Automated Music Analysis Using Dynamic Graphical Models

by

Brian Kenneth Vogel

B.S.E.E. (University of Michigan) 1996
M.S. (University of California, Berkeley) 2001

A dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Michael Jordan, Chair
Professor David Wessel
Professor Nelson Morgan

Fall 2005

The dissertation of Brian Kenneth Vogel is approved:

Chair

Date

Date

Date

University of California, Berkeley

Fall 2005

Automated Music Analysis Using

Dynamic Graphical Models

by

Brian Kenneth Vogel

ABSTRACT

Automated Music Analysis Using

Dynamic Graphical Models

by

Brian Kenneth Vogel

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Michael Jordan, Chair

Music transcription refers to the process of listening to a piece of music and generating the underlying musical score. This is a task that has traditionally required a musician trained in transcription. A robust automated music transcription system would have applications to the music information retrieval (MIR) community, such as automated indexing of musical recordings and query by humming, for example. Good results on the automated music transcription problem might also carry over to the related problem of automated speech recognition. An automated transcription system would also be a useful component in software that teaches one how to play an instrument by allowing the software to transcribe performances and then provide feedback. Other applications include transcription of improvised performances and automated score following.

The problem of polyphonic transcription with instrument identification is a topic that has not been explored in the literature. In this thesis, we propose and implement dynamic graphical models (DGMs) for multi-instrument polyphonic transcription. The graphical models formalism provides for families of probability distributions to be modeled in a concise, modular, and intuitive manner. This thesis is concerned with making some progress on the problem of automated polyphonic multi-instrument music transcription. By multi-instrument transcription, we mean a system capable of listening to a recording in which two or more instruments are playing, and identifying the notes played by each instrument.

Our transcription system models both the musical signal behavior as well as some musical structure. We use an instrument-specific timbre model for musical signals that models both the spectral behavior and the variation in the overall sound intensity with time within a note event. We also illustrate the modularity of the graphical models approach by making some modifications to our multi-instrument transcription system to create a system for guitar transcription.

DEDICATION

Dedicated to my parents, Kenneth and Emilene Vogel.

ACKNOWLEDGMENTS

Foremost, I would like to thank my advisors, Michael Jordan and David Wessel, for supporting me and giving me the freedom to work on this project. I would like to thank Nelson Morgan for serving on my qualifying exam and dissertation committees. I would like to thank Pravin Varaiya for serving on my qualifying exam committee.

I would like to thank Peter Kassakian and Adrian Freed for their helpful advice and for their help in providing guitar recordings. I would like to thank Francis Bach for his discussions and advice early on in this research. I also wish to acknowledge financial support of the Jerry and Evelyn Hemmings Chambers Chair in Music.

Contents

Chapter 1

Introduction

Music transcription refers to the process of listening to a piece of music and generating the underlying musical score. This is a task that has traditionally required a musician trained in transcription. A robust automated music transcription system would have applications to the music information retrieval (MIR) community, such as automated indexing of musical recordings and query by humming, for example. Good results on the automated music transcription problem might also carry over to the related problem of automated speech recognition. An automated transcription system would also be a useful component in software that teaches one how to play an instrument by allowing the software to transcribe performances and then provide feedback. Other applications include transcription of improvised performances and automated score following.

1.1 Overview of the music performance process

In order to understand the issues involved in music transcription, it is first necessary to understand what is involved in the process of performing music from a score. Therefore, we now give a brief overview of the music performance process. Figure 1.1 shows this process graphically.

The starting point is the score, which represents the composer's instructions for performance. A score is a symbolic human-readable representation of a musical piece. Figure 1.2 shows an example of a musical score. We provide a more precise definition of a score in

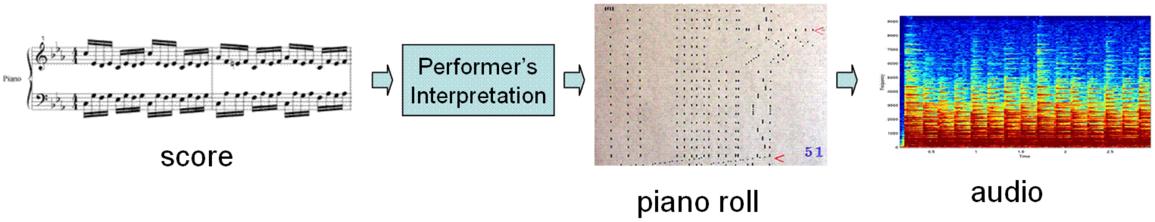

Figure 1.1: A graphical representation of the music performance process. A performer reads from a score, interprets the score expressively, and supplies musical control gestures to an instrument, producing sound.

Section 1.2.

It is common for music to be performed expressively. Music played without expressiveness can sound cold and mechanical. Examples of expressive performance attributes include tempo and rhythmic deviations from the note durations as notated in the score. The amount and type of expressiveness can depend on the particular musical genre, as well as on the performer. The score can therefore be regarded as a somewhat abstract representation of a musical piece, since some expressive performance details are often left out. There are several reasons why such a representation can be desirable: A score that is notated using simple and regular note duration symbols tends to be more readable. Such a representation can also allow the performer more expressive freedom in performing a piece. For example, in notating trills, it is common to simply place the word “trill” or “tr” next to the principle note of the trill in the score. This leaves the execution style of the trill up to the performer (which note to start on, speed of the trill, etc.). Whereas, if the notes of the trill were noted explicitly in the score, it might be considered incorrect for the performer to perform the trill differently than notated. It is up to the composer (and in some cases, the editor) to decide how specific the performance instructions in the score should be.

For example, Frederic Chopin generally used a trill symbol in his written piano music to indicate a trill. An example of this appears in Figure 1.2. However, we can see that in this particular edition, the editor chose to provide an example of a sequence of notes for executing the trill. Note also that the phrase *ritenuto* (a gradual decrease in tempo) appears in the score. This indicates the Chopin would like the performer to reduce the tempo, but the amount of tempo reduction is left to the performer's discretion. In some cases, the composer may choose to make tempo changes more explicit by using more complex notation for the note durations. Brahms decided to make the amount of ritenuto explicit in the last few measures of his Rhapsody for piano in G-minor, which is shown in Figure 1.3.

The score, therefore, typically contains both explicit and implicit performance instructions. Some of the implicit performance information is also contained in the history of performance styles for a particular musical genre. For example, a Bach piece is typically performed with less rhythmic expressiveness than a Chopin piece.

In summary, the musical performance process operates as follows: a composer writes a score, a performer then interprets the score by performing the score expressively. The performance consists of playing an instrument, which consists of supplying musical control gestures to an instrument, causing it to produce sound.

For example, in the case of playing a piano, the musical control gestures would involve depressing the keys with some velocity at the note onset, and releasing the keys (or sustain pedal) at the note offset. We will use the term *piano roll* to refer to a symbolic representation of a musical performance at the level that is used to directly control (i.e., play) a musical instrument. We use this term since an actual piano roll used in player pianos is a concrete example.

1.2 Terminology

We now provide definitions for the music transcription terminology used in this thesis.

1. *Partial*

   A *partial* is a narrow bump in the spectrum, corresponding to resonant frequencies in non-percussive acoustic instruments. Many musical instruments produce sounds such that the partials are harmonically related. That is, the partials are evenly spaced in frequency. In this case, the partials can more specifically be referred to as *harmonics*. The first (lowest frequency) partial is often referred to as the *fundamental frequency* or simply the *fundamental* or the *pitch*.

2. *Pitch*

   Pitch is the perceptual attribute of sound corresponding to the physical attribute of frequency. The pitch of a sound can be measured in psychoacoustic experiments by presenting a listener with a sound and allowing the listener to adjust the frequency of a pure tone generator until the two sounds have the same perceived frequency. The pitch of the sound is then taken to be the frequency of the corresponding pure tone that is the best match.

   Pitch perception is a complex process. For example, consider the notion of *virtual pitch* [Ter74], which is where a sound has a distinct pitch but has no energy at frequencies near the perceived pitch. A sound exhibiting virtual pitch can be easily created by synthesizing only the higher harmonics while leaving the fundamental out. For example, a sound that consists of the 3rd through 10th harmonics will have a pitch corresponding to the 1st harmonic, even though the fundamental and 2nd harmonic are missing. We note that some acoustic instruments produce little energy at the fundamental relative to the higher harmonics. Notes played in the lower registers of the piano are an example.

3. *Note event*

   We define a *note event* as the sounding of a single note, from onset to offset. A note event is the acoustic event that corresponds to a note symbol in a score.

4. *Intensity envelope*

   The *intensity envelope*, also known as the *amplitude envelope*, refers to the variation in sound intensity within a note event.

5. *Onset time* We refer to *onset time* as the time at which a note begins to sound. Depending on the instrument, we may be able to define the onset time in terms of a corresponding physical action (modeled as instantaneous) performed on an instrument. For example, we might define the onset time of a piano note to be the time at which the key is depressed. In evaluating transcription performance, we will simply define onset time as the time of the corresponding onset event in the MIDI file (truth score) that was used as the input to the transcription system.

   The notion of onset time can become ambiguous in the case of instruments that are capable of gradual changes in sound intensity, or that can change their timbre suddenly. In order to avoid such issues, we will only consider instrument whose onset sounds are characterized by a sudden increase in sound intensity followed by either a relatively constant intensity or a gradually decreasing intensity. We will assume that the timbre (e.g., brightness) does not change suddenly within a note event. Gradual changes in brightness are allowed, however.

6. *Offset time*

   We define the *offset time* as the time at which a note is no longer sounding. Some musical instruments, such as the harpsichord, are characterized by a gradual decay in sound intensity during the sounding of a note. For such instruments, the offset time can be ambiguous. The offset time turns out to be much mess perceptually significant than onset time, however [Bre90].

7. *Timbre*

   Timbre refers to the perceptual attributes of musical sounds that allows us to distinguish one instrument from another. Timbre can sometimes also vary within a single instrument. For example, the timbre of the piano changes as a function of the velocity at which a key is struck. We will use the definition of timbre used by Sterian in [Ste99] and defined by the American National Standards Institute as: “that attribute of auditory sensation in terms of which a listener can judge that two sounds similarly presented and having the same loudness and pitch are dissimilar."

Some attributes of timbre include spectral envelope (or relative partial intensities), sound intensity variation vs time, transient noise at the note onset, expressive variation in loudness and/or pitch, location of partials (harmonic, relatively harmonic, inharmonic) etc..

8. *Traditional score*

   A *traditional score* or simply *score*, is a human-readable representation of a musical piece. The score represents the composer's instructions for performing a piece. A traditional score is a written representation of music in which standard traditional music notation is used. An example of a score is shown in Figure 1.2. Note symbols are used to represent pitch and note duration. Rests are used to represent periods of silence. The score has a notion of tempo, and of time signature, which implies some duration for a beat. The note and rest durations are notated in fractions of the beat length. The time signature implies some number of beats per measure. The key signature along with the horizontal placement of a note symbol on the five-line staff represents its pitch. The score also typically contains information such as instrument label, dynamic level, and various expressive performance instructions.

9. *Piano roll score*

   We use the term *piano roll representation* or *piano roll score* to refer to a symbolic representation of a musical performance at the level that is used to directly control (i.e., play) a musical instrument. We consider a piano roll representation to be a machine readable or instrument readable musical representation. We do not consider a piano roll representation to be a very human readable representation. Unlike a traditional score, in a piano roll score, note durations are not quantized into some fraction of a basic beat length, and there may not even be a notion of tempo in the score. Rather, notes are notated by specifying their pitch (and possible instrument label), dynamic level, onset time, and offset time. Onset and offset times are typically notated relative to the starting time of the piece. Examples of a piano roll representation of music include a MIDI file and of course an actual piano roll.

Note that both the traditional score and the piano roll score can be considered to instructions for performing a musical piece. The difference is that we consider the traditional score to serve as performance instructions for a human performer that allow for further expressive interpretation by the performer. Whereas, we consider the piano roll score to serve as performance instructions for a machine or instrument that does not allow for further expressive performance.

10. *Pitch tracking*

    We consider *pitch tracking* or *pitch detection* to consist of the process of analyzing an audio signal to obtain pitch as a function of time. Typically, a pitch tracker will provide pitch updates at regular uniformly spaced instants in time. Some pitch trackers also have an intensity output, and/or a voiced/unvoiced output. For example, in the context of speech signal analysis, a pitch tracker might provide pitch update values every 25 msec or so, along with a voiced/unvoiced decision output. Polyphonic pitch tracking refers to the process of identifying pitch as a function of time for two or more sound sources (e.g., human speakers or musical instruments) in the input audio signal.

    It is common for researchers to use the terms *pitch tracking* and *transcription* interchangeably. For example, *polyphonic pitch tracking* is often used interchangeably with *polyphonic (audio to piano roll) music transcription* in the literature. However, in this thesis, we consider audio to piano roll transcription to be different than pitch tracking. The difference is that audio to piano roll transcription goes a step further than pitch tracking, because symbolic note events (with associated onset time, musically quantized pitch, etc.) are identified. Whereas in pitch tracking, we only have numerical pitch estimates that are updated at regular intervals.

11. *MIDI*

Musical Instrument Digital Interface (MIDI) is a protocol for representing musical instrument control signals [Rot92]. MIDI can be thought of as a protocol for representing a piano roll score. Examples of musical control events that can be represented in MIDI include onset time, offset time, note number (pitch), volume, instrument number, etc.. Many digital musical instruments, particularly keyboards, are capable of receiving and transmitting MIDI events. Many personal computers are capable of playing piano roll transcriptions in the MIDI file format using either a hardware synthesizer in the sound card, or a software synthesizer.

1.3 Overview of the music transcription problem

We consider music transcription to be the inverse of the music performance process discussed in Section 1.1. Music transcription refers to taking the acoustic signal as input and producing the underlying score as the output. However, there is then the question of what the transcribed score should represent. Should the score consist of a symbolic representation of the performance at the level required to directly control an instrument (the "piano roll" box in Figure 1.1)? Or, should the score consist of a more human readable symbolic representation of the performance where certain performance information that is deemed expressive is not explicitly represented (the "score" box in Figure 1.1)?

The former has the advantage that it may more accurately represent what was actually performed. However, it has the disadvantage of possibly being quite unreadable to a (human) musician. The latter has the advantage of being more readable to a musician. However, it has the disadvantage in that some of the explicit expressive performance information may be lost. It also implies that a decision needs to be made as to just what constitutes "expressive performance."

Note that in the latter, the transcriber must undo the expressive performance information, and this is generally not a well-formulated problem. It involves making musically reasonable tradeoffs between producing a readable score and accurately representing what was actually performed. Given a recording of a piece, different human transcribers may each produce different but arguably musically reasonable transcriptions.

Motivated by the above discussion, we propose breaking the transcription problem into three basic problems: the general audio to score problem, and two subproblems.

1. *Audio to score transcription*

   In the audio to score transcription problem, the goal is to take the audio signal as input and produces the underlying score in standard musical notation. The score contains symbols representing the notes that were played and rests, with their durations expressed in fractions of a beat. The score also can contain information such as key signature, time signature, measure boundaries, expressive performance instructions, dynamic level, tempo, phrasing, and instructions for expressive performance, for example.

   A goal in audio to score transcription is to produce a score that is both readable by a musician and that is also a musically reasonable representation of what was performed. We note that this is not a well-defined problem, since it is not possible in general to undo a performer's expressive performance and arrive back at the original score. However, musicians would argue that there are certainly musically reasonable ways of transcribing expressively performed musical pieces. In fact, there may be many acceptable transcriptions for a given input recording, such as the trill and ritenuto examples in Section 1.1.

2. *Audio to piano roll transcription*

   The audio to piano roll problem takes the audio signal as input and produces a piano roll score. Although a piano roll score contains symbolic note events, it typically contains no musicalogical information other than constraining the pitches of the transcribed notes to be contained in some set of allowable pitches (e.g., the equal-tempered 12 tone scale). That is, unlike a traditional score, a piano roll score has no notion of key signature, musical meter, measure boundaries, tempo, etc.. The note durations are not quantized into musically meaningful durations (such as fractions of a beat).

A piano roll score may be appropriate when the goal of transcription is to provide an accurate representation of what was actually performed (or should be performed) without consideration for human readability. A piano roll score may be appropriate when we are interested in obtaining a compact symbolic representation of a musical performance.

3. *Piano roll to score transcription*

   The goal of piano roll to score transcription is to take the piano roll score as input and produce a traditional score as the output. This problem involves the same musical issues as audio to score transcription (e.g., undoing expressive performance), but the properties of the acoustic signal do not come into play.

1.4 What makes music transcription hard?

A musician trained in transcription is capable of transcribing music much more robustly than any present automated transcription systems. The current state of the art in automated transcription systems is still far from a solution to the problem of robustly transcribing most interesting music. However, partial solutions to the transcription problem may still find many practical uses at the present.

In Chapter 2, we present a review of the existing literature on automated music transcription, where we will discuss recent work and results on the various transcription subproblems. Here, we discuss some of the properties of recorded music that tend to make the audio to score transcription problem hard.

1.4.1 Musical signal knowledge

Many musical instruments are capable of producing sounds that have a distinct perceived pitch. These include instruments such as the piano, guitar, violin, etc.. We will refer to this class of instruments as the *melodic instruments*. Instruments such as drums and other percussive instruments do not produce pitched sounds.

In a typical musical recording, there may be several melodic instruments and drums playing simultaneously. In order to identify the instruments, we must of course have some prior knowledge of the instruments. Probably even most non musicians would have little difficulty identifying common instruments such as piano, trumpet, and drums in a musical recording, even if the instruments are sounding simultaneously. Depending on the type of music and polyphony, a trained musician might to able to correctly transcribe most of the notes. Automated transcription systems, however, have great difficulty on this problem.

In order to identify instruments in a recording, we must somehow make use of prior instrument-specific knowledge, which we refer to as an instrument timbre model, or simply a timbre model. Incorporating this knowledge into an automated system in a way that is robust to interfering instrument sounds (polyphonic recordings) has turned out to be a difficult problem. As of this writing, there appear to be no published results on the problem of joint polyphonic transcription with instrument identification.

Even the simpler problem of polyphonic transcription without instrument identification is still considered a hard problem. It is quite common for simultaneously sounding notes to have common partials. This is because in many musical genres, chord pitches tend to be related by ratios of small integers and most musical instruments have relatively harmonic partials. For such sounds, identifying the set of pitches that best explains the observed partials can become difficult. We note also that sometimes partial corresponding to the higher harmonics will have more energy than the fundamental. Therefore, even in a monophonic recording, the most prominent partial does not always correspond to the fundamental. Also it is common for simultaneously-played notes to share a common onset time. Therefore, we cannot often use common onset as a cue for the onsets of individual notes.

Many melodic instruments produce some transient noise at the note onset. The onset transient could potentially be useful in identifying the onset times. However, it can also cause problems for polyphonic transcription because at the times where note onsets and/or percussive sounds are sounding, there may be broadband noisy processes that obscure other sounding notes and make the identification of partials more difficult.

Consider also two notes played simultaneously an octave apart. The higher note will share all of its harmonics with the lower note. Therefore, if nothing is known about the relative distribution of energy among the harmonics, then the presence of the higher note becomes ambiguous. Therefore, a suitable instrument timbre model is required in order to identify simultaneously played notes that are one or more octaves apart.

Even monophonic transcription is not a solved problem. Transcription from an audio recording to either a piano roll or traditional score involves identifying symbolic note events and is a harder problem than monophonic pitch tracking. In order to identify onsets in a musical signal, we must take into account how both the pitch and the intensity change with time. As an example, consider a piano key that is repeatedly struck while the sustain pedal is depressed. The pitch will remain constant as a function of time. Therefore, unless we take into account the sound intensity variation with time, it will not be possible to identify the repeated note onsets. Depending on the instrument, it may also be possible to produce several distinct types of onsets. For example, in guitar playing, an onset can correspond to a plucked string, a hammer-on, or pull-off. In transcribing guitar music, we might be interested in distinguishing between these different kinds of onsets and notating them appropriately.

1.4.2 Musical structure knowledge

A traditional score contains musical structure information (also referred to as musicological information in the literature, e.g.,[Kla99]) such as the notes that were played and the rests, with the durations of both notated in fractions of a beat. The score also has measure boundaries, instrument assignments for all notes, key signature, tempo, and time signature indications. Recall that the score often abstracts away explicit expressive performance information.

One problem for automated transcription systems is that the task of undoing expressive performance to arrive back at the underlying score is not well defined. This is because if the performer is given complete freedom in interpreting a piece expressively, the resulting performance could be quite different from the original score, and it would be impossible to know which aspects of the performance are due to the score and which are due to the performer’s expressive interpretation.

However, given enough musical knowledge, it becomes possible to create a “reasonable” score. We can define a reasonable score as one that musicians would agree makes an appropriate tradeoff between human readability and accurately representing performance details. Of course, this is not a definition that one could use to objectively measure the performance of an automated transcription system, since it is defined in terms of subjective notions of what is musically reasonable. We also note that for a given performance, there may exist several distinct scores that could be judged as musically reasonable.

We can now see some of the difficulties involved in automating this process. First of all, it is necessary to track the tempo. Depending on the musical genre, it might be common for the tempo to change suddenly, and the tempo might also vary significantly within individual measures as well. This can make it difficult to quantize the note and rest durations to fractions of the beat length in a way that is not overly complex so as to remain readable to a musician.

1.5 Scope of this thesis

The problem of polyphonic transcription with instrument identification is a topic that has not been explored in the literature. In this thesis, we propose and implement dynamic graphical models (DGMs) for multi-instrument polyphonic transcription.

The graphical models formalism provides for families of probability distributions to be modeled in a concise and intuitive manner. Inference and learning algorithms exist that can be applied to general graphical models. Exact inference is sometimes not tractable, but progress is being made on approximate inference algorithms. Graphical models are also flexible in that small changes to a model typically require only small changes to the implementation. The formalism is also modular in the training can be performed on small models, and these small models can then be used to construct larger models in which inference is performed. That is, we can often break down the task of training a large graphical model into a number of simpler subproblems.

This thesis is concerned with making some progress on the problem of automated polyphonic multi-instrument music transcription. By multi-instrument transcription, we mean a system capable of listening to a recording in which two or more instruments are playing, and identifying the notes played by each instrument.

We focus on the audio to piano roll transcription problem. We do not consider the piano roll to score problem. The latter problem, while interesting and worthy of research, involves creating scores that satisfy a musician's subjective ideas of what constitutes a "musically reasonable" score. By focusing only on audio to piano roll transcription, we can make use of more objective measures of transcription performance.

Our transcription system models both the musical signal behavior as well as some musical structure. We use an instrument-specific timbre model for musical signals that models both the spectral behavior and the variation in the overall sound intensity with time within a note event. We focus more on the signal modeling approach and less on the modeling of musical structure. However, the graphical models formalism seems to be well suited to modeling both of these aspects of musical signals. We also illustrate the modularity of the graphical models approach by making some modifications to our multi-instrument transcription system to create a system for guitar transcription. We expect that modifying and extending our models to model more musical structure as well as more signal structure will be an interesting and somewhat straightforward endeavor for future researchers.

1.6 Outline of this thesis

In Chapter 2, we present a review of the existing literature on automated music transcription.

In Chapter 3, we present DGMs for multi-instrument polyphonic transcription. We discuss the issue of modeling instrument timbre and present an instrument timbre model for modeling both the spectral and temporal attributes of timbre.

In Chapter 4, we present DGMs for guitar transcription using a hexaphonic pickup. We discuss some modeling issues specific to guitar transcription.

In Chapter 5, we present a novel way of performing music transcription and musical parameter estimation using a non-negative matrix factorization algorithm.

The appendices contain derivations and some related material. Appendix A shows the derivation of M step updates for the key DGM in Chapter 3.

Appendix B present a Gaussian process model for harmonic musical signals that can be used to estimate the spectral envelope.

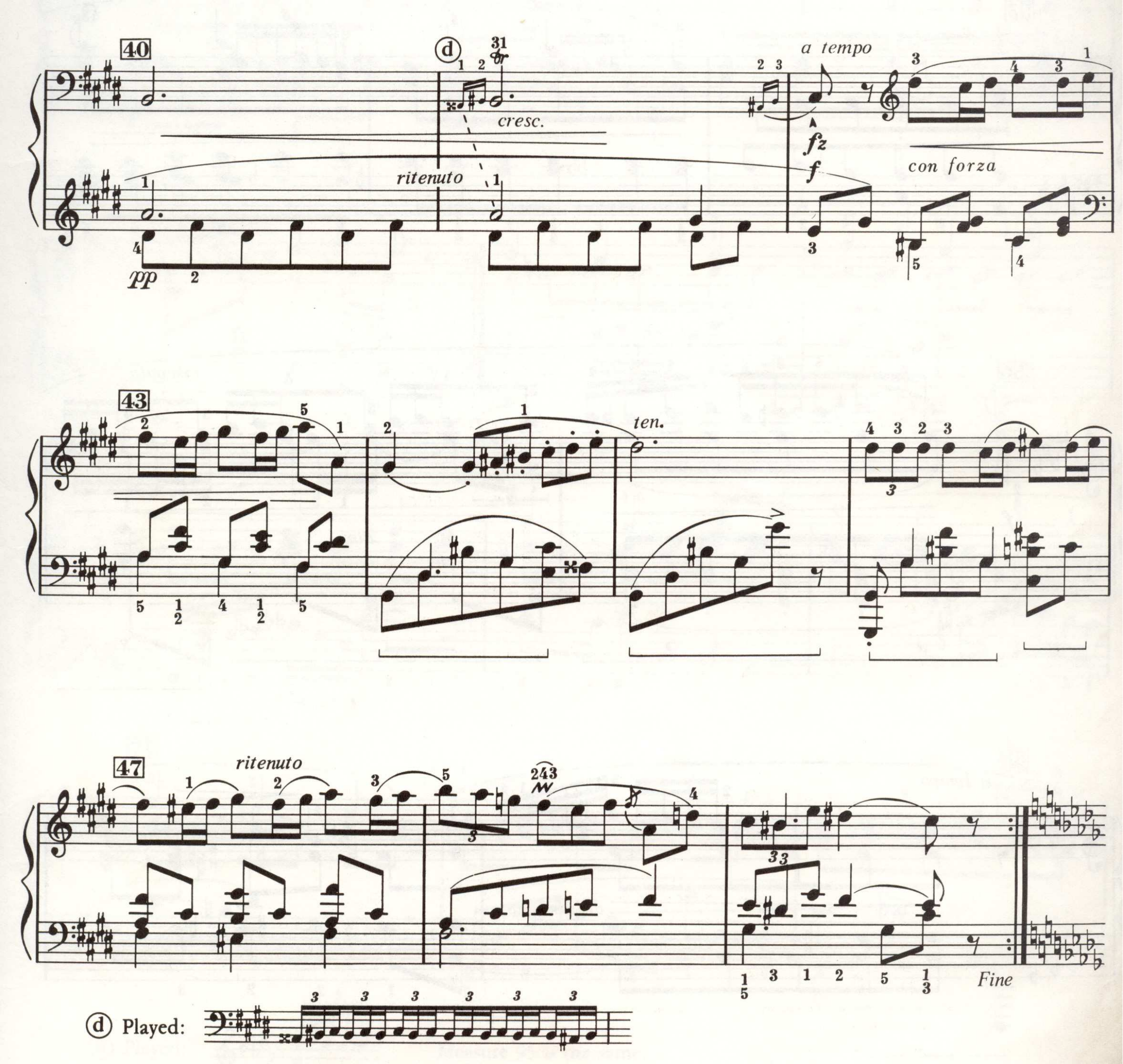

Figure 1.2: An example musical score using traditional standard music notation. This is an excerpt from Frederic Chopin’s Polonaise No. 1 in C-sharp minor.

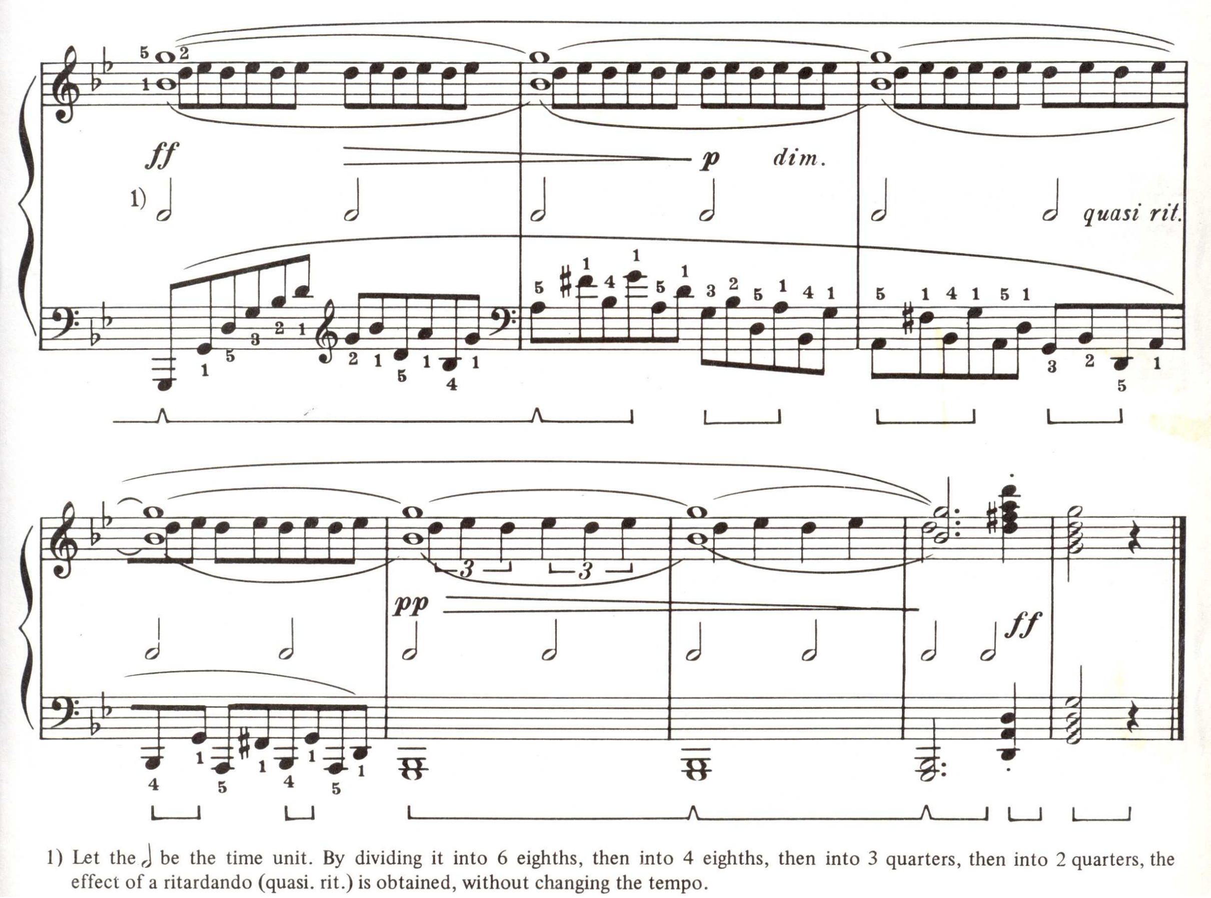
ff
p
dim.
1)
quasi rit.
pp
ff
1) Let the 𝅗𝅥 be the time unit. By dividing it into 6 eighths, then into 4 eighths, then into 3 quarters, then into 2 quarters, the effect of a ritardando (quasi. rit.) is obtained, without changing the tempo.

