Towards Automatic Rhythm Description of Musical Audio Signals — Representations, Computational Models and Applications

by

Fabien Gouyon

Submitted to the Department of Technology
in partial fulfilment of the requirements for the

Diploma of Advanced Studies, Doctorate in
Computer Science and Digital Communication

at the Universitat Pompeu Fabra,
Barcelona, Spain

Thesis supervisor: Dr. Xavier Serra i Casals

6th November 2003

Abstract

The aim of this document is to give an overview of part of my research activities during the last years and to advocate a research proposal for the fulfilment of my Ph.D. dissertation.

This document is concerned with the automatic description of rhythmic aspects of musical audio signals. First, we formulate motivation statements to this topic and comment on terminology issues and inherent difficulties to the problem of automatic rhythm description. Then, we propose a review of rhythm representation schemes, rhythm description computational models and applications for rhythmic description in the context of music content-based processing. In this review we highlight a set of requirements that we believe of interest for rhythm representations; we also propose an original comparison of many state-of-the-art computational models with respect to the functional blocks of a comprehensive diagram. We then report on our initial research objectives and our contributions to date: We implemented algorithms for the automatic extraction from musical audio signals of the tempo and beats at three different metrical levels, as well as the swing ratio. We proposed enhancements to existing rhythm representation schemes. Evaluations are also provided: the tick induction algorithm reaches an error rate of about 20%, in tactus induction the error rate is about 25% and in downbeat induction it is about 10%. Finally, given our initial aims and current achievements we formulate a research proposal for future work: the main objective will be the design of a computational model for the description of audio signal metrical structure, implementing influential schemes between metrical levels.

Author contact: fgouyon _at_ iua.upf.es
This document was written with LyX.

Acknowledgements

The work reported in this thesis is really not the work of a single person. Ok, I did some of it, but it truly would have been impossible to carry on in any other place and surrounded by other people. There are so many people without whom this work would not exist, I have troubles thanking them all.

First and foremost, I would first like to thank my advisor, Xavier Serra, for inviting me to join the Music Technology Group in Barcelona, back in 1999, and for encouraging and supporting me since then. Thanks also for creating this wonderful research center in which I am able to carry on my work. I am also grateful for the multiple opportunities I had to make my work public (present it in international conferences, submit it to journals).

For their enormous influence on my work, for being amazing sources of knowledge and always being willing to spend a couple of hours (with or without beer) discussing this or that aspect of some scientific problem, or proof-reading stuff of mine, my deepest thanks go to Perfecto Herrera and Pedro Cano. Many other people in the MTG also had a great influence on my scientific trajectory and gave me great support e.g. in implementation matters, I want to thank them here: Emilia Gómez, Jordi Bonada, Martin Kaltenbrunner, Günter Geiger and Lars Fabig. Thanks also to an anonymous reviewer, Simon Dixon, Harvey Thornburg, Stefan Bilbao, Tamara Smyth, Bob Sturm and Stefania Serafin.

Thanks to Joana Clotet and Cristina Garrido for helping in all administrative tasks and, most importantly, for creating such a nice atmosphere around them.

For providing the means to work in amazingly smooth material conditions, thanks to Marteen de Boer, Ramón Loureiro and Carlos Atance. Another technical thanks goes to the CLAM team for their unconditional support, especially David Garcia and Miquel Ramírez.

Part of the research reported here has been supported by the European IST project CUIDADO.

For being wonderful friends and making Barcelona such a nice place to live, I would like to thank Pedro, Martin K, Larsito, Emilia, Jordis (B & J), Jojo, Line, Marife, Daniele, Alvaro, Alejandro, GNUter, Nadine, Rossella, Marion (la loca), Álex, Oscars (both), Ramón, Chiara, Aurora, Laura, José Pedro, Vadim, Joana, Marteen, Perfe, Begonya, Hagar, Arantxa. And Claudia.

Last but obviously not least, my gratitude goes to my family, especially both my parents, Va, PF, Lise and Fanny for loving and supporting.

Contents

List of Figures

Chapter 1

Introduction

When listening to music, or even barely hearing it, people perceive rhythm. People without musical training (other than having been exposed to music since childhood) as well as trained musicians (trained to play Jazz, Pop or Baroque, etc.) hear out rhythm from musical audio signals. Musical signals convey rhythm. Hence, any part of the musical communication chain (composing, performing, listening, commenting, buying, searching, etc.) has to do with rhythm. Involving a machine at any point calls for an explicit and precise handling of musical rhythm. Along this standpoint, we propose to address in this dissertation three topics:

- The definition of musical rhythm, or rhythmic elements
- The design of algorithms for the automated extraction of such elements from audio data
- The usage of these elements in specific application contexts

Roughly put, this dissertation is about extracting automatically a specific rhythm representation from audio signals, and demonstrating its usefulness in application contexts. Each chapter below (review, objectives, contributions and conclusions) refers to these three research topics. However, all along this dissertation, a special focus is being put on the second point (the implementation in the form of computer softwares of algorithms for describing automatically the musical rhythm). The type of musical signals is restricted to audio signals, from a relatively recent Western repertoire.

This document is divided in five chapters. In this chapter, we present the research context. More precisely, in Section 1.1, we advocate that the topic addressed is relevant to computer music research and specifically to music content processing research (i.e. we propose an answer to the question *Why describe musical rhythm?*). In Section 1.2, we present concepts of first relevance to this thesis (*What are we talking about, exactly?*). In Section 1.3, we detail issues inherent to the problem at hand (*Why is the task difficult?*). In Section 1.4,

we discuss the computational modeling paradigm in musical rhythm research (*Why make a computer model?*). In Chapter 2, we present a review of past and current research in musical rhythm and present state-of-the-art computational models (*What has already been achieved?*). Then, commenting on previous research achievements, Chapter 3 states in details our research objectives (*What are the author's research aims?*). Chapter 4 presents in details researches done by the author during the first years of Ph.D. and their evaluations (*What are the author's contributions so far?*). Finally, given our aims and current achievements, we formulate in Chapter 5 a proposal of future research directions for the fulfilment of the Ph.D. dissertation (*What is left to do?*).

1.1 Motivations

1.1.1 Music content processing applications

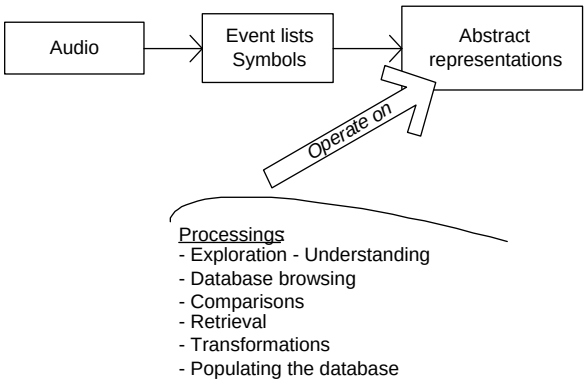

Figure 1.1: Overview of music content processing. From musical data to meaningful representations.

The standardization of personal computers and worldwide low-latency networks has spurned developments in digital audio dissemination. Hence, there are ample applications for content-based analysis of music, in particular rhythmic content. It is now *technically* possible to use PCs to browse and retrieve large databases of music files in audio and symbolic formats, analogous to text databases. In addition, computers are now powerful enough to serve as high-quality home-studios. For Aigrain [1], in the near future, content-processing technologies will provide "new aspects of listening, interacting with music, finding and comparing music, performing it, editing it, exchanging music with others or selling it, teaching it, analyzing it, and criticizing it."

Therefore, as is illustrated in Figure 1.1 on page 7, "processing" musical data may mean:

- Exploring, understanding items in a database

- Browsing a database
- Comparing two or more musical items
- Retrieving any desired selection of items from a database
- Transforming items
- Introducing new items to a database

1.1.2 Emulation of a human capability

Tapping in time along with some music is a very common scene. As is the synchronization of different musicians in the performance of some musical piece. Perceiving or producing rhythm is ubiquitous in our lives. Hence, studying musical rhythm is another paradigm to the study of human. And designing machines that would perform a certain task just as we do it (no matter how well or badly) has always been a source of motivation to science.

1.1.3 Musical interactions man-machine

A complementary motivation to the previous stands in the design of machines one could interact with. How amazing would it be to just play a melody on some instrument and have a machine listening and producing an accompaniment, or being able to point on a score the precise note one is playing. Or maybe to compose jointly with a machine. "Trading fours" with a computer would also seem like fun (i.e. one plays a musical phrase, then the computer answers, this process is iterated, each one learning about the other's playing style, imitating, etc.).

Let us provide a few pointers towards some systems that implement similar functionalities: [33], [133], [105], [111], [14], [55].

1.2 Terminology

Rhythm: "[...] a precise, generally accepted definition of rhythm does not exist" [54, p.149]. The same consideration can be found in the very introduction of [93]. In its more general meaning, 'rhythm' is another term for 'musical time'. Just as it has a melodic dimension, music has a rhythmic dimension. Here, the word 'rhythmic' implicitly encompasses all the temporal aspects of music, from small- to large-scale temporal phenomena. It is indeed difficult to draw a clear line regarding the temporal scope of rhythm. Some use this word when referring to the duration of a note, expressed relatively to a reference pulse, as in "this note is an eighth-note" (see "Quantized duration" on page 12). Others refer to the rhythm of a musical piece, as the tempo and the time signature. Others refer to the rhythm of a pattern of notes, as e.g. a typical Waltz pattern, this usually entails the notion of quantized duration as well as large-scale properties

(as the tempo and the time signature, even some expressiveness features). Others use the related adjective ("rhythmic") to describe somehow a percept that leads (or not) to dance to the music. Confusion abounds.
Rhythm is commonly defined indirectly. For instance, when stating rhythm is grounded in the architectonic organisation of the musical events in time and in the accentuation (or differentiation) of some musical events in opposition to some others. Thus, rhythm involves regularity (or organisation) and also differentiation [54, p.151].
Rhythm is also indirectly defined in its usual opposition to the 'meter' and the 'form'. The three terms involve regularity and differentiation. Yet, the distinction lies in the concept of 'perceptual present'. For London [93], "rhythm involves the pattern of durations that is phenomenally present in the music, while meter involves our perception and anticipation of such patterns." He also puts it differently: "meter [is] a mode of attending, while rhythm is that to which we attend." London considers that rhythm's proper meaning refers to the "smaller-scale features of musical experience." The reason for this would be that rhythm "is apprehended within the span of the perceptual present", unlike the form and the meter that would "engage one's long-term memory of the piece at hand as well as one's musical background and knowledge." Similarly, Clarke [27] makes the distinction between "small- to medium- scale temporal phenomena" (rhythm) and "large-scale temporal phenomena" (form). These definitions are foreshadowed in Fraisse's work. Fraisse defines the 'perceptual present' as "the temporal extent of stimulations that can be perceived at a given time, without the intervention of rehearsal during or after the stimulation." [27, p.474]. In other words, it is the dividing line between the perception of time (temporal phenomena extending to no more than about 5 seconds), and the estimation of time, that relies primarily on the reconstruction of temporal estimates from information stored in memory. Cooper et al. [32] extend the definition of rhythm to all the temporal scopes of description (from single note to entire movement). They define rhythm as "the way in which one or more unaccented beats are grouped in relation to an accented one." Five basic groupings would be permitted, those of prosody (*iamb*, *anapest*, *trochee*, *dactyl*, *amphibrach*). In this definition, the underlying meaning of the word 'beat' is very broad. Notes could be grouped together according to their respective accentuations, as could be groups of notes, phrases and so on [32, p.6].

Accent: 'Accentuation' commonly refers to the human ability to perceptually apply a mark on some events in the musical flow, in opposition to other such events. The way we actually carry out this marking process is still not well understood. Pitch, intensity, duration [127], harmony and timbre perception certainly have an influence on our way to hear out rhythm from music [129]. But one cannot state unequivocally that one of these factors prevails, nor that these are the sole factors of rhythm perception. [93] defines an 'accent' as a "means of differentiating events and thus giving them

a sense of shape or organisation." One can also find various definitions of accentuation in the literature.[1]

Pulse - Beat: Cooper et al. [32] define a pulse as "[...] one of a series of regularly recurring, precisely equivalent stimuli. [...] Pulses mark off equal units in the temporal continuum." Commonly, 'pulse' and 'beat' are often used indistinctly and refer *both* to one element in such a series and to the whole series itself.[2]

A musical composition usually exhibits a reference time unit (e.g. the reference time division used in standard Western rhythmic notation). The 'beat' corresponds to this time unit. It is often represented by the quarter-note (or 'crotchet'), or more generally by the denominator of the score time signature (e.g. if the time signature is $\frac{12}{8}$, then the beat would be an eighth-note).

However, such a purely formal definition of the beat is not entirely satisfying as it is generally considered that "the sense of pulse may exist subjectively" [32]. In addition, humans tend to perceive a beat in stimuli that were not generated with a "rhythmic intention", e.g. a watch, machines, ocean waves, etc. (see 'Subjective Rhythmisation' [54, p.155]). In this context, the 'beat' corresponds to that time unit that can be apprehended in an immediate, or reactive, fashion (see aforementioned notion of 'perceptual present' on page 8), and that leads one to tap his feet or fingers accordingly. Madison writes that "[p]ulse is the subjective experience of isochrony, which is typically elicited by series of sensory events with close to isochronous spacing [...]" [97]. For Scheirer [117], the beat is "the sequence of equally spaced phenomenal impulses which define a tempo for the music." Human anatomy and motor-behavior naturally account for pulses (walking, heart-beat, breathing). It is commonly thought that there is an intimate connection between these physiological properties and the reactive aspect of rhythm [54, pp.151-155], [83, pp.40-47]. Lerdahl et al. [90] use the term 'tactus' to refer to the perceptually most prominent pulse.

Tempo - Tactus: Mirroring the ambiguity in the definition of 'pulse' on this page, "[t]he concept of tempo often leads to confusion [...] because it corresponds to both a musical [concept] (the number of defined beats per minute) and a psychological concept (perceived rate of events)" [46]. The tempo refers to the pace of a musical excerpt (how fast or slow it is). If it is counted in beats per minute (BPM), it explicitly refers to a specific pulse (Drake et al. refer to the "physical tempo" or "musical tempo" [46]): it is inversely proportional to the pulse period. Here, the 'pulse' can correspond to the score temporal unit, in this case, one refers to M.M. tempo (Maelzel metronome).

However, from a psychological point of view, thing are less clear. Tempo

[1] see http://www.music.indiana.edu/som/courses/rhythm/illustrations/accent.html

[2] In this thesis, 'beats' will have the same meaning as 'beat positions' or 'beat indexes'.

can also mean tactus [90]: as Scheirer [117, p.56] writes, "[t]he tempo of a sound is the *perceptual* sense that the sound is recurrent in time at regular intervals, where the interval length is between 250 ms and 2 s" (emphasis ours). Furthermore, when hearing a musical excerpt, one can give an appreciation whether it is fast or slow without referring explicitly to the beats of a specific pulse (Drake et al. refer to the "perceived tempo" [46]); in addition to the perception of a specific pulse, the rapidity of an excerpt relies on the perception of "event density" [46].

Beat "phase": A beat is characterized by a period and a phase. Its period is the distance between two beats (it is inversely proportional to the tempo) and its phase is specified by the temporal location of one beat (usually the first beat).

Metric structure - Meter: The metric structure of a musical piece is based on the coexistence of several (often more than two) pulses (or 'metrical levels'). The segmentation of time by a given pulse provides the basic time span to measure musical event accentuations. Periodic recurrences of these accents define other, higher, metrical levels. Along this line, [32] define the meter as "the number of pulses between the more or less regularly recurring accents." However, this definition does not specify whether this would mean that the "meter imposes an accent structure on beats" [12], or conversely that the meter would emerge from event accentuations, inherent to the excerpt's melody, dynamics, harmony and timbre. [27] and [59] combine both concepts, the metric structure would be an abstraction from the stimulus properties, a construct with no rigid reality in the music itself. Pulses are sometimes perceived as sounded events and sometimes as unsounded time points [86], and they need not be *precisely* equivalent (in a physical meaning) to be perceived as elements of the same metrical level.

The "Generative Theory of Tonal Music" (GTTM) of Lerdahl et al. [90] further formalizes properties of the metric structure in Western tonal music: Pulses of a metrical level must be equally spaced. Levels range from *low level* ones (small interval between consecutive pulses) to *high level* ones (longer interval between consecutive pulses). There must be a pulse of the metric structure for every note. A pulse at a high level must also be a pulse at each lower level. Pulses obey a discrete time grid, time intervals being all multiple of a common duration: the smallest metrical level (see 'tick' on 12). Figure 2.2 on page 20 illustrates GTTM metric structure representation on an audio waveform.

Time signature - Measure - Downbeat: Restricting the notion of meter to two levels, Yeston defines it as "an outgrowth of the interaction of two distinct levels (two differently-rated strata), the faster of which provides the elements and the slower of which groups them" [143]. This definition seems close to the usual description of meter that can be found in a score, given by the time signature and the bar lines. The bar lines define the

slower of the two levels (the 'measure') and the time signature defines the number of faster pulses that make up one measure. For instance, a $\frac{6}{8}$ time signature indicates that the basic temporal unit is an eighth-note (a 'note' referring to a 'whole', or 'semi-breve') and that between two bar lines there is room for six of them. Two categories of meter are generally distinguished: duple and triple. This notion is contained in the numerator of the time signature: if the numerator is a multiple of two, then the meter is duple, if not a multiple of two but of three, the meter is triple. For instance, $\frac{2}{4}$ and $\frac{4}{4}$ signatures are duple, $\frac{3}{4}$ and $\frac{9}{8}$ are triple.
The 'downbeat' corresponds to the first beat in a measure. Any other beat in a measure is called an 'off-beat'. The beat that immediately precedes the downbeat is the 'upbeat'.

Tick - Tatum: The metric structure smallest level lacks a commonly accepted name. Bilmes uses the term 'tatum' [7]. Schloss refers to the 'attack-point' [119]. In our understanding, [110]'s 'basic time unit' and[76]'s 'chronota' refers to the same concept. We propose the term 'tick'.
The tick seems better defined as "the regular time division that most highly coincides with all note onsets" [7, p.22] than as the shortest interval between notes. Indeed, in syncopated musical excerpt for example, the tick may not be explicit in the list of successive note intervals, it may rather be implied by the relationships between those intervals.

Quantized duration - Metrical point: The GTTM (aforementioned on the page before) specifies that there must be a pulse of the metric structure for every note. Accordingly, given a list of note onsets, the quantization (or rhythm-parsing) task aims at making it fit into Western music notation. Viable time points (metrical points) are those defined by the different coexisting pulses. Quantized durations are then rational numbers (e.g. 1, $\frac{1}{4}$, $\frac{1}{6}$) relative to a chosen time interval: the time signature denominator. Cemgil et al. [18] defines quantization as "the extraction of an acceptable description (music notation) from a music performance", 'acceptable' meaning 'easy to read while representing the timing information accurately'.

Swing - Groove: The term 'swing' originates in jazz music. For [56], one characteristic aspect of the swing is that "consecutive eighth-notes are performed as long-short patterns." [87] defines it as a "slight delay of the second and fourth quarter-beats" (in this article, 'beat' refers to half-note). The swing ratio refers to a mathematical expression: the duration of the first eighth-note divided by that of the second.
The term 'groove' resists precise definitions, but, as the 'feel', it usually refers to a rhythmic phenomenon, resulting from the conflict between a fixed pulse and various timing accents played against it; or resulting from the "musician moving in non-metronomical ways" [138]. The swing (as defined above) is a particular case of groove.

IOI: Short for inter-onset interval. Some authors define an IOI as the time difference between two *successive* onsets, e.g. [3]. Some others as the time difference between any two onsets, not necessarily successive, e.g. [24, p.64], [40, p.44].

1.3 Difficulties in automatic rhythm description

Automatically describing musical rhythm is not obvious. First of all, as Desain et al. argue in [37], because it seems to entail two dichotomic processes: a bottom-up process enabling very rapidly the percept of pulses from scratch, and a top-down process (a persistent mental framework) that lets this induced percept guide the organisation of incoming events. Embodying in a computer program both reactivity to the environment and persistence of internal representations seems a challenge.

Second, it seems difficult to decouple rhythm description from other auditory processes modeling as for instance stream segregation, instrument recognition, harmonic structure parsing. The disassembly of music into a rhythmic part, a melodic part, a harmonic part and an orchestration part (i.e. repartition of timbres) is an analytical rationale that seems suitable for doing research. However, it is an artificial framework that does not necessarily correspond to the human experience of music. Furthermore, rhythm description does not solely call for the handling of timing features, a multiplicity of different features are relevant to the task (e.g. intensities, pitches).

Third, rhythm involves two dichotomic aspects, there are both a heavy structure and inexact timings. It is difficult to describe rhythm as there always exist distortions between when a musical event should occur and when it actually does. Indeed, inexact timings always occur because of expressive performances, sloppy performances and inaccurate collection of timing data (e.g. onset detection may have poor time precision and suffer false-alarms).

Fourth, the structure of rhythm is relatively complex. Hearing music always involves the perception of diverse pulses (or metrical levels) that coexist in music. One may focus attention on a short and rapid group of notes, or on a larger time span.

Fifth, there is still not a complete agreement on what the rhythm description task really means, indeed, explicit representational elements of rhythm are still the object of active research. For instance, the ambiguity of tempo perception makes it a difficult process to model.

Sixth, beats are strongly correlated to onset times. However, beats do not necessarily line up exactly with them.

1.4 Research methodologies regarding rhythmic aspects of music

Rhythmic aspects of music have been studied for many centuries, within many disciplines and by means of different methodologies. In order to roughly order the bulk of research, the following methodologies can be identified:

- Theoretical analysis
- Psychological studies (experiments with human subjects)
- Computational modeling

Theoretical analysis Theoretical analyses of music formulate theories about the syntactic structure of music. Rhythmic aspects have been widely studied via this research paradigm. Music Theory accounts for many contributions to the study of rhythm and musical time, see e.g. [32], [93], [143].

Psychological studies Some researchers point out that the rhythm concept should not refer to intrinsic properties of the musical notation system, but should rather refer to the experience aspects of listeners. As written by Honing [77]:

> "Most theories agree that there is more in music than what is written in the score. [...] The question here is whether a piece of music resides in the notation, in the air, or in the people's minds, or in other words, whether music is cognitive or not."

Hence, a very important body of work is concerned with psychological studies of time apprehension. The aim in this field of research is to explore the role of *memory* and *perception* and their interaction in time apprehension. This usually entails empirical investigations: an hypothesis is formulated, stimuli are designed in accordance with this hypothesis, listeners are selected, a procedure is defined (apparatus, indications to listeners, etc.), data is collected and analysed in order to accept or reject the hypothesis. See e.g. [54], [27], [38, Part III].

Computational modeling One may consider the computational modeling paradigm from a pragmatic viewpoint: indeed, it entails the building of computer tools that can prove very useful in application contexts (see on page 7).

From a more scientific point of view, one may consider several formalisms. For instance, embracing Signal Processing or Artificial Intelligence formalisms usually entails respectively either a bottom-up or top-down approach. In the former field, researchers focus principally on properties of the musical stimuli (and not so much on the musical notation system or the experience aspects of listeners). In the latter field, researchers mostly tackle the issue of rhythm description by building computer models of our minds. This difference echoes the discussion existing in psychological researches regarding whether the apprehension of rhythm would be a physiological or a cognitive process.

In sum, we believe that it is important, while seeking computational models of musical rhythm description, to distinguish between the aim to extract what would be phenomenally present in the signal vs. the aim to identify in the signal the musical constructs that could exist in our minds.

Chapter 2

Review

2.1 Representing musical time

Imagine the following musical scene. Somebody (or some machine) is making music: musical events are generated at given instants (onset times). The first way of representing the rhythm one would probably think of is to specify an exhaustive and accurate list of onset times, maybe together with some other musical feature characterizing those events as e.g. durations, pitches or intensities (as is done in MIDI, see on page 18). However, the problem to this representation is the *lack of abstraction*. There is more to rhythm than the absolute timings of successive musical events, namely:

- There exists an objective, regular, *temporal structure* underlying those event occurrences, one must consider an event timing relatively to the others [90].
- Because it passes through the filter of listeners' auditory systems, this succession of musical events is subject to interpretation and gives rise to a *perceived* rhythm (this is indeed why it was produced in the first place). Part of the structure (at least beats at one level of the metrical hierarchy) is readily perceived [86] and it does not necessarily line up with onset times [15], [42].
- *Timing features* are of very first relevance: they emerge from the opposition between the structure regularity and violations of that regularity. For instance, variations in execution speed (long-time timing deviations), or event shifts while keeping a constant execution speed (short-time timing deviations) [7]. These are very relevant, especially to skilled performances, and must be somehow part of a rhythm representation [27, p.489], [6], [78].
- Further, this succession of musical events generate highly abstract rhythmic qualities in the listeners' minds (*emotive aspects*).

