4

Beat Tracking and Musical Metre Analysis

Stephen Hainsworth

Tillinghast-Towers Perrin, 71 High Holborn, London WC1V 6TH, UK
swh21@cantab.net

4.1 Introduction

Imagine you are sitting in a bar and your favourite song is played on the jukebox. It is quite possible that you might start tapping your foot in time to the music. This is the essence of beat tracking and it is a quite automatic and subconscious task for most humans. Unfortunately, the same is not true for computers; replicating this process algorithmically has been an active area of research for well over twenty years, with reasonable success achieved only recently.

Before progressing further, it would be useful to define beat tracking clearly. This involves estimating the possibly time-varying tempo and the locations of each beat. In engineering terms, this is the frequency and phase of a time-varying signal, the phase of which is zero at a beat location (i.e., where one would tap one's foot). When musical audio signals are used as an input, the aim of 'beat-tracking' algorithms is to estimate a set of beat times from this audio which would match those given by a trained human musician. In the case where a notated score of the music exists, the musician is used as a proxy for it (hopefully the musician's set of beats would align with those in the score). Where no score exists, the musician's training must be accepted to return a metre equivalent to how the music would be notated. Note that this implies that it is the intended rather than the percieved beat structure that is the focus here.

Beat tracking as just described is not the only task possible. Some algorithms attempt only tempo analysis—finding the average tempo of the sample; others attempt to find the phase of the beat process and hence produce a 'tapping signal'. Meanwhile, some methods also attempt a full rhythmic transcription and attempt to assign detected note onsets to musically relevant locations in a temporally quantized representation. This is often considered in terms of the score which a musician would be able to read in order to recreate the musical example [352]. MIDI signals are also commonly used as

inputs and, assuming that the signal is of an expressive performance, all of the above tasks are again possible aims.

This chapter is organized as follows. Section 4.2 gives an overview of methods and approaches to beat tracking. However, as with any engineering system which is trying to replicate a real-world process, it is useful to examine the actual process before trying to build a model. Section 4.3 of this chapter briefly discusses some of the musical background behind beat tracking. Next, detection of onsets in musical audio signals is discussed in Section 4.4 before some of the more influential approaches to beat tracking are presented in Sections 4.5 to 4.9. Probabilistic models are examined in more detail in Section 4.10. Section 4.11 presents trials of various algorithms on a comprehensive test database and conclusions will be drawn in Section 4.12.

There are many immediate and commercial applications of a successful beat tracking program which have perhaps motivated some of the research. Some of these are: automatic accompaniment of a solo performance [538], synchronization of two music streams (e.g. for DJing [94]), correctly timed recovery from CD skipping (see [660] for a similar application), intelligent time stretching of musical samples [151], determination of good points for looping algorithms (useful for studio samplers which are heavily utilized in the creation of dance music) and adding tempo synchronous effects. Other uses include database retrieval [633] and metadata generation [566], provision of a 'rhythmic similarity' function to listeners (either in playback or for purchase recommendation) and rhythmic expressiveness transformations (e.g. adding swing to a musical example [244]). In addition, beat tracking can form a good basis for any automated transcription program (e.g. [126], [231], [353], [611]) from which to begin its analysis.

4.2 Summary of Beat-Tracking Approaches

Beat tracking with computers has been an active area of research since the early 1980s, though psychological models of human rhythmic perception predate this. The early work was undertaken in the fields of music perception and computer science, though the emphasis shifted towards engineering and statistics as computing power increased.

As a result of this paradigm shift, the aims and approaches of the methods described below vary considerably. It would hence be useful to categorize them. The first and most important distinction is by type of input; most of the earlier algorithms for beat tracking used a symbolic[1] or MIDI input while audio signals have been used more recently. This is at least partly because the signal processing required to extract rhythmic cues from the audio was beyond the power of early computers. It should be noted, however, that many of the more recent methods implicitly convert an audio stream to a set of MIDI-type inputs via the use of a pre-processing onset-detection algorithm.

[1] Symbolic data usually consists of a quantized set of note start times.

The second important differentiation between approaches is the intended purpose of the algorithm. Much of the early work was conducted with the music psychology goal of understanding how humans perceive music and attempting to model this. Other approaches have goals based more in engineering and attempt to capture information in the signal without direct reference to human perception. Specifically, those studies undertaken within the framework of automated transcription attempt to return to the underlying score rather than any human perception of the performance.

The next major distinction between the algorithms is the broad approach used. Categorizations here could include

- rule-based;
- autocorrelative;
- oscillating filters;
- histogramming;
- multiple agent;
- probabilistic;

though there are methods which do not fall neatly into any of these classes. Descriptions of these six broad approaches can be found later in Sections 4.5 to 4.10.

Another, more subtle method of classifying algorithms is by *causal* [572] operation. In a causal model, the estimate of the metre at a given time depends only on past and present data. A non-causal model allows the use of future data and backward decoding. Another way to consider it is that a causal algorithm attempts to mimic human tapping and uses data only up to the current time to decide whether a beat should be marked or not. Semi-causal algorithms have also been produced where the estimate is made after a short time-lag, typically around 20 ms. These can often give a 'strict' causal estimate but at the cost of optimality.