Figure 1.3: An excerpt from Brahms Rhapsody for piano in G-minor. Here, the composer explicitly represents retardando by using successively longer duration note symbols in the score.

Chapter 2

Literature Review

2.1 Introduction

In this chapter, we review the recent work related to the transcription problem. We would ideally like to compare the performance of the various approaches in the literature. However, the transcription problem is very broad, and researchers often work on non-overlapping problems, making objective comparisons difficult. Also, an objective performance measure may not even be reasonable for some interesting subproblems, such as audio to score transcription or piano roll to score transcription. For these reasons, we will avoid discussing the relative performance of the various approaches. Instead, our goal in this chapter is to provide an overview of the recent modeling approaches and types of problems being researched.

2.2 Audio to score transcription

In the audio to score transcription problem, the goal is to take the audio signal as input and produces the underlying score in standard musical notation. The score contains symbols representing the notes that were played and rests, with their durations expressed in fractions of a beat. The score also can contain information such as key signature, time signature, measure boundaries, expressive performance instructions, dynamic level, tempo, phrasing, and instructions for expressive performance, for example.

A goal in audio to score transcription is to produce a score that is both readable by a musician and that is also a musically reasonable representation of what was performed. We note that this is not a well-defined problem, since it is not possible in general to undo a performer's expressive performance and arrive back at the original score. However, musicians would argue that there are certainly musically reasonable ways of transcribing expressively performed musical pieces.

At the present, all of systems in the literature that take an audio recording as input and produce a score in traditional notation involve generating a piano roll score as an intermediate step. Thus, all of the published work on the audio to score problem can conceptually be broken done into two systems: a system that performs audio to piano roll transcription followed by a system that performs piano roll to score transcription. For this reason we will discuss the audio to score literature in the following sections under the appropriate subproblem.

2.3 Audio to piano roll transcription

The audio to piano roll problem involves taking the audio signal as input and producing a piano roll score. Although a piano roll score contains symbolic note events, it typically contains no musicological information other than constraining the pitches of the transcribed notes to be contained in some set of allowable pitches (e.g., the equal-tempered 12 tone scale). That is, unlike a traditional score, a piano roll score has no notion of key signature, musical meter, measure boundaries, tempo, etc.. The note durations are not quantized into musically meaningful durations (such as fractions of a beat).

Most of the recent literature audio to piano roll transcription focuses on the case of polyphonic transcription. All of the work we have come across on polyphonic transcription is for the case of melodic instruments only (no percussive instruments). Also, instrument identification is not performed. The transcription work that focuses on instrument identification constrains the input recording to be monophonic.

2.3.1 Monophonic audio to piano roll transcription

There is a large literature on algorithms for robust monophonic pitch tracking of speech and musical signals [Hes91] [Tal95] [dCK02] [SLIL03] [BXM04]. However, monophonic transcription is still considered to be a difficult problem. This is because some knowledge of instrument signal properties is required in order to identify symbolic note events. Transcription systems can also benefit from musical structure knowledge.

A recent system that combines rule-based algorithms and probabilistic models was proposed by Ryynanen and Klapuri [RK04]. Their system performs monophonic transcription. They presented results for recordings of male and female singers. Their system first performs musical feature extraction using rule-based algorithms to extract pitch, voicing, accent, and meter estimates from the input signal. The second stage of the system consists of two probabilistic models: a note event model and a musicological model. The note event model is represented as a HMM. The HMM only models transitions within a note event. Transitions between note event HMMs are modeled by the musicological model, which uses a token passing algorithm to infer the most likely state settings. Thus, in their system, the note pitch, voicing, accent, and meter are assumed known and are used as features for the two probabilistic models, which are then used to infer note onset and offset events.

2.3.2 Polyphonic pitch tracking

Although robust algorithms exist for monophonic pitch tracking, the problem of polyphonic pitch tracking of musical signals is still considered to be a challenging problem. We now give an overview of some recent approaches to this problem.

Goto

Goto [Got04] proposed a system for two-voice pitch tracking and presented results for CD recordings of classical and popular music. His system modeled a single time slice of the spectrogram as a probability distribution. Goto proposed a weighted Gaussian mixture model for the contributions of instrument tones of all possible fundamental frequencies. An instrument tone is modeled as a weighted sum of Gaussian centered at the harmonics of the fundamental. His system attempts to find the predominant fundamentals in the spectrum by minimizing the Kullback-Leibler divergence between the observed spectrum (interpreted as a pdf) and his tone model. Manually specified non-overlapping pitch ranges for the melody and bass lines are used. The system attempts to find the fundamental that is most predominant in each pitch range.

Saul

Saul [SS05] proposed a system for polyphonic pitch tracking that uses a nonnegative matrix factorization (NMF). His system is related to the NMF pitch tracker presented by Brown in [SB03] and is also related to our work in Chapter 5.

Bilmes

Bilmes [BXM05] presented a dynamic graphical model (DGM) for two-voice pitch tracking. His system models the pitch and formants of speech signals. In his system, the fundamental and formants are represented by latent variable in a DGM.

Davy et al.

Davy and Godsill [DG02] proposed a polyphonic pitch tracking system that models musical signal behavior in the time domain. No musical structure (or *contextual* information, as they refer to it) is modeled. In their system, the parameters of a sinusoidal signal model are modeled as random variables. Various smoothness and other regularization constraints are imposed on the random variables corresponding to the sinusoidal model parameters. The posterior distribution is estimated using MCMC techniques. They reported good results for up to 3 note polyphony on saxophone and trumpet sounds.

Smaragdis et al.

Smaragdis and Brown proposed an algorithm for polyphonic piano transcription. Their algorithm is is based on a nonnegative matrix factorization (NMF) of the spectrogram.

Their work is similar to our work in Chapter 5. Although they refer to their system as performing transcription, we consider their system to perform polyphonic pitch tracking, using our definition from Chapter 1. The output of their system consists of a set of "spectral basis vectors" and corresponding intensity versus time vectors. Some further processing is required in order to produce an output consisting of labeled note events, however. An obvious and simple way to obtain a transcription is to perform thresholding of the intensity vectors and manual labeling of the spectral basis vectors. We note that the NMF approach is a purely signal based approach. No musical structure is modeled. We implemented a similar system in Chapter 5 and observed that the transcription results are good considering the simplicity of the NMF algorithm.

2.3.3 Polyphonic audio to piano roll transcription

Polyphonic audio to piano roll transcription is a challenging problem since it requires some knowledge of instrument signal properties and would also benefit from some knowledge of musical structure. There appears to be nothing in the literature on polyphonic piano roll transcription with instrument identification other than our work in [VJW05].

In this section we review some of the recent approaches to audio to piano roll transcription. The earliest published work on polyphonic transcription is Moorer's 1975 PhD thesis [Moo75]. Moorer presented some results for piano and guitar recordings with two-voice polyphony. A detailed review of the older literature (from 1975 through 2001) can be found in [Hai01]. A common feature of much of the work from this period is the use of a time-frequency representation of the input signal, followed by various rule-based algorithms for performing peak finding and assigning note events. The more recent work is characterized by an increasing use of probabilistic models, although rule-based approaches are still common.

Raphael

Raphael [Rap02] used an HMM approach for transcription of piano music. The states of the HMM correspond to piano chords. Raphael proposed various simplifications in order to achieve a tractable model. He presented results from a Mozart Sonata and also made re-synthesized transcriptions available on his web site.

Klapuri

Klapuri [Kla04] used an approach in which the input recording is first converted to a time-frequency representation. He then used a polyphonic pitch tracking algorithm that operated by looking for harmonic structure in the spectrogram, canceling it, and iterating. Note events were estimated by using the pitch tracker in combination with an onset and meter estimation system [Kla99]. His system was primarily rule-based, but also used some probabilistic modeling.

Klapuri presented objective results for single-instrument polyphonic transcription on both MIDI synthesized and acoustic instrument recordings. Percussive instruments were allowed in some of the input recording but were not modeled and transcribed by his system.

Unlike most other researchers, Klapuri chose to make listening examples of his results available on his web site. He provides both the input sound file and the re-synthesized transcription for a variety of classical and popular music examples.

Cemgil et al.

Cemgil et al. [CKB04] presented a dynamic graphical model (DGM) for the transcription of piano music. Their approach is notable in that a time-domain sinusoidal model is used. Their system makes use of a damped sinusoid note event model. Their system is one of the few approaches in the literature that avoids rule-based methods in favor of a statistical model of instrument signal properties. They make use of a DGM that is a special case of the switching Kalman filter [Mur02]. Exact inference is intractable, so approximate inference techniques were employed.

2.4 Piano roll to score transcription

The goal of piano roll to score transcription is to take the piano roll score as input and produce a traditional score as the output. This problem involves the same musical issues as audio to score transcription (e.g., undoing expressive performance), but the properties of the acoustic signal do not come into play.

2.4.1 Cemgil and Kappen

Cemgil and Kappen [CK03] presented a probabilistic generative model for tempo tracking and rhythm quantization in expressively performed music. Their system takes a piano roll score as input. They propose a probabilistic model for timing deviations in expressive musical performances. Specifically, they make use of a DGM in the form of a switching Kalman filter. Their model objective is formulated as a maximum a posteriori (MAP) state estimation problem. Since exact inference is intractable for their class of models, approximate inference techniques are used. They compared the performance of several MCMC methods. They presented results on some popular music.

2.5 Related work

2.5.1 Instrument identification

In this section, we review two approaches to instrument identification. Note events are not transcribed, however.

Martin

Martin [Mar98b] [Mar98a] worked on the problem of instrument identification in monophonic musical signals. Martin proposed a set of acoustic features for instrument identification. He used statistical pattern recognition techniques for classification using the acoustic feature vectors. In his work, the input consisted of monophonic single instrument recordings, which his system then attempted to classify. His system did not identify the notes present in the input.

Eggink

Eggink and Brown [EB04] described a system for instrument identification in polyphonic music. Their system attempts to identify a single instrument (the solo instrument) in a piece with piano or orchestral accompaniment. Their system relies on the empirical observation that the solo instrument often has a higher dynamic level than the accompanying instruments. Their system takes a recording of an entire musical piece as input and then makes a decision as to whether or not the instrument is present in the recording. No attempt is made to determine the notes that are sounding.

In their system, rule-based algorithms are used to perform peak finding on the time slices of the spectrogram and to locate the most prominent fundamental. With this information, the corresponding harmonic amplitudes are then estimated and input to an instrument classifier. Thus their system relies on a spectral model of instrument timbre. There is no notion of a note event model, so timbre components such as dynamic intensity versus time are not modeled. Their system is mostly rule-based, but includes a Gaussian mixture model classifier.

2.5.2 Automated meter estimation

A subproblem in piano roll to score transcription is the estimation of musical meter, also referred to as *rhythmic parsing*. The input can consist either of an audio file or a piano roll score such as a MIDI file. The goal in meter estimation is to identify the tempo (beat rate) measure boundaries, beat, and tatum locations. The *beat* or *tactus* refers to the basic time unit of a musical piece, corresponding to the “foot tapping” rate. The *tatum* (*temporal atom* [Bil93]) refers to the shortest-duration notes that occur in a piece. All notes are taken to have a duration that is an integer multiple of the tatum.

Lerdahl and Jackendoff proposed a rule-based system for meter estimation in [JL83]. However, they did not provide an implementation.

In [KEA06], Klapuri presented a system for meter estimation using both rule-based algorithms and probabilistic models. His system consisted of a time-frequency analysis of the input signal followed by probabilistic model for pulse periods. He proposed rule-based algorithms for representing musical knowledge that operate on the filtered time-frequency image as well as on the output of the probabilistic model. He claimed good results for a variety of musical examples.

Chapter 3

DGMs for polyphonic multi-instrument music transcription

3.1 Introduction

In this chapter, we present dynamic graphical models (DGMs) (also known as dynamic Bayesian networks (DBNs) [Mur02]) for multi-instrument polyphonic audio to piano roll transcription. The instruments are assumed to be melodic. We do not consider percussive instruments, such as drums. Given an input sound file, our system attempts to identify the notes that were played as well as the instruments that played them. Existing research on automated transcription has focused mainly on the problem of single-instrument polyphonic transcription. Although the input sound file may contain multiple instruments, the output of existing transcription systems lacks instrument labels associated with the note events. The key novel contribution of this chapter is a transcription system that performs joint polyphonic transcription and instrument identification. A key feature of our model is the use of a note-event timbre model that includes both a spectral model and a dynamic intensity versus time model (i.e., a time envelope model).

We have tried to produce a transcription system with the following in mind:

1. The modeling assumptions and objectives should be clear and well defined.

2. Ad hoc, rule based algorithms should avoided since they tend to lead to systems that are difficult to extend.

3. The number of parameters that must be specified manually should be kept small.

4. Any preprocessing or post precessing should be well motivated.

5. The system model should incorporate both musical signal knowledge and musical structure knowledge.

6. The system should be capable of learning from training examples.

7. The system should be straightforward to extend to model more complex musical structure and signal behavior.

We hope that Chapter 1 gave the reader some appreciation of the difficulties involved in formulating the transcription problem. When humans transcribe music, they rely on subjective notions of what is musically reasonable. The corresponding underlying mechanisms are not well understood and can be difficult to formulate in objective terms. There are a variety of interesting subproblems in transcription, and for any particular problem, it may not immediately be clear what should be modeled or in what amount of detail.

In our work, the emphasis has been on a model-based approach, with less of an emphasis on an algorithm-centered approach. We have tried to formulate a transcription system in terms of a clear model objective. This is in contrast to much of the previous work in the literature, in which the transcription problem is often formulated in terms of rule-based algorithms. Although systems that make heavy use of rule-based algorithms can often perform well, such systems tend to also be difficult to interpret and to extend.

When humans transcribe music, musical signal properties as well as musical structure knowledge are used. Our transcription system therefore models both the musical signal properties in the spectrogram as well as musical structure. We take musical signal properties to mean properties of musical instrument sounds that depend on the physics of sound production and not on a particular musical genre. Examples of signal properties include the location of partials as a function of pitch (harmonic vs. inharmonic, etc.), spectral envelope, transient onset noise, and variation in sound intensity after the note onset. Most musical instruments tend to produce notes characterized by a sudden increase in intensity at the onset, followed by a somewhat regular pattern of intensity variation as the note continues to sound, until the note offset. For some instruments, such as the piano and other keyboard instruments, the pitch does not change as the note sounds. For other instruments, depending on the playing style, there can be some expressive variation in pitch as the note sounds.

We take musical structure to mean attributes of a recorded musical performance that are specific to a particular musical genre or piece. Examples include key signature, musical meter, tempo, harmonic structure (chord progressions), rhythmic structure, and melody.

We make use of an explicit note event model that is defined in the context of a DGM. This allows us obtain the transcription by performing inference in the DGM. Formulating our note event model in a DGM makes our modeling objectives clear and concise. In our system, the output transcription is defined as the mode of the posterior of a DGM. This is in contrast to the typical approach in which note events are estimated by applying thresholding or other ad hoc postprocessing to the output of a polyphonic pitch tracker. The modularity of the graphical models approach means that we can often make changes to one component of a model while keeping the other components unchanged.

This chapter is organized as follows: In order to motivate our note event model, we first review the sinusoidal model of musical signals in Section 3.2. In Section 3.3 we briefly review and motivate time-frequency representations for musical signal analysis. In Section 3.4 we present a spectrogram model of musical signals based on the time-frequency representation of the sinusoidal model. In Section 3.5 we discuss ways of measure transcription performance. In Section 3.6 we give a brief overview of graphical models formalism that we use for modeling the transcription problem. In the remaining sections we will present dynamic graphical models of generally increasing complexity for pitch tracking and transcription systems.

By gradually increasing the model complexity we will see how components of simple models can then be reused in more complex models. Our main interest in this work was to create a transcription system that is capable of performing polyphonic transcription with instrument identification. We present a DGM for such a system in Section 3.12 for the case of two instruments. This DGM is also presented in [VJW05]. We present some quantitative results for two-instrument transcription using this DGM.

3.2 Sinusoidal modeling of musical signals

Our note event model is based on the *sinusoidal model* (also known as the *additive synthesis model* or *sinewave model*) [Ste99] [Goo97]. The sinusoidal model is useful because it provides a compact representation of musical sound that captures perceptually relevant attributes such as sound intensity, pitch, and spectral structure. Our interest with this model is in finding a reasonable simplification with a small number of parameters that models attributes of timbre sufficient for instrument identification.

We observe empirically that the sounding of a note by a melodic musical instrument tends to be well-modeled as the sum of a small number a sinusoids with time-varying amplitudes. The sinusoid amplitude functions tend to vary slowly with respect to the period of the fundamental. For many instruments, a sinewave model with anywhere from a few sinusoids to around 20 sinusoids is able to capture most of the energy in the signal.

However, a sinewave model fails to capture some attributes of instrument sounds such as transient onset noise. For this reason, a sinewave plus noise model is sometimes used for instrument synthesis [Goo97]. However, instrument sounds synthesized using a sinewave model alone often still sound much like the original instrument to human listeners.

Let $x(t)$ denote the acoustic waveform of a single note event. In the expressions to follow, we will assume that all of the parameters can depend on the fundamental frequency f_1 of the note. However, will not explicitly denote this dependence in the expressions in order to improve readability. A somewhat general sinewave model for $x(t)$ is then given by

$$x(t) = \sum_{h=1}^{H} \alpha_h(t) cos(2\pi\theta_h(t) + \phi_h)$$

where

$$\theta_h(t) = \int_0^t f_h + f_{vibrato_h}(\tau)\, d\tau$$

H is the number of partials to model. $\alpha_h(t)$ is the time varying amplitude envelope for the h'th partial. ϕ_h is the initial phase of the h'th partial. ϕ_h represents the initial phase of the h'th partial. f_h is the frequency of the h'th partial, andf_1 denotes the fundamental frequency. $f_{vibrato_h}$ represents a possible small (with respect to f_h) and slowly time varying frequency component, corresponding to expressive frequency deviations such as vibrato.

Many melodic instruments are well-modeled as having harmonically related partials. Examples of such instruments include bowed string instruments, horns, the human voice, wind instruments, and pipe organs [FR91]. In this case, the expression for $\theta_h(t)$ in the sinewave model becomes

$$\theta_h(t) = \int_0^t h(f_1 + f_{vibrato}(\tau))\, d\tau$$

If we also ignore vibrato, then the expression for $x(t)$ simplifies to

$$x(t) = \sum_{h=1}^{H} \alpha_h(t) cos(2\pi h f_1 t + \phi_h)$$

However, for some instruments, we may be interested in explicitly modeling the inharmonicity. Instruments such as the plucked string instruments, and particularly the piano, can have more significant inharmonicity. For example, in the case of the piano, the first several partials are very close to harmonic, but the higher partials tend to become “stretched.” In [FR91] an approximation for the inharmonicity I_h for the piano is given as

$$I_h \equiv f_h / h f_1 = B h^2$$

where f_1 is the fundamental frequency, and B is the *inharmonicity constant*. The value of B tends to vary with note number and a table of typical values can be found in [FR91].

In addition to the harmonic partials assumption, we can also simplify the sinewave model by observing that the amplitudes of the various partials tend to vary together (*amplitude comodulation* [Bre90]). We might therefore consider making the simplifying assumption that the relative amplitudes of the various harmonics do not vary significantly during the playing of a note. In this case we can model each of the $\alpha_h(t), n = 1..H$ as a single harmonic amplitude weight α_h that is scaled by an overall note event *intensity envelope* $Intensity(t)$. We then have that $\alpha_h(t) = Intensity(t)\alpha_h, h = 1..H$. We can think of the harmonic amplitude weights $\alpha_h, h = 1..H$ as defining a *harmonic template* for a particular instrument at a particular fundamental frequency, f. Under this modeling assumption, the sinewave model becomes

$$x(t) = Intensity(t) \sum_{h=1}^{H} \alpha_h cos(2\pi h f_1 t + \phi_h) \tag{3.1}$$

The phase parameters ϕ_n are important if one is interested in modeling the time domain signal $x(t)$ directly. However, since the human auditory system is relatively insensitive to phase information, the ϕ_n parameters are not perceptually important. The ϕ_n parameters can therefore be ignored if our goal is timbre modeling or synthesis. In this case, the sinewave model simplifies to

$$x(t) = Intensity(t) \sum_{h=1}^{H} \alpha_h cos(2\pi h f_{pitch} t) \tag{3.2}$$

Our note event model is based on the sinewave model in Equation 3.2. We make the modeling assumption that the parameters $\{\alpha_h, h = 1...H\}$, which can vary with note pitch f_1, contain sufficient timbre information for discriminating between instruments. Depending on the particular instruments this may or may not be a valid assumption. However, our empirical observations suggest that it is at least valid for instruments with significant perceptual differences in timbre, such as the guitar and trumpet, for example. We present a model for the note event intensity envelope $Intensity(t)$ in Section 3.9.2. We make use of this model in the DGM in Section 3.12 and also in the guitar DGM in Chapter 4.

For many instruments, the timbre tends to become less bright as a note sustains, however. Quantitatively, this is caused by a relative decrease in the amplitudes of the higher partials. In Chapter 5, we propose an analysis method that could be used to extract parameters for a sinewave model in which the brightness can change within a note event.

3.3 Time-frequency representation of musical signals

The time domain acoustic signal is sometimes modeled directly, as in [CK03, DG02]. However, a time domain representation does have some potential disadvantages in the context of musical signal analysis:

1. The shape of the time domain waveform can have little relation to its perceived timbre. For example, by changing the phase information, it is possible to generate several acoustic waveforms that look very different, yet would be judged as sounding the same or very similar by human listeners.

2. Extracting perceptually important parameters from the time domain signal $x(t)$ by fitting it to a sinewave model requires that we model the phase information as well.

In the literature on music transcription, it is common to perform some kind of preprocessing on the acoustic signal to obtain a more suitable representation for analysis. Recall from the sinusoidal model in Section 3.2 that melodic musical signals are modeled as consisting of a sum of a small number of sinusoids with amplitudes that tend to vary slowly with respect to the pitch period.

The short-time Fourier transform (STFT) [OS89] is a commonly used time-frequency representation. The STFT is obtained by applying a window function $b(t)$ to successive overlapping fragments of the input signal $x(t)$. The Fourier transform is then applied to the windowed input signal fragments, resulting in a series of time-localized frequency representations.

The continuous-frequency STFT of a continuous-time signal x(t) is given by

$$X_i(f) = \int_{-\infty}^{\infty} x(iL + t)b(t)exp(-j2\pi ft)\,dt \tag{3.3}$$

where i is the time slice number, and $L \leq length(b(t))$ is the hop length. A Hamming window is often used for $b(t)$. Since the phase information is not perceptually relevant, a magnitude representation is typically used. A magnitude time-frequency representation is often

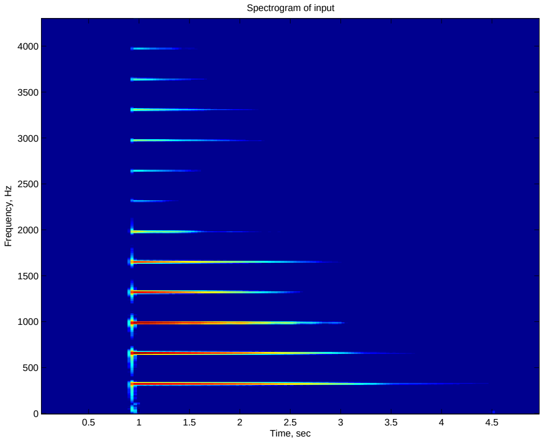

Figure 3.1: A spectrogram of a guitar waveform.

referred to as a *time frequency image* in the literature. The magnitude of the STFT,$|X_i(f)|$, is referred to as the *spectrogram*.

In order to implement the STFT on a computer, however, both time and frequency must be made discrete. We sample $x(t)$ uniformly in time to obtain the discrete time signal $x(n)$. The STFT at time slice i and frequency bin k is given by

$$X(k,i) = \sum_{m=0}^{M-1} x(iL+m)b(m)exp(-j(2\pi/M)km), 0 \leq k \leq M-1 \tag{3.4}$$

where M is the DFT size. In an actual implementation, it is common to choose $M = 2^p$ where p is a positive integer so that the FFT algorithm can be used for computational efficiency. For music signals, typical parameter choices are a sampling rate of 44.1 kHz, and M = 2048 or 4096. Typically the hope length is taken to be $L \leq M/2$. Figure 3.1 shows a spectrogram of a guitar waveform. Figure 3.2 shows a guitar spectrum, which corresponds to a time slice of the spectrogram shortly after the note onset. The bumps in the spectrum correspond to the harmonically related sinusoids in the time domain.

3.4 Time-frequency interpretation of the sinusoidal model

Consider one time slice $X_i(f)$ of the STFT in Equation 3.3. The spectral representation of a short-time sinusoid consists of the Fourier transform of the window function shifted to

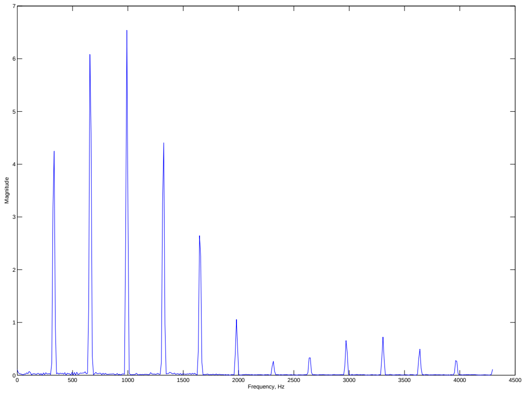

Figure 3.2: Spectrum of a guitar waveform. This is a time slice of the spectrogram of Figure 3.1 shortly after the note onset.

the frequency of the sinusoid. We assume that the window function $b(t)$ is real and even so that its Fourier transform $B(f)$ will also be real and even. We assume that $b(t)$ is of sufficient length to allow the lowest-frequency sinusoids in the input signal to be resolved. We also assume that the parameters of the sinusoidal model (Equation 3.1) can be assumed constant over the duration of $b(t)$.

We first consider the case where a single instrument k is present in the input signal. Under the harmonic sinusoidal model, the spectrum then consists of a series of bump functions that are uniformly spaced in frequency. The magnitude spectrum of instrument k at time slice i is therefore given by

$$|X_i(f)| = Instrument_i^k(f) = Intensity_i^k \sum_{h=1}^{H} \alpha_h^k(f_{pitch}) B(f - hf_{pitch}) \tag{3.5}$$

Now let us consider the multi-instrument case. For modeling simplicity, we assume that the magnitude spectrum of a mixture of instruments is given by the sum of the magnitude spectra of the individual instruments. Since superposition is not linear in the magnitude spectrum, the above model will not hold at harmonics common to both pitches. However, we feel that it is reasonable for the case where the input signal has at most two pitches sounding simultaneously. For the case of two instruments, we then have

$$|X_i(f)| \approx Instrument_i^1(f) + Instrument_i^2(f)$$

When the STFT is implemented on a computer, the discrete-time, discrete-frequency STFT of Equation 3.4 is used. A spectrogram time slice gives us samples of $|X_i(f)|$ at the uniformly spaced frequencies $f_i, i = 1...M$ corresponding to the M spectrogram frequency bins.

3.4.1 Modeling assumptions and constraints on the class of input signals

In this section, we state our modeling assumptions and the constraints that we place on the input musical signal. These constraints are intended to reduce the complexity of the transcription problem to a more manageable level. We make the following modeling assumptions:

1. All notes in the musical signal come from a discrete pitch space. We further assume that all notes come from an equal-tempered 12 tone scale and the all instruments are in-tune (using A = 440 Hz). We do not model vibrato or other expressive changes in pitch during the playing of a note.

2. The polyphony of the musical signal is limited to two simultaneous notes.

3. The musical signal contains at most two distinct instruments.

4. The musical instruments produce melodic sounds with either harmonic partials or known inharmonicity (sounds where the partials locations are known, given the pitch). We do not model percussive instruments, such as drums.

5. The musical instruments produce sounds that have unchanging brightness as a note sustains. That is, the relative partial intensities are constant within a note event, but can vary as a function of pitch and instrument. The sound intensity within a note event after the note onset can either remain constant or gradually decay as the note sustains.

6. Except where otherwise noted, all musical signals are synthesized from a MIDI file using a wavetable synthesizer. This is done so that we can objectively measure transcription performance by comparing the input true score (input MIDI file) to the transcription (output MIDI file). A wavetable synthesizer performs synthesis by playing back a recording of an actual acoustic instrument, so as a result, musical signals synthesized from MIDI files often sound somewhat similar to recordings made from acoustic instruments. In Chapter 4, all the musical input signals are recorded from actual guitars.

Some of these modeling assumptions might seem rather restrictive. However, the input musical signal does not have to strictly conform to our modeling assumptions in order to achieve good transcription performance. This is because our goal is not to synthesize instrument sounds, but rather to discriminate between instruments and pitches. Therefore, it is only required that we model the differences between instruments and note pitches sufficiently well to discriminate between them. The relative harmonic magnitudes (the spectral envelope), and the sound intensity envelop are attributes that do tend to vary significantly between instruments [FR91].

We also note that good transcription performance is still sometimes possible even when the sound signal contains expressive performance attributes such as vibrato. We present results for (synthesized) violin sounds that are played with a small amount of vibrato and observe that good transcription performance can still be achieved. However, for the case of very expressive performances or discriminating between similar sounding instruments in a performance, a different or more complex transcription model may be needed.

3.5 Measuring transcription performance

Recall from our definition of audio to score transcription in Chapter 1 that the audio to score transcription problem involves the task of undoing expressive performance to arrive back at the underlying score. Given enough musical knowledge, it becomes possible to create a “reasonable” score, where we define a reasonable score as one that musicians would agree makes an appropriate tradeoff between human readability and accurately representing performance details. Of course, this is not a definition that one could use to objectively measure the performance of an automated transcription system, since it is defined in terms of subjective notions of what is musically reasonable. We also note that for a given performance, there may exist several distinct scores that could be judged as musically reasonable. It is not immediately clear how one might define a reasonable objective measure of performance for such a problem. This raises the question of whether an objective measure of performance is even be appropriate for such a problem.

However, our work focuses on the problem of audio to piano roll transcription. This problem is perhaps better suited to an objective measure of performance since it focuses more on the signal properties of the input signal. In the literature on audio to piano roll transcription, it is therefore common for researches to present their results using an objective measure of performance.

For the problem of audio to piano roll transcription, MIDI files provide a convenient way to objectively measure transcription performance. An input MIDI file can be taken to be the "true" piano roll score. The MIDI file is then synthesized to an audio file using a high quality wave table synthesizer. A wave table synthesizer performs synthesis be playing back a recording of an actual acoustic instrument. Therefore, for a restricted class of musical input signals where expressive performance is not a requirement, synthesized MIDI file performances can sometimes sound very similar to an actual acoustic instrument performance. We also point out that if an automated transcription system performs poorly on MIDI synthesized performances, there is little hope that the transcription performance will be good for an actual acoustic performance. For these reasons, we mainly use MIDI synthesized performances for our work in this chapter. We present results for acoustic instrument recordings in Chapter 4.

We now have the true score and a corresponding sound file. The sound file is given as input to the transcription system, which then transcribes the sound file to an output piano roll score, such as a MIDI file. An objective measure of transcription performance is then obtained by comparing the input MIDI file to the output MIDI file (transcription) using an error rate measure analogous to the error rate used in automated speech recognition systems.