2.1.1 Requirements in a content-based processing framework

Representing rhythm in an abstract manner is a central issue in music content-based processing. Indeed, accessing and processing musical data would seem much easier (and enjoyable) if one could directly handle meaningful (e.g. rhythmic) descriptions of the data, rather than the data itself. However, there is a large semantic gap between the features that can currently be computed from musical data and the rich meaningful descriptions that users wish to handle. Indeed, it turns out to be very hard to automatically derive the semantics of musical data (even in symbolic format), much harder than that of e.g. text data. In fact, it is hard just to precisely define musical contents. As music affects our minds and our emotions, it is difficult to represent explicitly.

Hence, for content-based processing matters, the rhythm must be represented at a high level of abstraction. Building upon this point, let us propose the following list of *requirements for a rhythm representation.* Depending on the meaning one gives to the term "processing" (exploring, comparing, retrieving, transforming, etc., see on page 7), the relative relevance of these requirements may vary.

- Cognitively relevant (the representation is intuitive to the end-user, it facilitates the apprehension of the data semantics)
- Perceptually relevant (it models the data processing achieved in the listeners' perceptual apparatus and therefore intends to replicate the structures that exist in listeners' minds)
- Correct - Discriminative (music that have similar rhythms are represented almost alike whereas the representations of music whose rhythms strongly differ show clear differences)
- Compact (the representation summarizes the data and requires less memory than the very data for storage)
- Precise (important descriptive power, fine details are accounted for)
- Comprehensive (accounts for *all* the aspects of rhythm)
- Powerful (suitable to many purposes or applications)
- Automatically derivable from musical data. There, other important factors are:
  - Computational cost (can the representation be computed rapidly?)
  - Robustness to distortion in the data (does it change when the musical data is slightly degraded, e.g. compression/decompression, adding of background noise, etc.?)
  - Versatility (can it be computed from different input format as e.g. audio, MP3, MIDI, score?)

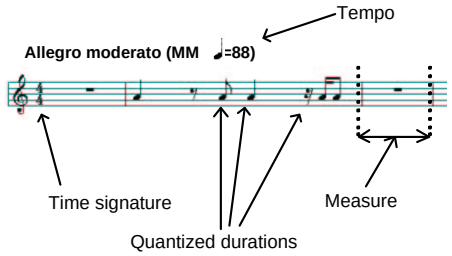

Figure 2.1: Score representation

2.1.2 Representing musical time in MIDI format

The MIDI format provides a means of storing (but most importantly transmitting in real-time) the *Tempo* (i.e. the number of microseconds per quarter-note), the *Meter* (more exactly, the time signature), and the *MIDI Timing Clock*, all of which are dimensions relative to rhythm. The latter permits to represent time by a discrete temporal unit that depends on the notion of tempo (unlike the *MIDI Time Code*).[1] This is a message –the status byte F8– sent from 24 to 480 times per quarter note. If the tempo changes, the MIDI Timing Clocks will pass at a faster rate, but the number of messages per quarter-note will stay the same.

This representation standard is rather designed as an efficient real-time communication method between electronic instruments than as an abstract representation standard. As Honing points out [77], "a distinction can be made between representations designed for real-time systems that are process-oriented [...], and non-real-time systems that have a static global view of the music [...]." Musical events (onset times and other features) can be precisely represented, however, the only step towards a more abstract representation lies in the *Tempo* and the *Meter*. Hence, the MIDI format forgets many of the important aspects evoked on page 16: e.g. the notion of temporal structure, the time interval hierarchy and timing features. Also, it satisfies some of the requirements listed in 2.1.1 (at least compactness, precision being arguable), but many others are not satisfied (e.g. it is not comprehensive and not automatically derivable from audio data).

2.1.3 Representing musical time on a score

We refer here to the common Western music notation. It has evolved over centuries and is nowadays relatively stable. On a score, the rhythmic elements are: the M.M. *tempo* of a reference metrical level (the metronomic level, tempo indication may just be a verbal indication as "Allegro", or a more explicit indication "120 eighth-notes per minute"), the *time signature* (the denominator of which corresponds to the aforementioned metronomic level, while the numerator indicates how many metronome beats make up a measure), *quantized durations* for notes and silences (w.r.t. the metronome) and *bar lines* (*what* metronome beats are measure boundaries), see Figure 2.1 on page 18.

The score representation captures important aspects of rhythm (see page 16), but it does not account for timing features (therefore, it does not entirely satisfy the "comprehensiveness" and "precision" requirements). Further, though clearly a very useful (and widely used) representation, its usefulness in some applications is not obvious. For instance, in the context of browsing and retrieval,

[1] The MIDI Time Code –based on SMPTE Time Code– is a representation of *absolute* time in that it follows hours, minutes and seconds and cannot be speeded up or slowed down.

just as text data is not sufficient for text retrieval in large databases (see e.g. [5]), the use of score data does not allow for direct retrieval of musical information. The score must be further parsed, segmented into coherent indexed structures over which "themes" can be sought –see e.g. [99], [102], [126]. Rules must also be designed to define distances between scores (e.g. [89] address the issue of "transposition invariance"), satisfying the "correctness" requirement depends thus upon those, "correctness" is not inherent to the score representation. The score representation is indeed cognitively-relevant to musical scholars, but much less to the non-trained listener.

"The majority of Western classical music is scored and many of these scores exists in electronic form" [92], this obviously is an important argument for the use of this representation. However, there is an enormous (and augmenting every day) amount of musical pieces that are not available in score format and will probably never be; it is indeed commonly agreed that no computer system can currently achieve the automatic transcription from audio to score (other than in simple cases).

2.1.4 Representing musical time with the GTTM

Lerdahl et al.'s "Generative Theory of Tonal Music" (GTTM, [90]) proposes a formal description of the "musical intuitions of a listener experienced in a musical idiom." The particular musical idiom they focus on is the classical Western tonal music, and they restrict themselves to those aspects of musical intuition that they consider hierarchical in nature. They formalise the listener's understanding of the musical hierarchical structures following a methodology based on generative linguistics.[2] Four main components make up the GTTM: "grouping structure", "metrical structure", "time-span reduction" and "prologational reduction". They propose a "musical grammar", expressed as a set of rules, that detail explicitly each of these components. Of special interest here are the rules relative to the metrical structure. While the grouping structure deals with time spans, the metrical structure deals with durationless points in time: the beats. Beats must be equally spaced: they are defined as the boundaries of the time axis regular division. A division according to a specific duration corresponds to a metrical level. Several levels coexist, from low levels (small time divisions) to high levels (longer time divisions). There must be a beat of the metrical structure for every note. A beat at a high level must also be a beat at each lower level. Beats obey a discrete time grid, time intervals being all multiple of a common duration: the smallest metrical level. Figure 2.2 on page 20 illustrates GTTM metrical structure representation on an audio waveform.

The temporal structure underlying musical events is explicitly handled here. Listeners' experienced aspects are also central to this representation. However, it may be considered inadequate (or insufficient) for the handling of timing features. Indeed, as there must be a beat for every note and as beats must be

[2] That is, seeking a set of rules (a grammar) to model the construction of a set of structure realizations (the subject's knowledge of a familiar idiom).

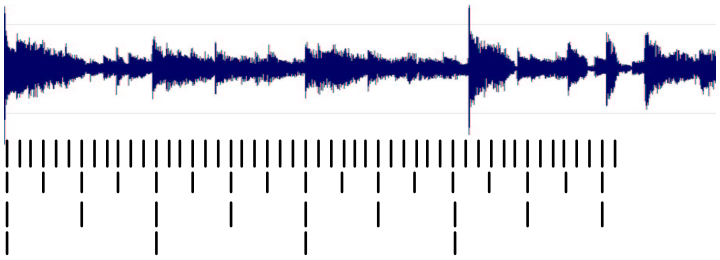

Figure 2.2: Representation of a four level metrical structure corresponding to an audio file

equally spaced, features emerging from violations of the structure are explicitly out of scope.

2.1.5 Representing musical time in MPEG7 format

Current elements of the MPEG-7 standard that convey a rhythmic meaning are the following:

- The *BeatType*
- The *MeterType*
- The note relative duration (*NoteRelDuration*)

The *BeatType* and note relative duration are embedded in the melody description (respectively in the melody contour and in the note, as we can see in Figure 2.3 on page 21).

The *MeterType* is the time signature, it consists in a numerator and a denominator, both integers. Values of the latter are restricted to powers of 2, from 1 to 128. It is illustrated in [31][3] in the description of a melody. It may however be used as a descriptor for any audio segment, but let us stress that it would not be a relevant descriptor at any temporal scope of description.

The *BeatType* refers to the pulse indicated in the *MeterType* denominator. It is a series of integers, an integer being assigned to each note. Assuming a given segmentation in beats (the metrical level is not specified, hence they do not necessarily correspond to beats at the tactus level), each element of the series represents the beat segment index a note falls in. That is, the *BeatType* represents the note quantized positions, with respect to the first note of the excerpt, the positions are expressed as integers, multiples of a timing reference which actual value in seconds is *not given* (this would be the beat period, i.e. the inverse of the tempo).

The note relative duration is the "logarithmic ratio of the differential onsets for the notes in the series" [31].

[3]See also http://www.chiariglione.org/mpeg/working_documents.htm

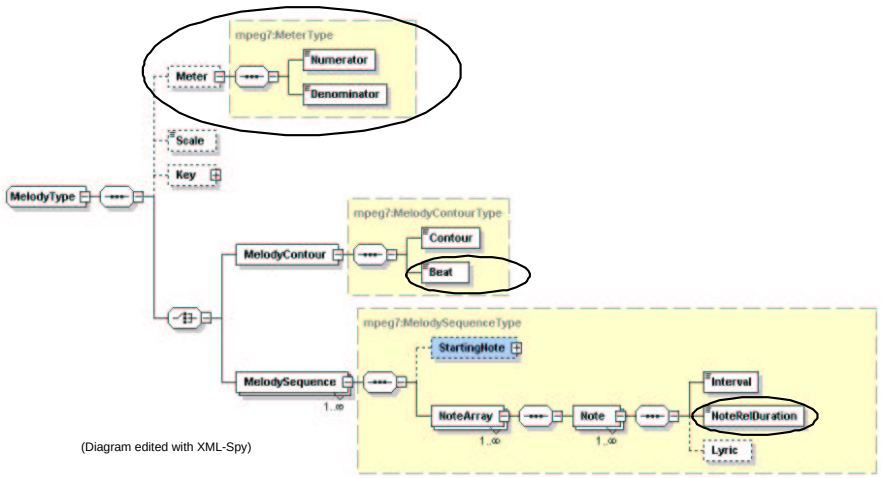

Figure 2.3: MPEG7 elements of rhythm, in the melody description.

This rhythm representation has proven useful for improving query-by humming applications. But let us wonder what, beyond the scope of this application, could be its limitations.

First comes to mind the fact that the context of rhythm description is, in the current standard, that of a monophonic melody, which seems to be a restriction.

The time signature can be represented, but not the bar lines (*where* is the downbeat, the "one" in e.g. "one-two-three-one-two-three-.."). Nor can be represented the symbolic values (e.g. "quarter-notes", etc.) of the notes. Indeed, as events are characterized by which beat they belong to, this is not accurate enough to represent already quantized music where sub-multiples are commonly found (e.g. "eighth-notes"). Therefore a score could not be represented by means of the current MPEG-7 standard.

The speed of execution of a musical piece (as in a performance, and sometimes suggested in a score) is absent from the standard. Also, the exact timing occurrence of each note cannot be represented. Therefore, there would be no possibility to encode a MIDI data stream for instance.

One may notice that a proposal of tempo descriptor has recently been made to the MPEG consortium: the *AudioTempo* [74]. It is not presently part of the standard, but is candidate for a forthcoming version. It is a scalar value assigned to an audio segment, the number of beats per minute (BPM). The *AudioTempo* can be useful as a global descriptor of a piece of music, to account for its global pace. However, the evolution of the tempo is also a very important rhythmic feature, representing it by means of the AudioTempo would entail a segmentation of the music piece at hand in many audio segments whose only reason of being would be their tempo differences. This is questionable. Moreover, the variation of the tempo is a continuous phenomenon, one may wish to envision it with different levels of accuracy, depending on the application. Considering

the assumption of constant tempo being relevant to a given piece, the "phase" of the beat, i.e. the actual point in time where the first beat occurs (in a temporal window length proportional to the inverse of the tempo) is another important feature that is lacking in this proposal. It is indeed completely different to tap one's foot on the beat than against the beat, or slightly ahead or before of it. More important, improving the current standard by adding a single metrical level disregards the fundamental notion of hierarchy in the rhythmic structure of music: several metrical pulses coexist and are tightly related (see 2.1.4). As argued on page 16, a representation scheme for rhythm should account for this property.

When dealing with expressive performance data (audio or MIDI), quantizing a note time occurrence through the use of the *BeatType*, a rounding towards $-\infty$ occurs; thus, in the case where an event is slightly before the beat (as it can happen in expressive performance) it is attributed to the preceding beat, which in some cases could be harmful.

Finally, provided the underlying metrical structure could be derived from a musical performance, the current representation cannot serve for exploring fine deviations from the structure, which would be useful in applications of e.g. performance comparisons.

A last practical point could be made: current elements of rhythm description in the MPEG-7 standard are extremely sensitive to the determination of the time signature, which is still a difficult task for the state-of-the-art in rhythm computational models (see Section 2.2). And no algorithms are suggested for the determination of these features; even though this is not mandatory in the official MPEG references, informative extraction procedures are provided for many other descriptive features.

2.1.6 Representing musical time in our minds

As mentioned on page 16, an important aspect of musical rhythm is its experienced aspect. Listeners perceive and represent in their minds (i.e. measure) rhythm in ways that might differ from the ways it has been notated, thought by a composer, or intended by a performer. Our perceptual apparatus and memory capabilities govern our internal representations ("[a musical event] must be situated in time in relation to surrounding events, and events must be grouped together in order to overcome memory constraints" [45]). Part of the research pursued in the field of music psychology addresses the role of memory and perception and their interaction in time apprehension. They usually entail empirical studies over human subjects.

A widespread aspect of the work done on mental representation of musical rhythm is the intent to determine differences in perception according to listeners' culture, musical background, age or sex [83, pp.53-64, Chapter IV] and [84], [44], [48], [57] and [58]. An alternative rationale is to seek representational elements, or processes, that would stand as "universals" or "innate" (i.e. functioning at birth, independent of environmental influence) [45].

Regularity perception An important aspect of human apprehension of auditory sequences is that we tend to perceive them as regular [86]. Even when the sequences are not regular [97], or do not have rhythmic "intentions" [79], as machine noises, ocean waves, etc. ("subjective rhythmisation" [54, p.155]).
[45] argue that our "predisposition towards regularity" should be regarded as a universal of music temporal processing. They argue that "Processing is better for regular than irregular sequences. We tend to hear as regular sequences that are not really regular". We would also "spontaneously search for temporal regularity and organise events around the perceived regularity".
This regularity does not necessarily correspond to a "physical tempo", that would be a property of the sequence, or to the tempo notated at the beginning of a score [46].
Another important finding is that the perceived beats do not necessarily line up exactly with onset times [15], [42].
Some researchers propose that this regularity perception would be consistent over time [30], [29, p.188] and also independent of musical training [91], arguing that our memory would store *absolute* tempo. Others rather consider that this would be true for few special cases (as when the sequences are well-known, or for a relatively restricted number of persons, as e.g. professional musicians), the general case being that regularity perception would be an unstable feature, relative to many factors: age, musical training, musical preferences, general listening context (as e.g. tempo of a previously heard sequence, subject's activity, instant of the day), etc. [83, 84], [47], [44], [48], [101].

Preferred tempo This regularity perception necessarily has upper and lower boundaries (e.g. [103] proposes 1500 and 200 ms, respectively). They are imposed by the mechanical capacities of our perceptual apparatus and our short-term memory limits. [110, p.424] refers to an "existence region of pulse sensation." In between the boundaries, durations are all possible candidates to a perceived regularity, but *with different probabilities*. That is, we consider tempi with some "a priori" preference, this, independently of the auditory sequence. Commonly, a tempo preference distribution is modeled as a unimodal distribution (e.g. a Gaussian) with a maximum, for [54] and [110, p.438], at 600 ms. [103] proposes a resonance curve (as that of a physical resonator) with a resonance period of 480 ms. Some also propose to consider multimodal distributions [46, p.201].
[45] argue that a "temporal zone of optimal processing" around 600 ms may be considered as another universal feature.

Hierarchy perception Listeners can perceive more than one regularity. They have a sense of hierarchy when listening to a complex auditory sequence. The Dynamic Attending Theory [80, 47] proposes that listeners spontaneously focus on a "referent level" of periodicity, and they then can later switch to other levels to track events occurring at different time spans (for instance, longer-span harmony changes, or a particular shorter-span fast motive). In accordance with

the GTTM (see 2.1.4), these levels share harmonic relationships and beats at a high level are also beats at lower levels.
Therefore, listeners perceive, at least, part of the objective temporal structure. However, here again, this is strongly dependent on musical training [47]. Related to that, [45] advocate a universal "predisposition toward simple duration ratio", they claim that "we tend to hear a time interval as twice as long or short as previous intervals".

Perception of deviations Psychological research found evidences that listeners also perceive performers' intentional timing deviations. [25] showed that our "categorical perception" mechanism permits us to differentiate expressive timing from rhythmic structure: we would characterize with a small number of categories the *continuously* variable temporal transformation of the structure (*discrete* and based on integer ratios).
Further, timing and structure are tightly linked. [113] confirms listeners' sensitivity to timing deviations, but, most importantly, also shows that this sensitivity is a variable of the position in the metrical structure. Complementary to this finding, there is strong evidence that performers do not produce timing deviations at whatever point in time [108]. They rather deviate from pure mechanical performance in specific ways (the metrical structure provides "anchor points" for timing deviations, "every aspect of musical structure contributes to the specification of an expressive profile for a piece" [27, p.492]), and in systematic ways (expressive timing in repeated performances can be very stable over a period measured in years, as shown in [29, pp.181-187]).
Some particular patterns of timing deviations have been studied. For instance, the typical slowing down at the end of phrases in Romantic music. [96] provides a study of "groove" perception, and [56] detail what in a Jazz ensemble performance defines the swing.

Other percepts There are undoubtedly other perceptual dimensions of rhythm. Gabrielsson [57, 58] identifies subjective dimensions of rhythm from various perceptual tests. Multidimensional scaling permits to reduce the inherent dimensionality of similarity ratings, and verbal descriptions are used to qualify the resulting dimensions: "meter", "rapidity", "tempo" and "uniformity-variation" (or "simplicity-complexity"), then "forward movement", "length" and "accent on the first beat" would be specific to monophonic rhythms, and "basic pattern" and "movement character" to polyphonic ones.
[123] advocate the "complexity" as another relevant perceptual feature of musical rhythm.
[135] propose to consider the "beat strength" and "loosely" define it as "the rhythmic characteristic(s) that allows us to discriminate between two pieces of music having the same tempo." It resembles the "pulse metric" proposed by [118]. Both features characterize somehow the fuzzy notion of "rhythmicness".

2.1.7 Section summary — Metric structure vs. timing

There seems to be a consensus regarding rhythm representation matters in that, in addition to absolute timings of musical events, the metric structure as well as timing features must be represented, somehow. (More arguments along this line can be found in e.g. [26], [78], [27].)

However, providing *explicit* representational elements for these concepts is still object of active research. Here, there is a lack of agreement caused by the diversity of research paradigms: some address this problem aiming to extract what is phenomenally present in musical signals —rhythm as auditory events— while others aim at identifying the musical constructs that exist in our minds — rhythm as mental representations. For instance, the rate at which beats pass in time is quite an ubiquitous feature in the literature. However, as some consider it as an objective feature and others as a subjective construct, researchers do not always refer to the same concept when writing about "the beat."

The lack of agreement regarding representational elements may also be due to the diversity of interests. Indeed, representing musical time is the act of making explicit *part* of the temporal information conveyed in a musical signal (either in terms of entities or in terms of processes —see "declarative vs. procedural knowledge" [77]). It necessarily entails a preferential point of view: focusing on some aspect, some other information is "pushed into the background" [77]. Therefore, useful representational elements might differ w.r.t one's interest in the musical communication chain (composition, performance, listening, selling or buying, etc.).

2.2 Computational models of rhythm description

The chief goal of a rhythm description model is the automatic parsing of auditory events that occur in time (i.e. one after the other with specific timings) into the more abstract notions of metric structure and timing features (see Section 2.1). The metric structure covers the whole set of metrical levels and their relationships (in accordance with the GTTM, see page 19), while timing features refer to any distortion with respect to this structure (both are illustrated in Figure 2.4 on page 26, where the particular timing feature is a time warping of the structure, i.e. a slowing down of the execution speed).

Particularly, in the framework of music content processing, computational models of rhythm description provide the representational elements (metric structure elements and timing features) over which content processing is achieved (as depicted in Figure 1.1 on page 7).

If the notions of structure and timing are quite consensual, their constituent features are less so. This is a problem because computer programs demand precise definitions, and these concepts are not explicit and accurate enough to define the aim of a computational model (the same concern regarding computational modeling is expressed in [110, p.423], [24, p.19] and [114, p.12]). Given a musical signal, how many metrical levels should one focus on? Is there one level

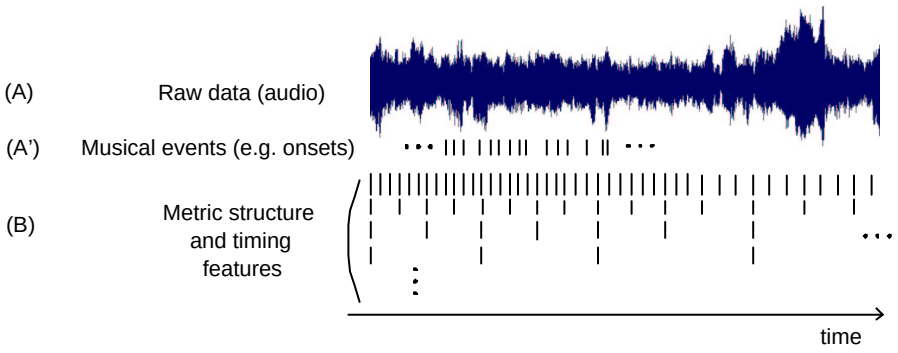

Figure 2.4: Example of an audio signal, a list of its onsets and its corresponding metric structure

more important than the others (is there solely one *perceptual* metrical level)? Which metrical level should one focus on to define the tempo of the music? Which are the two levels that define the time signature? What are relevant categories of timing deviations?

In other words, if it is clear that describing the rhythm is, somehow, deriving (B) from (A) in Figure 2.4 on page 26; what elements of (B) are relevant, what can these elements be called and how can they be clearly defined?

These are mainly unanswered questions. However, in the course of providing a clearer problem definition, many computer programs have been implemented that derive the beats and the tempo of *one* metrical level from a musical performance, the metrical level being usually loosely defined. Others aim at deriving complete rhythmic transcriptions (i.e. scores, see 2.1.3) from musical performances. Also, some programs aim at determining some timing features from musical performances, namely, tempo changes, event-shifts (or short-time timing deviations, tempo being constant) and swing factors. Although aiming at different goals, all these computer programs share some functional aspects. For instance, a prevalent aspect of these computer programs is the handling of symbolic data, i.e. event lists, instead of (or derived from) raw audio data. Those event lists are usually made up of onset times (see (A') in Figure 2.4 on page 26), some models also handle other features (temporal, timbral, harmonic or melodic). Therefore, we propose to organise this review of computational models w.r.t. the following functional aspects: event-list creation (e.g. onset detection), pulse induction (including periodicity computation and handling of event-shifts), pulse tracking, time signature and quantized duration determination and finally swing estimation. See Figure 2.5 on page 27.

2.2.1 Event list creation

Part of the models deal with symbolic data, such as MIDI or manually parsed scores (e.g. in the format of files containing solely onset times and durations

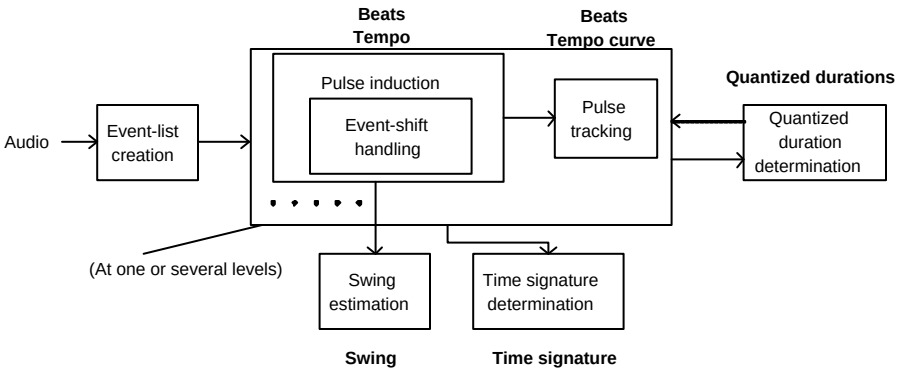

Figure 2.5: General diagram of rhythm description computational models

[12], [95]). Dealing directly with audio data was difficult when the first computer models were designed (e.g. [95]), because e.g. of storage requirements, hence the handling of symbolic data, presenting musical information in a compressed form. However, at some point in computer hardware progress, it became possible to manage large amounts of audio data as easily as symbolic data. Currently, recent models tend to deal directly with acoustic signals (early intents should be noted though, in the models designed at the Center for Computer Research in Music and Acoustics (CCRMA) at Stanford University [23] and [119]). Either starting from MIDI, other symbolic formats, audio data, or some compressed audio format (as MP3 [140]), the first analysis step is the creation of an event list, i.e. the parsing, or "filtering", of the data at hand into a sequence of symbolic events. These events are assumed to convey the predominant information relevant to a rhythmic analysis (see Figure 1.1 on page 7).

