Classification-Based Music Transcription

Graham E. Poliner

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY
2008

(This page intentionally left blank)

(This page intentionally left blank)

Abstract

Classification-Based Music Transcription

Graham E. Poliner

Music transcription is the process of resolving the musical score from an audio recording. The ability to generate an accurate transcript of a musical performance has numerous practical applications ranging in nature from content-based organization to musicological analysis. Although trained musicians can generally perform transcription within a constrained setting, the process has proven to be quite challenging to automate since the recognition of multiple simultaneous notes is generally obfuscated by the harmonic series interaction that renders music aurally pleasing.

In contrast to model-based approaches that incorporate prior assumptions of harmonic or periodic structure in the acoustic waveform, we present a classification-based framework for automatic music transcription. The proposed system of support vector machine note classifiers temporally constrained via hidden Markov models may be cast as a general transcription framework, trained specifically for a particular instrument, or used to recognize higher-level musical concepts such as melodic sequences. Although the classification structure provides a simple and competitive alternative to model-based systems, perhaps the most important result of this thesis is that no formal acoustical prior knowledge is required in order to perform music transcription.

We report a series of experiments, with corresponding comparisons to alternative approaches, in which the proposed framework is used to transcribe real-world polyphonic music ranging in diversity from ensemble orchestral recordings to popular music tracks. In addition, we describe several methods for extending a limited set of labeled training data, thereby improving the generalization capabilities of the classification system. Finally we relate a demonstrative experiment in which the classification posteriors (i.e. the outputs of the proposed framework) are used as an acoustic feature representation.

(This page intentionally left blank)

Contents

List of Figures

(This page intentionally left blank)

List of Tables

(This page intentionally left blank)

Acknowledgments

Firstly, I would like to thank my advisor, Professor Dan Ellis, from whom I have learned so much more than signal processing techniques. I am profoundly grateful that he was willing to take an unwarranted chance on me, and I will continue to strive toward the standard he set. I cannot imagine a more erudite advisor or a mentor better suited to my disposition.

I would also like to thank Professors Juan Bello, Shih-Fu Chang, Brad Garton, and John Kymissis for serving on my Ph.D. committee and for their influential instruction, comments, and discussions over the past few years.

While studying in LabROSA, I have been extremely fortunate to be a part of a warm community of brilliant researchers. I am very grateful to all the members of the group for their insights and friendship, and I am proud to be among their peers.

Finally, I would like to thank my family: my sister Caitlin, for whom boundaries do not exist and from whom I derive so much inspiration; Laurie Ortiz, whose love and infinite optimism have buoyed my spirits an uncountable number of times; and my parents, who have ensured that my wildest dreams have all become reality and to whom I would like to dedicate this thesis. Please accept this work as a token of my appreciation for all the love and support you have provided me.

(This page intentionally left blank)

Chapter 1

Introduction

Music elicits a plethora of responses from listeners, and as such, has received research consideration in fields ranging from philosophy to signal processing. Recently, the pervasiveness of music data led to the establishment of an entirely new research arena, music information retrieval, specifically concerned with developing methods for the organization and analysis of the rapidly growing musical universe. This thesis is concerned with one such method, automatic music transcription, and its application to content-based audio retrieval.

Music transcription is the process of resolving the musical score (i.e. a symbolic representation) from an audio recording. Thus, transcription entails recovering the list of note times and pitches generated by the performer or ensemble. In this thesis, transcription is specifically defined as estimating the fundamental frequency for the set of notes present within a frame of audio.

The ability to generate an accurate transcript of a performance has numerous practical applications in content-based organization and musicological analysis. For example, estimated transcripts may be used to identify multiple performances of the same piece of music within an audio database. Alternatively, an analysis of deviations from a reference score may be used as an instructive device or to examine stylistic variations between a set of performances. In addition, an automated transcription system could be used as the front-end to a source transformation system (e.g. synthesizing an audio recording with different instrumentation).

Trained musicians can typically transcribe polyphonic recordings within a constrained setting (though the undertaking is often arduous); however, the process has proven to be quite challenging to automate since the recognition of multiple simultaneous notes is generally obfuscated by the harmonic series interaction that renders music aurally pleasing. While a single musical note may be represented by a set of harmonics at integer multiples of the fundamental frequency under Fourier analysis as displayed in the left pane

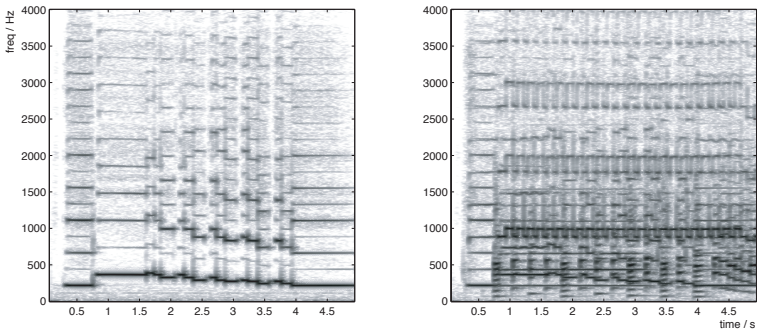

Figure 1.1: *Left*: Short-time Fourier transform spectral representation of a monophonic clarinet recording. *Right*: Spectral representation of a polyphonic quintet recording.

of Figure 1.1, ensemble music may consist of multiple notes (with fundamental frequencies at simple ratios) that overlap in time. The coincidence of the harmonics results in complex patterns of constructive and destructive interference in a narrowband spectral analysis as displayed in the right pane of Figure 1.1. That is, the underlying phenomena in musical harmony significantly complicate the corresponding analysis.

This thesis also considers the subject of melody transcription, a special case of music transcription in which the fundamental frequency of the most salient pitch is estimated. The melody of a piece of music is the principal part of a composition – informally, the sequence of tones that a listener might whistle or hum. As such, melody provides a concise and natural description of music that serves as an intuitive basis for communication and retrieval (e.g. query-by-humming). Although the fundamental mechanism required to deploy organizational systems based on melodic content faces similar challenges to general transcription systems, melody transcription systems face the additional challenge of discriminating the predominant note from within the polyphony.

1.1 Contributions

In this dissertation, we propose a machine learning approach to automatic music transcription. The proposed framework consists of a system of support vector machine classifiers temporally constrained via hidden Markov models. The classification-based system may be generalized to perform polyphonic pitch estimation or trained specifically to recognize the predominant melody. This learning-based approach to pitch transcription stands in stark contrast to previous approaches that incorporate prior assumptions of harmonic or periodic structure in the acoustic waveform. While the assumption that pitch arises from harmonic components is strongly grounded in musical acoustics,

it is not strictly necessary for transcription. As such, the main contribution of this thesis is a demonstration of the feasibility and simplicity of a purely data driven approach to music transcription.

In addition to the presentation of the classification-based framework and functional factors that influence the performance of the approach, we propose the use of classification posteriors as features for related music information retrieval tasks. An illustrative experiment is reported in which the classification posteriors (i.e. estimated transcripts) are used as acoustic features to synchronize musical scores to audio recordings. The resulting audio-transcript pairs may be used to bootstrap the original classification system.

In order to demonstrate the plausibility of the proposed framework, we created a corpora of labeled data for training and testing transcription systems. The labeled testing data and evaluation metrics described in this thesis were used to organize an international evaluation of melody transcription systems and constituted a portion of the test data used for an similar evaluation of polyphonic pitch estimation algorithms.

The work directly related to this thesis was reported in three journal articles [26, 54, 56] and two conference proceedings [53, 55].

1.2 Overview and Organization

The remainder of the thesis is structured as follows:

In Chapter 2, we provide a background discussion on polyphonic pitch estimation, melody transcription, and score to audio alignment, as well as a summary of previous work.

In Chapter 3, we introduce the concept of classification-based music transcription in the context of melody transcription. A description of the general framework consisting of a system of support vector machines and hidden Markov models is presented along with a corresponding analysis of the data collection, feature selection, and classification experiments conducted.

In Chapter 4, we extend the single-estimate classification framework described in Chapter 3 in order to perform polyphonic pitch transcription. The proposed framework is presented first as a system for polyphonic piano transcription then generalized for instrument-independent pitch estimation.

In Chapter 5, we examine several methods based on semi-supervised learning and multiconditioning for enhancing a limited training set thereby increasing the generalization capabilities of the proposed framework.

In Chapter 6, we explore the use of classification posteriors as acoustic features for score to audio alignment and present a keystone experiment in which the synchronized score/audio pairs are used to bootstrap the supervised classification system.

Finally in Chapter 7, we make concluding remarks regarding the merits and limitations of the classification-based framework and propose directions for future work.

Chapter 2

Background

In this chapter we provide background information and a discussion of prior research in the areas of polyphonic pitch estimation, melody transcription, and score to audio alignment. Although providing an exhaustive catalog of previous work is impractical, we have, to the best of our knowledge, surveyed a number of representative approaches for each of the research problems considered.

2.1 Music Transcription

Music transcription is the process of resolving a musical score from an audio recording. As such, transcription involves recovering the list of note times and pitches generated by a performer or ensemble. In order to automate the transcription process, a system must estimate the set of fundamental frequencies that correspond to the notes played within a given period of time (i.e. detecting the pitch, onset, and duration of each note).

Automated music transcription has a rich signal processing history dating back into the 1970s. In [47], Moorer proposed a limited system for duet transcription. Since then, a long thread of research has gradually improved transcription accuracy and reduced the scope of constraints (e.g. limitations on the number of concurrent notes or confinement to a specific instrument) required for successful transcription[1]; however, we are still far from a system that can automatically convert a recording into an accurate transcript in an unconstrained setting. Nonetheless, automatic music transcription has garnered a significant amount of research attention since such a system would have numerous practical implications in musicological analysis and content-based retrieval.

[1] A recent summary of the field is available in [40].

System	Front end	Multi-pitch	Note events	Post-processing
Ryynänen [62]	$\|STFT\|$	Harmonic sieve	HMM	
Smaragdis [68]	$\|STFT\|$	NMF	–	–
Marolt [43]	Harmonic oscillators	Neural Nets	Onset detection	*ad hoc* algorithms
Kameoka [37]	Power spectrum clustering via EM		–	–
Martin [45]	Auditory correlogram	Blackboard hypotheses		heuristics
Davy [46]	AR/harmonic model	Bayesian network		–
Cemgil [8]	Stochastic processes	Bayesian network		–
Bello [2]	Time domain	Mixing matrix, phase-alignment to a database	–	–

Table 2.1: Representative polyphonic transcription algorithms. For brevity, systems are referred to by their first author alone.

The algorithm structure and characteristic design parameters for a representative set of (western tonal music) polyphonic pitch transcription systems is displayed in Table 2.1. For example, all transcription systems must select a domain in which to examine the audio signal (e.g. spectrum or time domain), adopt an approach for handling temporal overlap of simultaneous notes with different periods, and may include further processing to organize frame-level pitch estimates into structured note events. The first column of the table, "Front end", describes the various signal processing approaches applied to the input audio in oder to reveal the pitch content. The most common technique is to apply the magnitude of the short-time Fourier transform (denoted $|STFT|$ in the table). In the $|STFT|$ representation, pitched notes appear as a 'ladder' of more-or-less stable harmonics in the spectrogram as displayed in Figure 1.1. Unlike the time waveform itself, $|STFT|$ is invariant to relative or absolute time or phase shifts in the harmonics because the STFT phase is discarded. This result is convenient since perceived pitch has essentially no dependence on the relative phase of (resolved) harmonics, and it makes the transcription invariant to the alignment of the analysis time frames. Since the frequency resolution of the STFT improves with temporal window length, these systems tend to employ long windows (e.g. 50 to 100 ms or more).

As an alternative to the STFT, Martin applies the log-lag correlogram [25], which is based on the short-time autocorrelation correlogram described in [67]. Like the $|STFT|$, the autocorrelation (which may be calculated by taking the inverse Fourier transform of the $|STFT|$) is phase invariant. Rather than explicitly calculating a Fourier transform, Davy proposed an autoregressive-based polyphonic harmonic model in order to represent the acoustical energy. Although the system proposed by Cemgil does not strictly calculate a Fourier transform or implement a sinusoidal model, the signal is modeled by a stochastic process that typically results in periodic oscillations.

In stark contrast to the systems discussed above, Bello does not attempt to model the frequency domain characteristics of the signal at all. Instead, segments under consideration are phase-aligned in the time domain and transcription is performed via a database comparison to previously seen notes. As such the proposed time-domain implementation is necessarily restricted to cases in which the phase relationship between partials in a given note may be assumed to be reproducible (e.g. piano notes) and essentially limited to the monophonic case for practical purposes due to the computational expense of calculating and storing representative phase combinations.

Kameoka proposed harmonic temporal structured clustering (HTC), a method with similarities to earlier work by Goto [33], which attempts to perform the front-end feature extraction and multi-pitch estimation cooperatively. The HTC model decomposes the energy patterns of the power spectrum (as calculated using a Gabor-based wavelet transform) into clusters such that each group corresponds to a single source. The sources are then modeled by a mixture of two dimensional Gaussians that are constrained harmonically in frequency and continuously in time. Transcription is performed by fitting mixtures of the source models to the observed power spectrum by updating model parameters and clustering the energy patterns via expectation maximization (EM).

The "Multi-pitch" column of Table 2.1 addresses how the representative systems deal with estimating the multiple periodicities present in the polyphonic audio. Systems that apply the $|STFT|$ typically perform transcription by identifying the set of fundamental frequencies corresponding to the observed harmonic series. This operation is generally performed by implementing a 'harmonic sieve' [31, 23], which, in principle, considers each possible fundamental by integrating evidence from every predicted harmonic location. One weakness of this approach is its susceptibility to reporting a spectrum one octave too high, since if all the harmonics of a fundamental frequency f_0 are present, then the harmonics of a putative fundamental $2f_0$ will also be present. The multi-pitch identification stage of Ryynänen's implementation [39] is essentially an iterative harmonic sieve; however, lower fundamentals are identified first and the spectrum is modified at each iteration in order to remove the energy associated with the identified pitch, thereby removing evidence for octave transpositions.

Martin performed multi-pitch detection and note event modeling simultaneously by implementing a blackboard system [25]. The proposed framework incorporated knowledge ranging from the low-level correlogram features described above to hypotheses of note structure and musical rules in order to perform transcription.

The remaining representative systems perform multi-pitch estimation using conventional machine learning techniques. In addition to many others, Smaragdis performs polyphonic pitch estimation via non-negative matrix factorization (NMF) [41], an unsupervised learning technique popular in audio scene analysis that learns harmonic structure from the magnitude spectra. In the sys-

tem proposed by Marolt, transcription is achieved by using neural networks to classify the outputs of adaptive harmonic oscillators. Likewise, Davy employs a Bayesian framework based on Markov Chain Monte Carlo sampling of harmonic oscillator posterior distributions. Finally, the graphical model proposed by Cemgil emulates sound generation by incorporating prior information on music structure with low-level acoustical analysis in a switching Kalman filter framework.

The "Note events" and "Post-processing" columns of Table 2.1 relate how, if at all, the representative multiple fundamental frequency transcription systems convert pitch estimates to the note-level of abstraction. Whereas Martin, Davy, and Cemgil consider notes (or at least onsets) in tandem with pitch estimation[2], a number of transcription systems integrate musicological considerations in a separate stage. Systems such as those proposed by Marolt and Martin employ heuristics in order to incorporate a representation of musical knowledge or common errors (e.g. removing octave transpositions). In contrast, Ryynänen resolves note events with a hidden Markov model (HMM) that incorporates musicological considerations (e.g. key estimates and bigram models) and imposes temporal consistency on the multi-pitch estimations.

2.2 Melody Transcription

Melody transcription is a special case of music transcription that entails estimating the fundamental frequency of the 'predominant pitch' within a polyphony, loosely defined as the dominant perceived melody note. In the context of identifying the main melody within multi-instrument music, the music transcription problem is further complicated because although multiple pitches may be present at the same time, at most just one of them will be the melody. Thus, all approaches to melody transcription face two problems: identifying a set of candidate pitches that appear to be present at a given time, then discriminating which (if any) of those pitches correspond to the melody.

For the scope of this thesis, we define melody as the single (monophonic) pitch sequence that a listener might reproduce if asked to whistle or hum a piece of polyphonic music (i.e. the sequence a listener would recognize as being the 'essence' of a piece of music). In particular, much of popular music contains a 'lead vocal' line, a sung contour which is typically the most prominent source in the mixture, that listeners have no trouble distinguishing from the background accompaniment. However, classical orchestral music and richly polyphonic piano compositions commonly possess a single, prominent melody line that can be agreed upon by most listeners. Thus, while we are in the dangerous position of setting out to quantify the performance of automatic systems seeking to transcribe something that is not strictly defined, there is some hope we can conduct a meaningful evaluation.

[2] The temporal clustering proposed by Kameoka may also be akin to note-level segmentation.

System	Front end	Multi-pitch	No. pitch	Onset events	Post-processing	Voicing
Dressler [21]	$\|STFT\|$+sines	Harmonic model fit	5	Fragments	Streaming rules	Melody+ local thresh.
Marolt [44]	$\|STFT\|$+sines	EM fit of tone models	> 2	Fragments	Proximity rules	Melody grouping
Goto [33]	Hierarchic $\|STFT\|$+sines	EM fit of tone models	> 2	–	Tracking agents	continuous
Ryynänen [63]	$\|STFT\|$	Harmonic sieve	2	Note onsets	HMM	Background model
Paiva [50]	Auditory correlogram	Summary autocorrelation	> 2	Pitches	Pruning rules	Melody grouping
Vincent [76]	YIN / Time windows	Gen. model inference	5 / 1	–	HMM	continuous

Table 2.2: Representative melody transcription algorithms. For brevity, systems are referred to by their first author alone.

In [35, 33], Goto proposed identifying a single, dominant periodicity over the main musical spectral range (plus a single low-frequency bass line estimate) which he referred to as "Predominant-Fo Estimation" or PreFEst. In Goto's system, the predominant fundamental is generally recognizable as the melody of the polyphonic music, and as such, the system provides a representative "sketch" of popular music. Such a representation may be used to implement a number of practical systems such as query-by-humming [30] or as a tool to analyze musicological primitives, and as a result, a great deal of research has recently taken place with respect to automatic melody transcription as summarized by the representative systems in Table 2.2.

The "Front end" column of Table 2.2 describes the various signal processing approaches applied to input audio in oder to reveal the pitch content. As was the case for general music transcription, the most common technique is to apply the magnitude of the short-time Fourier transform. In a slightly more complex implementation, Goto uses a hierarchy of STFTs in order to improve frequency resolution, down-sampling the original 16 kHz audio through 4 factor-of-2 stages resulting in a 512 ms window at the lowest (i.e. 1 kHz) sampling rate. Since musical semitones are logarithmically spaced with a ratio between adjacent fundamental frequencies of $2^{1/12} \approx 1.06$, to preserve semitone resolution down to the lower extent of the pitch range (i.e. below 100 Hz) requires these longer windows. Dressler, Marolt, and Goto further reduce their magnitude spectra by recording only the sinusoidal frequencies estimated as relating to prominent peaks in the spectrum, using a variety of techniques (such as instantaneous frequency [29]) to exceed the resolution of the STFT bins.

A number of systems apply autocorrelation as an alternative to the STFT. In the representative systems listed, Paiva uses the Lyon-Slaney auditory model up to the summary autocorrelation [67], and Vincent uses a modified version

of the YIN pitch tracker [18] to generate candidates for time-domain model inference. The Lyon-Slaney model calculates autocorrelation on an approximation of the auditory nerve excitation, which separates the original signal into multiple frequency bands, then sums the normalized results. In order to perform multi-pitch detection, Paiva simply identifies the largest peaks in the summary autocorrelation. Although YIN incorporates autocorrelation across the full frequency band, Vincent performs the calculation based on the STFT representation, and reports gains from some degree of across-spectrum energy normalization. Interestingly, because the resolution of autocorrelation is a function of the sampling rate rather than the window length, Paiva uses a significantly shorter window of 20 ms, and considers periods only out to 9 ms lag (110 Hz).