We take this approach and define the percentage *transcription error rate* as

$$100\frac{\text{Insertions + Substitutions + Deletions}}{\text{Total Notes in Score}}$$

This is a commonly used measure for the error rate for polyphonic single-instrument audio to piano roll transcription (e.g., [Rap02]). However, since we have not come across a similar error rate defined for multi-instrument transcription, we extend it as follows for the multi-instrument case. The insertions, substitutions, and deletions are computed separately for each instrument and then summed. Therefore, an instrument misclassification of a note onset event counts as one insertion and one deletion.

We present objective transcription results for several input recordings under various modeling assumptions. We do not compare the performance of our system against other published transcription systems, however. This is because we are not aware of any other transcription systems that are capable of multi-instrument polyphonic transcription, even for only two voices.

An objective measure of transcription performance can be useful in comparing models. However, the human auditory system is complex and we perceive music in ways that are hard to quantify objectively. For example, some pitch and rhythm errors may be perceived as more serious than others, although they might might be considered equal under the above error rate measure. Incorporating musicalogical knowledge into a transcription system might cause it to make more musically reasonable errors (similar to a human). However, the above measure of error rate penalizes all errors equally. One solution to this problem might consist of proposing a more complex error rate measure that attempts to take musicalogical structure and human auditory perception into account. However, there seems to be no widespread agreement in the transcription community on how to go about doing this.

Therefore, we feel that another useful way of presenting transcription results for model comparison purposes is to synthesize the transcription to an audio file. Listeners can then judge transcription performance themselves by listening to both the input audio file and the synthesized transcription.

3.6 Graphical models

The graphical models formalism provides for families of probability distributions to be modeled in a concise and intuitive manner. The graphical models formalism is used to represent latent variable probabilistic models. The modeling objective is typically specified in terms of a solution to an inference problem. A nice feature of the graphical models formalism is that it decouples the modeling objective of a problem from the algorithm used to solve the objective.

A *graphical model,* short for *probabilistic graphical model,* represents a class of probability distributions with a factorization given by the graph. In a graphical model, a *node* represents a random variable. An unshaded node denotes a *latent* or *hidden* random variable and a shaded node denotes an *observed* random variable. An *edge* or *arc* represents a conditional dependence relation. Informally, an arc can be thought of as implying causation between the parent and child nodes.

To apply the graphical models formalism to a particular data analysis problem, we first decide on a particular graph structure and corresponding conditional probability distributions. In some cases, we may be interested in parameter learning, which involves obtaining point estimates for some of the parameters in the model given data. We may also be interested in performing posterior inference, which involves computing the probability of some subset of the hidden nodes given the observed nodes.

Inference and learning algorithms exist that can be applied to general graphical models. There is a large literature on junction tree algorithms for exact inference as well as algorithms for approximate posterior inference. Depending on the graph structure, exact inference is sometimes intractable, but progress is being made on approximate inference algorithms. The graphical models formalism is also modular in the training can be performed on small models, and these small models can then be combined to form larger models in which inference is performed. The graphical models formalism therefore allows us to break down the task of training a large graphical model into a number of simpler subproblems.

We work with a specific class of graphical model, called a dynamic graphical model (DGM), also known as a dynamic Bayesian network (DBN) [Mur02]. DGMs are useful for modeling sequential data. A DGM is a graphical model in which a subgraph is repeated over time, with directed arcs from each time slice to the next. Figure 3.3 shows an example of a DGM.

3.6.1 Our approach to transcription using dynamic graphical models

Our approach in this chapter is to construct a latent variable dynamic graphical model that assigns a probability to a spectrogram. The latent variables represent musical quantities of interest, such as the pitch and intensity state for each time slice of the spectrogram. We take the transcription to consist of the settings of the latent variables that maximize the probability of the spectrogram data. However, in order to interpret the hidden states as symbolic note events, we must have a model that contains explicit states corresponding to note onsets and offsets. It will also be necessary to place various constraints on the hidden states so that only physically realizable settings are possible. For example, the pitch should remain constant within a note event (from onset to offset).

We will illustrate how the graphical models formalism allows us to define our transcription system in a modular way. We specify a set of components, each consisting of a probabilistic model for a particular musical signal or structure property. We then combine the components to form an overall probabilistic model by specifying independence relations between the components. For example, we will propose a probabilistic interpretation of the sinusoidal model of melodic musical instruments in the spectrogram. One component of our transcription system will then assign a probability to a spectrogram time slice conditional on the latent intensity and pith state at the corresponding time slice. Another component of the model will define a pitch transition process that tends to favor small interval changes in note pitch. Yet another component will constrain the intensity and pitch state to evolve over time in a way that is consistent with our prior knowledge of the physics of instrument sound production. For example, if we a modeling a plucked stringed instrument, then we will constrain the intensity to gradually decrease after the initial string pluck and we will also constrain the note pitch to remain constant while the intensity decays.

The modularity of this approach makes it straightforward to obtain point estimates for the parameters using an EM-based estimation procedure. We can perform parameter learning on simple graphical models, and then combine these to form a more complex model in which we perform inference.

The modularity of the graphical models formalism also makes it straightforward to extend our model. For example, although we will present models and results for the two-instrument transcription case, our models extend immediately to the case of higher polyphony. It is straightforward to extend a portion of our model while keeping the rest of the system unchanged. For example, one might consider using a hierarchical HMM to model musical melody motifs or intensity envelopes with hierarchical structure. This could be achieved while making use of the same observation model and the same pitch transition model. In the following sections we present DGMs of gradually increasing model complexity. This will allow us to justify various modeling assumptions by comparing results under different assumptions. However it will also illustrate the modularity of our approach, as we will often be able to reuse model components across many DGMs.

3.7 A HMM for monophonic pitch tracking

Our primary interest in this thesis is in models for multi-instrument polyphonic pitch tracking. In order to motivate such system, we feel that it is instructive to first consider some systems that try to solve a simpler problem. We will then gradually increase the model complexity until we arrive at a system for multi-instrument transcription. This presentation approach will also allow us to illustrate the modularity of graphical models, by showing how model components can be reused in more sophisticated graphical models.

A simple but still musically interesting problem is monophonic pitch tracking. We begin by presenting a system for monophonic pitch tracking. In this section we present an HMM pitch tracker that illustrates our basic modeling approach that will be used throughout this chapter. The HMM pitch tracker allows both musical structure knowledge and musical signal knowledge to be jointly modeled. This pitch tracker may also be useful for monophonic pitch tracking and is easy to implement, given the wide availability of software for inference and learning in HMMs.

3.7.1 Model

Figure 3.3 shows a HMM for monophonic pitch tracking. The $Audio_t$ nodes represent the observed audio feature data at time t. The $Pitch_t$ nodes represent the instantaneous pitch at time t. Here, $Pitch_t$ is a multinomial random variable with each state denoting a note pitch on some discretized musical tuning (e.g., A, C. etc.).

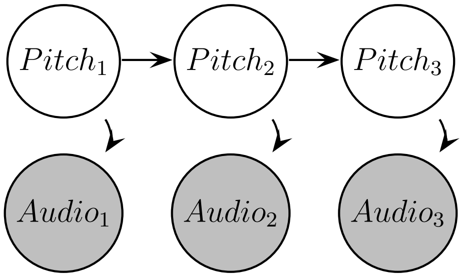

Figure 3.3: A HMM for monophonic transcription. Shaded nodes are observed audio data; unshaded nodes represent the hidden pitch state.

3.7.2 Pitch Transition Model

In music, small interval changes in pitch tend to occur more often than large interval changes. We incorporate this musical structure knowledge into a HMM by modeling the pitch as a hidden Markov process using a transition model that favors small pitch changes over large pitch changes. We refer to this as the *pitch line* model. The transition matrix can be constructed as follows: Let $Note(i)$ and $Note(j)$ denote the MIDI note number of pitch states i and j. For example, if i corresponds to A 440, then $Note(i)$ is 69. For each i and j, we set the $(i,j)th$ element of a K x K matrix T equal to the following Gaussian kernel evaluated at $Note(i)$ and $Note(j)$:

$$T(i,j) = \exp(-\frac{1}{2\sigma^2_{ptran}}(Note(i) - Note(j))^2)$$

The rows of T are then normalized to make the matrix stochastic. The resulting transition matrix depends on a single scalar parameter.

We can alternatively use a method for specifying the state transition probabilities inspired by Shepard's notion of the pitch helix [She82]. Perceived musical dissimilarity between two pitches is taken to be proportional to their Euclidean distance on the helix. We can easily construct a transition matrix based on the pitch helix as follows: Let $x(i)$ and $x(j)$ in R^3 represent the locations of pitches i and j in the space in which the helix is embedded, where i and j can range over the K possible pitch values. For each i and j, we set the $(i,j)th$ element of a K x K matrix T equal to the following Gaussian kernel evaluated at $x(i)$ and $x(j)$:

$$T(i,j) = \exp(-\frac{1}{2}(x(i) - x(j))^T C^{-1}(x(i) - x(j)))$$

The rows of T are then normalized to make the matrix stochastic. C is diagonal, with two of the parameters tied to maintain rotational symmetry about the helix. So, the resulting transition matrix depends on two scalar parameters.

Finally, we make a few notes regarding the choice of pitch transition model. Under the pitch line model, the probability of a pitch transition is a monotonically decreasing function the the difference between the note number of the pitches (or equivalently, the difference in log frequency). Whereas in the pitch helix model, the probability of a pitch transition generally decreases as with increasing pitch separation while tending to favor pitch transitions of approximately one octave. We will discuss the choice of pitch transition model more in Sections 3.7.4 and 3.7.5.

3.7.3 Observation Model

Our observation model is based on spectral model of a single melodic instrument with harmonic partials in Equation 3.5. We must assign a probability to the observed spectral data conditional on the pitch state. We use the following Gaussian process model for the spectrum at time slice t:

$$|X_t(f)| = Instrument_t(f) + \xi(f) \tag{3.6}$$

where

$$Instrument_t(f) = Intensity_t \sum_{h=1}^{H} \alpha_h(f_{pitch}) B(f - hf_{pitch}) \tag{3.7}$$

where $\xi(f)$ is a zero mean Gaussian noise process. We take $B(f) = \exp(-f^2/\sigma)$ where σ is chosen manually. This choice was made for modeling simplicity. We feel that this is a reasonable simplifying assumption since our transcription system only needs to discriminate between pitches and instruments. It is not necessary to produce an extremely accurate spectral model. In a system where a more accurate spectral model is required, such as in frequency-domain synthesis, we $B(f)$ can be well-approximated by computing an oversampled DFT of $b(n)$ [Goo97].

However, the audio data at time slice t is obtained from the STFT in Equation 3.4. A spectrogram time slice therefore gives us $Audio_t = [|X_t(f_1)|, ..., |X_t(f_M)|]^T$, corresponding to samples of the continuous-frequency STFT at the M uniformly spaced spectrogram bin frequencies.

Conditional on the hidden state pitch variable we then have the following Gaussian observation model:

$$p(Audio_t|Pitch_t) = \mathcal{N}(Audio_t|\mu(Pitch_t), \sigma_\xi^2 I)$$

where $\mu(Pitch_t) = [\mu_1, ..., \mu_N]^T$ and $\mu_i = Instrument_t(f_i)$.

Note that we are overloading our notation here so that f_i now refers to the frequency of the i'th spectrogram bin, not the i'th partial.

Since the HMM does not have an intensity state, we fixed the $Intensity$ variable in Equation 3.7 to a constant. We added a “no pitch” state to allow the system to identify unpitched regions of the input signal. The “no pitch” state corresponds to a zero-valued spectral template ($\mu_k = 0$ for $k = 1...M$).

3.7.4 Experiments

The $\alpha_h(f_{pitch})$ harmonic magnitude parameters and observation noise variance σ_ξ^2 parameters are learned by an EM-based estimation procedure by training on monophonic single-instrument audio recordings. The harmonic width parameter σ^2 was chosen manually. The audio recordings consist of chromatic scales played over the entire pitch range of the instrument.

We implemented the HMM pitch tracker and observed its pitch tracking performance on a few short monophonic musical recordings. The output pitch was taken to be the mode of the posterior (Viterbi path). The choice of pitch transition model (pitch line vs. pitch helix) appeared to have little effect on performance. We therefore only present results for the case of the pitch helix transition model. The two scalar parameters of the pitch transition model were chosen manually. We present results for this model on guitar recordings in Section 4.5.

3.7.5 Remarks

We presented a HMM for monophonic pitch tracking of harmonic musical signals. The HMM modeled musical signal behavior in the spectrogram using a harmonic-partials model and musical structure using a pitch transition model. The HMM contained a single hidden pitch state variable.

The HMM system presented here performs pitch tracking, but not transcription, since note events are not identified. Also, the HMM does not model intensity. An explicit “no pitch” state was added to allow the system to identify unpitched regions of the input signal. However, a better solution might consist of a system that models intensity explicitly.

We would ideally like the pitch transition model to reflect the transition probabilities at the level of note events. However, in the HMM model the pitch transition model reflects the transition probabilities at the level of individual spectrogram time slices, rather than the transition probabilities between note events. When we attempt to learn the pitch transition model from musical input recordings, the resulting pitch transition matrices tend to have very small off-diagonal elements. In Section 3.12, we present a DGM that incorporates a pitch transition model for note events.

3.8 A FHMM for multi-instrument polyphonic pitch tracking

In this section we present a factorial hidden Markov model (FHMM) [GJ94] for multi-instrument polyphonic pitch tracking of two harmonic musical instruments. The instruments are individually modeled as monophonic. The key features of this model are the use of a learned instrument timbre model and a pitch state transition model to model musical pitch similarity. Our system learns the relative harmonic magnitudes as a function of pitch (fundamental frequency, f_{pitch}) for each instrument. This set of learned harmonic magnitudes indexed by f_{pitch} constitutes our instrument model. We do not model sound intensity. A FHMM that models intensity level will be discussed in Section 3.9.

3.8.1 Model

Figure 3.4 shows the decoding model. Each hidden Markov chain models the pitch evolution of a single monophonic instrument. The top hidden Markov chain models the pitch evolution of one instrument (e.g., piano) while the lower hidden Markov process models the evolution of another instrument (e.g., violin). The hidden state variables correspond to the discrete set of allowable pitch values. We take the pitch versus time output of our system to be the mode of the posterior (Viterbi path).

3.8.2 Transition Model

An advantage to using a FHMM over a regular HMM is the reduction in the size of the transition model state space, provided that the actual musical instruments are reasonably well modeled as having independent pitch processes. If each instrument is capable of playing K different pitches, then a regular HMM would require a K^2 x K^2 transition matrix. However, a FHMM would require a K x K transition matrix.

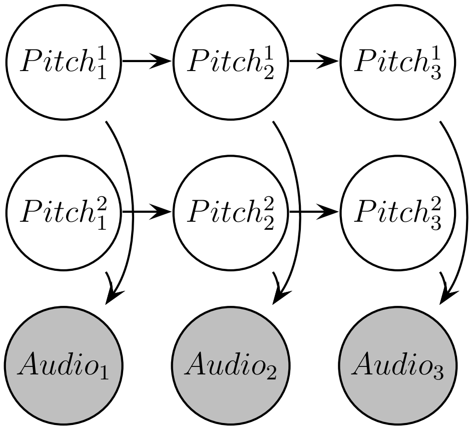

Figure 3.4: A FHMM for two-instrument polyphonic pitch tracking.

Each instrument makes use of the pitch transition model discussed in Section 3.7.2. Under this model, small interval pitch transitions tend to be more likely than large interval transitions, with transitions to the same note being the most likely. Provided that the two instruments are well-separated in pitch and tend to play small-interval pitch transitions, this transition model should improve instrument identification performance over a time-independent graphical model.

3.8.3 Observation Model

We make use of the HMM observation model from Section 3.7.3 by generalizing it to the case of two instruments. For two instruments, Equation 3.6 for a spectrogram time slice becomes

$$|X_t(f)| = Instrument_t^1(f) + Instrument_t^2(f) + \xi(f)$$

where

$$Instrument_t^i(f) = Intensity_t^i \sum_{h=1}^{H} \alpha_h^i(f_{pitch}) B(f - hf_{pitch})$$

Conditional on the hidden state pitch variable we then have the following Gaussian observation model:

$$p(Audio_t | Pitch_t^i, Pitch_t^j) = \mathcal{N}(Audio_t | \mu(Pitch_t^i, Pitch_t^j), \sigma_\xi^2 I)$$

where $\mu(Pitch_t^i, Pitch_t^j) = [\mu_1, ..., \mu_N]^T$ and $\mu_k = Instrument_t^1(f_k) + Instrument_t^2(f_k)$.

Since intensity level is not modeled, we set the $Intensity_t^i$ variable to a constant. We added a “no pitch” state to allow the system to identify unpitched regions of the input signal. The “no pitch” state corresponds to a zero-valued spectral template ($\mu_k = 0$ for $k = 1...M$).

3.8.4 Experiments

The $\alpha_h(f_{pitch})$ harmonic magnitude parameters and observation noise variance σ_ξ^2 parameters are learned by an EM-based estimation procedure by training the HMM in Figure 3.5 on monophonic single-instrument audio recordings. The audio recordings consist of chromatic scales played over the entire pitch range of the instrument. We then combine the the HMMs into the FHMM model in Figure 3.4. We used the same 12 allowable pitch states for each instrument, corresponding to the 12 semi-tones from A-flat below middle C to A-flat above middle C. The output pitch estimates are obtained by computing the Viterbi path for the FHMM.

Note that the FHMM in this section performs pitch tracking, and not transcription. In the experiments that follow, we present the pitch estimates along with the truth score. The truth score is taken to be the MIDI file that was used to generate the input sound file. This will allow is to see if our modeling approach is at least able to perform multi-instrument polyphonic pitch tracking.

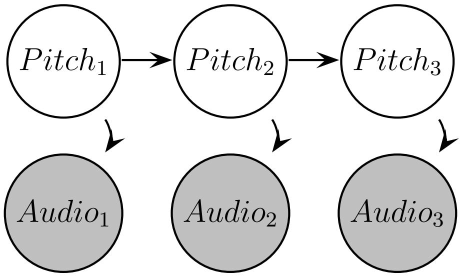

Figure 3.5: The training HMM. Shaded nodes are observed audio data; unshaded nodes represent the hidden pitch state.

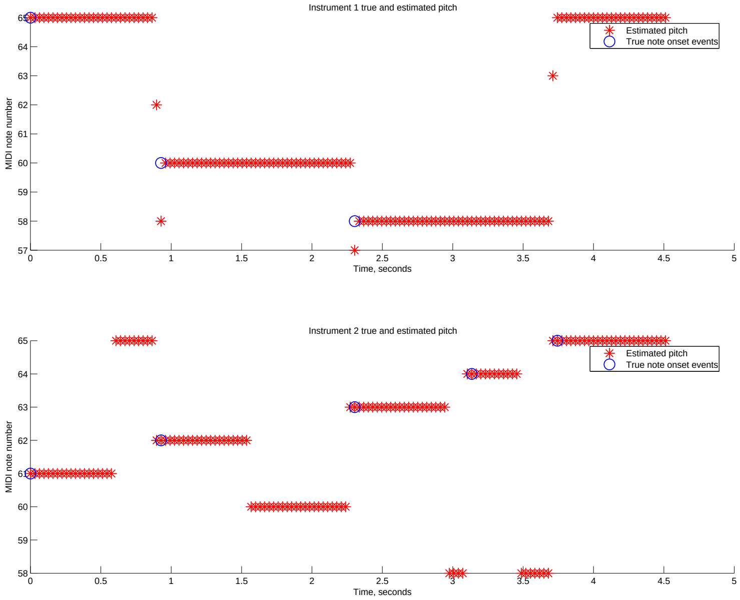

Figure 3.6: Pitch tracking results for the piano and violin example (clip 1). Instrument 1 corresponds to the violin, and Instrument 2 corresponds to the piano. The true note onsets are denoted by blue circles and the pitch estimates are denoted by the red stars.

Piano and violin results

We now present results for a recording of synthesized piano and violin sounds. The input sound file was synthesized from a MIDI file, which is taken to be the truth score. We will refer to this sound file as “clip 1” throughout this chapter. Clip 1 consists of several piano and violin notes that have note durations greater than 0.5 seconds. Clip 1 has a maximum polyphony of two, and each instrument is monophonic. Figure 3.6 shows the pitch tracking results. The true note onset events from the MIDI file are also shown on the same graph for reference.

We observe that the pitch estimates are generally in agreement with the pitch of the corresponding truth score onsets. However, we see that there are some erroneous pitch estimates near the onsets. We also observe that there are some instrument identification errors, such as the region from 1.5 to 2.2 seconds where the chain two pitch estimate corresponds to the true pitch of chain one.

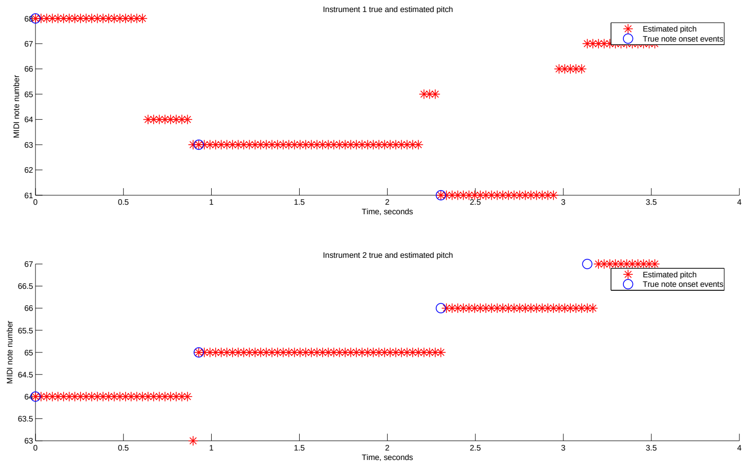

Figure 3.7: Pitch tracking results for the church organ and trumpet example (clip 2). Instrument 1 corresponds to the church organ, and Instrument 2 corresponds to the trumpet. The true note onsets are denoted by blue circles and the pitch estimates are denoted by the red stars.

Organ and trumpet results

We now present results for synthesized church organ and trumpet sounds. The input sound file was synthesized from a MIDI file, which is taken to be the truth score. We will refer to this sound file as “clip 2” throughout this chapter. Clip 2 consists of several church organ and trumpet notes that have note durations greater than 0.5 seconds. Clip 2 has a maximum polyphony of two, and each instrument is monophonic. Figure 3.7 shows the pitch tracking results. The true note onset events from the MIDI file are also shown on the same graph for reference.

We observe that the pitch estimates are generally in agreement with the pitch of the corresponding truth score onsets. However, we see that there are some erroneous pitch estimates near the onsets. We also observe that there are some instrument identification errors, such as the region from 3.2 to 3.5 seconds where the chain 1 pitch estimate corresponds to the true pitch of chain 2.

3.8.5 Remarks

We presented a FHMM for two-instrument polyphonic pitch tracking. The FHMM made use of a pitch process Markov chain for each instrument. Like the HMM system in Section 3.7, the FHMM system presented here performs pitch tracking, but not transcription, since note events are not identified. Also, intensity level is not modeled. A “no pitch” state could be used to allow the system to identify regions in the input signal where no note is sounding. However, we think a better solution would consist of a system that models intensity explicitly. We will address this issue in the following sections by including an explicit intensity level state.

We would like the pitch transition model to reflect the transition probabilities at the level of note events. However, our pitch transition model reflects the transition probabilities at the level of individual spectrogram time slices. When we attempt to learn the pitch transition model from musical input recordings, the resulting pitch transition matrices tend to have very small off-diagonal elements. In Section 3.12, we present a DGM that incorporates a pitch transition model for note events.

3.9 A FHMM for monophonic pitch tracking with intensity estimation

The models presented in the previous sections performed pitch tracking for both the single and multi-instrument case. However, intensity level was not modeled. However, the intensity level is also musically relevant. In order to identify note events, we music take the intensity level into account. Our goal is to create a system that includes a note event model. In this section, we present a system that brings us a step closer to a note event model. We present a FHMM system that includes a note intensity model. We consider the FHMM system in this section to implement a partial note event model, since we model the pitch and intensity processes as marginally independent. We therefore still consider this system to perform pitch tracking and not transcription. We present a more complete note event model in Section 3.11.

We propose an intensity model for note events that allows only physically realizable intensity envelopes. A nice feature of our modeling approach is that it is intuitive and straightforward to create new intensity envelope models by simply editing a single state transition matrix. We illustrate this by providing a concrete example of how to define two different types of intensity envelopes and we present some experimental results.

3.9.1 Model

Figure 3.8 shows the FHMM. We model the intensity and pitch evolution of a single monophonic instrument. In the FHMM, the intensity and pitch state processes are modeled as being marginally independent. Of course, in an actually musical signal the pitch and intensity are not independent since the pitch tends to only change at note onset instants. We present a DGM that takes this dependence into account in Section 3.11.

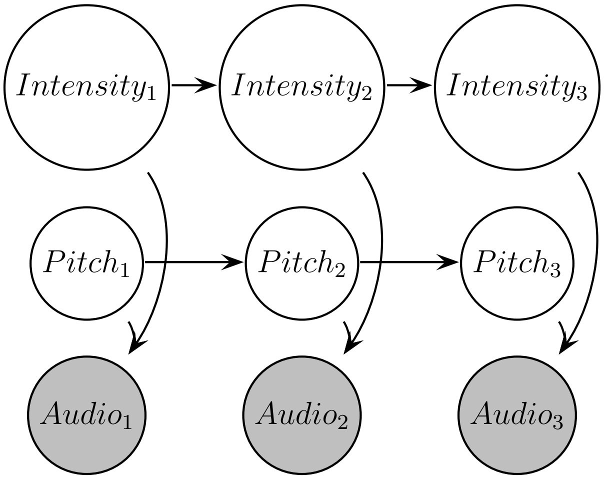

Figure 3.8: A FHMM for monophonic pitch and intensity tracking.

3.9.2 Intensity Envelope Transition Model

The timbre of a musical instrument is influenced by both the spectral content, and the way in which the sound level changes over time. In the piano, the sound level immediately begins decaying after the initial hammer strike. This is also the case for plucked string instruments such as the guitar. However, in instruments where energy is continually supplied during the

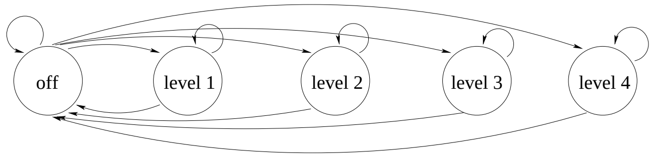

Figure 3.9: A state transition diagram for an instrument characterized by a constant sound level after the note onset. Five discretized intensity levels are shown for clarity. Our actual implementation uses ten intensity levels. The “off” state denotes zero intensity.

playing of a note, such as bowed string instruments, brass instruments, wind instruments and organs, the sound level fluctuates less during the playing of a note.

We propose two models of intensity envelopes for musical instruments. The “constant envelope model” is for instruments characterized by a steadier sound level after the note onset. The “decaying envelope model” is for instruments characterized by a gradual sound decay after the note onset.

The constant envelope model is shown in Figure 3.9. The state transition diagram in the figure comprises five discretized intensity levels, including the note-off state (zero intensity). Transitions from the note-off state to any nonzero intensity level are allowed. Self-loop transitions are allowed on all states. However, any outgoing transition from a nonzero intensity state must return to the note-off state. Thus, realizations of this state transition model will always result in note intensity envelopes consisting of a transition from the note-off state to some nonzero intensity level, followed by some number of self loops while the note sustains, followed by a transition back to the note-off state. This model defines a geometric distribution over note durations. Specifically, if the self-loop probability is p_{self}, then the probability that we remain at the same intensity for n time slices is $p_{self}(n) = (1 - p_{self})p_{self}^{n-1}$. Thus different expected note durations can be modeled by adjusting p_{self}.

The decaying envelope model is shown in Figure 3.10 for modeling the intensity envelope of instruments characterized by a decaying sound level after the note onset. In this model, transitions from the note-off state to any nonzero intensity level are allowed. Self-loop transitions are allowed on all states. However, any outgoing transition from a nonzero

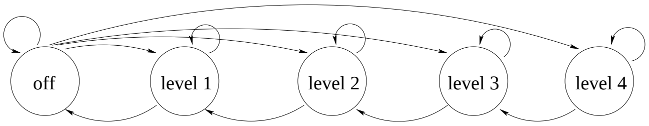

Figure 3.10: A state transition diagram for an instrument characterized by a decaying sound level after the note onset. Five discretized intensity levels are shown for clarity. Our actual implementation uses ten intensity levels. The “off” state denotes zero intensity.

intensity state must lead to the next lower intensity state. Thus, realizations of this state transition model will always result in note intensity envelopes consisting of a transition from the note-off state to some nonzero intensity level, followed by some number of self loops, followed by a transition to the next lower intensity state, and so on, until the note-off state is reached.

Our modeling intention in the above two envelopes was to present two different and relatively simple models to illustrate how the intensity envelope can be modeled within the DGM framework, while still being a reasonable model for some acoustic instruments.. Recall that our goal is not to produce a spectral model that is accurate enough to be used for synthesis. It is only required that our model be able to discriminate between instruments.

It is then straightforward to construct the state transition matrix

$$p(Intensity_t = j | Intensity_{t-1} = i) = Intensity\_CPT(i, j)$$

by setting all entries $Intensity\_CPT(i, j)$ for which there is no arc from state i to state j in the state transition diagram to zero. The remaining entries can be specified manually or learned from training data. This approach to modeling the intensity envelope allows us to experiment with different intensity envelope constrains by simply editing the the entries of $Intensity\_CPT$.

3.9.3 Pitch Transition Model

We make use of the pitch transition model discussed in Section 3.7.2. Under this model, small interval pitch transitions tend to be more likely than large interval transitions, with transitions to the same note being the most likely. We make use of the pitch helix model.

3.9.4 Observation Model

Our observation model is based on the observation model from Section 3.7.3, except that it is now also a function of the intensity state. We use the same Gaussian process model for the spectrum at time slice t, repeated here:

$$|X_t(f)| = Instrument_t(f) + \xi(f) \tag{3.8}$$

where

$$Instrument_t(f) = Intensity_t \sum_{h=1}^{H} \alpha_h(f_{pitch}) B(f - hf_{pitch}) \tag{3.9}$$

Conditional on the hidden $Intensity_t$ and $Pitch_t$ state variables, we then have the following Gaussian observation mode:

$$p(Audio_t | Intensity_t, Pitch_t) = \mathcal{N}(Audio_t | \mu(Intensity_t Pitch_t), \sigma_\xi^2 I)$$

where $\mu(Intensity_t, Pitch_t) = [\mu_1, ..., \mu_N]^T$ and $\mu_i = Instrument_t(f_i)$.

We must also specify a mapping from the (discretized) hidden intensity state variable to the $Intensity_t$ parameter in Equation 3.9. We propose that the intensity state values correspond to intensity levels that are discretized uniformly in log magnitude over the effective dynamic range of the input signal. The smallest intensity level is then considered to correspond to the note-off state.

3.9.5 Experiments

The $\alpha_h(f_{pitch})$ harmonic magnitude parameters and observation noise variance σ_ξ^2 parameters for each instrument are learned by an EM-based estimation procedure on the FHMM in Figure 3.8. Training is carried out on monophonic single-instrument audio recordings. The audio recordings consist of chromatic scales played over the entire pitch range of the instrument.

We used 10 intensity states, discretized uniformly in log magnitude over a 60 dB dynamic range. The lowest intensity state corresponds to the note-off event. Performing inference on the FHMM to compute the path of maximum posterior probability gives us the intensity and pitch estimates.

