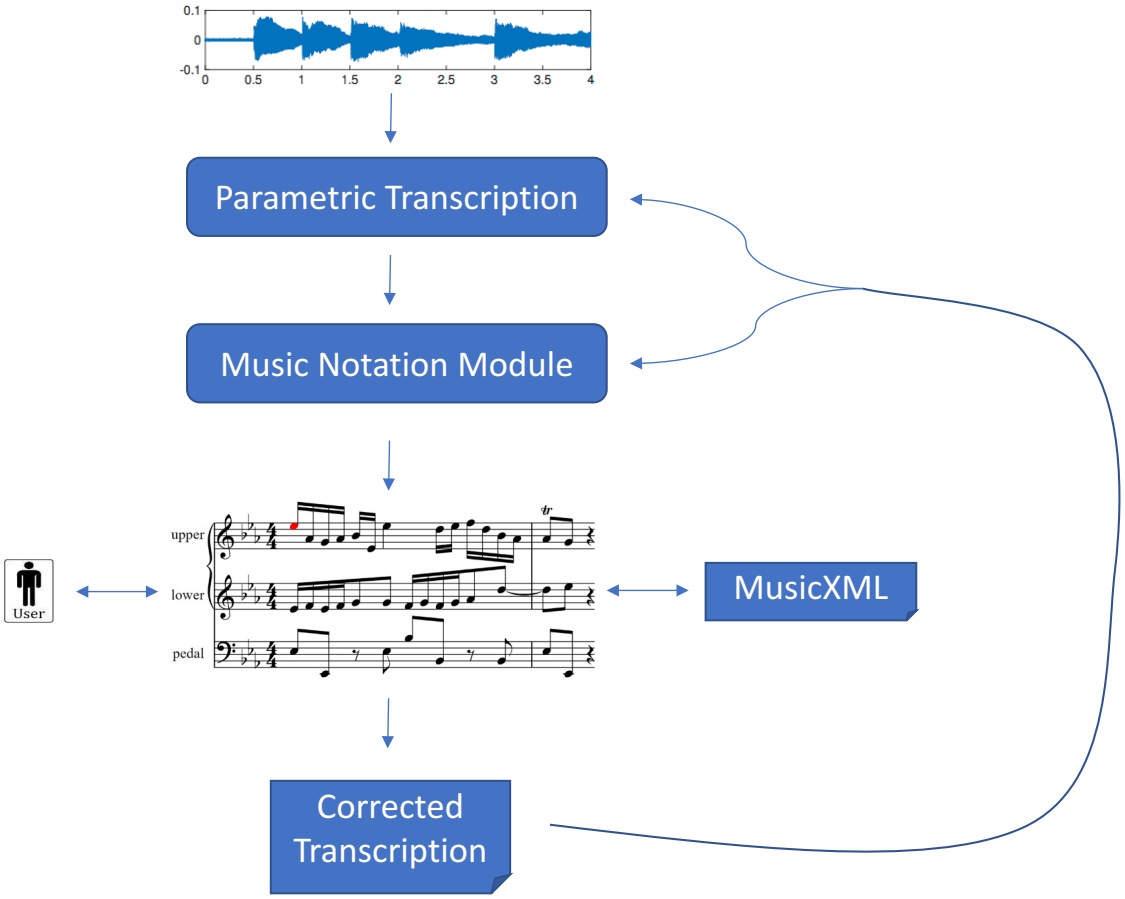


Figure 8.1: Interactive Music Transcription Dataflow.

The main challenges of such a system are how to create an interactive editor for MusicXML files and how to exploit the corrections to improve the accuracy of the future transcriptions. For the first problem, we can leverage existing public technologies, such as MuseScore[1], an open-source music editor, or Verovio[2], a lightweight open-source library for engraving music scores into SVG. Verovio looks very appealing for an IMT system as it is provided as a library that can be easily incorporated

[1] `https://musescore.com`
[2] `http://www.verovio.org`

into other programs. The developers provide two different toolkits for the library, one in Python and one in JavaScript. The Python library can be used to create a desktop application, while the JavaScript library can be used to create an interactive editor in a web browser. The second approach was tested at the AIR Lab during a Research Experience for Undergraduates (REU) by undergraduates Arlen Fan, of the University of Rochester, and Andrew V. Smith, of the University of Central Florida. The outcome of the project was an interactive music editor capable of loading a music score encoded as MusicXML and performing some basic editing functions, such as modifying pitches and durations, and adding or removing notes and rests. The editor is also capable of exporting the modified score and detecting the corrections performed by the user.