The "Multi-pitch" column of Table 2.2 addresses how the representative systems deal with distinguishing the multiple periodicities present in the polyphonic audio, and the following column, "No. pitch", quantifies the number of simultaneous pitches reported at any time. Systems that apply the $|STFT|$ transcribe the melody note by identifying the fundamental frequency of the harmonic series (even though there need not be any energy at that fundamental for humans to perceive the pitch), generally performed by implementing a harmonic sieve. Ryynänen's melody transcription implementation employs the same iterative harmonic sieve multi-pitch stage as the polyphonic system described above.

Goto proposed an expectation maximization technique for estimating weights over all the possible fundamentals in order to jointly explain the observed spectrum. As such, the different fundamentals effectively compete for harmonics, a process that is largely successful in resolving octave ambiguities. Marolt modified the EM procedure slightly to incorporate perceptual principles and to consider, exclusively, fundamentals that are equal to (or one octave below) observed frequencies. As a result, EM assigns weights to every possible pitch (most of which are very small), and the largest weighted frequencies are taken as the potential pitches at each frame (with two to five pitches typically considered).

Although Vincent uses autocorrelation in order to estimate up to five candidate pitches, the core of his system is a generative model for the time-domain waveform within each window that includes parameters for fundamental frequency, overall gain, amplitude envelope of the harmonics, the phase of each harmonic, and a background noise term that scales according to local energy in a psychoacoustically-derived manner. The optimal parameters are inferred for each candidate fundamental, and the one with the largest posterior probability under the model is chosen as the melody pitch at that frame.

The "Onset events" column of Table 2.2 reflects that only some of the representative systems attempt to incorporate note (or note-series) level analysis. The systems proposed by Goto and Vincent simply estimate a single melody pitch at every frame and do not attempt to form them into higher-level note-type structures. Dressler and Marolt, however, track the amplitude variation

in the harmonic sets (since there may still be multiple candidate notes) in order to form distinct fragments of more-or-less continuous pitch and energy. Paiva attempts to resolve the continuous pitch tracks into piecewise-constant frequency contours, thereby removing effects such as vibrato and slides between notes in order to provide a representation closer to the underlying, discrete melody sequence.

Ryynänen uses a hidden Markov model that provides distributions over features including an 'onset strength' related to the local temporal derivative of total energy associated with a pitch. The first, "attack", state models the sharp jump in onset characteristics expected for new notes, although a bimodal distribution also allows for notes that begin more smoothly; the following "sustain" state is able to capture the greater salience (energy), narrower frequency spread, and lesser onset strength associated with continuing notes. Thus, new note events can be detected simply by noting transitions through the onset state for a particular note model in the best-path (Viterbi) decoding of the HMM.

The "Post-processing" column of Table 2.2 examines how the raw (multi) pitch tracks are further refined in order to produce the final melody estimates. In the systems proposed by Dressler, Marolt, and Paiva, post-processing involves selecting a subset of the notes or note fragment elements to form a single melody line, including gaps where no melody note is selected. In each case, the post-processing is achieved by applying sets of rules that attempt to capture the continuity of realistic melodies in terms of energy and pitch (e.g. avoiding or deleting large, brief, frequency jumps). Rules may also include some musical insights, such as preference for a particular pitch range, and for the highest or lowest (outer) voices in a set of simultaneous pitches (a polyphony). Although the system proposed by Goto does not employ an intermediate stage of note elements, it does distinguish between multiple pitch candidates via a set of interacting "tracking agents" – alternate hypotheses of the current and past pitch – that compete to acquire the new pitch estimates from the current frame, and that live or die based on a continuously-updated penalty that reflects the total strength of the past pitches they represent; the strongest agent determines the final pitch reported.

Ryynänen and Vincent both use HMMs in order to limit the dynamics of their pitch estimates (i.e. to provide a degree of smoothing that favors slowly-changing pitches). Ryynänen simply connects the per-note HMMs described above through a third, noise/background, state, and incorporates musicologically-informed transition probabilities that vary depending on an estimate of the current chord or key [74]. Vincent uses an HMM simply to smooth pitch sequences, training the transition probabilities as a function of interval size from the ground-truth melodies in the 2004 evaluation set.

The "Voicing" column of Table 2.2 considers how, specifically, the systems distinguish between the intervals where the melody is present and those where it is silent (gaps between melodies). Goto and Vincent simply report their best pitch estimate at every frame and do not admit gaps. As discussed

System	Features	Similarity	Synchronization
Raphael [60]	"Activity" and $\|STFT\|$	HMM	
Orio & Schwarz [49]	Peak Structure Distance		DP
Hu et al. [36]	Chroma	Euclidian Distance	DP
Turetsky & Ellis [72]	$\|STFT\|$	Cosine Distance	DP

Table 2.3: Representative score to audio alignment algorithms.

above, the selection of notes or fragments in the systems proposed by Dressler, Marolt, and Paiva naturally leads to gaps where no suitable element is selected; Dressler augments this with a local threshold to discount low-energy notes.

2.3 Score to Audio Alignment

Score to audio alignment is the process of synchronizing a symbolic representation with a recording. For many recordings, a corresponding score is available in the form of sheet music or a MIDI transcript. Since a recorded performance is not an exact recreation of the score, expressive and stylistic variations exist between different interpretations of the same piece of music. As such, developing a time mapping between the note labels and audio events in a given recording enables an analysis of variations between performances and has a number of practical applications ranging from content-based indexing to automatic music accompaniment. We note that the basic theory of score to audio alignment is very similar in nature to string matching in speech recognition [58] and biological sequence analysis [24].

In the majority of cases, score to audio alignment algorithms may be broken down into three stages: acoustic feature analysis, feature similarity (or distance) calculation, and time synchronization. Typically, a set of acoustic features is calculated for both the recorded audio and a synthesis of the reference transcript. Then, a similarity calculation is performed by comparing the pairs of acoustic feature vectors at discrete time steps, a process that results in a distance matrix. Finally, time alignment is accomplished by identifying the least cost path through the distance matrix. Table 2.3 displays the characteristic attributes for several score to audio alignment systems.

Like [17, 73], Raphael [60] sought to provide a framework for automatic musical accompaniment. Monophonic recordings were aligned to a reference score by identifying the optimal sequence of local note estimates via a hidden Markov model. The note sequence was observed by estimating the fundamental frequency of the performance in the magnitude-STFT domain, gated by a normalized energy, "activity", measure. In contrast to the other representative approaches, Raphael uses a HMM to perform the time-alignment. Although the HMM framework has the potential to learn sequence structure, it is di-

rectly interchangeable with dynamic time warping (DTW) [58] for pairwise sequence alignment.

Whereas the remaining approaches perform feature analysis on a feature-domain realization generated from the score by some kind of synthesis, Orio and Schwarz [49] attempted to avoid employing an explicit score synthesis to achieve alignment. As such, they proposed a specialized similarity measure, the peak structure distance (PSD). For a given set of notes from the score, PSD hypothesizes the locations of associated harmonics in the spectrum (taking for example the first 8 multiples of the expected fundamentals), then calculates the similarity of the observed spectral frames to the set of notes as the proportion of the total spectral energy that occurs within some narrow window around the predicted harmonics. As the actual spectrum tends towards pure sets of harmonics at the correct frequencies, the similarity tends to 1. This is then converted to a distance by subtracting the similarity estimate from 1. As a result, the measure neatly avoids having to model the relative energies at each harmonic.

Hu et al. [36] and Turetsky and Ellis [72] calculate acoustic features based on the magnitude-STFT; however, Hu et al. map each bin of the fast Fourier transform (FFT) into the corresponding chroma classes (i.e. the 12 semitones within an octave) they overlap, and Turetsky and Ellis explore a number of magnitude-STFT feature normalizations in order to reduce the timbral dependency on the consistency of the synthesis. In addition, the approaches differ in that Turetsky and Ellis calculate the similarity matrices based on the inner product (i.e. cosine distance) whereas Hu et al. apply a Euclidian distance metric.

Identifying the least cost path through a large distance matrix can become quite computationally expensive. As such, a number of methods have been proposed in order to optimize the dynamic programming (DP) search such as [20, 38].

2.4 Summary

In this chapter, we presented a background discussion and analysis of representative research pertaining to polyphonic pitch estimation, melody transcription, and score to audio alignment. A wide variety of approaches were reported; however a number of common themes were identified as well (e.g. the popularity of the $|STFT|$ feature representation). In the following chapters, we too employ the $|STFT|$ front-end; however, we adopt an agnostic approach to transcription in which classifiers are left to infer whatever regularities may exist in the representation of training examples taken from real music audio recordings.

Chapter 3

Melody Transcription

In this chapter, we introduce the concept of classification-based music transcription in the context of melody note discrimination. Supervised classifiers trained directly from acoustic features are used to identify the predominant melody note in a frame of audio, and the overall note sequence is smoothed via a hidden Markov model in order to reflect the temporal consistency of actual melodies. The training data has the single greatest influence on any classification system, and as such, we begin our investigation by describing the collection and generation of the audio data. We present several acoustic features and normalizations for classification and make feature comparisons based on a baseline all-versus-all support vector machine framework. In order to examine the effect of classification structure on transcription accuracy, we explore different frame-level pitch classifiers and consider the problem of distinguishing voiced (melody) and unvoiced (accompaniment) frames. Finally, we describe the addition of temporal constraints from hidden Markov models and provide an empirical analysis of the classification-based system with comparisons to alternative approaches.

3.1 Audio Data

Supervised training of a classifier requires a corpus of labeled feature vectors. In general, larger quantities of eclectic training data will give rise to more accurate classifiers. In the classification-based approach to transcription, then, a significant challenge becomes collecting suitable training data. Although the availability of digital scores aligned to real recordings is very limited, there are a number of alternative sources for obtaining relevant data. We investigated using multitrack recordings and MIDI files as training data, and we evaluated the proposed approach on recently developed standard test sets.

3.1.1 Multitrack Recordings

Popular music recordings are typically created by layering a number of independently recorded audio tracks. In some cases, artists (or their record companies) make available separate vocal and instrumental tracks as part of a CD or 12" vinyl single release. The *a capella* vocal recordings can be used to create ground truth for the melody in the full ensemble music, since a solo voice can usually be tracked at high accuracy with standard pitch tracking systems [70, 18]. Therefore, we can construct a set of ground truth labels as long as we can identify the temporal alignment between the solo track and the full recording (melody plus accompaniment). Note that the *a capella* recordings are only used to generate ground truth; the classifier is not trained on isolated voices since we do not expect to use it to transcribe such data.

A collection of multitrack recordings was obtained from genres such as jazz, pop, R&B, and rock. The digital recordings were read from CD, then downsampled into monaural files at a sampling rate of 8 kHz. The 12" vinyl recordings were converted from analog to digital mono files at a sampling rate of 8 kHz. For each song, the fundamental frequency of the melody track was estimated using the fundamental frequency estimator in WaveSurfer, which is derived from ESPS's get_f0 [66]. Estimations of the fundamental frequency were calculated at frame intervals of 10 ms and limited to the range 70–1500 Hz.

Dynamic Time Warping was used to align the *a capella* recordings and the full ensemble recordings following the procedure described in [72]. This time alignment was smoothed and linearly interpolated in order to achieve a frame-by-frame correspondence. The alignments were manually verified and corrected using WaveSurfer's graphical user interface in order to ensure the integrity of the training data. Target labels were assigned by calculating the closest MIDI note number to the monophonic estimation. An illustration of the training data generation process is displayed in Figure 3.1.

The collection of multitrack recordings resulted in 12 training excerpts ranging in duration from 20 s to 48 s. Only the voiced portions of each excerpt were used for training (we did not attempt to include an 'unvoiced' class at this stage), resulting in 226 s (i.e. 3:46) of training audio, or 22,600 frames at a 10 ms frame rate.

3.1.2 MIDI Audio

MIDI was created by the manufacturers of electronic musical instruments as a digital representation of the notes, times, and other control information required to synthesize a piece of music. As such, a MIDI file amounts to a digital music score that can easily be converted into an audio rendition. Extensive collections of MIDI files exist consisting of numerous transcriptions from diverse genres. The MIDI training data used in the following experiments was composed of several frequently downloaded pop songs from

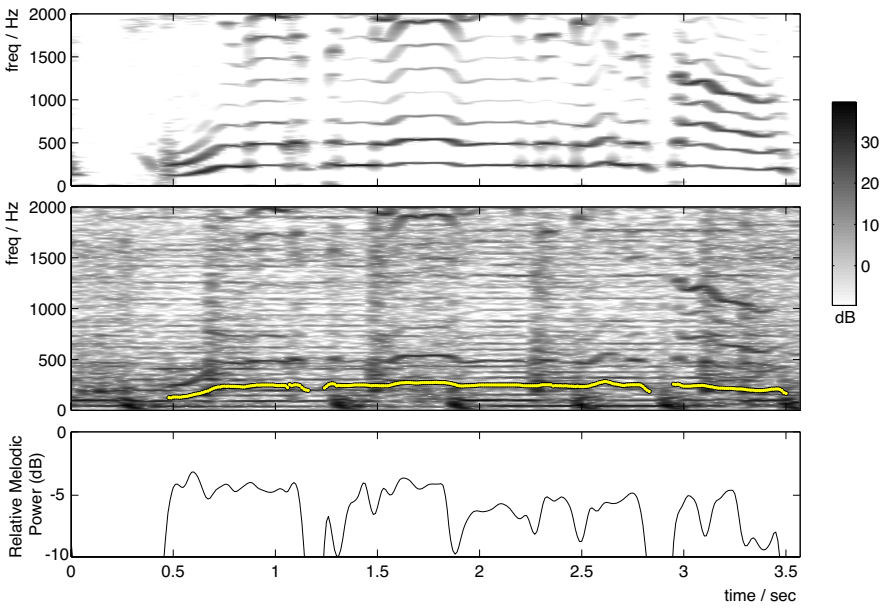

Figure 3.1: Examples from training data generation. The fundamental frequency of the isolated melody track (top pane) was estimated and time-aligned to the complete audio mix (center). The fundamental frequency estimates (overlaid on the spectrogram), rounded to the nearest semitone were used as target class labels. The bottom panel shows the power of the melody voice relative to the total power of the mix (in dB); if the mix consisted only of the voice, this would be 0 dB.

http://www.findmidis.com. The training files were converted from the standard MIDI file format to monaural audio files with a sampling rate of 8 kHz using the MIDI synthesizer in Apple's iTunes. Although completely synthesized (with the lead vocal line often assigned to a wind or brass voice), the resulting audio is quite rich, with a broad range of instrument timbres and production effects such as reverberation.

In order to identify the corresponding ground truth, the MIDI files were parsed into data structures containing the relevant audio information (i.e. tracks, channels numbers, note events, etc), and the melody was isolated and extracted by exploiting MIDI conventions. Commonly, the lead voice in pop MIDI files is stored in a monophonic track on an isolated channel. In the case of multiple simultaneous notes in the lead track, the melody was assumed to be the highest note present. Target labels were determined by sampling the MIDI transcript at the precise times corresponding to the analysis frames of the synthesized audio.

We selected five MIDI excerpts for training, each around 30 s in length. 125 s (12,500 frames) of training audio remained after we removed the unvoiced frames from the training pool.

Category	Style	Melody Instrument
Daisy	Pop	Synthesized voice
Jazz	Jazz	Saxophone
MIDI	Folk (2), Pop (2)	MIDI instruments
Opera	Classical opera	Male voice (2), Female voice (2)
Pop	Pop	Male Voice

Table 3.1: Summary of the ADC 2004 melody contest test data. Each category consists of 4 excerpts, each roughly 20 s in duration. The 8 segments in the *Daisy* and *MIDI* categories were generated using a synthesized lead melody voice, and the remaining categories were generated using multitrack recordings.

3.1.3 Resampled Audio

When the availability of a representative training set is limited, the quantity and diversity of musical training data may be extended by resampling the recordings to effect a global pitch shift. The multitrack and MIDI recordings were resampled at rates corresponding to symmetric semitone frequency shifts over the chromatic scale (i.e. $\pm 1, 2, \ldots 6$ semitones); the expanded training set consisted of all transpositions pooled together. The ground truth labels were shifted accordingly and linearly interpolated to maintain time alignment (because higher pitched transpositions also acquire a faster tempo). Using this approach, we created a smoother distribution of the training labels and reduced bias toward the specific pitches present in the training set. The classification approach relies on learning separate decision boundaries for each individual melody note with no direct mechanism to ensure consistency between similar note classes (e.g. C4 and C#4), or to improve the generalization of one note-class by analogy with its neighbors in pitch. Using a transposition-expanded training restores some of the advantages we might expect from a more complex scheme for tying the parameters of pitchwise-adjacent notes: although the parameters for each classifier are separate, classifiers for notes that are similar in pitch have been trained on transpositions of many of the same original data frames. Resampling expanded the total training pool by a factor of 13 to around 456,000 frames.

3.1.4 Validation and Test Sets

Research progress benefits when a community agrees on a consistent definition of their problem of interest, then goes on to define and assemble standard tests and data sets. Recently, the music information retrieval community developed formal evaluations for the melody transcription problem, starting with the Audio Description Contest at the 2004 International Conference on Music Information Retrieval (ISMIR/ADC 2004) [32] and continuing with the Music Information Retrieval Evaluation eXchange (MIREX) [56]. The ADC

Melody Instrument	Style
Human voice (8 f, 8 m)	R&B (6), Rock (5), Dance/Pop (4), Jazz (1)
Saxophone (3)	Jazz
Guitar (3)	Rock guitar solo
Synthesized Piano (3)	Classical

Table 3.2: Summary of the MIREX 2005 melody evaluation test data.

2004 test set for melody estimation is composed of 20 excerpts, four from each of five styles, each lasting 10-25 s, for a total of 366 s of test audio. A description of the data used in the 2004 evaluation is displayed in Table 3.1. The corresponding reference data was created by using SMSTools [6] to estimate the fundamental frequency of the isolated, monophonic melody track at 5.8 ms steps. As a convention, the frames in which the main melody is unvoiced were labeled 0 Hz. The transcriptions were manually verified and corrected using the graphical user interface in order to ensure the quality of the reference transcriptions. Unless otherwise noted, the ADC 2004 test set was used as the development set in the experiments described in this chapter.

Since the ADC 2004 data was distributed after the competition, an entirely new test set of 25 excerpts was collected for the MIREX 2005 evaluation consisting of 25 excerpts ranging in length from 10-40 s, providing 536 s of total test audio. The same audio format was used as in the 2004 evaluation; however, the ground-truth melody transcriptions were generated at 10 ms steps (in order to accommodate non-frame-based approaches) using the ESPS get_f0 method implemented in WaveSurfer [66]. As displayed in Table 3.2, the 2005 test data was more heavily biased toward a pop-based corpora rather than uniformly weighting the segments across a number of styles or genres as in the 2004 evaluation. The shift in the distribution was motivated both by the relevance of commercial applications for music organization and by the availability of multitrack recordings in the specified genres. Since the 2005 test set is more representative of real-world recordings, it is inherently more complex than the preceding test set. In Section 3.6, we evaluate the classification-based system on the MIREX 2005 test set and provide comparisons to a number of alternative approaches to melody transcription.