We implemented the FHMM pitch tracker and observed its pitch tracking performance on a few short monophonic musical recordings. The output pitch was taken to be the mode of the posterior (Viterbi path). We present results for this model on guitar recordings in Section 4.5.

3.9.6 Remarks

In this section we presented a FHMM for single instrument monophonic pitch tracking with intensity level estimation. This was accomplished by using both a pitch process and an intensity process model. We made use of a constrained intensity transition model that was intended to model the intensity envelope of note events. We proposed a "decaying envelope" model as well as a "constant envelope" intensity model. The DGM framework also makes it straightforward to implement more complex intensity envelopes, such as those with hierarchical structure. For example, a hierarchical HMM could be used to model an intensity envelope in which there are distinct regions such as attack, sustain, and decay, for example.

Observe that the pitch and intensity states are modeled as evolving independently in the FHMM. However, in an acoustic instrument, a change in pitch is generally accompanied by a sudden increase in intensity level, and the pitch tends not to change suddenly except when the intensity level also changes suddenly. We therefore consider this model to implement a partial note event model. We will implement a complete note event model in Section 3.11.

We consider the FHMM system in this section to implement a partial note event model, since we model the pitch and intensity processes as marginally independent. We therefore still consider this system to perform pitch tracking and not transcription. We present a more complete note event model in Section 3.11.

3.10 A FHMM for polyphonic multi-instrument pitch tracking with intensity estimation

We now generalize the FHMM from Section 3.9 to the case of two instruments. This system models the pitch and intensity level for two instruments, each capable of playing at most one note at a time. We train two of the single-instrument FHMMs from Section 3.9 on recorded audio data from isolated instrument recordings. We then combine the two trained FHMMs to form a larger two-instrument FHMM. The pitch and intensity output on each instrument is obtained by computing the Viterbi path for the two-instrument FHMM.

We consider the FHMM system in this section to implement a partial note event model, since we model the pitch and intensity processes as marginally independent. We therefore still consider this system to perform pitch tracking and not transcription. We present a more complete note event model in Section 3.11.

3.10.1 Model

Figure 3.11 shows the decoding FHMM for a two-instrument pitch and intensity tracking system. The top two chains model the intensity and pitch evolution of one instrument, while next two lower chains model the intensity and pitch evolution of another instrument. The audio feature data at each time step is dependent on all four hidden state variables.

3.10.2 Observation Model

Our observation model is based on the observation model from Section 3.8.3, except that it is now also a function of the intensity state. For two instruments, Equation 3.6 for a spectrogram time slice becomes

$$|X_t(f)| = Instrument_t^1(f) + Instrument_t^2(f) + \xi(f)$$

where

$$Instrument_t^i(f) = Intensity_t^i \sum_{h=1}^{H} \alpha_h^i(f_{pitch}) B(f - hf_{pitch})$$

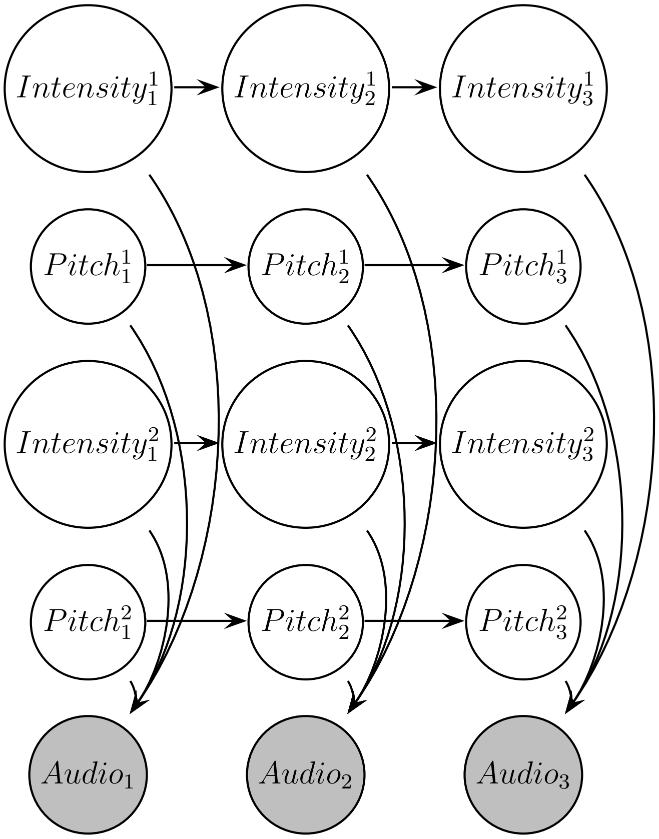

Figure 3.11: The decoding FHMM for a two-instrument pitch and intensity tracking system.

Conditional on the hidden state intensity and pitch variables, we then have the following Gaussian observation model:

$$\begin{aligned} & p(Audio_t|Intensity_t^i, Pitch_t^i, Intensity_t^j, Pitch_t^j) \\ = \quad & \mathcal{N}(Audio_t|\mu(Intensity_t^i, Pitch_t^i, Intensity_t^j, Pitch_t^j), \sigma_\xi^2 I) \end{aligned}$$

where $\mu(Intensity_t^i, Pitch_t^i, Intensity_t^j, Pitch_t^j) = [\mu_1, ..., \mu_N]^T$ and
$\mu_k = Instrument_t^1(f_k) + Instrument_t^2(f_k)$.

3.10.3 Experiments

The intensity transition probabilities, observation noise variance σ_ξ^2, and the $\alpha_h(f_{pitch})$ harmonic magnitude parameters for each instrument are learned by an EM-based estimation procedure on the single-instrument FHMM in Figure 3.8. We then combine the the single-instrument FMMs into the two-instrument FHMM model in Figure 3.11.

We used 10 intensity states for each instrument, discretized uniformly in log magnitude over a 60 dB dynamic range. The lowest intensity state corresponds to the note-off event. Performing inference on the FHMM to compute the mode of the posterior probability gives us the intensity and pitch estimates.

Note that the FHMM in this section performs pitch tracking, and not transcription. In the experiments that follow, we present the pitch estimates along with the truth score. The truth score is taken to be the MIDI file that was used to generate the input sound file. We also plot the estimated note onsets from our intensity model and compare with the true onsets.

Piano and violin results

We now present results for synthesized piano and violin sounds, using clip 1 from Section 3.8.4. In this experiment, we use the constant envelope transition model from Figure 3.9 for instrument 1 (violin). We use the decaying envelope transition model from Figure 3.10 for instrument 2 (piano). Our expectation is that the decaying envelope model is closer to being a reasonable transition model for a piano than a violin. Figure 3.12 shows the pitch tracking results. The true note onset events from the MIDI file are also shown on the same graph for reference.

We observe that the pitch tracking results are very similar to the results using the FHMM without an intensity state. That is, we observe that the pitch estimates are generally in agreement with the pitch of the corresponding truth score onsets. However, we see that there are some erroneous pitch estimates near the onsets. We also observe that there are some instrument identification errors, such as the region from 1.5 to 2.2 seconds where the chain two pitch estimate corresponds to the true pitch of chain one.

Figure 3.13 shows the intensity envelope estimation results. Figure 3.14 shows the onset estimation results for the FHMM along with the true note onsets. We observe that only two onsets (the first onset on each chain) are identified correctly. Note also that the true pitch changes several times during the decay of the estimated intensity envelopes. We

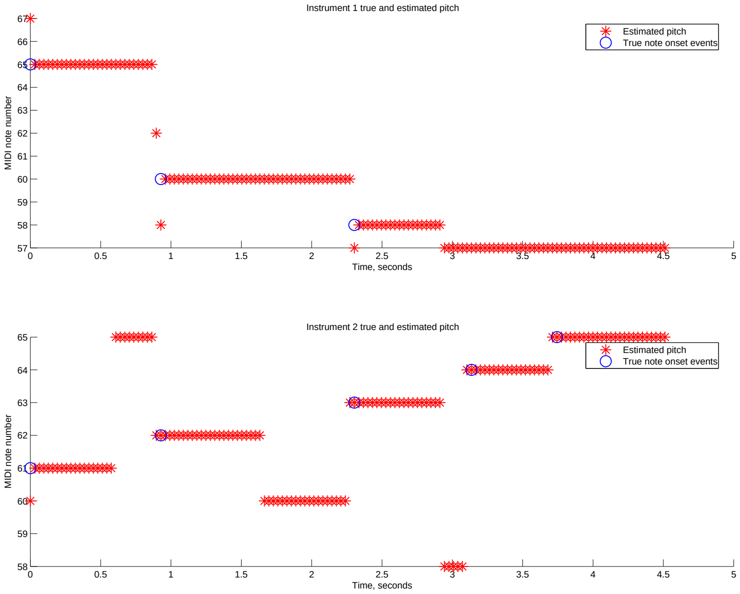

Figure 3.12: Pitch tracking results for the piano and violin example (clip 1). The true note onset events are denoted by blue circles and the pitch estimates are denoted by the red stars.

therefore see that interpreting the estimated intensity envelope onsets as note event onsets would lead to poor performance for this model.

Organ and trumpet results

We now consider an example where both instruments share the same intensity transition model. In this case, instrument identification is performed by making use of the spectral model alone. We present results for synthesized church organ and trumpet sounds, using clip 2 from Section 3.8.4. In this experiment, we use the constant envelope transition model from Figure 3.9 for instrument 1 (church organ) as well as for instrument 2 (trumpet). These instruments have the characteristic that a sustaining note tends to have a relatively constant intensity level, at least when not performed expressively. Therefore, the constant envelope model seems to be a reasonable choice. This also makes for an interesting experiment since our system must rely on the spectral timbre model to perform instrument identification.

Figure 3.15 shows the pitch tracking results. The true note onset events from the MIDI file are also shown on the same graph for reference. We observe that the pitch tracking

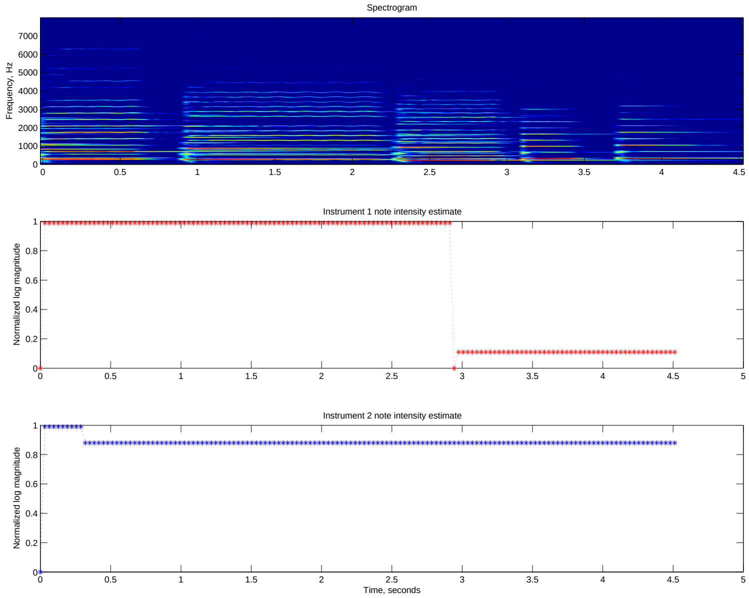

Figure 3.13: Intensity envelope estimation results for the piano and violin example (clip 1). Instrument 1 corresponds to the violin, and Instrument 2 corresponds to the piano. The spectrogram of the input audio file is shown at the top.

results are very similar to the results using the FHMM without an intensity state. That is, we observe that the pitch estimates are generally in agreement with the pitch of the corresponding truth score onsets. However, we see that there are some erroneous pitch estimates near the onsets. We also observe that there are some instrument identification errors, such as the region from 3.2 to 3.5 seconds where the chain 1 pitch estimate corresponds to the true pitch of chain 2.

However, we now also have an intensity estimate. Figure 3.16 shows the intensity envelope estimation results. Figure 3.17 shows the onset estimation results for the FHMM along with the true note onsets. We observe that only two onsets (the first onset on each chain) are identified correctly. Note also that the true pitch changes several times during the decay of the estimated intensity envelopes. We therefore see that interpreting the estimated intensity envelope onsets as note event onsets would lead to poor performance for this model.

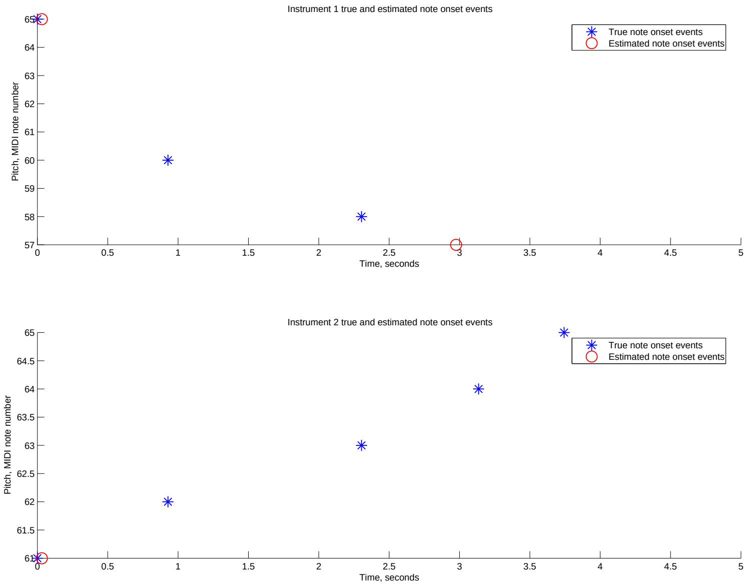

Figure 3.14: Onset estimation results for the piano and violin example (clip 1).

Time-independent graphical model results

We now present some pitch tracking and intensity level estimation results for the piano and violin clip (clip 1) under a simpler graphical model. We use the time-independent graphical model in Figure 3.18. This will allow us to observe the effect of the lack of a dynamical pitch and intensity model. In this model, we use a uniform multinomial prior for all of the hidden variables.

Figure 3.19 shows the estimated note intensity envelopes for clip 1. Figure 3.20 shows the estimated pitch versus time, and also shows the true note onsets. Note that there are several pitch errors, many of which occur around the note onsets. Note also that an increase in intensity between two successive time steps in Figure 3.19 often does not correspond to a note onset event in the truth score. In order to obtain note event estimates, some kind of post-processing would need to be performed on the Viterbi path results.

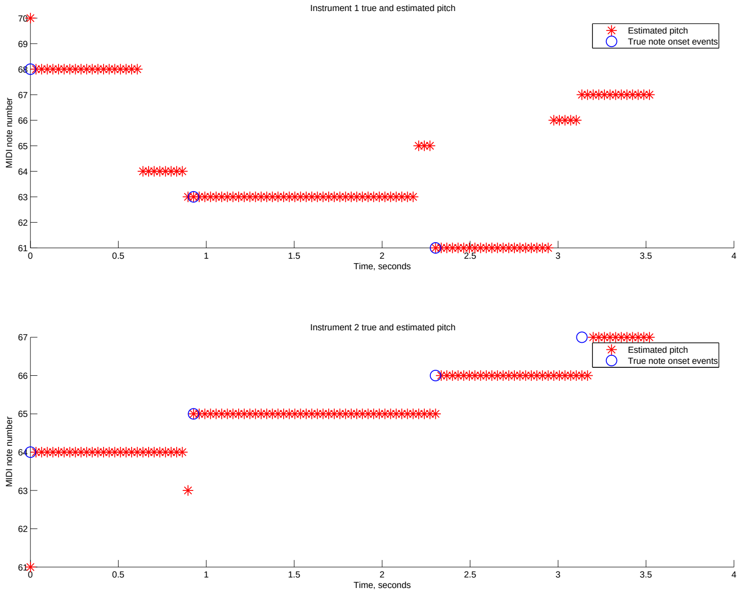

Figure 3.15: Pitch tracking results for the church organ and trumpet example (clip 2). The true note onset events are denoted by blue circles and the pitch estimates are denoted by the red stars.

3.10.4 Remarks

In this section we presented a FHMM for two-instrument polyphonic pitch tracking with intensity level estimation. This was accomplished by using a pitch process and an intensity process model for two instruments. Our approach consisted of using the EM algorithm to train two copies of the single-instrument FHMM from Section 3.9 on single instrument audio recordings. We then combined two single-instrument FHMMs to form the larger two-instrument FHMM in Figure 3.11. The pitch and intensity level output was obtained as the mode of the posterior.

We consider the FHMM system in this section to implement a partial note event model, since we model the pitch and intensity processes as being marginally independent. We therefore still consider this system to perform pitch tracking and not transcription.

Based on our experimental results in this section, we might consider defining a note event onset to occur when either an intensity envelope onset or a change in pitch state occurs. However, we observe that sometimes a pitch error occurs for only a single time

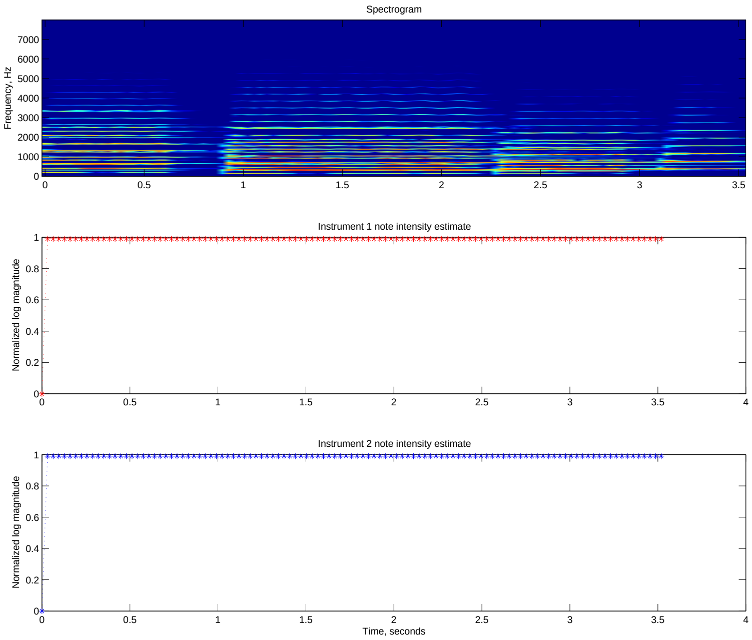

Figure 3.16: Intensity envelope estimation results for the church organ and trumpet example (clip 2). The spectrogram of the input audio file is shown at the top.

slice. We might then consider changing our definition so that the pitch state must remain the same for at least a few time slices. However, this seems somewhat ad hoc. We think a better solution would consist of a note event model in which the pitch state is tied to the intensity state in such a way that the pitch cannot change within a single note event intensity envelope. That is, we think that a more elegant approach would be to create a graphical model that makes the intensity and pitch states dependent in such a way that only physically plausible intensity and pitch combinations are possible. We will implement such a model in the following section.

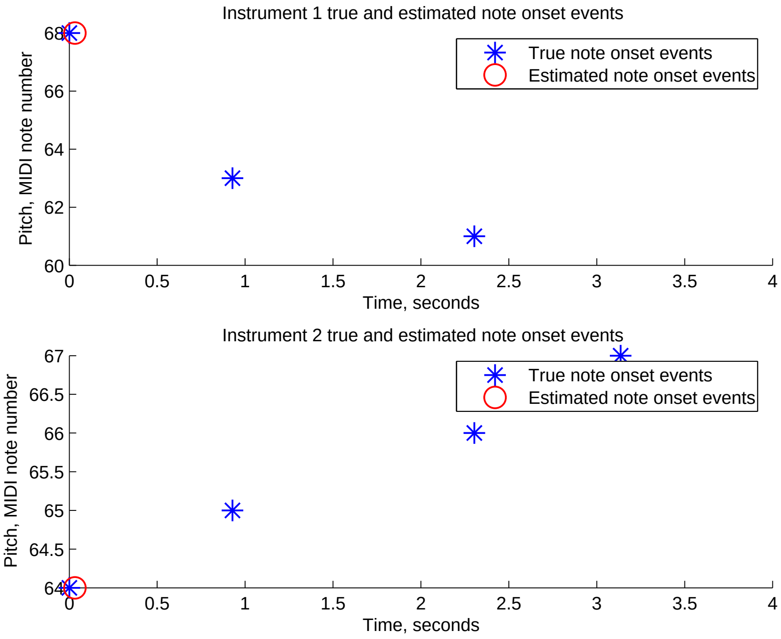

Figure 3.17: Onset estimation results for the church organ and trumpet example (clip 2).

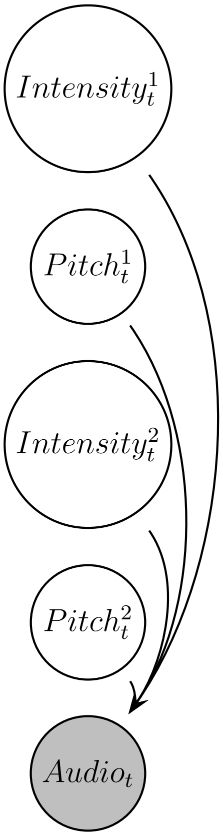

Figure 3.18: A graphical model for polyphonic two-instrument transcription. One time slice is shown.

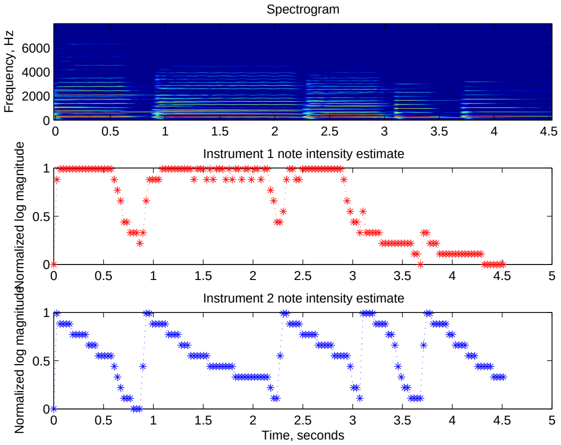

Figure 3.19: Estimated note intensity envelopes for synthesized piano and violin sounds (clip 1). Instrument 1 corresponds to the violin, and Instrument 2 corresponds to the piano. The spectrogram of the input audio file is shown at the top.

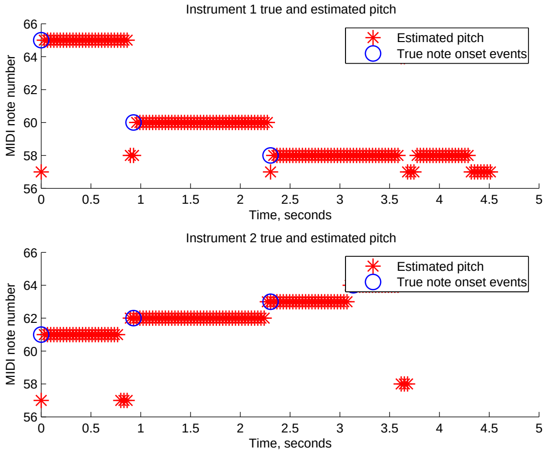

Figure 3.20: Estimated pitch versus time for synthesized piano and violin sounds (clip 1). Instrument 1 corresponds to the violin, and Instrument 2 corresponds to the piano.

3.11 A DGM for monophonic audio to piano roll transcription

In this section we present a DGM for monophonic audio to piano roll transcription. Unlike the models presented in the previous sections, the DGM in this section makes use of a note event model. In a piano roll score, a symbolic note event corresponds to the notion of an onset time, pitch, and offset time. Our modeling assumption is that within a note event, the pitch remains constant while the intensity envelope evolves according to either the constant envelope or the decaying envelope model from Section 3.9.2.

Our note event model constrains the intensity and pitch state to evolve according to a hierarchical state transition model. That is, the pitch is only allowed to change if the intensity level is zero (note-off state). Once a note onset occurs, the pitch must then remain constant while the intensity state evolves according to the state transition diagram in Figure 3.9 or Figure 3.10. The pitch may then change again once the note-off state is reached. The DGM that we use is essentially a hierarchical hidden Markov model (HHMM) [Mur02].

The DGM in this section also provides for a more musically reasonable pitch transition model than the models in the previous sections. We use the same pitch helix transition model. However, our note event model now constrains the pitch to remain constant during the sustain or decay of a note. Thus, we model pitch transitions at the level of whole note events. In our previous models, we modeled pitch transitions at the level of individual spectrogram time slices.

3.11.1 Model

Figure 3.21 shows a DGM for monophonic audio to piano roll transcription. This DGM differs from the FHMM in Section 3.9 only in that there is an additional diagonal arc from $Intensity_{t-1}$ to $Pitch_t$. The additional arc makes the pitch transition CPD a function of both the previous pitch and the previous intensity state. This allows us to constrain the pitch to remain constant within a note event.

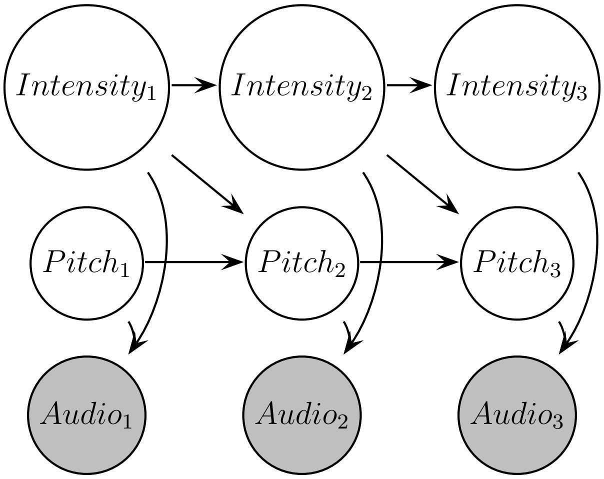

Figure 3.21: A DGM for for monophonic audio to piano roll transcription. The diagonal arcs constrain the note pitch to remain constant during the decay or sustain portion of the intensity envelope.

3.11.2 Pitch transition model

We place constraints on the times at which the pitch state can change. Specifically, pitch state change events within a single note envelope event should be disallowed. This is done by making the pitch state conditional probability distribution a function of both the previous pitch state and the previous intensity state. In our model, the pitch state is only allowed to change when the previous intensity state was the "note-off" state. In particular, the pitch transition model for instrument m is given by

$$\begin{aligned} & p(Pitch_t^m = j | Intensity_{t-1}^m = k, Pitch_{t-1}^m = i) \\ &= \begin{cases} \delta(i,j) & \text{if } k > 0 \text{ (stay in the same state)} \\ T^m(i,j) & \text{if } k = 0 \text{ (pitch transition)} \end{cases} \end{aligned} \tag{3.10}$$

where the $Intensity_{t-1}^m = 0$ state denotes the "note-off" state. $T^m(i,j)$ represents the pitch transition model for whole note events for instrument m. That is, $T^m(i,j) = p(Pitch_t^m = j|Pitch_{t-1}^m = i)$. We choose to make $T^m(i,j)$ instrument specific to reflect the fact that the set of allowable pitches can depend on the instrument. In our current implementation, we set $T^1(i,j) = T^2(i,j)$ so that instrument classification performance only depend on the intensity transition model and the observation model (and not on the pitch ranges of the instruments).

3.11.3 Experiments

The $\alpha_h(f_{pitch})$ harmonic magnitude parameters, intensity transition probabilities, and observation noise variance σ_ξ^2 parameters are learned by an EM-based estimation procedure by training on monophonic single-instrument audio recordings. The audio recordings consist of chromatic scales played over the entire pitch range of the instrument.

We used 10 intensity states for each instrument, discretized uniformly in log magnitude over a 60 dB dynamic range. The lowest intensity state corresponds to the note-off event. Performing inference on the DGM to compute the mode of the posterior probability gives us the output transcription. We implemented the DGM and observed its transcription performance on a few short monophonic musical recordings. We present results for this model on guitar recordings in Section 4.5.

3.11.4 Remarks

In this section we presented a DGM for monophonic audio to piano roll transcription. This was accomplished by making use of a note event model in which we place constraints on both the shape of the intensity envelope and the times at which the note pitch is allowed to change. We made use of the intensity envelope model from Section 3.9.2 which contains explicit “onset” and “offset” states and in which the envelope must be either decaying or constant-valued between the onset and offset states.

A note onset is defined as a transition away from the note-off (zero intensity) state. In a realization of our note event model, the intensity must therefore decay to zero before a new note onset event can occur. This may be a reasonable model for a slow musical passage in which long duration notes or regions of silence between adjacent notes is common. However, it can also be common for a note re-articulation to occur before the intensity decays to zero. An example of this is in a sequence of notes played via the hammer-on or pull-off guitar technique. In Chapter 4 we show that the current note event model can perform poorly for such signals. In Section 4.4 we propose a different note event model in the context of guitar transcription that makes us of a more reasonable model for the case of re-articulated notes.

3.12 A DGM for polyphonic multi-instrument audio to piano roll transcription

We now generalize the DGM from Section 3.11 to the case of two instruments. The previous DGMs modeled monophonic input signals. We now present a DGM for polyphonic multi-instrument transcription. We present a DGM for multi-instrument polyphonic transcription consisting of two musical instruments, each capable of playing at most one note at a time. A key feature of this model is the use of a note-event timbre model that includes both a spectral model and a dynamic intensity versus time model (i.e., a time envelope model). In a piano roll score, a symbolic note event corresponds to the notion of an onset time, pitch, and offset time. Our modeling assumption is that within a note event, the pitch remains constant while the intensity envelope evolves according to either the constant envelope or the decaying envelope model from Section 3.9.2.

3.12.1 Model

Figure 3.22 shows the decoding DGM for a two-instrument transcription system. This model is a factorial version of the monophonic DGM from Section 3.11. The hidden state variables $Intensity_t^i$ and $Pitch_t^i$ represent the instantaneous intensity and pitch, respectively, of instrument i at time t.

3.12.2 Observation model

We use the observation model from Section 3.10.2.

3.12.3 Experiments

The intensity transition probabilities, observation noise variance σ_ξ^2, and the $\alpha_h(f_{pitch})$ harmonic magnitude parameters for each instrument are learned by an EM-based estimation procedure on the single-instrument DGM in Figure 3.21. The harmonic width parame-

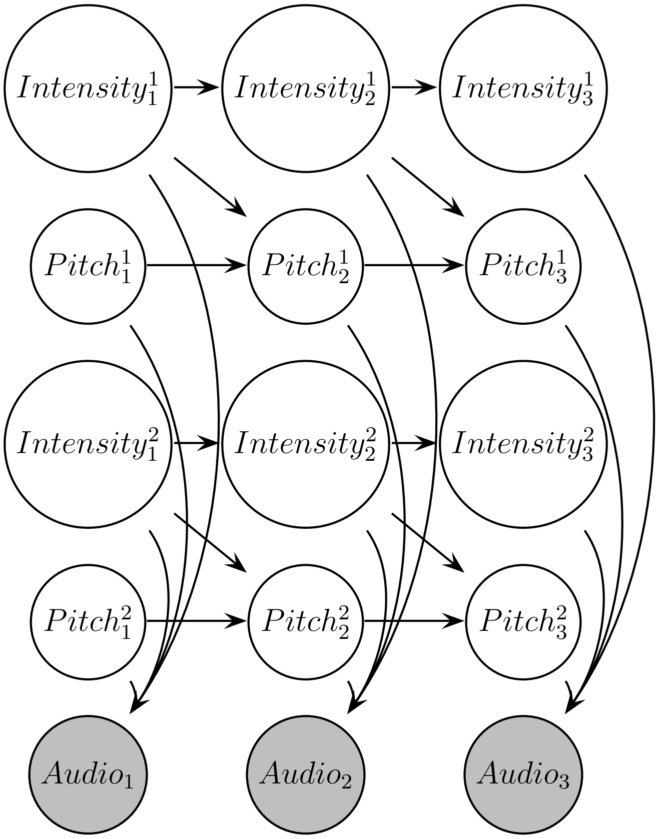

Figure 3.22: The decoding DGM for a polyphonic two-instrument audio to piano roll transcription system. The diagonal arcs constrain the note pitch to remain constant during the decay or sustain portion of the intensity envelope.

ter σ^2 and the pitch transition parameters were chosen manually. We then combined the single-instrument DGMs to form the larger two-instrument DGM in Figure 3.22. Performing inference on the DGM to compute the mode of the posterior gives us the transcription in the form of explicit note events.