Finally, the algorithms can be grouped by their intended output; some only produce a best estimate of tempo while others evaluate phase as well, therefore giving the beat. Gouyon [242] separates these into *tempo induction*, the estimation of the most likely tempo given a segment of data, and *beat tracking*, which is the following of the beat through an extended example. Some methods also extract the super-beat and/or sub-beat structure (that is, slower and faster pulses than the beat, respectively), while some only attempt estimation of either the super- or sub-beat and not the actual beat.

Table 4.1 summarizes some methods found in the literature, indicating the type of input used and any causal nature. Others which do not fall into any particular category are Sethares and Staley [578], Smith [601], Miller et al. [464], and Bilmes [37]. Two other studies which present surveys or reviews of beat tracking are [243], [249].

Table 4.1. Summary of beat-tracking methods. Key for *Input* column: A = audio, M = MIDI, and S = symbolic.

Approach	Author and year [Ref]	Input	Causal
1) rule-based	Steedman 1977 [607]	S	
	Longuet-Higgins & Lee 1982 [418]	S	
	Povel & Essens 1985 [529]	S	
	Parncutt 1994 [497]	S	
	Temperley & Sleator 1999 [622]	M	
	Eck 2000 [165]	S	
2) autocorrelative	Brown 1993 [55]	S	
	Tzanetakis et al. 2001 [632]	A	
	Foote 2001 & Uchihashi [194]	A	
	Mayor 2001 [445]	A	
	Paulus & Klapuri 2002 [503]	A	
	Alonso et al. 2003 [15]	A	
	Davies & Plumbley 2004 [118]	A	X
3) oscillating filters	Large 1994 [390]	M	X
	McAuley 1995 [450]	M	X
	Scheirer 1998 [564]	A	X
	Toiviainen 1998 [626]	M	X
	Eck 2001 [166]	A	
4) histogramming	Gouyon et al. 2001 [245]	A	
	Seppänen 2001 [573]	A	X
	Wang & Vilermo 2001 [661]	A	
	Uhle & Herre 2003 [635]	A	
	Jensen & Andersen 2003 [318]	A	X
5) multiple agent	Allen & Dannenberg 1990 [14]	M	
	Rosenthal 1992 [546]	M	
	Goto et al. 1994 [221]	A	X
	Dixon 2001 [148]	A/M	
6) probabilistic	Laroche 2001 [392]	A	
	Cemgil et al. 2000 [75], [76]	M	
	Raphael 2001 [537]	A/M	
	Sethares et al. 2004 [577]	A	
	Hainsworth & Macleod 2003 [266]	A	
	Klapuri 2003 [349]	A	X
	Lam & Godsill 2003 [386]	A	
	Takeda et al. 2004 [617]	M	
	Lang & de Freitas 2004 [387]	A	

4.3 Musical Background to Rhythmic Structure

Typically, music consists of sounds generated concurrently by a number of different sources (usually musical instruments of varying kinds). These are organized in a temporal manner, the structure of which forms the 'rhythm' of the piece. Most music has a coherent temporal structure, as this is pleasing to most listeners. Thus the rhythm of a piece more readily lends itself to analysis than the harmonic structure, which can often be much more complex.

At the top level, the rhythm describes the timing relationships between musical events within a piece. The Oxford English Dictionary [624] gives the definition of rhythm as

a. *The aspect of musical composition concerned with periodical accent and the duration of notes.*
b. *A particular type of pattern formed by this.*

Cooper and Meyer [102] define it as the way in which one or more unaccented beats are grouped in relation to an accented one. The term *metre* is sometimes used in an equivalent manner to rhythm, though in music psychology it takes on a different meaning. Here, metre is the number of pulses between the more or less regularly recurring accents in a piece of music [102]. Thus, the metre is a constituent of the rhythm of a piece of music; however, the grouping of accents into patterns and the interaction of this process and the metre are closer to describing the rhythm of a piece.

Some further analysis can be made; Bilmes [37] breaks down musical timing into four subdivisions. The first is the hierarchical *metrical structure*, which relates the idealized timing relationships as they would exist in a musical score, i.e., quantized to a grid.[2] Next is *tempo variation*, which gives the possibly time-varying speed at which the events are sounded. Another level of abstraction gives *timing deviations*, which are individual timing discrepancies around the time-varying metrical grid (e.g. 'playing ahead of the beat'; swing[3] can also be considered a timing deviation). Finally there are *arrhythmic sections*, where there is no established rhythm. These will be ignored from now on as fundamentally impossible to analyse rhythmically, except as a collection of unrelated note start times.

The metrical structure can also be broken down into a set of three hierarchical levels. Klapuri [349] describes the *beat* or *tactus* as the preferred (trained) human tapping tempo and is what most of the beat-tracking algorithms attempt to extract at a minimum. This usually corresponds to the 1/4 note or *crotchet* when written out in common notation, though this is not always the case: in fast jazz music, the pulse is often felt at half this rate (1/2 note or *minim*), while hymns are often notated with the beat given in minims.