Exploiting the corrections performed by the user to improve future transcriptions is a much hard problem to solve, and it is related to the problem of incorporating music knowledge into an automatic music transcription system. A possible way to incorporate musical knowledge into the parametric transcription system described in Chapters 3 and 4 is to introduce more constraints on the activation coefficients in the form of regularization terms in eq. (3.2) and eq. (4.11). The kind of constraints to add can be determined by analyzing a large amount of corrections performed by users of the system in order to detect common systematic errors made by the systems, possibly through the application of machine learning algorithms.

Bibliography

[1] Andrea Cogliati, Zhiyao Duan, and Brendt Wohlberg. Piano transcription with convolutional sparse lateral inhibition. *IEEE Signal Processing Letters*, 24(4): 392–396, 2017.

[2] Andrea Cogliati, Zhiyao Duan, and Brendt Wohlberg. Context-dependent piano music transcription with convolutional sparse coding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2016.

[3] Andrea Cogliati, Cristina Canavesi, Adam Hayes, Patrice Tankam, Virgil-Florin Duma, Anand Santhanam, Kevin P. Thompson, and Jannick P. Rolland. MEMS-based handheld scanning probe with pre-shaped input signals for distortion-free images in Gabor-domain optical coherence microscopy. *Optics Express*, 24(12):13365, 6 2016.

[4] Andrea Cogliati and Zhiyao Duan. A metric for music notation transcription accuracy. In *Proc. of International Society for Music Information Retrieval (ISMIR)*, pages 407–413, 2017.

[5] Andrea Cogliati, David Temperley, and Zhiyao Duan. Transcribing human piano performances into music notation. In *Proc. of International Society for Music Information Retrieval (ISMIR)*, pages 758–764, 2016.

[6] Andrea Cogliati, Zhiyao Duan, and Brendt Wohlberg. Piano music transcription with fast convolutional sparse coding. In *Proc. of IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 9 2015.

[7] Andrea Cogliati and Zhiyao Duan. Piano music transcription modeling note temporal evolution. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 429–433, Brisbane, Australia, 4 2015. IEEE.

[8] David Temperley. *Music and probability*. The MIT Press, 2007.

[9] Valentin Emiya, Roland Badeau, and Bertrand David. Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. *IEEE*

Transactions on Audio, Speech, and Language Processing, 18(6):1643–1654, 2010.

[10] Tian Cheng, Simon Dixon, and Matthias Mauch. Modelling the decay of piano sounds. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 594–598, Brisbane, Australia, 2015.

[11] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407–434, 2013.

[12] James A Moorer. On the transcription of musical sound by computer. *Computer Music Journal*, pages 32–38, 1977.

[13] Martin Piszczalski and Bernard A Galler. Automatic Music Transcription. *Computer Music Journal*, 1(4):24–31, 1977.

[14] Manuel Davy and Anssi Klapuri. *Signal Processing Methods for Music Transcription*. Springer, 2006.

[15] Kazuki Ochiai, Hirokazu Kameoka, and Shigeki Sagayama. Explicit beat structure modeling for non-negative matrix factorization-based multipitch analysis. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 133–136, 2012.

[16] Tom Collins, Sebastian Böck, Florian Krebs, and Gerhard Widmer. Bridging the audio-symbolic gap: The discovery of repeated note content directly from polyphonic music audio. In *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*, 2014.

[17] Emilios Cambouropoulos. From MIDI to traditional musical notation. In *Proc. of the AAAI Workshop on Artificial Intelligence and Music: Towards Formal Models for Composition, Performance and Analysis*, volume 30, 2000.

[18] David Temperley. A unified probabilistic model for polyphonic music analysis. *Journal of New Music Research*, 38(1):3–18, 2009.

[19] Mert Bay, Andreas F Ehmann, and J Stephen Downie. Evaluation of multiple-F0 estimation and tracking systems. In *Proc. of International Society for Music Information Retrieval (ISMIR)*, pages 315–320, 2009.