We used 10 intensity states for each instrument, discretized uniformly in log magnitude over a 60 dB dynamic range. The lowest intensity state corresponds to the note-off event. Performing inference on the FHMM to compute the path of maximum posterior probability gives us the intensity and pitch estimates. We used the same 12 allowable pitch states for each instrument, corresponding to the 12 semi-tones from A-flat below middle C to A-flat above middle C. We used the transcription error rate measure from Section 3.5 to measure transcription performance.

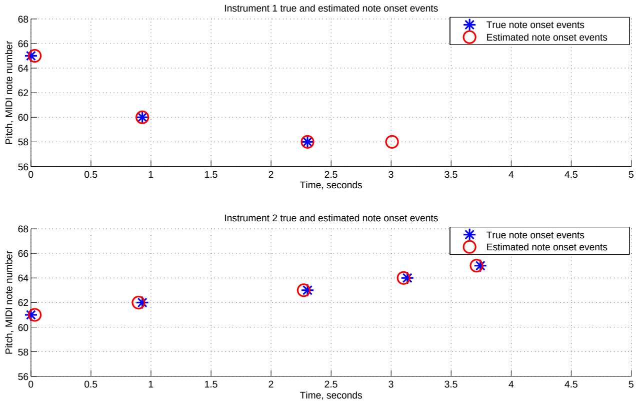

Figure 3.23: Transcription results for the piano and violin example (clip 1). Instrument 1 corresponds to the violin, and Instrument 2 corresponds to the piano.

Piano and violin results

We now present results for synthesized piano and violin sounds, using clip 1 from Section 3.8.4. In this experiment, we use the constant envelope transition model from Figure 3.9 for instrument 1 (violin). We use the decaying envelope transition model from Figure 3.10 for instrument 2 (piano). Our expectation is that the decaying envelope model is closer to being a reasonable transition model for a piano than a violin.

Figure 3.23 shows the transcription results. The true note onset events from the MIDI file are also shown on the same graph for reference. Figure 3.24 shows the intensity envelope estimation results. The transcription error rate was 12.5%. There were 8 notes total, with a single insertion error (on instrument 1 at around 3 seconds).

Organ and trumpet results

We now consider an example where both instruments share the same intensity transition model. In this case, instrument identification is performed by making use of the spectral model alone. We present results for synthesized church organ and trumpet sounds, using clip 2 from Section 3.8.4. In this experiment, we use the constant envelope transition model from Figure 3.9 for instrument 1 (church organ) as well as for instrument 2 (trumpet). These instruments have the characteristic that a sustaining note tends to have a relatively constant

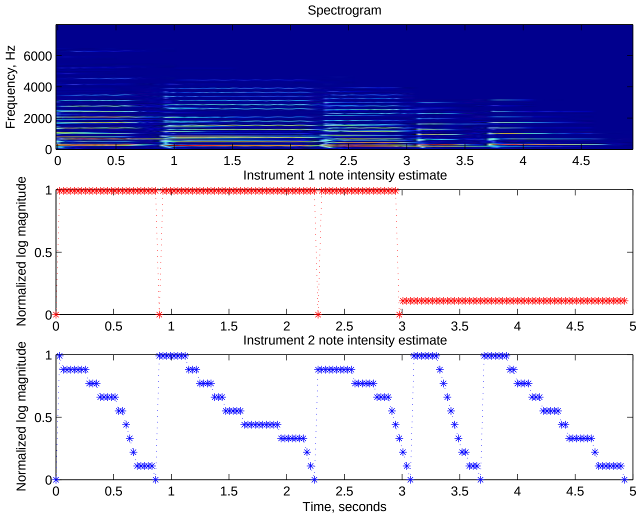

Figure 3.24: Estimated note intensity envelopes for the piano and violin example (clip 1). Instrument 1 corresponds to the violin, and Instrument 2 corresponds to the piano. The spectrogram of the input audio file is shown at the top.

intensity level when not performed expressively. Therefore, the constant envelope model seems to be a reasonable choice. This also makes for an interesting experiment since our system must rely on the spectral timbre model alone to perform instrument identification.

Figure 3.25 shows the transcription results. The true note onset events from the MIDI file are also shown on the same graph for reference. Figure 3.26 shows the intensity envelope estimation results. The transcription error rate was 28.6 %. There were 7 notes total. All pitches were correctly identified. However, there was a single instrument classification error on the last note played by the trumpet. This error corresponds to 1 deletion and 1 substitution.

Short-duration note results

We now present results for another recording of synthesized piano and violin sounds, which we refer to as clip 3. Clip 3 is a 1 second excerpt from Bach's two-part Invention #8, consisting of 16 notes total, 8 for each instrument. The right-hand part is played on the violin and the left-hand part is played on the piano. This is a more challenging transcription problem for two reasons: The notes have much shorter durations (approximately 0.13 seconds) than in the previous examples. Also, all of the note onsets consist of two-note pairs where

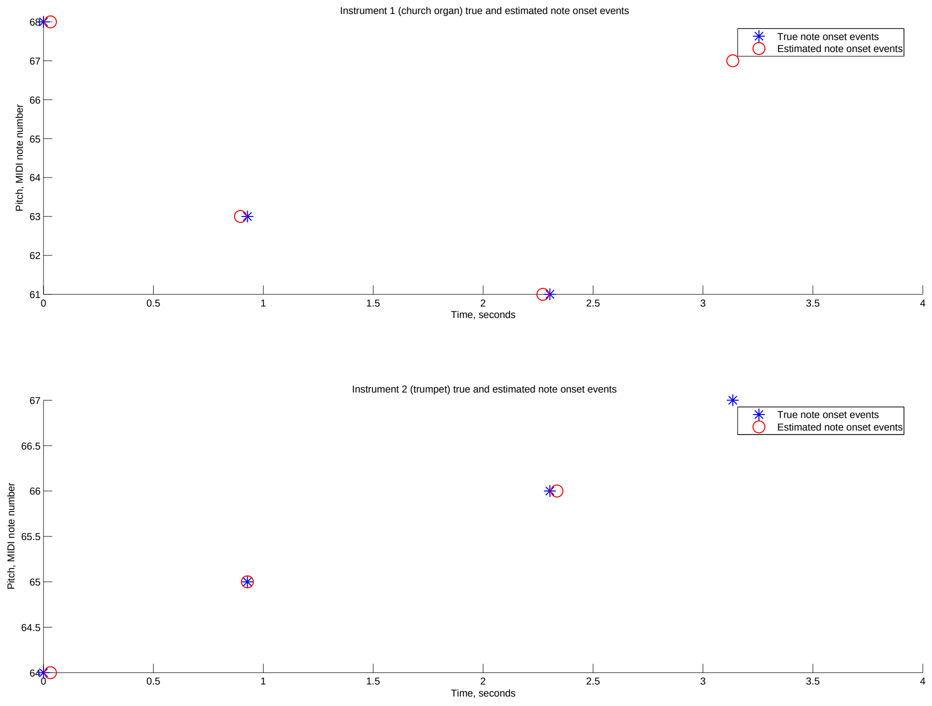

Figure 3.25: Transcription results for synthesized church organ and trumpet sounds (clip 2). Instrument 1 corresponds to the church organ, and Instrument 2 corresponds to the trumpet.

one note of each instrument is simultaneously played. Notes played with simultaneous onsets can be difficult even for a musician to transcribe.

In this experiment, we use the constant envelope transition model from Figure 3.9 for instrument 1 (violin). We use the decaying envelope transition model from Figure 3.10 for instrument 2 (piano). Our expectation is that the decaying envelope model is closer to being a reasonable transition model for a piano than a violin.

Figure 3.27 shows the transcription results. Figure 3.27 shows the intensity envelope estimation results. The transcription error rate on clip 3 was 93.75%. There were 16 notes total, with 1 substitution, 14 deletions, and no insertions. We believe that the poor performance on clip 3 reflects a poor choice of parameters for the short time scale. However, we our note event model is not well-suited to modeling short duration notes with rapid re-articulations. In Chapter 4 we will implement a more reasonable model for sequences

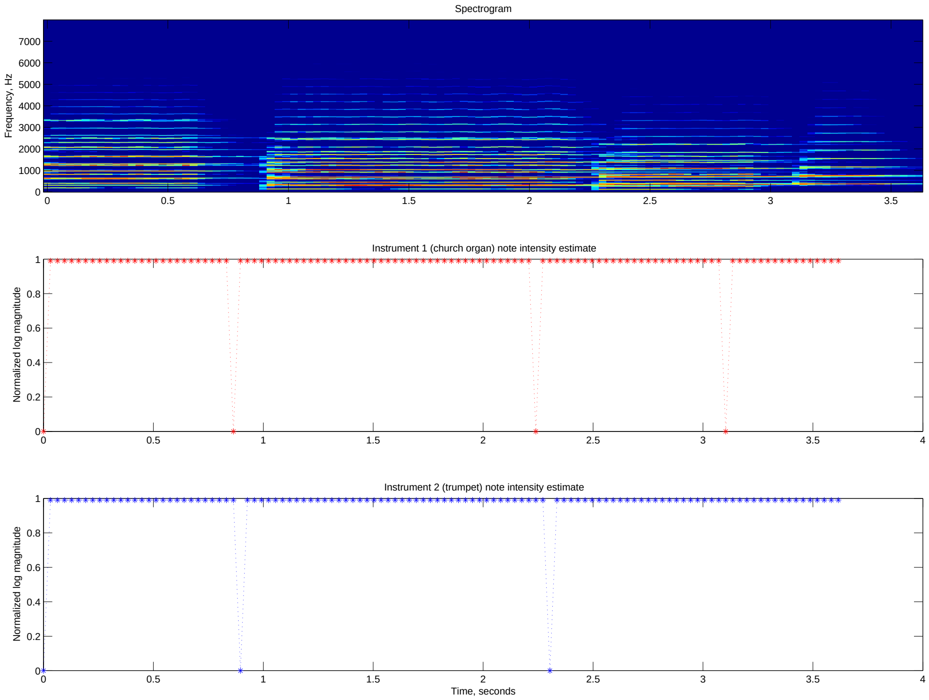

Figure 3.26: Estimated note intensity envelopes for synthesized church organ and trumpet sounds (clip 2). Instrument 1 corresponds to the church organ, and Instrument 2 corresponds to the trumpet. The spectrogram of the input audio file is shown at the top. Here, we use the constant envelope model for both instruments.

of notes in which the note onset is not preceded by a region of silence. We implemented the two-instrument version of this model and tested its performance on clip 3, but the transcription error rate was the same.

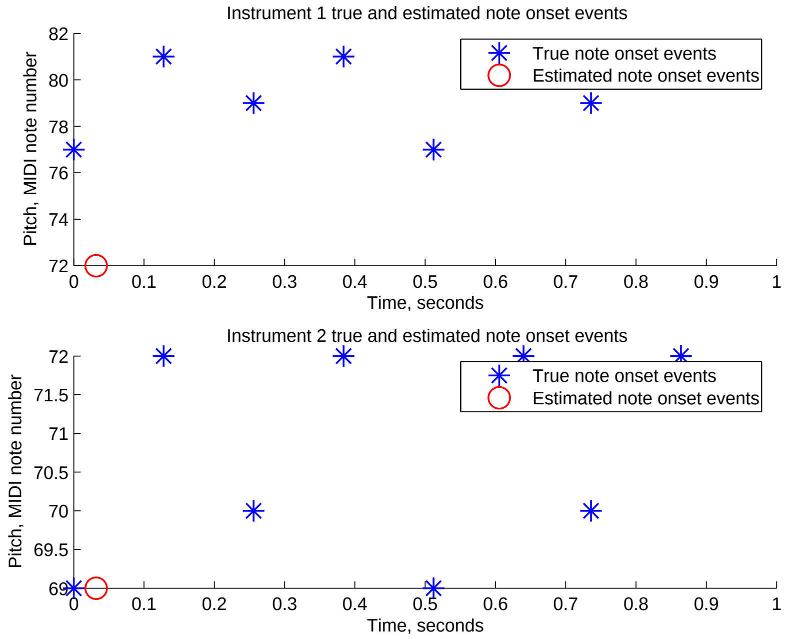

Figure 3.27: Transcription results for the short-duration notes example (clip 3). Instrument 1 corresponds to the violin, and Instrument 2 corresponds to the piano.

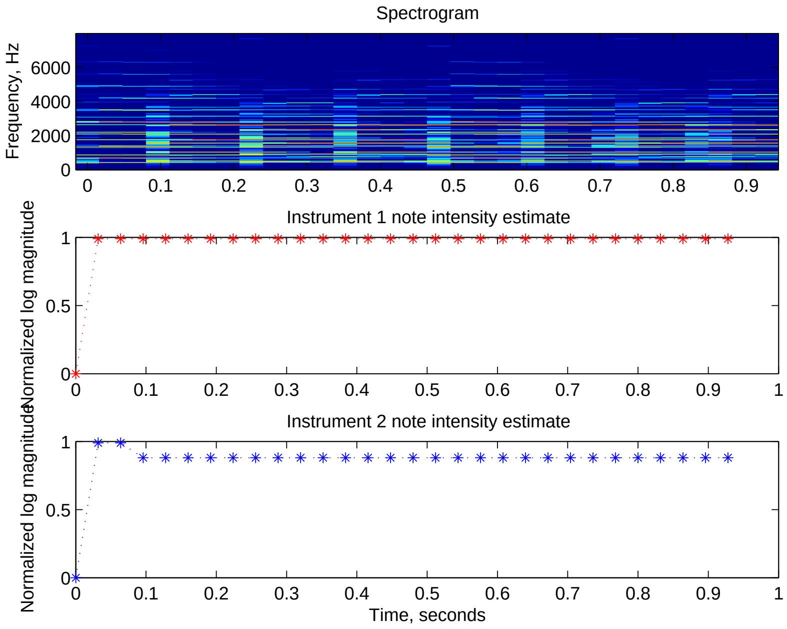

Figure 3.28: Intensity envelope estimation results for the short-duration notes example (clip 3). Instrument 1 corresponds to the violin, and Instrument 2 corresponds to the piano. The spectrogram of the input audio file is shown at the top.

3.13 Conclusions

In this chapter, we have presented a DGM for automated multi-instrument musical transcription and presented results for synthesized instrument sounds. A key feature of our model is a timbre model that includes both a spectral model and a time envelope model, yielding a combined approach to polyphonic transcription and instrument classification. Our model also has the feature that computing posterior modes for the DGM in Figure 3.22 yields explicit note-on events, as well as dynamic level versus time. If we had modeled the intensity state as being continuous-valued, some kind of post-processing would be required to estimate the note-on events.

We have observed in our experiments that the transcription results tend to be strongly influenced by the observation model, while the Markov chain transition probabilities tend to have less of an effect provided that all of transition probabilities are nonzero. The results for the HMM in Section 3.7 and for the FHMM in Section 3.8 are similar to the results for the corresponding time-independent graphical models. Also, the particular choice of pitch transition model (pitch line vs pitch helix) has little effect on performance. The models become more interesting, however, when we only allow certain state transitions. The DGM in Section 3.12 implemented a note event model that disallowed certain physically unrealizable pitch and intensity combinations, allowing for the explicit modeling of note events.

Our models extends immediately to the case of more than two instruments. However, the complexity for exact inference in a DGM is exponential in the number of hidden nodes in a time slice for our class of models. Specifically, for our DGM the time complexity is $O(TMK^{M+1})$ where T is the number of time slices, M is the number of hidden nodes per time slice, and K is the number of states for a hidden node. Thus, if we restrict ourselves to exact inference, the model is limited in practice to a small number of instruments. However, there is a large literature on algorithms for approximate posterior inference in large-scale dynamic graphical models [Mur02]; these algorithms are directly applicable to our model.

Chapter 4

Dynamic Graphical Models for Guitar Transcription Using a Hexaphonic Pickup

4.1 Introduction

The transcription problem that we consider in this chapter assumes the use of an electric guitar fitted with a hexaphonic pickup. An electric guitar has six strings and commonly uses a pickup with a single audio output channel. Each string has a pitch range of 23 semitones including the open string. This corresponds to a single instrument transcription problem with six-voice polyphony. A hexaphonic pickup has one audio output for each of the six strings. Neglecting the small amount of crosstalk, a hexaphonic pickup allows us to simplify the transcription problem to six independent monophonic transcription problems, once for each guitar string. This allows our guitar transcription system to transcribe some actual guitar recordings with reasonable performance, provided that the note durations in the input recording are of sufficient duration.

We present an audio to piano roll transcription system for a guitar equipped with a hexaphonic pickup. We will extend our note event model from Chapter 3 to better handle note onsets that are not preceded by silence, which is quite common in guitar music. Although we deal with with single instrument transcription in this chapter, identifying note events, even in a monophonic signal, can still be a difficult problem. This chapter also serves to illustrate the flexibility of our DGM-based modeling approach. It is straightforward to make changes to the note event model while the other components of the DGM, such as the observation model and pitch transition model remain unchanged.

4.2 Guitar technique

We now present a brief summary of some basic guitar technique that is relevant to our transcription work.

1. *Pull-off*

   A pull-off is a technique for playing a descending slur on a single string from a single string pluck. This technique is used to play a descending (in pitch) sequence of notes in which each successive note begins sounding immediately after the previous note. This technique results in a legato sound in which the dynamic level decreases as the notes are played.

   A two-note pull-off can be produced as follows: The player starts with two fingers on different frets of the same string. The player then plucks the string, causing the higher pitched note to sound. The finger on the higher pitched note is then quickly pulled off the string, allowing the lower pitched note to begin sounding.

2. *Hammer-on*

   A hammer-on is basically the opposite of a pull-off. This technique is used to play an ascending slur on a string. To play a hammer-on, the player places a finger on a fret and plucks the string. Then the player quickly places another finger on a higher pitched fret.

3. *Bends*

   The bend technique is used to produce a continuous change in pitch after plucking a fretted note. A bend is obtained by pushing the string along the neck after the string is plucked. The string can also be pushed along the neck before plucking and then smoothly released after the string pluck. This technique can be used to produce a shift in pitch of up to three or four semitones.

4. *Vibrato*

   The vibrato technique is used to produce a sound in which the pitch fluctuates slightly over time around the fretted pitch. Vibrato is produced by pulling and releasing a fretted string along the fret at a relatively constant rate.

4.3 Modeling assumptions

We assume that note pitches in the input signal are well modeled as coming from the discrete pitch set corresponding to the standard E-A-D-G-B-E tuning. That is, we assume that the guitar is in tune and that expressive pitch deviations from the standard tuning such as bends or vibrato are not present in the input signal.

We place no constraints on the total polyphony of the input signal. The combined output from the hexaphonic pickup can therefore contain up to six simultaneously sounding notes. However, we assume that crosstalk is negligible so that the individual hexaphonic pickup output signals can each be assumed monophonic. We observed that there is only a very small amount of crosstalk in recordings from the hexaphonic pickup.

4.4 Model

Our guitar DGM is based on the DGM for monophonic audio to piano roll transcription from Section 3.11 of Chapter 3. Our primary interest in this DGM was in training it on monophonic recordings to learn the parameters for an instrument timbre model. We then constructed the two-instrument DGM in Section 3.12 using these learned parameters. We used the junction tree algorithm to compute the transcription as the mode of the posterior distribution of intensity levels and notes (the Viterbi path).

An actual guitar note event begins with a string pluck or a hammer-on or pull-off, and ends when either the sound intensity decays to zero or a new pluck, hammer-on, or pull-off occurs. If bends and vibrato are disallowed then the pitch remains constant during a note event. In music transcription, note onset events are perceptually more important than note offset events. For this reason we will mainly be concerned with identifying note onset events.

Given a guitar recording, we would like to infer both the onset time and pitch of each note played in the input recording. We now present our signal modeling assumptions for a guitar note event. The following sections will describe how our transcription DGM realizes these modeling assumptions. We define a note onset event as an instantaneous significant increase in sound intensity during which time the pitch can possibly take on a new value. The pitch then remains constant as the intensity gradually decreases until either zero intensity is reached (note offset) or a new note onset event occurs. Furthermore, the pitch must take on a value from the discrete set of pitches corresponding to the fretted and open string pitches of a guitar tuned to the standard E-A-D-G-B-E tuning. We do not attempt to model vibrato or bends.

We have found through empirical observation that crosstalk is not a significant issue, and we therefore chose not to model it. By ignoring crosstalk, we can simplify the transcription problem into six independent monophonic transcription problems. We have found note onset detection to be a more significant issue.

Figure 4.1 shows the guitar transcription DGM for a single string. The complete transcription DGM therefore consists of six independent DGMs (one for each string). Note the similarity between this DGM and the monophonic transcription DGM in Figure 3.21 of Chapter 3.

4.4.1 Pitch transition model

Recall that our definition of a note event consists of an onset, followed by a sustaining or decaying intensity envelope, during which time the pitch remains constant. The note event ends when either the intensity reaches zero, or a new note onset occurs. We would like to enforce these constraints in the graphical model. The intensity transition model in

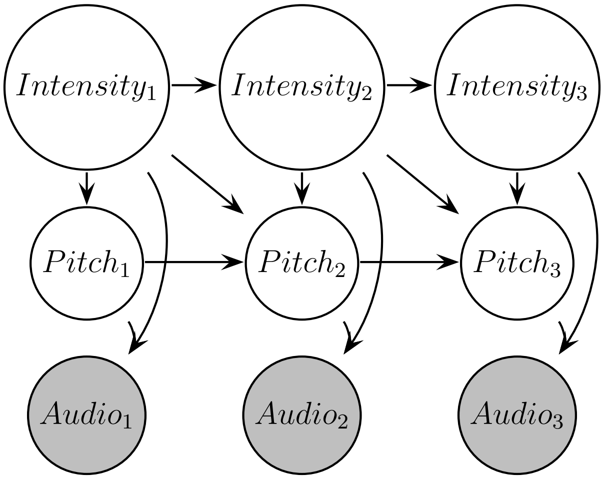

Figure 4.1: A DGM for guitar transcription with a hexaphonic pickup. The transcription model for a single string is shown. The complete transcription model contains six copies of the model shown (one for each string).

Section 4.4.2 places constraints on the types of intensity envelopes the may occur.

We place constraints on the times at which the pitch state can change. Specifically, pitch state change events within a single note event should be disallowed. This is done by making the pitch state conditional probability distribution a function of both the previous pitch state, the previous intensity state, and the current intensity state. In our model, the pitch state is only allowed to change if the arc (k, n) in the state transition diagram beginning in state $Intensity_{t-1} = k$ and ending in state $Intensity_t = n$ is in the set $Onsets$ of note onset arcs. We use the following pitch transition model for the guitar:

$$\begin{aligned} & P(Pitch_t = j | Intensity_{t-1} = k, Intensity_t = n, Pitch_{t-1} = i) \\ &= \begin{cases} \delta(i,j) & \text{if } (k,n) \notin Onsets \text{ (stay in the same state)} \\ T^m(i,j) & \text{if } (k,n) \in Onsets \text{ (pitch transition)} \end{cases} \end{aligned} \tag{4.1}$$

where the $Intensity^m_{t-1} = 0$ state denotes the "note-off" state. $T^m(i,j)$ represents the pitch transition model for whole note events for instrument m. That is, $T^m(i,j) = P(Pitch^m_t = j | Pitch^m_{t-1} = i)$. We choose to make $T^m(i,j)$ instrument specific to reflect the fact that the set of allowable pitches can depend on the instrument. In our current implementation, we set $T^1(i,j) = T^2(i,j)$ so that instrument classification performance only depend on the intensity transition model and the observation model (and not on the pitch ranges of the instruments).

4.4.2 Intensity transition model

The timbre of a musical instrument is influenced by both the spectral content, and the way in which the sound level changes over time. In the guitar and other plucked string instruments, the sound level immediately begins decaying after the initial hammer strike. We propose an intensity envelope model for the guitar that is similar to the "decaying envelope model" from Section 3.9.2.

Recall that in the decaying envelope model, a valid envelope consisted of an onset transition from the note-off state (zero intensity) to any intensity state level. Each successive transition then consisted of either a self-loop to the same intensity level, or a transition to the next-lower intensity state, until the zero-intensity state was reached. This seems to be a reasonable model for the case of a recording in which the note intensities always decay away completely before new notes are played. However, it can be common in music a new note onset to occur while the preceding note is still sounding. This is particularly common in guitar music, where the hammer-on or pull-off technique is used.

We now specify an intensity envelope model for the guitar that allows a note onset event to occur while the preceding note is still sounding (i.e., the starting intensity is nonzero). The decaying envelope model for the guitar is shown in Figure 4.2. We define an onset transition to be any transition from a lower intensity state to a higher intensity state. We define an offset transition to be any transition such that the end state is the note-off (zero intensity) state. In this model, onset transitions from the note-off state to any nonzero intensity level are allowed. Self-loop transitions are allowed on all states. Transitions from any nonzero intensity state to the next lower intensity state are allowed. Onset transitions from a nonzero intensity state to a higher intensity state are allowed (e.g., level 2 $\rightarrow$ level 4 onset), provided that some minimum intensity change threshold is exceeded.

Thus, a realization of this state transition model will result in an onset transition to some nonzero intensity level, followed by some number of self loops, followed by a transition to the next lower intensity state, and so on, until either the note-off state is reached or another onset transition occurs.

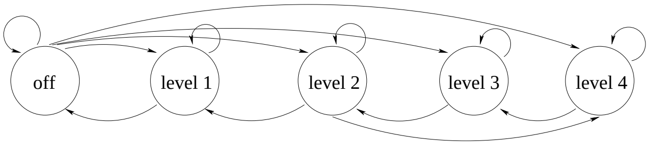

Figure 4.2: A state transition diagram for an instrument such as a guitar that is characterized by a decaying sound level after the note onset. Note onset events that start at nonzero intensity are allowed (e.g., level 2 → level 4 onset), provided that some minimum intensity change threshold is exceeded. The "off" state denotes zero intensity.

It is then straightforward to construct the state transition matrix $p(Intensity_t = j|Intensity_{t-1} = i) = Intensity\_CPT(i, j)$ by setting all entries $Intensity\_CPT(i, j)$ for which there is no arc from state i to state j in the state transition diagram to zero. The remaining entries can be specified manually or learned from training data. This approach to modeling the intensity envelope allows us to experiment with different intensity envelope constrains by simply editing the the entries of $Intensity\_CPT$.

4.4.3 Observation model

We use the observation model from Section 3.9.4, which we repeat here. We use the following Gaussian process model for the spectrum at time slice t:

$$|X_t(f)| = Instrument_t(f) + \xi(f) \tag{4.2}$$

where

$$Instrument_t(f) = Intensity_t \sum_{h=1}^{H} \alpha_h(f_{pitch}) B(f - hf_{pitch}) \tag{4.3}$$

Conditional on the hidden $Intensity_t$ and $Pitch_t$ state variables, we then have the following Gaussian observation mode:

$$p(Audio_t|Intensity_t, Pitch_t) = \mathcal{N}(Audio_t|\mu(Intensity_t Pitch_t), \sigma_\xi^2 I)$$

where $\mu(Intensity_t, Pitch_t) = [\mu_1, ..., \mu_N]^T$ and $\mu_i = Instrument_t(f_i)$.

We must also specify a mapping from the (discretized) hidden intensity state variable to the $Intensity_t$ parameter in Equation 4.3. We propose that the intensity state values correspond to intensity levels that are discretized uniformly in log magnitude over the effective dynamic range of the input signal. The smallest intensity level is then considered to correspond to the note-off state.

4.5 Experiments

We now present some transcription results on hexaphonic recordings of guitar sounds. The intensity transition probabilities, observation noise variance σ_ξ^2, and the $\alpha_h(f_{pitch})$ harmonic magnitude parameters for each instrument are learned by an EM-based estimation procedure on the single-string DGM in Figure 4.1. Each single-string DGM was trained on a recording of guitar sounds played on the corresponding string. The training data consisted of chromatic scales played over the entire pitch range of the corresponding strings. The durations of the training notes were approximately three seconds per note with a brief period of silence between successive notes.

We use 15 discrete intensity levels, discretized uniformly in log magnitude over a 60 dB dynamic range. The lowest intensity state corresponds to the note-off state. The pitch range is taken to be the complete pitch range of the corresponding string (23 semitones including the open string).

4.5.1 Hammer-on example

In this section, we present transcription results on a recording of several two-note groups played with a hammer-on technique on the D-string. We show how different DGMs corresponding to different note event modeling assumptions affect the transcription results.

HMM pitch tracking results

We start with the HMM pitch tracker from Section 3.7. Recall that this model contains a single Markov chain for modeling the pitch and that intensity level is not modeled. Figure 4.3 shows the pitch tracking results. The true note onset events are denoted by blue circles and the pitch estimates are denoted by the red stars.

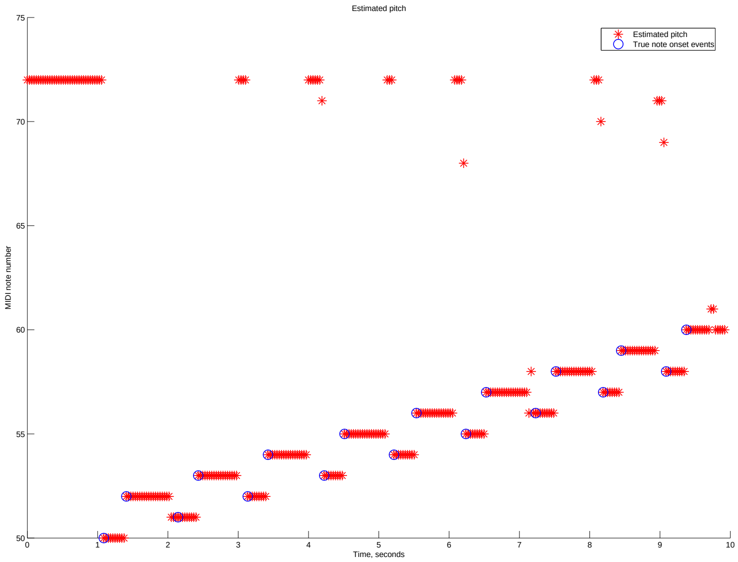

Figure 4.3: Pitch tracking results for the hammer-on example using the HMM in Section 3.7. The true note onset events are denoted by blue circles and the pitch estimates are denoted by the red stars.

We observe that the pitch estimates are generally in agreement with the pitch of the corresponding truth score onsets. However, we see that there are some erroneous pitch estimates near the onsets.

FHMM pitch tracking results

We now present pitch tracking results using the FHMM pitch tracker from Section 3.9. Figure 4.4 shows the pitch tracking results. The true note onset events are denoted by blue circles and the pitch estimates are denoted by the red stars. Figure 4.5 shows the corresponding intensity envelope estimation results.