3.2 Acoustic Features

The acoustic feature representation described in this chapter is based on the ubiquitous and well-known spectrogram, which converts a sound waveform into a distribution of energy over time and frequency. The spectrogram is commonly displayed as a pseudo-color or grayscale image as shown in the middle pane of Figure 3.1, and the basic acoustic features for each time-frame may be considered vertical slices through such an image. Specifically, the original music recordings (melody plus accompaniment) were combined into

a single (mono) channel and down-sampled to 8 kHz. The short-time Fourier transform was applied using $N = 1024$ point transforms (i.e. 128 ms), an N-point Hanning window, and an 80 point advance between adjacent windows (for a 10 ms hop between successive frames). Only the coefficients corresponding to frequencies below 2 kHz (i.e. the first 256 spectral bins) were used in the representative feature vector.

An analysis of preprocessing schemes was made by measuring the influence of feature normalization on a baseline classifier. A C-way, all-versus-all (AVA) algorithm for multi-class discrimination based on support vector machines trained by sequential minimal optimization [52] as implemented in the Weka toolkit [78] was used as the baseline pitch classifier in the acoustic feature comparison. In this scheme, a majority vote was taken from the output of $(C^2 - C)/2$ discriminant functions, comparing every possible pair of classes. For computational reasons, the AVA classification experiments were restricted to a linear kernel.

Each audio frame was represented by a 256-element input vector, with $C = 60$ classes corresponding to five-octaves of semitones from G2 to F#7. In order to classify the dominant melodic pitch for each frame, we assume the melody note at a given instant to be solely dependent on the normalized frequency data below 2 kHz. For the acoustic feature analysis experiments, we further assume each frame to be independent of all other frames. Additional experiments and details regarding the classification framework will be presented in Section 3.3.

Separate AVA classification systems were trained using six different feature normalizations. Of these, three feature sets were based on the STFT, and three were based on the (pseudo)autocorrelation. In the first feature representation, the audio data was simply represented by the magnitude of the STFT normalized such that the the energy in each frame was bounded by zero and one. For the second case, the magnitudes of the spectral bins were normalized by subtracting the mean and dividing by the standard deviation calculated in a 71-point sliding frequency window along the columns of the spectrogram. The goal of the 71-point normalization feature is to remove some of the variational influence due to differences in instrumentation and context between the training and testing data. For the third STFT-based normalization scheme, cube-root compression, which is commonly used as an approximation to the loudness sensitivity of the ear, was applied to the magnitude STFT in order to make larger spectral magnitudes appear more similar.

In order to create the fourth set of features, the autocorrelation was calculated by taking the inverse Fourier transform (IFT) of the magnitude of the STFT for the original windowed waveform. Similarly, the fifth feature set, the cepstrum, was generated by calculating the IFT of the log-STFT-magnitude. Note that the cepstrum also performs a sort of timbral normalization because the overall gain and broad spectral shape are separated into the first few cepstral bins whereas periodicity appears at the higher indexes. In addition, we at-

	Training data		
Normalization	**Multitrack**	**MIDI**	**Both**
STFT	56.4%	50.5%	62.5%
71-pt norm	54.2%	46.1%	**62.7%**
Cube root	53.3%	51.2%	62.4%
Autocorr	55.8%	45.2%	62.4%
Cepstrum	49.3%	45.2%	54.6%
LiftCeps	55.8%	45.3%	62.3%

Table 3.3: Effect of normalization: raw pitch accuracy results on a withheld portion of the training set for each of the normalization schemes considered, trained on either multitrack audio alone, MIDI syntheses alone, and both data sets combined. (The size of the training sets was held constant, so the results are not directly comparable to the other results reported in this chapter.)

tempted to normalize the autocorrelation-based features by liftering (scaling the higher-order cepstral by an exponential weight).

A comparison of the raw pitch accuracy for the classifiers trained on each of the different normalization schemes is displayed in Table 3.3. We show separate results for the classifiers trained on multitrack audio alone, MIDI syntheses alone, and both data sources combined. The raw pitch accuracy results correspond to melodic pitch transcription to the nearest semitone.

The most obvious result displayed in Table 3.3 is that all the features, with the exception of the cepstrum, result in a very similar transcription accuracy (although the across-frequency local normalization provides a slight performance advantage). This result is not altogether surprising since all the features contain largely equivalent information, but it also raises the question as to how effective the normalization (and hence the system generalization) has been. It may be that a better normalization scheme remains to be discovered. Looking across the columns in the table, we see that the more realistic multitrack data forms a better training set than the MIDI syntheses, which have much lower acoustic similarity to most of the evaluation excerpts. Using both, and hence a more diverse training set, always gives a significant accuracy boost – up to 9% absolute improvement, as observed for the best-performing 71-point normalized features.

The impact of including the training data transposed by resampling over ± 6 semitones is displayed in Table 3.4. The inclusion of the resampled data results in a substantial 7.5% absolute improvement in raw pitch accuracy, an effect that underscores the value of broadening the range of data seen for each individual note.

Training Set	# Training Frames	Raw Pitch Acc
No resampling	8,500	60.2%
With resampling	110,500	**67.7%**

Table 3.4: Impact of resampling the training data: raw pitch accuracy results on the ADC 2004 test set for systems trained on the entire training set, either without any resampling transposition, or including transpositions out to ± 6 semitones (i.e 500 frames per transposed excerpt, 17 excerpts, 1 or 13 transpositions).

3.3 Melody Classification

In the previous section, we showed that classification accuracy seems to depend more strongly on training data diversity than on feature normalization. It may be that the SVM classifier applied in the acoustic feature analysis was better able to generalize than the explicit feature normalizations. In this section, we examine the effects of different classifier types on transcription accuracy and the influence of the total amount of training data used.

The support vector machine (SVM) [15] was selected as the learning method to be used in the following classification experiments. The SVM is a supervised classification system that employs a hypothesis space of linear functions in a high-dimensional feature space in order to learn separating hyperplanes that are maximally distant from all training patterns. As such SVM classification attempts to generalize an optimal decision boundary between classes of data. Labeled training data in a given space are thus separated by a maximum-margin hyperplane through SVM classification. Complete tutorials and the underlying mathematical formulation for the SVM may be found in [5, 16].

3.3.1 C-way All-Versus-All SVM Classification

Our baseline classifier is the AVA SVM as described in Section 3.2. Given the large amount of training data used in the evaluation (over 10^5 frames), we selected a linear kernel, which requires training time on the order of the number of feature dimensions cubed for each of the $O(C^2)$ discriminant functions. More complex kernels (such as radial basis functions, which require training time on the order of the number of *instances* cubed) were computationally infeasible due to the size of the training set.

We sought to determine the number of training instances to include from each audio excerpt in the first classification experiment. The number of training instances selected from each song was varied using both incremental sampling (taking a limited number of frames from the beginning of each excerpt) and random sampling (selecting the frames from anywhere in the excerpt), as displayed in Figure 3.2. Randomly sampling feature vectors to train on

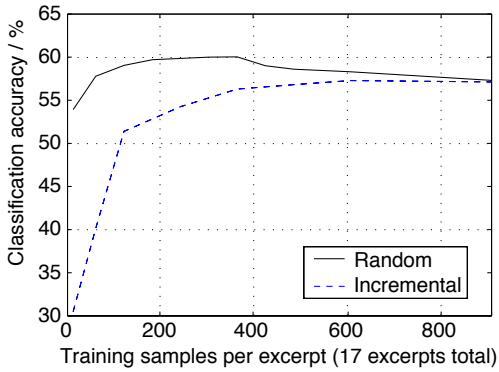

Figure 3.2: Variation of classification accuracy with number of training frames per excerpt. Incremental sampling takes frames from the beginning of the excerpt; random sampling takes them from anywhere. The training set does not include resampled (transposed) data.

approaches an asymptote much more rapidly than adding the data in chronological order. In addition, random sampling appears to exhibit symptoms of over-training.

The observation that random sampling achieves peak accuracy within approximately 400 samples per excerpt (out of a total of around 3000 samples for a 30 s excerpt with 10 ms hops) may be explained by both signal processing and musicological considerations. Firstly, adjacent analysis frames are highly overlapped, sharing 118 ms out of a 128 ms window, and thus their feature values will be very highly correlated (10 ms is an unnecessarily fine time resolution to generate training frames, but it is the standard used in the evaluation). From a musicological point of view, musical notes typically maintain approximately constant spectral structure over hundreds of milliseconds; a note should maintain a steady pitch for some significant fraction of a beat to be perceived as well-tuned. If we assume there are on average 2 notes per second (i.e. around 120 bpm) in the pop-based training data, then we expect to see approximately 60 melodic note events per 30 s excerpt. Each note may contribute a few usefully different frames to tuning variation such as vibrato and variations in accompaniment. Thus we expect many clusters of largely redundant frames in the training data, and random sampling down to 10% (or closer to one frame every 100 ms) seems reasonable.

The observation that analysis frames are highly overlapped also gives us a perspective on how to judge the significance of differences in these results. For example, the ADC 2004 test set consists of 366 s, or 36,600 frames using the standard 10 ms hop. A simple binomial significance test may be used to compare classifiers by estimating the likelihood that random sets of *independent* trials could produce the observed differences in empirical error rates from an equal underlying probability of error. Since the standard error of such an observation falls as $1/\sqrt{T}$ for T trials, the significance interval depends

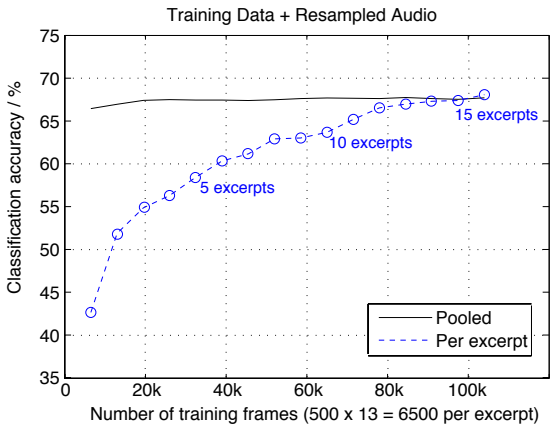

Figure 3.3: Variation of classification raw pitch accuracy with the total number of excerpts included, compared to sampling the same total number of frames from all excerpts pooled. This data set includes 13 resampled versions of each excerpt, with 500 frames randomly sampled from each transposition.

directly on the number of trials. However, the arguments and observations above show that the 10 ms frames are anything but independent; to obtain something closer to independent trials, we should test on frames no less than 100 ms apart, and 200 ms sampling (5 frames per second) would be a safer choice. This corresponds to only 1,830 independent trials in the test set; a one-sided binomial significance test suggests that differences in frame accuracies on this test of less than 2.5% are not statistically significant at the accuracies reported in this paper.

In the second classification experiment, we examined the incremental gain from adding novel training excerpts. The effect of increasing the number of excerpts (from one to 16) used to train the classification system on raw pitch accuracy is displayed in Figure 3.3. In this case, each additional excerpt consisted of adding 500 randomly-selected frames from each of the 13 resampled transpositions described in Section 3.1, or 6,500 frames per excerpt. Thus, the largest classifier was trained on 104k frames as compared to the approximately 15k frames used to train the largest classifier in Figure 3.2. The solid curve in Figure 3.3 displays the result of training on the same number of frames randomly drawn from the pool of the entire training set. Again, we notice that the system trained from pool of total frames appears to reach an asymptote by 20k total frames, or fewer than 100 frames per transposed excerpt. We suspect, however, that the level of this asymptote was determined by the total number of excerpts. That is, we believe that the "per excerpt" trace will continue to climb upwards if additional novel training data was available.

Classifier	Kernel	Raw Pitch	Chroma
AVA SVM	Linear	67.7%	72.7%
OVA SVM	Linear	69.5%	74.0%
OVA SVM	RBF	**70.7%**	**74.9%**

Table 3.5: Raw pitch accuracy for multi-way classification systems based on all-versus-all (AVA) and one-versus-all (OVA) structures. Accuracy results are provided for both raw pitch transcription and chroma transcription (which ignores octave errors).

3.3.2 Multiple One-Versus-All SVM Classification

In addition to the C-way melody classification, 60 binary one-versus-all (OVA) SVM classifiers were trained representing each of the notes present in the resampled training set. The distance-to-classifier-boundary hyperplane margins were treated as a proxy for a log-posterior probability for each of the classes. Pseudo-posteriors (up to an arbitrary scaling power) were obtained from the distance-to-classifier boundary by fitting a logistic model to the data. In the OVA framework, transcription was achieved by selecting the most probable class at each time frame. While OVA approaches are generally viewed as less sophisticated, [61] presents evidence that they can match the performance of more complex multi-way classification schemes. An example 'posteriorgram' (note-class-versus-time image showing the posteriors of each class at a given time step) for a pop excerpt is displayed with the ground truth labels overlaid in the bottom pane of Figure 3.4.

Since the number of classifiers required in the OVA framework is $O(C)$ (rather than the $O(C^2)$ classifiers required for the AVA approach) it becomes computationally feasible to experiment with alternative classifier kernels. The best result classification rates for each of the SVM systems examined are displayed in Table 3.5. Both OVA classifiers provide a marginal performance advantage over the pairwise classifier (with a slight edge favoring the OVA SVM system that uses an RBF kernel).

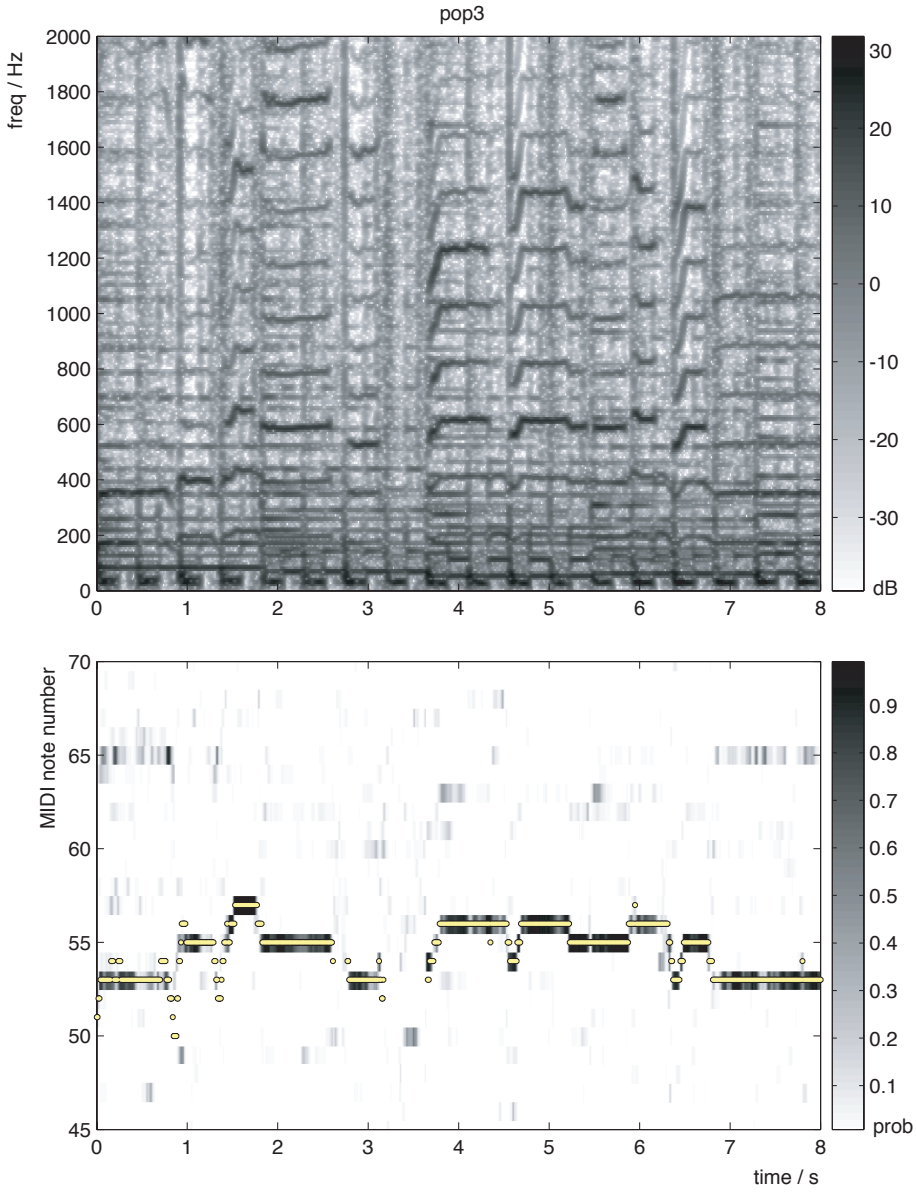

Figure 3.4: Spectrogram and posteriorgram (pitch probabilities as a function of time) for the first 8 s of pop music excerpt "pop3" from the ADC 2004 test set. The ground-truth labels, plotted on top of the posteriorgram, closely track the mode of the posteriors for the most part. However, this memoryless classifier also regularly makes hare-brained errors that can be corrected through HMM smoothing.

Classifier	Voicing Detection	Voicing FA	Voicing d'	Voicing Frame Acc
All Voiced	100%	100%	0	85.6%
Energy Threshold	88.0%	32.3%	**1.63**	78.4%
Linear SVM	76.1%	46.4%	0.80	73.0%
RBF SVM	82.6%	48.1%	0.99	78.3%

Table 3.6: Voicing detection performance. "Voicing Detection" is the proportion of voiced frames correctly labeled; "Voicing FA" is the proportion of unvoiced frames incorrectly labeled, so labeling all frames as voiced scores 100% on both counts, as displayed in the first row. d' is a measure of a detector's sensitivity that attempts to factor out the overall bias toward labeling any frame as voiced. "Voicing Frame Acc" is the proportion of all frames given the correct voicing label.

3.4 Voiced Frame Detection

Complete melody transcription involves not only deciding the note of frames where the main melody is active, but also discriminating between melody and non-melody (accompaniment) frames. In this section, we briefly describe two approaches for classifying instants as voiced (dominant melody present) or unvoiced (no melody present).

In the first approach we considered, voicing detection was performed by implementing a simple energy threshold. Spectral energy in the range $200 < f < 1800$ Hz was summed for every 10 ms frame. Each energy sum value was normalized by the median energy in that band for the given excerpt, and instants were classified as voiced or unvoiced by a global threshold as tuned on a small development set. Since the melody instrument is usually given a prominent level in the final musical mix, this approach is generally quite successful (particularly after we have filtered out the low-frequency energy of bass and drums).

In keeping with the classification-based approach, we also attempted to train binary SVM classifiers (using both linear and RBF kernels) based on the normalized magnitude of the STFT. The voiced melody classification statistics are displayed in Table 3.6. Although we had hoped that the classifiers would learn particular spectral cues as to the presence of the melody, the corresponding data shows that the simple energy threshold provides better voicing detection results. However none of the voicing detectors investigated resulted in a higher frame-level accuracy than simply labeling all the frames as voiced. Due to the fact that the ADC 2004 test data is more than 85% voiced, any classifier that attempts to identify unvoiced frames risks making more mislabeling mistakes than unvoiced frames correctly detected. As such, we also report the performance in terms of d', a measure of a detector's sensitivity that attempts to factor out the overall bias toward labeling any frame as voiced (complete definitions of the evaluation metrics are provided in Section 3.6).

While the proposed voicing detection scheme is simple and not particularly accurate, it is not the main focus of the current work. The energy threshold enables the identification of nearly 90% of the melody-containing frames, without resorting to the crude choice of simply treating every frame as voiced. However, more sophisticated approaches to learning classifiers for tasks with highly-skewed priors offer a promising future direction [10].

3.5 Hidden Markov Model Post Processing

The posteriorgram in Figure 3.4 clearly illustrates both the strengths and weaknesses of the classification approach to melody transcription. The success of the approach in estimating the correct melody pitch from audio data is clear in the majority of frames. However, the result also displays the obvious fault of classifying each frame independently of its neighbors: the inherent temporal structure of music is not exploited. In this section, we attempt to incorporate the sequential structure that may be inferred from musical signals by using hidden Markov models (HMMs) to impose temporal constraints[1].

