Tiago Fernandes Tavares

# Computational Models for Rhythm and Applications on Human-Machine Interactions

# Modelos Computacionais para o Ritmo e Aplicações em Interatividade Homem-Máquina

Campinas
2013

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação

Tiago Fernandes Tavares

# Computational Models for Rhythm and Applications on Human-Machine Interactions

# Modelos Computacionais para o Ritmo e Aplicações em Interatividade Homem-Máquina

Doctorate thesis presented to the School of Electrical and Computer Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering. Concentration area: Computer Engineering. Tese de doutorado apresentada à Faculdade de

Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: Engenharia de Computação.

Orientador (Supervisor): Prof. Dr. Romis Ribeiro de Faissol Attux
Co-orientador (Co-supervisor): Dr. Jayme Garcia Arnal Barbedo

Este exemplar corresponde à versão final da tese defendida pelo aluno Tiago Fernandes Tavares, e orientada pelo Prof. Dr. Romis Ribeiro de Faissol Attux

Campinas
2013

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

T197c

Tavares, Tiago Fernandes, 1984-
Computational models for rhythm and applications on human-machine interactions / Tiago Fernandes Tavares. – Campinas, SP : [s.n.], 2013.

Orientador: Romis Ribeiro de Faissol Attux.
Coorientador: Jayme Garcia Arnal Barbedo.
Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Inteligência artificial. 2. Processamento digital de sinais. 3. Música eletrônica. 4. Transcrição músical. I. Attux, Romis Ribeiro de Faissol,1978-. II. Barbedo, Jayme Garcia Arnal. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** Modelos computacionais para o ritmo e aplicações em interatividade homem-máquina
**Palavras-chave em inglês:**
Artificial intelligence
Digital signal processing
Electronic music
Transcription of music
**Área de concentração:** Engenharia de Computação
**Titulação:** Doutor em Engenharia Elétrica
**Banca examinadora:**
Romis Ribeiro de Faissol Attux [Orientador]
Marcelo Gomes de Queiroz
Eduardo Néspoli
Leonardo Tomazeli Duarte
Rafael Santos Mendes
**Data de defesa:** 17-10-2013
**Programa de Pós-Graduação:** Engenharia Elétrica

## COMISSÃO JULGADORA - TESE DE DOUTORADO

**Candidato:** Tiago Fernandes Tavares

**Data da Defesa:** 17 de outubro de 2013

**Título da Tese:** "Computational Models for Rhythm and Applications on Human Machine Interactions"

Prof. Dr. Romis Ribeiro de Faissol Attux (Presidente): [illegible]

Prof. Dr. Marcelo Gomes de Queiroz: [illegible]

Prof. Dr. Eduardo Néspoli: [illegible]

Prof. Dr. Leonardo Tomazeli Duarte: [illegible]

Prof. Dr. Rafael Santos Mendes: [illegible]

# Abstract

This thesis describes investigations towards the use of computational models for rhythm in the context of human-machine interactions. I propose a model for semi-automatic musical transcription, a model for automatic musical transcription and two interfaces for musical expression, all of them based in concepts related to rhythm. This novel study highlights the importance of using domain-specific knowledge in music information retrieval systems, as it allows not only more accurate systems to be built, but also the development of new ways of musically interacting with computers.
Key-words: Artificial Intelligence. Digital Signal Processing. Electronic Music. Automatic Transcription of Music.

# Resumo

Esta tese descreve investigações sobre a aplicação de modelos computacionais para o ritmo na interatividade homem-computador. São propostos um modelo para transcrição musical semi-automática, um modelo para transcrição automática e duas interfaces para expressão musical, todos baseados em conceitos ligados a ritmo. Este estudo, inédito, evidencia a importância do uso de conhecimentos especialistas em sistemas de recuperação de informações musicais, uma vez que são possibilitados não somente sistemas mais eficazes como também novas maneiras de interagir musicalmente com o computador.
Palavras-chave: Inteligência Artificial. Processamento Digital de Sinais. Música Eletrônica. Transcrição Automática de Música.

# Contents

Dedico este trabalho a todos aqueles que já acordaram um dia com o único propósito de construir um mundo melhor para si mesmo e para as pessoas que amam.

# Agradecimentos

Agradeço,
Ao Amauri Lopes, ao Romis Attux, ao Jayme Barbedo e ao George Tzanetakis, pelo excelente trabalho de orientação e inspiração ao longo de todo o tempo que trabalhamos juntos;
À FEEC, à Unicamp e à Universidade de Victoria pelo apoio institucional;
À CAPES e ao CNPq pelo apoio financeiro;
Aos meus familiares, pelo apoio incondicional;
Aos meus amigos de infância, da faculdade, de Victoria e de outros lugares, porque sem eles eu não teria conseguido.

Chapter 1

# Introduction

Automatic music transcription (AMT) is a task that consists of detecting what musical notes must be played to describe a particular audio signal. It is assumed that a musical piece may be described by a set of notes, which can be played, leading to the realization of the piece.

Different description marks must be used for the adequate description of pieces in different styles. Traditional musical notes, for example, are a remarkable tool for this purpose, but are unable to describe many kinds of expression marks, such as micro-timing and contemporary techniques like digital audio processing. It is unlikely that a musical piece can be entirely described by any set of pre-defined parameters, because the musician's interpretation at the time of playing may contain subtle variations, causing great change in the overall feeling of the piece.

Most work in AMT focuses on finding musical notes, which are described by their *onset* and their *pitch*, ignoring expression marks such as dynamics or changes in timbre. Note pitches and onsets are useful information for documentation and for the development of content-based search and interaction tools. Nevertheless, state-of-the-art transcription systems still cannot reach near-100% accuracy even in this simple case [1].

Aiming at these applications, recent efforts on AMT have relied on innovative algorithms derived from Digital Signal Processing (DSP) and Machine Learning (ML), as will be discussed in detail in Chapter 2. These algorithms are able to learn decision rules from a training dataset. However, after a thorough review of the state of the art of transcription algorithms, it was observed that the long-term temporal organization of musical notes is, to a great extent, neglected.

This organization is clearly present in many important musical genres. For example, in several contemporary songs it is possible to find a similar verse-verse-chorus-verse-chorus-chorus form. In general, information about one of these sections may be used to obtain objective clues about the most likely contents of the other sections.

The instrumental parts within Western popular music, particularly in the song form, are even more remarkable cases. It is common to find musical structures such as verse, bridge and chorus, in which the same sequence of musical notes are explicitly repeated. Hence, the notes in a verse of a song are useful clues about the notes in the following verse and so on.

Another important aspect in the temporal organization of musical notes concerns beat and tempo. Musical events, especially in contemporary Western popular music, tend to be organized in a hierarchical and periodic fashion. This gives a feeling of pulse, which makes it possible to clap hands to the beat of a piece or to dance to a tune without knowing it beforehand.

The aforementioned characteristics imply that some clues about the musical notes within a piece may be obtained from the overall organization of the piece. It may be possible, for example, to forecast continuations for musical pieces, based solely on the previous notes. It is also possible to expect that musical notes are organized as to generate a feeling of pulse, which means some note onsets times are more likely than others.

The use of this kind of explicit rhythmic knowledge is novel in the context of musical transcription. This work consists of developing models that quantify how much a given excerpt may be explained by a particular hypothesis, and applying such models in the context of musical transcription. The following sections will present a brief description of them.

## 1.1 Forecasting musical notes

As discussed before, in some musical genres, the explicit repetition of sections is a common composition strategy. In such genres, like contemporary Western popular music, it is very likely that two sections of the same type (for example, two verses) will be played using the exact same notes. This characteristic may be exploited to produce an algorithm capable of forecasting musical notes.

The algorithm, thoroughly described in Chapter 3, is based on the idea that, if two (or more) sections of a piece are played likewise, then it is not necessary to re-write them all. Instead, it is possible to generate a set of notes that will be played each time this type of section is used. This allows forecasting notes each time a particular musical section is repeated.

Since there is no prior information about the beginning or ending of each musical section, it was necessary to employ an automatic algorithm for finding it. The algorithm proposed in this thesis implicitly finds musical organizations in many levels. Then, a musical note that is considered more adequate to continue a sequence is yielded.

The forecasting algorithm is shown to correctly predict the majority of the musical notes in a piece, particularly, as experiments show, when the bass in popular music is considered. Therefore, the algorithm may be applied on an interactive transcription tool, in which the user provides some notes and the system forecasts continuations. When applied to musical genres that rely on the explicit repetition of musical notes, such as Western popular music, this demands less effort from the user than if an automatic transcription system was applied and had its outcomes manually corrected.

The disadvantage of using such an algorithm is that user interaction is necessary, even if an imperfect transcription suffices. When dealing with search in big databases, for example, it is better to have imperfect descriptions for all elements in the database than having exact transcriptions for just a few elements. Because of that, another model was developed, aiming at the problem of completely automatic transcription of music.

## 1.2 Finding periodicity in musical note onsets

If a particular piece causes a feeling of pulse, it is reasonable to assume that note onsets will be temporally organized in a more or less periodic fashion. This is correlated to the feeling of pulse that some pieces, especially those composed specifically for dancing, trigger in human listeners. A literature review has shown that this characteristic, although present in many musical genres such as Classical or Popular, has been neglected in the context of automatic transcription.

The proposed solution also approaches a technical problem that is present in many state-of-the-art transcription systems: the user has to set a sensitivity parameter that controls the tradeoff between false positives and false negatives yielded by the system. This parameter is either set manually or trained from a labeled database, often leading to sub-optimal choices, due to differences in timbre that appear when different instruments (*e.g.*, two distinct pianos) are used.

The proposed solution [2] consists of choosing the sensitivity parameter that maximizes the periodicity of the yielded note sequence. This approach is shown to retrieve solutions that are more robust to timbre changes than if a pre-defined (fixed) threshold was used. Also, it is an unsupervised method, hence no training dataset is necessary.

## 1.3 Interfaces for Musical Expression

During the development of this thesis, it became clear that AMT is one among many forms of content-based human-computer interaction that can be provided by modelling musical rhythm. This kind of algorithm may also be used in live performances, providing new interfaces for musical expression. Two algorithms for this purpose were developed and are discussed in detail in Chapter 5.

The first algorithm [3] is capable of repeating loops in control data. This kind of data, derived from faders, knobs, accelerometers and other sensors, is often used for the control of digital effects or other electronic parameters. The proposed algorithm extends motion loops, allowing more complex control-based variations to be performed live and expanding the possibilities of musical creation.

The second algorithm [4] is capable of quickly learning rhythms tapped in a digital interface and continuing it indefinitely. The algorithm forecasts repetitive sequences, using a process similar to that aimed at interactive transcription, and feeds back the predicted outcomes, creating a loop. This allowed building a drum sequencing device that does not require a visual interface, thus providing musicians with an interface for drum sequencing that is closer to the physical act of playing drums.

## 1.4 Objectives

This section exposes the objectives of this thesis. They will be discussed again in Chapter 6.

1. To develop models for aspects of rhythm that are present in some specific musical genres;
2. To apply these models to computer-assisted musical transcription;
3. To explore applications for these models other than automatic transcription of music.

## 1.5 Contributions

This thesis presents many contributions to the state of the art of automatic music transcription, as well as to the field of interfaces for musical expression. However, none of the problems presented here is entirely solved: chapter 6 is entirely dedicated to discussions over the unsolved problems and perspectives for future work.

For the reader's convenience, the main contributions of this thesis are summarized in Table 1.1. Each of these contributions is discussed in depth in a specific chapter.

| Contribution | Chapter | Description |
|---|---|---|
| Survey on Automatic Music Transcription | 2 | A representative survey on the state of the art of AMT is presented. It is shown that using long-term characteristics has improved the results in transcription. |
| Interactive transcription of music | 3 | A method for interactive transcription of music is presented. |
| Using periodicity for improving AMT | 4 | The concept of periodic organization of musical events is directly applied to improve the robustness of musical transcription algorithms. |
| Interfaces for Musical Expression | 5 | Derived from the previously discussed contexts, new interfaces for musical expression are proposed, allowing novel interactions in musical performance. |
| Discussions | 6 | Discussion on what was learned from this thesis and what are the perspectives for future work. |

Table 1.1: Summary of the contributions of this thesis.

## 1.6 Published papers

All papers published during this doctorate are listed here.

1. TAVARES, T. F.; TZANETAKIS, G.; DRIESSEN, P. Factors in Factorization: Does better Audio Source Separation Imply Better Polyphonic Music Transcription? In: 2013 IEEE International Workshop on Multimedia Signal Processing, Pula, Sardinia, Italy, September 2013.

2. TAVARES, T. F.; BARBEDO, J. G. A. ; ATTUX, R.; LOPES, A. Survey on Automatic Transcription of Music. In: Journal of the Brazilian Computer Society. Online.

3. TAVARES, T. F.; MONTEIRO, A.; BARBEDO, J. G. A.; ATTUX, R.; MANZOLLI, J. Real-time event sequencing without a visual interface. In: Sound and Music Computing (SMC2013). Stockholm, Sweden, July 2013.

4. SCHAUB, S.; SIMURRA, I.; TAVARES, T. F. Mixing Symbolic And Audio Data In Computer Assisted Music Analysis: A Case Study From J. Harvey's Speakings (2008) For Orchestra And Live. In: Sound and Music Computing (SMC2013). Stockholm, Sweden, July 2013.

5. COLLARES, L.; TAVARES, T. F.; Feliciano, J.; Gao, S.; Tzanetakis, G.; Gooch, A. SoundAnchoring: Content-based Exploration of Music Collections with Anchored Self-Organized Maps. In: Sound and Music Computing (SMC2013). Stockholm, Sweden, July 2013.

6. TAVARES, T. F.; GODOY, A. Sonification of population behavior in Particle Swarm Optimization. 2013 Genetic and Evolutionary Computation Conference. Amsterdam, Netherlands. July 2013.

7. TAVARES, T. F.; BARBEDO, J. G. A. ; ATTUX, R.; LOPES, A. Unsupervised training of detection threshold for polyphonic musical note tracking based on event periodicity.38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP). Vancouver, British Columbia, Canada, May 2013.

8. TAVARES, T. F.; ODOWICHUK, G. ; ZEHTABI, S.; TZANETAKIS, G. Audio-visual vibraphone transcription in real time. 2012 IEEE International Workshop on Multimedia Signal Processing. Banff, Alberta, Canada, September 2012.

9. GODLOVITCH, D. ; TAVARES, T. F. ; TRAIL, S.; TZANETAKIS, G. Physical Modeling and Hybrid Synthesis for the Gyil African Xylophone. Sound and Music Computing (SMC 2012). Copenhagen, Denmark, July 2012.

10. TRAIL, S. ; DEAN, M. ; TAVARES, T. F. ; ODOWICHUK, G.; DRIESSEN, P. ; SCHLOSS, A. W.; TZANETAKIS, G. Non-invasive sensing and gesture control for pitched percussion hyper-instruments using the Kinect. New Interfaces for Musical Expression (NIME 2012). Ann Arbour, Michigan, U.S.A., May 2012.

11. TRAIL, S. ; TAVARES, T. F. ; GODLOVITCH, D. ; TZANETAKIS, G. Direct and surrogate sensing for the Gyil african xylophone. New Interfaces for Musical Expression (NIME 2012). Ann Arbour, Michigan, U.S.A., May 2012.

Chapter 2

# Automatic Music Transcription

This chapter brings a review of state-of-the-art methods for Automatic Music Transcription. It was planned to put the following chapters in a proper perspective. Most of this content was published in a review paper [1].

This chapter begins by describing basic concepts on psycho-acoustics, discussing how physical models and objective measures can be related to auditory sensations. Section 2.1 also concerns conventions used for depicting musical information in the context of automatic transcription.

Section 2.2 shows the latest techniques used for automatic music transcription. The section describes how digital signal processing (DSP) and machine learning (ML) techniques have been used for the purpose of automatic transcription, with emphasis not only on technical descriptions, but also on the inspiration behind these techniques.

Section 2.3 discusses how automatic music transcription systems are evaluated. This is an important methodological issue that allows the objective comparison of different proposals for AMT.

In Section 2.4, an analysis of the results reported in the literature show that considering other musically relevant features has provided a greater performance improvement than the use of different algorithms. This fact is an evidence that the basic assumption of this work is valid, justifying its investigation.

Section 2.5 concludes the chapter with a brief summary of the main discussions.

## 2.1 Signals, sensations and notation

An audio signal is a stimulus that can trigger auditory sensations in human beings. In general, an audio signal is assumed to be a variation in the air pressure $x(t)$ with frequency components in the hearing range, that is, 20 Hz to 20 kHz [5]. The characteristics of this sound pressure variation may be controlled by a musician, either using his own voice and body or interacting with musical instruments using proper gestures. When the sound pressure variation is a harmonic signal, that is, $x(t)$ is the sum of $M$ sinusoidal components whose frequencies are multiples $mF$ of a fundamental frequency (F0) $F$, and phases $\phi_m$ and amplitudes $A_m$ are arbitrary, as in

$$x(t) = \sum_{m=1}^{M} A_m \cos(2\pi mFt + \phi_m), \tag{2.1}$$

it triggers an auditory sensation called pitch [6], which allows classifying sounds in a scale that goes from bass to treble [5].

In Western culture, music is traditionally played using a discrete set of pre-defined pitches [6]. Each one of these elements is called a musical note. Although there are different techniques for tuning notes, it is generally accepted, for automatic transcription purposes, that F0 values are drawn from an equal-tempered scale, defined by steps (or semitones) corresponding to a frequency ratio of $\sqrt[12]{2}$. For historical reasons, notes were called A, A# (or Bb)[1], B, C, C# (or Db), D, D# (or Eb), E, F, F# (or Gb), G and G# (or Ab), in a total of twelve different tones, comprising one octave. In that notation, it is possible to refer to the octave of a specific note using a number, *e.g.*, A4 is note A in the fourth octave. The fundamental frequency assigned to the note A3 is half the one related to A4 and so on. Conventionally, the note A4 is tuned so that its fundamental frequency is 440 Hz, and this allows the definition of the F0s for all other notes by applying the $\sqrt[12]{2}$ ratio.

In order to execute a musical piece, the musician generally follows instructions to play a particular sequence of musical notes. These instructions may be taught by oral tradition, but written notations were developed across history. Different forms of musical notation have arisen, each one being more adequate to specific forms of playing, understanding and composing music [7].

In Western culture, a common form of notation is the score, an example of which is shown in Figure 2.1. In this notation, each note to be played is represented by a symbol (a vertical line and a notehead) in a staff with five lines. Details on the symbol describing each note are used to represent its duration, that is, for how long the note should be played, and the vertical positioning of each notehead describes which note should be played. There are many other characteristics of music that may be written in a musical score, but a complete description is beyond the scope of this thesis.

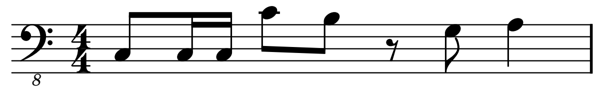

Figure 2.1: Example of traditional Western musical score.

It is important to notice that the score notation presents relative timing, that is, note

[1]A# is read as "A sharp" and Bb is read as "B flat".

durations are represented not as absolute values (in seconds), but as a ratio of a reference. This allows a particular piece to be played faster or slower, according to the interpretation of the musician. Less freedom is given to the interpreter by using a notation that is more accurate in time, for example a piano-roll. In this notation, each note event is represented by its onset and offset (that is, the exact time when the note should start and stop playing), and its pitch. For the purposes of inter-device communication using the MIDI (Musical Instrument Digital Interface) protocol, the pitch of each note is described by an integer [8] (called MIDI number) calculated by:

$$p = 69 + 12 \log_2 \frac{F}{440}, \tag{2.2}$$

where $p$ depicts a hypothetical sorting of piano notes in which A4 corresponds to 69, A#4 to 70 and so on, and $F$ is the corresponding fundamental frequency, in Hz.