We observe that the pitch estimates are generally in agreement with the pitch of the corresponding truth score onsets. However, we see that there are some erroneous pitch estimates near the onsets. There are fewer pitch errors compared to the HMM pitch tracker. Observe that the FHMM identified a single note onset corresponding to the first note played. The intensity envelope has significant energy during the playing of all 18 notes. This seems reasonable considering that the notes were played quickly enough that the intensity did not

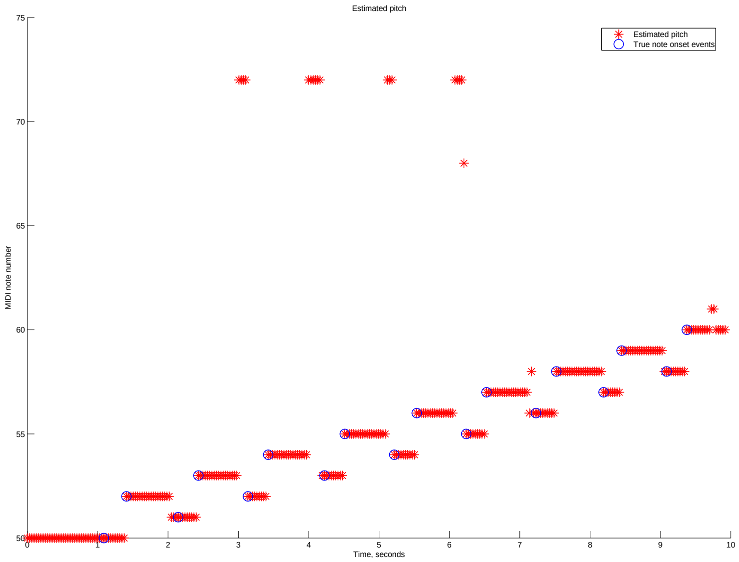

Figure 4.4: Pitch tracking results for the hammer-on example using the FHMM from Section 3.9. The true note onset events are denoted by blue circles and the pitch estimates are denoted by the red stars.

decay significantly from one note to the next.

Figure 4.6 shows the onset estimation results. If we interpret the intensity onsets as being note event onsets (ignoring pitch state changes), we would have a transcription error of 94.4 %. There were 18 notes total, with 17 deletions, 0 insertions, and 0 substitutions. Since we would like to associate intensity onset events with note onset events, this motivates using a model in which the pitch cannot change within a single intensity envelope.

DGM transcription results

We now present transcription results using the DGM from Section 3.11. Figure 4.7 shows the true and estimated note onset events. Figure 4.8 shows the corresponding intensity envelope estimate. The transcription error rate was 27.8%. There were 18 notes total, with 5 deletions, 0 insertions, and 0 substitutions.

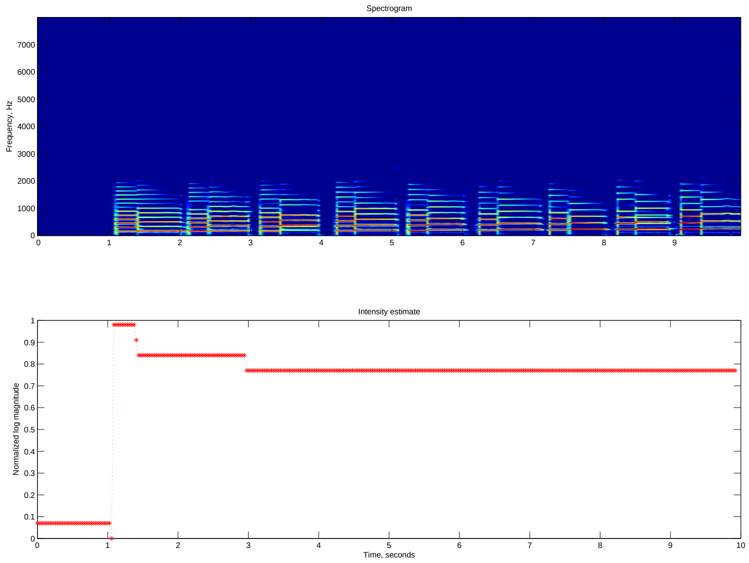

Figure 4.5: Intensity envelope estimation results for the hammer-on example using the FHMM from Section 3.9. The spectrogram of the input audio file is shown at the top.

Guitar DGM transcription results

We now present transcription results using the guitar DGM in Figure 4.1. Figure 4.9 shows the true and estimated note onset events. Figure 4.10 shows the corresponding intensity envelope estimate. The transcription error rate was 0 %. We observe that this is a significant improvement of the transcription results using the DGM from Section 3.11.

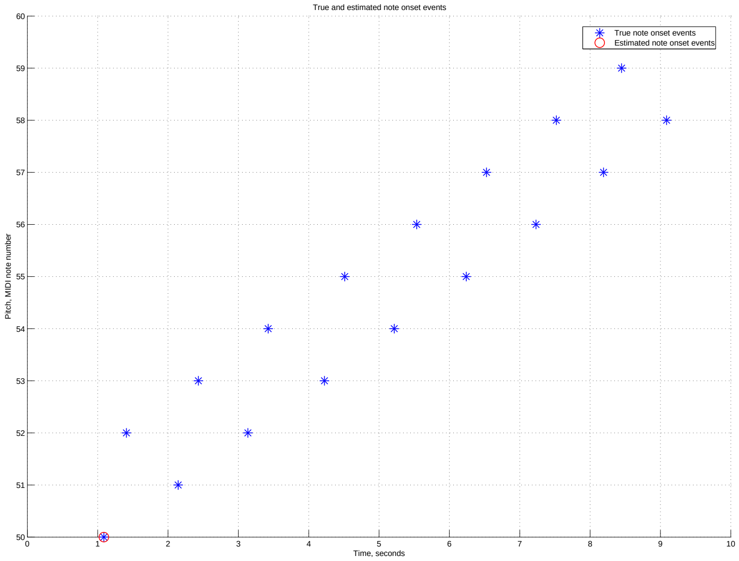

Figure 4.6: Onset estimation results for the hammer-on example using the FHMM from Section 3.9.

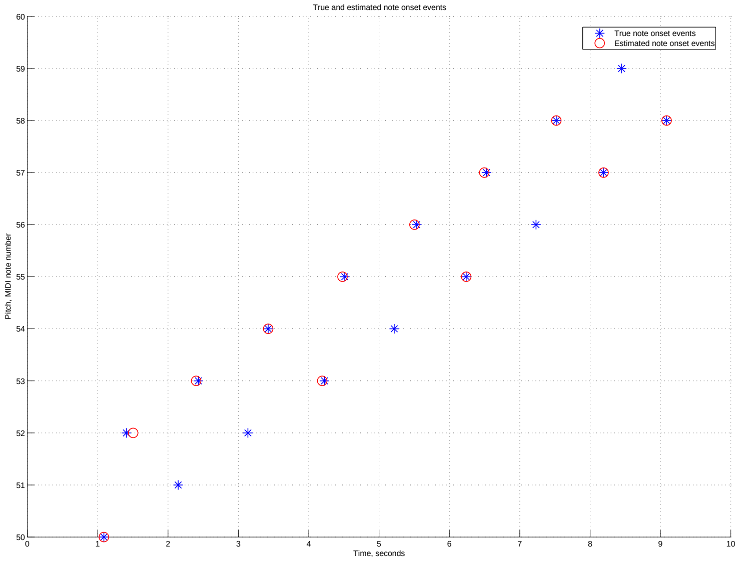

Figure 4.7: Transcription results for the hammer-on example using the DGM from Section 3.11. Here we show the true and estimated note onsets.

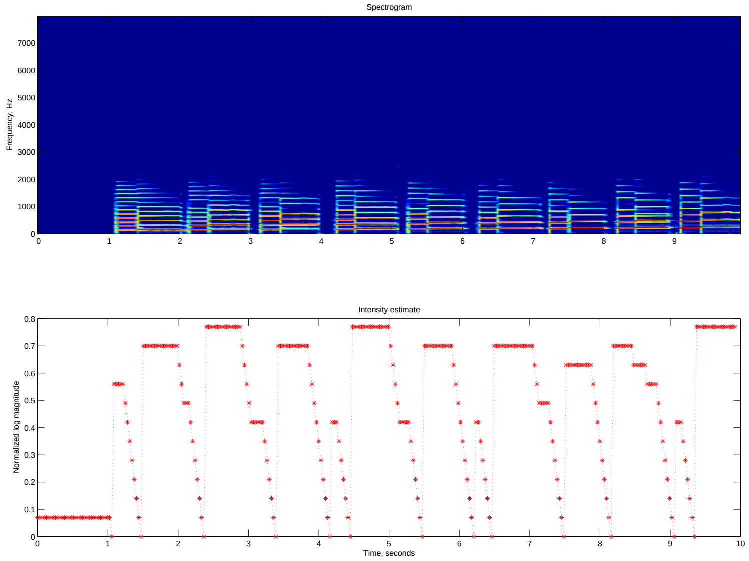

Figure 4.8: Transcription results for the hammer-on example using DGM from Section 3.11. Here we show the estimated note intensity envelope. The spectrogram of the input audio file is shown at the top.

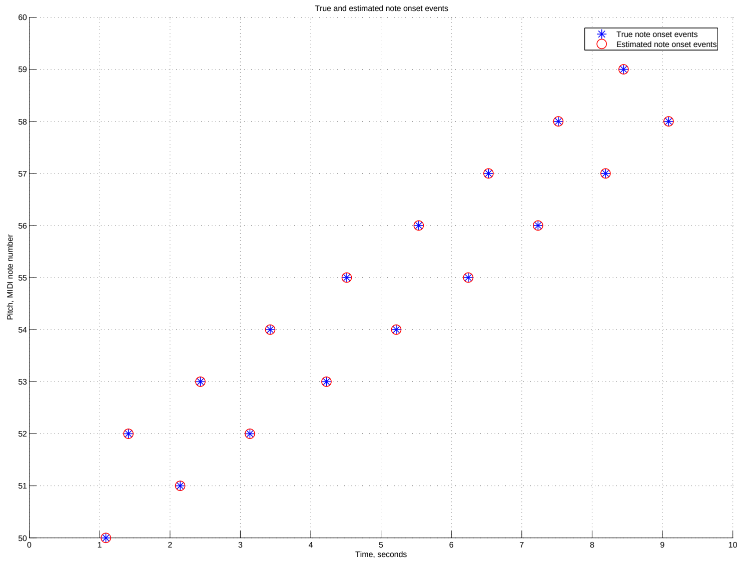

Figure 4.9: Transcription results for the the hammer-on example using the guitar DGM in Figure 4.1. Here we show the true and estimated note onsets.

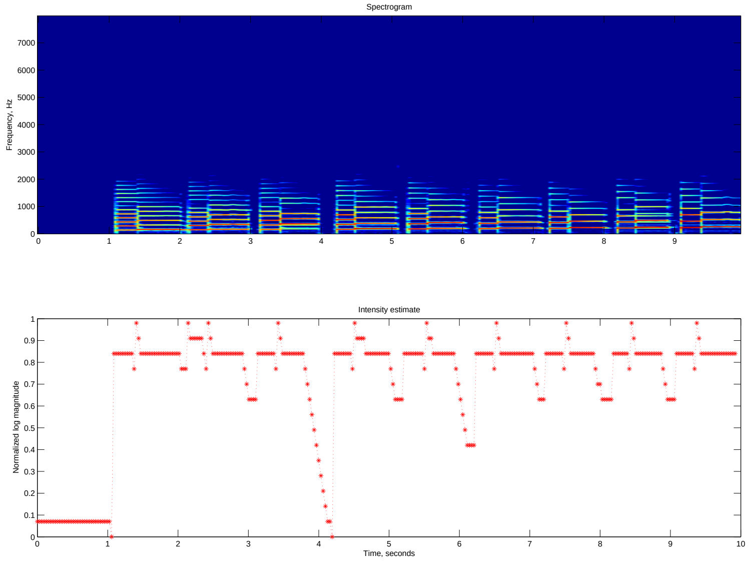

Figure 4.10: Transcription results for the hammer-on example using the guitar DGM in Figure 4.1. Here we show the estimated note intensity envelope. The spectrogram of the input audio file is shown at the top.

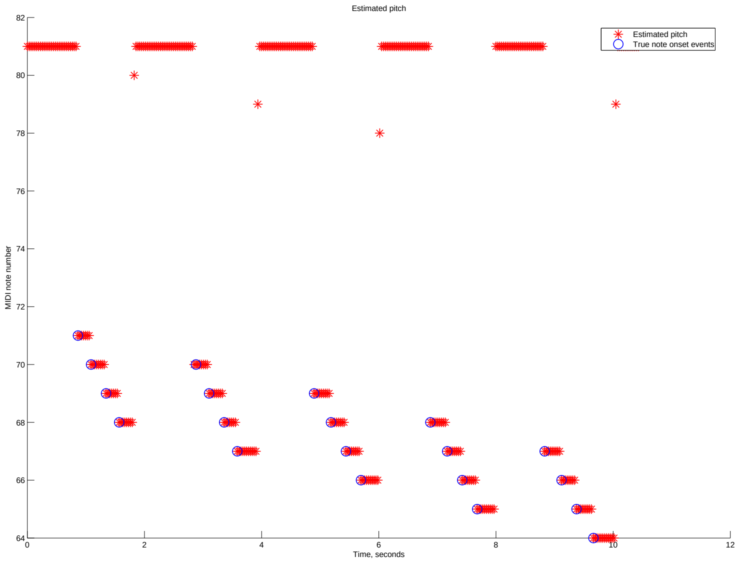

Figure 4.11: Pitch tracking results for the pull-off example using the HMM in Section 3.7. The true note onset events are denoted by blue circles and the pitch estimates are denoted by the red stars.

4.5.2 Pull-off example

In this section, we present transcription results on a recording containing several four-note groups played using the pull-off technique on the B-string. We show how different DGMs corresponding to different note event modeling assumptions affect the transcription results.

HMM pitch tracking results

We start with the HMM pitch tracker from Section 3.7. Recall that this model contains a single Markov chain for modeling the pitch and that intensity level is not modeled. Figure 4.11 shows the pitch tracking results. The true note onset events are denoted by blue circles and the pitch estimates are denoted by the red stars.

We observe that the pitch estimates are generally in agreement with the pitch of the corresponding truth score onsets. However, we see that there are some erroneous pitch estimates near the onsets. Pitch errors tend to occur when the intensity level is low, during the regions of silence separating the pull-off groups.

FHMM pitch tracking results

We now present pitch tracking results using the FHMM pitch tracker from Section 3.9. Figure 4.12 shows the pitch tracking results. The true note onset events are denoted by blue circles and the pitch estimates are denoted by the red stars. Figure 4.13 shows the corresponding intensity envelope estimation results.

We observe that the pitch estimates are generally in agreement with the pitch of the corresponding truth score onsets. However, we see that there are some erroneous pitch estimates near the onsets. As in the HMM pitch tracker, pitch errors tend to occur when the intensity level is low, during the regions of silence separating the pull-off groups.

We observe that the FHMM identified five note onsets, corresponding to the first note played of each pull-off group. This seems reasonable since the intensity envelope has significant energy during the playing of all 4 notes in each group.

Figure 4.14 shows the onset estimation results. Only the pull-off pluck events are identified. If we interpret the intensity onsets as being note event onsets (ignoring pitch state changes), we would have a transcription error of 75.0 %. There were 20 notes total, with 15 deletions, 0 insertions, and 0 substitutions.

DGM transcription results

We now present transcription results using the DGM from Section 3.11. Figure 4.15 shows the true and estimated note onset events. Figure 4.16 shows the corresponding intensity envelope estimate. The transcription error rate was 40.0%. There were 20 notes total, with 8 deletions, 0 insertions, and 0 substitutions. This is still an improvement of the FHMM, where only the onset of each pull-off group was detected.

Guitar DGM transcription results

We now present transcription results using the guitar DGM in Figure 4.1. Figure 4.17 shows the true and estimated note onset events. Figure 4.18 shows the corresponding intensity envelope estimate. The transcription error rate was 0 %. We observe that this is a significant

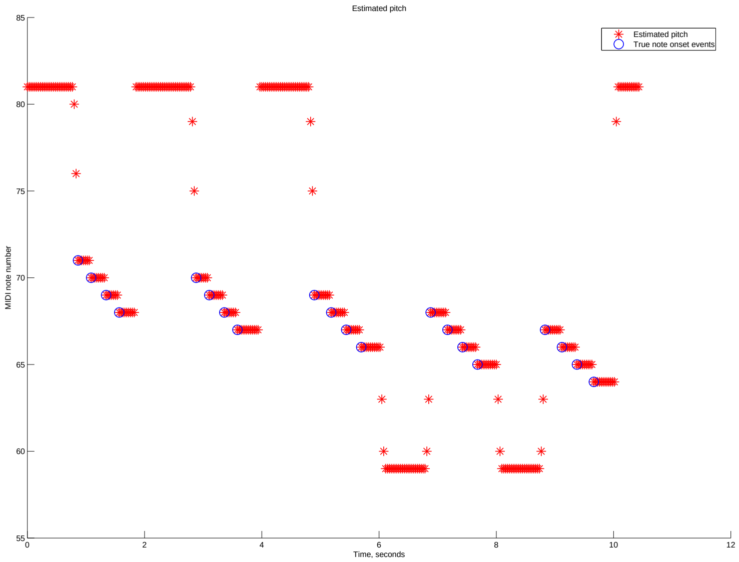

Figure 4.12: Pitch tracking results for the pull-off example using the FHMM from Section 3.9. The true note onset events are denoted by blue circles and the pitch estimates are denoted by the red stars.

improvement of the transcription results using the DGM from Section 3.11.

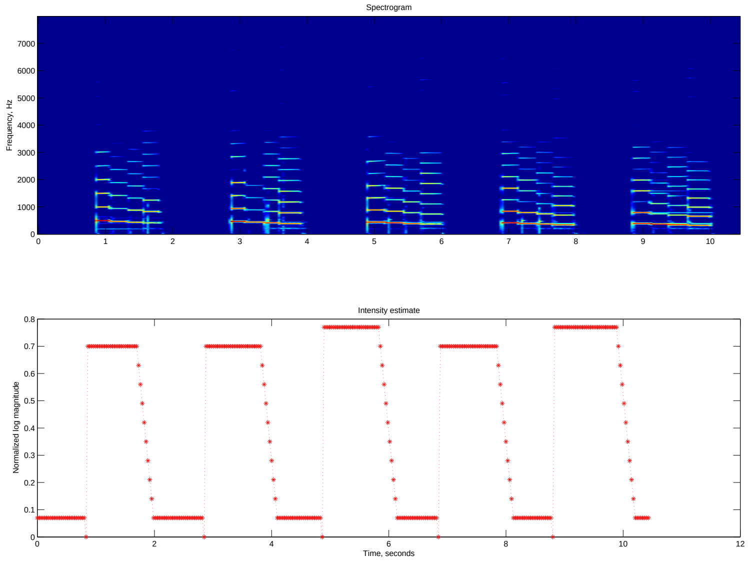

Figure 4.13: Intensity envelope estimation results for the pull-off example using the FHMM from Section 3.9. The spectrogram of the input audio file is shown at the top.

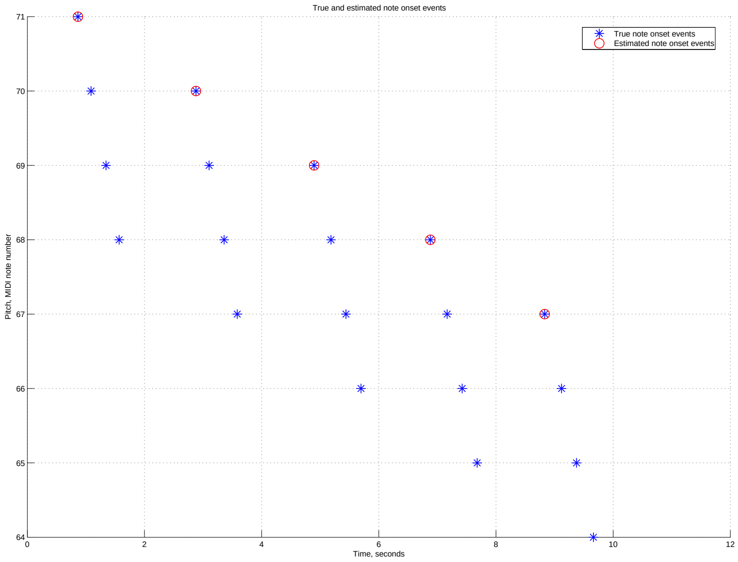

Figure 4.14: Onset estimation results for the pull-off example using the FHMM from Section 3.9. The spectrogram of the input audio file is shown at the top.

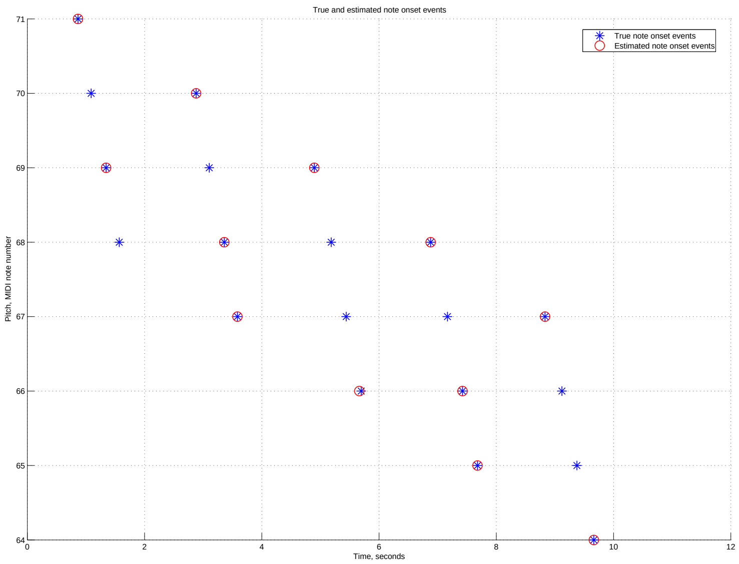

Figure 4.15: Transcription results for the pull-off example using the DGM from Section 3.11. Here we show the true and estimated note onsets.

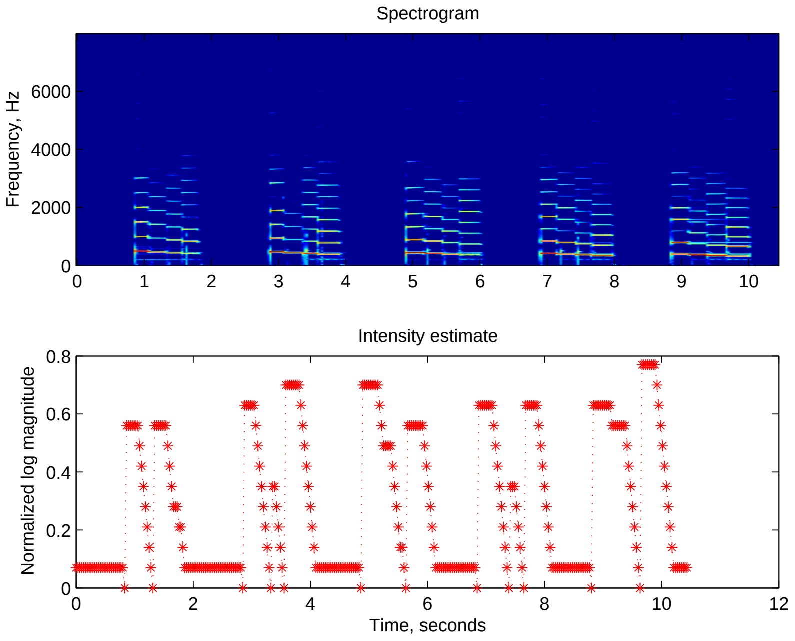

Figure 4.16: Transcription results for the DGM from Section 3.11. Here we show the estimated note intensity envelope. The spectrogram of the input audio file is shown at the top.

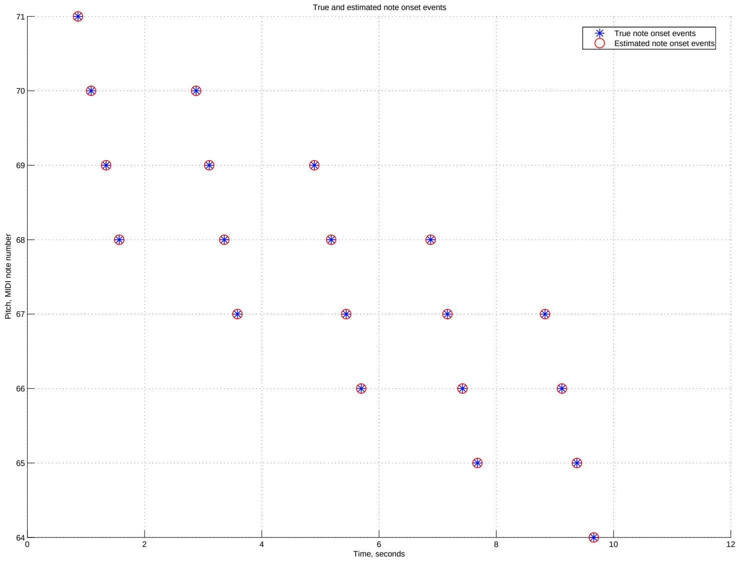

Figure 4.17: Transcription results for the pull-off example using the guitar DGM in Figure 4.1. Here we show the true and estimated note onsets.

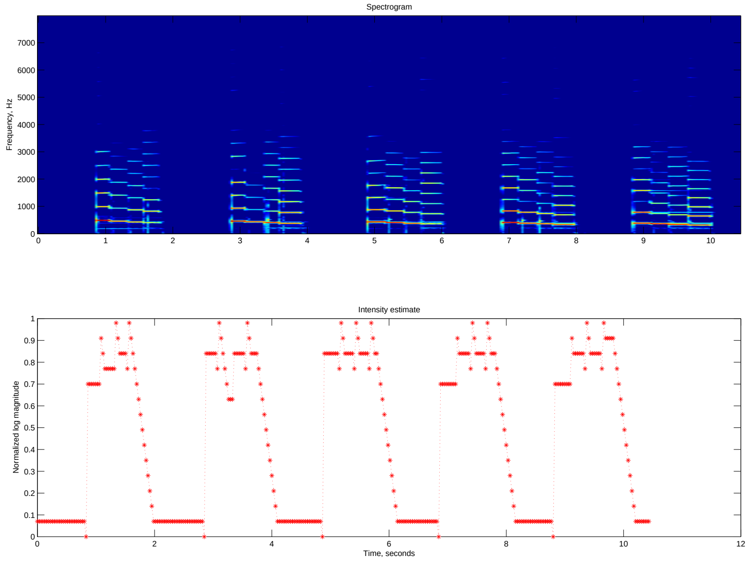

Figure 4.18: Transcription results for the pull-off example using the guitar DGM in Figure 4.1. Here we show the estimated note intensity envelope. The spectrogram of the input audio file is shown at the top.

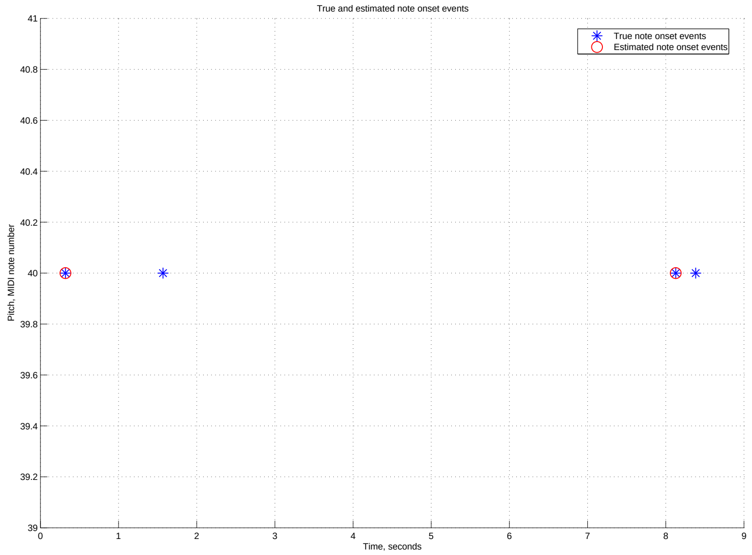

Figure 4.19: String 1 transcription results for the hexaphonic pickup example using the guitar DGM in Figure 4.1. Here we show the true and estimated note onsets.

4.5.3 Hexaphonic pickup example

We now present transcription results on data from the hexaphonic pickup using the guitar DGM in Figure 4.1. The input recording contains several strums with six note polyphony. Our results indicate that crosstalk is note an issue. We present results for each string.

String 1: low E-string

Figure 4.19 shows the true and estimated note onset events. Figure 4.20 shows the corresponding intensity envelope estimate. The transcription error rate for this string was 50.0%. There were 4 notes total with 2 deletions, 0 insertions, and 0 substitutions. We observe that the deleted notes were of the same pitch as the preceding notes and that the preceding notes still had nonzero intensity when the new onsets occurred.

String 2: A-string

Figure 4.21 shows the true and estimated note onset events. Figure 4.22 shows the corresponding intensity envelope estimate. The transcription error rate for this string was 33.3%. There were 3 notes total with 1 deletion, 0 insertions, and 0 substitutions. We observe that

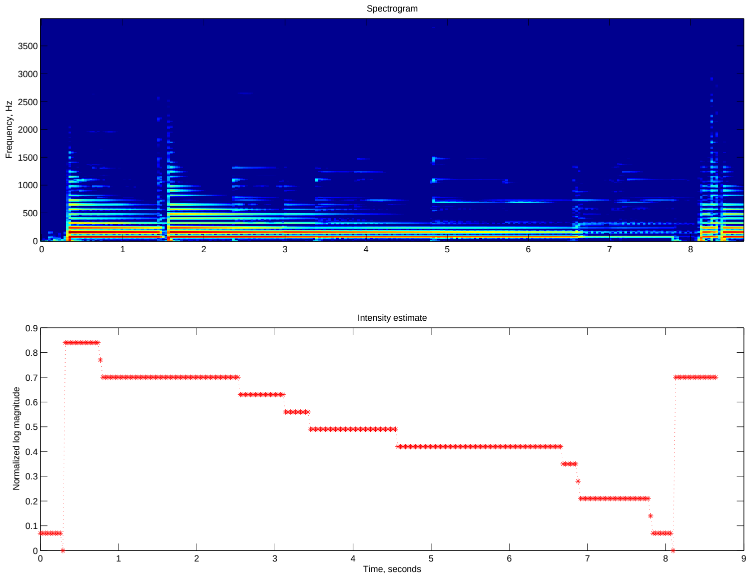

Figure 4.20: String 1 transcription results for the hexaphonic pickup example using the guitar DGM in Figure 4.1. Here we show the estimated note intensity envelope. The spectrogram of the input audio file is shown at the top.

the deleted notes were of the same pitch as the preceding notes and that the preceding notes still had nonzero intensity when the new onsets occurred.

String 3: D-string

Figure 4.23 shows the true and estimated note onset events. Figure 4.24 shows the corresponding intensity envelope estimate. The transcription error rate for this string was 42.9%. There were 7 notes total with 3 deletions, 0 insertions, and 0 substitutions. We observe that the deleted notes were of the same pitch as the preceding notes and that the preceding notes still had nonzero intensity when the new onsets occurred.