3.5.1 HMM State Dynamics

Similarly to our data driven approach to classification, we attempt to learn the temporal structure of music directly from the training data. In the proposed framework, the HMM states correspond directly to a given melody pitch. As such, the state dynamics (transition matrix and class priors) can be estimated from the 'directly observed' state sequences (i.e. the ground-truth transcriptions of the training set). The note class prior probabilities, generated by counting all frame-based instances from the resampled training data, and the note class transition matrix, generated by observing all frame-to-frame note transitions, are displayed in Figure 3.5 (a) and (b) respectively.

Note that although some bias has been removed in the note priors by symmetrically resampling the training data, the sparse nature of a transition matrix learned from a limited training set is likely to generalize poorly to novel data. In an attempt to mitigate this lack of generalization, each element of the transition matrix was replaced by the mean of the corresponding matrix diagonal. This generalization is equivalent to assuming that the probability of making a transition between two pitches depends only on the interval between them (in semitones) and not on their absolute frequency. The resulting normalized state transition matrix is displayed in Figure 3.5 (c).

[1] A complete tutorial on the formulation and use of hidden Markov models is available in [57].

3.5.2 Smoothing Discrete Classifier Outputs

We can use an HMM to apply temporal smoothing even if we only consider the labels assigned by the frame-level classifier at each stage and entirely ignore any information on the relatively likelihood of other labels that might have been available prior to the final hard decision being made by the classifier. If the model state at time t is given by q_t, and the classifier output label is c_t, then the HMM will achieve temporal smoothing by finding the most likely (Viterbi) state sequence i.e. maximizing

$$\prod_t p(c_t|q_t)p(q_t|q_{t-1}) \tag{3.1}$$

where $p(q_t|q_{t-1})$ is the transition matrix estimated from the ground-truth melody labels as described in the previous subsection. However, we still need $p(c_t|q_t)$, the probability of seeing a particular classifier label c_t given a true pitch state q_t. Since we cannot directly observe $p(c_t|q_t)$, we may estimate it from the confusion matrix of classified frames (i.e. counts normalized to give $p(c_t, q_t)$) on a development corpus. For practical purposes, we reused the training set to generate the normalized counts. Reusing the training data might lead to an overoptimistic belief about how well c_t will reflect q_t, but it was the only reasonable option due to the limited nature of labeled data. The raw confusion matrix normalized by columns (q_t) to give the required conditionals is displayed in Figure 3.5 (d), and the normalized confusion matrix regularized such that all elements along a given diagonal are equal is displayed in Figure 3.5 (e). Other than a small moving average, a normalization was not applied to the zero (unvoiced) state. From the confusion (observation) matrix, we can see that the most frequently confused classifications are between members of the same chroma (i.e. separated by one or more octaves in pitch) and between notes with adjacent pitches (separated by one semitone).

For the total transcription problem (dominant melody transcription plus voicing detection), the baseline (memoryless) transcription was estimated by simply gating the pitch classifier output with the binary energy threshold. If at each instant we use the corresponding column of the observation (confusion) matrix in the Viterbi decoder dynamic-programming local-cost matrix, we can derive a smoothed state sequence that removes short, spurious excursions of the raw pitch labels. Despite the paucity of information obtained from the classifier, a small, yet robust, absolute improvement of 0.9% in overall accuracy is obtained from using this approach as displayed in Table 3.7.

3.5.3 Exploiting Classifier Posteriors

The OVA classification system was constructed in order to approximate log-posteriors for each pitch class, and we can use this detailed information to improve the HMM decoding. Rather than guessing the local likelihood of a particular note given a single output from the classification system, the

Classifier	Voicing Acc	Raw Pitch Acc	Overall Acc
Memoryless	85.1%	71.2%	70.7%
MemlessAllVx	86.1%	76.8%	66.1%
Discrete	83.6%	71.9%	71.6%
Posteriors	86.2%	74.5%	**73.2%**
PostAllVx	86.1%	**79.4%**	68.3%

Table 3.7: Melody transcription frame accuracy percentages for different systems with and without HMM smoothing. "Voicing Acc" is the proportion of frames whose voicing state is correctly labeled. "Raw Pitch Acc" is the proportion of pitched frames assigned the correct pitch label. "Overall Acc" is the proportion of all frames assigned both the correct voicing label, and, for voiced frames, the correct pitch label. All results are based on the one-versus-all SVM classifier using an RBF kernel. The "Memoryless" classifier simply takes the most likely label from the frame classifier after gating by the voicing detector; "MemlessAllVx" ignores the voicing detection and reports a pitch for all frames (to maximize pitch accuracy). "Discrete" applies HMM smoothing to this label sequence without additional information, "Posteriors" uses the pseudo-posteriors from the OVA SVM to generate observation likelihoods in the HMM, and "PostAllVx" is the same except the unvoiced state is excluded (by setting its frame posterior to zero).

likelihood of each note may be directly observed from each binary classifier. Thus, if the acoustic data at each time is x_t, we may regard our OVA classifiers as giving us estimates of

$$p(q_t|x_t) \propto p(x_t|q_t)p(q_t) \tag{3.2}$$

i.e. the posterior probabilities of each HMM state given the local acoustic features. Thus, by dividing each (pseudo)posterior by the prior of that note, we obtain scaled likelihoods that can be employed directly in the Viterbi search for the solution of Equation 3.1. The unvoiced state needs special treatment, since it is not considered by the main classifier. We attempted several approaches for incorporating an estimate of the unvoiced state including decoding the pitch HMM with the unvoiced state excluded (setting its observation likelihood to zero), then applying voicing decisions from a separate voicing HMM, and setting the observation posterior of the unvoiced state to $1/2 \pm \alpha$, where α was tuned on the development (training set), which provided significantly better results.

As shown in Table 3.7, using the posteriors as HMM features results in an additional absolute improvement of 1.6% total frame accuracy over using only the 1-best classification information. More impressively, the absolute accuracy on the pitched frames jumps 3.3% as compared to the memoryless case, since knowing the per-note posteriors helps the HMM to avoid very unlikely notes when it decides to stray from the one-best label assigned by the classifier. If we focus only on raw pitch accuracy (i.e. exclude the frames whose

ground truth is unvoiced from scoring), we can maximize pitch accuracy with a posterior-based HMM decode that excludes the unvoiced state, achieving a pitch accuracy of 79.4%, or 2.6% better than the comparable unsmoothed case.

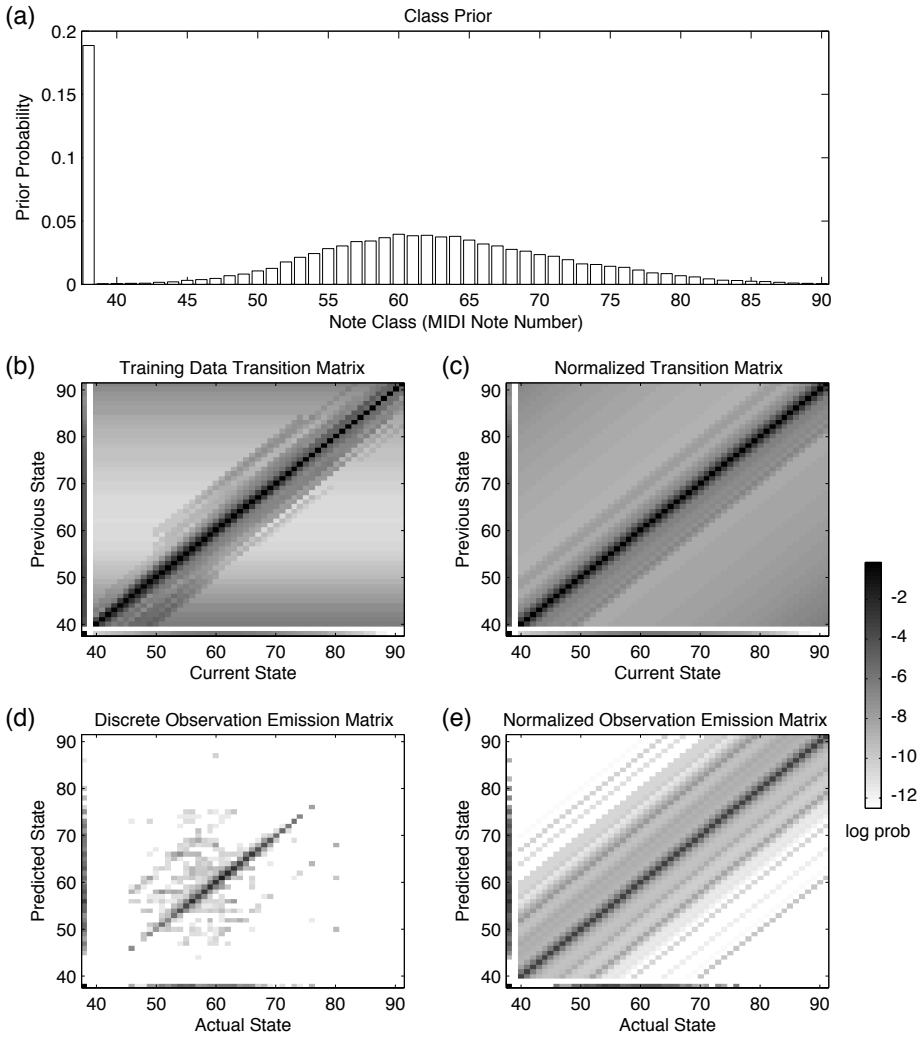

Figure 3.5: Hidden Markov parameters learned from the ground truth and confusion matrix data. Top (pane (a)): class priors. Middle: state transition matrix, raw (pane (b)), and regularized across all notes (pane (c)). Bottom: Observation likelihood matrix for the discrete-label smoothing HMM, raw (pane (d)) and regularized across notes (pane (e)). For panes (b) to (e), the state densities are displayed in the first column and last row of each figure.

3.6 Experimental Analysis

In this section, we present the results of a full-scale melody transcription evaluation, the MIREX melody transcription contest, that sought to provide an objective comparison of approaches to melody transcription by developing a standardized test set and a consensus regarding evaluation metrics. The evaluation was conducted in 2005 in association with the International Conference on Music Information Retrieval and repeated in 2006.

3.6.1 Evaluation Metrics

Algorithms submitted to the MIREX melody transcription contest were required to estimate the fundamental frequency of the predominant melody on a regular time grid. In order to enable more detailed insight into the structure of each submitted system, participants were allowed to perform pitch estimation and voicing detection independently, i.e., each algorithm could give its best guess for a melody pitch even for frames that it reported as unvoiced. An attempt was made to evaluate the lead voice transcription at the lowest level of abstraction, and as such, the concept of segmenting the fundamental frequency predictions into notes was omitted from consideration. The metrics used in the evaluation were agreed upon by the participants in a discussion period prior to the algorithm submission deadline. A brief description of the evaluation metrics is provided below:

- The algorithms were ranked according to the **overall transcription accuracy**, a measure that combines the pitch transcription and voicing detection tasks. It is defined as the proportion of frames correctly labeled with both raw pitch accuracy and voicing detection.
- The **raw pitch accuracy** is defined as the proportion of voiced frames in which the estimated fundamental frequency is within $\pm 1/4$ tone of the reference pitch (including the pitch estimation for frames estimated unvoiced).
- The **raw chroma accuracy** is defined in the same manner as the raw pitch accuracy; however, both the estimated and reference frequencies are mapped into a single chroma in order to forgive octave transpositions.
- The **voicing detection rate** is the proportion of frames labeled voiced in the reference transcription that are estimated voiced by the algorithm.
- The **voicing false alarm rate** is the proportion of frames that are not truly voiced that are estimated as voiced.
- The **discriminability** d' is a measure of the sensitivity of a detector that attempts to factor out the overall bias toward labeling any frame as

Participant	Overall Accuracy	Raw Pitch	Raw Chroma	Voicing Detection	Voicing FA	Voicing d'
Dressler 06 [21]	**73.2%**	**77.7%**	**82.0%**	89.3%	28.8%	1.80
Dressler	71.4%	68.1%	71.4%	81.8%	17.3%	**1.85**
Ryynänen 06 [63]	67.9%	71.5%	75.0%	78.2%	16.5%	1.75
Ryynänen	64.3%	68.6%	74.1%	90.3%	39.5%	1.56
Poliner 06 [26]	63.0%	66.2%	70.4%	93.5%	45.1%	1.64
Poliner	61.1%	67.3%	73.4%	91.6%	42.7%	1.56
Paiva [50]	61.1%	58.5%	62.0%	68.8%	23.2%	1.22
Marolt [44]	59.5%	60.1%	67.1%	72.7%	32.4%	1.06
Sutton [69]	53.7%	56.4%	60.1%	64.5%	13.8%	1.46
Goto * [33]	49.9%	65.8%	71.8%	99.9%	99.9%	0.59
Vincent * [76]	47.9%	59.8%	67.6%	96.1%	93.7%	0.23
Brossier * [4]	31.9%	41.0%	56.1%	99.5%	98.2%	0.46

Table 3.8: Results of the MIREX Audio Melody Transcription evaluation. Results marked with a * are not directly comparable to the others because those systems did not perform voiced/unvoiced detection. For brevity, systems are referred to by their first author alone.

voiced (which can move both hit rate and false alarm rate up and down in tandem). It converts the hit rate and false alarm into standard deviations away from the mean of an equivalent Gaussian distribution, and reports the difference between them. A larger value indicates a detection scheme with better discrimination between the two classes [22].

Each algorithm was evaluated on the 25 test songs described in Table 3.2, and the results of the evaluation are presented in the following subsection.

3.6.2 Empirical Results

The combined results of the 2005 and 2006 MIREX melody transcription evaluations are displayed in Table 3.8. Since each research group was allowed to make multiple submissions, only the top performing algorithm for each group in a given year is presented. The memoryless, C-Way AVA SVM system was submitted to the 2005 evaluation, and a linear kernel (for computational efficiency), OVA classification system with a posterior-based HMM smoothing stage was submitted in 2006.

In both the 2005 and 2006 competitions, the classification-based approach was the third best performing melody transcription submission for each of the evaluation metrics considered. Comparing the proposed approach to the other submissions, we note that, although the Dressler system outperformed all other submissions, the classification-based system performed near the top based on overall and raw pitch accuracy. As such, the classification-based systems appear to be one of the better approaches for generating candidate note estimates.

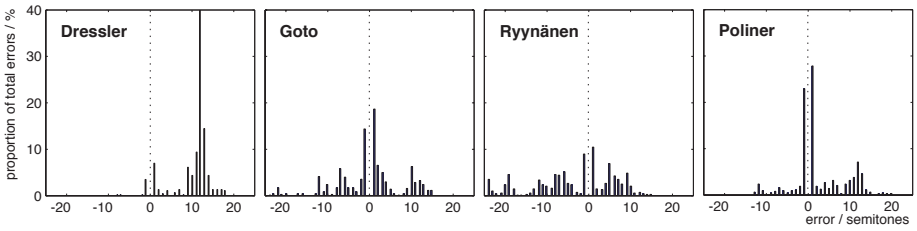

Figure 3.6: Transcription error histograms where the relative frequency of errors is plotted against the number of semitones deviation from the reference pitch.

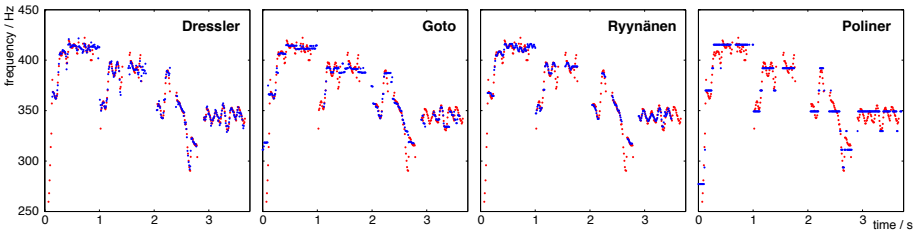

Figure 3.7: Examples of melody transcription estimations from several of the submitted systems as compared to the ground truth (light dots) for a 3.7 s excerpt from "Frozen" by Madonna.

We may gain additional insights into the nature of the classification approach by more closely examining the output of the proposed transcription system. Figure 3.6 displays a note error histogram for the classification-based system as compared with a few of the top performing MIREX submissions. In stark contrast to the other systems, the classification approach exhibits a significant number of adjacent note errors due to discretizing estimates to the nearest semitone. We suspect that this result is due to the fact that the note-level classifiers may have more difficulty resolving acoustic effects such as vibrato and 'mistuned' notes. These effects are illustrated in Figure 3.7 in which a number of example transcriptions are provided for an excerpt of "Frozen" by Madonna. Whereas algorithms that track the fundamental frequency of the lead melody voice (e.g. Dressler) follow the reference transcript quite closely and provide a clear representation of the acoustic effects, the classification-based system, which discretizes estimates to the nearest semitone, provides a representation more closely associated with the note level of abstraction.

Contrasting the classification submissions year over year, we observe that, in discordance with the classification results reported in Section 3.3.1, the AVA system provided a slight advantage in raw pitch accuracy as compared with the OVA system. However, we suspect that this result is an artifact of the statistical significance limitations of the ADC and MIREX data sets. Turning our attention to the raw chroma metric, we note that this measure indirectly evaluates the candidate note identification stage and hints at the potential for improvement in post processing. The HMM post-processing stage minimizes

the difference between the raw pitch and chroma accuracy by including a model of the melody note transitions thereby reducing erroneous octave transpositions. In addition, the HMM improves the voicing detection estimates by imposing temporal context on the voiced/unvoiced transitions. Thus, the HMM provided a modest improvement in overall accuracy by limiting the local melody steps and improving the voicing detection estimates.

3.7 Summary

In this chapter, we presented a classification approach to melody transcription. We have shown that a pure machine learning approach to melody transcription is viable and that such an approach can be successful even when based on a modest amount of training data. Although the quality of diverse training data was demonstrated to have the greatest impact on transcription accuracy, modest improvements in performance may be obtained by implementing more complex classification structures and by temporally constraining the classification estimates via hidden Markov models. The concepts formulated herein will serve as the underpinnings for a general classification-based music transcription framework to be investigated in the following chapters.

Chapter 4

Polyphonic Music Transcription

In this chapter, we present a classification approach to polyphonic music transcription. Whereas melody transcription, as described in Chapter 3, consists of estimating the single pitch corresponding to the most salient note, the polyphonic transcription problem entails resolving multiple simultaneous notes in a given period of time. As such, we extend the C-way, one-versus-all classification system developed in Chapter 3 to transcribe the *set* of notes for which the pseudo-posterior probability exceeds a threshold in a given frame. We first examine the complete transcription problem in the context of polyphonic piano transcription, then we generalize the approach to an instrument agnostic multiple fundamental frequency estimation framework. Similarly to Chapter 3, we begin our investigation with the description of our data collection and feature analysis. Support vector machines trained on spectral features are used to classify frame-level note instances, and the classifier estimates are temporally constrained via hidden Markov models. The polyphonic pitch classification system is used to transcribe piano and ensemble recordings, and empirical analyses, as well as comparisons to alternative approaches, are presented.