[20] Perry R Cook. *Music, cognition, and computerized sound*. Cambridge, MA: Mit Press, 1999.

[21] Hideo Suzuki and Isao Nakamura. Acoustics of pianos. *Applied Acoustics*, 30 (2):147–205, 1990.

[22] Alain De Cheveigné and Hideki Kawahara. YIN, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.

[23] Dennis Gabor. Theory of communication. Part 1: The analysis of information. *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, 93(26):429–441, 1946.

[24] MIREX 2016: Multiple Fundamental Frequency Estimation & Tracking Results - MIREX Dataset, 2016. URL `http://music-ir.org/mirex/wiki/2016:MIREX2016_Results`.

[25] Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig. The Microsoft 2016 conversational speech recognition system. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 5255–5259. IEEE, 3 2017.

[26] Brian C. J. Moore. *Hearing*. Academic Press, 1995.

[27] Stanley. Sadie and John Tyrrell. *The new Grove dictionary of music and musicians*. Grove, 2001.

[28] Lawrence E Kinsler, Austin R Frey, Alan B Coppens, and James V Sanders. *Fundamentals of acoustics*, volume 1. 1999.

[29] Michael Good. MusicXML for notation and analysis. *The virtual score: representation, retrieval, restoration*, 12:113–124, 2001.

[30] Meinard Müller. *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. Springer, 2015.

[31] Jean Baptiste Joseph Fourier. *Théorie analytique de la chaleur*. Chez Firmin Didot, père et fils, 1822.

[32] Alan V Oppenheim, Alan S Willsky, and S. Hamid Nawab. *Signals and Systems*, volume 2nd ed. Pearson, 1997.

[33] Alan V Oppenheim and Ronald W Schafer. *Discrete Time Signal Processing*. 2009.

[34] Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1047, 2005.

[35] Sebastian Böck and Gerhard Widmer. Local group delay based vibrato and tremolo suppression for onset detection. In *Proc. of International Society for Music Information Retrieval (ISMIR)*, pages 361–366, 2013.

[36] MIREX2015 Results. URL `http://music-ir.org/mirex/wiki/2015:MIREX2015_Results`.

[37] Bruce Croft and John Lafferty, editors. *Language modeling for information retrieval*. Springer Netherlands, 2003.

[38] Yuta Ojima, Eita Nakamura, Katsutoshi Itoyama, and Kazuyoshi Yoshii. A hierarchical bayes model of chords, pitches, and spectrograms for multipitch analysis. In *Proc. of International Society for Music Information Retrieval (ISMIR)*, 2016.

[39] Siddharth Sigtia, Emmanouil Benetos, and Simon Dixon. An end-to-end neural network for polyphonic piano music transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(5):927–939, 5 2016.

[40] Adrien Ycart and Emmanouil Benetos. A study on LSTM networks for polyphonic music sequence modelling. In *Proc. of International Society for Music Information Retrieval (ISMIR)*, pages 421–427, 2017.

[41] Emmanouil Benetos. Polyphonic note and instrument tracking using linear dynamical systems. In *Proc. of AES International Conference on Semantic Audio*, 2017.

[42] Christopher Raphael. Automatic transcription of piano music. In *Proc. of International Society for Music Information Retrieval (ISMIR)*, 2002.

[43] Anssi P Klapuri. Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. *IEEE Transactions on Speech and Audio Processing*, 11(6):804–816, 2003.

[44] Chunghsin Yeh, Axel Röbel, and Xavier Rodet. Multiple fundamental frequency estimation of polyphonic music signals. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3, pages iii–225, 2005.

[45] Karin Dressler. Multiple fundamental frequency extraction for MIREX 2012. *Eighth Music Information Retrieval Evaluation eXchange (MIREX)*, 2012.

[46] Graham E Poliner and Dan P W Ellis. A discriminative model for polyphonic piano transcription. *EURASIP Journal on Advances in Signal Processing*, (8): 154–162, 1 2007.

[47] Antonio Pertusa and José M Iñesta. Multiple fundamental frequency estimation using Gaussian smoothness. In *IEEE International Conference on Audio, Speech, and Signal Processing*, pages 105–108, 4 2008.