String 4: G-string

Figure 4.25 shows the true and estimated note onset events. Figure 4.26 shows the corresponding intensity envelope estimate. The transcription error rate for this string was 20.0%. There were 5 notes total with 1 deletions, 0 insertions, and 0 substitutions. We observe that the deleted notes were of the same pitch as the preceding notes and that the preceding notes

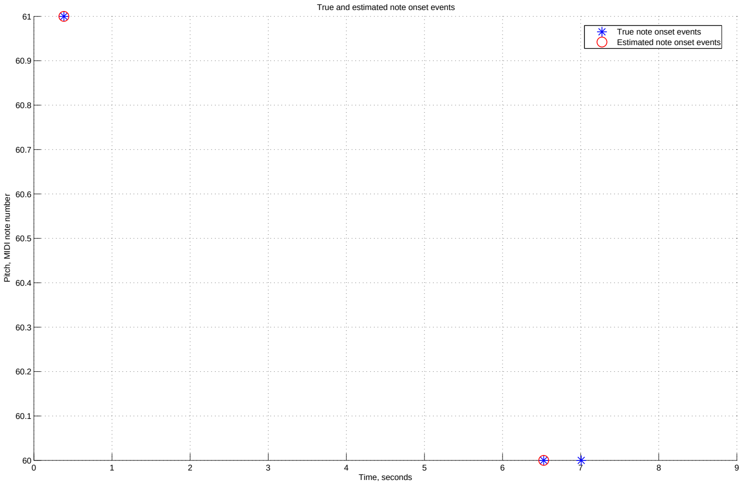

Figure 4.21: String 2 transcription results for the hexaphonic pickup example using the guitar DGM in Figure 4.1. Here we show the true and estimated note onsets.

still had nonzero intensity when the new onsets occurred.

String 5: B-string

Figure 4.27 shows the true and estimated note onset events. Figure 4.28 shows the corresponding intensity envelope estimate. The transcription error rate for this string was 16.6%. There were 6 notes total with 1 deletions, 0 insertions, and 0 substitutions. We observe that the deleted notes were of the same pitch as the preceding notes and that the preceding notes still had nonzero intensity when the new onsets occurred.

String 6: high E-string

Figure 4.29 shows the true and estimated note onset events. Figure 4.30 shows the corresponding intensity envelope estimate. The transcription error rate for this string was 14.3%. There were 7 notes total with 1 deletions, 0 insertions, and 0 substitutions. We observe that the deleted notes were of the same pitch as the preceding notes and that the preceding notes still had nonzero intensity when the new onsets occurred.

The overall transcription error (across all six strings) was 25.0 %. There were 32 notes total with 8 deletions, 0 substitutions, and 0 insertions. We observe that all of the deletions

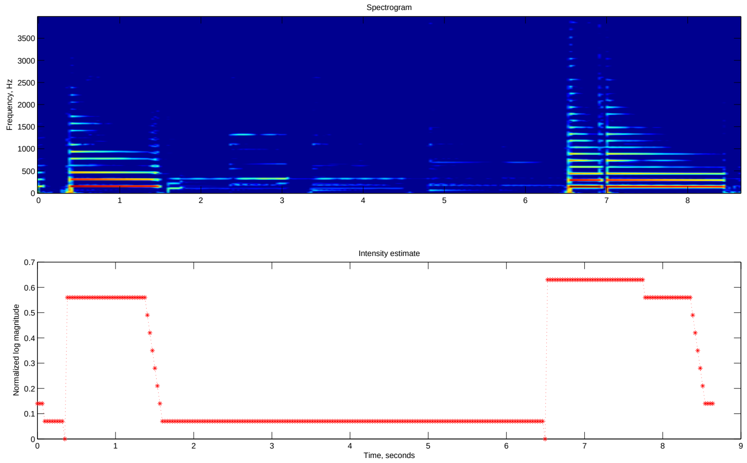

Figure 4.22: String 2 transcription results for the hexaphonic pickup example using the guitar DGM in Figure 4.1. Here we show the estimated note intensity envelope. The spectrogram of the input audio file is shown at the top.

consisted of repeated notes.

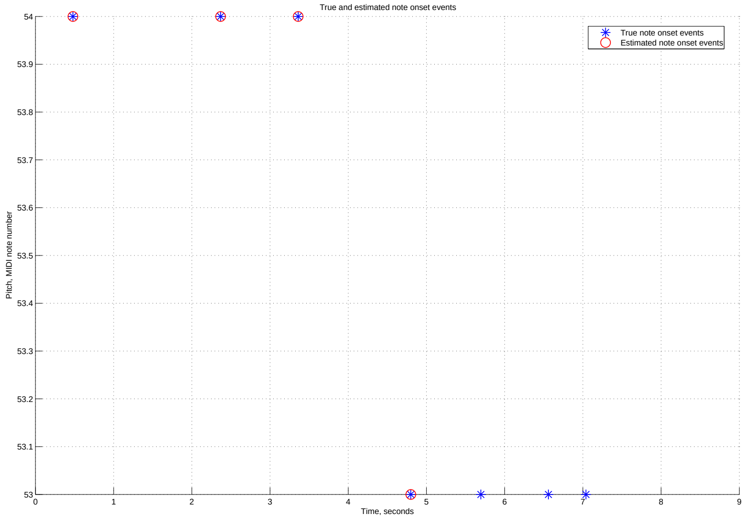

Figure 4.23: String 3 transcription results for the hexaphonic pickup example using the guitar DGM in Figure 4.1. Here we show the true and estimated note onsets.

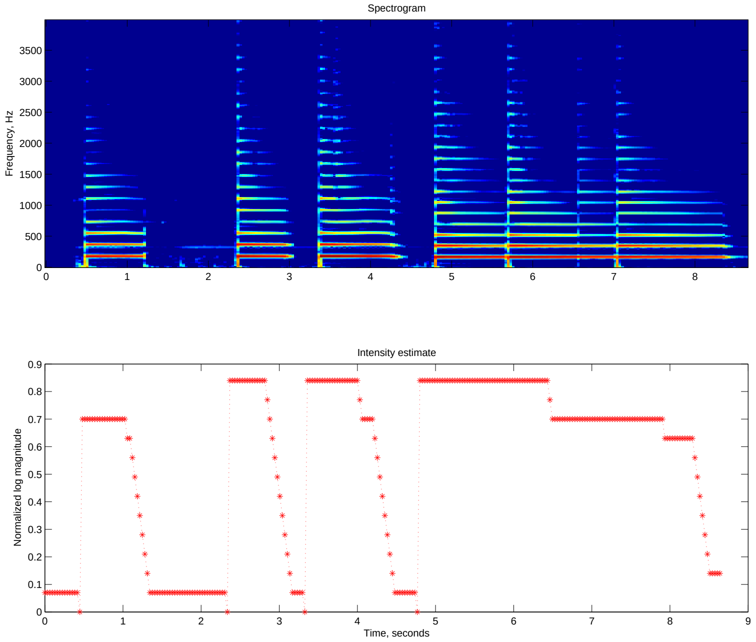

Figure 4.24: String 3 transcription results for the hexaphonic pickup example using the guitar DGM in Figure 4.1. Here we show the estimated note intensity envelope. The spectrogram of the input audio file is shown at the top.

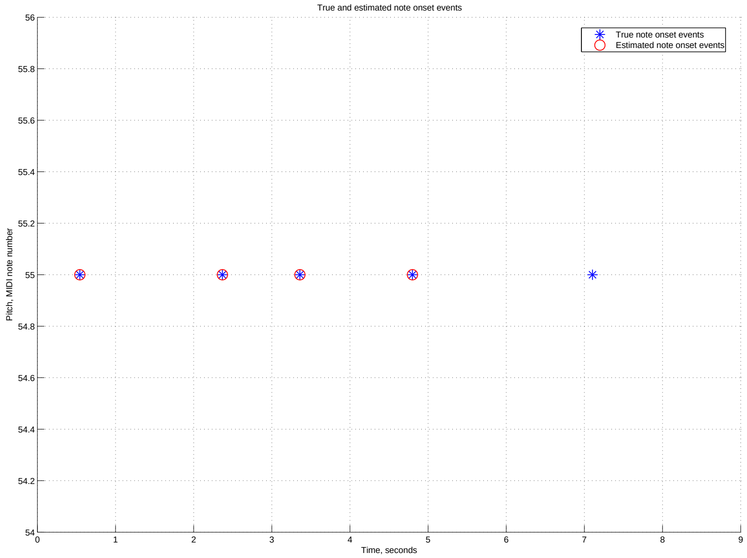

Figure 4.25: String 4 transcription results for the hexaphonic pickup example using the guitar DGM in Figure 4.1. Here we show the true and estimated note onsets.

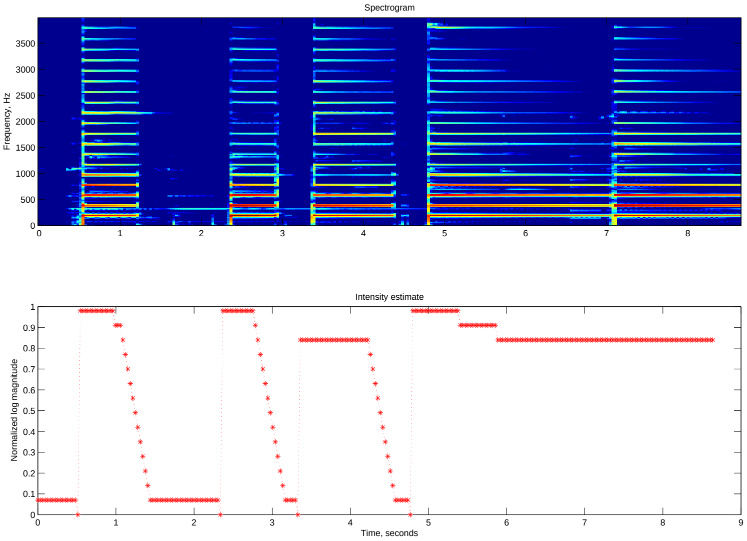

Figure 4.26: String 4 transcription results for the hexaphonic pickup example using the guitar DGM in Figure 4.1. Here we show the estimated note intensity envelope. The spectrogram of the input audio file is shown at the top.

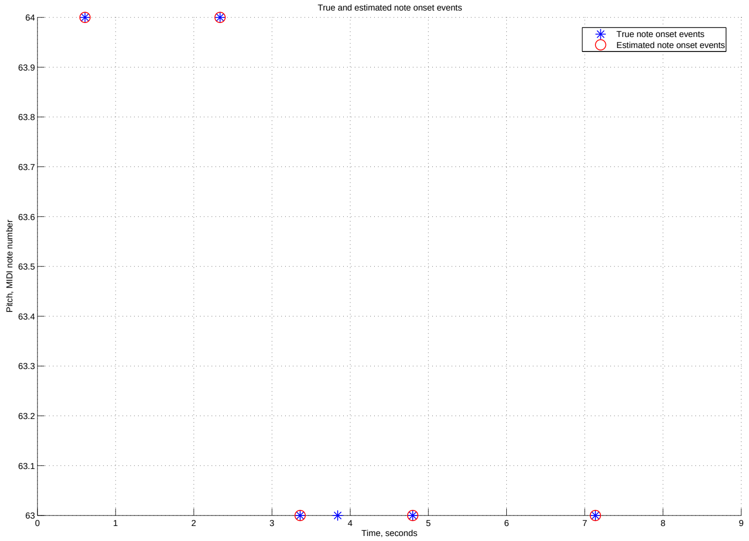

Figure 4.27: String 5 transcription results for the hexaphonic pickup example using the guitar DGM in Figure 4.1. Here we show the true and estimated note onsets.

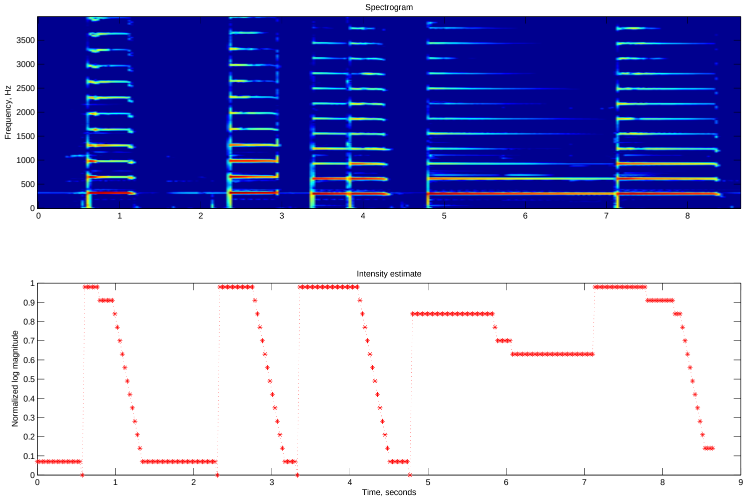

Figure 4.28: String 5 transcription results for the hexaphonic pickup example using the guitar DGM in Figure 4.1. Here we show the estimated note intensity envelope. The spectrogram of the input audio file is shown at the top.

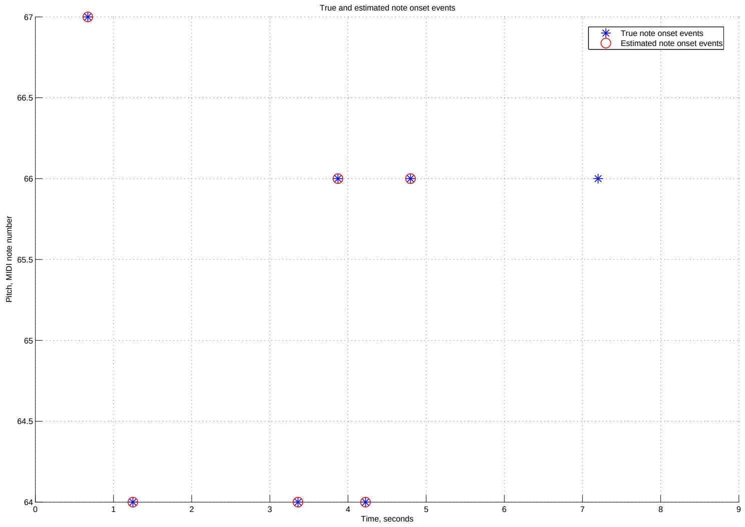

Figure 4.29: String 6 transcription results for the hexaphonic pickup example using the guitar DGM in Figure 4.1. Here we show the true and estimated note onsets.

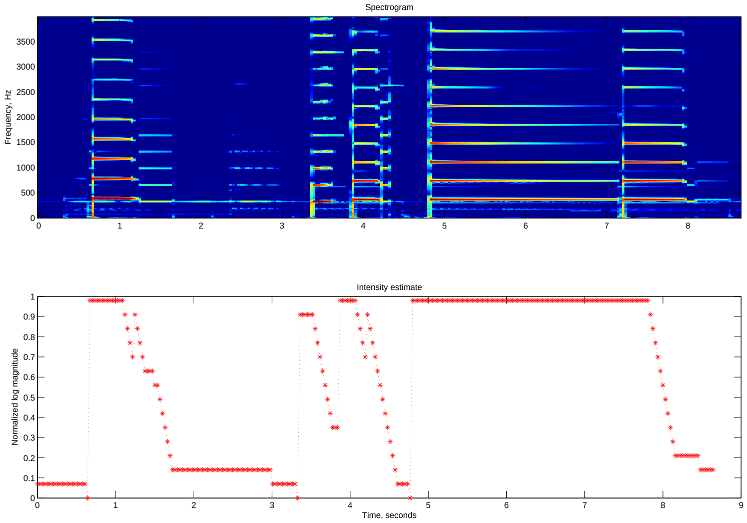

Figure 4.30: String 6 transcription results for the hexaphonic pickup example using the guitar DGM in Figure 4.1. Here we show the estimated note intensity envelope. The spectrogram of the input audio file is shown at the top.

4.6 Conclusions

In this chapter we presented a DGM for audio to piano roll guitar transcription using a hexaphonic pickup. We observed from actual recordings that the amount of crosstalk from the hexaphonic pickup was small and therefore chose not to model it. We then tried an approach that used six independent copies of the DGM from Section 3.11, each one trained on data from the corresponding string. We observed that the onset estimation performance was poor for the case of notes played with hammer-on and pull-off technique.

Based on these results, we chose to modify our note event model to allow onsets to occur from a nonzero starting intensity. This led to the guitar DGM in Figure 4.1. We observed that this model performed better than the model from Section 3.11. We presented transcription results on single-string examples of notes played using hammer-on and pull-off techniques. We presented results using all of the single instrument DGMs from Chapter 3 so that the effect of model choice on pitch tracking and transcription performance could be compared. Finally, we presented transcription results for a portion of a polyphonic piece recorded with a hexaphonic pickup.

Incorporating a note event model into the graphical model has a number of advantages over the commonly used approach of identifying note events by performing ad hoc post processing on the output of a pitch tracker. The graphical models approach allows us to make our modeling assumptions more clear and also makes it more straightforward to make changes to components of our model, as we illustrated in this chapter by modifying the note event model from Chapter 3.

Our model was able to identify notes in hexaphonic guitar recordings, provided that the note durations were sufficiently long (at least approximately 150 msec). There is still interesting future work that can be done on the guitar transcription problem. For example, there is still other information that one might be interested in extracting from the input signal. Guitarists can often recognize the type of technique that is used. For example, they can sometimes classify onsets as either plucked, or hammer-on/pull-off. Guitarists are also often capable of inferring the hand and finger positions used to play the notes. It would be interesting to extend our models to automate these tasks.

Chapter 5

Polyphonic Transcription and Musical Parameter Estimation Using a Non-negative Matrix Factorization Algorithm

5.1 Introduction

Nonnegative matrix factorization (NMF) [LS99] is a method that is capable of learning a parts-based representation of objects. Given a set of nonnegative input vectors, NMF attempts to find a set of nonnegative basis vectors and weights such that each input vector is approximated as a nonnegative linear combination of the basis vectors. When the input vectors correspond to images of faces, the basis vectors discovered by NMF sometimes consist of parts of faces (noses, mouths, eyes, etc.) [LS99]. NMF was recently used for pitch tracking of speech signals [SS05] and polyphonic piano transcription [SB03]. In both of these audio applications, NMF was applied to the magnitude spectrogram of the input audio data, resulting in basis vectors consisting of spectral templates. In [SB03], the basis vectors consisted of spectral templates for individual piano notes and chords.

Our results suggest that the ability of NMF to discover spectral template basis vectors from audio data may be useful for learning parameterized timbre models for musical instruments. We show some results of NMF applied to guitar sounds and discuss some ways of incorporating the learned parameters into a graphical model-based transcription system.

An NMF-based polyphonic pitch tracking system could be used as a preprocessing step to reduce the search space for a DGM-based transcription system. Exact inference for the DGM in chapter 3 quickly becomes intractable as more instruments and/or wider pitch ranges are modeled. So, performing preprocessing on the input signal to remove unlikely pitches and/or intensities from the search space would make inference on DGMs for multi-instrument polyphonic transcription more tractable. We present an NMF-based polyphonic transcription system and present qualitative results for piano music. Our NMF-based polyphonic transcription system is similar to the work of [SB03].

5.2 Nonnegative matrix factorization

Nonnegative matrix factorization (NMF) is an algorithm that takes a nonnegative mxn matrix X as input and attempts to find nonnegative matrix factors W of size mxr and H of size rxn such that

$$X \approx WH$$

Typically, r is chosen to be smaller than n so that the factorization will provide a compact approximate representation of X. If the columns of X represent m dimensional data vectors, then we can think of W as containing r nonnegative basis vectors for the data. If we let x_i denote the i'th column of X and h_i denote the i'th column of H, then we can represent each column of X as a nonnegative linear combination of the r nonnegative basis vectors in W. That is, $x_i \approx Wh_i, i = 1...r$

The following generalization of the Kullback-Leibler (KL) divergence is often used as the cost function for the approximation error between the two nonnegative matrices [LS99]:

$$D(X||Y) = \sum_{i,j}(X_{ij} log \frac{X_{ij}}{Y_{ij}} - X_{ij} + Y_{ij})$$

The NMF factorization is given by

$$\min_{B,H} D(X||BH)$$
$$\text{s.t. } B, H \geq 0$$

The following multiplicative update rules are guaranteed to converge to a local minimum of $D(X||H)$ [LS99]:

$$\begin{aligned} H_{jk} &\leftarrow H_{jk}\frac{\sum_i W_{ij}X_{ik}/(WH)_{ik}}{\sum_l W_{lj}} \\ W_{ij} &\leftarrow W_{ij}\frac{\sum_k H_{jk}X_{ik}/(WH)_{ik}}{\sum_l H_{jl}} \end{aligned}$$

5.3 Analysis of instrument sounds

In this section we apply NMF to the magnitude spectrogram of musical sounds. That is, we set X equal to the magnitude spectrogram. We discuss how the choice of r affects the results. Setting r to equal the number of distinct pitches that appear in the spectrogram might seem like the obvious choice of compactness of NMF representation. However, we will show that other values of r can also result in NMF representations that discover interesting musical structure.

5.3.1 Guitar

We begin by performing NMF on the spectrogram of a single note guitar recording. Figure 5.1 shows the spectrogram of a single note played on a guitar with a magnetic pickup. We tried different numbers of basis vectors for the factorization and found interesting results for $r = 1..3$. Figure 5.2 shows shows the estimated spectral template and corresponding intensity vs. time estimate for $r = 1$ basis vectors. Note that a single spectral template forces the spectral content to remain constant over the duration of the note, although the overall intensity can change. Note also how the single row of H summarizes the time behavior of the spectrogram, giving a sound intensity vs. time representation. We note also that the $r = 1$ choice gives a spectral template model of similar complexity to the one used on our multi-instrument DGMs in this thesis. However, there is also some signal structure that is not being captured by the $r = 1$ model. Note in the spectrogram that there is what appears to be a wideband spectrum at note onset. Note also that the brightness of the guitar (as well as many other instruments) decreases over time as the note sustains and decays,

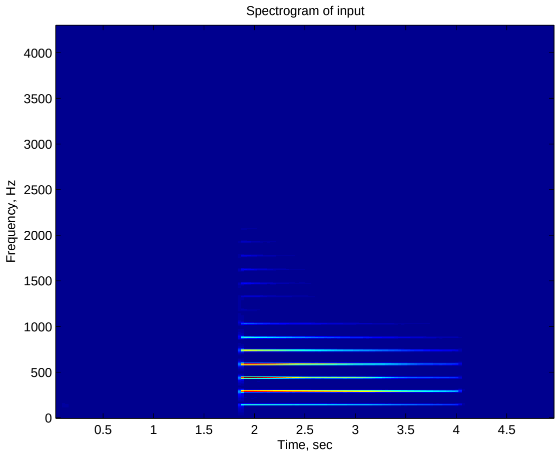

Figure 5.1: Spectrogram of a guitar note.

and cannot be captured by a single harmonic template.

We now increase the value of r by one to see if any additional interesting structure is captured. Figure 5.3 shows shows the estimated spectral template and corresponding intensity estimate for $r = 2$ basis vectors. Note that the basis vectors now consist of a wideband spectral template (column 1 of W) and a harmonic spectral template (column 2 of W). Row 1 of H contains a spike at the note onset, corresponding to the transient noise of the string pluck. So in summary, the $r = 2$ model extracts both the transient onset noise as well as a single harmonic template that is scaled in intensity over time.

We now again increase the value of r by one. Figure 5.4 shows shows the estimated spectral template and corresponding intensity estimate for $r = 3$ basis vectors. Note that like the $r = 2$ case, there is a single wideband template for representing the transient onset noise. However, there are now two distinct harmonic templates. Upon close inspection, we see that column 3 of W represents the brighter spectral content (more energy in the higher harmonics) shortly after the note onset, while column 2 of W represents the less bright spectral content after the note has sustained for a while. The $r = 3$ model is therefore able to capture some information about change in spectral content during the playing of a note. Based on empirical results using other instrument sounds, we found that the $r = 3$ choice appears to model similar interesting signal information in the other instrument

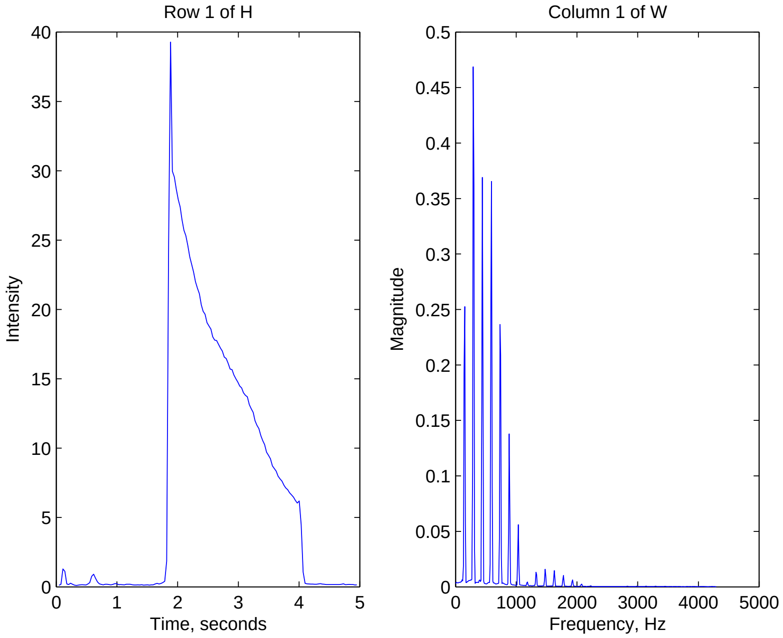

Figure 5.2: Rows of H (intensity vs. time) and columns of W (spectral templates) estimated by NMF for r=1 basis vector.

sounds characterized by onset and/or sustain noise and evolving brightness as the note sustains.

5.3.2 Piano

We now present NMF results on an audio file containing piano sounds. We chose the sequence of notes to be C → C together with F → G → D together with G. Figure 5.5 shows the spectrogram of the input signal. The two-note pairs were played simultaneously so that onset time cannot be used as a grouping cue to resolve them. Note also that while C and G are heard by themselves, F and D are never heard by themselves; they only occur as part of a two-note pair. Therefore we might expect NMF to have difficulty resolving the four notes since it lacks any knowledge of musical spectral structure. Somewhat surprisingly, we observed that for $r = 5$ basis vectors, NMF is able to resolve all four notes, along with a single wideband template for representing transient onset events. Figure 5.6 shows the NMF representation. However, the quality of factorization found by NMF depends

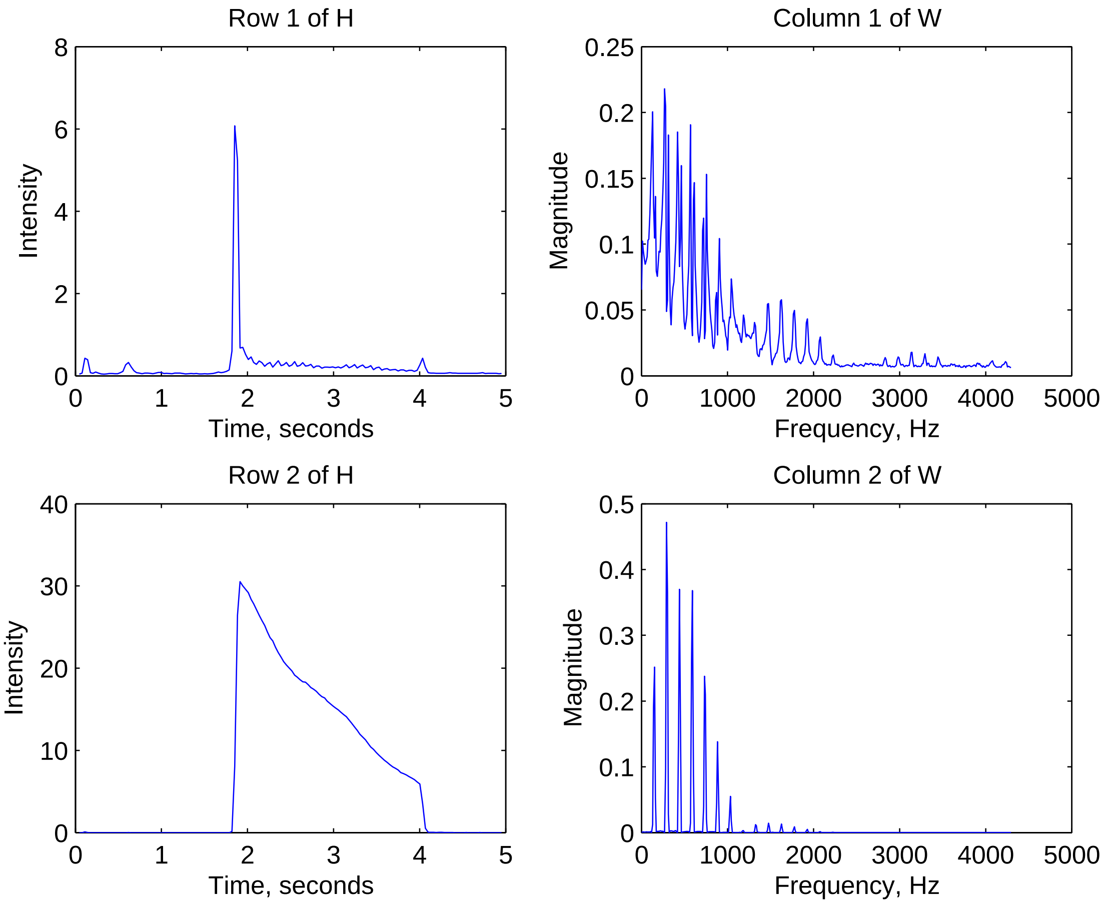

Figure 5.3: Rows of H (intensity vs. time) and columns of W (spectral templates) estimated by NMF for r=2 basis vectors.

an appropriate choice of r. [SB03] showed that some notes will typically not be resolved while others may appear more than once as basis vectors when the total number of distinct pitches in the input spectrogram is large.