In this step, monophonic excerpts are often parsed into sequences that resemble note features (with e.g. an onset time, a duration and a pitch). Regarding polyphonic music, one can intend to separate instrumental streams (a very challenging process) and build an event list for each monophonic stream (as proposed by [3]), further rhythmic analyses of these streams can be merged in a subsequent step. This is not the only possibility though, one can also describe a polyphonic excerpt by a single event list, events ("summary events" for [114, p.29]) representing a global (polyphonic) view of musical chunks (e.g. chords, energy components).

Onsets The processing of onset times for rhythmic analyses is ubiquitous in the literature. Most systems account for onsets. Musical event occurrence instants are indeed thought to be a very important cue to rhythm perception. Onsets can be extracted (with more or less reliability) from virtually any musical format. For instance, [94] processes onsets manually parsed from scores. They can also be easily parsed from MIDI data (see e.g. [35], [18], [20] and [112]).

More complex is their automatic extraction from audio signals. Onset[4] detection is an old research topic in the Signal Processing community. Providing a review of onset detection techniques is clearly out of the scope of this dissertation. Let us solely provide a few pointers of interest: for instance, early algorithms tuned explicitly for musical signals can be found in [23], [119], other more recent and interesting algorithms can be found in e.g. [131], [9], [87] and [49]. An example of model dealing solely with onsets extracted from audio signals is [40].

Durations In addition to onset times, some systems also handle durations (or IOIs as a substitute). For instance [12] and [95] parse durations from scores. Building upon Mont-Reynaud et al.'s model [104] that processes onset times, [33] and [3] models also use durations in creating event lists. They parse MIDI onsets in two groups: "weak" and "healthy" onsets, the model then solely processes the latter. "Weaks" being those onsets whose duration is either shorter than some fixed threshold (20 to 50 ms for [3]) or much shorter than the preceding one. Chung's "note importance agencies" [24, pp.61-62] parse MIDI data into note onsets and durations (Chung argues however that these "agencies" are not restricted to such information, they could "be influenced by every aspect of music that can affect beat perception", e.g. loudness, repetition of melody). [110, p.426-432] weights onsets proportionally to their subsequent IOI, this is achieved by the use of a "saturation function" that is perceptually justified (in [110, p.426], loudness, timbre and pitch are also promoted, but the model's implementation depends solely on IOIs).

Of interest here, [127]'s perceptual experiments show that solely time information (onsets and durations) are sufficient for the perception of a pulse (in Ragtime music).

As onsets, durations can be easily parsed from MIDI or scores, but more difficultly computed from audio data (all the more if it is polyphonic). It is often calculated as the (more easily computable) IOI, considered a rough equivalent [12].

Timing patterns Mont-Reynaud et al. propose to define musical events as patterns of note timings [104]. Dealing with scores, [104] demonstrate the difficulty of the pattern elaboration process and its inherent multiplicity of solutions. It is therefore difficult to envisage using this rationale when using MIDI data or audio signals (all the more if polyphonic).

Amplitudes Smith et al. [125, 124] handle lists of onset times extracted from monophonic drum MIDI signals, weighted by their amplitudes (i.e. MIDI velocity). Similarly, [59]'s model assumes that the perceptual accentuation of rhythmic events in music lies in inter-onset intervals, and dynamics in the input sequence (input pulses have different amplitudes). Building upon [40]'s system, [43] also considered onset amplitudes together with their positions.

[4]also referred to as transients, or abrupt changes

Amplitudes are easily computable from MIDI data and audio signal (provided a good onset detection), but are absent from the score notation.

Pitches For the analysis of piano melody lines, [23] extract from audio signals note onsets, durations and pitches. This data makes up the "acoustic maps" that are subsequently processed in order to determine complete score transcriptions of musical performances. [41] include duration, amplitude and pitch in the computation of the "salience" of musical events; these features are parsed from MIDI data (when events are made up of several notes, the longest duration, the amplitude summation and the lowest pitch are kept as representative).

Pitch is easily readable in scores and MIDI data. However, depending on the musical excerpt (monophonic or polyphonic, with specific timbres), it can more difficult to derive from audio data [62].

Chords Dealing with MIDI data, [114] defines two types of events, collectively called "summary events": single events (i.e. notes) and chords as collections (i.e. 2 or more) of simultaneous notes. Pitches do not seem to be of first relevance here, the importance is rather given to whether there is one or more notes at a specific point in time. Also for MIDI data, [40] enhances a previous system [41] by the addition of a chord representation, the "simultaneous note density". [66] focus on drumless polyphonic audio signals. A front-end performs onset detection and frequency analysis, yielding a harmonic transcription of the audio.

Just like pitch, chords are easily readable in scores and MIDI data, but much more hard to derive from audio data.

Percussive instrument classes Designed for the analysis of audio recordings of percussion instruments, Schloss's system [119] differentiates between "high drums" and "low drums". Similarly, [7] proposes to automatically differentiate several conga sounds.

The system by Goto et al. [65, 64] deals with audio signals whose beat is maintained by drum sounds. Here, in addition to onset detection, the front-end performs a discrimination between drum sounds: bass drum or snare drum. The classification is based on a frequency analysis of regions surrounding onsets.[5] Similarly, [106], [67] and later [144] ground rhythmic analyses on the automatic extraction from audio signals of bass drum and snare drum occurrences. The issue of drum timbres classification in polyphonic signals is addressed by [73]: the authors argue that drum timbre characterization should be relative to musical excerpts rather than absolute and depending on predefined clusters in a "freezed and universal" timbre space. Therefore, they discriminate occurrences of snare-like and bass drum-like timbres by means of non-supervised clustering techniques. The determination of two series of indexes is based on the progressive identification of the source sound (the percussive sound to find) during the analysis process. More precisely, templates of synthetic sounds are refined

[5] It can be seen as a generalization of the spectrum low-frequency component detection explained on the following page.

within several iterations, in order to most closely match the actual recurrent percussive timbre of the audio signal. Unfortunately, it should be noted that both [65, 64] and [73] provide poor evaluations of the drum-sound classification issue.

Percussive events can be extracted from MIDI (MIDI channel 10 is normally used for such events). Their extraction from audio data is still ongoing research. Isolated samples can be automatically classified with a high reliability [75], however, the state-of-the-art in percussive events recognition in polyphonic mixture leaves room for improvement [67], [73], [144], [51].

Frame features In [117], Scheirer criticizes the "transcriptive metaphor" according to which music perception would consist in a hierarchy of tasks, the first one being the transcription of audio data into an accurate note-list representation to be analyzed. A central argument to Scheirer's discourse is that modeling very low-level procedures of our perception should not entail symbolism, he maintains that the first stages of a perceptual mechanism modeling should not handle abstract symbols (such as note accurate durations and pitches, or chords) neither as a starting point (when dealing with e.g. MIDI data) nor a mid-level representation (when dealing with audio). In [77], Honing comments that "there seems to be a general consensus on the notion of discrete elements (e.g. notes, sound events or objects) as the primitives of music. [...] but a detailed discussion and argument for this assumption is missing from the literature." In his opinion, when building a system that in some way mimic perceptual capabilities, on must put a special focus on the issue of what in this system is chosen to be innate or learned. Honing argues that "a distinction has to be made between [...] the existence of possibly innate perceptual mechanisms and learned divisions of continuous time." Precisely, [117] argues that solely well-trained musicians hear the music in terms of its conventional musicological structures, and that the "transcription" assumption may not be relevant regarding the actual perception of music by the human auditory system. In [116], Scheirer aims at confirming this hypothesis with a particular musical aspect of interest here, the tempo. He presented to listeners an amplitude-modulated noise constructed by "vocoding a white noise signal by the subband envelopes of the musical signal" instead of the musical signal itself. In this synthesized signal, the notion of onset features are purposely filtered out. From his conclusions, the sense of tempo was similar in both the cases.

Along this rationale, some models do not focus on note onsets and onset features, but refer to a data granularity of a lower level of abstraction: frames, i.e. chunks of audio.[6] Polyphonic audio is handled in the very same way as monophonic. The frame size is of particular relevance here, frames are usually multiplied by a window and a particular hop size is defined.

Energy Regarding energy feature computation, several rationales can found in the literature. Some systems simply extract frame energy, in the time or fre-

[6]This approach is not suited to score or MIDI data analyses.

quency domain; events thus correspond to one value per frame.

Others (as [2] and [8]) compute Fourier representations and derive one event per analysis frame in focusing on the energy in low-frequency components (thus implementing the assumption that rhythm is mainly communicated by instruments such as bass drums or bass guitars, in a somehow more simplistic way than the methods mentioned on page 29).

Others decompose the signal in several subbands, compute energy in each subband, then optionally postprocess them (e.g. assign them different weights) and *sum* them back; here also, events correspond to a value per frame. This is the case of [135] and [136] (10 ms-long frames and no overlap for the latter).

Finally, another procedure is the computation of energy in frequency subbands, there are thus as many events per frame as there are subbands, e.g. 6 for [74][7] and [140] (in the latter, MP3 bitstreams are processed, hence frames are 13 ms-long and there is no overlap, the correspondence between subband frequency interval and MDCT coefficients depends on whether short or long windows have been used in the MP3 coding), 20 for [109] and 23 for [122]. Here, frequency subbands are further processed *separately*, an integration of their different processes is achieved at the end, [116] precisely argues that:

> "A rhythmic processing algorithm should treat frequency bands separately, combining results at the end, rather than attempting to perform beat-tracking on the sum of filterbank outputs."

Frame-to-frame energy variation Rather than focusing on single frame energy values (in subbands or the whole frequency range), some systems define events, still at the frame time-span, as some measure of variation of the energy from one frame to the consecutive one. For instance, [53] parameterize signal frames (11 ms-long, no overlap) by the magnitude of their Fourier transform. Then, they compute a similarity measure (normalized product, i.e. cosine distance) between consecutive frames. In [88] the magnitude spectra (10 ms frames, no overlap) is transformed by a compression function (e.g. a hyperbolic sinus), giving thus more weight to high frequencies than low frequencies; then a first-order difference is computed. [82]'s "registral accent" computation follows a process resembling that of [135], aforementioned (i.e. subband decomposition —here 36—, weighting and sum), with the difference that a first-order difference of frames energy values replaces the mere energy value.

In [116], Scheirer also computes the first-order difference of frame energy values, however, this is done in 6 different frequency subbands and, differently from [88] and [82], frequency subbands are further processed *separately* (an integration of their different processes is achieved at the end), there are thus 6 events per frame, not just one. It may be noted that [116] does not explicitly refer to a frame-by-frame analysis. However, in our understanding, the signal envelope

[7] This paper indeed details an algorithm based on a subband integration posterior to the periodicity detection. However, a MATLAB implementation provided on the MPEG7-audio mailing list ftp site by its very authors features a subband integration *prior* to further processing, as [135].

extraction (by convolution with a half-Hanning window) and downsampling is similar to a framing of the signal (with a frame-size equal the window size, i.e. 200 ms) and a step of analysis (hop size) of 5 ms (if the downsampling frequency is 200 Hz; 13.3 ms when downsampling at 75 Hz, value advocated by [116] for reaching real-time performances).[8]

2.2.2 Pulse induction

A metrical level (a pulse) is defined by the periodic recurrence of some musical event. Therefore, computer programs generally seek periodic behaviors in event-lists (Subsection 2.2.1 makes explicit the meaning of "events") in order to select one (or some) pulse period(s) and also sometimes phase(s). This is the process of *pulse induction*. The resulting pulse(s) often serves as input to a *pulse tracker* (whose explanation is the object of Subsection 2.2.3). This division in the processing is motivated by Desain et al. [37] who argue that human perception of pulse exhibits two dichotomic processes: a bottom-up process that enables very rapidly a percept from scratch, and a top-down process (a persistent mental framework) that lets this induced percept guide the organisation of incoming events.

In pulse induction, a fundamental assumption is made: The pulse period (and phase) is *stable* over the data used for its computation. That is, there is no speed variation in that part of the musical performance used for inducing a pulse. In that part of the data, remaining timing deviations (if any) are assumed to be short-time ones (considered as either errors or expressivity features). They are either "smoothed out" (see on page 40) or cautiously handled within the pulse induction process so as to derive patterns of short-time timing deviations as e.g. the swing (see on page 46).

For pulse induction, computer programs either proceed by:

- Pulse selection (page 38): evaluating the importance (or "salience" [110]) of a *restricted number* of possible periodicities (pulses)
- Computing a periodicity function (page 33): generating a *continuous* function plotting pulse "salience" versus pulse period (or frequency)

Models handling symbolic data (parsed from e.g. MIDI or scores) often resort to the former procedure, while models handling finer-grained data (as frame features) often implement the latter procedure.

Inducing the pulse with part of the data In many cases, a hypothesis is made on the maximum duration over which the pulse period can be considered stable (e.g. 5 s). In this case, the induction process serves as a front-end to the tracking process. The resulting pulse (there also might be several candidates) is

[8] One could see in [88] and [116]'s procedures (amplitude envelope extraction and first-order differentiation) a focus on the onsets, but the important point to focus on is that there is no discretization of events performed on the envelope signals (no explicit thresholding and peak-picking), the models deal with continuous data until the final decision stage.

propagated over the remaining data (i.e. for t>5 s) and a process of comparison between predicted beats and actual musical events outputs all the beat positions and a tempo curve (see on page 40). Most systems resorting to the pulse selection method (page 38) process a small amount of data for pulse induction and rather translate the overall difficulty onto the subsequent tracking process; e.g. reporting on potential problems of their induction technique, [3] argue that "[it does not seem] to be a problem since [their tracking] model incorporates a great deal of flexibility." Some systems relying on the computation of a periodicity function also consider it as a first processing stage, previous to the pulse tracking [40, 114]. Those typically use around 5 s of data for pulse induction; additionally, in some cases, some emphasis can also be given to most recent samples (e.g. by multiplying the data with an exponentially decreasing window, or by the intrinsic exponential behavior of a comb filter impulse response [116], [20]'s "tempogram" also implements this feature in its parameter α).

This rationale is suitable for streaming application where one does not know a priori the amount of data to process.

Inducing the pulse with the whole data If the pulse induction process is achieved on the whole data (e.g. an entire audio recording, or MIDI file), a strong assumption is made, namely that the tempo is constant all over. Pulse tracking is simply not addressed in this case. This is suitable to some musical excerpts, but much music violate this assumption. This is typically done by systems that rely on a periodicity function computation.

This rationale is not suitable for streaming applications, but it may be relevant for specific offline applications, where one knows that the the tempo stability assumption makes sense.

Inducing the pulse with no data One could think of fixing the pulse period and phase to some probable values and rely completely on the pulse tracking step to correct the (necessary) error made at this step. This would be in accordance with the notion of preferred tempo (see on page 22) and the Dynamic Attending theory [80, 47]. Some systems restrict tempo seeking to a quite small range (e.g. 60 to 120 BPM). One could think of sampling this range into a small number of hypotheses (e.g. 6 hypotheses sampled every 10 BPM) that would be extended, refined, and eventually discarded, all but one.

2.2.2.1 Computing a periodicity function

Here, the event-list is processed in a bottom-up manner in order to highlight its intrinsic periodicities. To each period (or frequency) in the periodicity continuum corresponds a magnitude.

Possible approximate periodicities, due to short-time timing deviations, can be handled in different ways, see 2.2.2.3 on page 40.

Optionally, once computed, the periodicity function may be multiplied by a tempo preference probability distribution (this is what does e.g. [110, p.439, equation 7]), thus implementing the fact that humans consider tempi with

some "a priori" preference, this, independently of the auditory sequence (see on page 22).

Some methods also let large periodicities affect rationally-related periods (e.g. a τ-periodicity in the event-list contributing to the raising of several peak magnitudes: at τ, $\frac{\tau}{2}$, etc.), thus encoding aspects of the metric hierarchy.

Diverse periodicity functions

Fourier transform A classical tool for highlighting periodicities in Signal Processing is the Fourier Transform. An advantage of this transform is its relative rapidity of computation by the FFT. The system described in [8] is an example that makes use of this tool over event-lists defined by low-frequency energy values. [109] also uses the FFT to detect periodicities in 20 frequency channels.

Wavelets In [125, 124], Smith et al. perform wavelet analysis to explore the concepts of architectonic rhythmic strata. Their purpose is to show that the wavelet analysis is well-adapted to capture temporal organisations at different scales and visualize the hierarchies between the different organisational levels. The choice of wavelet representation is not made to suggest that human perception actually proceeds by means of such signal representation; rather, "the intention is to make explicit that information which is inherent in the rhythm."

Autocorrelation function (ACF) The use of ACFs for pulse induction is widespread in the literature. In [12], Brown computes a sample-by-sample ACF of a sequence of onsets (with a sampling rate of 200 per second), weighted by their durations. She presents results for different values of the integration time (time span for the estimation of one correlation coefficient), an important parameter in the ACF computation, among other factors because it defines the statistical reliability of the estimate [34]. [12]'s results are better for long integration times. Scheirer et al. [118] also compute ACF of "onset trains". In another article, Scheirer argues that summing ACFs computed over several frequency channels is adequate to the modeling of pulse induction [115].

In [136] Vercoe proposes the use of the "Phase-Preserving Narrowed Autocorrelation" for detecting periodicities. The "Narrowed ACF" (NACF) was introduced by Brown et al. in [13]: the computation of correlation coefficients includes terms at periodic lags (in addition to the term at a single lag —ACF case—). M being the "number of added terms", the computation of the NACF coefficient corresponding to the lag k accounts for $(M-1)$ terms of delay k, $(M-2)$ terms of delay $2 \times k$, etc., and 1 term of delay $(M-1) \times k$. The NACF thus encodes implicitly aspects of the metric hierarchy (a $2 \times k$-periodicity has an effect on the correlation coefficient of lag k) and gives better period precision (at the expense of worse time resolution). This is a useful feature for signals that contain close partials. However, in the context of pulse induction (as for [136]'s), one might wonder if "close period" is a potential rhythmic situation

(i.e. should one consider that pulses with close periods may coexist in a musical signal?). It may be noted that in [12, p.1955], Brown recognizes that the NACF was not necessary. In our understanding,[9] [136]'s "zero-phase" feature (i.e. ability to keep time localization normally lost in computing an ACF) lies in the computation of a simplified NACF: the integration time is set to a very small value; this may have an influence on the statistical reliability of the estimate.

Foote et al. [53] propose to compute periodicities ("self similarity") in event list (i.e. there, matrix diagonals); one could also consider collection of events (hence changing the data granularity) by means of different kernels. They propose two ways to derive a measure of self-similarity: performing either sums or correlations of the matrix diagonal elements. Interestingly, the first of these two options can be seen as a continuation of an ACF-based approach, indeed, the sum over the i^{th} diagonal is similar to the (normalized) autocorrelation of the signal frame parameters with a lag i. As the NACF, the latter option goes further and accounts for aspects of the metric hierarchy.

Among other recent models, [106, 67, 144] compute ACFs of series of bass-drum and snare-drum occurrences, as well as a cross-correlation function between these two series. [74] and [135] also implement ACFs.

Sum of comb filterbank outputs In [116], detecting periodicities is achieved by banks of resonators, i.e. comb filters (one filterbank per frequency channel). Each filter has a specific resonance frequency. Scheirer promotes the number of 150 filters, covering a logarithmically spaced frequency range from 3 Hz to 1 Hz (i.e. 60 to 240 BPM), one filter thus corresponding to one possible pulse period. Hence, here, the periodicity function plots filter output power (actually the sum over frequency channels) versus filter resonance frequency. [116] and [115] detail similarities and differences of the NACF and comb filter approaches, see also 4.2.2 on page 65. This method also "encodes implicitly aspects of the rhythmic hierarchy" [117, p.91].

Time interval histogram The use of histograms of time intervals between similar elements in the event-list is also widespread, mostly when events are defined as onset times (with the interesting exception of [104] who builds histograms of time intervals between temporal *patterns*, resembling thus somehow an ACF).

[23, pp.17-19] and [119, p.90] generate a smoothed histogram by associating a Dirac delta function to each IOI, assigning it a weight proportional to its value (i.e. longer IOIs are emphasized) and convoluting them with a "bell shaped curve of appropriate bandwidth." Similarly, [114, p.40] builds a discrete IOI histogram and smear it with a Gaussian curve.

Dixon's IOI clustering scheme [39, 40] is essentially similar to the building of an IOI histogram (figure 3 in [43] makes this similarity explicit).[10] Cluster scores

[9] Interested readers should check the CSound implementation of the "Phase-Preserving Narrowed Autocorrelation" in the "tempest" method for tempo estimation (see http://www.lakewoodsound.com/csound/hypertext/manual.htm).

[10] With the difference that it already includes the parsing into a "ranked list of periodicities."

are first defined by their number of elements (i.e. the number of recurrences of a specific IOI). An adjustment of the scores (and cluster representative interval) then favors rationally-related clusters; in this process also, aspects of the metric hierarchy are encoded.

Seppanen's model also implements an IOI histogram [121, 120]. Its computation is sequential, it is updated at each new event, emphasis being given to most recent ones.

"Periodicity transform" Sethares et al. propose the use of the "periodicity transform" [122]. As the Fourier Transform, this transform also decomposes the signal (here made up of frame energy in a subband) onto a set of basis vectors, but at the difference with Fourier or wavelet transforms, the vectors are not specified a priori, the transform is adaptive in that it defines "its 'best' set of basis elements" from the signal itself. Mathematical developments and examples of basis vectors are provided in [122].

Pulse track matching function Seeking periodic behaviors of the event list can also be done by computing a similarity measure between the list and several pulse tracks. Indeed, pulse tracks are fundamentally periodic. That is, generating many possible pulse tracks and measuring how well (or bad) each one matches the event list. This procedure is foreshadowed in Povel et al. "clock model" which assumes that "people perceive, remember and reproduce temporal patterns by structuring their representation according to an internal clock" [101, p.178], that is, a pulse track. In fact, Povel et al.'s clock model does not entail the computation of a periodicity function, it rather belongs to the "pulse selection" models (see on page 38); indeed, it considers a sampling of the time axis with a basic time unit corresponding to the smallest IOI. Candidate pulse periods hence are few. This rationale suits well the type of musical sequences they consider: parsed scores or artificially created sequences. In these sequences, event and silence positions are *exact* integral multiples of this time unit. Nevertheless, it is of interest to introduce this model at this point as it is easy to extrapolate the computations it entails onto the computation of a periodicity function: one may think of sampling the time axis with a much finer granularity than this time unit, the candidate pulse period incremental step having thus no relation whatsoever with the sequence rhythmic structure, hence generating a continuum of possible pulse periods.

As detailed in [101], there are several ways to compute a pulse track matching function. All are based on computing a matching score for *each pulse track element* and then summing them up. When considering *discrete* event-lists, one way is to focus on *negative evidences* (basically counting the number of pulse track elements that do not match any events, the best match is that corresponding to the lowest value). Another one is to focus on *positive evidences* (the opposite). A third way would *combine* both positive and negative evidences. When events are considered on a *continuous* representation (e.g. when adding some degree of tolerance for onset times, or when using frame energies, etc.,

see 30), negative and positive evidences loose their meaning, the function is basically just a multiplication of the pulse track with the continuous event-list.

"Tempogram" In [20], Cemgil et al. define the "tempogram" to induce a pulse period and phase from an event-list (onset times). It represents in a 2-dimensional space the probability of each pair {pulse period, pulse phase} given the onsets. This probability (*posterior* distribution) is computed in a Bayesian framework, therefore it is proportional to the *likelihood* of the observed onsets under given period and phase hypotheses, weighted by a *prior* distribution (actually flat, i.e. all tempos are equiprobable). For given period and phase, the likelihood is computed as the integration, over all the onsets, of the product between a constant pulse track (with appropriate period and phase) and a continuous representation of the onsets (onsets are smeared with a Gaussian curve). It implements the assumption that a good pulse track is one which matches well all the onsets. The tempogram marginal probability function $p(\omega \mid t)$ (integration of the tempogram w.r.t. the phases) provides a 1-dimensional representation of periodicities resembling those aforementioned, see figure 4 in [20] (Cemgil et al. suggest a resemblance with the response of a comb filter bank and a multiscale representation as the wavelet transform).

In our understanding, computing a tempogram is conceptually similar to computing a pulse track matching function considering all possible pulse periods *and phases*.