4.1 Audio Data and Features

4.1.1 Audio Data

The audio data used to train and test the instrument specific piano transcription system was derived directly and indirectly from MIDI transcripts collected from the Classical Piano Midi Page, http://www.piano-midi.de/. The 130 piece data set was randomly split into 92 training, 25 testing, and 13 val-

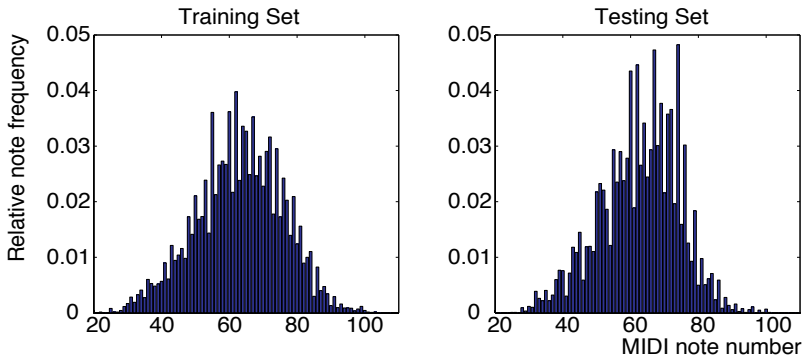

Figure 4.1: Note distributions for the piano transcription training and testing sets.

idation pieces. Table 4.5 gives a complete list of the composers and pieces used in the experiments.

Following the procedure described in Chapter 3, The MIDI files were converted from the standard MIDI file format to monaural audio files with a sampling rate of 8 kHz using the synthesizer in Apple's iTunes. In order to identify the corresponding ground truth transcriptions, the MIDI files were parsed into data structures containing the relevant audio information (i.e. tracks, channels numbers, note events, etc). Target labels were determined by sampling the MIDI transcript at the precise times corresponding to the analysis frames of the synthesized audio.

In addition to the synthesized audio, piano recordings were made from a subset of the MIDI files using a Yamaha Disklavier playback grand piano. 20 training files and 10 testing files were randomly selected for recording. The MIDI performances were recorded as monaural audio files at a sampling rate of 44.1 kHz and time-aligned to the MIDI score by identifying the maximum cross-correlation between the recorded audio and the synthesized MIDI file.

The first minute from each song in the data set was selected for experimentation which provided us with a total of 112 minutes of training audio, 35 minutes of testing audio, and 13 minutes of audio for parameter tuning on the validation set. This amounted to 56497, 16807, and 7058 note instances in the training, testing, and validation sets respectively. The note distributions for the training and test sets are displayed in Figure 4.1

4.1.2 Acoustic Features

In Section 3.2, it was observed that classification accuracy appears to exhibit a very weak dependence on variations in feature representations. A number of spectral feature normalizations were attempted for melody classification; however, none of the representations provided a significant advantage

in classification accuracy. As such, we did not repeat the acoustic feature experiments described in Section 3.2 for polyphonic transcription. Instead, we have selected the best performing normalization from the melody transcription feature analysis; however, as we will show in the following section, the greatest gain in classification accuracy is obtained from a larger and more diverse training set.

In order to generate the acoustic features, the short-time Fourier transform was applied to the audio files using $N = 1024$ point Discrete Fourier Transforms (i.e. 128 ms), an N-point Hanning window, and an 80 point advance between adjacent windows (for a 10 ms hop between successive frames). In an attempt to remove some of the influence due to timbral and contextual variation, the magnitudes of the spectral bins were normalized by subtracting the mean and dividing by the standard deviation calculated in a 71-point sliding frequency window. The live piano recordings were down-sampled to 8 kHz using an anti-aliasing filter prior to feature calculation in order to reduce the spectral dimensionality. Separate one-versus-all SVM classifiers were trained on the spectral features for each of the 88 piano keys with the exception of the highest note, MIDI note number 108. For MIDI note numbers 21 to 83 (i.e. the first 63 piano keys), the input feature vector was composed of the 256 coefficients corresponding to frequencies below 2 kHz. For MIDI note numbers 84 to 95, the coefficients in the frequency range 1 kHz to 3 kHz were selected, and for MIDI note numbers 95 to 107, the frequency coefficients from the range 2 kHz to 4 kHz were used as the feature vector.

4.2 Piano Note Classification

The piano transcription system is composed of 87 OVA binary note classifiers that detect the presence of a given note in a frame of audio, where each frame is represented by a 256-element feature vector as described in Section 4.1. The distance-to-classifier-boundary hyperplane margins were treated as a proxy for a log-posterior probability for each of the classes. In order to classify the presence of a note within a frame, we assume the state to be solely dependent on the normalized frequency data. At this stage, we further assume each frame to be independent of all other frames.

The SVMs were trained using Sequential Minimal Optimization [52], as implemented in the Weka toolkit [78]. A Radial Basis Function (RBF) kernel was selected for the experiments, and the γ and C parameters were optimized over a global grid search on the validation set using a subset of the training set. In the experiments described in this section, all of the classifiers were trained using the 92 MIDI training files and classification accuracy is reported on the validation set.

In the first classification experiment, we sought to determine the number of training instances to include from each audio excerpt. The number of training excerpts was held constant, and the number of training instances selected

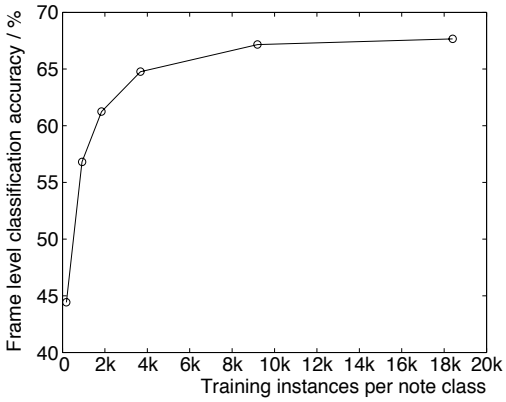

Figure 4.2: Variation of classification accuracy with number of randomly selected training frames per note, per excerpt.

from each piece was varied by randomly sampling an equal number of positive and negative instances for each note. As displayed in Figure 4.2, the classification accuracy[1] begins to approach an asymptote within a small fraction of the potential training data. The observation that random sampling approaches an asymptote within a couple of hundred samples per excerpt (out of a total of 6000 for a 60 s excerpt with 10 ms hops) is consistent with and reinforces the results reported in Section 3.3.1 in which we noticed many clusters of largely redundant training frames due to musicological and signal processing considerations. Since the RBF kernel requires training time on the order of the number of training instances cubed, 100 samples per note class, per excerpt was selected as a compromise between training time and performance for the remainder of the experiments. As noted in Section 4.1, there are an average of 8 note events per second in the training data; thus, random sampling down to 2% (roughly equal to the median prior probability of a specific note occurrence) is a reasonable approximation.

In the second classification experiment, we examined the incremental gain in classification accuracy from adding novel training excerpts. In this case, the number of training excerpts was varied while holding the number of training instances per excerpt constant. Figure 4.3 shows the variation in classification accuracy with the addition of novel training excerpts. Each additional excerpt corresponded to 100 randomly-selected frames per note class (50 each positive and negative instances). Thus, the largest note classifiers were trained on 9200 frames. The solid curve displays the result of training on the same number of frames randomly drawn from the pool of the entire training set. In contrast to the effects of adding additional training excerpts described in Section 3.3 and illustrated in Figure 3.2, the two curves in Figure 4.3 are very closely related. This result appears to be a combined artifact of the limited timbral variance in MIDI piano synthesis and the fact that the piano notes are

[1] A detailed description of the classification metrics is provided in Section 4.4

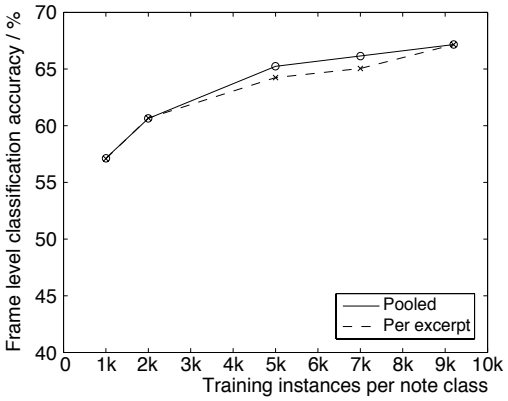

Figure 4.3: Variation of classification accuracy with the total number of excerpts included, compared to sampling the same total number of frames from all excerpts pooled.

discretized to specific fundamental frequencies (whereas instruments such as the human voice may vary over a bounded range). In Section 4.4.2, we begin to investigate the effect of training data generalization, a concept that becomes the focus of Chapter 5.

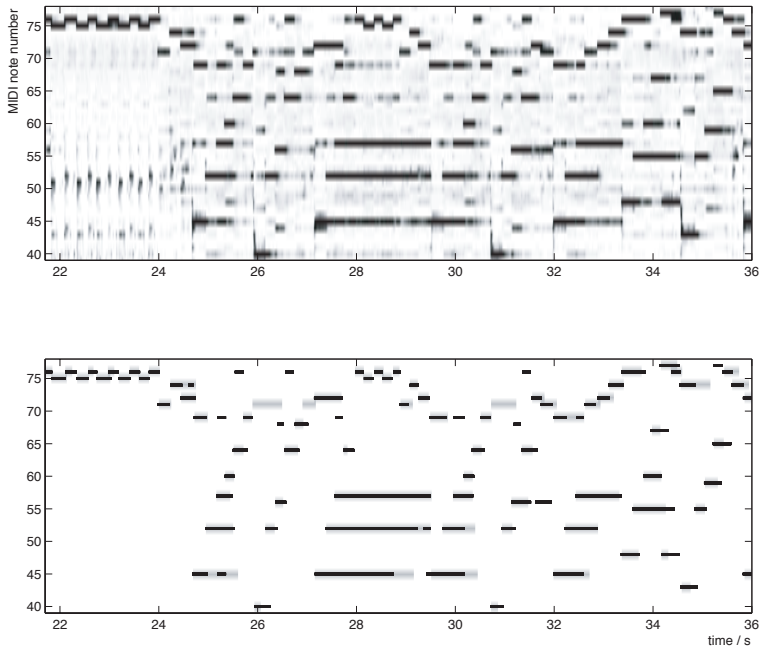

Figure 4.4: *Top*: Posteriorgram (pitch probabilities as a function of time) for an excerpt of Beethoven's Für Elise. *Bottom*: The HMM smoothed estimation (black) plotted on top of the ground truth labels (gray).

4.3 Hidden Markov Model Post Processing

Similarly to the procedure described in Section 3.5.3, we attempted to induce temporal constraints on the independent note classifications by implementing a hidden Markov model post processing stage. Each note class was independently modeled with a two state, 'on'/'off', HMM where the state dynamics (i.e. transition matrix and class priors) were estimated from the directly observed state sequences, the ground-truth transcriptions of the training set. Whereas the posterior-based melody transcription HMM consisted of identifying the single maximum likelihood sequence through the C allowed states where the observation $p(c_t|q_t)$ was represented by columns of the posteriorgram, the posterior-based polyphonic HMM consists of identifying the C maximum likelihood binary sequences for each of the C states where the observation $p(c_t|q_t)$ is represented by the cell of the posteriorgram (and its complement) corresponding to the particular class under consideration.

HMM post-processing results in a 2.8% absolute improvement thus yielding a frame-level classification accuracy of 70% on the validation set. Although the improvement in frame-level classification accuracy is relatively modest, the

HMM post-processing stage reduces the total onset transcription error by over 7%, primarily by alleviating spurious onsets. A representative posteriorgram and the result of the HMM post processing for an excerpt of "Für Elise" are displayed in Figure 4.4.

4.4 Experimental Results

In this section, we present a number of metrics to evaluate polyphonic music transcription algorithms and provide empirical comparisons to a number of alternative systems for piano transcription and multiple fundamental frequency estimation.

4.4.1 Evaluation Metrics

For each of the evaluated algorithms, a 10 ms frame-level comparison was made between the algorithm output and the ground-truth MIDI transcript. As such, the reference scores are represented by a binary "piano-roll" matrix that consists of one row for each note considered and one column for each 10 ms time step. A brief description of the evaluation metrics is provided below:

- The **overall accuracy** is a frame-level interpretation of the metric proposed in [19] and is defined as

$$Acc = \frac{TP}{(FP + FN + TP)} \tag{4.1}$$

  where TP ("true positives") is the number of correctly transcribed voiced frames, FP ("false positives") is the number of unvoiced note-frames transcribed as voiced, and FN ("false negatives") is the number of voiced note-frames transcribed as unvoiced. Overall accuracy is bounded by 0 and 1, with 1 corresponding to perfect transcription. This measure does not, however, facilitate an insight into the trade-off between notes that are missed and notes that are inserted.

- The frame-level **transcription error score** is a metric based on the "speaker diarization error score" defined by NIST for evaluations of 'who spoke when' in recorded meetings [48]. A meeting may involve many people, who, like notes on a piano, are often silent but sometimes simultaneously active (i.e. speaking). NIST developed a metric that consists of a single error score which further breaks down into substitution errors (mislabeling an active voice), "miss" errors (when a voice is truly active but results in no transcript), and "false alarm" errors (when an active voice is reported without any underlying source). This three-way decomposition avoids the problem of 'double-counting' errors where a

note is transcribed at the right time but with the wrong pitch; a simple error metric as used in earlier work, and implicit in **Acc**, biases systems towards not reporting notes, since not detecting a note counts as a single error (a "miss"), but reporting an incorrect pitch counts as two errors (a "miss" *plus* a "false alarm"). Instead, at every time frame, the intersection of N_{sys} reported pitches and N_{ref} ground-truth pitches counts as the number of correct pitches N_{corr}; the total error score, integrated across all time frames t is then:

$$E_{tot} = \frac{\sum_{t=1}^{T} max\big(N_{ref}(t), N_{sys}(t)\big) - N_{corr}(t)}{\sum_{t=1}^{T} N_{ref}(t)} \tag{4.2}$$

which is normalized by the total number of active note-frames in the ground-truth, so that reporting no output will entail an error score of 1.0. Frame-level transcription error is the sum of three components: substitution, "miss", and "false alarm" errors.

- **Substitution errors** are defined as:

$$E_{subs} = \frac{\sum_{t=1}^{T} min\big(N_{ref}(t), N_{sys}(t)\big) - N_{corr}(t)}{\sum_{t=1}^{T} N_{ref}(t)} \tag{4.3}$$

  which counts, at each time frame, the number of ground-truth notes for which the correct transcription was not reported, yet *some* note was reported – which can thus be considered a substitution. It is not necessary to designate *which* incorrect notes are substitutions, merely to count how many there are.

- **Miss errors** are defined as:

$$E_{miss} = \frac{\sum_{t=1}^{T} max\big(0, N_{ref}(t) - N_{sys}(t)\big)}{\sum_{t=1}^{T} N_{ref}(t)} \tag{4.4}$$

- **False alarm errors** are defined as:

$$E_{fa} = \frac{\sum_{t=1}^{T} max\big(0, N_{sys}(t) - N_{ref}(t)\big)}{\sum_{t=1}^{T} N_{ref}(t)} \tag{4.5}$$

The error equations sum, at the frame level, the number of ground-truth reference notes that could not be matched with any system outputs (i.e. misses after substitutions are accounted for) or system outputs that cannot be paired with any ground truth (false alarms beyond substitutions) respectively.

The error measure is a score rather than a probability or proportion i.e. it can exceed 100% if the number of insertions (false alarms) is very high. In line with the universal practice in the speech recognition community, we feel this is the most useful measure since it gives a direct feel for the quantity of errors that will occur as a proportion of the total quantity of notes present.

It aids intuition to have the errors break down into separate, commensurate components that add up to the total error, expressing the proportion of errors falling into the distinct categories of substitutions, misses, and false alarms.

In addition to the frame-level error metrics described above, the MIREX 2007 evaluation described in Section 4.4.3 reported note-level results using a number of common signal detection metrics as well as a measure of the total similarity to the reference score.

- The **precision** or positive predictive value is defined as:

$$P = \frac{TP}{TP + FP} \tag{4.6}$$

  which measures the proportion of assigned labels that were correct.

- The **recall** is defined as:

$$R = \frac{TP}{TP + FN} \tag{4.7}$$

  which measures the proportion correct labels that were assigned.

- The $\mathbf{F_1}$ **measure** combines precision and recall with equal weighting:

$$F_1 = \frac{2PR}{P + R} \tag{4.8}$$

  in order to balance the bias towards omitting note label estimates in order to maximize precision and including spurious note label estimates in order to maximize recall.

- The **overlap ratio** as proposed in [62] is defined as:

$$\text{overlap ratio} = \frac{\min\{\text{offsets}\} - \max\{\text{onsets}\}}{\max\{\text{offsets}\} - \min\{\text{onsets}\}} \tag{4.9}$$

  where "onsets" refers to the onset times of both the reference and estimated note, and "offsets" refers to the onsets of the reference and estimated note.

For the note-level analysis described in Section 4.4.3, an estimated note is scored as correct if the onset is within 50 ms of the reference onset, the fundamental frequency is within a quarter tone of the reference fundamental, and the offset value is within the larger of 20% of the duration of the note or 50 ms.

4.4.2 Piano Transcription

The classification-based piano transcription system was used to estimate the musical score for the 35 (25 synthesized and 10 recorded) songs in the testing

Algorithm	Acc	E_{tot}	E_{subs}	E_{miss}	E_{fa}
SVM [54]	65.9%	34.2%	5.3%	12.1%	16.8%
Ryynänen and Klapuri [62]	46.3%	52.3%	15.0%	26.2%	11.1%
Marolt [43]	36.9%	65.7%	19.3%	30.9%	15.4%

Table 4.1: Frame-level piano transcription results.

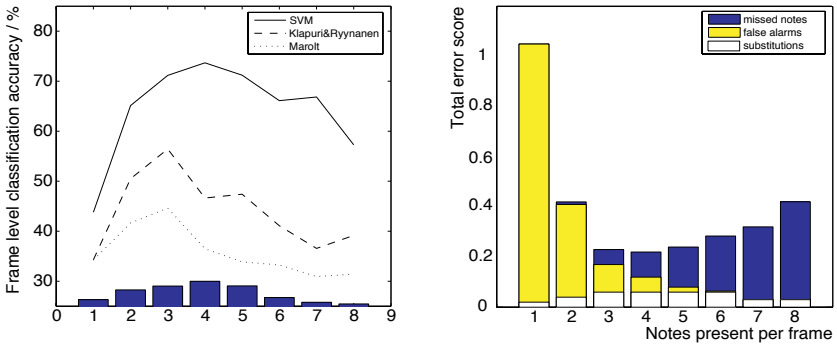

Figure 4.5: *Left*:Variation of classification accuracy with number of notes present in a given frame and relative note frequency. *Right*: Error score composition as a function of the number of notes present.

set, and the results of the frame-level evaluation are displayed in Table 4.1. In addition, direct comparisons are provided to the systems proposed in [62] and [43]. We note that the Ryynänen and Klapuri system was developed for general music transcription, and the parameters have not been tuned specifically for piano recordings.

As displayed in Table 4.1, the classification system provides a significant performance advantage on the test set with respect to frame-level accuracy and the error scores – outperforming the other two systems on 33 out of the 35 test pieces. Since the transcription problem becomes more complex as the number of simultaneous notes increases, we have also plotted the frame-level classification accuracy versus the number of notes present for each of the algorithms in the left panel of Figure 4.5; the total error score (broken down into the three components) with the number of simultaneously occurring notes for the proposed algorithm is displayed in the right panel. As expected, there is an inverse relationship between the number of notes present and the proportional contribution of false alarm errors to the total error score. However, the performance degradation is not as severe for the proposed method as it is for the comparison systems.

In Table 4.2, a breakdown of the transcription results is reported between the synthesized audio and piano recordings. The proposed system exhibits the most significant disparity in performance between the synthesized audio and piano recordings; however, we suspect that this is because the greatest

Algorithm	**Piano** (10)	**Synthesized** (25)	**Both** (35)
SVM (Piano Only)	59.2%	23.2%	33.5%
SVM (MIDI Only)	33.0%	74.6%	62.7%
SVM (MIDI & Piano)	56.5%	70.0%	65.9%
Ryynänen and Klapuri	42.2%	47.9%	46.3%
Marolt	35.4%	37.5%	36.9%

Table 4.2: Classification accuracy comparison for the MIDI test files and live recordings. The MIDI SVM classifier was trained on the 92 MIDI training excerpts, and the Piano SVM classifier was trained on the 20 piano recordings.

proportion of the training data was generated using synthesized audio. In addition, Table 4.2 displays the classification accuracy results for SVMs trained on MIDI data and piano recordings alone. The specific data distributions perform well on more similar data, but generalize poorly to unfamiliar audio. These results clearly indicate that the implementations based only on one type of training data are over-trained to the specific timbral characteristics of that data and may provide an explanation for the poor performance of neural network-based system. However, the inclusion of both types of training data does not come at a significant cost to classification accuracy for either type. As such, it is likely that the proposed system may be generalized to different types of piano recordings when trained on a diverse set of training instances.