Due to the extensive use of MIDI devices in the context of electronic music, the piano-roll notation is often referred to as *MIDI notation*. The piano-roll related to a particular piece may be visualized in a Cartesian plane, where the y-axis represents the pitch $p$ and the x-axis represents time. This representation may be seen in Figure 2.2.

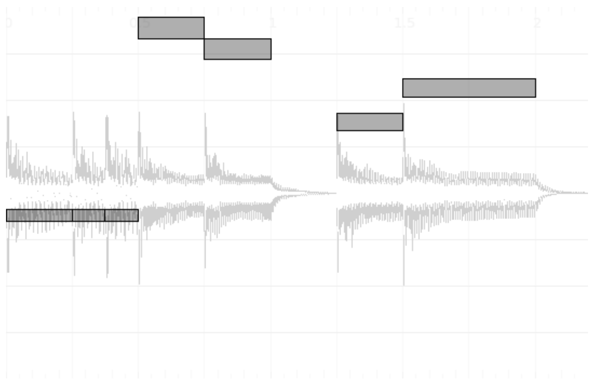

Figure 2.2: Example of a piano-roll and the related acoustic signal, which was synthesized from the score in Figure 2.1.

All kinds of notations have their own advantages and drawbacks. For example, although the piano-roll notation reveals the exact duration intended for each note, it is not as easily readable or interpretable by a human being as the traditional score.

In order to obtain the transcription of a particular piece, it is necessary to define the symbols that best describe the piece considering the desired notation. When the transcription process is performed by a device without human interference, it is called automatic transcription. Transcription of music is only possible because auditory sensations related to different musical notes are distinguishable. To build an automatic transcriber, it is necessary to understand the relationship between these sensations and particular signal models, as well as the conditions under which these models work.

### 2.1.1 Pitch

The harmonic model in Expression 2.1 is known to be overly simplistic, as audio synthesis derived directly from it is easily recognized as artificial. Nonetheless, it remais a useful tool in many audio-related applications. There are several methods to detect the fundamental frequency (and, therefore, the pitch) of a harmonic signal. If only one note is played at each time instant (*i.e.*, the audio signal is said to be monophonic), the application of the model in Expression 2.1 is straightforward. When $J$ notes are played concurrently, the resulting signal may be described as a sum of signals derived from Expression 2.1:

$$y(t) = \sum_{j=1}^{J} \sum_{m=1}^{M_j} A_{m,j} \cos(2\pi m F_j t + \phi_{m,j}), \tag{2.3}$$

where $F_j$ is the fundamental frequency of the $j$-th note and $\phi_{m,j}$ and $A_{m,j}$ are the phases and amplitudes of its $m$-th partial.

The sensation that arises from hearing this signal is that of a sum of sounds with different pitches. This is what happens when more than one note of a piano, for example, is played at the same time.

Expressions 2.1 and 2.3 can only be used in stationary excerpts, that is, while musical notes are sustained. When transient behavior is found, e.g., during note onsets, different signal characteristics are found and, therefore, different techniques are necessary to detect them, as it will be discussed later.

### 2.1.2 Onsets

Onsets of new musical notes may be detected by a change on the stationary behavior of the signal. A complete tutorial on the detection of onsets was written by Bello [9]. Note onsets may be detected using typical characteristics of starting notes, which are:

1. Increase in the signal power, for notes with sharp attacks, like a plucked string, and

2. Change in the spectral content (that is, the frequency distribution of the signal power), for soft attacks, which can be achieved, for example, in wind or bowed instruments.

These assumptions may be used as an inspiration to build DSP algorithms for finding onsets. Similarly to the pitch, the onset models give continuous values. Therefore, techniques for finding discrete events among these continuous signals must be applied.

Due to this demand, the structure of most AMT systems comprises two parts, as shown in Figure 2.3. First, a DSP algorithm calculates objective features of the audio signal related to models of auditory phenomena. After that, a machine learning algorithm uses these features to make binary decisions about what notes are present in the signal.

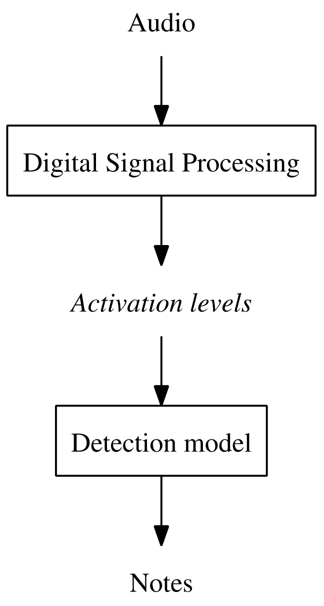

Figure 2.3: Flowchart for a generic AMT system.

After inspecting the literature, it was found that the most recent advances in digital signal processing have been derived from concepts and techniques related to machine learning. They will be discussed in the next section.

## 2.2 Techniques for Automatic Music Transcription

The final goal of an automatic music transcriber is to obtain a proper description of the musical notes that have been played to generate the signal received as input. This corresponds to inferring the pitch, the onset and the offset of each note. The harmonic model in Expression 2.1 implies that it is necessary to detect what are the frequency partials of a frame of audio in order to infer the pitches of the active notes in that frame.

The simplest technique to detect partials is based on peak-picking and thresholding in the frequency domain, which corresponds to selecting all local maxima whose absolute values are above a pre-defined threshold [10–14]. A slightly more complex approach involves low-pass filtering the DFT of the analyzed signal so that spurious peaks are eliminated [15]. Another simple operation that may be performed is to apply a parameter estimation algorithm. Hainsworth [16] uses the method proposed by MacLeod [17], which works by obtaining parameters according to a maximum likelihood criterion. In the system proposed by Moorer [18], sub-harmonic spurious components are eliminated by discarding fundamental tone candidates that do not present even harmonics.

Once the parameters (magnitude and frequency) of the existing partials are found, it is necessary to group them in order to find musical notes. Piszczalski [19] simply assumes that only a single note sounds at each time frame, without any other noise. This is not true in the general case, but can be representative of some specific and pertinent case, like those engendered by a solo flute or voice. Sterian [11] proposes using multiple time frames, grouping partials that do not change in magnitude or frequency by more than a pre-defined threshold. The result of this grouping process is a set of tracks, which are grouped using the harmonicity criterion. Tanaka [12] uses a rule-based system that groups partials considering that they start and end in similar instants, their amplitudes change slowly and they are harmonically related. Lao [13] and Triki [14] add a rule according to which all partials of the same series must have a similar amplitude. Hainsworth [16] separates the tasks of finding notes and classifying them, so that an algorithm based on framewise energy variation finds note onsets and, afterwards, notes are classified according to the found partials. Similar approaches were also used by Rao and Rao [20] and Uchida and Wada [21]. Finally, Dressler [22] used a chain of pitch formation rules to detect the predominant pitch in an existing mixture.

The process of peak-picking and thresholding is interesting because it filters a great amount of noise and yields few data points. This allows the extraction of information with rule-based systems that rely on psycho-acoustic principles. On the other hand, when a greater number of musical notes are present, this process is compromised by two phenomena. First, peaks related to partials of different notes will merge, which means that peak-picking becomes more likely to fail. Second, since there may exist a great difference between the loudness of the mixed notes, finding a suitable threshold level will tend to be a harder task.

When the coefficients of a particular frame are interpreted as a vector, the problem of characterizing events in that frame may be considered as a pattern classification problem [23], and many techniques, specifically designed for solving classification problems, may be used.

A pattern classification technique is an algorithm that receives as input a set of characteristics $\boldsymbol{o}$ and yields a label $s(\boldsymbol{o})$ as output. This label characterizes the class to which the object – in this case, the frame – belongs, based on the feature vector $\boldsymbol{o}$. The transcription systems discussed up to this point may be considered as rule-based pattern classification systems, which are designed taking into account specialist knowledge. The techniques that will be discussed in the following, however, also gather information from a training dataset.

Artificial Neural Networks (ANNs) are broadly used classifiers. Among the existing ANN architectures, it is safe to consider the Multi-Layer Perceptron (MLP) as most emblematic representative [24]. This ANN architecture calculates the function:

$$\boldsymbol{y} = \boldsymbol{A} \begin{bmatrix} f(\boldsymbol{B} \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix}) \\ 1 \end{bmatrix}, \tag{2.4}$$

where the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, obtained by supervised training, contain weight parameters, $\boldsymbol{x}$ is an input vector, $\boldsymbol{y}$ is an output vector and $f(.)$ is a sigmoidal function [24], like $\tanh(.)$.

The mathematical model in Expression 2.4 is capable of universal approximation [24], *i.e.*, it is mathematically capable of modelling any function. Obtaining the parameters that yield the approximation of a particular function depends on a supervised learning algorithm, generally using a gradient descent approach that iteratively minimizes the approximation error for a given dataset, but often leads to local minima [23], which may harm the system performance.

It is important to notice that, in the domain of a transform that presents the characteristics discussed in Section 2.1.1, the magnitude of a particular coefficient may indicate the existence of a frequency component, its non-existence or a state of doubt about its existence. A possible mathematical model for these states is a sigmoidal function $f(x)$. For $x \to \pm\infty$, the function assumes negative or positive saturation values, which represent either existence or non-existence, and for intermediate values the sigmoid function presents a smooth transition, which represents an increasing value for the hypothesis of existence of the component. MLP networks were used in [25–29], in which one complete network is designed to detect if a particular note is active or inactive – therefore, there are as many networks as possible notes to detect. By using multiple networks, the dimension of the input is considerably reduced, which decreases the susceptibility of the network to converging to local minima and having sub-optimal performance.

In order to avoid local minima, recent works have favoured the use of Support Vector Machines (SVM). SVM is a machine learning technique that does not minimize an approximation error considering a dataset, but maximizes the decision margin between clusters to be classified. The training step is unimodal and performed in a single pass (that is, it is not necessary to iterate several times over data, as it is the case for MLP networks). SVM variations were used in [30–38] to classify musical notes and chords using short time spectral representations, and in [38, 39] to classify the timbre of already labeled musical notes.

Recently, deep-belief Networks (DBNs) were applied to the problem of transcription by Nam *et al.* [40]. These networks can be briefly explained as multi-layer structures in which each layer is trained independently. DBNs have shown to yield better results than SVMs in the databases used by the authors.

Neural networks tend to produce a black box machine, that is, the final transformation obtained may not correspond to the way a specialist would reason over the problem. A more semantically meaningful approach to obtain detection functions is to use probabilistic techniques, which also play an important role in music transcription. Among those techniques, it is important to discuss the Bayesian classifier [23]. This decision technique relates a vector of observations $\boldsymbol{o}$ to the strength of the hypothesis that the observed object belongs to the class $s_j$, using two factors. The first is the probability that $\boldsymbol{o}$ is generated by using a known model of the class behavior, which gives $P(\boldsymbol{o}|s_j)$. The second is the prior probability of the class $s_j$.

Using Bayes' Theorem, the probability of finding the class $s_j$ given the observation vector $\boldsymbol{o}$ is:

$$P(s_j|\boldsymbol{o}) = \frac{P(\boldsymbol{o}|s_j)P(s_j)}{P(\boldsymbol{o})}. \tag{2.5}$$

Since $P(\boldsymbol{o})$ is equal for all candidate classes $s_j$, the decision algorithm may discard it and simply choose the class given by $\arg\max_j P(\boldsymbol{o}|s_j)P(s_j)$.

Expression 2.5 produces a decision system with theoretical minimum probability of error (MPE), which means that Bayesian decision systems may, ideally, reach the best possible classification performance, given the input data. However, both $P(\boldsymbol{o}|s_j)$ and $P(s_j)$ are unknown and must be estimated, which relates the performance of the classification systems to the quality of the estimation [23]. Bayesian decision systems were used in [41–45].

When dealing with framewise classification of musical signals, it must be noted that there is a strong correlation between the classes of adjacent frames, because there is a good chance that no new note events happened between these frames. This means that the prior probability of observing a specific class depends on the previous decision. This premise is used in Hidden Markov Model (HMM) structures [46]. HMMs are discrete-time systems in which the $q$-th state depends on the state in the instant $q-1$, that is, $P(s_{j,q}) = P(s_{j,q}|s_{j,q-1})$, which is a property ignored by the Bayesian decision process. Each state may be interpreted as a classification decision, and, with a carefully chosen topology, states will represent the desired musical events. HMMs were used in [30, 47–51], with different topologies and data sets.

Linear algebra can also provide a sound theoretical framework of such kind. In this case, a very common model for the detection of musical notes is:

$$\boldsymbol{X} = \boldsymbol{B}\boldsymbol{A}, \tag{2.6}$$

where $\boldsymbol{B}$ and $\boldsymbol{A}$ are non-negative and $\boldsymbol{B}$ has as many columns as the number of notes that are going to be detected.

In this model, each column $\boldsymbol{x}_q$ of $\boldsymbol{X}$ must contain a short time description of the input signal, in a vector representation that does not change significantly if the pitch itself does not change – for example, the absolute value of the DFT that description. The representation $\boldsymbol{x}_q$ is factorized as a combination $\boldsymbol{a}_q$ of the basis vectors stored as columns of the $\boldsymbol{B}$ matrix, hence $\boldsymbol{x}_q \approx \boldsymbol{B}\boldsymbol{a}_q$. Therefore, $a_{d,q}$ is the strength of activation of the $d$-th basis vector during time frame $q$. Also, the model inherently has a non-negativity constraint, which holds at least for $\boldsymbol{A}$, since it does not make any sense to have a note "subtracted" from the mixture. The representation used in this model is, in general, some kind of spectrogram, hence the values of $\boldsymbol{X}$ and $\boldsymbol{B}$ are also non-negative. Therefore, finding the factors $\boldsymbol{B}$ and $\boldsymbol{A}$ is often referred to as "Non-Negative Matrix Factorization" (NMF) [52].

Since the factorization model in Expression 2.6 is inexact, an approximation must be calculated. This depends on the choice of a suitable error measure. The first one that has been used in AMT was the Euclidean distance, that is, the error $\epsilon$ is given by:

$$\epsilon = \|\boldsymbol{X} - \boldsymbol{B}\boldsymbol{A}\|. \tag{2.7}$$

Smaragdis [52] used an iterative gradient-based approach to obtain both $\boldsymbol{B}$ and $\boldsymbol{A}$ from a spectrogram $\boldsymbol{X}$. Although it is a non-supervised learning technique, experiments show that $\boldsymbol{B}$ usually converges to basis vectors corresponding to notes and $\boldsymbol{A}$ converges to their corresponding activation weights. Aditionally, Smaragdis [52] observed that note events that are never found separately (notes that are always in a chord) are not recognized by the system (only the corresponding chords are recognized), and, at the same time, some spurious events are recognized, which represents a significant drawback of the factorization system. Later, Bertin [53] showed that NMF yields better results than a Singular Value Decomposition (SVD). Sophea [54] and Vincent [55] proposed using of prior knowledge on the pitches, both restricting the values on the base matrix $\boldsymbol{B}$ so that only values corresponding to the fundamental frequency and the overtones would be allowed to be different than zero. In parallel, Grindlay [56] used instrument-specific data to obtain the basis matrix $\boldsymbol{B}$, hence it would not have to be updated while searching for the values of $\boldsymbol{A}$.

Bertin [57] converted the usual NMF approach to a probabilistic framework, in which the values of the spectrogram and the factor matrices are interpreted as probabilities. This was useful to connect the factorization results to a hypothesis of temporal smoothness, that is, the expectancy of notes lasting for long time intervals was incorporated to the system. This probabilistic framework is called "Bayesian Non-Negative Matrix Factorization" and was later used in experiments for automatic transcription of polyphonic music [58]. A similar technique, namely Probabilistic Latent Component Analysis (PLCA), was used by Han to solve the problem of transcription [59]. Although the calculations performed in the Bayesian approach can be modelled as matrix factorization problems, the use of a probabilistic framework may allow not only a more meaningful interpretation of both the results and the hypothesis over which the system is built on, but also the use of an important set of tools and algorithms, as seen above.

Under the hypothesis that the base matrix $\boldsymbol{B}$ is informed *a priori*, it is possible to interpret the factorization of each column of $\boldsymbol{X}$, in Expression 2.7, as an independent problem, that is:

$$\min_{\boldsymbol{a}_q} \epsilon_q = \|\boldsymbol{x}_q - \boldsymbol{B}\boldsymbol{a}_q\| \tag{2.8}$$

with the constraint that $a_{q,d} \geq 0$, $\forall d$, thus obtaining the corresponding weight vector $\boldsymbol{a}_q$.

The hypothesis of time independency between columns allows the construction of causal algorithms for transcription, which is a pre-requisite for real-time applications. A gradient-based rule may be used, but an algorithm specifically designed to solve the problem in Expression 2.8

was proposed by Hanson and Lawson [60]. It was used in the context of AMT by Niedermayer [61], Mauch and Dixon [62] and Tavares *et al.* [63].

Bertin [64] observed that the Euclidean distance may not be the best error measure for the approximation of Expression 2.6, using the Itakura-Saito divergence instead, defined as:

$$\epsilon_I(x|y) = \frac{x}{y} - \log\frac{x}{y} - 1. \tag{2.9}$$

Bertin [64] argues that this divergence could be more suitable for transcription because it is invariant to a linear gain (that is, $\epsilon(x|y) = \epsilon(\lambda x|\lambda y)$).

However, the Itakura-Saito distance is not convex, in contrast to the Euclidean distance. For this reason, Bertin proposes using the $\beta$-divergence, given by:

$$\epsilon_\beta(x|y) = \begin{cases} \frac{x^\beta + (\beta-1)y^\beta - \beta x y^{\beta-1}}{\beta(\beta-1)}, & \beta \in \Re\backslash\{0,1\} \\ x - \log\frac{x}{y} + (x-y), & \beta = 1 \\ \frac{x}{y} - \log\frac{x}{y} - 1, & \beta = 0. \end{cases} \tag{2.10}$$

Since the $\beta$-divergence is convex for $\beta \in [1,2]$, Bertin [64] proposes minimizing the cost function for $\beta = 1$ and then gradually reducing it until it converges to zero, in which case the $\beta$-divergence is equivalent to the Itakura-Saito divergence. The $\beta$-divergence was also used by Dessein *et al.* [65], simply assuming $\beta = 0.5$.

The factorization problem in Expression 2.6 is akin to a source separation problem, in which a number of sources is mixed into different channels. In this case, the sources are the individual notes, each with its own activation level, and the mixture is the final spectrogram. Obtaining the factorization $\boldsymbol{BA}$, under this assumption, is equivalent to solving a Blind Source Separation problem. This interpretation was used by Abdallah [66] and, later, by Boulanger-Lewandowsky *et al.* [67], to build a transcription system based on the assumptions that the activation of the notes is independent and the activation matrix is sparse, that is, few notes are active at each frame. A similar idea was used by Vincent [68], who also incorporated the idea of time dependency for notes and applied the system to recordings instead of the synthesized harpsichord samples used by Abdallah [66]. The fact that few notes can be played at the same time was exploited by Tavares *et al.* [63], who applied computer vision techniques to detect which notes can be played in a vibraphone given the position of the mallets, restricting the search space and reducing false positives. Lee *et al.* [69] incorporated sparseness in the factorization process by minimizing the $l_1$ norm, shown in Expression 2.11, in the factorization process:

$$l_1(\boldsymbol{x}|\boldsymbol{y}) = \sum_{n=1}^{N} \|x_n - y_n\|. \tag{2.11}$$

Benetos and Dixon [70] observed that, in the Constant-Q Transform (CQT) domain, templates assigned to different notes can be obtained by shifting other templates in the frequency domain. Later, Kirchoff *et al.* [71] improved this approach, allowing the use of more than one spectral template per note. The property of shift-invariance was indirectly used by Argenti *et al.* [72], who used 2D (time-frequency) templates for musical notes, hence considering the spectral changes that happen during the execution of a musical note.