[48] Shoichiro Saito, Hirokazu Kameoka, Keigo Takahashi, Takuya Nishimoto, and Shigeki Sagayama. Specmurt analysis of polyphonic music signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(3):639–650, 3 2008.

[49] Juhan Nam, Jiquan Ngiam, Honglak Lee, and Malcolm Slaney. A classification-based polyphonic piano transcription approach using learned feature representations. In *Proc. of International Society for Music Information Retrieval (ISMIR)*, pages 175–180, 2011.

[50] Sebastian Böck and Markus Schedl. Polyphonic piano note transcription with recurrent neural networks. In *IEEE International Conference on Audio, Speech, and Signal Processing*, pages 121–124, 3 2012.

[51] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling temporal dependencies in high-dimensional sequences: application to polyphonic music generation and transcription. In *29th International Conference on Machine Learning*, Edinburgh, Scotland, UK, 2012.

[52] Masataka 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(4):311–329, 2004.

[53] Zhiyao Duan, Bryan Pardo, and Changshui Zhang. Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions. *IEEE Transactions on Audio, Speech, and Language Processing,*, 18(8):2121–2133, 2010.

[54] Paul H Peeling and Simon J Godsill. Multiple pitch estimation using non-homogeneous poisson processes. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1133–1143, 10 2011.

[55] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[56] Paris Smaragdis, Bhiksha Raj, and Madhusudana Shashanka. A probabilistic latent variable model for acoustic modeling. *In Workshop on Advances in Models for Acoustic Processing at NIPS*, 2006.

[57] Paris Smaragdis and Judith C Brown. Non-negative matrix factorization for polyphonic music transcription. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2003.

[58] Graham C Grindlay and Dan P W Ellis. Transcribing multi-instrument polyphonic music with hierarchical eigeninstruments. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1159–1169, 2011.

[59] Emmanouil Benetos and Simon Dixon. A shift-invariant latent variable model for automatic music transcription. *Computer Music Journal*, 36(4):81–94, 2012.

[60] Samer A Abdallah and Mark D Plumbley. Polyphonic music transcription by non-negative sparse coding of power spectra. In *Proc. of International Society for Music Information Retrieval (ISMIR)*, pages 318–325, 2004.

[61] Ken O'Hanlon, Hidehisa Nagano, and Mark D Plumbley. Structured sparsity for automatic music transcription. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 441–444, 2012.

[62] Ken O'Hanlon and Mark D Plumbley. Polyphonic piano transcription using non-negative matrix factorisation with group sparsity. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3112–3116, 2014.

[63] Sebastian Ewert and Mark B Sandler. An augmented Lagrangian method for piano transcription using equal loudness thresholding and LSTM-based decoding. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 146–150, 2017.

[64] Ray Meddis and Michael J Hewitt. Virtual pitch and phase sensitivity of a computer model of the auditory periphery. I: pitch identification. *Journal of the Acoustical Society of America*, 89:2866–2882, 1991.

[65] T Tolonen and M Karjalainen. A computationally efficient multipitch analysis model. *IEEE Transactions on Speech and Audio Processing*, 8(6):708–716, 11 2000.

[66] Paul J Walmsley, Simon J Godsill, and Peter J W Rayner. Polyphonic pitch tracking using joint Bayesian estimation of multiple frame parameters. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 119–122. IEEE, 1999.

[67] Ali Taylan Cemgil, Hilbert J Kappen, and David Barber. A generative model for music transcription. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(2):679–694, 3 2006.

[68] Manuel Davy, Simon Godsill, and Jerome Idier. Bayesian analysis of polyphonic western tonal music. *The Journal of the Acoustical Society of America*, 119(4): 2498–2517, 2006.

[69] Juan P Bello, Laurent Daudet, and Mark B Sandler. Automatic piano transcription using frequency and time-domain information. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(6):2242–2251, 2006.

[70] Li Su and Yi-Hsuan Yang. Combining spectral and temporal representations for multipitch estimation of polyphonic music. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 23(10):1600–1612, 10 2015.

[71] Mark D Plumbley, Samer A Abdallah, Thomas Blumensath, and Michael E Davies. Sparse representations of polyphonic music. *Signal Processing*, 86(3): 417–431, 2006.