Parsing the periodicity function Periodicity representations provide hints on what periodicities are objectively present in the musical signal. They are inherently continuous. As the desired output of the induction process is the value of one (or a few) pulse period, and optionally its phase, another step is needed: resulting graphs must be further parsed in order to actually derive rhythmic information (one, or a list of ranked, pulse period). Usually, this is achieved by a peak-picking algorithm: peak positions and magnitudes can be detected with e.g. an N-point running window method: local maxima are detected at indexes whose magnitudes are higher than that of their direct neighbors (N/2 on the left and N/2 on the right). Peaks must be subsequently interpreted with respect to what periodicities one wishes to identify. As [125, 124] suggest, heuristics[11] are needed to give a *musical interpretation to periodicity representation*, e.g. derive the tactus ("While it is tempting to draw hypotheses for methods of derivation of the tactus by 'ridge-tracing' or the well-formedness of the global continuation of a voice, further research is required to build a model of tactus in respect of perceptual issues").

For instance, [23, pp.17-19] and [119, p.90] explicitly achieve peak-picking on a smoothed IOI histogram, and keep the highest peak, qualifying it as the "important duration" (recognizing that secondary peaks may be significant too).

[11] For instance, consider the maximum peak in a *restricted* region (say, between 61 and 120 BPM [65, 64]).

So does [114, p.41] who sets the tactus to the maximum peak; there, the peak-picking algorithm implements a bias towards smaller IOIs. Picking the maximum peak in the histogram is also implicitly suggested in [104].[12] In [53]'s "beat- spectrum", the pulse period is determined as the maximal peak, also by peak-picking. In [12], the pulse of interest is the downbeat (measure). All the peaks in the ACF are detected and the winning pulse period is that corresponding to the peak whose height is greater than those of all previous peaks and all subsequent peaks up to twice its period.

In some cases (e.g. comb filter, tempogram), a special care can be taken to determine the pulse *phase* (hence all beat positions) jointly with the period. In other cases, the computation of the period entails the loss of time localization (e.g. ACF), the phase has to be computed in a second step. For instance, one can generate pulse tracks (with a fixed pulse period and varying phases) and determine which matches best the event-list e.g. by some distance measure (i.e. computing a pulse track matching function, see 36).

2.2.2.2 Pulse selection

The alternative to the computation of a periodicity function is to select a pulse (evaluating its importance, or "salience" [110]) among a small number of possible pulses. This is usually a simpler procedure than the former, and it is most of the time considered jointly with a subsequent tracking.

For instance, [95] consider the first two events as the first two beats. [33] refer to this process as "creating a predictor." They achieve it as follows: considering sequentially the first events, a pulse is defined (and the induction process stopped) as soon as the time occurrences of 3 successive events are in "quasi-arithmetic progression" (a "rhythmic alignment is noticed"). The pulse period is then defined as the time difference between two of these successive events and its phase as the position of the first of these 3 events. [3]'s system must be given the metrical value of the first event. The initial pulse period (and incidentally phase) is then derived from this value and the time difference between the first two events (i.e. the first IOI).

In Chung's model [24], a number of pulse periods and phases are derived from the event-list in a sequential manner. Similarly as Longuet-Higgins et al. [95], the first two events are considered as potential beats, their IOI being the pulse period. Then, the next event is considered in the light of this potential pulse: if its position is an integral multiple of the period (with some tolerance, see on page 40), then the system jumps to the next incoming event, otherwise, a second potential pulse is created (its period being set to this new IOI, the phase being specified by the last two events). This process keeps on as the event list unfolds, each pulse being propagated onto the list and the number of potential pulses logically increasing. Limiting the number of pulses is achieved by assigning to each pulse a score depending on: the "importances" (i.e. durations) of its constituent events, the timing deviations that have been considered in its

[12] This is not explicitly stated, but probable considering subsequent work by Mont-Reynaud [33].

propagation and the number of syncopations. Solely highest-scored pulses are selected. The actual number of events necessary to achieve the pulse induction process is not specified in [24], however, Chung argues that "[g]enerally, after a few bars, simple beat-levels will have been created for all of the IOIs [...], and unless a significantly different interval appear, no new simple beat-levels will be created" (p.77). In sum, [24]'s selection of pulse resembles [33]'s, improvements being that it is not restricted to *three* events in "quasi-arithmetic progression", and that more than one pulse are considered.[13]

Another example of pulse selection is that implemented by Parncutt [110, pp.433-436]. In this model, pulse period and phase are measured w.r.t. a "basic time unit", that is, the shortest IOI, and the considered patterns are cyclically repeating. For instance, for a Waltz pattern (half-note, quarter-note, half-note, quarter-note, etc.), the time unit is set to the quarter-note and the cycle length (C) is 3 time units. In our understanding, candidates for pulse period and phase are integral multiples of the time unit. Therefore, there is a restricted number of candidates, namely, C pulse period candidates (each of which having C phase candidates) —in the Waltz example, the total would be 9. No period continuum is considered. [110, equation 4], the "pulse-match salience", implements a process of matching between onsets and elements of the pulse track candidates that resembles a pulse track matching procedure that focuses on positive evidence rather than negative evidence (see on page 36), with pulse track lengths set to C cycles (pulse track elements having unity amplitudes and onsets having *specific weights*). As previously argued on page 36, this computation of the "pulse-match salience" seems difficult to adapt to non strictly metronomical and non cyclically repeating musical patterns (as are [110]'s stimuli). Indeed, in such cases, the "basic time unit" would be difficult to determine. However it is interesting to remark that one may overcome this difficulty by sampling the time axis with a much finer granularity than this time unit, the candidate pulse period incremental step having thus no relation whatsoever with the sequence rhythmic structure, hence generating a continuum of possible pulse periods. [110, equation 4] could be replaced by an actual inner product, dependency on C could be overcame by considering a maximum period, set a priori. One may also remark that this would accentuate the conceptual similarity between the "pulse-match salience" and the aforementioned "tempogram." The difference is still that the former does not entail a continuous representation of onset times and does not apply weights on pulse track elements.

[13] This *sequential* pulse induction mechanism could be thought of as some kind of tracking. But, In the words of the author, there is no tempo tracking [24, pp.61 and 87], the tempo is considered constant. Chung argues that if there were some tempo or time signature changes, his model would eventually discover it, as the pulse induction process is always running (previous "winning pulse" scores would diminish), but "there is no mechanism for expecting repeated changes" (p.88). This would finally result in a some *switching* of agents (and agencies) rather than in a proper *evolution* of a single agent.

2.2.2.3 Handling short-time timing deviations

In contrast to continuous event-lists (e.g. frame energy list), the spiky nature of discrete event-lists (e.g. onsets) forbids the consideration of approximate periodicities (due to short-time timing deviations that always exist in any musical data other than parsed scores and artificial sequences) and of artifacts in the list creation. Indeed, in discrete event-lists, values are not related to their direct neighbors whatsoever. To overcome this dead end, elements of discrete event-lists are usually considered with a "tolerance interval" [94].

For instance, [40] considers a tolerance interval of 25 ms (the "cluster width") for IOIs, [43] considers tolerance intervals proportional to IOIs, so that long IOIs allow for greater variations. Similarly, [24, p.65] considers durations with a 30% tolerance, allowing thus more variability to long durations. In our understanding, in [121, 120], the quantization of the IOI histogram into a specific number of bins is also similar to the previous procedures. The exact number of bins, thus the tolerance interval, is not given in the original documents. A tolerance interval can also be considered in the very creation of the event-list. [114, p.29]'s "summary events" can represent a polyphonic view of musical chunks, two isolated events are merged into a single one (a chord) if their onset times are equal, with a timing tolerance of 10 ms. Similarly, [41] use a tolerance of 70 ms to define onset simultaneity.

The previous procedures can be interpreted as smearing the event-list with a rectangular window (e.g. by convolution). It does permit to consider short-time timing deviations, however, the resulting representation is still discontinuous (the series of Dirac has been transformed into a step function). This can be improved by using a smooth curve for smearing. For instance, among others, [23], [119], [114], [20] make use of a Gaussian window (they do not precise its variance value, but [20, equation 10] propose a procedure to estimate it from the data). In our understanding, [128]'s coding of durations is comparable to the use of a triangular window for smearing a list of durations. [33, p.245] suggest the use of an exponential window.

Some expressiveness timing features lie in short-time timing deviations. Therefore, instead of "smoothing them out", one may think of handling them cautiously to derive *patterns* such as e.g. the swing (see on page 46).

2.2.3 Pulse tracking

Pulse tracking models are often seen as complementary to pulse induction models ([33] refer to "adjusting the predictor"). A fundamental difference between tracking and induction lies in the handling of time deviations. The latter handles short-time timing deviations and the former handles long-time timing deviations (e.g. tempo curves). While the latter considers timing deviations as some noise to be eventually smoothed out (see on this page), deviations are part of the very model in the former case. That is, the goal is to determine changes in a pulse period and phase, assuming from the very outset that this pulse is probably unstable.

Pulse tracking models follow top-down approaches and the processing in done online (thus opening the way to real-time implementations): that is, previous data provides pulse period and phase (evidences) that are used as predictions propagated onto incoming data, tracking is then a process of reconciliation between these predictions and observed features of incoming musical data. A suitable framework to describe and compare pulse trackers is to consider them as systems defined by:

- A set of state variables
- An initial situation (initial values for these variables)
- Observations (incoming data)
- A goal situation (explaining observations to the best)
- A set of actions (adapt the state variables in order to reach the goal situation) and methods to discriminate good and bad actions.

Tracking a pulse involves the notion of entrainment, and accounting for some noise in the data is inherent in such a process. Different models implement a different balance between *reactiveness* and *inertia*, that is, in adapting the state variables, they give different importances to incoming data and past evidences (context).

Diverse formalisms have been used in the design of pulse trackers: rule-based, problem-solving, agents, adaptive oscillators, dynamical systems and Bayesian statistics. In the remainder of this section, we intend to provide an overview of how diverse models (following diverse formalisms) deal with the adaptation of state variables to the observations.

2.2.3.1 Observations

Observed musical events are usually onset features: onset times, durations (or IOIs) and dynamics. We are not aware of models accounting for different events, as those listed in Subsection 2.2.1.

Tracking models follow two different rationales regarding observations. They either *consider events sequentially* (i.e. an observation is noticed at each incoming event) or *consider predicted beat positions* (i.e. some events might be disregarded, the important ones are those around predicted beat positions).

2.2.3.2 State variables

State variables usually account for:

- The pulse period
- The pulse phase as either the current beat position or the first beat (or both)

Some models also include:

- A metrical position
- A performance measure, or score

2.2.3.3 Actions

Adaptive oscillators predict the next beat position as the current beat position plus the pulse period, they then choose the closest event (there, onset) to this predicted position and adapt the state variables accordingly [85], [100]. For instance, Large et al. [85] aim at constructing networks of basic oscillatory units, or resonators; these units having the principal feature to "embody the notion of metrical pulse, or beat." A simple oscillator, called the "driven" unit, embodies the period and phase variables and will adapt to incoming events emitted by the "driver" unit. Each event from the driver perturbs the phase of the driven to an amount determined by a coupling strength. This coupling monitors the aforementioned balance between reactiveness and inertia of the model. The resulting instantaneous period of the driven eventually differs slightly from its preferred period without coupling. However, in this "phase-pulling" scheme, if the driver stops (i.e. no more input to the driven), then the driven instantaneous period recovers its previous value. The stability of such a system is function of the driven/driver period ratio and the coupling strength ([85] provides insightful diagram illustrations). In order to prevent the oscillator from returning to its former period if the driver stops, a "frequency-locking" procedure is also needed. Phase and frequency locking is achieved by minimizing the gap between the current beat prediction and the actual subsequent event of the driver, according to the method of gradient descent. If this gap is too big, the new event will not be taken into account. Several coupling values are tested and the results are detailed in [85]. [100] and [85] agree on the fact that, ideally, several oscillators should be connected in a network and interact in some way, in order to model several metric levels jointly (on this issue, see [59] and [50]).

In the rule-based approach (e.g. [95], [37]), state variables are pulse period as well as first and current beats. A set of "if-then" rules previously defined adapt these variables considering events sequentially (as soon as an event is observed) as well as considering one (just one) predicted beat. For instance, the system detailed in [95] works as follows. A beat is predicted at the current beat position plus the pulse period, the pulse period is then adapted by two rules: "conflate" and "stretch." The former achieves a doubling of the pulse period when an onset is observed on the predicted beat, the latter changes the period if an onset is observed before the predicted beat (then the period is set to the distance between this new onset and the last but one beat). Pulse phase is adapted by the rule "update": if no onset is observed at the predicted beat (neither before it), the first beat is shifted to the current beat and the current beat to the predicted beat (regardless of the fact that there is no onset there). This approach seems rather biased towards reactiveness than inertia;

which seems acceptable considering input data in which noise is solely due to short-time timing deviations.

In [33], incoming events are considered sequentially and solely the pulse period is updated, not the phase. An integer divisor (or multiple) of the pulse period is assigned to the next observation (e.g. 1, 1/2, 1/3, 2, etc.) as the closest metrical position to the actual event position. The resulting deviation then serves to update the pulse period. This updating mechanism depends also on the event position in the metrical hierarchy: i.e. events close to multiples of the expected pulse period have a greater impact on the updating mechanism than other events, e.g. half-periods (see [33] "Confidence" parameter). Finally, the balance between reactiveness and inertia is explicitly monitored by the "Decay" parameter.

Allen et al. [3] propose to add some flexibility to the previous model by fine-tuning the "Decay" and "Confidence" parameters, depending on the musical style. However, observing that this model does not possess the capability to recover after an error, they embrace the problem-solving formalism and introduce the notion of concurrent hypotheses (one hypothesis being a *sequence of states*). In this model also, incoming events are considered sequentially, but no definitive decision is taken at each observation. Rather, the evolutions of several concurrent hypotheses are evaluated with some delay with respect to real-time (i.e. decisions are not taken on the basis of a given state, but on the basis of a sequence of states). In this framework, the number of hypotheses increases with the number of observations, resulting in a "search tree." In addition to the period and phase variables, a metrical position and a "credibility" (performance measure) are now part of the state variables. The tree is restrained to an acceptable size by discarding some hypotheses (i.e. pruning the tree) via diverse methods:

1. Some basic rules implementing simple aspects of musical knowledge (e.g. "quarter-notes must start on the downbeat or the upbeat", see [3])

2. Expand the best hypotheses first: setting a maximum number of concurrent hypotheses and, when this number is reached, use a heuristic function to determine how well each hypothesis performed so far (e.g. sum of state credibilities in that hypothesis —best-first search—), and keep the best ones.

3. Discard (all but one) hypotheses showing duplicate current states. Keep the one with the "best score" (in our understanding, defined by the sum of state credibilities in that hypothesis).

4. Limit the number of likely metrical positions. That is, if an incoming event corresponds to a "complicated" metrical positions (e.g. a triplet in the context of very simple music) with respect to a given hypothesis (pulse period and current beat position), then this hypothesis is unlikely.

Rosenthal's rationale [114] is comparable to that of Allen et al. [3]. A first important difference is that states account for diverse metrical levels simulta-

neously. Several pulses are tracked simultaneously. A second difference is that observations are not defined by sequential events, but rather events around beats (at several levels). Hypotheses are created for each event around a beat (in a rectangular window which width depends on the pulse period, see [114, p.46]). The pruning techniques are comparable to those used by [3], but adapted to the fact that states (and therefore hypotheses) are more complex ([114, pp.57-68]).

Dixon's agents [40] are comparable to the aforementioned hypotheses by Allen et al. [3]. Each agent has a state (state variables are period and phase of a pulse) and a history. An agent history is "the sequence of beat times selected to date by the agent." At the difference with [3]'s hypotheses, and along with [114], observations are events that occur around predicted beats ("around" having the same meaning as in Rosenthal's case: within a window whose width depends on the pulse period). In order to prevent an explosion of the number of agents, Dixon uses a method similar to [3]'s third pruning method (see above).

Cemgil et al. [20] address pulse tracking through the use of a dynamical system, a "metronome model" that updates state variables at each inferred beat. The system is defined with two hidden state variables: the period and the phase of a metronome. Transition from one metronome beat to the next is modeled by a simple set of state equations. Their model is fully determined if the initial state variables are given. To this deterministic model, they add a noise term (a Gaussian random vector whose covariance matrix will be estimated through a training phase) that models the likely tempo variations. Observations to the dynamical system ("noisy metronome beats") are given by the computation of a "tempogram" from incoming onsets (see page 37). The hidden state variables are estimated by means of a Kalman filter (extensions to the Kalman filter are proposed in [20]).

2.2.4 Further in the metric structure

Deriving a complete rhythmic transcription of an audio signal (i.e. producing a score) requires the determination of a reference metronomic level and its tempo (the M.M. tempo), the time signature (the denominator of which corresponds to the aforementioned metronomic level, while the numerator indicates how many metronome beats make up a measure), quantized durations for notes and silences (quantized w.r.t. the metronome) and bar lines (*what* metronome beats are measure boundaries), see on page 18. The first and fourth points (determination of two different pulse periods and phases) has already been addressed in 2.2.2. Let us now address the two remaining points.

2.2.4.1 Time signature determination

Very few algorithms for time signature determination exist. A usual procedure (see e.g. [12]) is to determine the time signature as a by-product of two pulse *period* induction processes (not phase). A periodicity function may be parsed (see on page 37): two peaks being selected, one corresponding to a fast pulse

(the time signature denominator) and a slower one (the numerator). The ratio between the two pulse periods defines the time signature.

2.2.4.2 Rhythm parsing (quantization)

Given the periods and phases of several metrical levels, one may propagate those, generating therefore a metrical grid. Parsing the rhythm of a given onset sequence can be done by assigning each onset (independently of its neighbors) to the closest grid element. In this framework, rhythm parsing is seen as a by-product of the induction of several metrical levels. For Chung [24, p.21], "obtaining the correct metric and rhythmic interpretation are part of the same process." [24] follows this rationale in a system that assumes constant tempo. The quantization is considered within the constraints imposed by the induction of several pulses (the "metric interpretation"). He argues that "rhythmic and metric interpretations proceed simultaneously, but a correct rhythmic interpretation is not strictly necessary for a correct metric one" [24, p.83].

It is easy to imagine that such a method is bound to fail in the case of music with tempo changes. Rosenthal [114] also consider the quantization process as a by-product of the metrical structure determination, however, in his case, this determination involves pulse induction *and* tracking (at diverse levels).

Raphael [112], Cemgil et al. [19] and Thornburg [130][14] argue convincingly that rhythm parsing should not be seen as a subsequent process to tempo tracking, but that these are really intertwined processes, and that a common model should formalize their interdependencies. Indeed, representing an expressive performance with a too simple tempo curve will result in a complicated quantization, and representing it with a too simple rhythm will result in unrealistic tempo changes. Accordingly, [112], [19] and [130] implement models grounded in Bayesian statistics to estimate jointly the evolution of a rhythm process and a tempo process. These models are foreshadowed in [3]'s model (see on page 42) in which a metrical position is assigned to each onset, but they propose a more elegant formalism. [112] and [19] enhance the probabilistic model proposed in [20] in adding a second (discrete) hidden layer corresponding to the rhythm process. [130] also follows the same rationale, however, an interesting aspect of his model stands in the consideration, from the very outset, of the segmentation of audio data. In the framework of rhythm tracking, his model progressively learns how to produce relevant lists of onsets from the audio. In his point of view, segmentation should not solely be seen as a preprocessing to rhythm description matters, it should rather be intertwined with rhythm tracking (i.e. rhythm parsing and tempo tracking). These tasks should be considered jointly: polyphonic audio segmentation (deriving onset times) is a necessary step to be taken to provide data to the rhythm tracker; similarly, rhythm tracking should orient (i.e. provides priors to) the segmentation task. As "in the presence of rhythmic structure, the pattern of musically relevant change points, as note onsets, becomes highly regular", exploiting structure might help to "better adapt

[14] See also
http://www-ccrma.stanford.edu/~jos/mus423h/Bayesian_Approach_Segmentat.html

segmentation to the problem of onset detection."

Raphael as well as Cemgil et al. models account inherently for noise in onset timing. However, one may note that they process MIDI onset data and therefore do not explicitly incorporate robustness to *spurious onsets* as those that always appear as artifacts to polyphonic audio segmentation. In this respect, Thornburg's rationale seems of first relevance.

2.2.5 Swing estimation

In the pulse induction process, short-time timing deviations can be "smoothed out" (see 2.2.2.3) or cautiously handled so as to derive patterns of short-time timing deviations, for instance, the swing.

In [53], Foote et al. suggest that the swing could be measured by inspection of a periodicity function (there, the "Beat spectrum") within the pulse induction process. This is illustrated by the positions of secondary peak with respect to some higher ones on [53, figure 3]. Unfortunately, no extraction procedure is provided.

In [87], Laroche proposes to estimate the swing jointly with tempo and beats at the half-note level (i.e., he defines swing as a "slight delay of the second and fourth quarter-beats"). The tempo is assumed constant. In our understanding, the procedure is conceptually similar to a pulse induction making use of a pulse track matching function (see on page 36), but considering all possible pulse periods *and phases*, like Cemgil et al.'s tempogram (see on page 37). Concretely, many pulse tracks are generated, the winning one is that which matches best the events (here, onsets). However, here, the number of candidate tracks (the search space) is bigger as tracks have a third parameter to be estimated —the swing— in addition to the two previous parameters —tempo and phase. The pulse tracks are no more periodic, they correspond to the timing pattern that we wish to find in the data: a long-short-long-short pattern. The amount of deviation from a periodic track defines the swing ratio.

2.2.6 Section summary and discussion

The computational modeling paradigm calls for systematic evaluations based on some "ground truth." However, we did not provide quantitative comparisons of models here. This is because many papers do not originally provide performances of their models, also, because such a comparison would require a common test database, and researchers usually set up their own databases to test their systems, but most importantly because there is a fundamental issue in the definition of the very problem: the *lack of agreement regarding which explicit features would serve for model evaluation* (see 2.1.7 on page 25).

For instance, it is difficult to compare programs that extract the tempo if their definitions of "tempo" do not explicitly refer to the same metrical level. Indeed, tempo induction models typically make errors in simple integer ratios (as reported by many authors and in Section 4.2), e.g. 60 BPM instead of 120 or 180. The most common errors are tempo doubling and halving, ubiquitous

when dealing with music with duple time signatures (however, factors of three are also commonly found when dealing with triple time signature excerpts). Also, even if existing scores can be taken as "ground-truth" references to the quantization or time signature determination tasks, "correct" time signatures or quantized durations can always be object of controversy ([18] defines music transcription as "the extraction of an *acceptable* [...] music notation" —original emphasis). Additionally, tempo functions are by far the most common timing features in the computational model literature; however, their relevance has been convincingly questioned [36].

Therefore, this review solely provided a qualitative comparison of models, w.r.t. the functional blocks of the general diagram proposed in 2.5 on page 27. Let us provide a short summary below:

A common aspect of all computational models is the handling of event lists, either as a starting point (e.g. scores, MIDI) or as a mid-level representation (e.g. models that process audio). These events (onsets, durations, timing patterns, amplitudes, pitches, chords, percussive instrument classes, frame energy, frame-to-frame energy variations) are assumed to convey the predominant information relevant to a rhythmic analysis. Depending on which events are chosen and their time span, these lists entail more or less "implicit symbolism" [117]. The first stages of human rhythm perception achieve a comparable parsing of auditory streams in event lists, however, an actual modeling of these perceptual processes (the definition of perceptually relevant events) is still ongoing research.

Regarding pulse induction, we depicted two procedures (pulse selection and periodicity function computation) and gave examples of different implementations. Computing a periodicity function is usually more powerful than just selecting a pulse. However, there is a trade-off between how much data can be used for induction and how relevant is the tempo-stability assumption. Using few data (typical of pulse selection methods) lowers the assumption but usually generates unreliable predictors, whereas using much data (typical of periodicity computation methods) generates more reliable predictors solely when the tempo-stability assumption is relevant. An important aspect to account for in the design of a pulse induction algorithm is the amount of tolerance regarding short-time timing deviations.

Some pulse induction methods encode (implicitly or explicitly) aspects of the metric hierarchy by letting large time-scale phenomena influence responses at smaller time-scales (and inversely), e.g. comb filters, NACF, etc. In fact, this encodes the assumption that the perception of high metrical levels, e.g. the measures, should orient the perception of lower metrical levels (and inversely). However, one might precisely want to question this assumption. This is what Parncutt does when writing "each pair of events in a rhythmic sequence initially contributes to the salience of a *single* pulse sensation [...]" (emphasis ours), and later that "pulse sensation can enhance the salience of other, consonant pulse sensation" [110, p.434]. One may understand the "initially" above as an indication not to implement influential schemes between metrical levels in the induction process, but indeed to do it in the tracking process (note that this is

in agreement with the Dynamic Attending Theory [80, 47], see also 2.1.6).