Although factorization methods have shown to be effective, they are computationally expensive and may become prohibitive for large, real-time problems. An approximate solution, however, may be obtained by using the Matching Pursuit algorithm proposed by Mallat and Zhang [73]. This algorithm consists of iteratively selecting, within the dictionary $\boldsymbol{B}$, the vector whose inner product with the input vector is the greatest. The base function is subtracted from the input so that it becomes maximally uncorrelated to that base function, and then the algorithm proceeds by analyzing the remainder of that subtraction. The Matching Pursuit technique, which may be seen as a "greedy" heuristic to minimize Expression 2.8, may converge to a local minimum, and it will tend to give results that differ from the ones yielded by NNLSQ algorithm when the basis functions are more correlated. It was used for the task of transcription by Derrien [74]. A mixed algorithm developed by O'Hanlon *et al.* [75] uses a greedy approach to select the notes that are the most efficient to model a mixture, but applies NNLSQ, with a sparsity constraint, to improve the estimation of their ratios.

The decision algorithms described above are those that, historically, have been more frequently used in AMT, but it is also important to mention other relevant techniques. Knowledge-based blackboard systems, like the one proposed by Martin [10], are algorithms that successively incorporate information from multiple, independent agents. Each agent is responsible for dealing with a particular situation, and its output may generate a situation that will be dealt with by other agent. After multiple iterations with all agents, the system yields the final transcription. Reis [76] uses a genetic algorithm, which generates random musical piece candidates, estimates their spectrum and uses the Euclidean distance from the estimated spectrum to the true spectra to evaluate their fitness. Different candidates are combined, changed, replicated and rejected using evolutionary strategies, and after some iterations the desired transcription is obtained. Later, Reis *et al.* [77] incorporated variance on the spectral envelope and a dynamic noise level analysis, significantly outperforming the previous approach [76].

All these proposals, although using considerably different techniques, are based on similar premises, which were discussed in Section 2.1. In the last decades, the scientific community has agreed on some evaluation methods that can be used to measure which techniques work best. Section 2.3 brings a discussion on this subject.

The information presented in this section is condensed in Tables 2.1 and 2.2. These tables list, respectively, techniques based on peak-picking and vector classification.

| Peak detection | |
|---|---|
| References | Technique |
| [10], [11], [12], [13], [14] | Peak-picking and thresholding |
| [15] | Peak-picking and thresholding, low-pass filter in frequency domain eliminates spurious notes |
| [16] | Peak-picking, maximum likelihood criterion [17] is used |
| [18] | Peak-picking, specialist rule system filters peak hypothesis |

| Peak grouping | |
|---|---|
| References | Technique |
| [19] | Partial grouping with monophonic assumption |
| [11] | Assumes that partial amplitudes and frequencies do not change significantly across frames |
| [12] | Partials of same notes begin and end in similar instants |
| [13], [14] | All partials of the same series should have similar amplitudes |
| [16], [20], [21] | Notes are classified after onset/offset is detected |
| [22] | Chain of pitch formation rules highlight predominant pitch |

Table 2.1: Reference table for peak-picking based classification techniques.

## 2.3 Evaluation

Evaluating an AMT system means detecting how well it performs, and, furthermore, collecting evidence about which techniques work best for a given database. The first AMT systems [18, 19] were evaluated by visual comparison between ground-truth with automatically obtained scores. This practice remained for some time, but, with the growth of the research field, it became necessary to develop objective performance measurements that could not only be automatically calculated over large databases, but also be exploited to compare different transcribers.

The most frequently used measures are *Recall*, *Precision* and *F-Measure*. They derive from information retrieval [80] and are used in the Music Information Retrieval Exchange (MIREX) [81]. They are defined in Expressions 2.12, 2.13 and 2.14, respectively.

$$\text{Recall} = \frac{\text{\# of correctly transcribed notes}}{\text{\# of notes in ground-truth}}. \tag{2.12}$$

$$\text{Precision} = \frac{\text{\# of correctly transcribed notes}}{\text{\# of notes in automatic transcription}}. \tag{2.13}$$

$$\text{F-Measure} = 2\frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}. \tag{2.14}$$

| References | Technique |
|---|---|
| [25], [26], [27], [28], [29] | Multi-Layer Perceptron |
| [30], [31], [32], [33], [34], [35], [36], [37], [38] | Support Vector Machines |
| [52], [53], [54], [55], [57], [78], [56], [64], [58], [37], [79], [58], [65] | Non-Negative Matrix Factorization |
| [40] | Deep-belief Networks |
| [59] | Probabilistic Latent Variable Analysis |
| [66], [67], [68], [69] | Non-Negative Matrix Factorization with sparsity constrain |
| [70], [71], [72] | Shift-invariant factorization |
| [61] | Non-Negative Least Mean Squares |
| [41], [42], [43], [44], [45] | Bayesian decision system |
| [47], [48], [30], [49], [50], [51] | Hidden Markov Models |
| [10] | Blackboard system |
| [74], [75] | Matching pursuit |
| [76], [77] | Genetic algorithm |

Table 2.2: Reference table for vector-based pattern classification techniques.

There is an inherent tradeoff between Recall and Precision. If the sensitivity is too high, many notes will be yielded, the probability of correctly transcribing notes in the ground-truth will be higher, hence Recall will tend to grow. The increased sensitivity tends to lead the system to yield more false positives, which will cause the Precision to decrease. Conversely, a lower sensitivity parameter tends to lead to higher Precision, with the drawback of lowering the Recall. The F-Measure accounts for this tradeoff, representing the harmonic mean between Recall and Precision.

These measurements, however, depend on a definition of correctness for a transcribed note. Most modern AMT systems focus on converting audio to a MIDI-like representation in which notes are described by their onsets, offsets and pitches. This allows transcribing tempo fluctuations as played by the musician, but requires a manual annotation of evaluation datasets.

For a yielded note to be considered correct, it is frequently required that its pitch be within half a semitone from the pitch of the corresponding note in the ground-truth. This is justified because deviations of less than half semitone lead to the correct pitch by simple rounding. The requirements regarding time, however, are harder to define, as there are many aspects to consider.

The human perception is an important one. An onset difference starts to be noticed above about 10 ms [82]. Onset deviations of more than 20 ms tend to compromise the listening experience, but are still acceptable for database search purposes. A deviation over 100 ms is harmful to most applications.

It is also necessary to consider the feasibility aspect. In most pitch-tracking systems, the length of the analysis window is between 12 ms and 43 ms. In live applications, this length is the minimum delay between playing a note and obtaining an associated response from the system. In offline scenarios, it becomes a timing error, as it is impossible to know at which point, within a window, a given event occurred.

Last, there is a practical aspect to be considered. It is hard to manually annotate onsets in an audio file below a precision of around some tenths of milliseconds. This aspect was partially solved by using either MIDI synthesizers, MIDI-controlled instruments or by recording musicians that were playing against a MIDI recording.

All of these aspects become even more important when offsets are considered. They are harder to identify (both manually and automatically), and the auditory relevance of their deviations is harder to define numerically. For this reason, many AMT systems ignore offsets and only work with onsets.

Once the timing tolerances are defined, Recall, Precision and F-Measure may be immediately calculated. Although they allow a quick comparison between methods, they reveal little about the nature of the errors. Furthermore, the impact of the tolerances in the final outcome is not measured, which may generate misleading results.

To account for that, another approach consists on evaluating the transcription in short frames, of around 10 ms [81]. For each frame, the number of active notes in the ground-truth and in the automatic transcription are counted, as well as the number of correctly transcribed notes. These numbers are summed, allowing to calculate Recall, Precision and F-Measure. In this case, the measures do not depend on a subjective choice of timing, and notes are simply considered proportionally to its duration.

Both the notewise and the framewise evaluation methods, however, do not provide information about the characteristics of the algorithm failures. Tavares *et al.* [83] developed a method that yields histograms for time and pitch deviations. This method highlights information that may be useful, such as detecting the typical time delay, the number of pitch errors for each pitch class and so on. On the other hand, it is unable to provide a unique performance measure.

Daniel and Emiya [84] observed that Recall, Precision and F-Measure do not necessarily reflect the perceptual accuracy of a transcription. In reality, different types of errors are perceived differently, for example, timing deviations are less perceptible than pitch deviations. To account for that, Fonseca and Ferreira [85] proposed a perceptually-motivated evaluation method based on applying different evaluation processes to the decaying and sustained parts of a note. This method has shown to be perceptually more accurate than the usual measures.

The standardization of performance measures comprises an important step towards the comparison of AMT systems. However, performance comparison also requires executing different methods over the same dataset. Henceforth, significant effort has been made to provide adequate datasets aiming at future research.

An important step towards test standardization was taken by Goto *et al.* [86], who developed the Real World Computing (RWC) database. It comprises audio and MIDI files for 115 popular music songs, 50 classical pieces and 50 jazz pieces. All of them were recorded especially for the database. This database was used by Ryynanen and Klapuri [49, 50, 87, 88], Benetos and Dixon [70], Raczynski *et al.* [89], Simselki and Cemgil [90] and Argenti *et al.* [72]. However, each one of these works used a different subset of the RWC database.

Aiming at evaluating transcriptions of piano solos, Poliner and Elis [32] used a dataset consisting of 92 pieces downloaded from the Classical Piano MIDI Page (`http://www.piano-midi.de/`) and rendered using standard software (Apple Itunes). A similar dataset was used by Costantini *et al.* [34], who not only used the same database construction method, but also published an explicit list of all MIDI files used in the process. Nam *et al.* [40] also used this database, and performed additional tests on it with previous methods [29, 49]. The evaluation performed on this database was carried out using frame-level note detection. Table 2.3 shows the average accuracy (F-Measure) achieved by each method, as well as a brief description of the employed techniques.

| References | Accuracy | Technique |
|---|---|---|
| [32] | 70% | SVM |
| [34] | 85% | SVM with memory |
| [40] | 79% | DBN |
| [49] | 46% | HMM and specialist signal processing |
| [29] | 39% | MLP networks |

Table 2.3: Performance comparison for methods using the Poliner-Ellis database.

Using synthesized data may lead to scenarios devoid of aspects such as room reverberation and instrument resonances. The MAPS (standing for MIDI Aligned Piano Sounds) dataset [91] provides high-quality recordings of a Yamaha Disklavier, that is, an automatic piano. It contains 31 GB of recordings with corresponding ground-truths, and may be freely downloaded. It was used by Dessein *et al.* [65], Nam *et al.* [40] and O'Hanlon *et al.* [75]. Table 2.4 shows the F-Measure (both notewise and framewise) reported in each work.

## 2.4 Discussion

This review shows that there are two main approaches to solve the problem of transcription. The first one is to program sub-systems that apply specialist knowledge aiming to detect psy-

| References | Results | | Technique |
|---|---|---|---|
| | Notewise | Framewise | |
| [65] | 71.5% | 65.5% | NMF with *beta*-divergence |
| [40] | - | 74.4% | DBN |
| [29] | - | 63.6% | MLP networks |
| [75] | 78.2% | 76.3% | Sparse NMF decomposition |

Table 2.4: Performance comparison for methods using the MAPS database. Unreported results are identified with a dash.

choacoustic clues that may lead to the correct transcription, and then combine the results into the desired transcription. If the signal behavior that triggers a particular sensation is known and properly modeled, then its use will imply in good results. However, that assumption holds only partially, both because of errors resulting from inexact mathematical modelling and because of errors due to the operation of some algorithms in ill-conditioned situations. Also, as noted by Hainsworth [92], it must be considered that music transcription, when performed by human beings, demands a high level of training and specialization, hence psychoacoustic models obtained by analysis of non-expert subjects may be ignoring some aspects that are important to build an automatic transcriber. This leads to the second approach to AMT, which consists of defining a machine learning process that automatically estimates key parameters from a data corpus, as it is the case for NMF or SVM-based methods. These will present all training problems that are typical of machine learning processes, that is, the need for a great amount of data for training, the fact that parameters are not likely to be easily interpretable and the impossibility of theoretically granting that the results are not subject to particularities of a given database. Machine learning algorithms, however, have been shown to lead to a good performance, not rarely outperforming systems built from specialist knowledge.

As can be seen in Tables 2.3 and 2.4, the best transcription results have been obtained when machine learning algorithms are mixed with specialist knowledge. The use of memory for SVMs [34] and sparsity for the NMF [75] lead these systems to outperform previous ones. This indicates a promising direction for future AMT research.

## 2.5 Conclusion

This chapter has presented an overview on automatic transcription of music. It presents historical remarks on how techniques have evolved in the last twenty years, and the concepts and inspirations behind the most commonly used techniques. The specific transcription tasks that are dealt with by most AMT systems were also discussed.

It was noted that techniques that consider long-term characteristics of the signals tend to outperform the others – HMM outperforms MLP, SVN works best if a memory unit is used, and sparsity constraints have improved the results when NMF is used. This suggests that future work in AMT should focus on the development of machine learning techniques that exploit long-term characteristics that are often found in music, such as sparseness in time and frequency and a tendency for continuity in spectral representations. The next chapters will discuss how to apply this idea in practice.

Chapter 3

# Interactive Transcription of Music

In many musical genres, there is a visible use of explicit repetitions of notes. In the context of contemporary Western popular music, this allows piece analysis using concepts such as “verse” and “chorus”. This chapter presents a technique that exploits this characteristic to deliver a fast way to transcribe polyphonic music using computer assistance.

In the proposed workflow, the user is required to input the first notes of a piece. After that, the system predicts another note, being manually corrected, if necessary. As a result, an accurate documental transcription is obtained.

The inspirations behind this system are discussed in Section 3.1. After that, the theoretical basis for several design decisions of this system are presented in Section 3.2. Section 3.3 discusses the proposed method in detail, presenting the forecasting algorithm.

The accuracy of the proposed method is evaluated in Section 3.4. It is shown that, for some genres, the proposed method provides an accurate transcription with less human effort than if an automatic system were used.

Section 3.6 concludes the chapter by presenting a brief summary of the discussed topics.

## 3.1 Context

Over the last years, increasing attention has been given to tools capable of detecting, without human interference, the musical notes present in an acoustic recording. These tools are called automatic music transcribers, and have been used for many purposes, such as query-by-content database search algorithms [93], tutoring devices [94, 95], and for providing scores for documentation and analysis purposes [88, 96]. These last two applications – documentation and analysis – require a great accuracy on the part of the automated system, which cannot yet be provided by most state-of-the-art systems. Hence, it is necessary to perform manual corrections to the transcription yielded by the automatic systems.

Aiming at these applications, a different workflow for obtaining documental transcriptions is proposed: the user provides a few notes in the beginning of the score and an algorithm predicts what is the most likely continuation. Then, the user either corrects or accepts the predicted continuation and the algorithm yields another note candidate. These iterations happen until the transcription is completed. This workflow is different from the usual one, in which the user receives a candidate note sequence and then performs corrections on it. Experimental results show that the effort required from the user when employing the proposed transcription workflow, namely Interactive Computer-Assisted Music Transcription (ICAMT), is significantly lower than the effort required when using the traditional approach.

The forecasting algorithm relies on the detection of structure repetitions in the notes provided by the user, an approach based on the *modus operandi* of the universal compression algorithm proposed by Lempel and Ziv [97]. It does not require a training stage based on a corpus, as all information is obtained from the particular piece being transcribed. Also, it does not have great memory or processor requirements, hence it can be implemented even in mobile devices.

As will be seen later, the algorithm is, in general, capable of predicting correctly a significant amount of notes in popular music databases, outperforming state-of-the-art automatic transcription systems in the sense that the user is required to provide less notes than if the usual workflow – automatic transcription preceding corrections – is used. It is also shown that, for this purpose, the proposed algorithm performs better than an existing note forecasting algorithm [98].

## 3.2 Theoretical basis

Musical pieces, for the purpose of transcription, may be interpreted as sets of discrete events, or symbols, each one related to a particular musical gesture – for example, one possible event is “play note C for one beat of the song, starting after the preceding note ends”. This set of events is generally crafted by a human being (unless an algorithmic composition process is used), hence it is, in a certain sense, unpredictable. On the other hand, musical pieces tend to be based on guidelines, which are a result of both formal learning (such as studies on harmony and counterpoint) and intuition derived from experience (such as listening to other compositions and copying structures that sound better than others). These guidelines (both formal and intuitive) direct the composition process of a musical piece so that certain structures tend to be more likely to occur than others. In fact, as widely discussed by Howell *et al.* [99], music is essentially a social activity; therefore, it demands rules and structure. This means that, in a certain sense, specific parts of a musical piece may be predicted based on other parts.

The relationship between the existence of an underlying structure of a sequence of symbols and the predictability of that sequence was studied by Shannon [100]. In his work, he used Markov chains to forecast characters of texts in English, obtaining significant results. Shannon [100] states that this gives support to the notion that part of the characters in a text are freely chosen, while the remainder is bounded to the structure of the language. In another important contribution, Solomonoff [101, 102] studied the predictability of generic symbol strings from the perspective of their known statistical properties.

Later, Ziv and Lempel [97] observed that, while Markov chains model a probabilistic output, it is possible to develop a deterministic algorithm that predicts sequences of symbols, hence allowing compression of strings without loss of information. Their algorithm iteractively builds a codebook of sequences of bits that are always followed by the same bit in the string, as if it were a variable-order Markov chain where all transition probabilities are either zero or one. Ziv [103] showed that the compression capacity of the algorithm, for long strings, converges to Shannon's limit [104].

The codes obtained by Ziv and Lempel's algorithm [97] allow building models which are more flexible than classical Markov chains, due to their variable length. Cleary and Witten [105] developed an algorithm that combines both the ability of acquiring previous knowledge of the Markov chains and the variation of the code length by allowing strings to be partially (instead of fully) matched to an n-order Markov hypothesis. This gives rise to a new problem, which is that of defining what are the differences that will be allowed to exist between previously learned sequences and what should be modelled as new sequences.

In the context of music, Cleary and Witten's algorithm was used by Conklin and Witten [106, 107], who explored the idea of partial string matching to predict the scores for the soprano voice of Bach's chorales. Such prediction scheme, called Prediction by Partial Matching (PPM), combines both information from a learned corpus and from previous events of the piece to predict. Later work by Witter *et al.* [108] compares the performance of Conklin's algorithm to the performance of human beings in the same chorale prediction problem.

Ron *et al.* [109] observed that the number of nodes in Markov chains grows exponentially as its order increases, even when using partial string matching. Ron proposed a prediction algorithm in which low-order strings are used as basis for variable-order predictions, that is, symbols are used to predict, simultaneously, a certain number of future symbols. The algorithm, called Probabilistic Suffix Automata (PSA), preserves low complexity even when a high number of simultaneous predictions are used. Trivino and Moralez-Bueno [110] applied the PSA to the problem of forecasting music, and observed that, although it is impossible to predict in a deterministic fashion the events in a certain musical piece, it is possible to determine a probability distribution related to the next symbol of that same sequence. Trivino and Moralez-Bueno [110] used the same musical corpus as Conklin and Witten [106] to perform experiments.

Assayag *et al.* [98] use an idea similar to Conklin's [106] to build codebooks from existing data. After that, the next note of a given sequence is predicted using the most common continuation found in the codebook for that given excerpt. Differently from Conklin's [106], Witten's [108] and Trivino and Moralez-Bueno's [110] works, the system proposed by Assayag *et al.* was developed aiming to produce a plausible continuation for a certain piece, and not necessarily the continuation that was found in the original score.

For the same purpose, Pearce and Wiggins [111], proposed to use the PPM algorithm and a finite alphabet of rhythms and pitches to generate music. Later, the generated pieces were mixed with ones composed by human beings, and an audience was asked to identify which pieces were generated by computer and which were not.

Maxwell *et al.* [112] included the explicit development of tree structures that link similar sections of the song in order to improve the quality of the predictions. Furthermore, the evaluation was qualitative and based on the description of the generated pieces.