In order to investigate the generalization assumption further, the proposed system was used to transcribe the test set prepared by Marolt in [43]. This set consists of six recordings from the same piano and recording conditions used to train his neural network and is different from any of the data in our training set. The classification results on the Marolt test set are displayed in Table 4.3. The SVM system commits a greater number of substitution and miss errors compared to its performance on the relevant portion of our test set, reinforcing the possibility of improving the stability and robustness of the SVM with a broader training set. Marolt's classifier, trained on data closer to his test set than to ours, outperforms the SVM here on the overall accuracy metric, although interestingly with a much greater number of false alarms than the SVM (compensated for by many fewer misses). The system proposed by Ryynänen and Klapuri outperforms the classification-based approaches on the Marolt test set; a result that underscores the need for a diverse set of training recordings for a practical implementation of a classification-based approach.

Frame-level accuracy is a particularly exacting metric. Although offset estimation is essential in generating accurate transcriptions, it is likely of lesser perceptual importance than accurate onset detection. In addition, the problem of offset detection is obscured by relative energy decay and pedaling effects. In order to account for these effects and to reduce the influence of note duration on the performance results, we report an evaluation of note

Algorithm / test set	Acc	E_{tot}	E_{subs}	E_{miss}	E_{fa}
SVM / our piano	56.5%	46.7%	10.2%	15.9%	20.5%
SVM / Marolt piano	44.6%	60.1%	14.4%	25.5%	20.1%
Marolt / Marolt piano	46.4%	66.1%	15.8%	13.2%	37.1%
Ryynänen and Klapuri / Marolt piano	50.4%	52.2%	12.8%	21.1%	18.3%

Table 4.3: Frame-level transcription results on recorded piano only (ours and the Marolt test sets).

Algorithm	Acc	E_{tot}	E_{subs}	E_{miss}	E_{fa}
SVM	69.1%	43.2%	4.5%	16.4%	22.4%
Ryynänen and Klapuri	60.2%	46.0%	6.2%	25.3%	14.4%
Marolt	33.6%	87.5%	13.9%	41.9%	31.7%

Table 4.4: Piano note onset detection results.

onset detection. A note onset was labeled as correct if the estimated onset was within 100 ms of the ground-truth onset. The systems were scored on the metrics described above with respect to note onsets rather than frame-level transcription accuracy, and the note onset transcription results are reported in Table 4.4. When scoring the systems, substitutions were counted first by associating unattached system outputs and ground-truth notes. Even without a formal onset detection stage, the proposed algorithm provides a slight advantage over the comparison systems on the 35 song test set.

4.4.3 Multiple Fundamental Frequency Estimation

In 2007, the music information retrieval community conducted a multiple fundamental frequency estimation evaluation as part of MIREX 2007[2]. Like the melody transcription evaluations that preceded it, the goal of the multiple fundamental frequency evaluation was to provide a unified set of test data and evaluation metrics for the community of researchers working on automated transcription. For the frame-level analysis, the 28 song test set against which the algorithms were evaluated consisted of 20 ensemble pieces (with various combinations of woodwind, brass, and string instruments) and eight synthesized recordings from the real world computing (RWC) database [34]. The ground-truth transcriptions for the 20 ensemble pieces were generated using a procedure similar to that described in Section 3.1.1. The results of the frame-level multiple fundamental frequency evaluation are presented in Table 4.6.

[2] http://www.music-ir.org/mirex/2007/index.php/Multiple_Fundamental_Frequency_Estimation_&_Tracking_Results

In this experiment, the piano specific classification system was generalized in order to perform instrument-independent music transcription. The classification-based submission to the multiple fundamental frequency evaluation was trained on the pop-based melody transcription training data[3] described in Section 3.1, the piano transcription training data described in Section 4.1, and 20 additional synthesized MIDI files with varying instrumentation. In order to maximize efficiency in classification time, linear kernel SVM classifiers were trained for each of the C classes corresponding to the first 87 piano notes. A two state HMM, as described in Section 4.3, was used to temporally constrain the classification posteriors based on the state dynamics observed in the training data. Despite the fact that the data used to train the classification system bore little resemblance to the 20 ensemble pieces, the SVM-based approach to music transcription performed relatively well as compared to the other submissions finishing 6th out of 12 research groups on the total frame-level error score.

In addition to the frame-level evaluation, an analysis of note-level transcription systems was performed on a set of 30 recordings. The note-level test set consisted of 16 ensemble recordings, eight synthesized recordings from the RWC database, and six piano recordings selected from the test set described in Section 4.1. Seven of the participants attempted to formulate the fundamental frequency estimates into notes, and the results of the note-level transcription analysis are displayed in Table 4.7. The note-level classifiers, which discretize the fundamental frequency estimates to the nearest semitone, performed near the top of the evaluation as a result of operating on the note level of abstraction. The proposed system likely received an advantage on the six piano test cases that were created in the same recording environment as a fraction of the training data; however, the SVM approach was able to generalize to a number of unseen instruments and recording environments represented in the remaining testing cases.

[3]Only positive training instances were selected from the melody transcription training data since we could not exclude the possibility of a positive accompanying note instance for any given training frames.

Composer	Training	Testing	Validation
Albéniz	España (Prélude†, Malagueña, Sereneta, Zortzico) Suite Española (Granada, Cataluña, Sevilla, Cádiz, Aragon, Castilla)	España (Tango), Suite Española (Cuba)	España (Capricho Catalan)
Bach	Well-Tempered Clavier 1: Prelude & Fugue No. 5	Well-Tempered Clavier 1: Prelude & Fugue No. 2	Well-Tempered Clavier 1: Prelude & Fugue No. 1
Balakirew	Islamej†		
Beethoven	Appassionata 1-3, Moonlight (1, 3), Pathetique (1)†, Waldstein (1-3)	Für Elise† Moonlight (2), Pathetique (3)†	Pathetique (2)
Borodin	Petite Suite (In the monastery†, Intermezzo, Mazurka, Serenade, Nocturne)	Petite Suite (Mazurka)	Réverie
Brahms	Fantasia (2†, 5), Rhapsodie	Fantasia (6)†	
Burgmueller	The pearls†, Thunderstorm	The Fountain	
Chopin	Opus 7 (1†, 2), Opus 25 (4), Opus 28 (2, 6, 10, 22), Opus 33 (2, 4)	Opus 10 (1)†, Opus 28 (13)	Opus 28 (3)
Debussy	Suite bergamasque (Passepied†, Prélude)	Menuet	Clair de Lune
Granados	Danzas Españolas (Oriental†, Zarabanda)	Danzas Españolas (Villanesca)	
Grieg	Opus 12 (3), Opus 43 (4), Opus 71 (3)†	Opus 65 (Wedding)	Opus 54 (3)
Haydn	Sonata in G major 1†	Sonata in G major 2 †	
Liszt	Grandes Etudes de Paganini (1†-5)	Love Dreams (3)	Grandes Etudes de Paganini (6)
Mendelssohn	Opus 30 (1)†, Opus 62 (3,4)	Opus 62 (5)	Opus 53 (5)
Mozart	Sonata in C Major (1†-3), Sonata in B Flat Major (3)	Sonata in B Flat Major (1)†	Sonata in B Flat Major (2)
Mussorgsky	Pictures at an Exhibition (1†,3,5-8)	Pictures at an Exhibition (2,4)	
Schubert	Sonata in A Minor (1†,2), Fantasy in C Major (1-3), Sonata in B Flat (1,3)	Fantasy in C Major (4)†	Sonata in B Flat (2)
Schumann	Scenes from Childhood (1-3, 5, 6†)	Scenes from Childhood (4) †	Opus 1 (1)
Tchaikovsky	The Seasons (February, March, April†, May, August September, October, November, December)	The Seasons (January†, June)	The Seasons (July)

Table 4.5: MIDI compositions from http://www.piano-midi.de/. † denotes songs for which piano recordings were made.

Algorithm	E_{tot}	E_{subs}	E_{miss}	E_{fa}	Acc
Pertusa [51]	44.5%	9.4%	29.8%	5.3%	58.0%
Yeh [79]	46.0%	10.8%	23.8%	11.5%	58.9%
Ryynänen [62]	47.4%	15.8%	13.3%	18.3%	60.5%
Zhou [80]	49.8%	14.1%	19.7%	16.0%	58.2%
Vincent [75]	53.8%	13.5%	24.0%	16.3%	54.3%
Poliner	63.9%	12.0%	37.5%	14.4%	44.4%
Leveau [42]	63.9%	15.1%	43.2%	5.5%	39.4%
Raczyński [59]	67.0%	18.5%	21.9%	26.5%	48.4%
Cao [7]	68.5%	20.0%	12.8%	35.6%	51.0%
Emiya [28]	95.7%	7.0%	76.7%	12.0%	14.5%
Cont [12]	99.0%	34.8%	22.1%	42.1%	31.1%
Egashira [37]	118.8%	40.1%	5.2%	73.4%	33.6%

Table 4.6: MIREX 2007 frame-level multiple fundamental frequency evaluation results. For brevity, systems are referred to by their first authors alone.

Algorithm	Precision	Recall	F_1	Overlap
Ryynänen	0.312	0.382	0.337	0.884
Poliner	0.305	0.278	0.277	0.879
Pertusa	0.206	0.262	0.226	0.844
Vincent	0.162	0.277	0.204	0.859
Egashira	0.071	0.130	0.09	0.847
Emiya	0.098	0.052	0.076	0.804
Cont	0.015	0.044	0.023	0.831

Table 4.7: MIREX 2007 note-level multiple fundamental frequency evaluation results.

4.5 Summary

In this chapter, we presented a classification approach to music transcription. We have shown that a data driven approach may be used to classify the notes of a specific instrument or generalized to an instrument agnostic transcription framework. We observed that the relevance of the training data had the single greatest impact on classification accuracy; however, representative training data is often limited. As such, we seek to investigate methods for improving classification generalization and extending a limited training set in the following chapter.

Chapter 5

Improving Generalization for Classification-Based Transcription

In this chapter, we present methods to improve the generalization capabilities of the classification-based approach to music transcription. Although the system proposed in Chapter 4 compared favorably with model-based approaches when both the training and testing recordings were made from the same set of pianos, the classification-based system exhibited a performance degradation when presented with piano recordings made under different conditions. As described in the preceding chapters, classifier performance is limited by the amount and diversity of the labeled training data available; however, a great deal of relevant, yet unlabeled, audio data exists. In this chapter, we seek to exploit the vast pool of unlabeled data and to improve the value of the limited labeled data. To that end, we investigate semi-supervised learning and multiconditioning techniques for improving generalization.

5.1 Audio Data

The 92 synthesized MIDI files and 20 piano recordings described in Section 4.1 were used as the labeled training data in the generalization experiments. The validation set used to tune classifier parameters was collected from the RWC database [34], and the ground-truth transcripts for the three validation files were aligned by Cont [11] following the method described in [38]. The generalization test set consisted of 19 piano recordings made from three different pianos including six pieces from the test set generated by Marolt [43], two pieces created by Scheirer [64], and 11 pieces recorded on a Roland HP

330e digital piano downloaded from the Classical Piano Midi Page[1]. In cases where limitations in the MIDI file parsing resulted in a linear scaling between the labels and test audio, a compensating scaling constant was estimated to maximize the alignment between the reference score and a noisy transcription of the audio made by the baseline SVM system. In addition to the labeled audio, 54 unlabeled polyphonic piano files were collected from 20 different recording environments to be used as training data in the semi-supervised learning experiments.

5.2 Generalized Learning

Although the classification-based system performs well on different recordings made from the same set of pianos in the same recording environments, the success of the transcription system does not translate as well to novel pianos and unseen recording settings. In this section, we propose methods for improving generalization by learning from unlabeled training data and by augmenting the value of the data for which training labels are available.

5.2.1 Semi-Supervised Learning

Millions of music recordings exist, yet only a very small fraction of them are labeled with corresponding transcriptions. Since the success of the proposed transcription system is so heavily dependent on the quantity and diversity of the available training data, we have attempted to incorporate more of the data available to train new classification systems by applying different techniques to assign labels to unlabeled data.

Nearest neighbor clustering is a simple classification system in which a label is assigned to a particular point in the feature space based on its proximity, using a given distance metric, to its k-nearest neighbors. For each frame-level feature vector[2] in the unlabeled data set, a set of 87 binary labels was generated by calculating the Euclidian distance to each point in the training data for a given note class and assigning the label of the (majority vote of the) k-nearest neighbors to the unlabeled point. For each note, an equal number of positive and negative training instances generated from the unlabeled data was added to the original training data set, and a new system of SVM classifiers was trained.

In our semi-supervised SVM approach, labels were assigned to unlabeled data by classifying the unlabeled points with our baseline SVM system. Although formal semi-supervised support vector machine methods have been proposed for minimizing the structural risk by calculating the misclassification error for each unlabeled feature vector (i.e. minimizing the additional

[1] http://www.piano-midi.de/

[2] The acoustic features were calculated following the procedure described in Section 4.1.2.

risk of adding the data point as a positive or negative training instance) [3], classifying the unlabeled data using the framework described in Chapter 4 enables the incorporation of temporal constraints. As an alternative to using the raw classifier output as a proxy for training sample selection, the HMM post-processing stage described in Section 4.3 may be applied to the output of the unlabeled data classification. In some cases, the inclusion of the HMM stage results in class assignment updates due to temporal context, thus improving the insight of the trained classifier in ambiguous situations. Again, for each note, an equal number of positive and negative training instances generated from the unlabeled data was added to the original training set in order to create an updated system of classifiers.

5.2.2 Multiconditioning

The quantity and diversity of the training data was extended by resampling the audio to effect a global pitch shift. Each recording from the training set was resampled at rates corresponding to frequency shifts of a fraction of a semitone in order to account for differences in piano tunings. The corresponding ground-truth labels were unaffected (since the target note class remained the same); however, the time axis was linearly interpolated in order to adjust for the resulting time scaling. Symmetrically shifted frequency data was added to the original training set in order to create additional classifiers. As such, this method for extending the training data corresponds to a sub-semitone version of the resampling approach described in Section 3.1.

5.3 Experiments

In the first semi-supervised learning experiment, each frame of audio in the unlabeled data set was assigned the label of its k-nearest neighbors. From each song in the unlabeled set and for each note in the classification system, 50 negative training instances and 50 positive training instances (when available) were added to the original set of training data This addition increased the quantity of training data by approximately 50%. The amount of training data used was held constant while the number of nearest neighbors, k, was varied from 1 to 7 in odd increments. A classification system of SVMs was trained from each of the updated training sets; however, each resulted in a negligible change in transcription error on the validation set.

The baseline SVM system was then used to estimate transcriptions for each song in the unlabeled data set. Positive training instances were selected by varying the range of the distance to classifier boundary used for sampling selection. While holding the 50% increase in training data constant, we attempted sampling from a series of ranges by performing a grid search over the distance to classifier boundary, the best of which resulted in a 0.8 point decrease in total error score for the validation set. In addition to sampling

different distance to classifier ranges to generate training instances, the HMM post-processing stage was applied to the raw classifier transcriptions of the unlabeled data set. From each song, 50 positive and negative instances were selected for each note class and additional classifiers were trained resulting in an 1.1 point reduction in the total error on the validation set. In order to demonstrate the variation in classifier performance due to the addition of semi-supervised training instances, the amount of estimated training data was varied as a fraction of percent increase in total data from 10-100% (in 10% increments) resulting in a monotonically decreasing reduction in the total error score on the validation set up to 1.9 points for the training instances generated from the output of the SVM classifier with HMM smoothing.

Four additional classifiers were trained in order to investigate the effects of generating training data from resampled audio. Each recording from the training set was resampled at symmetric rates corresponding to $\pm$ 0.5, 1.0, 1.5, 2.0% deviations from the original tone (where a full semitone shift corresponds to a $\approx$ 6% deviation). In this experiment, the amount of resampled training data was held constant, while the range of resampled audio used to train the classifiers was varied. Incorporating the resampled audio resulted in 3.1, 1.2, 1.1, and 0.9 point reductions, respectively, in frame-level error score for the validation set. We suspect that the resampling rates closer to the original tone provide an advantage in performance because they are more likely to be in line with mild instrument de-tuning. The top performing resampled classifier was then used to generate labels for the unlabeled data set. The transcriptions were temporally smoothed via the HMM, and the estimated labels were sampled (50 positive and negative instances per class) to create additional training data for a final set of classifiers. The combination of the semi-supervised learning with the resampling technique resulted in a 4.7 point improvement in total error score on the validation set.

The parameters for each of the generalization techniques were optimized on the validation set by minimizing the frame-level transcription error score, and the top performing classification system from each of the proposed frameworks was used to transcribe the 19 songs in the test set. The corresponding frame-level transcription results are displayed in Table 5.1. The top performing system, a combination of the semi-supervised and multiconditioning techniques, provided a 10 point reduction in total frame-level error score on the test set.

Finally, both the baseline system and the system combining training data from multiconditioning and semi-supervised learning were used to transcribe the 10 test piano recordings described in Section 4.1. Including the diversifying training data resulted in a mild 0.4 point performance degradation in total error score for the original instruments; however, the 10 point improvement in generalization on the novel test set seems to warrant the addition.

	Frame-level transcription			
System	$\mathbf{E_{tot}}$	$\mathbf{E_{subs}}$	$\mathbf{E_{miss}}$	$\mathbf{E_{fa}}$
SVM (baseline)	69.7%	15.8%	36.3%	17.6%
k-NN	70.5%	15.1%	37.3%	18.1%
SVM	68.9%	10.2%	49.7%	9.0%
SVM + HMM	68.5%	15.6%	33.9%	19.0%
MC	63.0%	12.4%	39.5%	11.1%
MC + SVM + HMM	59.1%	8.6%	38.6%	12.3%

Table 5.1: Generalization experiment transcription error results for the 19 song test set. The systems reported correspond to methods for creating additional training data.

5.4 Summary

In this chapter, we presented a number of methods for improving classification generalization for piano transcription. We have shown that a reduction in total transcription error may be achieved by combining multiconditioning and semi-supervised learning in order to generate additional training data for a classification-based music transcription system. The proposed methods demonstrate that limited quantities of training data may be augmented in order to reduce classification error. In the following chapter, we investigate these concepts further by examining the effects of using classification posteriors as alignment features in order to facilitate bootstrap learning for developing training data and improving transcription accuracy.

Chapter 6

Score to Audio Alignment

In this chapter, we present a method for score to audio alignment based on synchronizing an estimated transcript with a reference score. The framework described in the preceding chapters is used to transcribe polyphonic audio recordings, and the classification posteriors are synchronized to a MIDI transcript by dynamic time warping. A key advantage of the proposed method for score to audio alignment is that the time-alignment is performed in the score domain and, as such, does not require an artificial synthesis of the score. We describe a novel method for generating test cases based on leveraging a small amount of aligned data by systematically distorting the reference score, and we report empirical comparisons to a number of alternative alignment approaches. Finally, we present a keystone experiment in which the proposed method for score to audio alignment is used to develop a semi-supervised classification system for music transcription.