A number of diverse formalisms have been used to implement pulse tracking models. An important aspect is the balance between inertia and reactiveness of the model. Models with a sufficient degree of inertia can be built by accounting for several concurrent hypotheses. This seems a must for preventing "garden-path errors" [114, p.11] and keeping the possibility to recover after an error. Recent models embodying this feature and involving a Bayesian framework [20] or agent-based architectures [40] seem to reach high levels of accuracy. Another important aspect lies in the consideration of incoming data on a "event-after-event" basis or a "predicted beat-after-predicted beat" basis. Following the former rationale is in fact making a first step towards quantizing the data, not solely tracking a pulse.

Elegant AI formalisms have been recently proposed to address jointly quantization and pulse tracking.

Very few algorithms for time signature determination exist. They usually entail the computation and parsing of a periodicity function, as in pulse induction.

The swing estimation is the object of few computational models. The usual rationale for swing estimation is to consider that the tempo is constant (i.e. no long-time timing deviations) and to seek patterns of short-time timing deviations within a pulse induction process, usually entailing the computation of a periodicity function.

2.3 Applications

Applications of automatic rhythm description are manifold. As already argued in 1.1.1 on page 7, in the context of "rhythmic content processing", diverse meaning can be given to the term "processing." Automatic rhythm description may be useful in (automatic, or interactive) composition, editing and transformation of musical data, performance analysis, musical interactions with a computer, defining new ways of listening to music (computer-guided exploration, browsing and design of personal musical databases), new ways of doing commercial deals with music (by helping potential customers to find their way in huge musical databases, facilitating comparisons and retrieval), etc.

2.3.1 Interesting functionalities

Let us provide here a (non-exhaustive) list of potential functionalities of interest for which rhythm description is a must:

- Synchronize two musical streams.
- "Smooth" sequencing of musical excerpts (in the sense of rhythmically coherent), this functionality is certainly a must for DJs.

- Determine time indexes that would stand as "looping" points, or references for "cut and paste" operations. This is a must in "loops-based" music production.
- Genre classification and retrieval: Classify rhythmic patterns w.r.t. categories (either templates as e.g. "salsa" or user-defined). Navigate by rhythmic patterns categories within a database of musical excerpts, compare and retrieve satisfying instances.
- Query-by-rhythm.
- Time-stretch a musical excerpt to match the tempo of another musical source.
- Apply a "human touch", that is, slight deviations from the perfect metrical structure.
- Apply tempo-synchronous audio effects.
- Determining the rhythm that fits best a given melody.
- Provide to a user the possibility to "define rhythmic similarity" by populating interactively a musical database. Let us consider a specific database, music titles being presented to the user with a given organisation, making use of predefined rhythmic similarities. If the user wants to add a new title, declaring it similar to –say– songs A and B, but dissimilar to song C, then, the application must learn somehow what is in that case the implicit meaning of 'rhythmically similar'.
- Compilation of musical playlists: build sequences of music titles satisfying particular properties (or constraints). For instance, one could wish a playlist with the following preferences: 'no slow or very slow tempos', 'medium to heavy beats', 'groove similar to that of "Gimme some more" by Busta Rhymes', etc. The song sequence would then be selected by satisfying constraints on rhythm descriptors –among others.
- Recover from channel transmission errors in streamed audio by beat-pattern based methods.
- Identification of musical excerpts.
- Rhythmic expressiveness transformations, i.e. modifying note timings of MIDI and audio signals. Quantization permits to adjust note rhythmic placements so that they would match precisely the underlying metrical structure, a given score, or a template (e.g. "specific-performer-like quantization"). Rhythmic expressiveness can also be modified so as to change the "groove."

2.3.2 Existing applications

Some literature exist regarding the aforementioned functionalities. [28], [142] and [74] advocate the synchronization and sequencing functionalities. [141] report on the need to provide better tools for looping and cut-and-paste operations. [22] provide an algorithm for determining the rhythm that would fit best a given melody. The compilation of musical playlists is discussed in [107] and [4]. The CUIDADO Music Browser prototype [137] implements this functionality, based, among other descriptors, on the tempo. Automatic recognition of audio excerpts calls for the extraction of some compact description of the data, a "fingerprint" [16], rhythm description may take a part in this process. [81] and [74] discuss the possibility to recognize audio excerpts by means of rhythm descriptors. Concealment of transmission errors in streamed audio by beat-pattern based methods in discussed in [139]. Query-by-rhythm is discussed in [21]. Among others, [134], [52] and [109] propose experimental prototypes for genre classification and audio retrieval based, among other descriptors, on rhythm descriptors.

However, very few of these functionalities are implemented in commercial softwares.

Rhythmic expressiveness transformation functionalities exist in many applications. For instance, any sequencer provides the means to adjust MIDI note timings. Most music producers frequently combine this functionality with the use of sample libraries. The typical problem they are confronted with appears when they want to combine samples that contain rhythmic patterns, e.g. a drum pattern with a percussion pattern. Mixing both will sound sloppy if they have been recorded with slightly different swings. MIDI quantization is used to control the playback of the constituting sounds. One can just modify the MIDI note timings to fit to a certain rhythmic template. Wanting to combine two different MIDI scores, one can do groove matching between both, applying the same rhythmic template to both scores. On the other hand, when dealing with general polyphonic audio one does not possess the MIDI score nor the constituting isolated instrument samples, this technique cannot be applied. Some commercial applications implement other techniques, they restrain rhythmic expressiveness transformations to *swing transformations*. Let us provide here a review of these techniques.[15]

2.3.2.1 Swing transformations

Two techniques are commonly used: "audio slicing" (MIDI score mapping and sample sequencing) and "time-compression / -expansion."

Audio Slicing This technique is based on slicing an audio recording into regions whose playback is controlled by a corresponding MIDI score (that can be edited with a sequencer):

[15] Review done with Lars Fabig and Jordi Bonada [68].

1. Segmenting: Onset detection is performed on the whole audio file to determine meaningful slice boundaries. These segments are identified by timing markers (see Figure 2.6 on page 51).

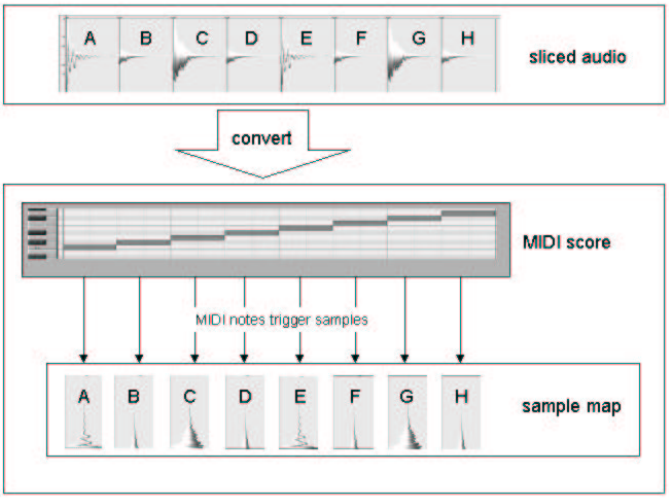

Figure 2.6: MIDI notes triggering polyphonic audio slices in a sample map

2. Slicing: When the user agrees with the proposed segmentation, the audio is then chopped up into slices using the marker information.
3. Instrument sample map generation: All slices are converted into a sampler compatible format, mapping the slices to MIDI notes.
4. Sequencer triggering: Corresponding to time position of each slice within the original audio a MIDI score is generated that contains the time information when each slice has to be played back and triggers the sample map.

The swing of the original audio can then controlled by modifying the MIDI score timings (quantizing it to a different rhythmic template). Whenever a slice is moved away from its original position, problems may arise: it may overlap with others, or at certain positions where there were sound, just silence will remain. Silence will cause a very unpleasant perception, especially for reverberant sounds. To avoid this critical issue, a "tail", derived from the original signal (e.g. using a reverberation technique), is usually added to each slice. That way, the decay phase of the original slice is lengthened. A representative implementation of this method is Propellerhead´s Recycle. The user can control each stage of the algorithm. The software is aimed to work for any audio file. A similar method is used by Spectrasonics. Their system "Groove Control"

permits to change the rhythmic expressiveness of audio files shipped into loop libraries.

Time-compression / -expansion Here, swing transformations are done as follows:

1. Onset detection is performed, as in the "Audio Slicing" method.
2. The onsets are used to segment the signal according to a regular grid: eighth-note positions. The quarter-note positions are also estimated.
3. Portions of audio corresponding to eighth-notes can be either shortened or lengthened by a time-compression/- expansion algorithm.

A critical issue with this method is the determination of the eighth-note and quarter-note positions. The quality and rapidity of the time-compression/-expansion algorithm is also of first importance. This technique is implemented for example in Emagic´s Logic Audio's "Groove machine."

Chapter 3

Initial Objectives

3.1 Regarding rhythm representations

3.1.1 Comments on previous research

Section 2.1 introduced the important aspects of the representation of musical rhythm: event occurrence instants, underlying temporal structure, perceived regularities, violations of the structure (e.g. distortions metrical points/onset times) and emotive aspects. 2.1.1 listed a set of requirements for a rhythm representation in a content-based processing framework. A conclusion to the review of rhythm representations (Section 2.1) is that there is no such thing as *one* rhythm representation that would address all these aspects and satisfy all these requirements. It is in fact difficult to satisfy the whole set of requirements. For instance, a representation could hardly be cognitively relevant to any listener, it could hardly be e.g. both compact and precise, or both correct and compact, or precise and be rapidly computable from audio data, regardless of its distortion level.

Most researchers concord on the importance to represent somehow the *metric structure* and the *timing features* (see 2.1.7). However, their *explicit* representations (by means of specific descriptors and/or description schemes) is still ongoing research.

3.1.2 Research proposal

Our rationale regarding rhythm representation is to focus on the following descriptors, based on an objective and accurate description of musical events. We account for structural aspects, following the GTTM (presented on page 19). Some timing aspects are also present. The emotive aspects are not addressed.

- The onset times
- A representation of periodicities (IOI histogram)

- Several pulses (beats and mean tempo of each):
  - The tick
  - The tactus
  - The downbeat
- Note quantized instants and quantized durations
- The deviations between onsets and metrical points
- The evolution of the performance speed
- The swing

Different applications deal with different rhythmic elements (see Section 2.3). Therefore, it seems relevant to provide *context-oriented rhythm representations*. That is, to use specific descriptor subsets depending on the application. Hence, we propose to organise rhythmic descriptors into different description schemes. Musical excerpts can be segmented w.r.t. different temporal scopes. To each scope of description corresponds a specific description scheme (with different rhythmic descriptors). See Section 4.6.

This rationale opens the way to specific, accurate, descriptions of musical excerpts. It also permits to focus on some specific requirements among those listed on page 17 and leave aside some others (e.g. precision vs. versatility).

3.2 Regarding computational models

3.2.1 Comments on previous research

It is our belief that existing computational models of rhythm description can be explained w.r.t. the general diagram we proposed on page 27. Hence, when developing new models for the automatic description of rhythm, a special care should be put on event-list creation, pulse induction at one or several metrical levels, event-shift handling, pulse tracking, rhythm parsing and time signature determination matters. Our comments regarding these diverse aspects are summarized in 2.2.6. Accordingly, our research proposal below addresses some of these aspects:

3.2.2 Research proposal

3.2.2.1 Rhythmic events

First, we propose to focus on onsets, durations and intensity (energy) for the design of a global representation of periodicities. See Section 4.1.

Second, focusing on frame features, we propose to investigate whether sub-band energies provide better rhythmic events for the induction of the tactus

than the energy computed on the whole frequency range (as [116] claims). See Section 4.2.

Third, we study the relevance of different time-spans for the creation of rhythmic events: from frames to tick and tactus beat segments. See Section 4.2 and Section 4.3.

Fourth, we aim to test the hypothesis that the very definition of rhythmic events (the discrimination between relevant and non-relevant features) should be relative to the excerpt at hand. Indeed, we challenge the assumption that specific musical features would be typical of beats (at any metrical level) in *any* type of music. Testing whether a specific feature is relevant to rhythm description should depend on the periodic (or non-periodic) behavior of this feature. See Section 4.3.

3.2.2.2 Pulse induction

We propose to focus on three different metrical levels: the tick, the tactus and the downbeat. Our models implement diverse periodicity functions: an IOI histogram, ACFs and a pulse track matching function. In pulse induction, models often have a bias towards small periods, either implicitly in the algorithm or explicitly by systematically preferring small periods when having to make a choice (see e.g. [114], [40], [20]). We propose to follow this rationale in the implementation of the algorithm for tick induction, but not for the induction of the tactus or the downbeat. As is suggested by [117, p.91], we account for the possibility to let large time-scale phenomena influence responses at smaller time-scales in the algorithms for tick induction (by using the TWM), tactus induction (by seeking periodicities in the ACF) and downbeat induction (by including high-order terms in the definition of the "decisional features"). Finally, the way we handle of short-time-timing deviations is not novel; however, we believe that it is among those that performs best.

An originality in our approach stands in the induction of several pulses in a bottom-up manner, low-level pulses helping the induction of higher level pulses.

These diverse research proposals are distributed over Section 4.1, Section 4.2 and Section 4.3.

3.2.2.3 Time signature determination

In our approach, the time signature determination is a by-product of the pulse induction process in which we address several metrical levels (see Section 4.3). We simplify the problem in assuming that the time signature denominator is the tactus, we also restrict the set of possible time signatures: duple (groupings of two tactus beats) or triple (groupings of three tactus beats). Eventually, we consider solely two possibilities: $\frac{2}{4}$ and $\frac{3}{4}$.

The reason for that is that we need a way to automatically and objectively assess our results. When dealing with written music, or MIDI, a reference can be taken as the score time signature. But there is no ground truth regarding the concept of time signature of audio signals. As an illustration, it is our belief

that there could be endless discussions on whether the beats of a given excerpt would be better grouped by 2, 4 or 8. But there would certainly be no doubt that for this particular excerpt, 2 would be a better grouping factor than 3. Furthermore, a rule of [90]'s GTTM states that at each metrical level, strong beats (and thus beats at the next level) are placed either two or three beats apart.

3.2.2.4 Swing estimation

Our approach to swing estimation is conceptually similar to that of the few models already existing. The estimation will be considered as a by-product of a pulse induction process, assuming constant tempo and cautiously handling short-time timing deviations in the computation of a periodicity function (i.e. controlling with cautious the size of the smoothing window). See Section 4.4.

3.3 Regarding applications

3.3.1 Comments on existing applications

Regarding swing transformations of audio signals, in our opinion, among the mentioned commercial systems, Spectrasonics's "Groove Control" provides the best sound quality, and it permits to reach important swing factors. But it is also the less flexible: solely proprietary sample libraries can be modified. Indeed, the phases of segmentation, slicing, instrument sample map generation, sequencer triggering and sample tail generation are achieved prior to the sound library shipping (probably in semi-automatic manners). One buys both the audio loops and the metadata attached to them (region boundaries, sample map, etc.). The remaining commercial systems, that work with any audio recording, show in general a poor sound quality, above all when used on polyphonic sounds more complex than drum loops, and at important swing ratio modifications. The main reasons are signal distortions caused by the time-scaling algorithm or unnatural sounding tails of audio slices. An additional disadvantage of the aforementioned systems is that the audio signal must start on a quarter-note; otherwise, the wrong eighth-note might be shortened (i.e. the first instead of the second, which is not at all the same). Also, as swing transformation are often performed jointly with a looping of the audio excerpt, another restriction is that the excerpt length must be an integer multiple of the measure length. This requires the use of a sound editor to adjust the audio file boundaries. Audio Slicing systems also require an additional hardware or software sampler. Finally, these systems require quite a lot of manual editing (by the user or the sample library manufacturer) to get reasonable results (e.g. in onset detection, segmenting the signal according to eighth-note positions). In sum, there does not seem to exist any fully automatic software featuring professional sound quality that would permit to transform the swing of unrestricted polyphonic audio signals.

3.3.2 Proposal

3.3.2.1 Content-based transformation example

Our aim is to illustrate the concept of content-based transformation in a system for rhythmic expressiveness transformations. We aim at grounding the transformation of audio signals on a previous description of their rhythmic content. We restrain ourselves to swing transformations.

We propose to develop a fully automatic system, the Swing Transformer, that would not require neither manual editing nor a software or hardware sampler. It is based on the time-compression / -expansion technique and its goal is to provide better sound quality than the standard applications. See Section 4.4.

3.3.2.2 Rhythmic similarity for audio retrieval

There seems to be few applications that address the notion of rhythmic similarity. However, this notion seems of very first importance in e.g. automatic Genre classification, compilation of musical playlists, etc. Our aim in this domain is not to implement a working and reliable application, but rather to explore the concept of "rhythmic space" that would represent in spatially (in 2 or 3 dimensions) a database of audio excerpts. That is, suggest an association between distances in an intuitive visual space and rhythmic distances. That way, exploration, browsing, comparison or transformation would have a visual meaning, as in the numerous interactive graphical displays described in [134]. See Section 4.5.

3.4 Chapter summary

Regarding rhythm representation, our main objective lies in the proposal of context-oriented representations, accounting for structural aspects and expressive timing aspects of rhythm, we do not address emotive aspects. In our research proposal regarding computational models, we address event-list creation, pulse induction, short-time timing deviation handling (particularly swing estimation) and time signature determination. But we do not consider pulse tracking neither quantization matters. Eventually, another objective in our research is that implementations in specific applications will demonstrate both the usefulness of the rhythmic features we address and the feasibility of their automatic extraction from audio. For instance, we aim at illustrating the concept of content-based transformation via the Swing Transformer.

Chapter 4

First contributions

4.1 Tick induction algorithm

Part of the material in this section has been previously published in a conference article written with Perfecto Herrera and Pedro Cano [71].

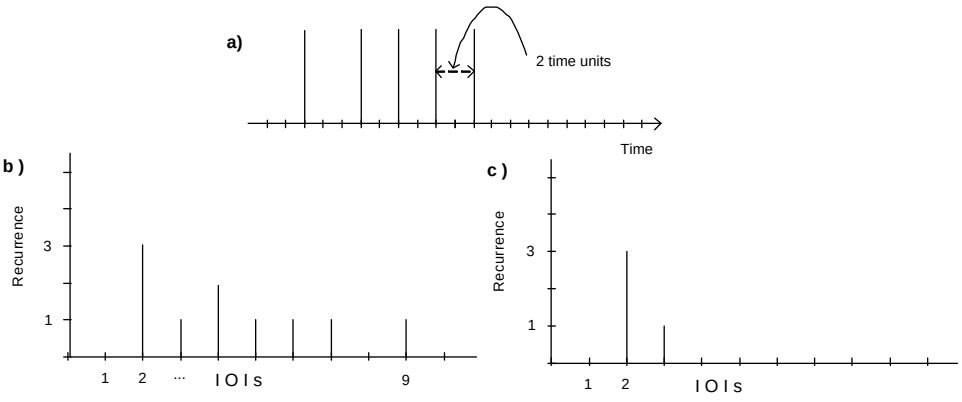

Figure 4.1: Onset sequence (a) — IOI histograms (b & c)

The design of this algorithm had two main aims:

- Provide a working and reliable model
- Test the relevance of onsets, durations and intensity (energy) for the induction of low levels in the metrical hierarchy

Fundamental to the tick induction algorithm is the computation of an audio signal IOIs (subsequent to a detection of onsets). In accordance with the tick definition, keeping the shortest IOI would not suffice to determine the tick. Indeed, in e.g. syncopated musical excerpt, the tick may not be explicit in the

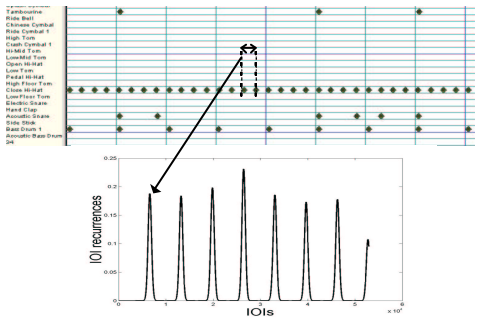

Figure 4.2: "Piano Roll" and IOI smoothed histogram of a MIDI drum track

IOI list, but rather be implied by the relationships between those intervals (see Figure 4.1 on page 58 for an illustration). Such cases are better handled when defining an IOI as the time difference between any two onsets (not necessarily successive) than between successive onsets. Here also, see Figure 4.1 on page 58 for an illustration: in b) IOIs are computed taking into account all pairs of onsets, in c) IOIs are computed taking into account solely successive onsets, the tick –of 1 time unit– is not explicit.

Therefore, the algorithm is based on an IOI recurrence measure. As there are integer timing ratios between metrical pulses, histograms of IOIs should show peaks at approximately harmonic positions. If one extracts note-on timing data from quantized MIDI drum tracks, then the fact that the smallest pulse do contribute to the raising of peaks in the histogram at the exact positions of all of its multiples can be clarified visually on Figure 4.1 on page 58 and Figure 4.2 on page 59.

Therefore, herein the tick is defined as the gap of the IOI histogram harmonic series –one could make an analogy with the notion of fundamental frequency.

4.1.1 Description

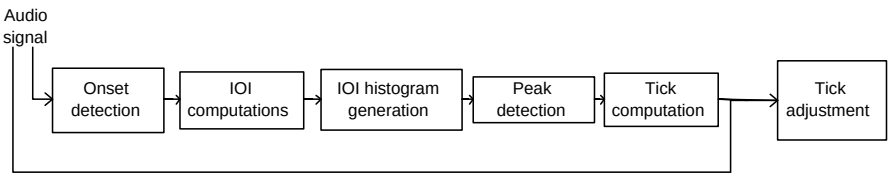

Figure 4.3: Tick induction algorithm flow diagram

The tick induction algorithm is divided in the following steps.

1. Onset detection
   The short-time energy is computed over non-overlapping frames (e.g. 11

ms) of signal. When the energy value is a certain percentage (e.g. 200%) higher than the energy average of a fixed number of previous frames (e.g. 8), an onset is detected. It is assumed that there are at least 60 ms in between two onsets. To each onset is associated a weight (i.e. a degree of confidence), corresponding to the number of after-onset successive frames whose energy is higher than the aforementioned averaged energy. The weight gives an indication whether the onset should be considered as an actual one or an artifact of the onset detection scheme, which can be useful for subsequent uses of the onset list. Optionally, a minimum number of onsets per second (e.g. 2.5) can be imposed to the algorithm. To reach this requisite, the aforementioned percentage is lowered step by step (step set to e.g. 10%).

2. IOI computations
   As mentioned earlier, we take into account the time differences between any two onsets. A weight is associated to each IOI, corresponding to the smallest weight among the two onsets used for the IOI computation.

3. IOI histogram generation
   In order to handle short-time timing deviations, the histogram is smoothed by convolution with a Gaussian function whose standard deviation was empirically adjusted.

4. IOI histogram peaks detection
   Peaks positions and heights are detected in the histogram with a 5-point running window method. A local maximum is detected at index i when the corresponding value is higher than four others: at indexes $i-2k$, $i-k$, $i+k$ and $i+2k$ (where k is set to e.g. 4).

5. Tick period computation
   The fundamental aspect of the tick computation is the use of a particular pulse track matching functions (see page 36): the two-way mismatch error function (TWM [98]). According to the previous definition of the tick, we seek the IOI that best predicts the harmonicity of the IOI histogram. The basic procedure is to generate many possible tick period candidates (all with phase set to zero) and to measure how well (or bad) each one matches the IOI histogram peaks. Candidate periods range e.g. from 80 ms to 700 ms (750 BPM to 85 BPM). For each candidate, a corresponding harmonic pulse track is generated and two error functions are computed. The first one illustrates how well the pulse track elements explain the peaks (resembling the "negative evidence" mentionned on page 36): a global deviation is computed as depicted on the following page. The second one illustrates how well the peaks explain the pulse track elements (resembling the "positive evidence" mentionned on page 36); it is computed as the number of unmatched elements of the pulse track. The TWM error function is a linear combination of these two functions (e.g. with equal weighting factors). The tick period is set to the gap of the pulse track corresponding to the TWM error function global minimum.

Algorithm 1 TWM first error function

```
deviation = 0
for each candidate pulse track period
  for each candidate pulse track phase
    for each peak
      deviation += distance to the closest pulse track element
```

6. Tick period adjustment and phase computation
   The tick period is refined and its phase computed by achieving a matching between several pulse tracks and the signal onsets. As in the previous step, the TWM error function is computed. Differences are that the matching is done on the onsets, not the histogram peaks, that the period candidates range e.g. from (first tick period estimation - 11 ms) to (first tick period estimation + 11 ms) and that we do seek the best phase.
   The reason for achieving the tick period adjustment is that the first tick period estimation is not very accurate because of the histogram smoothing. The smoothing permits to agglomerate IOIs that should be considered jointly, even if not exactly equal; it necessarily entails a tradeoff between precision and amount of IOI agglomeration. Note that the accuracy of the tick period computation is an important issue, even a very small error in this value does propagate in an additive manner in the prediction of future tick beats.