A common aspect of all these music prediction methods is that they are bounded to the use of a finite alphabet of rhythmic symbols. Moreover, except for the method proposed by Assayag *et al.*, such alphabet is defined *a priori* as a combination of symbols from the traditional score notation. The hypothesis that there is a small number of rhythmic structures does not hold in general for Western music. Musical events, in live performances, often present a continuous variation on their onsets and offsets, due to either intentional stylistic marks like interpretation of a score and swing or inherent variations due to the execution of musical gestures.

Aiming at the generation of an accurate score, this chapter presents a novel method for forecasting the continuation of musical performances. The method uses an idea similar to those proposed by Lempel and Ziv [97] and Assayag *et al.* [98], that is, searching for matching sub-sequences within the data and deducing a similar continuation. The matching algorithm, however, incorporates explicit musical knowledge, hence increasing the amount of the inherent structure of the music language that is modelled. As stated by Shannon [100], this is expected to increase the accuracy of the system in the prediction task.

The proposed method models the composer's decisions regarding the continuation of songs, which is different from modelling the expectation that is created in the listener when a certain sequence of notes is presented. Models for the listener's expectancy are broadly discussed by Huron [113] and Namour [114].

## 3.3 Proposed Method

As previously stated, the note forecasting method proposed in this chapter is based on the assumption that note sequences will be repeated within a particular piece. This means that a sequence of notes that is found twice within an excerpt is likely to be continued the same way. The assumption of repetition, as will be seen later, holds for many popular songs.

Before searching for repetitions, it is necessary to represent the musical piece in a convenient form. In this work, a musical note is defined by three numbers: *onset*, which is the time (in seconds) after the beginning of the execution of the piece in which the note will be played; *offset*, which is the time (in seconds) after the beginning of the execution of the piece in which the note will be damped, for example by releasing the key in a piano; and *pitch*, which is an integer that indicates to what key, in a piano keyboard, the note is related to. The whole piece is represented as an array of notes, which are sorted in ascending onset order. Notes with the same onset are sorted according to their pitch. This representation defines three arrays: $\boldsymbol{s} = [s_1, s_2, ..., s_N]$, $\boldsymbol{e} = [e_1, e_2, ..., e_N]$ and $\boldsymbol{p} = [p_1, p_2, ..., p_N]$, where $s_n$ contains the onset of the $n$-th note, $e_n$ contains its offset, and $p_n$ contains its pitch. All of these arrays have size $N$, which is the number of notes in the piece or excerpt being represented.

The algorithm relies on the values $M$, which marks the end of a subsequence, $K$, which is the length of the subsequence and $N$, which is the length of the whole symbol sequence. Thus, it will search, within the $N$ existing notes, the longest subsequence $[M-K+1..M]$ whose notes are equivalent to those in the final subsequence $[N-K+1..N]$. After the subsequence $[M-K+1..M]$ is found, it is reasonable to assume that the continuation of the excerpt (that is, note $N+1$) will be equivalent to the continuation of the subsequence (note $M+1$). A sequence of notes is equivalent to another if the $k$-th notes of both subsequences are equivalent for all $k$. The criteria for considering two notes equivalent depends on particular assumptions about the musical rules employed in the process.

The first assumption is that the final subsequence of the existing excerpt is being generated as an explicit repetition of another subsequence. In this case, two notes $m$ and $n$ are considered equivalent if they can be considered as a simple shift in time of each other, that is:

1. Their pitch is the same, that is, $p_m = p_n$,
2. Their duration is the same, within an allowed deviation, that is, $\|(e_n - s_n) - (e_m - s_m)\| < \alpha$, and
3. The interval between their onset and the onset of the previous note is the same, within an allowed deviation, that is, $\|(s_n - s_{n-1}) - (s_m - s_{m-1})\| < \beta$.

While rules one and two ensure that the notes are similar as symbols without context, rule three implies that the note is related to the previous one. This implicitly defines the strict repetitive structure of the subsequence. These rules are written in the form of the pseudo-code shown in Figure 3.1.

```
function EXPLICIT($n, m, \boldsymbol{s}, \boldsymbol{e}, \boldsymbol{p}, \alpha, \beta$)
    if ($p_n = p_m$) and
($\|(e_n - s_n) - (e_m - s_m)\| < \alpha$) and
($\|(s_n - s_{n-1}) - (s_m - s_{m-1})\| < \beta$) then
        return True
    else
        return False
```

Figure 3.1: Explicit$(n, m, \boldsymbol{s}, \boldsymbol{e}, \boldsymbol{p}, \alpha, \beta)$ detects if two notes are equivalent, considering the hypothesis of an explicit repetition.

The tolerance values $\alpha$ and $\beta$ compensate for a possible deviation inherent to the inputs. They should be set to values high enough so that most natural timing deviations will be shorter than both of them, but not as high as the length of most notes. Values between 100 ms and 200 ms for both $\alpha$ and $\beta$ have shown to work well in most cases, and $\alpha = \beta = 100$ ms was used for experiments.

The second assumption that can be made is that the final sequence of the existing excerpt is a transposition of another subsequence. This means that, although the pitches are not the same, the pitch variations should be equal. Hence, rules two and three related to the explicit repetition hypothesis still hold, but rule one must be changed to: $p_m - p_{m-1} = p_n - p_{n-1}$. The rules for detecting a transposed note equivalency are shown as a pseudo-code in Figure 3.2.

```
function TRANSPOSED($n, m, \boldsymbol{s}, \boldsymbol{e}, \boldsymbol{p}, \alpha, \beta$)
    if ($p_n - p_{n-1} = p_m - p_{m-1}$) and
($\|(e_n - s_n) - (e_m - s_m)\| < \alpha$) and
($\|(s_n - s_{n-1}) - (s_m - s_{m-1})\| < \beta$) then
        return True
    else
        return False
```

Figure 3.2: Transposed($n, m, \boldsymbol{s}, \boldsymbol{e}, \boldsymbol{p}, \alpha, \beta$) detects if two notes are equivalente, considering the hypothesis of a transposition.

The proposed algorithm for detection of transposed sequences considers chromatic transpositions, but it is possible to perform diatonic, modal or even other kinds of transpositions. Modelling each one of them would add complexity to the system, because the scale used for transposition would have to be detected. Hence, such models were not implemented.

If a long subsequence of explicit repetitions is found, it will also be explainable as a special transposed subsequence where the transposition is zero. However, using both rules may lead to ambiguous cases. In such cases, an explicit repetition of size $X$ is found to yield a different continuation than a transposed repetition of size $X$. We found that using the simpler (explicit repetition) model yielded good results.

The third and last assumption that will be considered is that the final sequence of the existing excerpt is a transposition of another subsequence, played at a different tempo. In this case, all notes are played equally faster or slower, with a constant ratio $\gamma$ between the durations of the compared notes of the subsequences, that is, $\gamma = (e[N] - s[N])/(e[M] - s[M])$. In this case, rule two must be substituted by $\|(e[n] - s[n]) - \gamma(e[m] - s[m])\| < \alpha\gamma$ and rule three must be substituted by $\|(s[n] - s[n-1]) - \gamma(s[m] - s[m-1])\| < \beta\gamma$. This set of rules is written as a pseudo-code in Figure 3.3.

Again, this new assumption is more general than the previous one. For the same reason stated before, the simpler, more specific rules must be preferred over this one.

Using the similarity rules defined above, as well as the premise of choosing the simpler

```
function DIFTEMPO($n, m, \boldsymbol{s}, \boldsymbol{e}, \boldsymbol{p}, \alpha, \beta$) $\gamma \leftarrow e[N] - s[N])/(e[M] - s[M]$
    if $((p_n - p_{n-1}) == (p_m - p_{m-1}))$ and
$(\|(e_n - s_n) - \gamma(e_m - s_m)\| < \gamma\alpha)$ and
$(\|(s_n - s_{n-1}) - \gamma(s_m - s_{m-1})\| < \gamma\beta)$ then
        return True
    else
        return False
```

Figure 3.3: Diftempo($n, m, \boldsymbol{s}, \boldsymbol{e}, \boldsymbol{p}, \alpha, \beta, \gamma$) detects note equivalencies based on the assumption of a transposed repetition played at a different tempo.

assumption prior to the more complicated ones, it is possible to develop a search algorithm capable of detecting the longest subsequence equivalent to the last subsequence of an excerpt. The algorithm progressively considers every note $M$, $M < N$, as a candidate ending for a given subsequence. A counter $k$, starting at zero, is increased while notes $M - k$ and $N - k$ are equivalent, considering the simplest rule (exact repetition). If at least one valid subsequence is found, the algorithm stops and returns the value of $M$ for which the value of $k$ was the greatest. If no equivalent subsequences are found, the same search is performed using the second simplest rule (transposition) and, again, if no equivalent subsequences are found, then the third rule is used.

Next, the algorithm must yield a note, namely, note $N + 1$. Given that it was found that the longest subsequence equivalent to the one at the end of the excerpt ends in note $\hat{M}$, it is reasonable to assume that note $N + 1$ is equivalent to note $\hat{M} + 1$. Hence, each one of the different assumptions for subsequence search will give a different continuation, as follows:

- an exact repetition implies that $p_{N+1} = p_{\hat{M}+1}$, $s_{N+1} = s_{\hat{M}+1} - s_{\hat{M}} + s_N$ and $e_{N+1} = e_{\hat{M}+1} - s_{\hat{M}+1} + s_{N+1}$,
- a transposed repetition implies that $p_{N+1} = p_{\hat{M}+1} - p_{\hat{M}} + p_N$, $s_{N+1} = s_{\hat{M}+1} - s_{\hat{M}} + s_N$ and $e_{N+1} = e_{\hat{M}+1} - s_{\hat{M}+1} + s_{N+1}$, and
- a transposed repetition with tempo variation implies that $p_{N+1} = p_{\hat{M}+1} - p_{\hat{M}} + p_N$, $s_{N+1} = \gamma(s_{\hat{M}+1} - s_{\hat{M}} + s_N)$ and $e_{N+1} = \gamma(e_{\hat{M}+1} - s_{\hat{M}+1} + s_{N+1})$, where $\gamma$ is calculated as shown in Figure 3.3.

There is a chance that none of the assumptions for finding repetitions will apply to the given array of notes. This is inevitable at some points – an excerpt containing only one note will never have any detectable repetitions, for example – so it is important to deal with it. In this case, the algorithm will yield a guess that is simply a copy of the last note, shifted in time so that it onsets at the original note's offset. This is based on the heuristic assumption that, unless there is some evidence showing otherwise, the most likely continuation for a note is the note itself.

The whole forecasting procedure is shown as a pseudo-code in Figure 3.4. There are three loops, each performing a linear search for a repetition ending at candidate notes $M$. The counter $k$ shows the length for which the note subsequence is equivalent to the subsequence ending in note $N$. The final value of $k$ for each candidate subsequence is stored in the $\boldsymbol{d}$ array. If no suitable repetitions are found within a loop, the algorithm proceeds to the next attempt, which will be built upon a more general similarity rule. If a repetition is found, then note $N+1$ is yielded.

There are more elegant and efficient ways to implement the algorithm shown in Figure 3.4, but the chosen form facilitates its visualization and interpretation.

In the next section, the accuracy of the algorithm is demonstrated experimentally, including a comparison with another note forecasting algorithm and with two systems for automatic music transcription.

## 3.4 Experiments and Results

This section shows three experiments. The first one aims at demonstrating the importance of each one of the repetition assumptions (explicit, transposed and with a tempo variation) in the forecasting process. The second one shows that the proposed algorithm outperforms a previous approach [98] when the number of correctly estimated notes is considered. The third experiment compares the proposed forecasting algorithm to two different state-of-the-art automatic transcription algorithms [65, 88], over two different databases, showing under what conditions the forecasting approach is valid for semi-automatic transcription.

The statistical analysis of the experiments is done by means of Fisher's randomization test, as recommended by Smucker *et al.* [115]. This is a non-parametric test that evaluates the probability $P$ that randomly using either of the two compared systems to perform each test would give better results than intentionally choosing the one with better average results. When this probability is too low (*e. g.*, $P < 0.05$), that affirmation, called null hypothesis, is rejected and both systems can be said to yield significantly different results. The tests are performed with 1901 permutations, which means that $P$ is calculated with an error of 10% [115].

### 3.4.1 Forecasting

The experiments described in this section aim to show the basic behavior of the forecasting algorithm, that is, how well it is capable of forecasting symbols. Exploring different composition styles, three different datasets (*Beatles*, *Bass* and *Piano*) were considered.

The *Beatles* dataset was built by downloading, from the MIDI Database [116], all files from the Beatles collection, and considering each MIDI track (hence, each instrumental line) separately. Drum tracks were manually excluded, resulting in a total of 722 different tracks.

```
procedure FORECAST($N, \boldsymbol{s}, \boldsymbol{e}, \boldsymbol{p}, \alpha, \beta$)
    $\boldsymbol{d} \leftarrow$ array of $N$ zeroes
    for $M = N - 1$ to 2 do                                         ▷ Search for exact repetitions
        $k \leftarrow 0$
        while $M - k > 0$ and EXPLICIT($M - k, N - k, s, e, p, \alpha, \beta$) do
            $k \leftarrow k + 1$
        $\boldsymbol{d}_M \leftarrow k$
        $M \leftarrow M - 1$
    if MAX(D) $> 0$ then                                            ▷ If found
        $\hat{M} \leftarrow \arg\max D$
        $p_{N+1} = p_{\hat{M}+1}$,
        $s_{N+1} = s_{\hat{M}+1} - s_m + s_N$
        $e_{N+1} = e_{\hat{M}+1} - s_{\hat{M}+1} + s_{N+1}$
        return                                                      ▷ Stop condition
    for $M = N - 1$ to 2 do                                         ▷ Search for transposed repetitions
        $k \leftarrow 0$
        while $M - k > 0$ and TRANSPOSED($M - k, N - k, s, e, p, \alpha, \beta$) do
            $k \leftarrow k + 1$
        $\boldsymbol{d}_M \leftarrow k$
        $M \leftarrow M - 1$
    if MAX(D) $> 0$ then                                            ▷ If found
        $\hat{M} \leftarrow \arg\max D$
        $p_{N+1} \leftarrow p_{\hat{M}+1} - p_{\hat{M}} + p_N$
        $s_{N+1} \leftarrow s_{\hat{M}+1} - s_{\hat{M}} + s_N$
        $e_{N+1} \leftarrow e_{\hat{M}+1} - s_{\hat{M}+1} + s_{N+1}$
        return                                                      ▷ Stop condition
    for $M = N - 1$ to 2 do                                         ▷ Search for tempo variations
        $k \leftarrow 0$
        $\gamma \leftarrow (e_N - s_N)/(e_M - s_M)$
        while $M - k > 0$ and DIFTEMPO($M - k, N - k, s, e, p, \alpha, \beta, \gamma$) do
            $k \leftarrow k + 1$
        $\boldsymbol{d}_M \leftarrow k$
        $M \leftarrow M - 1$
    if MAX(D) $> 0$ then                                            ▷ If found
        $\hat{M} \leftarrow \arg\max \boldsymbol{d}$
        $p_{N+1} \leftarrow p_{\hat{M}+1} - p_{\hat{M}} + p_N$,
        $s_{N+1} \leftarrow \gamma(s_{\hat{M}+1} - s_{\hat{M}} + s_N)$
        $e_{N+1} \leftarrow \gamma(e_{\hat{M}+1} - s_{\hat{M}+1} + s_{N+1})$
        return                                                      ▷ Stop condition
    $p_{N+1} \leftarrow p_N$
    $s_{N+1} \leftarrow e_N$
    $e_{N+1} \leftarrow e_N - s_N + s_{N+1}$
```

Figure 3.4: Pseudo-code for note forecasting.

This set was used as being representative of contemporary Western Popular music.

The *Bass* dataset was built by extracting the MIDI tracks corresponding to the bass lines in the RWC-POP database [86]. This is a special case for note forecasting, because bass lines in Western Popular music are typically repetitive, which means the forecasting algorithm is supposed to work better. The RWC-POP database was used instead of the MIDI Database because it also includes acoustic recordings related to the MIDI files, which contain voice and other aspects that are typically found in popular music, hence a comparative test considering a state-of-the-art bass line transcriber designed for this musical genre [88] may be performed, as it will be discussed in Section 3.4.3. This dataset consists of 68 tracks.

Finally, the *Piano* dataset was built using the files used by Poliner and Elis [32] in the evaluation of their music transcriber. This dataset was selected as representative of classical piano music. Due to style differences between classical piano and popular genres, it was expected that the algorithm will present a worse performance in this case.

In these experiments, the values for maximum inter-onset-intervals (IOI) and maximum duration deviations ($\alpha$ and $\beta$ in Section 3.3) were both set to 100 ms, but the forecasting system was found not to be very sensitive to these parameters, as long as they lie within a reasonable range.

The forecasting experiment was performed by providing the algorithm with the $N$ first notes of a track. The yielded result is considered correct if, when compared to the $(N+1)$-th note from the track, there is no pitch difference and the IOI and duration deviations are respectively smaller than $\alpha$ and $\beta$. This deviation means that the predicted note can be inserted into the score and the prediction process can continue autonomously. The experiment is repeated for each track, with $N$ varying from 1 to the number of notes in the track minus one. The ratio of notes correctly predicted for each track is stored, and the mean and standard deviation of the performances for each dataset is reported. This evaluation process does not consider the number of notes in each track; each track is statistically interpreted as a random sample of a musical idea.

It can be expected that the forecasting algorithm will work better when more evident repetitions can be found. To evaluate this behavior, each track was divided into four segments of equal duration. The forecasting experiment for each track was first performed considering only the initial segment, in an experiment labeled *Q1*. Then, the two first segments were used, for experiment *Q2*, then the three first segments (experiment *Q3*) and, finally, the whole track (experiment *Q4*).

The results for these experiments are shown in Figure 3.5. It can be seen that the performance for both popular music datasets (*Beatles* and *Bass*) presents a similar behavior: the forecasting accuracy increases significantly with time. The performance for the *Piano* dataset, on the other hand, remains constant for the whole track.

The error bars in Figure 3.5 show the standard deviation of the results within each dataset. For the Bass dataset, the standard deviation presents a decreasing trend when more data is analyzed. Since the analyzed bass lines are, as discussed before, repetitive, this result means that the forecasting system is working. For the Piano dataset, the standard deviation remains

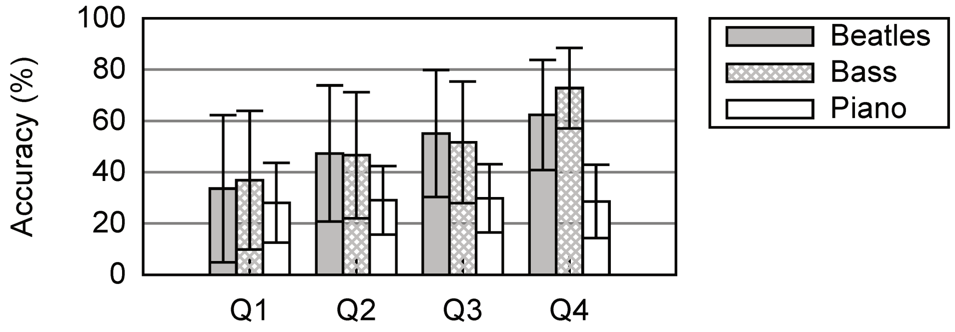

Figure 3.5: Experiment results (mean and standard deviation) for note forecasting.

constant, which also points to a constant rate of musical variations along the pieces. Last, the highest standard deviations are observed for the Beatles dataset. It can be observed, qualitatively, that this is caused by the presence of different instrument tracks in the dataset, some of which – like the bass lines – are more repetitive than others – like a trombone that only plays a short phrase.