6.1 Audio Data and Features

6.1.1 Audio Data

The typical method for evaluating score to audio alignment approaches involves hand-labeling the time mapping between a recording and the corresponding score for a small test set. In the typical case, the audio recording is considered a distorted version of the target reference score. Rather than limiting our evaluation to a small set of hand-labeled test files, we examined the converse approach of generating a reference recording directly from a transcript, then distorting the reference transcript by a known warping to create misaligned pairs of scores and recordings. In this complementary case, the recordings (and the transcripts used to generate them) are considered the reference, and we attempt to map the distorted scores onto the time axis of the reference recordings. This alternative method allows us to employ a larger

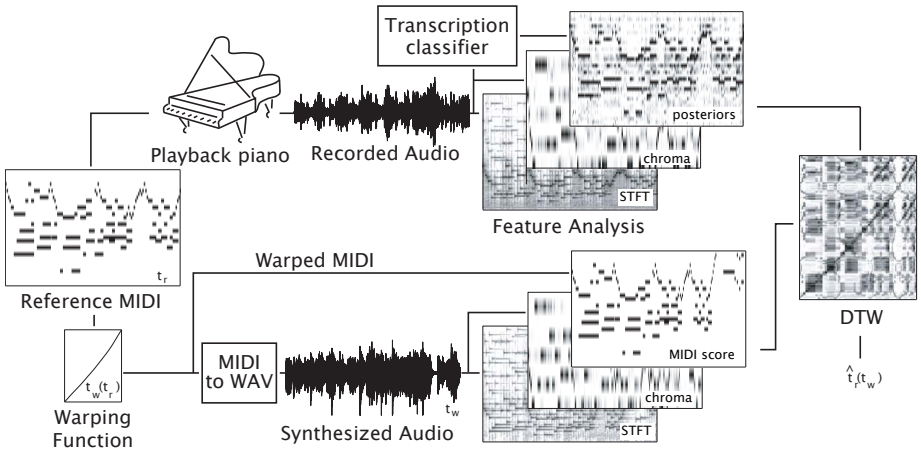

Figure 6.1: Flow chart of the data generation and feature analysis for score to audio alignment.

test set (potentially including a large number of distortions) without requiring a large number of recordings, since each recording is aligned to multiple, distorted, scores. Consequently, the statistical quality of our analyses are improved, and the time required for test set development is significantly reduced.

The experimental audio data was derived from the 10 piano recording/MIDI transcript pairs described in Section 4.1. We attempted to account for typical deviations that might be expected between scores and recordings when generating the test cases. Therefore, the reference MIDI files were distorted by varying the local tempo and by modifying the notes present in the reference transcript. A complete description of the experimental distortions is provided in Section 6.3. Figure 6.1 displays the development of the test data and time-warping analysis.

In addition to the set of piano recordings and distorted MIDI files, we compiled a small set of operetta songs from the 1993 D'Oyly Carte recording of Gilbert and Sullivan's *Mikado* and the corresponding MIDI karaoke files from The Gilbert and Sullivan Archive, `http://math.boisestate.edu/gas/`. The beginning of each line of lyrics was hand labeled for the recordings and read directly from the karaoke file transcripts.

6.1.2 Short-Time Fourier Transform

The magnitude short-time Fourier transform was used as our baseline alignment feature. The piano recordings and synthesized MIDI files were resampled to 8 kHz, and the STFT was applied to the audio files using 1024 point discrete Fourier transforms (i.e. 128 ms), a 1024 point Hanning window, and an 80 point advance between adjacent windows (for a 10 ms hop between

successive frames). The frequency coefficients below 2 kHz (i.e. the first 256 spectral bins) were used as the features in the calculation of the similarity matrix as described in Section 6.2.

6.1.3 Classification Posteriors – Transcription Estimate

The posterior features were calculated using the piano transcription system described in Chapter 4 (without the HMM post-processing stage) trained on the generalized data described in Chapter 5. As in Chapter 5, linear kernel SVMs were used in order to maximize classification efficiency. The system of SVM classifiers was used to detect the presence or absence of each note in a frame of audio, a process that resulted in an estimated transcript for each recording. The transcript estimations were limited to the first 63 piano notes (i.e. notes with a fundamental frequency less than 2 kHz) when calculating the similarity between the warped MIDI transcript and the estimated reference transcript.

6.1.4 Peak Structure Distance

Like us, the authors of [49] wished to avoid having to employ an explicit synthesized audio version of their score to achieve alignment. Their solution was to define a specialized similarity measure, the Peak Structure Distance (PSD). For a given set of notes from the score, PSD hypothesizes the locations of associated harmonics in the spectrum (taking for example the first 8 multiples of the expected fundamentals), then calculates the similarity of the observed spectral frames to the set of notes as the proportion of the total spectral energy that occurs within some narrow window around the predicted harmonics. As the actual spectrum tends towards pure sets of harmonics at the correct frequencies, the similarity tends to 1. This is then converted to a distance by subtracting the similarity metric from 1. Thus, the measure neatly avoids having to model the relative energies at each harmonic.

6.1.5 Chroma

Chroma features attempt to capture the dominant note as well as the broad harmonic accompaniment by folding all spectral information into the 12 semitones within an octave. Rather than using a coarse mapping of FFT bins to the chroma classes they overlap (which is particularly blurry at low frequencies), the phase-derivative (instantaneous frequency) within each FFT bin was used both to identify strong tonal components in the spectrum (indicated by spectrally-adjacent bins with close instantaneous frequencies) and to improve the resolution of the underlying frequency estimate [9, 1]. The chroma was generated on a 10 ms grid from the spectral components below 1 kHz.

In addition to calculating the chroma features from the synthesized MIDI files, we attempted to estimate chroma directly from the MIDI transcript. Reference power curves, created by varying the 'velocity' of individual piano notes and measuring the power versus time, were used to approximate the relative loudness and decay of each note. In order to calculate the chroma features directly from the notes in the MIDI file, the corresponding reference power was added to the feature matrix rather than using a binary representation of each note in the transcript.

6.2 Time Alignment

6.2.1 Similarity Matrix

In order to calculate the similarity between two feature matrices, we took the normalized inner product or cosine distance. The similarity matrix M is calculated by:

$$M = \frac{A^T B}{E_A^T E_B} \tag{6.1}$$

where A is the feature matrix for the reference audio recording (one column per time step) and B is the feature matrix for the time-warped recording in our analysis. E_A and E_B are row vectors containing the norms of each column of A and B respectively, and the division is applied elementwise. The elements of the resulting matrix M are bounded by 0 and 1, where a cell value of 1 indicates a region of high similarity between the cells of the feature matrices.

6.2.2 Dynamic Time Warping

Dynamic time warping was used to identify the least-cost time-alignment path through each similarity matrix. The least-cost path p is computed iteratively via dynamic programming by minimizing the cost function for the distance matrix, $D = 1 - M$:

$$p(i,j) = \min \begin{cases} p(i-1, j-1) + D(i,j) \\ p(i-1, j) + D(i,j) \\ p(i, j-1) + D(i,j) \end{cases} \tag{6.2}$$

An example similarity matrix and the corresponding least-cost path is displayed in Figure 6.2. A complete treatment of DTW for audio alignment is available in [58].

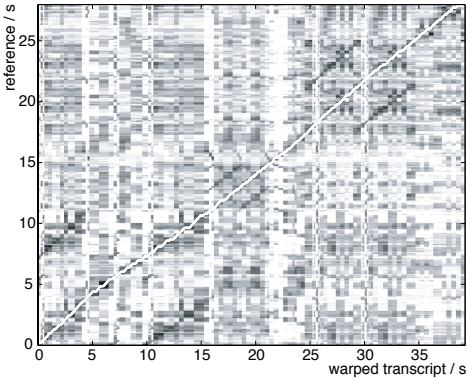

Figure 6.2: Similarity matrix calculated between an audio recording and a MIDI rendition of Tchaikovsky's January: At the Fireside from *The Seasons*. Dark regions indicate similarity between the feature matrices, and the least-cost path warping estimate is overlaid in white.

6.3 Alignment Experiments

6.3.1 Evaluation Metric

We assess the success of each of the feature representations considered by evaluating the timing error between the reference transcript and the estimated alignment. We report the average onset timing error which is defined as the mean of the timing error, $err_k = |t_k^{ref} - t_k^{est}|$, for each note onset k in the reference transcript. This metric is similar to the "average offset" reported in [49] and the "error" score described in [13].

6.3.2 Time Distortion

The tempo of the reference transcript was varied in order to account for local deviations (e.g. stylistic performance differences) and global shifts (e.g. playing the song more slowly). A Brownian walk $W(t)$ was applied to the reference tempo in order to generate the warped time t_w as a function of reference time t_r

$$W(t) = W(t-1) + N(0, \sigma^2) \tag{6.3}$$

$$t_w(t_r) = \sum_{t=0}^{t_r} \exp\left\{ B \cdot log(2) \cdot \frac{W(t)}{\max_\tau |W(\tau)|} \right\} \tag{6.4}$$

where B is a warp bound varied between 0.1 to 1. Thus, the warped time t_w is obtained as the sum of a sequence of time steps that can individually vary between 0.5 and 2.0 for the largest warp bound. 10 random walk iterations were performed for each warp bound resulting in 1000 time-warped MIDI and au-

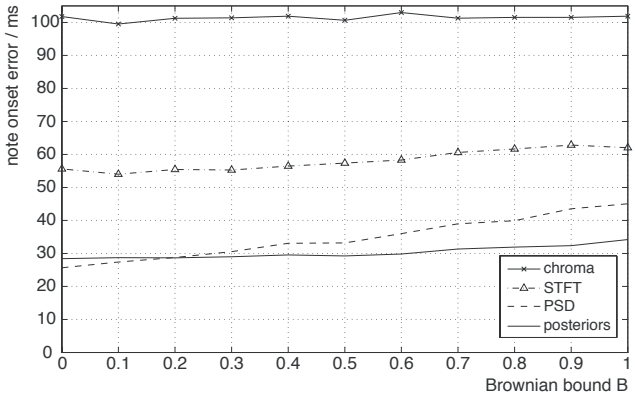

Figure 6.3: Time warp mean onset errors.

dio recording pairs. The resulting mean onset error evaluated on a 10 ms grid for each of the features considered is displayed in Figure 6.3. Although not displayed in Figure 6.3, the chroma features generated directly from the MIDI transcript result in an approximate 40 ms performance degradation as compared to the chroma calculated from the synthesized MIDI. Chroma features give relatively poor temporal alignment compared to STFT[1], but PSD and posterior features are significantly better, with posteriors achieving a slight edge for more drastic distortions.

Each of the comparison systems operates by making some kind of prediction of the spectrum from the score, then comparing with the actual signal in the spectral domain. The key advantage of using a classifier to generate score-like note posteriors appears to lie in its ability to generalize across all the different spectral realizations that a particular pitch may take. Synthetic audio derived from MIDI makes a single guess about the anticipated spectra of each note, and to the extent this fails to match the actual notes observed, the similarity matrix is compromised and alignment will suffer. The classifier, by contrast, has been trained on multiple instances of each note's spectra, and will map any of these versions to the same transcribed event. This result is most clearly illustrated in the substantial improvement between alignment based on STFT features and that using posteriors. The PSD, which also is able to accept a note regardless of its precise harmonic spectrum, performed comparably to posterior matching in our tests; however, posterior feature matching was superior in most cases.

[1] A single song containing a number of arpeggios contributed disproportionately to the total error for the Chroma features. This test case highlights the conspicuous weakness of folding the observations into a single octave representation; however, the chroma features provide a theoretical advantage when the recorded audio is played in a different octave than the score.

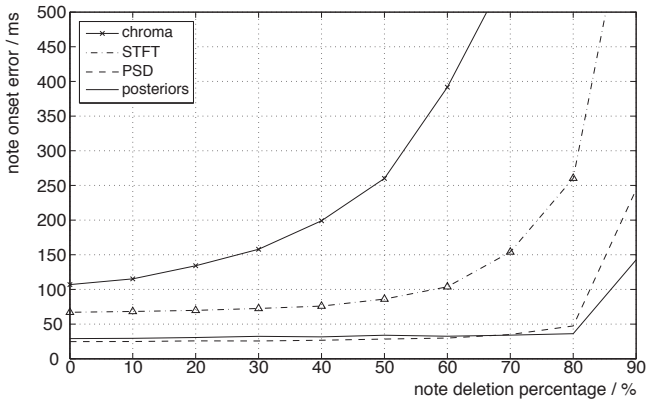

Figure 6.4: Note deletion mean onset errors.

6.3.3 Note Deletion

A portion of the notes present in the reference transcript were removed in order to account for variations in performance (e.g. misplayed or spurious notes). Uniform random sampling was used to delete a fraction of the notes while holding the original tempo of the reference transcript constant. For each fraction of notes deleted, 10 files were created resulting in 900 additional test cases. The results of the mean note onset error calculation for varying note deletion percentages are displayed in Figure 6.4.

Both the PSD and posterior features appear to gain an advantage from the dynamic programming bias towards jointly advancing a step in each time axis. As such, the note deletion experiment was repeated for those two features with a constant, linear time-scaling in the distorted transcripts while holding the specific notes deleted fixed. The test cases were scaled such that the distorted transcripts were 50% longer in duration, and the results of the supplemental note deletion experiment are displayed in Figure 6.5. The posterior features still resulted in note onset errors of approximately 100 ms or less when up to 50% of the notes were deleted from the linearly scaled transcript.

6.3.4 Variation in Instrumentation

In order to account for variations in instrumentation and synthesis quality between a MIDI transcript and an audio recording, a set of eight operetta recordings, for which the symphonic MIDI representation differed from the recorded audio, were hand labeled at the beginning of each line of lyrics. These instants were selected since hand-labeling every note onset is impractical for an opera recording and because the practical task of voice detection facilitates vocal performance analysis such as an examination of vowel-

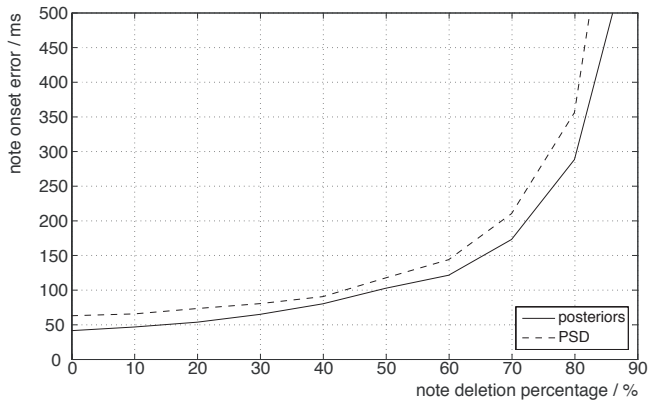

Figure 6.5: Note deletion mean onset errors for the linearly scaled test cases.

Feature	$\bar{err}$ (ms)	$\bar{err}$ (frames)
STFT	324	1.62
Posteriors	**316**	**1.58**
PSD	322	1.61
Chroma	375	1.87

Table 6.1: Error results for the hand-labeled opera recordings.

modification. The start times for each line of lyrics were labeled on a 100 ms grid, but the error evaluation between the recorded audio and karaoke transcript was performed on a 200 ms grid in order to account for inaccuracies in the hand-labeling. The mean onset error results for the hand-labeled data set are displayed in Table 6.1. In this experiment, the classification-based features were generated using the instrument agnostic classification system described in Section 4.4.3.

We suspect that the hop size used in the evaluation may be a limiting factor in the mean note onset errors (especially for the opera experiments). However, the average time alignment errors were limited to a small number of frames in the feature representation for the majority of cases.

6.4 Bootstrap Learning

The proposed method for score to audio alignment was used to synchronize 46 MIDI transcripts from the data set described in Section 3.1 to piano recordings made in a number of different recording environments (i.e. different studio, piano, performer, etc.) for which an exact transcript was unavailable. The note labels from the time-aligned score were associated with frames from

	Frame-level transcription			
System	$\mathbf{E_{tot}}$	$\mathbf{E_{subs}}$	$\mathbf{E_{miss}}$	$\mathbf{E_{fa}}$
SVM (baseline)	69.7%	15.8%	36.3%	17.6%
MC + SVM + HMM	59.1%	8.6%	38.6%	12.3%
MC + SVM + HMM + Bootstrap	58.1%	11.7%	31.8%	14.6%
Bootstrap	64.8%	12.8%	35.8%	16.2%
Bootstrap + MC	64.3%	16.6%	30.3%	22.4%

Table 6.2: Bootstrap generalization experiment transcription error results for the 19 song test set described in Section 5.1. The first two rows of the table have been repeated from Table 5.1.

the additional audio recordings and used to train a piano transcription classification system based on all of the available piano training data[2]. As displayed in Table 6.2, the final piano transcription system resulted in a 1.0 point reduction in total error score on the test set described in Section 5.1 as compared with the best performing system from Chapter 5.

Comparing the baseline system with the systems reported in the last two rows of Table 6.2 has more dramatic implications. Training on the bootstrap data alone resulted in a 4.9 point reduction in total error score as compared to the baseline system (despite learning from $\approx 1/3$ the amount of data which was unlabeled to begin with), a result that highlights both the merit of the bootstrapping method for generating labeled training data and the importance of relevant training data for successful music transcription. Incorporating a more diverse training set has a similar effect to the proposed methods for generalization, and as such, increasing the quantity of training data with multiconditioned bootstrap data results in a modest 0.5 point reduction in total error score. However, the systems based on bootstrap learning are significantly inferior to the system that incorporates all available training data. We speculate that this results stems from the vast difference in training data volume.

6.5 Summary

In this chapter, we described a method for score to audio alignment based on classification posterior features. The posterior features were used in a dynamic time warping framework to align MIDI transcripts to recorded audio under a number of distortion conditions, and the proposed features appear to provide a modest improvement in mean onset error for larger deviations between the transcript and a recorded performance. The score to audio alignment system was used to synchronize MIDI transcripts to unlabeled audio

[2]The training data for the final classification system consisted of the time-aligned 'bootstrap' recordings, the MIDI syntheses, the playback piano recordings, the semi-supervised training data, and the multiconditioned piano recordings.

recordings in order to facilitate bootstrap learning, a process that resulted in an improvement in classification generalization.

Chapter 7

Conclusion

We have presented a classification-based approach for automatic music transcription. The proposed system of support vector machine note classifiers temporally constrained via hidden Markov models may be cast as a general transcription framework, trained specifically for a particular instrument, or used to recognize higher-level musical concepts such as melodic sequences. Although the classification structure provides a simple and competitive alternative to model-based systems, perhaps the most important result of this thesis is that no formal acoustical prior knowledge is required in order to perform music transcription.

Music transcription has a number of practical applications in content-based retrieval/organization, signal transformation, and as a pedagogical tool. For examples, the transcription system described may be used to identify multiple performances of the same piece from within a music database, synthesize a recording with different instrumentation, or analyze a performance to identify stylistic interpretations or variations. In addition, the estimated transcripts may be used as acoustic features in order to solve related music information retrieval problems (e.g. score to audio alignment).

As with any artificial intelligence system, one may begin to wonder what characteristics the classifiers are learning. Although the mechanics of the classifiers are rather opaque (i.e. a SVM formulates classification decisions on input feature vectors in a black box framework), we may speculate about the underlying nature of the system by examining the common transcription errors. Figure 7.1 displays the log-probability of a note occurrence during an insertion error for the piano transcription validation set. Insertion errors commonly occur when (sub)octave and harmonically related notes are present[1]. As such, it appears that the classifier is learning empirical models of harmonic structure; however, the framework allows for the potential of a more generalized form since the classifier may learn to map many harmonic series

[1]Similar error patterns have been observed for note omission errors.