4.1.2 Evaluation

Comparison with pulse track matching functions Pulse track matching functions (see on page 36) are usually based on the computation of a matching score for *each pulse track element* followed by a sum. That would be a "one-way mismatch." A disadvantage in these procedures is that they have a bias towards high-frequency pulse tracks.[1] The TWM counterbalances this bias by computing two error measures, the first illustrating how well the pulse track elements explain the events (this one is similar to aforementioned pulse track matching functions), the second how well the events explain the pulse track elements (this error function has a natural increasing trend toward high-frequency pulse tracks).

Similarities between generic pulse track matching functions and both the TWM function and Cemgil et al.'s tempogram [20] and have been outlined respectively above and on page 37. Let us focus here on a comparison of a tempogram computation and the TWM procedure:

Computing the pulse *period* by a TWM procedure, as detailed above, is conceptually comparable to picking the maximum value of the tempogram marginal

[1] High-frequency pulse tracks sample the time axis in smaller elements, therefore they naturally provide a better match with any sequence than low-frequency ones.

probability function $p(\omega \mid t)$ (i.e. integration over phases, see [20, figure 4]). Differences are the following: first, the TWM seeks the best match between the elements of several candidate pulse tracks and the peaks of the IOI histogram while the tempogram seeks the best match between the elements of several candidate pulse tracks and the very onsets (actually onsets smeared with a Gaussian). Second, the tempogram applies a weighting scheme on the pulse track elements (monitored by the parameter α) while the TWM entails pulse track elements of equal amplitudes. Last but not least, the tempogram suffers a bias toward high-frequency pulse tracks (all the more for large values of α).

Computing the pulse *phase* by a TWM procedure, as detailed above, is conceptually comparable to the computation of "one frequency slice" of the tempogram (i.e. seeking the best phase with a fixed pulse period). One difference being that, in the TWM mismatch, there is no smearing of the onset representation by a Gaussian window, rather, for a given phase candidate τ, the TWM computation of distances between onsets and elements of the pulse track is similar to a tempogram inner product (see [20, equation 8]) with a *triangular* window. Another difference is, here also, that in the TWM procedure, there is no weighting of the pulse track (tempogram's parameter α). This seems an advantage as the TWM can consider less phase candidates than the tempogram, as many as fit in one period is sufficient.

Quantitative evaluations

Evaluation 1 (artificial drum tracks) We developed an algorithm to generate random audio drum tracks,[2] together with exact scores of these drum tracks. The resulting audio files do not really have a musical meaning, however they constitute a useful audio material for automatic evaluation. Roughly, the algorithm first defines a primary pulse at each integer multiple position of which a percussive instrument is added in an empty audio signal. Instruments are randomly chosen from a percussive sound database. Then, a tick is defined as an integer divisor of the previous pulse, at each position in the tick track is randomly assigned either an instrument or silence. To account for more realistic features, deviations of 1 to 10 ms from the exact tick track positions are allowed and white noise is subsequently added to the signal. The evaluation process is the following:

1. Drum tracks and corresponding scores are generated
2. Tick beats are computed from the drum tracks
3. The tick induction is assigned one of the following categories:
   (a) 'Good' if there is a exact match between estimated beats and score beats, with a possible deviation of $\pm 1\%$.

[2] Audio signals of restricted polyphonic complexity, containing few sets of timbres, e.g. acoustic bass drums, snare drums, hi-hats, toms and cymbals.

(b) 'Halves' if each score beat is matched but there are twice as much estimated beats as score beats.

(c) 'Multiples' if each estimated beats match a score beat but there are more score beats than estimated beats.

(d) 'Bad' in the remaining cases.

1000 five-seconds drum tracks have been generated, with a primary pulse of 500 ms (120 BPM), four tick sizes being considered (250, 166, 124 and 83 ms). The results are: 77.3% of the computed ticks have been assigned to the category 'good', 0.7% have been assigned to the category 'halves', 10.4% have been assigned to the category 'multiples' and 11.6% of the tracks have been assigned to the category 'bad'.

Evaluation 2 (real drum tracks) An analysis achieved over real audio drum tracks (that is, not generated automatically) has also been performed. The systematic evaluation is here more difficult as we do not have scores of the drum tracks that would provide a "ground truth." Here, the subjectivity of the listener enters in the evaluation process, all the more if the number of excerpts to evaluate is high. Nonetheless, it is interesting to mention the following results: over 57 drum tracks, ranging from 2 to 10 seconds, made up of different bass drums, snares, hi-hats, cymbals, toms, corresponding mainly to reggae, funk, hip-hop and rock styles, comparing subjectively extracted minimal pulses with estimated tick beats, the determination of the tick was considered good in 86% of the cases, almost good (multiples or rationally-related values) in 7%, and bad in 7%.

4.2 Tactus induction algorithms

Part of the material in this section has been previously published in a conference article written with Perfecto Herrera [69].

A first algorithm for tactus induction was implemented as a by-product of the tick induction algorithm described in Section 4.1. In this section, it is further referred to as 'algorithm 1'. It simply assumes that the maximum of the periodicity function (the IOI histogram), i.e. the most frequent IOI, is the tactus, hence following a rationale found in previous literature (see e.g. [119, p.90]'s "important pulse", [114, p.41], [53]).

The design of a second algorithm (that described below and referred to as 'algorithm 2') had 4 main aims:

- More reliability than the first algorithm
- Study the relevance of defining rhythmic events at the tick period time span

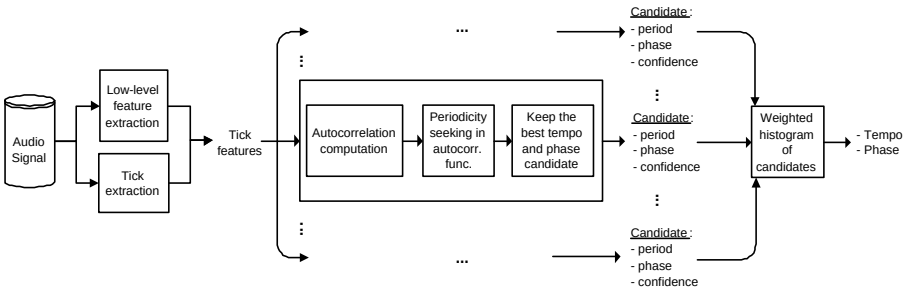

Figure 4.4: Tactus induction algorithm flow diagram

- Investigate whether subband energies provide better rhythmic events for the induction of the tactus than the energy computed on the whole frequency range (as [116] claims)
- Keep generality w.r.t. the feature used, so as to facilitate future investigations regarding the relevance of a specific low-level feature as cue to tactus induction

4.2.1 Description

The tactus period is sought as an integer multiple of the tick period. Precisely, that which best explains the periodicity of the ACFs of a set of low-level features. Upper and lower bounds for the tempo are parameters of the algorithm.

1. Computation of low-level features
   A frame size (e.g. 20 ms) and an overlap factor (50%) are set. Energy (whole band as well as in 26 Bark bands) is computed on each frame.

2. Tick computation
   See Section 4.1.

3. Computation of tick features
   Tick boundaries are matched with frame indexes. A tick contains e.g. between 10 and 20 frames —around 150 ms. Features are computed for every tick as frame feature statistics: mean and standard deviation. Standardized values are computed.

4. Feature autocorrelations
   Recall that the series at hand take one value per tick. The upper limit for the ACF computation (i.e. upper bound for the lag) is set to 20 ticks. The integration time is set to the number of elements in the series.

5. Periodicity seeking in ACFs
   The upper and lower bounds for tempo correspond to possible tick multipliers. We therefore derive upper and lower limits for "interesting" lags.

That is, those that, multiplied by the tick size, would yield tempo values respecting boundaries. For instance, if the tick size is 166 ms, the lower and upper bounds for tempo respectively 60 and 180 BPM (1s and 333ms), then the maximum "interesting" lag is 6, the lowest is 2. Each lag within these limits is a candidate. The goal is then to find which is the best "seed" for a harmonic grid matching the ACF peaks (peaks of the whole ACF, not solely those within tempo limits). For each possible candidate, a confidence is computed that reflects how well this candidate explains the periodic structure of the ACF. (If the ACF is far from being periodic, then all candidates will have low confidences.)
Since the tactus period is sought as an integer multiple of the tick period, a limited set of possible tactus phase corresponds to each lag candidate (i.e. "Which tick corresponds to the first beat?"; for instance, if k is a candidate, then there are just k possible time indexes for the first beat). For each candidate, we choose the phase whose corresponding comb grid best matches high amplitudes in the series (not in the ACF). Note that seeking periodicities in ACF resembles the computation of a NACF (see on page 34).

6. Candidate selection
   For each ACF (each feature), the outputs of the previous step are {tactus period, phase} pairs and confidence factors. For each feature, we select the {tactus period, phase} whose confidence is the highest.

7. Weighted histograms
   The integration of the results is achieved by building a weighted histogram out of the tactus periods, each one weighted by its confidence. Picking the maximum yields the final tactus period. Another weighted histogram is built out of the phase values (first discarding those that do not correspond to the tactus period selected above). The maximum is chosen as the final tactus phase.

4.2.2 Evaluation

Quantitative evaluation of algorithm 1 We performed a test over 80 drum tracks whose tactus beats had previously been manually annotated. A refining of the tactus period and the determination of its best phase were computed similarly as for the tick (see on page 61). The results are the following:

- In 63.7% of the cases, the most frequent IOI corresponds to the tactus
- In 25% of the cases, the most frequent IOI corresponds to integer multiples or divisors of the tactus
- In 11.3%, it is not related with the tactus

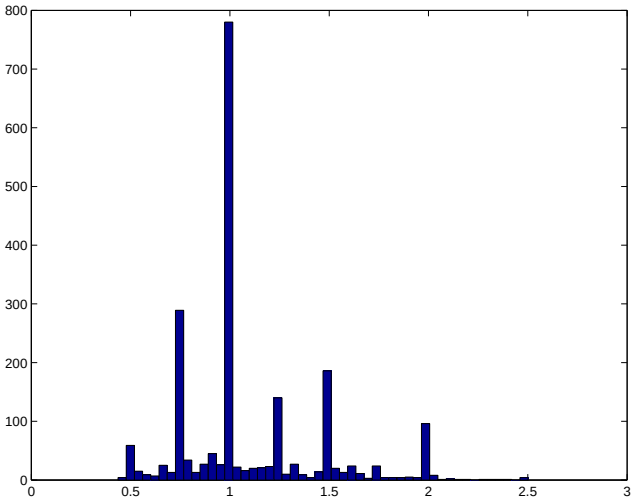

Figure 4.5: Tactus induction algorithm 1 performances

Another evaluation was done on an annotated commercial drumtrack database of 2071 instances.[3] The results are illustrated in Figure 4.5 on page 66: it is a histogram of matches between annotated tempo and tactus period estimation (X-axis is the ratio annotated/estimated, a good match thus corresponds to a value of 1, estimating half the tempo yielding a value of 2, estimating 2/3 of the tempo yielding a value of 1.5, etc.).

One can see in both experiments that even if the success rate is not very high, errors are not random, they are mainly errors of simple integer ratio. We believe that these results illustrate:

- The issue of defining the ground-truth for tempo induction algorithm (see 2.2.6 on page 46)
- As it is commonly thought, a model of tactus induction should take into account IOI data, but also other features of the signals (e.g. dynamic, melodic, harmonic or timbral).

Quantitative evaluation of algorithm 2 Table 4.1 on page 67 details a quantitative evaluation of our second algorithm. The database for evaluation contains 144 audio excerpts (7 to 20 s-long, *.wav* format, $Fs = 44.1kHz$, 16 bits) of polyphonic music, without restriction of styles nor timbres. These excerpts were not used during the design of the algorithm. Limits for the tempo have been set to 60 BPM and 180 BPM. A first evaluation of the segmentation in tick segments revealed 86% of correct segmentation (i.e. 124 excerpts). The first row gives performances using the standard deviation of the energy in 26

[3]Thanks to Pedro Cano for the data.

	Correct tactus period (correct tempo)	Correct tactus period and phase
"Correct tick" subset	91.9%	75.8%
Whole set	79.1%	65.2%
"Correct tick" subset	70.9%	60.4%
Whole set	63.2%	52.7%

Table 4.1: Tactus induction algorithm 2 performances

Bark bands (i.e. 26 features). The second row gives performances using the standard deviation of the energy in the whole frequency range (i.e. 1 feature).

These evaluations are done on a quite small database. Still, they permit to draw useful conclusions for further development of the system. A first conclusion is that (at least with this implementation) periodicities should be sought in frequency subbands and then combined rather than on the whole frequency range. This is in accordance with the point made by Scheirer in [116].

Qualitative evaluation of algorithm 2 This computational model shows theoretical resemblances with that of Scheirer [116], but also many differences in the implementation. Let us discuss some differences on a theoretical ground. Seeking periodicities in the autocorrelation functions is similar to using comb filters. Scheirer [117, p.91] argues that an advantage of comb filters over autocorrelation is that they "encode implicitly aspects of the rhythmic hierarchy, where autocorrelation does not." For instance, a non-null response of a ν Hz-periodic comb filter indicates that the stimulus at hand may show recurrences of τ ms, τ/i ms, *and/or possibly* $i \times \tau$ ms (with $\nu = 1000/\tau$ and i integer). On the other hand, a pronounced peak at τ ms in the autocorrelation function solely reveals that the stimulus may show recurrences of τ ms and/or τ/i ms, *not* $i \times \tau$ ms (a way to overcome this is to seek periodicities in the autocorrelation function, or compute a "narrowed" ACF [13]). Letting large time-scale phenomena influence responses at smaller time-scales is indeed encoding an aspect of rhythmic hierarchy. In fact, this encodes the assumption that the perception of the measures should orient the perception of the tactus. However, one might precisely want to test this assumption. Such tests can be done using autocorrelation (one might as well not seek periodicities in it, just seek the highest peak, and compare results); they cannot be done with the comb filter approach.

Our algorithm relies on the assumption that small time-scale phenomena (fast pulses as the tick) influence larger ones (the tactus). This does not seem to be in accordance with the idea that, when listening to music, one would focus neither on the fastest occurring events, nor on the slow metrical levels, but rather spontaneously on events occurring at an intermediate rate: a "referent" level, the tactus (as formalized in the Dynamic attending theory [80, 47]). Then, attention could be redirected towards other levels (i.e. faster or slower).

The system is open enough to let —or not— slow metrical levels have an

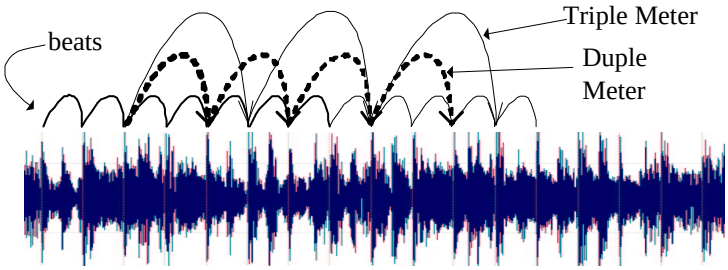

Figure 4.6: Illustration of the two possibilities for beat grouping: duple or triple?

influence on the tactus induction process. It also permits to evaluate the relevance of any low-level feature as a cue for tactus induction. Further evaluation and comparison with other models calls for the setting up of a large annotated database.

4.3 Downbeat and time signature determination algorithm

Part of the material in this section has been previously published in a conference article written with Perfecto Herrera [70].

The design of this algorithm had three main aims:

- Provide a working and reliable model
- Study the relevance of defining rhythmic events at the tactus period time span
- Test the assumption that feature relevance to rhythm description should depend on a measure of feature periodicity in specific excerpts.
  To do so, we experimented some pattern recognition and feature selection techniques, with a qualitative approach rather than a quantitative one. That is, we did not intend to specify what *values* a specific set of features should take to indicate the presence of a downbeat (i.e. a downbeat *model*), we rather intended to determine *which* are the features whose periodicities most likely match downbeat periodicities in a labelled database.

As can be seen in Figure 4.6 on page 68, we restrict the set of possible time signatures to duple (groupings of two tactus beats) or triple (groupings of three tactus beats).

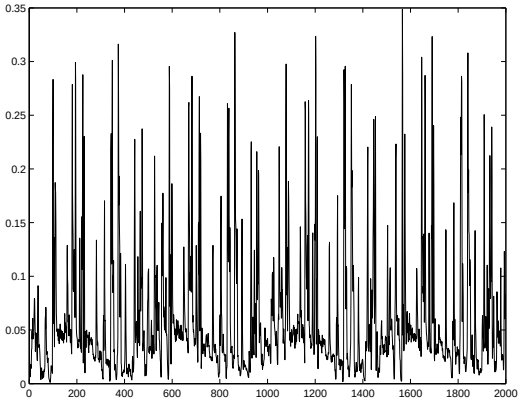

Figure 4.7: Evolution of the energy over the frames of 20 seconds of "A lo Cubano" (*Orishas*, Cuban Hip-Hop)

4.3.1 Description

1. Frame feature computation[4]
   We set a frame size of 20 ms, and a hop size of 10 ms. On each signal frame, a few low-level features of interest are computed. Our investigations led us to use:

   (a) f_1: Energy

   (b) f_2: Spectral flatness, i.e. ratio geometric mean/arithmetic mean (for this feature, frames are multiplied by a Hamming window before DFT computation)

   (c) f_3: Energy in upper-half of first Bark band (approximately 50-100 Hz)

2. Tactus induction
   e.g. [116], [40], Section 4.2

3. Region definition
   Tactus beats are matched with frame indexes. For each beat, three regions of interest are defined:

   (a) $R0$: The whole beat segment, recentered around the beat

   (b) $R1$: The 120 ms region surrounding the beat

   (c) $R2$: The rest of the beat segment, i.e. $\left(R0 \cap \overline{R1}\right)$

[4]More details on the choice of these features (and the beat descriptors below) by experiments with diverse feature selection techniques can be found in [70].

4. Beat segment descriptor computation
   Four beat descriptors are defined as the standard deviation of f_1 over $R0$, the average of f_2 and f_3 over $R1$, and the temporal centroid over $R0$ (this descriptor does not entail frame feature computation). Values of the descriptors are normalized (mean is subtracted and they are divided by the standard deviation). Each musical excerpt is then represented by 4 temporal sequences, whose lengths correspond to the number of beats of this excerpt. Each sequence is the evolution of a specific descriptor over the different beat segments.

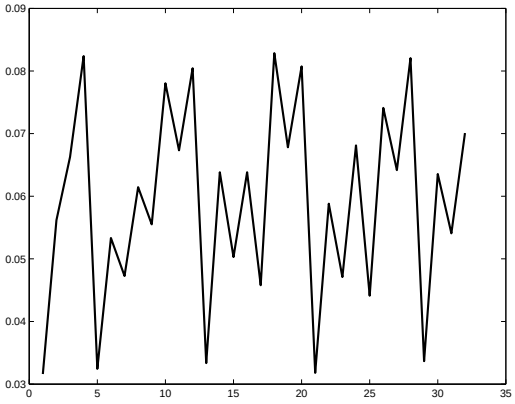

Figure 4.8: Evolution of the energy standard deviation over $R0$s (same song, same temporal scale on the X- axis as Figure 4.7 on page 69, but measured in beat indexes).

5. Periodicity detection
   The (normalized) autocorrelation is computed for each sequence as follows. Let x be the subsequence corresponding to beat indexes 0 to I, and y the subsequence corresponding to beat indexes l to $(l+I)$.

$$acf\,(l) = \frac{x \cdot y}{\sqrt{\sum_{i=0}^{I} (x_i)^2}\sqrt{\sum_{i=0}^{I} (y_i)^2}} \qquad \forall l \in \{0 \cdots U\}$$

   U is the upper limit for the lag l (e.g. 8 beats), and I the integration time (e.g. 10 beats). High peaks in a descriptor autocorrelation function indicate lags for whose this descriptor reveals recurrences along the sequence.

6. Computation of decisional features
   We specify the criterion M for making decisions regarding the 'duple' or 'triple' nature of excerpts:

$$M = \left(\frac{(acf\,(2) + acf\,(4) + acf\,(8))}{3}\right) - \left(\frac{(acf\,(3) + acf\,(6))}{2}\right)$$

M is a real number; the farther from zero in the positive values , the more it is representative of duple time signatures; the farther from zero in the negative values, the more it represents triple time signatures. There is one value of M for each descriptor. Henceforth, the relevant features for the 'duple'/'triple' decision are the values of M corresponding to each descriptor. In the example illustrated in the figures above, relative to a single feature (the evolution of the energy standard deviation over $R0$s), $M = 0.6313$, the time signature is effectively duple.

7. Classification
   Excerpts are represented by 4 features: the criteria M relative to the 4 beat descriptors. The decision regarding the time signature of a test excerpt shall be taken according to the set of values for these features. Deriving a class membership from a set of descriptor values can be achieved by several pattern recognition techniques. For instance, Discriminant Analysis (DA) derives regions of class memberships in the feature space from a statistical modeling of labelled data –in our case, by the very definition of M, the region boundaries are around zero. This technique gave us fairly good results. Even a simple rule, relative to a single feature, seems to give error rates relatively acceptable. Namely:
   "For a given excerpt, if $M_{temporal\ centroid(R0)} > -0.108046$, then this excerpt has a duple time signature, otherwise its time signature is triple."

4.3.2 Evaluation

Quantitative evaluation A database of 70 sounds (format 44100 Hz, 16 bit, mono) was used for experiments. Each excerpt lasts 20 seconds. Boundaries for beginnings and ends were set randomly. We intended to keep generality w.r.t. genres and timbres: excerpts are all polyphonic, the majority multitimbral (there are few monotimbral excerpts: guitar or piano), and the genres are diverse (Hip-hop, Pop, Opera, Classical, Jazz, Flamenco, Latin, Hard-rock, etc.). There are 34 triple time signatures and 36 duple time signatures. 33 excerpts have drums or percussion (10 triple, 23 duple) and 37 do not (24 triple, 13 duple).

We have tested different approaches to classification ranging from non-parametric models (kernel density estimation) to parametric ones (discriminant analysis), and including rule induction, neural networks, 1-Nearest Neighbor (1-NN), or Support Vector Machines (SVMs). Results are obtained by ten-fold cross-validation (average of the error rates over 10 trials with 10% of the samples for testing and 90% for training, randomly chosen). A discriminant analysis with the four features introduced above yielded a 5.2% error rate. With 6 different classifiers (Naïve Bayes, kernel density, 1-NN, SVM, C4.5, PART)[5] and a single feature, the feature M computed from the temporal centroid values over $R0$s, error rates were found to lie around 10%.

[5] Experiments have been done using the commercial software Systat (http://www.systat.com/) and the open-source software Weka (http://www.cs.waikato.ac.nz/~ml/).

4.4 Swing transformation system

Much of the material in this section has been previously published in a conference article written with Lars Fabig and Jordi Bonada [68].

The Swing Transformer consists in a content description module and a transformation module. The former achieves an offline pre-analysis. It does onset detection and rhythmic analysis (determination of tempi and beat indexes at the quarter-note and eighth-note levels, as well as estimation of the swing ratio, if there is any). For later use in the time-scaling algorithm, transient information is also extracted from the audio. Here, the distinction "onset" vs. "transient" is as follows: as detailed in [71], the rhythmic analysis input must consist in reliable note onsets. On the other hand, as explained in [9], the time-scaling algorithm apply different processings on stable and transient regions, there, the detection of non-stationarities does not have to be restricted to note onsets. In one case (rhythmic analysis) the detection of non-stationarities should be rather oriented towards "no false-alarms", in the other case (time-scaling) it should rather be oriented towards "no missed." The transformation module consists in analysis, time-scaling and synthesis of the audio in real-time. The time-scaling is controlled by a "User Swing Ratio." While playing back the audio file (in a loop), the user can continuously adjust the swing ratio in real-time: one can either increase or decrease the swing.

4.4.1 Description

Focusing on the swing of a musical excerpt requires the determination of two distinct metrical levels, a fast and a slow one. As swing is applied on eighth-notes, it is necessary to recognize which elements in the musical flow are eighth-notes. But this is not sufficient, one must also describe the excerpt at a higher (slower) metrical level. That is, determine the eighth-note "phase": in a group of two eighth-notes, determine which is the first one. Indeed, it is not at all the same to perform a long-short pattern as a short-long pattern. The existing swing ratio (if there is any) must also be estimated.

1. Onset detection
   Similar as on page 59. The algorithm is tuned towards no false-alarms. The transient detection for time-scaling implements a similar rationale in frequency subbands [9] and is rather tuned towards no missed.