### 3.4.2 Forecasting comparison

Elements of comparison with another forecasting method were obtained by applying the method proposed by Assayag *et al.* [98] to the same forecasting experiment described in Section 3.4.1, using the full length pieces. This method [98] consists of building a variable-order Markov model using all data in a set. Because building the model is computationally expensive, the model order was limited to 30. The model was required to yield the most likely continuation for each given note sequence.

The results, shown in Figure 3.6, were compared to those yielded by the proposed method using a randomization test [117] for the mean and the standard deviation-to-mean ratio, so a general idea of the accuracy and the typical performance deviation are informed. The P-Values obtained in these tests are shown in Table 3.1 (a P-value lower than 0.05 indicates a significant difference).

Table 3.1: P-Values comparing the result yielded by the proposed method and the method proposed by Assayag *et. al* [98]

| Dataset | Mean ($\mu$) | $\sigma/\mu$ |
|---|---|---|
| Beatles | 0.00 | 0.00 |
| Bass | 0.00 | 0.00 |
| Piano | 0.00 | 0.21 |

The results show that the proposed method consistently yields a higher mean result, with all P-Values equal to 0.0. Also, except for the Piano dataset, the proposed method yields

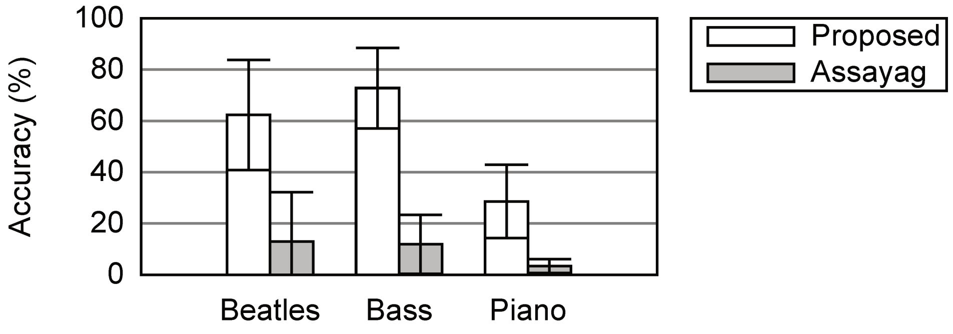

Figure 3.6: Comparative results (mean and standard deviation) for note forecasting.

a significantly lower standard deviation-to-mean ratio, which means its performance is more consistent from one piece to another. This can be explained by the high amount of variations present in Classical piano music, which makes it harder to forecast continuations using a single mathematical concept. On the other hand, it is clear that popular music may be better predicted by the proposed method.

This shows that the proposed deterministic approach proposed works better than the probabilistic approach proposed by Assayag *et. al* [98] in the sense of forecasting the continuation of a piece as intended by the author. However, the aesthetic appeal of the probabilistic continuation may be higher, since it may contain more variations and interesting outcomes.

### 3.4.3 Semi-Automatic Transcription

The usual computer-assisted method for obtaining a documental register of a particular musical piece relies on Automatic Music Transcription (AMT) systems. These systems generate a transcription that will have to be manually corrected, as they will generally have some kind of error.

To obtain a documental register, the user will be required to correct and delete notes yielded by the AMT system. The number of insertions and deletions may be calculated from the Recall and Precision (defined in Expressions 2.12 and 2.13) as:

$$\begin{aligned} \text{Insertions} =&\ (1-R)\times\text{Total number of notes} \\ \text{Deletions} =&\ (1-P)\times\text{Total number of notes} \\ \text{Total} =&\ \text{Insertions}+\text{Deletions}. \end{aligned} \tag{3.1}$$

In practice, it is observed that Recall and Precision tend to be similar, and yielded notes are often partially correct, which saves work from the user. Estimating an initial guess for a note was considered as more difficult than correcting a wrong not, hence, the effort to be made by the user was estimated as:

$$\text{Operations} = (1 - \text{F-Measure}) \times \text{Total number of notes}, \tag{3.2}$$

The proposed approach, called Interactive Computer-Assisted Music Transcription (ICAMT), relies on the interaction between a human user and a prediction system. In this case, the user provides some information – the first notes of a sequence – and the prediction system uses it to yield a possible continuation. The continuation will be manually accepted or rejected. If it is accepted, the system yields a new prediction, and if it is rejected the user will be requested to input the correct note. This means that the user will have to modify all notes that the system is unable to forecast correctly, that is:

$$\text{Operations} = (1 - \frac{\text{Correct notes}}{\text{Total number of notes}}) \times \text{Total number of notes}. \tag{3.3}$$

The following experiments show that employing ICAMT may, in some cases, require less manual labor than using state-of-the-art AMT systems. In the experiments, both the AMT systems and the forecasting system described in Section 3.4.1 were executed over a database. The experiments compare the F-measure associated with the AMT system with the forecast accuracy related to the ICAMT system, with the aim of detecting which will require a less pronounced correcting effort from the user. For this purpose, the tests with ICAMT used a simulated computer environment in which the human user was considered to give correct notes up to a particular point in each piece.

Two different cases were analyzed. The first one is the transcription of the bass line in popular music, based on the system proposed by Ryynanen and Klapuri [88]. The second one is the transcription of the piano in classical music, based on the system proposed by Dessein *et al.* [65]. The database, the evaluation criteria and the results related to each experiment are described below.

### Bass line in popular music

The transcription of the bass line in popular music is defined by Ryynanen and Klapuri [88] as detecting the onset and pitch of notes played by a bass, double bass or electric bass in the context of popular pieces. For the purposes of this experiment, the authors own implementation of the original system was executed over 45 second excerpts from the RWC-POP database [86]. A MIDI score was manually synchronized to each audio file in order to obtain a reliable ground truth. Pieces that did not have a bass line or that could not be synchronized due to tempo differences were not used. A total of 68 pieces were considered. Since the method proposed by Ryynanen and Klapuri [88] relies on training over a dataset, it was executed using a two-fold cross validation scheme.

In the Ryynanen and Klapuri's system, a note is considered correct if its pitch matches that of the ground truth and if its onset is within a 150 ms range from the onset in the ground truth [88].

When executing the forecasting system, however, a note can only be considered correct if its pitch matches that of the ground truth and if its onset and offset are within a 100 ms range of the onset and offset in the ground truth. The offset evaluation is needed because a correct offset is one of the requirements for forecasting without further user corrections, as discussed before.

The forecasting system tends to improve its performance when longer pieces are analyzed, as the chance of finding repetitions increase accordingly. Therefore, the ICAMT system was also tested with the full length ground-truth annotations for the analyzed pieces. This behaviour is not observed in AMT systems, which tend to keep a stable performance throughout the whole piece. This allows the use of excerpts, instead of whole pieces, for the evaluation of AMT systems, causing the testing process to be faster without harming the results.

The mean and variance of the results (F-Measure or forecast accuracy) are shown in Table 3.2. The reported P-values were calculated using Fisher's randomization test comparing the mean ($\mu$) and the standard deviation to mean ratio ($\sigma/\mu$) of the results yielded by AMT and ICAMT. A P-value lower than 0.05 indicates a significant difference.

Table 3.2: Experiment results (%) for the popular music bass line transcription.

| Experiment | Mean ($\mu$) | $\sigma/\mu$ |
|---|---|---|
| AMT (excerpts) | 48.83 | 38.33 |
| ICAMT (excerpts) | 52.58 (P=0.35) | 41.80 (P=0.54) |
| ICAMT (full) | 72.76 (P=0.00) | 21.43 (P=0.00) |

When the excerpts are used, the performance of the ICAMT is similar to the performance of the AMT. However, when the full length annotations are considered, the ICAMT performs significantly better than the AMT.

Also, the ICAMT system presents a significantly lower standard deviation to mean ratio when considering full pieces. This means that the performance for a particular piece tends to be closer to the average performance, which is a desirable characteristic in information retrieval systems.

The observed result is due to the specific characteristics of bass lines in popular music. They are known to be repetitive, which allows the forecasting algorithm to work well. As it will be seen, results are different when considering classical music.

**Piano in classical music**

For the following experiments, the automatic transcriber proposed by Dessein [65] was implemented. The system relies on supervised training using sound samples of each note of a piano, which were obtained from the IOWA musical instrument samples database. The pieces used in this experiment were the ones in the test dataset described by Poliner and Elis [32]. They were downloaded from the MIDI Database [116] and rendered using the same note samples used for training the AMT system, which emulates a best-case scenario.

Two different evaluations for the AMT system were performed. In the first one, a note is considered correct if its onset is within 50 ms of the onset of a note in the ground truth and its pitch is equal to the pitch in the ground truth. In the second evaluation, it is also required that a note have a duration within 20% of the duration of the ground truth note in order to be considered correct. These criteria were used in the 2010 edition of the Music Information Retrieval Exchange (MIREX 2010) [81].

The criteria for considering a note in the ICAMT test, however, were kept the same (no tolerance for pitch deviations, 100 ms tolerance for IOIs and durations) for the same reasons stated before.

Again, the mean and variance of the results are calculated and shown in Table 3.3. The reported P-values were calculated using Fisher's randomization test, comparing the mean ($\mu$) and the standard deviation to mean ratio ($\sigma/\mu$) of the AMT and ICAMT results.

Table 3.3: Experiment results (%) for the classical piano music transcription.

| Experiment | Mean ($\mu$) | $\sigma/\mu$ |
|---|---|---|
| AMT (onset only) | 76.32 (P=0.0) | 13.16 (P=0.003) |
| AMT (onset+offset) | 32.36 (P=0.42) | 71.87 (P=0.08) |
| ICAMT | 28.61 | 48.88 |

It can be seen that the AMT system is significantly more effective on detecting onsets than the ICAMT. However, when offsets are considered, no significant difference between the systems is observed.

The results for the bass in Popular music are visibly different from those in Classical music. In the first case, more explicit note repetitions were used, but the analyzed signals tend to present a great amount of different timbres and digital effects. Therefore, working directly with incomplete symbolic representation of the pieces may be more effective than trying to analyze signals. In the case of classical music, the musical ideas tend to be richer, but the only timbre present is the piano that will be transcribed. Hence, the information derived from the signal gives a more accurate information on musical notes than the symbolic information itself. This leads to the conclusion that ICAMT will perform better than AMT in musical pieces where the note progressions are more predictable (the symbolic representation is simpler) and more timbres are present (the signal is more complex).

## 3.5 Discussion

Experimental results show that semi-automatic transcription, that is, providing some notes to a forecast algorithm and iteratively correcting the outcomes, demands less effort from the user than correcting the results of an automatic music transcriber, when considering Western popular music. For the case of the piano in Classical music, no significant difference was observed when using the automatic and the semi-automatic approaches. This shows the importance of the semi-automatic transcription for obtaining a documental register of pieces, and points to the relevance of future work on developing a human-computer interface to explore it.

Since ICAMT does not depend on the existence of an acoustic signal, it can be used in an assisted music composition software. For this application, the user inserts musical notes in the system and the forecasting algorithm suggests continuations, which aim to save effort from the user on inserting subsequent notes. Again, by using a mobile interface, a better user experience can be delivered.

## 3.6 Conclusion

A method for forecasting note event continuations of musical pieces, based on the compression algorithm proposed by Ziv and Lempel [97], was presented. The proposed algorithm was shown to incorporate information along the piece, and to outperform another note event forecasting method.

When a documental transcription is the target, the interactive transcription algorithm is shown to outperform automatic algorithms. In the context of classical piano music, however, it is not as effective. This happens because of style differences between contemporary popular music and classical pieces, which employ different composition techniques.

Regardless of the analyzed piece, the presented algorithm cannot be used in an entirely automatic context. Although the algorithm can still be used for documentation and musical composition purposes, its application on content retrieval for big databases is limited. Entirely automatic algorithms, incorporating the same concept – repetition – will be discussed in the next chapter.

Chapter 4

# Polyphonic piano pitch tracking with explicit models for periodicity

As it was discussed in Chapter 2, a common approach for the detection of simultaneous musical notes in an acoustic recording involves defining a function that yields activation levels for each candidate musical note over time. These levels tend to be high when the note is active and low when it is not. Therefore, by applying a simple thresholding decision process, it is possible to decide whether each note is active or not at a given time.

Thresholding is a one-dimensional classification that consists of assigning an element to either group 1 if its value is higher than a threshold $\alpha$ or to group 0 otherwise. This method is used to separate significant source activities from noise, such as thresholding the output of a detector to avoid false positives.

However, there has not been, so far, a systematic discussion on how to find a suitable value for this threshold. In general, it is set manually or using supervised training. Using a fixed threshold ensures that the decision system can be executed in real time, but, at the same time, it may yield sub-optimal results.

This chapter discusses an unsupervised method for finding an optimal threshold for note detection. The proposed method, which has already been published [2], assumes that a piece of music probably presents some degree of pulse, thus the detected events should be organized as to resemble time-wise periodicities. Therefore, the desired threshold should make the detected events maximally periodic.

The performed experiments use as baseline the state-of-the-art note tracking algorithm proposed by Dessein *et al.* [65]. The method uses a pre-defined set of basis vectors $\boldsymbol{B}$, each consisting of a prototype spectrogram of a different note, to obtain an activation matrix $\boldsymbol{A}$ from a spectrogram $\boldsymbol{X}$. This is achieved by minimizing the divergence:

$$\epsilon_{\beta}(\boldsymbol{X}|\boldsymbol{BA}), \tag{4.1}$$

where $\epsilon_\beta$ is defined in Expression 2.10 and $\beta = 0.5$. Using a simple thresholding approach, active and inactive notes are found in $\boldsymbol{A}$.

This chapter is organized as follows. In Section 4.1, the proposed method for finding optimal thresholds is discussed. The experimental setup, the results and other discussions are shown in Section 4.3. Conclusive remarks are stated in Section 4.4.

## 4.1 Proposed method

Periodic signals are those that may be described by the expression $x(t) = x(t-\tau)$, where $\tau$ is the fundamental period of $x(t)$. In many applications, however, signals have only an approximate periodic behavior, due to either noise or natural dynamics of the system. For this reason, it is important to discuss the concept of *periodicity level*.

The concept of periodicity level is employed by many methods for fundamental frequency estimation in audio. Two emblematic cases are the Yin method [118], which consists of estimating a value of $\tau$ that minimizes $\|x(t) - x(t-\tau)\|^2$, and the multiple fundamental frequency estimation method developed by Klapuri [119], which consists of calculating the weighted sum of Discrete Fourier Transform coefficients to detect which fundamental frequency candidates are more prominent. In both cases, a definition for the periodicity level is built based on either time-domain or frequency-domain properties of the analyzed signals.

The method discussed in this section aims at finding the threshold $\alpha$ that, when applied to the activation matrix $\boldsymbol{A}$, yields a binary detection matrix $\boldsymbol{D}$ that presents the greatest periodicity. The periodicity of $\boldsymbol{D}$ is related to the timewise organization of the detected events, and the exact nature of such events may be ignored for this purpose. To represent these structures, the vector $c_j = \sum_{\forall i} d_{i,j}$ is calculated, and $c_j$ is the number of active events in the $j$-th signal frame.

True positives are expected to be organized in a more periodic structure than false positives, because wrong notes are more likely to have random onset times. The periodicity level is closely related to how much of the signal can be described using a Fourier series. It is calculated using the spectrum of $\boldsymbol{c}$, as follows.

The spectral representation $\boldsymbol{\gamma}$ of $\boldsymbol{c}$ is calculated by subtracting the mean value of $\boldsymbol{c}$ to avoid the DC component, multiplying it by a Hanning window to reduce spectral leakage, and then calculating its DFT. Both magnitude and power representations for $\boldsymbol{\gamma}$ were tested in order to determine which is more representative. The prominence of the Fourier series that can describe the spectrum is given by its Harmonic Sum Spectrum (HSS, $\gamma_H$), calculated as follows:

1. $\boldsymbol{\gamma}_H \leftarrow \boldsymbol{\gamma}$,

2. $w \leftarrow 2$,

3. $u[k] \leftarrow \max_{k \in [wk,(w+1)k]} \gamma[k]$,

4. $\boldsymbol{\gamma}_H \leftarrow \boldsymbol{\gamma}_H + \boldsymbol{u}$,

5. $w \leftarrow w + 1$,

6. If $w < 9$, go back to step 3. Else end.

The maximum value of the HSS corresponds to the prominence of the strongest Fourier series that can be used to describe $\boldsymbol{c}$. Thus, the ratio

$$y = \max \boldsymbol{\gamma}_H / (\sum_{\forall j} \gamma_j) \tag{4.2}$$

represents the periodicity level of $\boldsymbol{c}$, that is, an objective measure of how periodic is $\boldsymbol{c}$. It is necessary to define a search algorithm for the threshold $\alpha$, as it impacts directly in $y$.

Calculating the periodicity $y$ is an $O(Q \log Q)$-complexity operation, which is considerably faster than calculating the activation matrix $\boldsymbol{A}$, an $O(Q^3)$ operation. For this reason, the search for $\alpha$ may be conducted using exhaustive search, in a range in which the optimal threshold is likely to be found. In this work, the search was carried out between the mean and the maximum values of $\boldsymbol{A}$.

Experiments showing how the proposed method behaves in real applications will be described in the next section.

## 4.2 Experiments and Results

As stated before, the experiments performed in this section were based on the automatic piano transcriber proposed by Dessein *et al.* [65]. This transcriber calculates the linear approximation $\boldsymbol{X} \approx \boldsymbol{BA}$ by first obtaining the prototype vectors $\boldsymbol{B}$ using supervised training with pre-recorded samples, and then calculating $\boldsymbol{A}$ minimizing the beta-divergence between the measured spectrogram $\boldsymbol{X}$ and the approximation $\boldsymbol{BA}$. The exact details of this method are not essential here, as could be employed any other approach that uses the activation matrix as an intermediate step.

The database used in the following experiments consisted of 24 pieces for piano solo used for testing purposes by Polliner and Ellis [120]. This database is broadly used and freely available for download [116]. The samples for obtaining $\boldsymbol{B}$ were downloaded from the Iowa University Musical Instrument Samples database [121]. Audio files were rendered from the MIDI files using the Iowa University samples and were labeled as the IOWA dataset. These audio files were later processed by adding a digital chorus effect, generating the CHORUS set, which aims at simulating an analysis over a different timbre.

In all experiments, the performance measurements, as adopted in MIREX, were calculated as follows. A note is considered correct if its pitch matches (has the same MIDI number) and its onset is within 50 ms of the onset of a note in the ground truth. This allows calculating Recall, Precision and F-Measure as defined in Chapter 2.

The first test aimed at evaluating the correlation between the F-Measure and the periodicity level. For this purpose, the $\boldsymbol{A}$ matrix of a piece selected at random from the IOWA set was calculated, and the detection threshold was progressively increased. For each threshold value, the performance measurements and the periodicity level were measured, yielding the values seen in Figure 4.1.

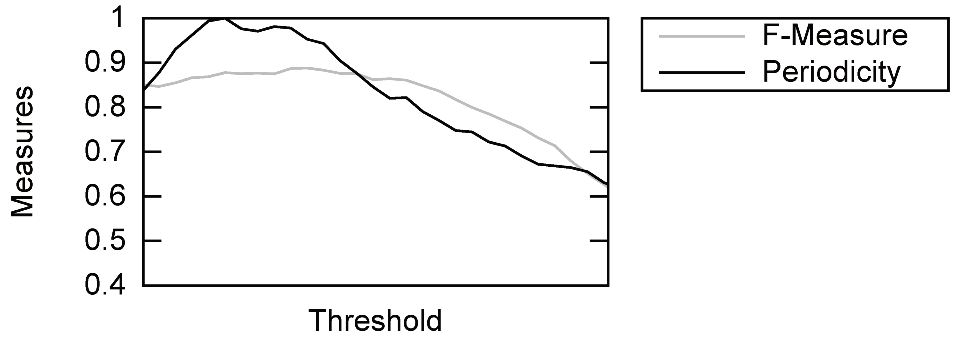

Figure 4.1: Periodicity of a detection matrix $\boldsymbol{D}$ and the related F-Measures, for different threshold levels considering a single piece. The periodicity value was normalized to provide a better visualization.