[72] Matti P Ryynänen and Anssi P Klapuri. Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, 32(3): 72–86, 4 2008.

[73] Zhiyao Duan and David Temperley. Note-level music transcription by maximum likelihood sampling. In *Proc. of International Society for Music Information Retrieval (ISMIR)*, pages 181–186, 2014.

[74] Matija Marolt, Alenka Kavcic, and Marko Privosnik. Neural networks for note onset detection in piano music. In *Proc. International Computer Music Conference*, 2002.

[75] Giovanni Costantini, Renzo Perfetti, and Massimiliano Todisco. Event based transcription system for polyphonic piano music. *Signal Processing*, 89(9):1798–1811, 2009.

[76] Hirokazu Kameoka, Takuya Nishimoto, and Shigeki Sagayama. A multipitch analyzer based on harmonic temporal structured clustering. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3):982–994, 2007.

[77] Taylor Berg-Kirkpatrick, Jacob Andreas, and Dan Klein. Unsupervised transcription of piano music. In *Advances in Neural Information Processing Systems*, pages 1538–1546, 2014.

[78] Sebastian Ewert, Mark D Plumbley, and Mark Sandler. A dynamic programming variant of non-negative matrix deconvolution for the transcription of struck string instruments. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 569–573, Brisbane, Australia, 2015.

[79] Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1): 33–61, 1998.

[80] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Robert Fergus. Deconvolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2528–2535, 6 2010.

[81] Brendt Wohlberg. Efficient algorithms for convolutional sparse representations. *IEEE Transactions on Image Processing*, 25(1):301–315, 1 2016.

[82] Thomas Blumensath and Mike Davies. Sparse and shift-invariant representations of music. *IEEE Transactions on Audio Speech and Language Processing*, 14(1):50–57, 2006.

[83] Roger Grosse, Rajat Raina, Helen Kwong, and Andrew Y Ng. Shift-invariance sparse coding for audio classification. *arXiv preprint arXiv:1206.5241*, 2012.

[84] Brendt Wohlberg. Efficient convolutional sparse coding. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 7173–7177, Florence, Italy, 5 2014.

[85] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1): 1–122, 2011.

[86] Hilton Bristow, Anders Eriksson, and Simon Lucey. Fast convolutional sparse coding. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 391–398, 2013.

[87] Brendt Wohlberg. SParse Optimization Research COde (SPORCO), 2015. URL `http://math.lanl.gov/~brendt/Software/SPORCO/`.

[88] Ping-Keng Jao, Yi-Hsuan Yang, and Brendt Wohlberg. Informed monaural source separation of music based on convolutional sparse coding. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 236–240, Brisbane, Australia, 4 2015.

[89] Emmanouil Benetos and Tillman Weyde. BW3 - MSSI-PLCA_fast_NoteTracking2. URL `http://www.music-ir.org/mirex/wiki/2013:Multiple_Fundamental_Frequency_Estimation_%26_Tracking_Results`.

[90] Emanuël Habets. RIR Generator. URL `https://www.audiolabs-erlangen.de/fau/professor/habets/software/rir-generator`.

[91] Pablo Sprechmann, Ignacio Ramirez, Guillermo Sapiro, and Yonina C. Eldar. C-HiLasso: a collaborative hierarchical sparse modeling framework. *IEEE Transactions on Signal Processing*, 59(9):4183–4198, 9 2011.

[92] Yann LeCun, Arthur D Szlam, and Karol Gregor. Structured sparse coding via lateral inhibition. In *Advances in Neural Information Processing Systems*, pages 1116–1124, 2011.

[93] Ewout van den Berg and Michael P. Friedlander. Joint-sparse recovery from multiple measurements. Technical report, Department of Computer Science, University of British Columbia, 4 2009.

[94] Brendt Wohlberg. Convolutional sparse representations as an image model for impulse noise restoration. In *2016 IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, pages 1–5. IEEE, 7 2016.

[95] Emmanouil Benetos and Tillman Weyde. An efficient temporally-constrained probabilistic model for multiple-instrument music transcription. *Proc. International Computer Music Conference*, pages 701–707, 2015.