2. IOI histogram computation
   IOIs are computed, taking into account the time differences between any two onsets. An IOI histogram is generated as on page 60. As detailed below, the standard deviation of the Gaussian smoothing window is an important parameter. Then, peak positions and heights are detected in the histogram with an N-point running window method. One can verify on Figure 4.9 on page 73 and Figure 4.10 on page 73 the intuitive idea that peaks corresponding to shortened and lengthened eighth-notes are closer

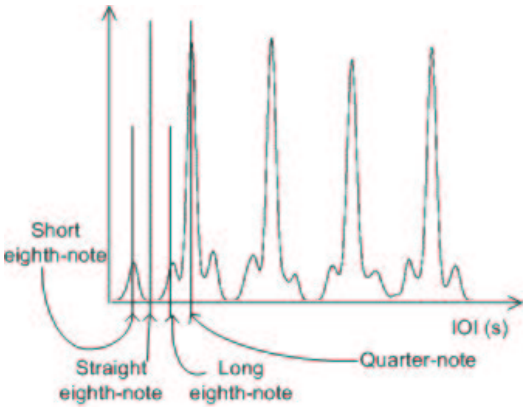

Figure 4.9: Example of an IOI histogram of an audio signal with a 2.7:1 swing ratio

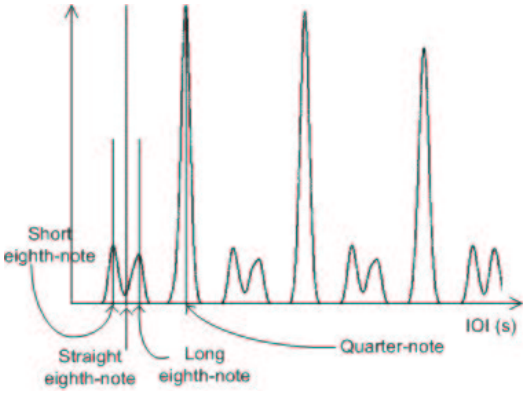

Figure 4.10: Example of an IOI histogram of an audio signal with a 1.5:1 swing ratio

to the straight eighth-note length for small swing ratio than for bigger ones. The estimation of the swing ratio relies on that observation.

3. Tick period estimation
   The tick computation implements the assumption that music shows approximate *integer timing ratios between metrical pulses*. IOI histograms should show peaks at approximately harmonic positions. Therefore, as on 60 tick is computed as the gap of the IOI histogram peak harmonic series. The use of the TWM purposely filters out the deviations from exact integer ratios between pulses that occur e.g. in the case of music that swings. In this stage, the Gaussian standard variation is set to a medium value (e.g. 35 ms) not to be led astray by small IOIs.

4. Eighth-note and quarter-note period estimation
   The quarter-note period is computed posterior to the tick as the maximum peak in the IOI histogram among four candidates: the tick, twice the tick, three times the tick and four times the tick (with additional boundary restrictions: the quarter-note minimum tempo is set to 50 BPM, and the maximum to 270 BPM).
   Then the eighth-note period is computed as half the quarter-note.
   Let us provide a justification of this procedure. The swing concerns the smallest metrical level present in the signal. As can be seen on Figure 4.9 on page 73 and Figure 4.10 on page 73, the amount of swing has a direct influence on the tick estimation. Let us consider the following cases:

   (a) If the audio has a swing ratio of 1:1 (i.e. "no swing"), the computation of the tick yields the actual smallest level. That is, the tick is the eighth-note.

   (b) If the swing ratio is 2:1 ("ternary feel"), the IOI corresponding to the straight eighth-note is not present in the signal (nor in the histogram), there are solely shortened eighth-notes (whose durations are 1/3 of that of a quarter-note) and lengthened eighth-notes (whose durations are 2/3 of that of a quarter-note). There, the tick computation yields 1/3 of the quarter-note length.

   (c) If the swing ratio is higher than 2:1 (above ternary feel), the IOI corresponding to the straight eighth-note is not present in the signal, there are solely shortened eighth-notes with durations smaller than 1/3 of that of a straight quarter-note, and lengthened eighth-notes with durations superior to 2/3 of that of a straight quarter-note. There, as it is restricted to integer ratios, the computation of the tick yields either 1/3 or 1/4 of the quarter-note length.

5. Swing ratio estimation
   Two different implementations of the swing ratio estimation are still under tests. They are both based on the computation of a second IOI histogram, with a Gaussian standard deviation smaller than in the previous step (e.g. 10 ms), in order to account for more peaks and also a better time precision in the peak positions.

   (a) First implementation
   It is based on the computation of deviations between all peaks in the IOI histogram and integer multiples of the eight-note length. An important observation is that the deviation distribution is *bimodal*: one mode is around 0 (it corresponds to deviations w.r.t. quarter-note positions) and the second mode does correspond to relevant deviations for swing estimation (see Figure 4.11 on page 75, in this example, the straight eighth-note length is 161ms, the deviation central tendency of the second mode is 68 ms; this results in a 2.45:1 swing ratio). The central tendency of the second mode deviations is computed (either

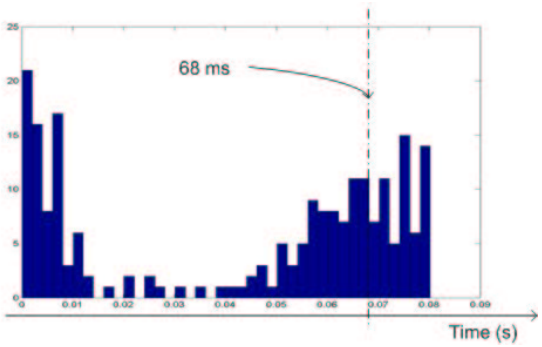

Figure 4.11: Distribution of the deviations {IOI histogram peaks / integer multiples of the eight-note length}

as the mean, the median or the mode). Finally, the swing ratio is computed as:

$$Swing\,ratio = \frac{Eighth\,note\,period + central\,tendency}{Eighth\,note\,period - central\,tendency}$$

(b) Second implementation
The basic concept in this implementation is the seeking of the best match between IOI histogram peaks and swing templates. As can be seen in Figure 4.12 on page 76, templates are built as harmonic comb grids with a gap equal to the quarter-note length and with varying offsets: between the eighth-note length for the first template and e.g. 3/4 of the quarter-note length for the last one (this is directly related to the maximum boundary the user can set for swing ratio seeking). For each template, the TWM error function is computed between the grid elements and the IOI histogram peaks (see on page 60), then the best template is chosen as that which yields the smallest error (i.e. which best matches the peaks). This procedure resembles somehow Laroche's method ([87], see also 2.2.5), the difference being that we do not achieve the matching over the onsets but the IOI histogram peaks.

6. Eighth-note and quarter-note position estimation
Given the quarter-note period and the audio signal onsets, the quarter-note positions are sought (and quarter-note period is slightly adjusted) so as to match to the best the onset positions similarly as on page 61. Logically, each quarter-note position is also an eighth-note position. The remaining eighth-note positions are simply determined as positions in between each pair of quarter-notes: half-way between two quarter-notes, adjusted with respect to the detected swing ratio.

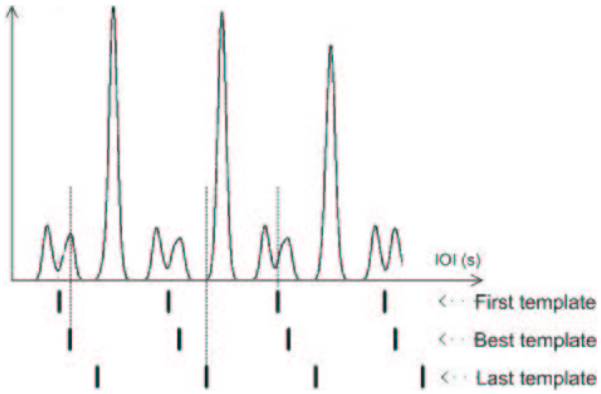

Figure 4.12: Illustration of the template-matching approach to swing estimation

7. Swing transformation by time-stretching
   Modifying the swing means moving onsets corresponding to eighth-notes from their original positions to different ones. In Figure 4.13 on page 77, an example is shown for an audio file that has no swing (signal in the upper half of Figure 4.13 on page 77). Quarter-notes are depicted by a simple number ('1', '2', '3' on the figure top). The eighth-notes are indexed with i ($i = 1$ means "in a subdivision of a quarter-note in two eighth-notes, this is the first eighth-note", and $i = 2$ means "in a subdivision of a quarter- note in two eighth-notes, this is the second eighth-note") and their corresponding sample positions n_i. The detected Swing Ratio (SR) in the example is 1:1 (i.e. "no swing"). When the user chooses a different swing ratio (for example 2.6:1), the regions between indexes $n_{i=1}$ and $n_{i=2}$ are expanded with the time-scale factor TS_{EXP} while the regions between $n_{i=2}$ and $n_{i=1}$ are compressed with TS_{COMP}. The scaling factors for expansion and compression are calculated as follows ($TS > 1$ means signal expansion, and $TS < 1$, signal compression):

$$TS_{EXP} = \frac{SR_{User} + SR_{Detected} \cdot SR_{User}}{SR_{Detected} + SR_{Detected} \cdot SR_{User}}$$

$$TS_{COMP} = 1 + SR_{Detected}\,(1 - TS_{EXP})$$

   When the original audio signal already has swing, the processing is slightly different because the regions of expansion and compression are not equal-sized anymore. The real onset positions of eighth-notes indexed by $i = 2$ deviate from the straight eighth-note grid. Regions to be expanded and to be compressed are adapted consequently.
   More details on the time-stretch algorithm can be found in [9].

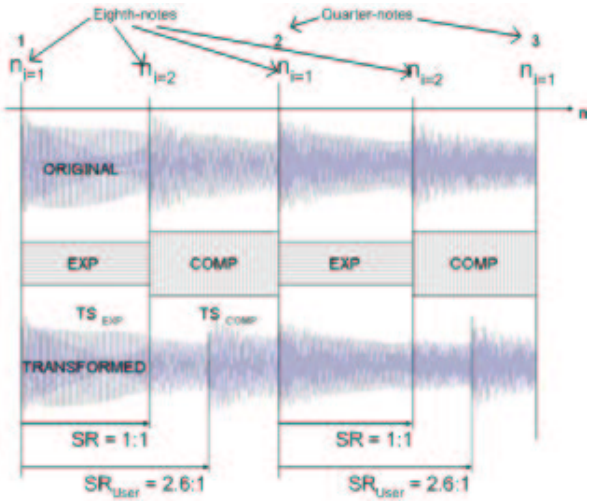

Figure 4.13: Adding swing to an audio file by time-scaling

4.4.2 Evaluation

The system provides very good sound quality for monophonic signals or polyphonic stereo mixes. It could be also extended to handle multi-channel signals.

Improvements are needed on the time-scale algorithm to reduce phasing and flanging for time-scale factors above 1.3 (note that this is an important scaling factor). If we want to apply very drastic swing ratios, scaling factors higher than 1.3 are very often exceeded and sound quality may decrease. Although at large time-scale factors an analysis frame is repeated many times until the next frame is chosen, the resulting signal possibly sounds quite metallic. This can be improved by interpolation of the magnitude spectrum between subsequent analysis frames. Another area for improvement is the handling of transients lying precisely on eighth-note positions. We assume that when the scaling factor is switched from expansion to compression (e.g. from 1.3 to 0.7) in a small group of frames belonging to a transient, this may cause a doubling of the transient (this is problematic especially for drum sounds). Finally, it is our belief that the analysis of the deviation distribution should be further pursued. Indeed, the number of modes, the mode variances and higher moments (skewness and kurtosis) are probably representative of important information regarding diverse systematic timing deviations [7].

4.5 Rhythmic similarity system

We[6] implemented a number of rhythmic distance measures, all based on the computation of an IOI histogram (see on page 60):

[6]Work done with Pedro Cano and Martin Kaltenbrunner

- Euclidean distance
- Cosine distance
- TWM distance

That is, the rhythmic distance between two musical excerpts is computed as the distance (one of the three above) of their respective IOI histograms. These distances inherently define a multidimensional space, the dimensions of which are unknown. In order to reduce the dimensionality and present the data in a space of two or three dimensions, we made use of a Multidimensional Scaling (MDS) algorithm.[7] We used the *Song Surfer* of Cano et al. [17] to test the overall relevance of the resulting data (i.e. to actually listen to the resulting 2-D mapping of the database).

In general, results were quite disappointing; mainly because we fell short on finding a way to objectively assess our results, other than simply browsing the 2-D map and asking people whether they agreed with the algorithm on the distance between songs A and B.

4.6 MPEG7 rhythm representation enhancement

Along the aims detailed in Section 3.1, we propose the following representation for musical rhythm. Associated to specific representation of melody and instrumentation, part of it provided the material of two articles written with Emilia Gómez, Perfecto Herrera and Xavier Amatriain [61, 60].

The concept of context-oriented rhythm representations was also developed in an article written with Benoît Meudic [72].

4.6.1 Description

Musical excerpts can be segmented w.r.t. different temporal scopes. Two segment are defined. The *NoteSegment* represents a note and the *MusicSegment* represents an audio excerpt, either monophonic or polyphonic, spanning a longer time interval. The *MusicSegment* can be decomposed in other *MusicSegments* (for example, a polyphonic segment could be decomposed in a collection of monophonic segments, as illustrated in [61, figure 2]) and in *NoteSegments*. This, by means of two fields (the *MusicSegmentTemporalDecompositionType* and *NoteSegmentTemporalDecompositionType*) whose types derive from the MPEG7 *AudioSegmentTemporalDecompositionType* [31] (see [61, figure 3]).

To each scope of description corresponds a specific description scheme, with different rhythmic descriptors (and also different melodic and instrumental descriptors). A *NoteDS* and a *MusicDS* are associated to, respectively, the *NoteSegment* and the *MusicSegment*. While the *NoteDS* accounts for those rhythmic descriptors specific to a single note, the *MusicDS* accounts for rhythmic descriptors that emerge from the timing relationships between several notes.

[7] See Malcolm Slaney's http://rvl4.ecn.purdue.edu/~malcolm/interval/2000-025/

Some rhythmic elements of the *MusicDS* are embedded in a structure describing the intertwined metrical levels that coexist in music, the *PulseDecomposition* (among other descriptors, it accounts for the tempo). Additionally, the DS accounts for possible variations of the musical pace and the time signature. Also, this DS accounts for simple series of letters permitting to describe a signal in terms of recurrences of events w.r.t. the rhythmic structure that organises musical signals (see [61, figure 5]). The exact temporal location of the music segment is also described by the MPEG7 *MediaTime* attribute derived from the MPEG7 *AudioSegment*.

Pulse decomposition: Several pulses coexist in a musical piece. In this respect, the *MusicSegment* rhythmic description accounts for a decomposition in pulses: each pulse has a name, a beginning time index, a gap value, a rate (which is logically proportional to the inverse of the gap; some might prefer to apprehend a pulse in terms of occurrences per minute, some others in terms of milliseconds per occurrence –as in MIDI) and a time function that defines the variation of the musical pace (the regular grid defined by the previous 'beginning' and 'gap' can be warped according to it in order to derive the beats). Among the hierarchy of pulses, no pulse is by any means as important as the tempo. In addition, the reference pulse for writing down the rhythm sometimes coincides with the perceptual pulse. Therefore, it seemed important to provide a special handling of the tempo. The *PulseDecomposition* type holds a mandatory pulse named *Tempo*, in addition to it, several other pulses can optionally be defined by using the *OtherPulse* container (additional pulses can be e.g., the tick, the downbeat, etc.).

Meter: It is similar to the MPEG7 *MeterType* (see 2.1.5). Related to the time signature, the *MeterType* does not specify the instant of occurrences of the downbeats. In our proposal, this information can be stored in the *OtherPulse* container, by defining a pulse as the downbeat.

Sequence: A simple series of letters can be added to the description of a *MusicSegment*. This permits to describe a signal in terms of recurrences of events, with respect to the rhythmic structure that organises musical signals. For instance, one may wish to categorize the succession of ticks in terms of timbres (e.g. this would look like the string 'abccacccabcd'), and then seek patterns (see [71] for an example). Categorize segments of the audio "chopped up" with respect to the tempo grid might also reveal interesting properties of the signal. One might want to describe a signal in the context of several pulses. Therefore, several sequences can be instantiated.[8] Rather than restricting one's time precision to that of a pulse grid, one might wish to categorize musical signals in terms of accurate time indexes of occurrences of particular instruments (e.g. bass drums and snares

[8]Note that this descriptor might also be used to describe any type of patterns, would they be rhythmic, melodic or harmonic (e.g. characterizing an A-B-A form).

as in [63], [73]). This, in order to post-process these series of occurrences so as to yield rhythmic descriptors. Here, the decomposition of a music segment in its constituent instrument streams (i.e. other music segments) is needed (see [61, figure 5]). Then, each of these music segments can be decomposed into note segments, thus keeping precise timings. For instance, a music segment can be attributed to the occurrences of the snare, another one to those of the bass-drum; timing indexes lie in the MPEG7 *TemporalMask*, inherited from the MPEG7 *AudioSegment*, that permits to describe a single music segment as a collection of sub-regions disconnected and non-overlapping in time.

The first important rhythmic elements of the *NoteDS* is the exact temporal location, described by the MPEG7 *MediaTime* attribute inherited from the MPEG7 *AudioSegment*. Low-level descriptors are also inherited from the *AudioSegment*. In addition are the following attributes:

Quantized instant: Given a reference pulse, that might be e.g. the tick, a note is attached a rational number representing the number of pulses separating it from the previous one.

Deviation: It represents the deviation of a note from its closest pulse. The deviation is expressed in percentage of a reference pulse, from -50 to $+50$. (Here, the reference pulse can be different than that used for quantizing, one might want to quantize at e.g. the beat level and express deviations with respect to e.g. the tick.)

4.6.2 Evaluation

Pros The proposed rhythm representation is not specific to an application nor to a type of signal (e.g. a monophonic melody). It does address metric structure elements, perceptual features and timing features. With the division in two different description schemes attached to different temporal scope, and the possibility for many descriptors to be instantiated or not, this rhythm representation proposal opens the way to the context-oriented rhythm representations advocated in Section 3.1. In the framework of a specific application, depending on the descriptor subset instantiated, specific requirements (see 2.1.1) may be satisfied.

The quantized instant proposal can be seen as a generalization of the MPEG7 *BeatType*. However, it improves it in the following aspects:

- One can choose the level of quantization (the reference pulse does not have to be the time signature denominator as in the *BeatType*). ·
- Even when a reference pulse is set, one can account for (i.e. represent without quantizing) durations that do not rely on this pulse (as in the case of e.g. triplets in a quarter-note-based pattern). This feature is provided by the fact that the quantized instants are rational numbers and not integers.

- The onset quantizations are done towards the closest beat (not towards $-\infty$).

Fine deviations from the structure can be represented, which may be useful for analyzing phrasing and expressivity.

This chapter provides algorithms to automatically extract most of the descriptors from audio signals and details applications that use this representation.

The *PulseDecomposition* structure permits to describe the intertwined metrical levels that coexist in music. We focused on three levels, the tick, the tempo and the downbeat, but the structure is general and other levels can be easily incorporated.

Cons Emotive aspects are not addressed. We believe that adding such higher level descriptors, linked to emotive aspects (e.g. mood), needs solid grounding and testing on the proposed descriptors, defining interdependency rules that currently cannot be easily devised.

In the proposed meter (the MPEG7 *MeterType*), an issue is that there is no direct link between this container and the pulses of the *PulseDecomposition*. The coherence between the meter and the different pulses is left to the user. Further, there is currently no other way to deal with a change of meter than defining a different audio segment.

Chapter 5

Conclusions and future work

5.1 Initial objectives and summary of contributions

We intended to highlight the current relative lack of agreement regarding explicit representational elements of rhythm. On this basis, our main objective regarding rhythm representations was the proposal of context-oriented representations. We argued that they should account for structural, expressive timing, perceptual and emotive aspects of rhythm. We also proposed a list of requirements for rhythm representations in the context of content-based music processing. Our contribution is the design of a representation scheme, built upon the rhythm descriptors that are currently defined in the MPEG7 standard, based on two description schemes, each one "specialized" for a given temporal scope of description. We believe that this rationale opens the way to specific, accurate, descriptions of musical excerpts. Indeed, the same musical item can be described (rhythmically) in different manners by the same representation scheme, just by exploring diverse layers of this scheme. This also permits to focus on specific application requirements.

The main aim in this dissertation was however to provide a comprehensive review of rhythm description computational models. To this purpose, we introduced a comprehensive diagram whose functional blocks permit to explain and compare (conceptually) different models. Our research proposal regarding computational models was to propose algorithms addressing four aspects of this diagram: event-list creation from audio, pulse induction (at three metrical levels), time signature determination and swing estimation. We aimed at demonstrating the feasibility of the automatic extraction from audio of these rhythmic features. Commenting on the inherent difficulty of evaluating rhythm description algorithms, we compared qualitatively our algorithms to state-of-the-art ones, we also evaluated quantitatively our algorithms on personal musical audio databases.

We also aimed at demonstrating the usefulness of these (automatically ex-

tracted) rhythmic features. We illustrated the concept of content-based transformation via a software application for rhythmic expressiveness transformations in musical audio signals. A fully automatic system has been developed, the Swing Transformer, it requires neither manual editing nor software or hardware sampler. It provides very good sound quality for monophonic signals as well as polyphonic stereo mixes, even for relatively high transformation factors.

5.2 Future work

5.2.1 Rhythm representation

Tempo ambiguity experiment[1] It is common to assign to pieces of music a scalar value, its tempo in BPM, supposed to be representative of its speed. However, we have seen that some psychological research asserts that the perception of tempo is an ambiguous phenomenon, ambiguity referring here to the rate of agreement among listeners. There doesn't seem to exist *one objective tempo* to attribute to a piece of music.

We will aim at determining if there exist cases for which tempo values are consensual via an experiment over a number of subjects. Then, the goal will be to investigate if there are (and what they are) signal features (i.e. *objective* features) that strongly influence this phenomenon. In sum, we will aim at testing to which extent can one consider that tempo perception is relative to the signal.

Hopefully, this experiment will result in an algorithm to measure tempo ambiguity directly on the signal.

High-level rhythmic descriptors Our rhythm representation proposal did not address emotive aspects of rhythm as e.g. the mood. We believe that adding such highly abstract rhythm descriptors to a description scheme needs solid grounding and testing on large amounts of human judgement data. Provided such data would be available, data mining techniques could probably be useful in their analyses.

Furthermore, measuring tempo ambiguity at several levels of the metrical hierarchy could be useful to the derivation of other high-level descriptors; maybe some measure of the "rhythmic complexity", if it is grounded on the comparison between metrical level ambiguities.

5.2.2 Computational model

First of all, we did not address all the aspects of automatic rhythm description proposed in Figure 2.4 on page 26. Namely, we did not consider pulse tracking nor quantization matters. This is a must for many applications. To name a few, tools for performance analysis or transformations require, among other things, the extraction of the speed evolution (e.g. tempo curves) and automatic score followers require quantized durations.

[1] We plan on designing this experiment with Diego d'Allosto

Many previous models envisaged rhythm description either as inducing and/or tracking *one* pulse, or quantizing performance data. In our opinion, it would probably be more efficient to focus on several metrical levels simultaneously. In this context, high levels provide "anchor points" to smaller ones and small levels provide basic time segmentations to the determination of higher ones. Additionally, in such a context, quantization can be seen as a by-product of metrical structure determination. We also believe that, in fact, this context would make more sense perceptually, as it has been shown that humans do not perceive a single pulse, but rather part of the metrical hierarchy. We can focus on one pulse and then redirect attention towards other ones if required. Moreover, we have previously commented that when we do focus on a single pulse, there is no evidence that we all do it on the *same* pulse. Therefore, a computational model of rhythm description should embody a capability to "perceive" several pulses.

Our main objective for future work will thus be the design of a computational model that will implement influential schemes between metrical levels. We will therefore mainly focus on the following research topics:

- Feature selection
- Intertwined pulse inductions
- Pulse tracking

5.2.2.1 On feature selection

As detailed in Section 4.3 and [70], we challenge the assumption that specific musical features would be typical of beats (at any metrical level) in *any* type of music. Testing whether a specific feature is relevant to rhythm description should depend on the periodic (or non-periodic) behavior of this feature. Furthermore, defining a set of relevant physical features for rhythm description should rather be grounded on observed relevance of features (on real musical signals) than on a declarative procedure or on intuitions.

Therefore, we will pursue our work on feature selection with a machine learning approach, in the direct following of [70] and [69].

5.2.2.2 Intertwined pulse inductions

An originality in our approach to pulse induction was to induce several pulses in a bottom-up manner, low-level pulses helping the induction of higher level pulses. There is a need of top-down constraints for the diverse pulses to be properly jointly induced.

We will therefore incorporate to the general induction scheme segmentation algorithms for large time spans (e.g. phrases, motives).

In the induction of the smallest pulse (the tick), our algorithm relies on a detection of onsets. This is a potential drawback as it is difficult to ensure reliable onset detection for any type of musical audio signals. To overcome this issue, we will adapt the periodicity function computation, the TWM, to the

matching between pulse tracks and continuous functions e.g. frame features (instead of the matching of two discrete grids).

Additionally, each metrical level will be characterized by a salience factor, directly related to the degree of perceptual ambiguity of this level (i.e. whether listeners would agree on this level as the perceptually more salient, i.e. the tactus), see on page 83.

5.2.2.3 Pulse tracking

So far, we did not address the issue of pulse tracking in our computational models. Future work on the topic will embrace the Bayesian framework advocated by [19].