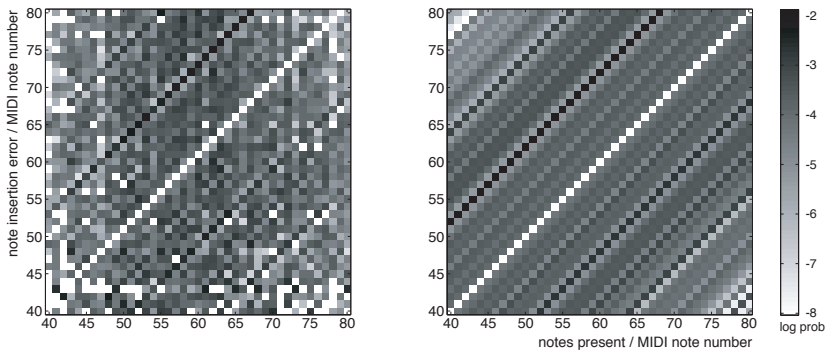

Figure 7.1: *Left*: Log-probability note accompaniment occurrence for note insertion errors. *Right*: Log-probability note accompaniment insertion errors normalized by averaging along the diagonals of the matrix in order to facilitate generalization.

weightings to a single note class (as long as representative patterns exist in the training data).

An examination of the transcription errors leads us to consider the upper bound of the proposed system. An attractive feature of the data driven approach is that improvements in classification accuracy may be obtained by adding novel, relevant training data as illustrated in Figures 3.3 and 4.3. We expect to observe these incremental gains in classification accuracy until the subtle nuances of the testing distribution are represented in the training data. Unfortunately, labeled training data is often difficult or expensive to obtain. As such, we were motivated to extend the value of the limited data through semi-supervised learning, multiconditioning, and bootstrapping, and subsequent improvements were achieved. Furthermore, dependency on representative data is also the key limitation of the proposed approach, and the classification system was observed to perform poorly in cases where exemplary data was unavailable. However, the training requirements for characteristic data were generally modest.

As with any classification approach, the proposed framework assumes a predefined system dependency. In the reported structure, we regard a note class as a conventional western tonal note that bears a resemblance to the data represented in the training set. As such, the described implementation cannot perform sub-note-level (e.g. quarter-tone) pitch estimation without training additional classifiers and may have difficulty resolving notes played in different contexts (e.g. unseen simultaneous note combinations). However, a related advantage of the proposed system is the potential to exploit higher-level concepts such as commonly related notes when performing transcription.

Although a strong dependency on training data was observed, there are a number of directions to pursue with respect to the classification framework. We recognize that separating the classification and temporal constraints is

somewhat *ad hoc*. A system for applying maximum-margin classification in a Markov framework was proposed in [71]; however, solving the entire optimization problem may be impractical for the scope of the evaluation. Moreover, treating each frame independently does not come at a significant cost to classification accuracy. Perhaps the existing SVM framework may be improved by optimizing the discriminant function for detection rather than maximum-margin classification as described in [65].

Despite the fact that a variety of acoustic features were implemented, none of the representations provided a significant advantage in terms of classification accuracy. This result is note entirely surprising since all of the features contain largely equivalent information; however, it may be that a better normalization scheme remains to be discovered. For example, a log-frequency spectral representation may allow for the incorporation of higher frequency harmonic information without an excessively large feature vector.

Figures 3.6 and 7.1 illustrate common note errors for the proposed melody transcription system and piano transcription system respectively. Many of the errors may be categorized as adjacent note confusion or octave transpositions. More advanced training sample selection methods such as disproportionately sampling members of the same chroma class, harmonically related notes, or adjacent note boundaries (i.e. note classes with the greatest probability of classification error) may result in improved transcription results.

Provided the appropriate training data is available, the proposed framework may be extended to perform a number of related tasks such as chord transcription and local key estimation. Independent estimates of the key and chord may be combined in a hierarchical model to improve the temporal context of the transcriptions. Similarly, the inclusion of a formal onset detection stage may reduce note detection errors occurring at rearticulations and provide a perceptual advantage for common onset clustering.

In addition to framework enhancements to the classification approach, there are several research directions that may benefit from investigating the use of the proposed system. We observed that posterior features provided an advantage for score to audio alignment. Classification posteriors could also serve as a replacement for chroma or spectral features for similar tasks such as cover song detection [27] and phrase segmentation [14]. The classification posteriors may even be used as features for another classification system (e.g. key or chord), similarly to the method proposed in [77] for speech/non-speech discrimination.

Finally, the proposed system may be used to learn the structure of musical composition. For example, the classification system may be trained on the accompaniment data *without* the lead melody mixed in, but still using the melody transcripts as the target labels. The resulting classification system would be trained to predict an 'appropriate' melody from an accompaniment alone. With suitable temporal smoothness constraints, as well as perhaps some random perturbation to avoid boring melody choices, the proposed framework could be used as a robotic improvisor or compositional aid.

(This page intentionally left blank)

Bibliography

[1] T. Abe and M. Honda. Sinusoidal model based on instantaneous frequency attractors. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1292–1300, 2006. (Cited on page 61.)

[2] J. Bello, L. Daudet, and M. Sandler. Automatic piano transcription using frequency and time-domain information. *IEEE transactions on audio, speech, and language processing*, 14(6):2242–2251, November 2006. (Cited on page 6.)

[3] K. Bennett and A. Demiriz. Semi-supervised support vector machines. In *Proc. Advances in Neural Information Processing Systems*, pages 368–374, Denver, December 1998. (Cited on page 55.)

[4] P. Brossier, J. Bello, and M. Plumbley. Fast labeling of notes in music signals. In *Proc. International Conference on Music Information Retrieval*, pages 331–336, Barcelona, October 2004. (Cited on page 34.)

[5] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, June 1998. (Cited on page 22.)

[6] P. Cano. Fundamental frequency estimation in the SMS analysis. In *Proc. COST-G6 Workshop on Digital Audio Effects*, Barcelona, November 1998. (Cited on page 19.)

[7] C. Cao, M. Li, J. Liu, and Y. Yan. Multiple fo estimation in polyphonic music (MIREX 2007). In *Proc. MIREX Multiple Fundamental Frequency Estimation Abstracts*, Vienna, September 2007. (Cited on page 51.)

[8] A. Cemgil, H. Kappen, and D. Barber. A generative model for music transcription. *IEEE transactions on audio, speech, and language processing*, 14(2):679–694, March 2006. (Cited on page 6.)

[9] F. Charpentier. Pitch detection using the short-term phase spectrum. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 113–116, Tokyo, April 1986. (Cited on page 61.)

[10] N. Chawla, N. Japkowicz, and A. Kolcz. Editorial: Special issue on learning from imbalanced data sets. *ACM Special Interest Group on Knowledge Discovery and Data Mining Explorations*, 6(1):1–6, June 2004. (Cited on page 28.)

[11] A. Cont. Realtime multiple pitch observation using sparse non-negative constraints. In *International Conference on Music Information Retrieval*, Victoria, October 2006. (Cited on page 53.)

[12] A. Cont, S. Dubnov, and D. Wessel. Realtime multiple-pitch and multiple-instrument recognition for music signals using sparse non-negative constraints. In *Proc. International Conference on Digital Audio Effects*, Bordeaux, October 2007. (Cited on page 51.)

[13] A. Cont, D. Schwarz, N. Schnell, and C. Raphael. Evaluation of real-time audio-to-score alignment. In *Proc. International Conference on Music Information Retrieval*, pages 315–316, Vienna, September 2007. (Cited on page 63.)

[14] M. Cooper and J. Foote. Summarizing popular music via structural similarity analysis. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 127–130, New Paltz, October 2003. (Cited on page 71.)

[15] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, November 1995. (Cited on page 22.)

[16] N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, New York, 2000. (Cited on page 22.)

[17] R. Dannenberg. An on-line algorithm for real-time accompaniment. In *Proc. International Computer Music Conference*, pages 193–198, Paris, October 1984. (Cited on page 12.)

[18] A. de Cheveigne and H. Kawahara. YIN, a fundamental frequency estimator for speech and music. *Journal Acoustic Society of America*, 111(4):1917–1930, 2002. (Cited on pages 10 and 16.)

[19] S. Dixon. On the computer recognition of solo piano music. In *Proc. Australasian Computer Music Conference*, Brisbane, July 2000. (Cited on page 43.)

[20] S. Dixon and G. Widmer. Match: A music alignment tool chest. In *Proc. International Conference on Music Information Retrieval*, London, September 2005. (Cited on page 13.)

[21] K. Dressler. Sinusoidal extraction using an efficient implementation of a multi-resolution fft. In *Proc. International Conference on Digital Audio Effects*, pages 247–252, Montreal, September 2006. (Cited on pages 9 and 34.)

[22] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley & Sons, New York, 2001. (Cited on page 34.)

[23] H. Duifhuis, L. Willems, and R. Sluyter. Measurement of pitch in speech: An implementation of Goldstein's theory of pitch perception. *Journal of the Acoustical Society of America*, 71(6):1568–1580, 1982. (Cited on page 7.)

[24] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Cambridge University Press*. Prentice Hall, Cambridge, 1998. (Cited on page 12.)

[25] D. Ellis. *Prediction-driven computational auditory scene analysis*. PhD thesis, Massachusetts Institute of Technology, Cambridge, 1996. (Cited on pages 6 and 7.)

[26] D. Ellis and G. Poliner. Classification-based melody transcription. *Machine Learning Journal*, 65(2–3):439–456, 2006. (Cited on pages 3 and 34.)

[27] D. Ellis and G. Poliner. Identifying cover songs with chroma features and dynamic programming beat tracking. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages IV 1429–1432, Honolulu, April 2007. (Cited on page 71.)

[28] V. Emiya, R. Badeau, and B. David. Multipitch estimation and tracking of inharmonic sounds in colored noise. In *Proc. MIREX Multiple Fundamental Frequency Estimation Abstracts*, Vienna, September 2007. (Cited on page 51.)

[29] J. L. Flanagan and R. M. Golden. Phase vocoder. *Bell System Technical Journal*, 45:1493–1509, 1966. (Cited on page 9.)

[30] A. Ghias, J. Logan, D. Chamberlin, and B. Smith. Query by humming – Music information retrieval in multimedia databases. In *Proc. ACM Multimedia*, pages 231–236, San Francisco, 1995. (Cited on page 9.)

[31] J. Goldstein. An optimum processor for the central formation of pitch of complex tones. *Journal of the Acoustical Society of America*, 54:1496–1516, 1973. (Cited on page 7.)

[32] E. Gomez, S. Streich, B. Ong, R. Paiva, S. Tappert, J. Batke, G. Poliner, D. Ellis, and J. Bello. A quantitative comparison of different approaches for melody extraction from polyphonic audio recordings. Technical Report MTG-TR- 2006-01, University Pompeu Fabra, Music Technology Group, 2006. (Cited on page 18.)

[33] M. Goto. A real-time music scene description system: Predominant-Fo estimation for detecting melody and bass lines in real-world audio signals. *Speech Communication*, 43(4):311–329, 2004. (Cited on pages 7, 9, and 34.)

[34] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. Rwc music database: Music genre database and musical instrument sound database. In *Proc. International Conference on Music Information Retrieval*, pages 229–230, Baltimore, October 2003. (Cited on pages 48 and 53.)

[35] M. Goto and S. Hayamizu. A real-time music scene description system: Detecting melody and bass lines in audio signals. In *Working Notes of the IJCAI-99 Workshop on Computational Auditory Scene Analysis*, pages 31–40, Stockholm, August 1999. (Cited on page 9.)

[36] N. Hu, R. Dannenberg, and G. Tzanetakis. Polyphonic audio matching and alignment for music retrieval. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 185–188, New Paltz, October 2003. (Cited on pages 12 and 13.)

[37] H. Kameoka, T. Nishimoto, and S. Sagayama. A multipitch analyzer based on harmonic temporal structured clustering. *IEEE transactions on audio, speech, and language processing*, 15(3):982–994, 2007. (Cited on pages 6 and 51.)

[38] H. Kaprykowsky and Xavier Rodet. Globally optimal short-time dynamic time warping application to score to audio alignment. In *Proc. IEEE International Conference on Audio Speech and Signal Processing*, Toulouse, May 2006. (Cited on pages 13 and 53.)

[39] A. Klapuri. A perceptually motivated multiple-fo estimation method. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 291–294, New Paltz, October 2005. (Cited on page 7.)

[40] A. Klapuri and M. Davy, editors. *Signal Processing Methods for Music Transcription*. Springer, New York, 2006. (Cited on page 5.)

[41] D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, October 1999. (Cited on page 7.)

[42] P. Leveau, D. Sodoyer, and L. Daudet. Automatic instrument recognition in a polyphonic mixture using sparse representations. In *Proc. International Conference on Music Information Retrieval*, Victoria, October 2006. (Cited on page 51.)

[43] M. Marolt. A connectionist approach to transcription of polyphonic piano music. *IEEE Transactions on Multimedia*, 6(3):439–449, October 2004. (Cited on pages 6, 46, 47, and 53.)

[44] M. Marolt. On finding melodic lines in audio recordings. In *Proc. International Conference on Digital Audio Effects*, 2004. (Cited on pages 9 and 34.)

[45] K. Martin. A blackboard system for automatic transcription of simple polyphonic music. Technical Report 385, MIT Media Laboratory Perceptual Computing Section, 1996. (Cited on page 6.)

[46] M.Davy and S. J. Godsill. Bayesian harmonic models for musical signal analysis (with discussion). In J. Bernardo, J. Berger, A. Dawid, and A. Smith, editors, *Bayesian Statistics VII*. Oxford University Press, 2003. (Cited on page 6.)

[47] J.A. Moorer. On the transcription of musical sound by computer. *Computer Music Journal*, 1(4):32–38, 1977. (Cited on page 5.)

[48] National Institute of Standards and Technology. Spring 2004 (RT-04S) rich transcription meeting recognition evaluation plan, 2004. (Cited on page 43.)

[49] N. Orio and D. Schwarz. Alignment of monophonic and polyphonic music to a score. In *Proc. International Computer Music Conference*, Havana, September 2001. (Cited on pages 12, 13, 61, and 63.)

[50] R. P. Paiva, T. Mendes, and A. Cardosa. Melody detection in polyphonic musical signals: Exploiting perceptual rules, note salience and melodic smoothness. *Computer Music Journal*, 30(4):80–98, 2006. (Cited on pages 9 and 34.)

[51] A. Pertusa and J. Iñesta. Multiple fundamental frequency estimation based on spectral pattern loudness and smoothness. In *Proc. MIREX Multiple Fundamental Frequency Estimation Abstracts*, Vienna, September 2007. (Cited on page 51.)

[52] J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Scholkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 185–208. MIT Press, Cambridge, 1999. (Cited on pages 20 and 39.)

[53] G. Poliner and D. Ellis. A classification approach to melody transcription. In *Proc. International Conference on Music Information Retrieval*, pages 161–166, London, September 2005. (Cited on page 3.)

[54] G. Poliner and D. Ellis. A discriminative model for polyphonic piano transcription. *EURASIP Journal on Advances in Signal Processing*, 2007, January 2007. (Cited on pages 3 and 46.)

[55] G. Poliner and D. Ellis. Improving generalization for classification-based polyphonic piano transcription. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 86–89, New Paltz, October 2007. (Cited on page 3.)

[56] G. Poliner, D. Ellis, A. Ehmann, E. Gomez, S. Strich, and B. Ong. Melody transcription from music audio: approaches and evaluation. *IEEE transactions on audio, speech, and language processing*, 15(4):1247–1256, May 2007. (Cited on pages 3 and 18.)

[57] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989. (Cited on page 28.)

[58] L. Rabiner and B. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, Englewood Cliffs, 1993. (Cited on pages 12, 13, and 62.)

[59] S. Raczyński, N. Ono, and S. Sagayama. MIREX 2007: Multiple fundamental frequency estimation and tracking harmonic nonnegative matrix approximation approach. In *Proc. MIREX Multiple Fundamental Frequency Estimation Abstracts*, Vienna, September 2007. (Cited on page 51.)

[60] C. Raphael. Automatic segmentation of acoustic musical signals using hidden markov models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):360–370, 1999. (Cited on page 12.)

[61] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004. (Cited on page 25.)

[62] M. Ryynänen and A. Klapuri. Polyphonic music transcription using note event modeling. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 319–322, New Paltz, October 2005. (Cited on pages 6, 45, 46, and 51.)

[63] M. Ryynänen and A. Klapuri. Transcription of the singing melody in polyphonic music. In *Proc. International Conference on Music Information Retrieval*, Victoria, October 2006. (Cited on pages 9 and 34.)

[64] E. Scheirer. *Music-Listening System*. PhD thesis, Massachusetts Institute of Technology, Cambridge, 2000. (Cited on page 53.)

[65] B. Schlköpf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001. (Cited on page 71.)

[66] K. Sjölander and J. Beskow. WaveSurfer - an open source speech tool. In *Proc. International Conference on Spoken Language Processing*, pages 464–467, Beijing, October 2000. (Cited on pages 16 and 19.)

[67] M. Slaney and R. F. Lyon. On the importance of time – A temporal representation of sound. In M. Cooke, S. Beet, and M. Crawford, editors, *Visual Representations of Speech Signals*. J. Wiley, 1993. (Cited on pages 6 and 9.)

[68] P. Smaragdis and J. Brown. Non-negative matrix factorization for polyphonic music transcription. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 177–180, New Paltz, October 2003. (Cited on page 6.)

[69] C. Sutton, E. Vincent, M. Plumbley, and J. Bello. Transcription of vocal melodies using voice characteristics and algorithm fusion. In *Proc. MIREX Audio Melody Extraction Contest Abstracts*, Victoria, October 2006. (Cited on page 34.)

[70] D. Talkin. A robust algorithm for pitch tracking (RAPT). In W. B. Kleijn and K. K. Paliwal, editors, *Speech Coding and Synthesis*, chapter 14, pages 495–518. Elsevier, Amsterdam, 1995. (Cited on page 16.)

[71] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *Proc. Neural Information Processing Systems*, Vancouver, December 2003. (Cited on page 71.)

[72] R. Turetsky and D. Ellis. Ground-truth transcriptions of real music from force-aligned midi syntheses. In *Proc. International Conference on Music Information Retrieval*, pages 135–141, Baltimore, October 2003. (Cited on pages 12, 13, and 16.)

[73] B. Vercoe. The synthetic performer in the context of live performance. In *Proc. International Computer Music Conference*, pages 199–200, Paris, October 1984. (Cited on page 12.)

[74] T. Viitaniemi, A. Klapuri, and A. Eronen. A probabilistic model for the transcription of single-voice melodies. In *Proc. Finnish Signal Processing Symposium*, pages 5963–5957, 2003. (Cited on page 11.)

[75] E. Vincent, N. Bertin, and R. Badeau. Two nonnegative matrix factorization methods for polyphonic pitch transcription. In *Proc. MIREX Multiple Fundamental Frequency Estimation Abstracts*, Vienna, September 2007. (Cited on page 51.)

[76] E. Vincent and M. Plumbley. Predominant-fo estimation using Bayesian harmonic waveform models. In *Proc. MIREX Audio Melody Extraction Contest Abstracts*, London, September 2005. (Cited on pages 9 and 34.)

[77] G. Williams and D. Ellis. Speech/music discrimination based on posterior probability features. In *Proc. European Conference on Speech Communication and Technology*, pages 687–690, Budapest, September 1999. (Cited on page 71.)

[78] I. Witten and E. Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco, 2000. (Cited on pages 20 and 39.)

[79] C. Yeh. Multiple fo estimation for (MIREX 2007). In *Proc. MIREX Multiple Fundamental Frequency Estimation Abstracts*, Vienna, September 2007. (Cited on page 51.)

[80] R. Zhou and M. Mattavelli. A new time-frequency representation for music signal analysis. In *Proc. International Conference on Information Sciences, Signal Processing, and its Applications*, Sharijah, February 2007. (Cited on page 51.)