[96] Matija Marolt. SONIC: transcription of polyphonic piano music with neural networks. In *Audiovisual Institute, Pompeu Fabra University*, pages 217–224, 2001.

[97] Emmanuel Vincent, Nancy Bertin, and Roland Badeau. Adaptive harmonic spectral decomposition for multiple pitch estimation. *IEEE Transactions on Audio, Speech and Language Processing*, 18(3):528–537, 2010.

[98] Xabier Jaureguiberry, Pierre Leveau, Simon Maller, and Juan Jose Burred. Adaptation of source-specific dictionaries in Non-Negative Matrix Factorization for source separation. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 5–8. IEEE, 5 2011.

[99] Siying Wang, Sebastian Ewert, and Simon Dixon. Identifying missing and extra notes in piano recordings using score-informed dictionary learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(10):1877–1889, 10 2017.

[100] Alfred M. Bruckstein, David L. Donoho, and Michael Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81, 2 2009.

[101] Julien Mairal, Francis Bach, and Jean Ponce. Sparse modeling for image and vision processing. *Foundations and Trends in Computer Graphics and Vision in Computer Graphics and Vision*, 8(2-3):85–283, 12 2014.

[102] Mehrdad Yaghoobi, Laurent Daudet, and Mike E Davies. Parametric dictionary design for sparse coding. *IEEE Transactions on Signal Processing*, 57(12):4800–4810, 12 2009.

[103] Yen T. H. Nguyen, Moeness G. Amin, Mounir Ghogho, and Des McLernon. Time-frequency signature sparse reconstruction using chirp dictionary. In Fauzia Ahmad, editor, *Compressive Sensing IV*, volume 9484. International Society for Optics and Photonics, 5 2015.

[104] Jeffrey C O'Neill and Patrick Flandrin. Chirp hunting. In *Proc. of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pages 425–428. IEEE, 1998.

[105] Sylvain Merlet, Emmanuel Caruyer, and Rachid Deriche. Parametric dictionary learning for modeling EAP and ODF in diffusion MRI. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 10–17. Springer, Berlin, Heidelberg, 2012.

[106] Rudolf A. Rasch and Vincent Heetvelt. String inharmonicity and piano tuning. *Music Perception: An Interdisciplinary Journal*, 3(2):171–189, 12 1985.

[107] Gabriel Weinreich. Coupled piano strings. *The Journal of the Acoustical Society of America*, 62(6):1474, 1977.

[108] Ralf Gunter Correa Carvalho and Paris Smaragdis. Towards end-to-end polyphonic Music transcription: transforming music audio directly to score. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 151–155, 2017.

[109] Haruto Takeda, Naoki Saito, Tomoshi Otsuki, Mitsuru Nakai, Hiroshi Shimodaira, and Shigeki Sagayama. Hidden Markov model for automatic transcription of MIDI signals. In *Proc. of IEEE Workshop on Multimedia Signal Processing*, pages 428–431, 2002.

[110] Ali Taylan Cemgil. *Bayesian music transcription.* PhD thesis, Radboud University Nijmegen, 2004.

[111] Ioannis Karydis, Alexandros Nanopoulos, Apostolos Papadopoulos, Emilios Cambouropoulos, and Yannis Manolopoulos. Horizontal and vertical integration/segregation in auditory streaming: a voice separation algorithm for symbolic musical data. In *Proc. 4th Sound and Music Computing Conference (SMC'2007)*, 2007.

[112] Harald Grohganz, Michael Clausen, and Meinard Müller. Estimating musical time information from performed MIDI files. In *Proc. of International Society for Music Information Retrieval (ISMIR)*, pages 35–40, 2014.

[113] Jonathan M. Keith, editor. *Bioinformatics*, volume 1525 of *Methods in Molecular Biology.* Springer New York, New York, NY, 2017.

[114] Gonzalo Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 33(1):31–88, 3 2001.

[115] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 2 1978.

[116] Michael Scott Cuthbert and Christopher Ariza. music21: a toolkit for computer-aided musicology and symbolic music data. In *Proc. of International Society for Music Information Retrieval (ISMIR)*, 2010.

[117] David Temperley. An evaluation system for metrical models. *Computer Music Journal*, 28(3):28–44, 2004.