It can be seen that the F-Measure curve presents a close-to-maximum value when the periodicity value is at its the global maximum. The shape of the periodicity curve is reasonably similar to the shape of the F-Measure curve, which indicates that it can be useful in finding an optimal threshold. This hypothesis was assessed by means of tests over the whole database.

These tests were performed as follows. First, using the IOWA set, the value for $\alpha$ that yields the greatest F-Measure was calculated by exhaustive search. This is the value that maximizes the performance of the system when using the same fixed threshold for all the dataset, considering the best case scenario. After that, new thresholds were obtained considering the periodicity maximization, using both the magnitude and the power spectrum representations for $\boldsymbol{\gamma}$. The mean value[1] of each performance measure, considering each one of these methods, is shown in Figure 4.2.

As can be seen, the performance measurements, when using the proposed approach, have only a small difference in comparison with the supervised training (fixed threshold) approach. This result is important, since the periodicity approach is unsupervised, and the optimal value for the fixed threshold can only be obtained if the ground-truth result is known *a priori*.

The final test analyzed the effects of using the same threshold for different datasets. For this purpose, two thresholds were used to detect notes in the CHORUS dataset: a fixed threshold

[1]The confidence intervals were calculated using the expression $\sigma/\sqrt{N}$, where $\sigma$ is the standard deviation of the measure over the dataset and $N$ is the number of pieces in the dataset.

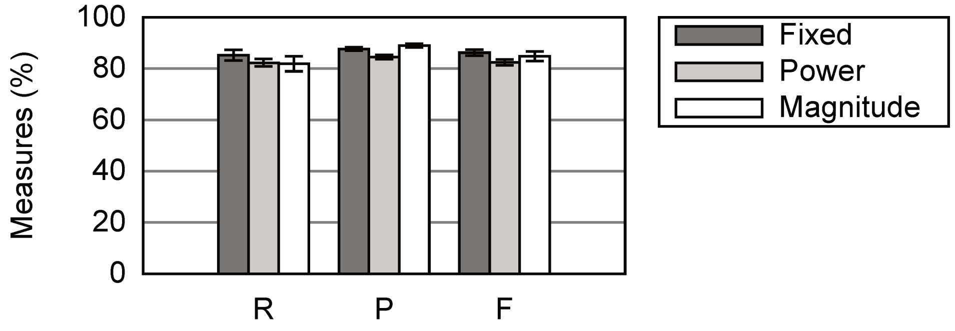

Figure 4.2: Recall (R), Precision (P) and F-Measure (F) obtained for the IOWA database when using a fixed threshold and using the periodicity approach, considering both the power and the magnitude spectral representations.

that maximizes the average F-Measure over the CHORUS dataset, and a trained threshold, which was obtained by maximizing the average F-Measure over the IOWA dataset. The results obtained using both thresholds were compared to those obtained using the periodicity method, as shown in Figure 4.3.

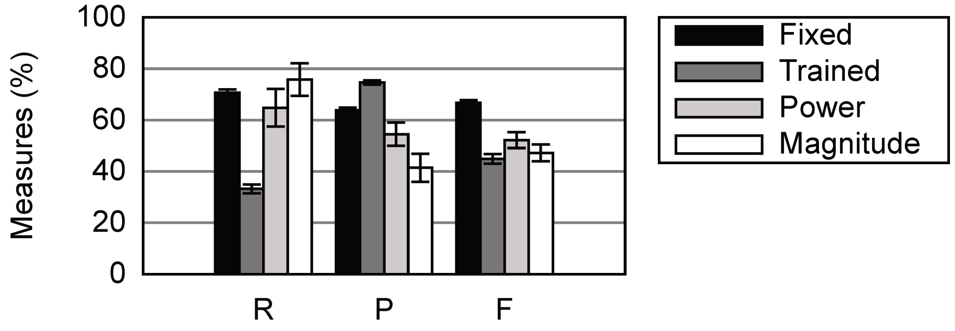

Figure 4.3: Recall (R), Precision (P) and F-Measure (F) obtained for the CHORUS database when using a fixed threshold, a threshold trained in the IOWA database (Trained) and using the periodicity approach, considering both the power and the magnitude spectral representations.

The comparison shows that using the trained threshold for a different database significantly compromised the F-Measure of the system. This is a consequence of the significant decrease in the Recall, which could not be balanced by the increase in the Precision. Thus, the optimal detection threshold tends to be different for different databases.

These differences in Recall and Precision were smaller when using the periodicity with a power spectrum representation. As a consequence, a slightly higher average F-Measure was observed. This shows that, although the periodicity cannot yield an optimal average result, it is more robust to variations in the database than a fixed threshold.

Periodicity has shown to be effective for finding a threshold that give a good average result over a database, especially when using the power spectrum representation. Also, it is obtained without the need for ground truth or manual setup, and is flexible towards different databases, which are desirable characteristics for music transcription applications.

## 4.3 Discussion

The presented method was tested using the note tracker proposed by Dessein *et al.* [65], chosen because of its simplicity. Nevertheless, it is applicable to any note tracking algorithm that uses an activation matrix $\boldsymbol{A}$, where $a_{i,j}$ has high values if the $i$-th note is active in the $j$-th frame and low values otherwise. This is a common step in many note tracking algorithms, which means that the periodicity-based threshold optimization is highly applicable in future research.

The obtained results indicate that the periodicity of detected events can be used as a feature in the context of blind source separation (BSS) for audio signals, the same way sparsity constraints are commonly used. The task, then, would consist of detecting a base $\boldsymbol{B}$ that minimizes some divergence $d(\boldsymbol{X}|\boldsymbol{BA})$ while also maximizing the periodicity of $\boldsymbol{A}$. This is left as an idea for future work.

## 4.4 Conclusion

This chapter discussed an algorithm for determining the detection threshold of note-tracking systems. The presented method uses an important aspect of music, the periodicity of events, to search for an optimal threshold by means of unsupervised learning.

By using the proposed method, it is possible to run transcription algorithms over different databases, without requiring a training corpus for each one of them. As it has been shown, the unsupervised training outperforms the approach in which a database is used for training and another one is used for testing. This shows that periodicity is an important feature to consider in note-tracking algorithms.

The discussion held in chapters 3 and 4 considers how models for rhythm can be used for the detection of events in existing audio files. These same concepts, however, can be used for music production, if similar algorithms are delivered in the form of musical interfaces. This possibility, and its implications, will be discussed in the next chapter.

Chapter 5

# Interfaces for Musical Expression Using Explicit Models for Repetitions

By incorporating domain-specific models into music transcription algorithms, more accurate hypotheses about the contents of a piece may be derived from the observation of the symbolic context. This means that musical events tend to be correlated to other nearby events. This characteristic was exploited in Chapter 3 for the purpose of transcribing musical notes, and this chapter will discuss applications of the same idea in the context of musical production.

The domain-specific knowledge that was applied in these algorithms came from direct interaction with music composers. This interaction allowed defining meaningful characteristics of the compositional process employed by the artist. In the two interfaces described in this chapter, the concept of repetitions is used in order to expand the range of possible ways the musician could make music.

Unlike the case of automatic transcription, the development of new musical interfaces was not evaluated using a ground truth. Instead, it was necessary to define how the newly developed tools could provide a novel way for musical expression. Also, the limitations of each tool were qualitatively analyzed and their practical effects were considered.

This chapter is organized as follows. In Section 5.1, an algorithm that allows repeating control data is described. Section 5.2 describes a drum sequencer that can be controlled by the natural playing of drums. Section 5.3 brings some discussions, and the chapter is concluded in Section 5.4.

## 5.1 Repeating control data

In the context of electronic music, it is common to use effect parameters as part of the musical composition. For example, a musician may choose to increase and decrease the amount of digital reverberation to create variations on the feeling of distance within a piece. These parameters, known as control data, may be either generated automatically, by an algorithm, or acquired from sensors, such as faders, potentiometers, light-dependent resistors, accelerometers or other electronic devices.

One problem that arises when using this approach is that the continuous variation of effect parameters demands considerable attention from the musician, harming the on-stage construction of a complex sonority. However, if control data can be predicted, then the musician may focus his attention on other aspects of the piece. This is the purpose of the tool, which has already been published [3], described in this chapter.

The tool is an implementation of a method capable of forecasting the continuation of a given control data array $x[n]$, where $x[n]$ is the value of the $n$-th reading of a particular sensor. The forecasting algorithm does not require any training on previous templates, which means that the musician has freedom to improvise and generate creative soundscapes and define the appropriate gestures while performing. The key idea is that the control data will be continued, extending repetitive variations.

When forecasting, the system provides data which aims to be equivalent to the control data that would be yielded if the musician continued interacting with the sensors. Thus, the performer may continue playing other parts of the piece while still getting the soundscape variations derived from that interaction.

The forecasting method is based on evaluating what time lag, in samples, is most likely to be the fundamental period of the control data. This means that, although the motion may be freely composed and improvised, only repetitive motions can be predicted by the system. The method begins by estimating the autocorrelation of the last $N$ input samples, which is defined as:

$$r_x[k] = \frac{2}{N} \sum_{n=0}^{\frac{N}{2}-1} x[n]x[n-k]. \tag{5.1}$$

An autocorrelation signal presents peaks at positions $k$ that correspond to the time lags to which $x[n]$ is most self-similar. However, there are some other peaks, especially at the zero time lag $k = 0$ that must be filtered out. In order to do that, a signal $r'_x[k]$ is calculated by upsampling $r_x[k]$ by a factor of two and subtracting it from the original signal, that is:

$$r'_x[k] = r_x[k] - r_x[left\lfloor k/2 \rfloor]. \tag{5.2}$$

The value $j = \arg\max r'_x[k]$ is obtained by a simple linear search. The forecasting, then, proceeds by yielding an estimate $\hat{x}[N+1] = x[N-j+1]$. The quality of the estimate may be evaluated by the ratio $r_x[j]/r_x[0]$, which will present values closer to 1 when the signal is significantly periodic.

It is important to note that the calculation of Expression 5.1 depends on storing the last $N$ data samples obtained from the sensor in an internal buffer. The size of that buffer must be chosen so that $N$ samples can be used to store at least two repetitions of the control data. At the same time, the time required for the algorithm to detected any periodicity is proportional to the square of the size of the buffer. In preliminary experiments, it was found that a good rule of thumb is configuring the buffer size to 3 times the duration of the longest expected interaction, which will give a considerable margin for timing variations.

### 5.1.1 Implementation issues and usability

The proposed forecasting method was implemented as a patch for the Max/MSP digital music processing environment [122], so that it could be used together with other tools for musical creation. The forecasting system works in three different modes: *learning*, *predicting* and *bypass*. The *learning* mode should be manually triggered when a new control sequence is to be acquired. While in this mode, the internal buffer is updated and the forecasting algorithm is executed at each new sample. Since *learning* mode assumes that the sequence is not yet learned, the system does not yield predicted data.

When the *prediction* mode is triggered, the system starts yielding forecasts, adding them to the internal buffer as if they were received as inputs. While in this mode, the system does not re-execute the forecasting algorithm, sticking with a single value for $j$ during the whole process.

The operation of *bypass* mode is to simply bypass input data to the output, while ignoring any operations regarding processing. Hence, if the *prediction* mode is triggered while the system is in *bypass* mode, the system will recall the last learned pattern. When operating in the *bypass* mode, the system preserves the internal buffer, hence it may be used again by triggering the *prediction* mode.

### 5.1.2 Experiments and performances

The internal operation of the system is visualized in Figure 5.1, which shows the input (bypassed) data from an one-dimensional sensor and the predicted data. The figure was generated while acquiring manually-driven data from the sensor . In the first five seconds, the system is in bypass mode, and the prediction system has received any samples yet. When the learning mode is triggered, the predicted data quickly synchronizes with the repetitive sensor data received as input. When the prediction mode is triggered, the forecasting system continues the learned gesture and ignores the sensor data.

After implementation, the proposed system was used in the piece *Mane Havn (mounhoun): An Exploration of Gestural Language for Pitched Percussion*, by Shawn Trail, Thor Kell and Gabrielle Odowichuk. This piece uses a vibraphone augmented with a hand motion sensor (namely, a Microsoft Kinect device) to control audio effect parameters. The proposed technique allows creating an improvised soundscape comprising both the sound from the vibraphone and the effect parameter modifications.

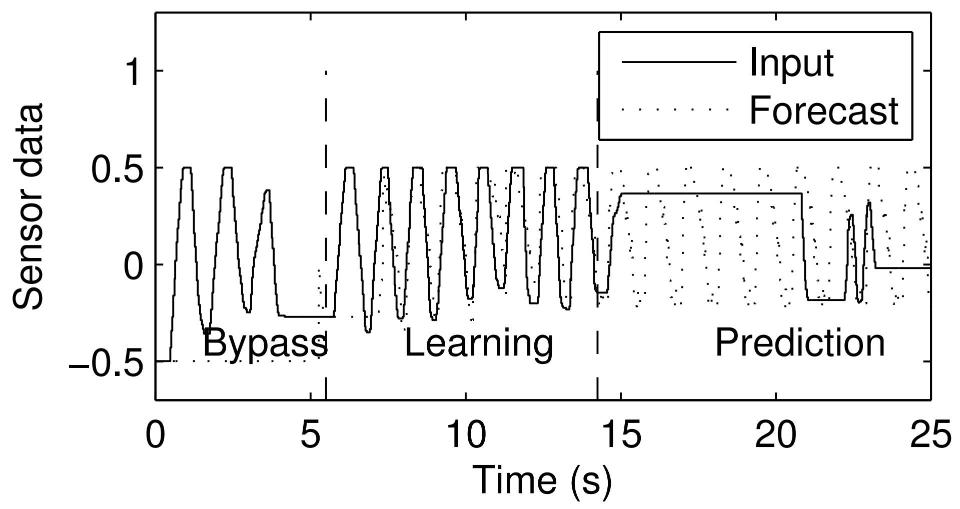

Figure 5.1: Visualization of different modes of operation for the sample forecasting system.

Also, it was used as part of a hyperinstrument called Wiikembe[1]. It is a small device which may be tilted to change its timbre or to control other transformations. By using the proposed technique, several layers of tilting motion may be employed at the same time.

The next section describes a drum looper that was built using a different theoretical ground, but serving a similar purpose.

## 5.2 Drum sequencer

In electronic music, it is often useful to build loops from discrete events, such as playing notes or triggering digital effects. This process generally requires using a visual interface, as well as pre-defining tempo and time quantization. This section describes a digital musical interface capable of looping events without using visual interfaces or explicit knowledge about tempo or time quantization. The interface [4] was built based on a prediction algorithm that detects repetitive patterns over time, allowing the construction of rhythmic layers in real-time performances. It has been used in musical performances, where it showed to be adequate in contexts that allow improvisation.

A technique that can be used to create meaningful repetitions for musical events consists of applying carefully arranged delay lines and feedback so that an audio sample is played in a pattern defined by the musician, as in RhythmDelay [123] or SDelay [124]. Using this technique, a musician can build audio loops on-the-fly, without the need for a grid interface. However, when using delay lines it is hard to change events (for example, changing all kick drums for cymbals) or to modify the musical tempo without affecting the timbres.

[1]`http://webhome.csc.uvic.ca/~trl77/wiikembe.html`

This section describes an interface that allows looping general event sequences that are played by the musician, without predefining or quantizing tempo. The instrument may be played using any interface that generates discrete events, from high-end MIDI drum interfaces to low-cost game controllers. It relies on a technique similar to the one presented in Section 5.1, with the difference that it works with discrete symbols, rather than continuous data.

The event looping process is based on an algorithm that is similar to the one for note forecasting described in Chapter 3, in the following manner. As the musician plays a sequence of events, a continuation for that is predicted. When the user triggers the automation, the system starts yielding the predicted continuations and feeding them back into itself, creating an event sequence that corresponds to continuing the pattern played previously by the musician, without any explicit inputs regarding tempo or time-quantization.

The proposed system, similar to the one described in Section 3.3, assumes that events repeat within a particular piece or excerpt. This assumption allows simple, intuitive interactions with the system, as the process of designing a new loop becomes similar to that of showing a rhythmic pattern to a human being. Hence, the musician can have great control over the outcomes of the system, as if the usual grid interface was being used.

In the context of this work, each musical event $n$ is represented by its *onset* $s_n$, that is, the time it happens, and its *label* $l_n$, which identifies the event. Using labels, events may be related to any description of discrete musical gestures desired by the musician, such as "play cymbal", "strongly play cymbal", "play random drum" or "activate reverb effect". As will be shown, the flexibility of the event label allows many creative uses of the system.

The system receives event-related messages through any device that yields discrete messages (such as MIDI instruments, OSC controllers or HID devices). The label and onset of the events are stored in an internal buffer of arbitrary size. The information in the internal buffer is used to predict the following event, as described below.

The forecasting algorithm is based on the assumption that the musician is playing a loop, a condition that has to be intentionally caused. When the user desires, the predictions may be used as inputs to the forecast system, creating a feedback loop and allowing the continuation of the event sequence. Preliminary tests showed that adding a reset functionality, which clears the internal buffer, made it easier to switch between different beats.

In the context of the prediction algorithm, two events $n$ and $m$ are considered equivalent if their label is the same ($l_n = l_m$) and their inter-onset intervals (IOIs), that is, the difference between their onset and the previous onsets, are within an allowed deviation ($\|s_n - s_{n-1} - (s_m - s_{m-1})\| \leq \alpha$). The algorithm, shown in Figure 5.2, aims at searching, within the last $N$ recorded events, the longest subsequence $[M-K+1 \ldots M]$ whose events are equivalent to those in the subsequence $[N-K+1 \ldots N]$. After the subsequence $[M-K+1 \ldots M]$ is found, it is reasonable to assume that the continuation of the recorded excerpt (that is, event $N+1$) will be equivalent to that of the subsequence (event $M+1$).

```
1: procedure FORECAST($N, \boldsymbol{s}, \boldsymbol{l}, \alpha$)
2:     $\boldsymbol{d} \leftarrow$ array of $N$ zeroes
3:     for $M = N - 1$ to 2 do                                    ▷ Search for repetitions
4:         $k \leftarrow 0$
5:         while $M - k > 0$ and $l_{N-k} = l_{M-k}$ and $|(s_{N-k} - s_{N-k-1}) - (s_{M-k} - s_{M-k-1})| < \alpha$ do
6:             $k \leftarrow k + 1$
7:         $\boldsymbol{d}_M \leftarrow k$
                                                    ▷ Check if a subsequence was found
8:     if MAX(D) $> 0$ then                                        ▷ If found
9:         $\hat{M} \leftarrow \arg\max D$
10:    else
11:        $\hat{M} \leftarrow N$
                                                                    ▷ Yield events
12:    $s_{N+1} = s_{\hat{M}+1} - s_m + s_N$
13:    $l_{N+1} = l_{\hat{M}+1}$
14:    return
```

Figure 5.2: Pseudo-code for event forecasting.

Figure 5.3 shows an example of a possible execution of the forecasting algorithm. The left column shows the internal buffer after the user has yielded a series of events, arbitrarily labeled "hit drum" and "switch reverb". After triggering event E9, the forecasting system detects that, if the buffer is delayed by three events, events E9 and E6, as well as the previous five events, are equivalent, which is a higher number of equivalent events than if any other delay was used.

In this example, event E7 is chosen as the most plausible continuation for the delayed sequence, generating the estimated E10 and is used to build the event to be yielded. If event E10 is used by the system as an input, the next event to be yielded would be a "hit drum", then a "switch reverb", then a "hit drum" again, creating a cycle of repetitions. Hence, feedback may be used to create event loops in real time.