5.2.2.4 Short-time timing deviations

In addition to refinements and final tuning of our algorithm for swing estimation, we will seek the automatic estimation of other short-time expressiveness features as e.g. drummers' "behind-the-beat" ways of playing, typical in Funk music.

5.2.2.5 Comparisons with other models

Another important objective in future work is to provide systematic, quantitative comparisons between models. This requires the setup of a common database of audio signals. Hopefully, results of the experiment proposed on page 83 will be useful for making a step towards the (still eagerly awaited) definition of a "ground truth" for model comparisons.

5.2.3 Applications

Rhythmic similarity system We detailed in Section 4.5 first experiments regarding the design of a system for browsing musical databases by rhythmic similarities. Results were quite disappointing. Future work will be oriented towards:

- The design of new rhythmic distances based on further parsing of periodicity functions (as is suggested in [43]). For instance, we may make use of diverse statistics regarding the position and heights of local maxima in the function. Another promising research direction is probably to take into account the evolution of periodicity function peaks over time (computing successive periodicity functions on relatively short windows, with overlap).
- Also following [43], define a database of "unambiguous" rhythmic patterns in order to possess a sounder ground truth for evaluation. That is, test our system over audio excerpts corresponding to established rhythmic categories, e.g. typical Waltz patterns, typical Tango patterns, typical Reggae patterns, etc.

A graphical interface for metrical structure editing We also aim at incorporating the whole set of our algorithms into a single application. It would mainly serve as a tool for understanding the metrical structure of musical signals. In a nutshell, it would present a time representation of the audio file and propose markers at specific positions corresponding to beats at diverse metrical levels. These markers could be deleted or adjusted by the user. They could also be used for looping, or cut-and-paste operations.

5.2.4 An "embodied" model of rhythm description

As early as 1987, [33] suggested that "[...] it seems that music understanding relies more heavily on pattern recognition capabilities, and on hierarchical grouping, than it does on logical reasoning and problem-solving techniques. It is possible that an emphasis on the latter is precisely the privilege of musical experts, who are able to name, and to reason about, entities which naive listeners also perceive, for the most part, but are unable to put into words."

Recent efficient computational models of rhythm description implement problem solving techniques and probabilistic approaches. State variables (e.g. beats, quantized durations) are considered hidden variables to be estimated from observations. Efficient models have been implemented (e.g. [19], [112]) to search wide hypothesis spaces, contributing to breakthroughs in recent AI research (e.g. graphical and switching state space models, particle filtering). In this powerful framework, knowledge is not explicitly constrained by fixed, deterministic sets of rules, however, it lies in probabilistic models learned from measured data during a supervised training step.

Different approaches can also be found in recent literature. Rhythm perception could be seen as a reactive process (i.e. where cognition matters would not enter) [132, 117]. Accordingly, bottom-up approaches to the derivation of rhythmic attributes from input data could be preferred to approaches including a modeling of the musical stimuli. An illustration of this rationale can be seen in [55], who provides an interesting use of both supervised learning and reinforcement learning. Bryson [14] espouses the so-called "embodied cognitive science", "behavior-based" or "new AI" rationale. The "Reactive Accompanist" architecture embodies principles developed by Brooks [11, 10] for designing and controlling mobile robots interacting with their environment.

Music is greatly structured and there exist established musical formalisms, hence it seems difficult to think about computational systems that would not have explicit symbolic representations and would rather just "interact" with a musical environment; except maybe precisely for rhythmic aspects of music. Rhythm perception is precisely very much linked to a physical apprehension of music.

Bibliography

[1] P. Aigrain. New applications of content processing of music. *Journal of New Music Research*, 28(4):271–280, 1999.

[2] M. Alghoniemy and A. Tewfik. Rhythm and periodicity detection in polyphonic music. Proc. IEEE Workshop on Multimedia Signal Processing, 1999.

[3] P. Allen and R: Dannenberg. Tracking musical beats in real time. Proc. International Computer Music Conference, 1990.

[4] J.-J. Aucouturier and F. Pachet. Scaling up music playlist generation. Proc. IEEE International Conference on Multimedia Expo, 2002.

[5] R. Baeza-Yates and B. Ribeiro-Neto. *Modern information retrieval*. Addison Wesley, 1999.

[6] L. Baggi. Neurswing: An intelligent workbench for the investigation of swing in jazz. *Computer*, 24(7):60–64, 1991.

[7] J. Bilmes. *Timing is of the Essence: Perceptual and Computational Techniques for Representing, Learning, and Reproducing Expressive Timing in Percussive Rhythm*. Thesis/dissertation, MIT, Cambridge, 1993.

[8] T. Blum, D. Keislar, A. Wheaton, and E. Wold. Method and article of manufacture for content-based analysis, storage, retrieval, and segmentation of audio information. USA Patent 5,918,223, 1999.

[9] J. Bonada. Automatic technique in frequency domain for near- lossless time-scale modification of audio. Proc. of the International Computer Music Conference, 2000.

[10] R. Brooks. Intelligence without representation. *Artificial Intelligence Journal*, 47:139–159, 1991.

[11] R. Brooks. New approaches to robotics. *Science*, 253:1227–1232, 1991.

[12] J. Brown. Determination of the meter of musical scores by autocorrelation. *Journal of the Acoustical Society of America*, 94(4), 1993.

[13] J. Brown and M. Puckette. Calculation of a narrowed autocorrelation function. *Journal of the Acoustical Society of America*, 85(4):1595–1601, 1989.

[14] J. Bryson. *The Subsumption Strategy Development of a Music Modelling System*. Thesis/dissertation, University of Edinburgh, Faculty of Science (Department of Artificial Intelligence), 1992.

[15] E. Cambouropoulos, S. Dixon, W. Goebl, and G. Widmer. Human preferences for tempo smoothness. Proc. International Symposium on Systematic and Comparative Musicology, International Conference on Cognitive Musicology, 2001.

[16] P. Cano, E. Batlle, T. Kalker, and J. Haitsma. A review of audio fingerprinting. *Journal of VLSI Signal Processing Systems*, To appear, 2004.

[17] P. Cano, M. Kaltenbrunner, F. Gouyon, and E. Batlle. On the use of fastmap for audio retrieval and browsing. Proc. International Symposium on Music Information Retrieval, 2002.

[18] A. Cemgil, P. Desain, and B. Kappen. Rhythm quantization for transcription. *Computer Music Journal*, 24(2), 2000.

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

[20] A. Cemgil, B. Kappen, P. Desain, and H. Honing. On tempo tracking: Tempogram representation and kalman filtering. *Journal of New Music Research*, 28(4):259–273, 2001.

[21] J. Chen and A. Chen. Query by rhythm: An approach for song retrieval in music databases. Proc. IEEE International Workshop on Research issues in Data Engineering, 1998.

[22] F. Chin and S. Wu. An efficient algorithm for rhythm-finding. *Computer Music Journal*, 16(2), 1992.

[23] J. Chowning, L. Rush, B. Mont-Reynaud, C. Chafe, A. Schloss, and J. Smith. Intelligent system for the analysis of digitized acoustic signals. Report STAN-M-15 CCRMA Stanford University, 1984.

[24] J. Chung. *An agency for the perception of musical beats or If I had a foot...* Thesis/dissertation, MIT, Cambridge, 1989.

[25] E. Clarke. Categorical rhythm perception: An ecological perspective. In *Action and perception in rhythm and music*, pages 19–34. Royal swedish academy of music, 1987.

[26] E. Clarke. Levels of structure in the organization of musical time. *Contemporary music review*, 2(1):211–238, 1987.

[27] E. Clarke. Rhythm and timing in music. In D. Deutsch, editor, *The Psychology of Music, 2nd edition*, Series in Cognition and Perception. Academic Press, 1999.

[28] D. Cliff. Hang the dj: Automatic sequencing and seamless mixing of dance-music tracks. Report HPL-2000-104 Hewlett Packard, 2000.

[29] M. Clynes and J. Walker. Neurobiologic functions of rhythm, time, and pulse in music. In Clynes M. and Walker J., editors, *Music, mind, and brain: The neuropsychology of music*, pages 171–216. Plenum, 1982.

[30] M. Clynes and J. Walker. Music as time's measure. *Music Perception*, 4(1):85–119, 1986.

[31] MPEG-7 consortium. Iso working draft - information technology - multimedia content description interface - part4: Audio. Report ISO/IEC 15938-4:2001, 2001.

[32] G. Cooper and L. Meyer. *The rhythmic structure of music.* University of Chicago Press, Chicago, 1960.

[33] R. Dannenberg and B. Mont-Reynaud. Following an improvisation in real-time. Proc. International Computer Music Conference, 1987.

[34] P. Desain and S. de Vos. Autocorrelation and the study of musical expression. Proc. International Computer Music Conference, 1990.

[35] P. Desain and H. Honing. The quantization of musical time: a connectionist approach. In Desain P. and Honing H., editors, *Music and Connectionism.* MIT Press, 1991.

[36] P. Desain and H. Honing. Tempo curves considered harmful. a critical review of the representation of timing in computer music. Proc. International Computer Music Conference, 1991.

[37] P. Desain and H. Honing. Computational models of beat induction: The rule-based approach. *Journal of New Music Research*, 28(1), 1999.

[38] P. Desain and L. Windsor. *Rhythm Perception and Production.* Swets and Zeitlinger, 2000.

[39] S. Dixon. A beat tracking system for audio signals. Proc. Conference on Mathematical and Computational Methods in Music (Diderot), 1999.

[40] S. Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30(1), 2001.

[41] S. Dixon and E. Cambouropoulos. Beat tracking with musical knowledge. Proc. European Conference on Artificial Intelligence, 2000.

[42] S. Dixon and W. Goebl. Pinpointing the beat: Tapping to expressive performances. Proc. International Conference on Music Perception and Cognition, 2002.

[43] S. Dixon, E. Pampalk, and G. Widmer. Classification of dance music by periodicity patterns. Proc. International Conference on Music Information Retrieval, 2003.

[44] C. Drake. Reproduction of musical rhythms by children, adult musicians and adult non-musicians. *Perception and Psychophysics*, 53(1), 1993.

[45] C. Drake and D. Bertrand. The quest for universals in temporal processing of music. *Annals of the New York Academy of Science*, pages 17–27, 2001.

[46] C. Drake, L. Gros, and A. Penel. How fast is that music? the relation between physical and perceived tempo. In Yi S., editor, *Music, Mind and Science*. Seoul National University Press, 1999.

[47] C. Drake, M. Jones, and C. Baruch. The development of rhythmic attending in auditory sequences: attunement, referent period, focal attending. *Cognition*, 77:251–288, 2000.

[48] C. Drake, A. Penel, and E. Bigand. Why musicians tap slower than non-musicians. In Drake C., Penel A., and Bigand E., editors, *Rhythm Perception and Production*. Swets and Zeitlinger, 2000.

[49] C. Duxbury, J. Bello, M. Davies, and M. Sandler. Complex domain onset detection for musical signals. Proc.of the Digital Audio Effects Conference, 2003.

[50] D. Eck, M. Gasser, and R. Port. Dynamics and embodiment in beat induction. In P. Desain and L. Windsor, editors, *Rhythm Perception and Production*. Swets and Zeitlinger, 2000.

[51] D. FitzGerald, E. Coyle, and B. Lawlor. Sub-band independent subspace analysis for drum transcription. Proc. Digital Audio Effects Conference, 2002.

[52] J. Foote, M. Cooper, and U. Nam. Audio retrieval by rhythmic similarity. Proc. International Conference on Music Information Retrieval, 2002.

[53] J. Foote and S. Uchihashi. The beat spectrum: A new approach to rhythm analysis. Proc. International Conference on Multimedia and Expo, 2001.

[54] P. Fraisse. Rhythm and tempo. In D. Deutsch, editor, *The Psychology of Music*, Series in Cognition and Perception, book chapter 6. Academic Press, 1982.

[55] J. Franklin. Multi-phase learning for jazz improvisation and interaction. Proc. Eighth Biennial Symposium on Arts and Technology (Perception and Interaction in the Electronic Arts), 2001.

[56] A. Friberg and J. Sundström. Swing ratios and ensemble timing in jazz performances: Evidence for a common rhythmic pattern. *Music Perception*, 19(3), 2002.

[57] A. Gabrielsson. Similarity ratings and dimension analyses of auditory rhythm patterns. part i. *Scandanavian Journal of Psychology*, (14):138–160, 1973.

[58] A. Gabrielsson. Similarity ratings and dimension analyses of auditory rhythm patterns, part ii. *Scandanavian Journal of Psychology*, (14):161–176, 1973.

[59] M. Gasser, D. Eck, and R. Port. Meter as mechanism: a neural network that learns metrical patterns. *Connection Science*, 1, 1999.

[60] E. Gomez, F. Gouyon, P. Herrera, and X. Amatriain. Mpeg-7 for content-based music processing. Proc. WIAMIS Special session on Audio Segmentation and Digital Music, 2003.

[61] E. Gomez, F. Gouyon, P. Herrera, and X. Amatriain. Using and enhancing the current mpeg-7 standard for a music content processing tool. Proc. Audio Engineering Society, 114th Convention, 2003.

[62] E. Gomez, A. Klapuri, and B. Meudic. Melody description and extraction in the context of music content processing. *Journal of New Music Research*, 32(1):23–41, 2003.

[63] M. Goto. An audio-based real-time beat tracking system for music with or without drums. *Journal of New Music Research*, 30(2), 2001.

[64] M. Goto and Y. Muroaka. Music understanding at the beat level - real-time beat tracking for audio signals. Proc. International Joint Conferences on Artificial Intelligence Workshop on Computational Auditory Scene Analysis, 1995.

[65] M. Goto and Y. Muroaka. A real-time beat tracking system for audio signals. Proc. International Computer Music Conference, 1995.

[66] M. Goto and Y. Muroaka. Real-time rhythm tracking for drumless audio signals: Chord change detection for musical decisions. Proc. International Joint Conferences on Artificial Intelligence, Workshop on Computational Auditory Scene Analysis, 1997.

[67] F. Gouyon. *Extraction automatique de descripteurs rythmiques dans des extraits de musiques populaires polyphoniques.* Thesis/dissertation, IRCAM, Paris, 2000.

[68] F. Gouyon, L. Fabig, and J. Bonada. Rhythmic expressiveness transformations of audio recordings: swing modifications. Proc. International Conference on Digital Audio Effects, 2003.

[69] F. Gouyon and P. Herrera. A beat induction method for musical audio signals. Proc. WIAMIS Special session on Audio Segmentation and Digital Music, 2003.

[70] F. Gouyon and P. Herrera. Determination of the meter of musical audio signals: Seeking recurrences in descriptor of beat segment descriptors. Proc. Audio Engineering Society, 114th Convention, 2003.

[71] F. Gouyon, P. Herrera, and P. Cano. Pulse-dependent analyses of percussive music. Proc. AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio, 2002.

[72] F. Gouyon and B. Meudic. Towards rhythmic content processing of musical signals - fostering complementary approaches. *Journal of New Music Research*, 32(1):41–65, 2003.

[73] F. Gouyon, F. Pachet, and O. Delerue. On the use of zero-crossing rate for an application of classification of percussive sounds. Proc. Digital Audio Effects conference, 2000.

[74] J. Herre, M. Cremer, C. Uhle, and J. Rohden. Proposal for a core experiment on audiotempo. Report MPEG2001/8415, 2002.

[75] P. Herrera, A. Dehamel, and F. Gouyon. Automatic labeling of unpitched percussion sounds. Proc. Audio Engineering Society, 114th Convention, 2003.

[76] L. Hofmann-Engl. Rhythmic similarity: A theoretical and empirical approach. Proc. International Conference on Music Perception and Cognition, 2002.

[77] H. Honing. Issues in the representation of time and structure in music. *Contemporary music review*, 9:221–239, 1993.

[78] H. Honing. From time to time: The representation of timing and tempo. *Computer Music Journal*, 25(3):50–61, 2001.

[79] V. Iyer. *Microstructures of Feel, Macrostructures of Sound: Embodied Cognition in West African and African- American Musics.* Thesis/dissertation, University of California, Berkeley, 1998.

[80] M. Jones and M. Boltz. Dynamic attending and responses to time. *Psychological Review*, 96(3), 1989.

[81] D. Kirovski and H. Attias. Beatid: Identifying music via beat analysis. Proc. International Workshop on Multimedia Signal Processing, 2002.

[82] A. Klapuri. Musical meter estimation and music transcription. Proc. Cambridge Music Processing Colloquium, 2003.

[83] E. Lapidaki. *Consistency of tempo judgments as a measure of time experience in music listening.* Thesis/dissertation, Northwestern University, Evanston, Illinois, 1996.

[84] E. Lapidaki. Stability of tempo perception in music listening. *Music Education Research*, 2(1), 2000.

[85] E. Large and E. Kolen. Resonance and the perception of musical meter. *Connection Science*, (6):177–208, 1994.

[86] E. Large and C. Palmer. Perceiving temporal regularity in music. *Cognitive Science*, 26:1–37, 2002.

[87] J. Laroche. Estimating tempo, swing and beat locations in audio recordings. Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2001.

[88] J. Laroche. Efficient tempo and beat tracking in audio recordings. *Journal of the Acoustical Society of America*, 51(4):226–233, 2003.

[89] K. Lemström, G. Wiggins, and Meredith D. A three-layer approach for music retrieval in large databases. Proc. International Symposium on Music Information Retrieval, 2001.

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

[91] D. Levitin and P. Cook. Absolute memory for musical tempo: Additional evidence that auditory memory is absolute. *Perception and Psychophysics*, 58, 1996.

[92] A. Lindsay and W. Kriechbaum. There's more than one way to hear it: Multiples representations of music in mpeg-7. *Journal of New Music Research*, 28(4):364–372, 1999.

[93] J. London. Rhythm. In *The new Grove Dictionary of Music and Musicians, 2nd edition.* 2001.

[94] C. Longuet-Higgins. *Mental processes.* MIT Press, Cambridge, 1987.

[95] C. Longuet-Higgins and C. Lee. Perception of musical rhythms. *Perception*, 11:115–128, 1982.

[96] G. Madison. Different kinds of groove in jazz and dance music as indicated by listeners' ratings. Proc. International Symposium on Systematic and Comparative Musicology and International Conference on Cognitive Musicology, 2001.

[97] G. Madison and B. Merker. On the limits of anisochrony in pulse attribution. *Psychological Research*, 66(3):201–207, 2002.

[98] J. Maher and J. Beauchamp. Fundamental frequency estimation of musical signals using a two-way mismatch procedure. *Journal of the Acoust. Soc. of America*, 95, 1993.

[99] D. O' Maidin and M. Cahill. Score processing for mir. Proc. International Symposium on Music Information Retrieval, 2001.

[100] J. McAuley. *Perception of time as phase: Towards an adaptive- oscillator model of rhythmic pattern processing.* Thesis/dissertation, Indiana University, Bloomington, 1995.

[101] J. McAuley and P. Semple. The effect of tempo and musical experience on perceived beat. *Australian Journal of Psychology*, 51(3):176–187, 1999.

[102] C. Meek and W. Birmingham. Thematic extractor. Proc. International Symposium on Music Information Retrieval, 2001.

[103] D. Moelants. Preferred tempo reconsidered. Proc. of the International Conference on Music Perception and Cognition, 2002.

[104] B. Mont-Reynaud and M. Goldstein. On finding rhythmic patterns in musical lines. Proc.International Computer Music Conference, 1985.

[105] F. Pachet. Interacting with a musical learning system: The continuator. Proc. International Conference on Music and Artificial Intelligence, 2002.

[106] F. Pachet, O. Delerue, and F. Gouyon. Automatic extraction of rhythmic structure from music. Report Sony Research Forum, 2000.

[107] F. Pachet, P. Roy, and D. Cazaly. A combinatorial approach to content-based music selection. Proc. IEEE International Conference on Multimedia Computing and Systems, 1999.

[108] C. Palmer. Music performance. *Annual review of psychology*, 48:115–138, 1997.

[109] E. Pampalk, A. Rauber, and D. Merkl. Content-based organization and visualization of music archives. Proc. ACM International Conference on Multimedia, 2002.

[110] R. Parncutt. A perceptual model of pulse salience and metrical accent in musical rhythms. *Music Perception*, 11(4):409–464, 1994.

[111] C. Raphael. Music plus one: A system for flexible and expressive musical accompaniment. Proc. International Computer Music Conference, 2001.

[112] C. Raphael. A hybrid graphical model for rhythmic parsing. *Artificial Intelligence*, 137(1-2):217–238, 2002.

[113] B. Repp. Probing the cognitive representation of musical time: structural constraints on the perception of timing perturbations. *Cognition*, 44:241–281, 1992.

[114] D. Rosenthal. *Machine rhythm: Computer emulation of human rhythm perception.* Thesis/dissertation, MIT, 1992.

[115] E. Scheirer. Pulse tracking with a pitch tracker. Proc. IEEE Worshop on Applications of Signal Processing to Audio and Acoustics, 1997.

[116] E. Scheirer. Tempo and beat analysis of acoustic musical signals. *Journal of the Acoustical Society of America*, 103(1), 1998.

[117] E. Scheirer. *Music-listening systems.* Thesis/dissertation, MIT Cambridge, 2000.

[118] E. Scheirer and M. Slaney. Construction and evaluation of a robust multifeature speech/music discriminator. Proc. IEEE-ICASSP, pages 1331–1334, 1997.

[119] A. Schloss. *On the automatic transcription of percussive music - From acoustic signal to high-level analysis.* Thesis/dissertation, CCRMA, Stanford University, 1985.

[120] J. Seppänen. *Computational models of musical meter recognition.* Thesis/dissertation, Tampere University of Technology, 2001.

[121] J. Seppänen. Tatum grid analysis of musical signals. Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2001.

[122] W. Sethares and T. Staley. Meter and periodicity in musical performance. *Journal of New Music Research*, 30(2):149–158, 2001.

[123] I. Shmulevich and D. Povel. Measures of temporal pattern complexity. *Journal of New Music Research*, 19(1), 2000.

[124] L. Smith. Modelling rhythm perception by continuous time- frequency analysis. Proc. International Computer Music Conference, 1996.

[125] L. Smith and P. Kovesi. A continuous time-frequency approach to representing rhythmic strata. Proc. International Conference on Music Perception and Cognition, 1996.

[126] L. Smith and R. Medina. Discovering themes by exact pattern-matching. Proc. International Symposium on Music Information Retrieval, 2001.

[127] J. Snyder and C. Krumhansl. Tapping to ragtime: Cues to pulse finding. *Music Perception*, 18(4):455–489, 2001.

[128] A. Tanguiane. A principle of correlativity of perception and its applications to music recognition. *Music Perception*, 11, 1994.

[129] M. Thiemel. Accent. In *The new Grove Dictionary of Music and Musicians, 2nd edition.* 2001.

[130] H. Thornburg. Bayesian segmentation and rhythm tracking. Draft report CCRMA Stanford University, 2001.

[131] H. Thornburg and F. Gouyon. A flexible analysis/synthesis method for transients. Proc. International Computer Music Conference, 2000.

[132] P. Todd, C. Lee, and D. Boyle. A sensory-motor theory of beat induction. *Psychological Research*, 66(1):26–39, 2002.

[133] P. Toiviainen. Real-time recognition of improvisations with adaptive oscillators and a recursive bayesian classifier. *Journal of New Music Research*, 30(2), 2001.

[134] G. Tzanetakis, G. Essl, and P. Cook. Automatic musical genre classification of audio signals. Proc. International Symposium for Audio Information Retrieval, 2001.

[135] G. Tzanetakis, G. Essl, and P. Cook. Human perception and computer extraction of musical beat strength. Proc. Digital Audio Effects Conference, 2002.

[136] B. Vercoe. Computational auditory pathways to music understanding. In I. Deliège and J. Sloboda, editors, *Perception and Cognition of Music*, pages 307–326. Psychology Press, 1997.

[137] H. Vinet, P. Herrera, and F. Pachet. The cuidado project. Proc. International Symposium on Music Information Retrieval, 2002.

[138] C. Waadeland. It don't mean a thing if it ain't got that swing - simulating expressive timing by modulated movements. *Journal of New Music Research*, 30(1), 2001.

[139] Y. Wang. A beat-pattern based error concealment scheme for music delivery with burst packet loss. Proc. International Conference on Multimedia and Expo, 2001.

[140] Y. Wang and M. Vilermo. A compressed domain beat detector using mp3 audio bitstreams. Proc. ACM Multimedia, 2001.

[141] P. Xiang. A new scheme for real-time loop music production based on granular similarity and probability control. Proc. Digital Audio Effects Conference, 2002.

[142] Y. Yamada, T. Kimura, T. Funada, and G. Inoshita. An apparatus for detecting the number of beats. USA Patent 5,614,687, 1995.

[143] M. Yeston. *The stratification of musical rhythm.* Yale University Press, New Haven, 1976.

[144] A. Zils, F. Pachet, O. Delerue, and F. Gouyon. Automatic extraction of drum tracks from polyphonic music signals. Proc. International Conference on Web Delivery of Music, 2002.