5.3.3 Using learned parameters in a DGM

In chapters 3 and 4, an EM algorithm was used to learn instrument-specific spectral templates, one per pitch. These spectral templates were scaled by the discretized intensity state. The intensity state transition model constrained the way in which intensity can change during a note event (from note onset to offset). We used this model of instrument timbre because it was captured information about both the spectral content and the way in which

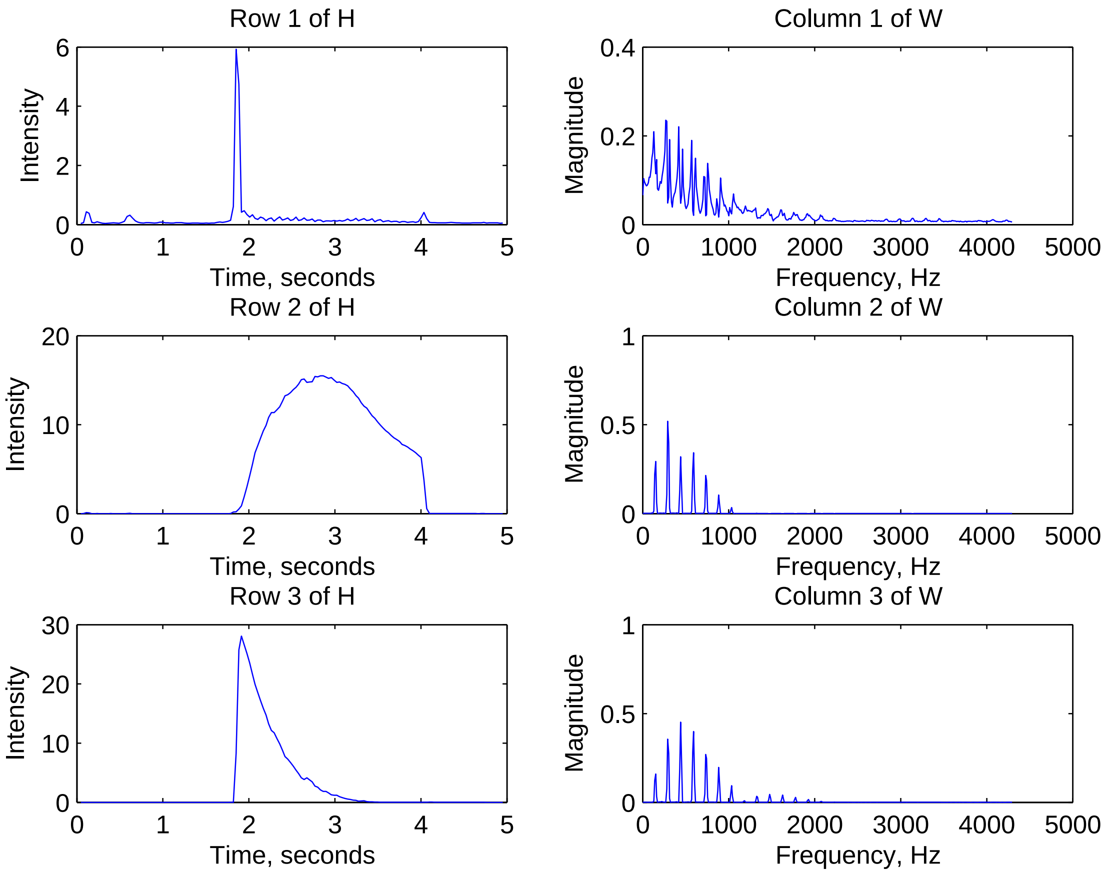

Figure 5.4: Rows of H (intensity vs. time) and columns of W (spectral templates) estimated by NMF for r=3 basis vectors.

the overall sound level changes over time. We showed that this model was capable of instrument identification given instruments with sufficiently different timbres. However, as future work, one might be interested in using more sophisticated models of instrument timbre. In particular, we think it might be interesting to model the transient noise spike at note onset. We showed that NMF framework can be used to extract parameters representing transient onset noise, spectral change during note sustain, and is not limited to harmonic musical sounds. We these reasons, we think it may be interesting to incorporate the parameters from an NMF analysis into a graphical model.

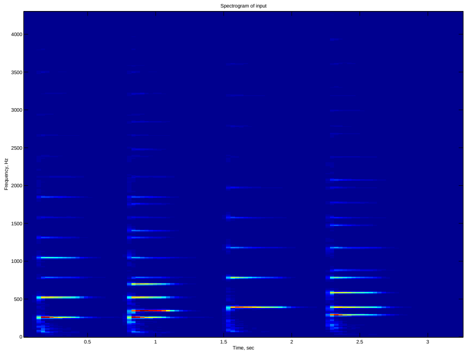

Figure 5.5: Spectrogram of the sequence of notes C, C-F, G, D-G played on a piano.

5.4 Polyphonic transcription using NMF

[SB03] presented a polyphonic piano transcription system using NMF techniques. Their system performed NMF on the magnitude spectrogram of piano sounds and identified unique sounds events in the piece, rather than a pitch vs. time transcription. Some of the sound events identified by their system consisted of unlabeled chords rather than distinct note events. This occurred because NMF does not assume a specific signal model and therefore sometimes cannot identify pitches that do not occur by themselves in the piece. Their system also required that the approximate number of distinct notes in the input piece by known a priori so the r could be set to the appropriate value.

We have implemented an obvious extension of their work, which consisted of learning the matrix W of spectral basis vectors on training data. Our training data consisted of a chromatic succession of notes covering the entire range of the piano. We then labeled the columns of W with the known pitch. Both r and W were then fixed so that only the matrix H of pitch weights vs. time would be updated during the transcription process. The output of the NMF algorithm is a matrix H consisting of a row of intensity vs. time values for each possible discrete pitch. However, the desired output of an audio to piano roll transcription system is a discrete set of onset and offset times for each pitch. Therefore, some post-processing of the rows of H is required to generate a piano roll transcription, such as a MIDI file. Our transcription system used simple thresholding of H to detect note

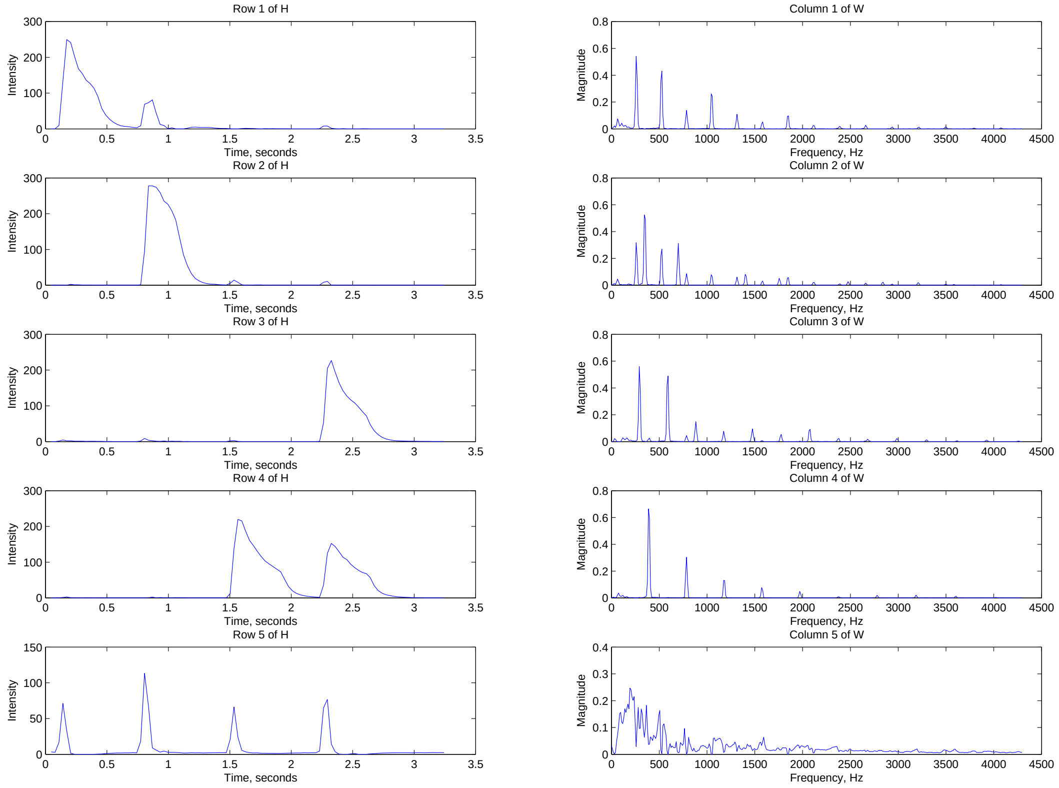

Figure 5.6: Rows of H (intensity vs. time) and columns of W (spectral templates) estimated by NMF for r=5 basis vectors.

onset events.

Since the other transcription models presented in this thesis have a maximum polyphony of two and are not well suited to polyphonic piano transcription, we chose not to make quantitative comparisons between the NMF-based system and the DGM-based system. However, our qualitative impression is that the NMF-based system's results are quite good considering the simplicity of the algorithm and implementation.

Our results suggest that an NMF analysis of the input spectrogram may be a useful preprocessing step to make inference in a DGM-based transcription system more tractable. An NMF analysis could be performed to provide a set a candidate pitches at each time step (by simple thresholding of the t'th column of H). This could significantly reduce the pitch search space required for exact inference in a DGM and might allow the modeling of more instruments and/or polyphony.

5.5 Conclusions

We have presented some results on the NMF analysis of spectrograms of musical signals. No signal model is assumed by NMF other than that the observed spectral time slices can be well represented as a nonnegative linear combination of a small number of spectral template basis vectors. Since a harmonic signal model is not assumed, the NMF approach can be used to learn spectral templates for inharmonic musical instruments such as bells, and potentially even drums and other percussive instruments. The NMF multiplicative update rules are easy to implement and rapid convergence was observed on our data sets. The update rules can be implemented in only two lines of matlab code, and we were able to implement a complete transcription system in only a few dozen lines of matlab code.

Our results suggest that an NMF analysis could be useful in obtaining instrument timbre parameters that could then be incorporated into a DGM transcription system. An NMF analysis might also be useful in making inference more tractable in a DGM transcription system by ruling out very unlikely pitch states at each time step.

Appendix A

M step for the transcription DGM

A.1 Parameter learning for the transcription DGM

We now describe how parameters are learned for the multi-instrument transcription DGM in Figure **??** in Chapter 3. Our approach is to perform training on simpler instrument-specific DGMs and then use the learned parameters in the full multi-instrument DGM. Parameter learning is performed by training an instrument-specific DGM on monophonic training data from a particular instrument. The instrument-specific DGMs are then combined to form the full multi-instrument DGM used for transcription in Figure **??**.

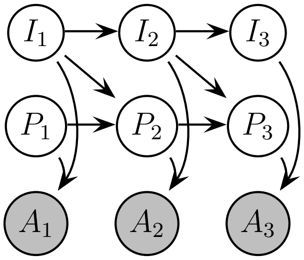

Figure A.1: The DGM used for parameter learning.

Figure A.1 shows the DGM used for parameter learning.

A.2 M-step for the pitch/intensity DGM

We now derive the M-step updates for the DGM in Figure A.1. We start by writing the complete log liklihood:

$$
\begin{aligned}
log(I,P,A) &= log\Bigg(p(I_0)p(P_0)\prod_{t=0}^{T-1}p(I_{t+1}|I_t)\prod_{t=0}^{T-1}p(P_{t+1}|I_t,P_t)\prod_{t=0}^{T}p(A_t|I_t,P_t)\Bigg)\\
&= log\Bigg(\prod_{i=1}^{M}p(I_0^i=1)^{I_0^i}\prod_{i=1}^{M}p(P_0^i=1)^{P_0^i}\prod_{t=0}^{T-1}\prod_{i,j=1}^{M}p(I_{t+1}^j=1|I_t^i=1)^{I_{t+1}^jI_t^i}\\
&\prod_{t=0}^{T-1}\prod_{i=1}^{N}\prod_{j=1}^{M}\prod_{k=1}^{N}p(P_{t+1}^i=1|I_t^j=1,P_t^k=1)^{P_{t+1}^iI_t^jP_t^k}\\
&\prod_{t=0}^{T}\prod_{i=1}^{M}\prod_{j=1}^{N}p(A_t|I_t^i=1,P_t^j=1)^{I_t^iP_t^j})\Bigg)\\
&=\sum_{i=i}^{M}I_0^ilogp(I_0^i=1)+\sum_{i=i}^{N}P_0^ilogp(P_0^i=1)\\
&+\sum_{t=0}^{T-1}\sum_{i,j=1}^{M}I_{t+1}^jI_t^ilog(p(I_{t+1}^j=1|i_{t^i=1}))\\
&+\sum_{t=0}^{T-1}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{N}P_{t+1}^iI_t^jP_t^klogp(P_{t+1}^i=1|I_t^j=1,P_t^k=1)\\
&+\sum_{t=0}^{T}\sum_{i=1}^{M}\sum_{j=1}^{N}I_t^iP_t^jlogp(A_t|I_t^i=1,P_t^j=1)
\end{aligned}
\tag{A.1}
$$

A.3 M-step for the observation model

The expected complete log likelihood for the observation model is given as

$$
\begin{aligned}
J_{obs}(\Theta_{obs}) &= \sum_{t=0}^{T}\sum_{i=1}^{M}\sum_{j=1}^{N}<I_t^iP_t^j>logp(A_t|I_t^i=1,P_t^j=1)\\
&=\sum_{t=0}^{T}\sum_{i=1}^{M}\sum_{j=1}^{N}<I_t^iP_t^j>log\mathcal{N}(A_t;\mu_{ij},\sigma_{obs}^2) \qquad \text{(A.2)}\\
&=\sum_{t=0}^{T}\sum_{i=1}^{M}\sum_{j=1}^{N}<I_t^iP_t^j>\Bigg[log(2\pi)^n\sigma_{obs}^2+\frac{1}{\sigma_{obs}^2}(A_t-\mu_{ij})^T(A_t-\mu_{ij})\Bigg]
\end{aligned}
$$

Now, maximizing for σ^2_{obs}, we obtain the M-step update for the observation noise:

$$\sigma^2_{obs} = \sum_{t=0}^{T}\sum_{i=1}^{M}\sum_{j=1}^{N} < I_t^i P_t^j > (A_t - \mu_{ij})^T (A_t - \mu_{ij})$$

We now derive the M-step update step for the harmonic magnitude parameters of the observation model. The observation component of J is

$$\begin{aligned} J(\mu_{ij}) &\propto \sum_{t=0}^{T}\sum_{i=1}^{M}\sum_{j=1}^{N} < I_t^i P_t^j > (A_t - \mu_{ij})^T (A_t - \mu_{ij}) \\ &= \sum_{t=0}^{T}\sum_{i=1}^{M}\sum_{j=1}^{N} < I_t^i P_t^j > \sum_{q=1}^{Q} (A_{tq} - \mu_{ijq})^2 \end{aligned}$$

where $\mu_{ijq} = \beta_i \sum_{h=1}^{H} \alpha_{hj} b(f_q - hF_{0j})$

The $\beta_i, i = 1 \ldots m$ are chosen uniformly spaced in log magnitude from the noise floor to the global maximum of the time frequency image. $F_{0j})$ is fundamental frequency (pitch).

The partial bump function $b(f)$, is a Gaussian kernel that models a partial (or harmonic, for harmonic musical instruments). The width of $b(f)$ is a function of the window length used for the STFT, and is chosen manually based on the window length.

The partial magnitude at the h'th harmonic at pitch j is denoted by α_{hj}. Writing J as a function of the partial magnitude parameters α, we obtain

$$\begin{aligned} J(\alpha) &= \sum_{t=0}^{T}\sum_{i=1}^{M}\sum_{j=1}^{N} < I_t^i P_t^j > \sum_{q=1}^{Q} \left[A_{tq} - \beta_i \sum_{h=1}^{H} \alpha_{hj} b(f_q - hF_{0j}) \right]^2 \\ &= \sum_{t=0}^{T}\sum_{i=1}^{M}\sum_{j=1}^{N} < I_t^i P_t^j > \parallel A_t - \beta_i B_j \boldsymbol{\alpha}_j \parallel^2 \end{aligned}$$

where

$$B_k = \begin{pmatrix} \mathbf{b}_1 & \cdots & \mathbf{b}_H \end{pmatrix}, \ \mathbf{b}_j = \begin{pmatrix} b(f_1 - jF_{0k}) \\ \vdots \\ b(f_Q - jF_{0k}) \end{pmatrix}$$

$$\boldsymbol{\alpha}_j = \begin{pmatrix} \alpha_{1j} \\ \vdots \\ \alpha_{Hj} \end{pmatrix}$$

is the vector of harmonic magnitudes corresponding to pitch state j.

$$A_t = \begin{pmatrix} A_{t1} \\ \vdots \\ A_{tq} \end{pmatrix}$$

is time slice t of the magnitude TFR. We have that

$$J(\alpha) = -\sum_{j=1}^{N} \parallel \mathbf{y}_j - C_j\boldsymbol{\alpha}_j \parallel^2$$

where

$$\mathbf{y}_j = \begin{pmatrix} \sqrt{< I_1^1 P_1^j >} A_1 \\ \sqrt{< I_1^2 P_1^j >} A_1 \\ \vdots \\ \sqrt{< I_1^M P_1^j >} A_1 \\ \sqrt{< I_2^1 P_2^j >} A_2 \\ \vdots \\ \sqrt{< I_T^M P_T^j >} A_T \end{pmatrix}$$

and

$$C_j = \begin{pmatrix} \sqrt{< I_1^1 P_1^j >} \beta_1 B_j \\ \sqrt{< I_1^2 P_1^j >} \beta_2 B_j \\ \vdots \\ \sqrt{< I_1^M P_1^j >} \beta_M B_j \\ \sqrt{< I_2^1 P_2^j >} \beta_1 B_j \\ \vdots \\ \sqrt{< I_T^M P_T^j >} \beta_M B_j \end{pmatrix}$$

For each $\boldsymbol{\alpha}_j$ we have the least squares problem $min\alpha_j \parallel \mathbf{y}_j - C_j\boldsymbol{\alpha}_j \parallel^2$. The solution is $\boldsymbol{\alpha}_j = (C_jC)^{-1}C_j^T\mathbf{y}_j$.

Appendix B

A Gaussian Process Model for Harmonic Musical Signals

B.1 Introduction

We present a probabilistic pitch model for harmonic musical instrument sounds. A probabilistic model is desirable since we are interested in using an appropriate musical signal model as a component of an automated musical transcription system based on probabilistic models. We show how the Gaussian processes regression framework can be used to estimate a spectral envelope function. We propose a method of performing pitch tracking using this model. We present experimental results on spectral envelope estimation and monophonic pitch tracking of violin sounds.

Most non percussive musical instruments produce sounds in which the partials are harmonically related to the fundamental frequency, or $F0$. The relative energy of the harmonics typically varies slowly with frequency, such that the magnitude of neighboring harmonics tends to be highly correlated. A smooth slowly varying function that approximately traces out the spectral peak locations is often referred to as the spectral envelope. In speech signal modeling, the shape of the spectral envelope contains information about the vowel structure. In musical instrument sounds, the timbre, or sound quality, is related to the spectral envelope. The variation in spectral envelope over the duration of a note event may be a useful component of systems for both pitch tracking and instrument classification.

B.2 Model

We will model a time slice of the log magnitude spectrogram as the product of a comb function $h_{F0}(f)$ and a spectral envelope function $t(f)$ plus a zero mean Gaussian noise process $\xi(f)$:

$$y(f) = t(f)h_{F0}(f) + \xi(f) \tag{B.1}$$

where $\xi(f)$ has variance σ_ξ^2 and

$$h_{F0}(f) \sum_{n=1}^{H} b(f - nF0) \tag{B.2}$$

We model a single harmonic bump as

$$b(f) = exp(-f^2/\sigma_b) \tag{B.3}$$

The spectral envelope function $t(f)$ should be smooth and slowly varying. By slowly varying, we mean that $t(f)$ should not vary widely between successive harmonics. We propose modeling $t(f)$ as a Gaussian process. We chose to use a Gaussian kernel for the covariance function of $t(f)$:

$$k(f, f') = exp(-\frac{(f - f')^2}{2\sigma_t^2}) \tag{B.4}$$

The covariance kernel width parameter σ_t^2 was chosen manually ($\sigma_t = 500Hz$). The harmonic bump width parameter σ_b^2 was also chosen manually (σ_b = 25 Hz). Note that it would also be possible to compute $b(f)$ from the window function used for the spectrogram. For simplicity, however, we chose to represent $b(f)$ as a Gaussian kernel.

A time slice of the spectrogram gives us samples of $y(f)$ at the uniformly spaced frequencies $f_i, i = 1...N$, where N is the number of spectrogram frequency bins. We use Gaussian processes regression [Smo02] to compute the spectral envelope as the MAP estimate. Let $\mathbf{y} = \{y(f_1), ..., y(f_N)\}$ denote the observed spectrogram time slice data.

The posterior is

$$\begin{aligned} p(\mathbf{t}|\mathbf{y}, F0) &\propto p(\mathbf{y}|\mathbf{t})p(\mathbf{t}|F0) && \text{(B.5)} \\ &= \left[\prod_{i=1}^{N} p(y(f_i)|t(f_i))\right] p(\mathbf{t}|F0) && \text{(B.6)} \end{aligned}$$

Taking logarithms, and substituting $\mathbf{t} = K\boldsymbol{\alpha}$, as on page 485 of [Smo02], we get

$$log(p(\boldsymbol{\alpha}|\mathbf{y}, F0)) = -\frac{1}{2\sigma_\xi}\sum_{i=1}^{N}(y(f_i) - h_{F0}(f_i)K(:,i)^T\boldsymbol{\alpha})^2 - \frac{1}{2}\boldsymbol{\alpha}^T K\boldsymbol{\alpha} \tag{B.7}$$

Here, $K(:,i))$ refers to the k'th column of K. Taking the gradient and solving for $\boldsymbol{\alpha}$, we get

$$\boldsymbol{\alpha} = (\sigma_\xi K + \sum_{i=1}^{N} h_{F0}(f_i)^2 K(:,i)K(:,i)^T)^{-1}\sum_{i=1}^{N} y(f_i)h_{F0}(f_i)K(:,i) \tag{B.8}$$

The spectral envelope estimate is then given as $\tilde{\mathbf{t}} = K\boldsymbol{\alpha}$.

B.2.1 Pitch tracking

Note that our model assumes that the pitch, $F0$, is given. If the pitch is unknown, we must perform a search over all possible $F0$. In some cases, it may be reasonable to constrain the set of admissible pitch values to those on some particular tuning. If we have a uniform prior over the pitch space, then the best pitch estimate can be taken to be the $F0$ that maximizes the likelihood in (7).

B.3 Experiments

Figure B.1 shows a spectrogram time slice from a violin recording. Since $t(f)$ is a zero mean process, some preprocessing should be performed on the raw spectrogram data so that the spectrogram noise floor will correspond to the mean of $t(f)$. This preprocessing consists of suppressing everything below some threshold (currently 60 dB below the global maximum) and then shifting the data so the minimum value is zero.

Figure B.2 shows the Gaussian process envelope fit to the violin spectrum. We consider the performance to be good provided that the input signal is well modeled as having harmonic partials. The violin is well modeled as having harmonic partials, so this is not a problem here. However, the piano is an instrument for which the inharmonicity can be significant. Fortunately, the partial locations as a function of $F0$ are still straightforward to compute [FR91].

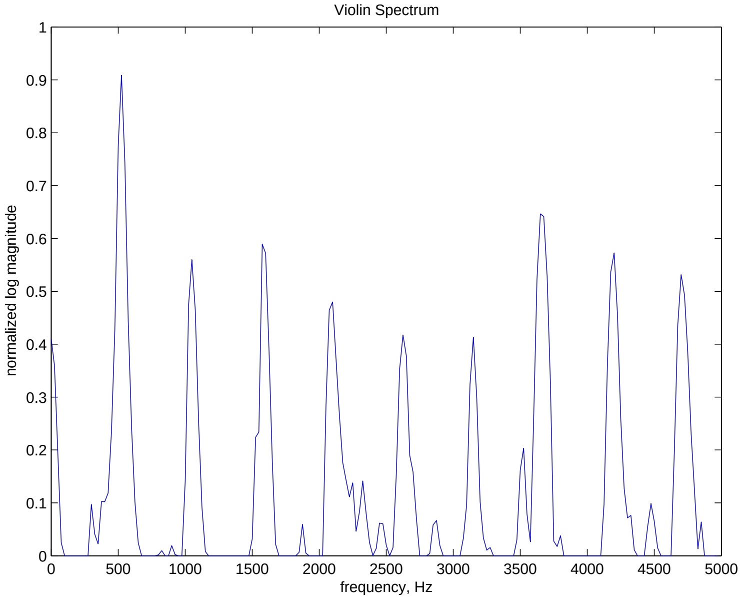

Figure B.1: A spectrogram time slice of a violin signal.

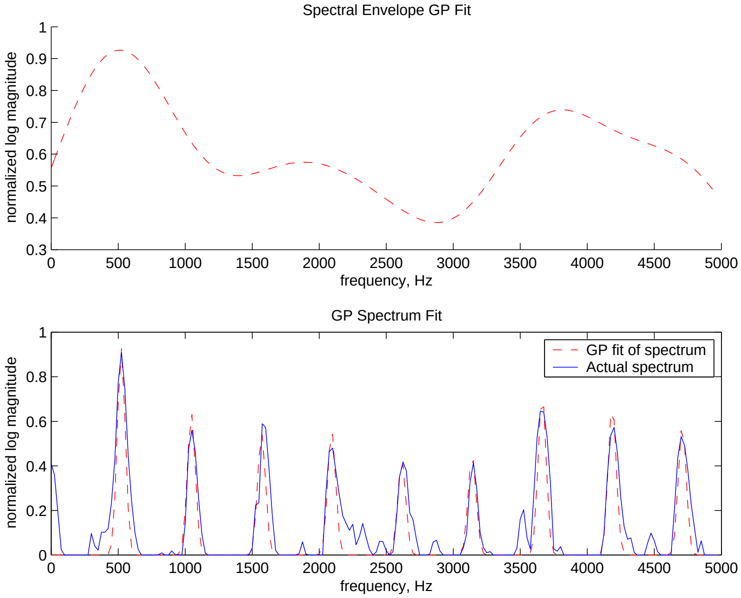

Figure B.2: Gaussian process envelope fit to a violin spectrum.

B.4 Conclusions

We have proposed a Gaussian process model for harmonic musical signals. We showed how the Gaussian processes regression framework can be used to estimate a spectral envelope function and we proposed a method of performing pitch tracking using this model. We presented experimental results on a violin signal.

There are some computational issues with Gaussian processes regression. In particular, it requires inverting an N x N matrix, where N is the number of data points. This is an $O(N^3)$ operation, limiting the number of data points to a few hundred. Approximation techniques are available, but we have not experimented with them yet.

It might be interesting to consider setting the width of the covariance function, σ_t, to be proportional to $F0$. This might be a more reasonable way to enforce that $t(f)$ should not vary widely between successive harmonics.

We should point out that there are other spectral envelope estimation techniques in the literature. Cepstral analysis is an example. However, our model is appealing in that it is probabilistic and could be used as a component in a more complex probabilistic model.

It may be interesting to consider extensions of this model for polyphonic pitch tracking. We are also currently experimenting with a similar probabilistic model of harmonic musical signals and applying it to the problem of automated musical transcription (polyphonic pitch tracking and instrument classification).

Bibliography

[Bil93] J. Bilmes. Timing is of the essence: Perceptual and computation techniques for representing, learning, and reproducing, expressive timing in percussive rythm. Master's thesis, Massachusetts Institute of Technology, 1993.

[Bre90] A. Bregman. *Auditory scene analysis*. MIT Press, Cambridge, MA, 1990.

[BXM04] J. Bilmes, L. Xiao, and J. Malkin. A graphical model approach to pitch tracking. In *International Conference on Spoken Language Processing (ICSLP)*, 2004.

[BXM05] J. Bilmes, L. Xiao, and J. Malkin. A graphical model for mormant tracking. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 2005.

[CK03] A.T. Cemgil and H.J. Kappen. Monte carlo methods for tempo tracking and rhythm quantization. *Journal of Artificial Intelligence Research*, 18:45–81, 2003.

[CKB04] A.T. Cemgil, H.J. Kappen, and D. Barber. A generative model for music transcription. In *IEEE Transactions on Speech and Audio Processing*, 2004.

[dCK02] A. de Cheveigne and H. Kawahara. Yin, a fundamental frequency estimator for speech and music. *J. Acoust. Soc. Am.*, 111:1917–1930, 2002.

[DG02] M. Davy and S.J. Godsill. *Seventh Valencia International meeting (Bayesian Statistics 7)*, chapter Bayesian harmonic models for musical signal analysis. Oxford University Press, 2002.

[EB04] J. Eggink and G. Brown. Instrument recognition in accompanied sonatas and concertos. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 2004.

[FR91] N. H. Fletcher and T. D. Rossing. *Physics of musical instruments*. Springer Verlag, New York, 1991.

[GJ94] Z. Ghahramani and M.I. Jordan. Factorial hidden markov models. *Machine Learning*, 1994.

[Goo97] M. Goodwin. *Adaptive signal models: Theory, algorithms, and audio applications*. PhD thesis, EE Division, U.C. Berkeley, 1997.

[Got04] M. Goto. A real-time music-scene-description system: Predominant-f0 estimation for detecting melody and bass lines in real-world audio signals. *Speech Communication (ISCA Journal)*, 43:311–329, 2004.

[Hai01] S. Hainsworth. Analysis of musical audio for polyphonic transcription 1st year phd report. Technical report, Univeristy of Cambrige, 2001.

[Hes91] W. Hess. *Advances in speech signal processing*, chapter Pitch and Voicing Determination, pages 3–48. Marcel Dekker, Inc., New York, 1991.

[JL83] R. Jackendoff and F. Lerdahl. *A generative theory of tonal music*. MIT Press, Cambridge, Massachusetts, 1983.

[KEA06] A. Klapuri, A. Eronen, and J. Astola. Analysis of the meter of acoustic musical signals. In *IEEE Trans. Speech and Audio Processing, in press since Dec. 2004 (to appear Jan. 2006)*, 2006.

[Kla99] A. Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 3089–3092, 1999.

[Kla04] A. Klapuri. *Signal processing methods for the automatic transcription of music*. PhD thesis, Tampere University of Technology, 2004.

[LS99] D.D. Lee and H.S. Seung. Learning the parts of object by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[Mar98a] K. Martin. Musical instrument identification: a pattern-recognition approach. In *136th Meeting of the Acoustical Society of America, Norfolk, VA*, 1998.

[Mar98b] K. Martin. Towards automatic sound source recognition - identifying musical instruments. In *Nato Computational Hearing Advanced Study Institute, Il Ciocco, Italy*, 1998.

[Moo75] J. Moorer. *On the segmentation and analysis of continuous musical sound by digital computer*. PhD thesis, Department of Computer Science, Stanford University, Stanford, 1975.

[Mur02] K. Murphy. *Dynamic Bayesian networks: Representation, inference and learning*. PhD thesis, CS Division, U.C. Berkeley, 2002.

[OS89] A. Oppenheim and R. Schafer. *Discrete-time signal processing*. Prentice-Hall, Inc., 1989.

[Rap02] C. Raphael. Automatic transcription of piano music. In *Proc. of ISMIR*, 2002.

[RK04] M. Ryynanen and A. Klapuri. Modelling of note events for singing transcription. In *Proc. ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing*, 2004.

[Rot92] J. Rothstein. *MIDI : a comprehensive introduction*. Madison, Wis. : A-R Editions, 1992.

[SB03] P. Smaragdis and J.C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *Proc. of IEEEWorkshop on Applications of Signal Processing to Audio and Acoustics*, pages 177–180, 2003.

[She82] R. Shepard. *The Psychology of Music*, chapter Structural representations of musical pitch. Academic Press, New York, 1982.

[SLIL03] L. K. Saul, D. D. Lee, C. L. Isbell, and Y. LeCun. Real time voice processing with audiovisual feedback: toward autonomous agents with perfect pitch. *Advances in Neural Information Processing Systems*, 15:1205–1212, 2003.

[Smo02] B. Scholkopf A.J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT Press, Cambridge, MA, 2002.

[SS05] F. Sha and L. K. Saul. Real-time pitch determination of one or more voices by nonnegative matrix factorization. *Advances in Neural Information Processing Systems*, 17, 2005.

[Ste99] A. Sterian. *Model-based segmentation of time-frequency images for musical transcription*. PhD thesis, University of Michigan, Ann Arbor, Dept. of Electrical Engineering and Computer Science, 1999.

[Tal95] D. Talkin. *A robust algorithm for pitch tracking (RAPT)*, chapter Speech Coding and Synthesis, pages 495–518. Elsevier Science, 1995.

[Ter74] E. Terhardt. Pitch, consonance and harmony. *Journal of the Acoustical Society of America*, 55:1061–1069, 1974.

[VJW05] B. Vogel, M.I. Jordan, and D. Wessel. Multi-instrument musical transcription using a dynamic graphical model. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 2005.