### 5.2.1 Implementation issues

The proposed instrument was implemented as a patch for PureData [125]. This allows users to employ their favorite interfaces and sound designs, as well as their own compositional ideas. It is an open-source project, which means that the algorithm may be easily ported to other contexts.

The patch works in two modes: **observation**, or manual, and **prediction**, or automatic. In the observation mode, the system receives inputs from the user and tries to predict the next event that will be received. When the prediction mode is triggered, the system receives its own predictions, instead of user-generated events, as inputs.

Additional functionality may be easily implemented by the user, but not as part of the system itself. Several predictor instances in parallel, for example, may be used to achieve polyrhythms.

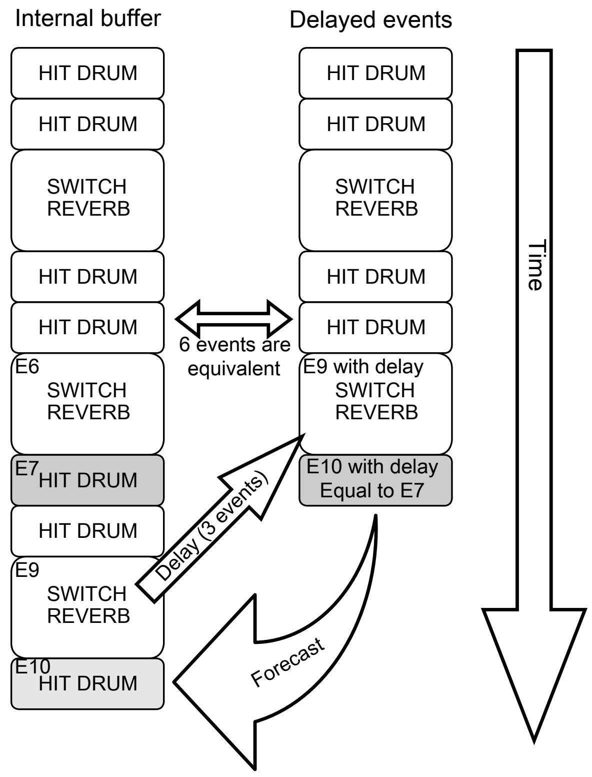

Figure 5.3: Example of a possible execution of the forecasting algorithm.

Also, the actions related to an event must be defined by the user: playing a drum sample or switching the configurations of an effect, for example.

### 5.2.2 Experimental performances

The experiments in this section aim at highlighting interesting features and also drawbacks of the proposed system, gathering information on how it can be helpful to musicians and how it may be improved. For this purpose, the system was used in a solo and an accompanied performance. The recorded audio material can be found at `http://www.dca.fee.unicamp.br/~tavares/Looper/index.html`.

The first performance – solo – aimed at showing the main capabilities of the system. The piece was intentionally composed so that a drum sequence and an effect switch were independently looped, allowing another layer of percussion to be freely played. Spectrograms are used to show the most important parts.

The second performance – duo – had the goal of showing that the drum loops could be quickly arranged and played in an arbitrary tempo. The piece was an improvisation in which digital effects were manually played while drums were looped using the proposed system. The piece was analyzed based on the impressions of musicians and on auditory characteristics of the recording.

### Solo performance

The first experiment was based on composing and playing a short piece using the proposed system. A low-cost gamepad controller was used to tap rhythms and to trigger the observation and prediction modes, as described in Section 5.2.1.

The piece is based on two drum synthesizers that yield samples to a distortion based on clipping (half-wave rectification) that, in turn, yields a change in timbre. The drum onsets and the use of the distortion are controlled by the musician and can be individually looped using separated instances of the proposed system. By looping drum patterns, a base rhythm for the piece can be generated, whereas looping switches in the distortion create different electronic ambiences that give more variation to the piece.

Figure 5.4 shows the instant the looping process is triggered by the musician as a vertical black line – the previous events are manually controlled. After triggering the prediction mode, the musician stopped playing. As it can be seen, the algorithm successfully continued the manually-played pattern.

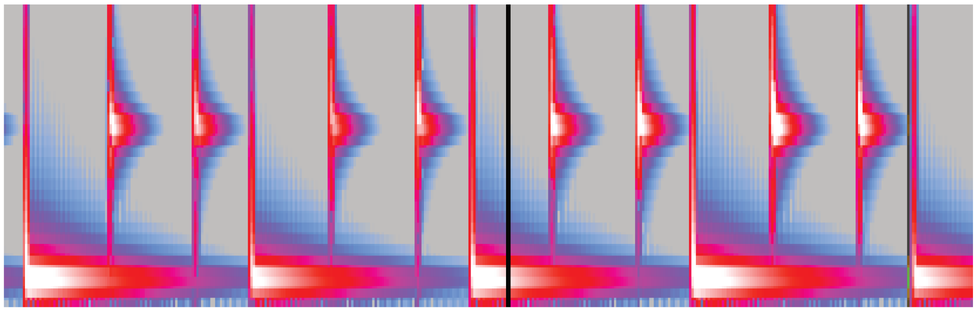

Figure 5.4: Spectrogram showing the prediction of a drum loop. Frequency is represented in the vertical axis and time is represented in the horizontal axis. Two different signals can be seen: a treble and a bass, each corresponding to a different drum. The prediction mode is triggered in the vertical line. After this, the same pattern observed before is maintained.

Figure 5.5 depicts the results of looping the distortion switch, which is triggered in the moment displayed as a vertical black line. As it can be seen, when the distortion is used more harmonics are present. This adds to the drum loops, creating a more complex rhythm.

Figure 5.6 shows a particular excerpt in which the loops regarding both the drums and the effect triggers are active. A simultaneous new layer of drum events is manually played. These events are not looped, but contribute to create a rhythm that would be hard to achieve without the overlap of automated loop patterns.

In the audio recording, it could be noted that the developed rhythms tend to sound natural, despite of the lack of dynamics caused by the use of a low-cost controller. As a consequence of having to play each new loop pattern that would be used, there were no sudden transitions between complex patterns. These characteristics were also observed in the accompanied performance, as shown below.

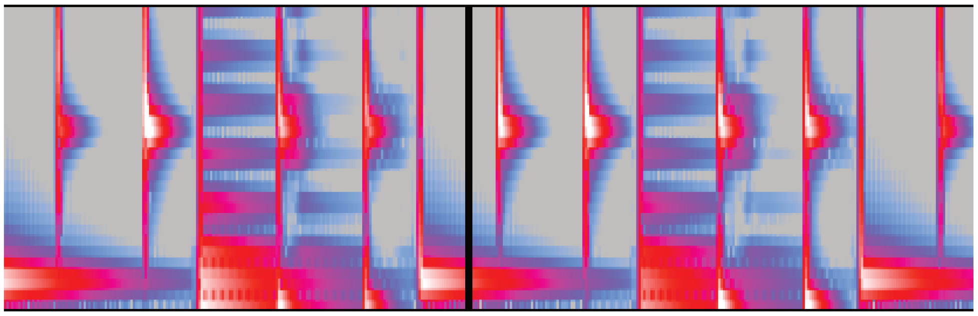

Figure 5.5: Spectrogram showing the prediction of distortion triggers. Frequency is represented in the vertical axis and time is represented in the horizontal axis. The prediction mode is triggered in the vertical line. After this, it can be seen that the distortion is switched on and off following the same pattern as before.

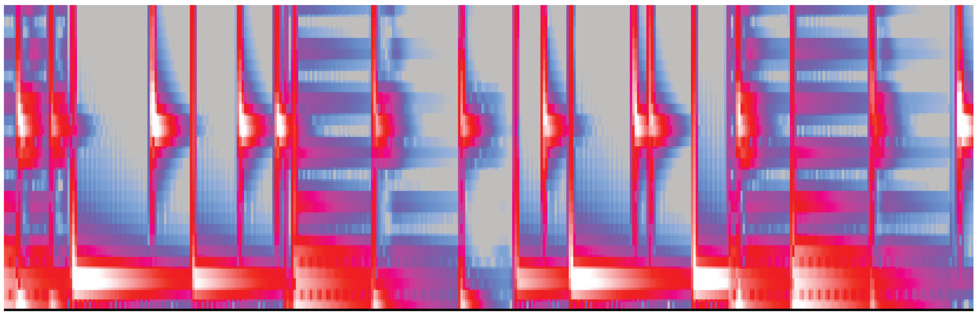

Figure 5.6: Spectrogram showing the use of an additional percussion layer on top of the event and the percussion loops. It can be seen that a more complex pattern arises.

### Accompanied performance

In the second experiment, the proposed system was used in a guitar-electronics duo. The electronic percussion accompaniment, based on two pre-recorded samples of tabla drums, was played over an improvised guitar. The electronic part was played using low-cost gamepad controller to tap rhythms and control the prediction algorithm.

The guitar player stated that, while playing, there was no need to explicitly think about tempo or musical sections, because they emerged naturally from playing. According to the statement, this flexibility allowed the musician to enhance the focus on other musical aspects, such as dynamics and phrasing. Ultimately, this led to a feel of flow [4], which is frequently not the case when playing with drum sequencers, as they are bounded to pre-defined tempo, swing and measures.

The electronics player observed that, as it is easy to switch between the automatic and the manual modes of operation, the percussion parts could be quickly re-arranged. The interaction with the guitar player felt more natural than if a grid interface were used, because using the system with a game controller is more similar to tapping rhythms. Also, while the automatic operation was used, other actions could be performed – manually playing another percussion layer or triggering other digital effects.

In the audio recording, it was noticeable that the feeling of steadiness, often present when using drum machines, was only present in few passages. Hence the proposed instrument allows using steady beats, but not as a requirement of the instrument.

## 5.3 Discussion

This section discusses characteristics of the proposed musical interfaces, highlighting the main differences regarding the previous techniques (drum sequencers and manual loopers).

A feature that must be considered is that neither of the proposed systems pre-define a physical interface for usage or a sound design. These parts of the interaction must be designed by the user, which allows for great flexibility, but also requires a certain level of expertise. A possible way to solve this is by developing pre-set configurations for common use cases, but it must be considered that customizing interfaces and sound designs is frequently an important part of modern musical composition processes.

Both proposed systems allow a quick development of loops, and do not require a visual interface or any pre-definition of tempo, beat or gesture vocabulary. Also, the same physical interface that is used to develop the loops can be used to build a layer of freely improvised variations. This means that the system is potentially more useful in contexts that include flexibility and improvisation.

Both systems have presented great musical potential in improvised performances. They reinforce the paradigm of using multiple layers of sonic transformations to build a piece. However, their applications to offline musical composition are limited, as there is no interface allowing the user to review the current content of the buffer.

An important feature of both interfaces is that they do not require learning a new vocabulary for interaction. Instead, they were built using as basis gestures that are already employed by several musicians. Hence, they represent a small change to the musician techniques when compared to other approaches.

## 5.4 Conclusion

This chapter presented two novel musical interfaces that are based on aspects of rhythm that are present in particular musical conceptions. The first interface aims at repeating continuous data acquired from a sensor, and the second one can be used to loop discrete events. Both interfaces were developed concerning needs of specific artists, but employ concepts that are general enough to be used in many different contexts.

This chapter concludes the technical exposition of this thesis. Further discussions on the implications of the results shown until this point will be held in the next chapter.

# Chapter 6

# Final Discussion

During the development of the entire work described in this thesis, there was a need for extensive theoretical studies and interactions with artists and other engineers. Together with the experimental results, they represented a great source of inspiration, as will be discussed in this chapter. Each topic is contained within a separate section, for the sake of textual organization.

## 6.1 About the objectives

The objectives of this work, as stated in Chapter 1, will be discussed in this section.

1. **To develop models for aspects of rhythm that are present in some specific musical genres**: this was the main subject of the whole text of the thesis. It was found that each situation and application demanded a different model, which would highlight particular aspects of rhythm.

2. **To apply these models to computer-assisted musical transcription**: two proposals for computer assisted transcription of music were presented, one interactive and one automatic. The performance of the proposed models was discussed.

3. **To explore applications for these models other than automatic transcription of music**: models for particular aspects of the musical creation were implemented as a way to provide novel interfaces for artistic expression. For this purpose, a new model was developed and a previously used one was applied.

In view of those facts, the objectives of this work can be considered to have been accomplished. The following discussion regards topics that relate to this thesis as a whole.

## 6.2 The use of domain-specific knowledge in AMT

This thesis is based on a project written in the beginning of the doctorate, in 2010. It was highly based on the Bayes Theorem, which can be stated as:

$$P(s_j|o) = \frac{P(o|s_j)P(s_j)}{P(o)}. \tag{6.1}$$

In one of the many possible interpretations, this expression means that the probability of a symbol (or, in the case of automatic transcription, a note) $j$ being found, given an observation $o$, is evaluated considering both a model of how $o$ is generated when the note is active and a prior model of the behaviour of the note. In this interpretation, $P(o|s_j)$ may be enhanced in the solution by improving the employed digital signal processing algorithm, while $P(s_j)$ depends on a model that indicates the context in which the note is expected to onset. The results in Chapter 3 show that models for the production of note onsets may be applied in some contexts, but the same solution is unlikely to hold for different musical genres. It is important to note that, in spite of the Bayesian inspiration, the solutions provided in this thesis are not Bayesian.

On the other hand, there are aspects of musical signals that are hard to model. This, in addition to the low availability of annotated musical data, except for contemporary Western popular and Classical music, indicate that the simple use of general-purpose classification algorithms, such as Support Vector Machines, is likely to be insufficient for good results in many applications. Hence, it is necessary to apply domain-specific knowledge in MIR-related tasks, despite the limitations that will appear.

In the decade of 2000, there was a huge scientific effort towards the development of novel DSP tools for music transcription. However, as shown in Chapter 2, the most significant improvements came from a more meaningful application of old algorithms. Once again, this indicates that music-related knowledge should be applied on MIR tasks.

In 2013, at the Sound and Music Computing conference, Prof. Xavier Serra[1] gave a keynote speech in which the need for domain-specific knowledge was highlighted. Besides the problems discussed above, he mentioned that a meaningful description to a musical piece is different from culture to culture. Hence, some anthropological studies would be necessary if a system aiming at a different musical genre (or cultural origin) is addressed.

This gives support to the meaningfulness of the hypothesis made in 2010. Although all possibilities regarding this idea were not explored, since that would demand too much time, important steps were taken. The presented work is the first one that focuses exclusively on long-term musical characteristics, like rhythm, and may be used as basis for further development.

[1]Universitat Pompeo Fabra, Barcelona, Spain. Prof. Xavier Serra currently works on several research projects on music information retrieval.

## 6.3 Computational models for rhythm

Rhythm, as a whole, involves many different aspects, many of them of high complexity. Therefore, this thesis does not claim to have modelled rhythm as a whole, but to have presented a series of models for rhythmic aspects that can be employed in useful tools for music information retrieval and musical production.

The models presented here are based on the idea of repetition. Rhythm may also be based on other concepts, like progressions and dialogues (as in some kinds of African drumming). If these cultural manifestations are analyzed, it will be necessary to develop specific models for them, considering the elements that compose the mindset of a musician playing that specific genre.

For the convenience of the reader, the models discussed in this thesis are compared in tabular form in Table 6.1.

| Method | Strengths | Weaknesses |
|---|---|---|
| Note forecasting using search for longest common subsequence (Chapter 3 and Section 5.2) | Capable of forecasting great part of the notes in contemporary Western popular music and provides an intuitive method for sequencing events in electronic music. | Sensitive to errors and cannot work with anything but explicit repetitions. |
| Evaluating periodicity of onsets (Chapter 4) | Is an unsupervised method, but provides results that are similar to the ones obtained with training, and has a greater generalization capability. | Is not completely correlated to the F-Measure, and may give misleading results. |
| Finding periodicity in sensor data using autocorrelation (Section 5.1) | Useful tool for performance in electronic or electroacoustic music. | Repetitive sensor data is not always meaningful to the performance, which reduces its applications. |

Table 6.1: Comparison between the models presented in this thesis.

## 6.4 The proximity between MIR and musical interfaces

One of the purposes of MIR algorithms is to provide users with a more intuitive way to interact with audio information. Employing MIR, users can, for example, sing a melody instead of typing the name of a song they wish to find. This involves knowledge about the user demands and the actual possibilities that can be implemented.

When musical interfaces are considered, algorithms may be used to model gestures that are already native to the musician's performance. As it was shown in this thesis, the reasoning behind these algorithms may be very close to the domain-specific knowledge applied to MIR algorithms. Hence, both applications are, in terms of back-end algorithms, very similar.

A very important difference, however, regards the evaluation methods. While music transcription and other MIR-related algorithms can be adequately evaluated by objective measurements, the evaluation of musical interfaces is inherently subjective and must focus on the new musical possibilities given to the user. In the latter case, the interaction with the potential users of the system is of great importance.

## 6.5 Perspectives for future research

During the previous chapters, some possibilities for future work were identified. These possibilities will be discussed here, with the goal of inspiring future research in the field.

There is a weakness in the algorithm discussed in Chapter 3. It is not robust to input errors, which may compromise its use. In the future, it would be interesting to solve this issue, which would directly impact the system for drum sequencing described in Chapter 5.

Another problem with the forecasting algorithm is that its measure for hypothesis strength is not a continuous function of any variable, thus it is not differentiable. This prevents it to be easily coupled with digital signal processing algorithms. The lack of a derivative, together with the vulnerability to errors, made necessary the development of another model for use in automatic transcription in Chapter 4.

The periodicity-based measure described in Chapter 4 could be used as a parameter for tasks other than automatic transcription. For example, a Blind Source Separation (BSS) system could use the assumption of periodicity in events as another function to maximize, the same way constraints like sparsity are being used. This may result in BSS systems that work better with musical data.

All systems described above lack the ability of detecting dynamics, that is, if a note is played softer or louder. These expression marks are often ignored in the context of automatic transcription. Nevertheless, they are important for documental, educational and musicological purposes, hence a possibility for future research is to address this specific issue.

Both interfaces described in Section 5 can be improved with the development of standalone programs that apply them directly. This would allow their use by musicians that are unfamiliar to the Max/MSP or PureData environments. Also, it could be interesting to add functionalities like “save” and “load”, which would allow users to save interesting states of the system and then load them when convenient.

Chapter 7

# Conclusion

This thesis has presented a series of investigations towards the use of rhythm-related mathematical models in the context of Automatic Music Transcription (AMT). The models were inspired in the idea that musical events tend to be periodically organized. They were initially proposed as means for the improvement of Music Information Retrieval (MIR) tools, but also proved useful for the development of novel interfaces for musical expression.

Previous results found in the literature had shown that using domain-specific knowledge in MIR tasks usually leads to higher performance improvements than applying more modern machine learning techniques. Also, no explicit models for rhythm had been used. Therefore, developing such models was a promising research option.

Two distinct models for music transcription were built. The first one assumes that sequences of musical notes are likely to be found multiple times within the same piece, and was used in the context of semi-automatic music transcription. The second one relies on the assumption that note onsets are periodically organized within a piece, and was applied on the improvement of an automatic transcription system.

Also, two models were developed under similar assumptions, and were applied on interfaces for musical expression. The first model finds repeating loops of sensor-acquired data to forecast it, allowing the live construction of multiple layers of digital effect variations. The second model finds patterns in discrete event loops, which allows sequencing loops without requiring a visual interface for such.

Although all developed models rely on similar assumptions, their applications are diverse. This shows that different, useful models can be obtained from ideas related to rhythm. Also, it was found that domain-specific models that improve music transcription can be adapted to work in the context of musical creation.

The developments achieved during this work indicate that future work should focus on developing domain-specific models for MIR, instead of relying solely on machine learning approaches. Domain-specific models tend to work better than general-purpose solutions when applied within its field. Also, the development of domain-specific models is more likely to inspire solutions in correlated fields, based on similar assumptions.

# Bibliography

[1] T. F. Tavares, J. G. A. Barbedo, R. Attux, and A. Lopes. Survey on automatic transcription of music. *Journal of the Brazilian Computer Society*, 19(4):589–604, 2013.

[2] T. F. Tavares, J. G. A. Barbedo, R. Attux, and A. Lopes. Unsupervised training of detection threshold for polyphonic musical note tracking based on event periodicity. In *38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 21–25, 2013.

[3] S. Trail, M. Dean, T. F. Tavares, G. Odowichuk, P. Driessen, A. W. Schloss, and G. Tzanetakis. Non-invasive sensing and gesture control for pitched percussion hyper-instruments using the kinect. In *New Interfaces for Musical Expression - NIME 2012*, Ann Arbour, Michigan, U.S.A., May 2012.

[4] T. F. Tavares, A. Monteiro, J. G. A. Barbedo, R. Attux, and A. Lopes. Real-time event sequencing without a visual interface. In *Sound and Music Computing conference*, Stockholm, Sweden, Jul 2013.

[5] H. F. Olson. *Music, Physics and Engineering.* Dover Publications Inc., 2 edition, 1967.

[6] H. Helmholtz. *On the Sensation of Tone.* Dover Publications Inc., 4 edition, 1885.

[7] E. Karkoschka. *Notation in new music: a critical guide to interpretation and realisation.* Praeger, 1972.

[8] MIDI Manufacturers Association. MIDI Specifications. ”http://www.midi.org/”, 1996.

[9] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler. A tutorial on onset detection in music signals. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(5):1035 – 1047, sep 2005.

[10] K. D. Martin. A blackboard system for automatic transcription of simple polyphonic music. Technical report, M.I.T. Media Laboratory Perceptual Computing Section, 1996.

[11] A. Sterian and G. H. Wakefield. Robust automated music transcription systems. In *1996 International Computer Music Conference*, Hong Kong, 1996.

[12] T. Tanaka and Y. Tagami. Automatic midi data making from music wave data performed by 2 instruments using blind signal separation. In *SICE 2002. Proceedings of the 41st SICE Annual Conference*, volume 1, pages 451 – 456 vol.1, aug. 2002.

[13] W. Lao, E. T. Tan, and A. Kam. Computationally inexpensive and effective scheme for automatic transcription of polyphonic music. In *Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on*, volume 3, pages 1775 – 1778 Vol.3, june 2004.

[14] M. Triki and D. Slock. Perceptually motivated quasi-periodic signal selection for polyphonic music transcription. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 305 –308, april 2009.

[15] J. Bello, L. Daudet, and M. Sandler. Automatic piano transcription using frequency and time-domain information. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(6):2242 –2251, nov. 2006.

[16] S. Hainsworth and M. D. Macleod. Automatic bass line transcription from polyphonic music. In *In Proc. International Computer Music Conference, Havana*, 2001.

[17] M. Macleod. Fast nearly ml estimation of the parameters of real or complex single tones or resolved multiple tones. *Signal Processing, IEEE Transactions on*, 46(1):141–148, jan 1998.

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

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

[20] V. Rao and P. Rao. Vocal melody extraction in the presence of pitched accompaniment in polyphonic music. *Audio, Speech, and Language Processing, IEEE Transactions on*, 18(8):2145–2154, 2010.

[21] Y. Uchida and S. Wada. Melody and bass line estimation method using audio feature database. In *Signal Processing, Communications and Computing (ICSPCC), 2011 IEEE International Conference on*, pages 1–6, 2011.

[22] K. Dressler. Pitch estimation by the pair-wise evaluation of spectral peaks. In *AES 42nd International Conference*, 2011.

[23] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* Wiley-Interscience, 2 edition, October 2000.

[24] S. Haykin. *Neural Networks: A Comprehensive Foundation.* Prentice Hall, 2 edition, 2000.

[25] M. Privosnik and M. Marolt. A system for automatic transcription of music based on multiple agents architecture. In *Proceedings of MELECON'98*, pages 169–172, Tel Aviv, 1998.

[26] R. Keren, Y. Y. Zeevi, and D.Chazan. Multiresolution time-frequency analysis of polyphonic music. In *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pages 565–568, Pittsburgh, PA , USA, 1998.

[27] M. Marolt. Transcription of polyphonic piano music with neural networks. In *10th Mediterranean Electrotechnical Conference, MEleCon 2000, Vol. 11*, 2000.

[28] J. P. Bello, G. Monti, M. Sandler, and M. S. Techniques for automatic music transcription. In *in International Symposium on Music Information Retrieval*, pages 23–25, 2000.

[29] M. Marolt. A connectionist approach to automatic transcription of polyphonic piano music. *Multimedia, IEEE Transactions on*, 6(3):439 – 449, june 2004.

[30] O. Gillet and G. Richard. Automatic transcription of drum loops. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, volume 4, pages iv–269 – iv–272 vol.4, may 2004.

[31] O. Gillet and G. Richard. Automatic transcription of drum sequences using audiovisual features. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, volume 3, pages iii/205 – iii/208 Vol. 3, march 2005.

[32] G. E. Poliner and D. P. Ellis. Improving generalization for classification-based polyphonic piano transcription. In *Applications of Signal Processing to Audio and Acoustics, 2007 IEEE Workshop on*, pages 86 –89, oct. 2007.

[33] O. Gillet and G. Richard. Transcription and separation of drum signals from polyphonic music. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(3):529 –540, march 2008.

[34] G. Costantini, M. Todisco, and R. Perfetti. On the use of memory for detecting musical notes in polyphonic piano music. In *Circuit Theory and Design, 2009. ECCTD 2009. European Conference on*, pages 806 –809, aug. 2009.

[35] G. Costantini, R. Perfetti, and M. Todisco. Event based transcription system for polyphonic piano music. *Signal Processing*, 89(9):1798 – 1811, 2009.

[36] A. Weller, D. Ellis, and T. Jebara. Structured prediction models for chord transcription of music audio. In *Machine Learning and Applications, 2009. ICMLA '09. International Conference on*, pages 590 –595, dec. 2009.

[37] G. Costantini, M. Todisco, R. Perfetti, R. Basili, and D. Casali. Svm based transcription system with short-term memory oriented to polyphonic piano music. In *MELECON 2010 - 2010 15th IEEE Mediterranean Electrotechnical Conference*, pages 196 –201, april 2010.

[38] A. Tjahyanto, Y. Suprapto, M. Purnomo, and D. Wulandari. Fft-based features selection for javanese music note and instrument identification using support vector machines. In *Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on*, volume 1, pages 439–443, 2012.

[39] Y.-S. Wang, T.-Y. Hu, and S.-K. Jeng. Automatic transcription for music with two timbres from monaural sound source. In *Multimedia (ISM), 2010 IEEE International Symposium on*, pages 314 –317, dec. 2010.

[40] J. Nam, J. Ngiam, H. Lee, and M. Slaney. A classification-based polyphonic piano transcription approach using learned feature representations. In *12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, 2011.

[41] A. Sterian, M. H. Simoni, and G. H. Wakefield. Model-based musical transcription. In *1999 International Computer Music Conference*, Beijing, China, 1999.

[42] A. Cemgil, B. Kappen, and D. Barber. Generative model based polyphonic music transcription. In *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.*, pages 181 – 184, oct. 2003.

[43] M. Goto. A real-time music-scene-description system: predominant-f0 estimation for detecting melody and bass lines in real-world audio signals. *Speech Communication*, 43(4):311 – 329, 2004. Special Issue on the Recognition and Organization of Real-World Sound.

[44] A. Kobzantsev, D. Chazan, and Y. Zeevi. Automatic transcription of piano polyphonic music. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pages 414 – 418, sept. 2005.

[45] H. Thornburg, R. Leistikow, and J. Berger. Melody extraction and musical onset detection via probabilistic models of framewise stft peak data. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(4):1257 –1272, may 2007.

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

[47] H.-H. Shih, S. Narayanan, and C.-C. Kuo. An hmm-based approach to humming transcription. In *Multimedia and Expo, 2002. ICME '02. Proceedings. 2002 IEEE International Conference on*, volume 1, pages 337 – 340 vol.1, 2002.

[48] C. Raphael. Automatic transcription of piano music. In *Proceedings of the Third International Conference on Music Information Retrieval: ISMIR 2002*, pages 15–19, Paris, France, 2002.

[49] M. Ryynanen and A. Klapuri. Polyphonic music transcription using note event modeling. In *Applications of Signal Processing to Audio and Acoustics, 2005. IEEE Workshop on*, pages 319 – 322, oct. 2005.

[50] M. Ryynanen and A. Klapuri. Automatic bass line transcription from streaming polyphonic audio. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–1437 –IV–1440, april 2007.

[51] D. ning Jiang, M. Picheny, and Y. Qin. Voice-melody transcription under a speech recognition framework. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–617 –IV–620, april 2007.

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

[53] N. Bertin, R. Badeau, and G. Richard. Blind signal decompositions for automatic transcription of polyphonic music: Nmf and k-svd on the benchmark. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 1, pages I–65 –I–68, april 2007.

[54] S. Sophea and S. Phon-Amnuaisuk. Determining a suitable desired factors for nonnegative matrix factorization in polyphonic music transcription. In *Information Technology Convergence, 2007. ISITC 2007. International Symposium on*, pages 166 –170, nov. 2007.

[55] E. Vincent, N. Berlin, and R. Badeau. Harmonic and inharmonic nonnegative matrix factorization for polyphonic pitch transcription. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 109 –112, 31 2008-april 4 2008.

[56] G. Grindlay and D. Ellis. Multi-voice polyphonic music transcription using eigeninstruments. In *Applications of Signal Processing to Audio and Acoustics, 2009. WASPAA '09. IEEE Workshop on*, pages 53 –56, oct. 2009.

[57] N. Bertin, R. Badeau, and E. Vincent. Fast bayesian nmf algorithms enforcing harmonicity and temporal continuity in polyphonic music transcription. In *Applications of Signal Processing to Audio and Acoustics, 2009. WASPAA '09. IEEE Workshop on*, pages 29 –32, oct. 2009.

[58] N. Bertin, R. Badeau, and E. Vincent. Enforcing harmonicity and smoothness in bayesian non-negative matrix factorization applied to polyphonic music transcription. *Audio, Speech, and Language Processing, IEEE Transactions on*, 18(3):538 –549, march 2010.

[59] J. Han and C.-W. Chen. Improving melody extraction using probabilistic latent component analysis. In *Proceedings of the ICASSP*, 2011.

[60] R. J. Hanson and C. L. Lawson. *Solving least squares problems.* Philadelphia, 1995.

[61] B. Niedermayer. Non-negative matrix division for the automatic transcription of polyphonic music. In *Proceedings of the ISMIR*, 2008.

[62] M. Mauch and S. Dixon. Approximate note transcription for the improved identification of difficult chords. In *11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, 2010.

[63] T. Tavares, G. Odowichuck, S. Zehtabi, and G. Tzanetakis. Audio-visual vibraphone transcription in real time. In *Multimedia Signal Processing (MMSP), 2012 IEEE 14th International Workshop on*, pages 215–220, 2012.

[64] N. Bertin, C. Fevotte, and R. Badeau. A tempering approach for itakura-saito non-negative matrix factorization. with application to music transcription. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 1545 –1548, april 2009.

[65] A. Dessein, A. Cont, and G. Lemaitre. Real-time polyphonic music transcription with non-negative matrix factorization and beta-divergence. In *11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, 2010.

[66] S. A. Abdallah and M. D. Plumbley. An independent component analysis approach to automatic music transcription. In *Proceedings of the 114th AES Convention*, 2003.

[67] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent. Discriminative non-negative matrix factorization for multiple pitch estimation. In *Proceedings of the ISMIR 2012*, 2012.

[68] E. Vincent and X. Rodet. Music transcription with ISA and HMM. In *5th Int. Conf. on Independent Component Analysis and Blind Signal Separation (ICA)*, pages 1197–1204, Granada, Espagne, 2004.

[69] C.-T. Lee, Y.-H. Yang, and H. Chen. Automatic transcription of piano music by sparse representation of magnitude spectra. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pages 1–6, 2011.

[70] E. Benetos and S. Dixon. Multiple-instrument polyphonic music transcription using a convolutive probabilistic model. In *Sound and Music Computing (SMC 2011)*, 2011.

[71] H. Kirchhoff, S. Dixon, and A. Klapuri. Multi-template shift-variant non-negative matrix deconvolution for semi-automatic music transcription. In *Proceedings of the ISMIR 2012*, 2012.

[72] F. Argenti, P. Nesi, , and G. Pantaleo. Automatic transcription of polyphonic music based on the constant-q bispectral analysis. *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, 19(6):1610–1630, 2011.

[73] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41(12):3397 –3415, dec 1993.

[74] O. Derrien. Multi-scale frame-based analysis of audio signals for musical transcription using a dictionary of chromatic waveforms. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 5, page V, may 2006.

[75] K. O'Hanlon, H. Nagano, and M. Plumbley. Structured sparsity for automatic music transcription. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 441–444, 2012.

[76] G. Reis, N. Fonseca, and F. Ferndandez. Genetic algorithm approach to polyphonic music transcription. In *Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on*, pages 1 –6, oct. 2007.

[77] G. Reis, F. Fernandez de Vega, and A. Ferreira. Automatic transcription of polyphonic piano music using genetic algorithms, adaptive spectral envelope modeling, and dynamic noise level estimation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(8):2313–2328, 2012.

[78] P. D. O'Grady and S. T. Rickard. Automatic hexaphonic guitar transcription using non-negative constraints. In *Signals and Systems Conference (ISSC 2009), IET Irish*, pages 1 –6, june 2009.

[79] S. Phon-Amnuaisuk. Transcribing bach chorales using non-negative matrix factorisation. In *Audio Language and Image Processing (ICALIP), 2010 International Conference on*, pages 688 –693, nov. 2010.

[80] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* ACM Press, Addison-Wesley, 1999.

[81] J. S. Downie. The music information retrieval evaluation exchange (mirex). *D-Lib Magazine*, 12(12), Dec 2006.

[82] B. Lincoln. An experimental high fidelity perceptual audio coder project in mus420 win 97. Technical report, CCRMA - Stanford, 1998.

[83] T. F. Tavares, J. G. A. Barbedo, and A. Lopes. Towards the evaluation of automatic transcription of music. In *Proceedings of the VI Brazilian Congress of Audio Engineering (AES2008)*, 2008.

[84] A. Daniel and V. Emiya. Perceptually-based evaluation of the errors usually made when automatically transcribing music. In *Proceedings of the ISMIR 2008*, 2008.

[85] N. Fonseca and A. Ferreira. Measuring music transcription results based on a hybrid decay/sustain evaluation. In *7th Triennial Conference of European Society for the Cognitive Sciences of Music (ESCOM 2009)*, 2009.

[86] M. Goto, H. Hashigushi, T. Nishimura, and R. Oka. RWC Music Database: Popular, classical, and jazz music databases. In *Proc. of the 3rd International Conference on Music Information Retrieval*, pages 281–288, Paris, France, October 2002.

[87] M. Ryynanen and A. Klapuri. Transcription of the singing melody in polyphonic music. In *Proc. 7th International Conference on Music Information Retrieval*, pages 222–227, Victoria, BC, Canada, October 2006.

[88] M. Ryynanen and A. Klapuri. Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, 32(3):72–86, 2008.

[89] S. A. Raczynski, E. Vincent, F. Bimbot, and S. Sagayama. Multiple pitch transcription using dbn-based musicological models. In *1th International Society for Music Information Retrieval Conference (ISMIR 2010)*, 2010.

[90] U. Simsekli and A. T. Cemgil. A comparison of probabilistic models for online pitch tracking. In *Sound and Music Computing (SMC) 2010*, 2010.

[91] V. Emiya, R. Badeau, and B. David. Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. *IEEE Transactions on Audio, Speech and Language Processing*, 2010.

[92] S. W. Hainsworth and M. D. Macleod. The automated music transcription problem, 2007.

[93] S. Inc. Soundhound. `http://www.soundhound.com/|`.

[94] J. Yin, Y. Wang, and D. Hsu. Digital violin tutor: an integrated system for beginning violin learners. In *Proceedings of the 13th annual ACM international conference on Multimedia*, MULTIMEDIA '05, pages 976–985, New York, NY, USA, 2005. ACM.

[95] W. Boo, Y. Wang, and A. Loscos. A violin music transcriber for personalized learning. In *Multimedia and Expo, 2006 IEEE International Conference on*, pages 2081 –2084, july 2006.

[96] Recognisoft. Solo explorer. `http://www.recognisoft.com/`.

[97] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *Information Theory, IEEE Transactions on*, 23(3):337–343, may 1977.

[98] G. Assayag, S. Dubnov, and O. Delerue. Guessing the composer's mind: Applying universal prediction to musical style. In *Proceedings ICMC 99*, Beijing, China, 1999.

[99] P. Howell, I. Cross, and R. West. *Musical Structure and Cognition.* London Academic Press, 1985.

[100] C. E. Shannon. Prediction and entropy of printed english. *The Bell System Technical Journal*, 30:50–64, 1951.

[101] R. Solomonoff. A formal theory of inductive inference, part i. *Information and Control*, 7(1):1–22, Mar 1964.

[102] R. Solomonoff. A formal theory of inductive inference, part ii. *Information and Control*, 7(2):224–254, Jun 1964.

[103] J. Ziv. Coding theorems for individual sequences. *Information Theory, IEEE Transactions on*, 24(4):405 – 412, jul 1978.

[104] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 1948.

[105] J. Cleary and I. Witten. Data compression using adaptive coding and partial string matching. *Communications, IEEE Transactions on*, 32(4):396 – 402, apr 1984.

[106] D. Conklin. Prediction and entropy of music. Master's thesis, Department of Computer Science, University of Calgary, Canada., 1990.

[107] D. Conklin and I. H. Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1):51–74, 1995.

[108] I. H. Witten, L. C. Manzara, and D. Conklin. Comparing human and computational models of music prediction. *Computer Music Journal*, 18(1):70–80, 1995.

[109] D. Ron, Y. Singer, and N. Tishby. The power of amnesia: Learning probabilistic automata with variable memory length. *Machine Learning*, 25:117–149, 1996.

[110] J. L. Triviño-Rodriguez and R. Morales-Bueno. Using multiattribute prediction suffix graphs to predict and generate music. *Computer Music Journal*, 25(1):62–79, 2001.

[111] M. T. Pearce and G. A. Wiggins. Improved methods for statistical modelling of monophonic music. *Journal of New Music Research*, 33(4), 2004.

[112] J. B. Maxwell, P. Pasquier, and A. Eigenfeldt. Hierarchical sequential memory for music: A cognitively-inspired approach to generative music. 2010.

[113] D. Huron. *Sweet Expectation: Music and the Psychology of Expectation.* MIT Press, 2006.

[114] E. Namour. *The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model.* University of Chicago Press, 1990.

[115] M. D. Smucker, J. Allan, and B. Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pages 623–632, New York, NY, USA, 2007. ACM.

[116] mididatabase.com. The midi database. `http://mididatabase.com/`.

[117] Ronald Fisher. *The Design of Experiments.* Ronald Fisher, 1935.

[118] A. de Cheveigné and H. Kawahara. Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.

[119] A. Klapuri and M. Davy. *Signal Processing Methods for Music Transcription.* Springer-Verlag, 2006.

[120] G. Poliner, D. Ellis, A. Ehmann, E. Gomez, S. Streich, and B. Ong. Melody transcription from music audio: Approaches and evaluation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(4):1247 –1256, may 2007.

[121] University of Iowa. Musical Instrument Samples. "http://theremin.music.uiowa.edu/MIS.html", 2005.

[122] Cycling74. Max-msp. `http://cycling74.com/products/max`.

[123] P. Community. Rhythmdelay. http://puredata.hurleur.com/sujet-8421-electronics-two-rhythm-delay-steroids/.

[124] AudioMulch. Sdelay. http://www.audiomulch.com/help/contraption-ref/SDelay.

[125] Pd-Community. Puredata. http://puredata.info/.