[2]Dixon [148] uses the term 'scoretime', measured in beats since the start of the sample to describe this representation.

[3]Swing is a style where the second 1/8th note of every beat is slightly delayed; it is a characteristic of jazz and some rock music.

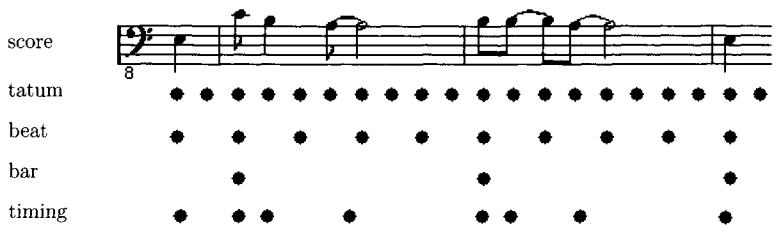


Fig. 4.1. Diagram of relationships between metrical levels.

However it is notated, the rate at which beats occur defines the tempo of the music [404].

At a lower level than the beat is the *tatum*, which is defined to be the shortest commonly occurring time interval. This is often defined by the 1/8th notes (*quavers*) or 1/16th notes (*semiquavers*). Conversely, the main metrical level above the beat is that of the *bar* or *measure*. This is related to the rate of harmonic change within the piece, usually to a repeated pattern of emphasis and also notational convention. Fig. 4.1 gives a diagrammatic representation of the above discussion. Included is a set of expressive timings for the score given. While obvious, it should also be noted that onsets do not necessarily fall on beats and that beats do not necessarily have onsets associated with them.

From here, metrical levels below the beat, including the tatum level, will be termed the *sub-beat* structure, while the converse—bar levels, etc.—will be labelled the *super-beat* structure. In between the tatum and beat, there may be intermediary levels, usually related by multiples of two or three (compound time divides the beat into three sub-beats, for instance). The same applies between the beat and bar levels. Gouyon [242] gives a comprehensive discussion of the semantics behind the words used to describe rhythm, pointing out many of the dualities and discrepancies of terminology. One point he raises is that the terms beat or pulse are commonly used to describe both an individual element in a series and the series as a whole.

An interesting point is raised by Honing [294], who discusses the duality between tempo variations and timing: the crux of the problem is that a series of expressively timed notes can be represented either as timing deviations around a fixed tempo, as a rapidly varying tempo, or as any intermediate pairing. This is a fundamental problem in rhythm perception and most algorithms arrive at an answer which lies between the extremes by applying a degree of smoothing to the processes—this usually means that estimated tempo change over an analysis segment is constrained by the algorithm and any additional error in expected timing of onsets is modelled as a timing deviation.

This leads to the concept of *quantization*, which is the process of assessing with which score location an expressively timed onset should be associated. Here, *score location* refers to the timing position the onset would take

when notated upon a traditional Western musical score or other equivalent representation. However, for most purposes, it can be reduced to the number of beats (and sub-beats) since the start of the sample. Quantization is an important problem and other specific studies on this topic include Cemgil et al. [75] and Desain and Honing [142].

The phase of the beat is determined by a series of stresses or accents, termed *phenomenal* accents [404], [497] or *salience* [148], [529]. These usually correspond to note starts, though not uniquely—it is possible that note ends or changes in intensity can indicate beat, too. It is generally assumed that stresses fall on the beat more often than not and that significant chordal changes also do so. While this is not always the case, and indeed many musical styles exhibit *syncopation*, where there are off-beat stresses, Steedman notes, 'No event inconsistent with either key or metre will occur in a piece until sufficient framework (of key or time signature) has been established for it to be obvious that it is inconsistent' [607]. There are counter-examples to this statement, but it holds in the main.

There is a large body of literature in the music psychology and neuroscience fields on how humans perceive rhythm. In particular, there is some literature on human tapping processes and the behaviour of musicians versus non-musicians (e.g. [155]). However, as the aim of most audio beat trackers is to return to the underlying score or performance intentions rather than replicate the perceptions of a listener, the general psychology literature will not be discussed in detail here.

4.4 Onset Detection

While the metre and tempo of a piece of music can be thought of as a constantly evolving signals, the musical events which underpin this are the starts of notes, and these are discrete events. Many methods for beat tracking deal with symbolic or MIDI data which represent these note start (onset) times. It is highly possible, and indeed common, to simply attach an onset detector to find the note starts in an audio signal and then track the resulting set of discrete impulses. When this approach is used, the success of any beat tracker is dependent upon the reliability of the data which is provided as an input. Thus, detecting note starts in the audio can be as important as the actual beat-tracking algorithm.

Note ends, even when played exactly as written in the score, can be ignored as unreliable indications of beat due to reverberation, sustain or at the opposite extreme, staccato events, where the note is cut short.

Note sources generally fall into two categories: harmonic and percussive. The former produce sounds which would be regarded as notes, have an identifiable pitch and harmonically related partials. Percussive sounds, in comparison, are more analogous to noise clouds. Drums and cymbals are the obvious examples of this class. It should be noted that many (indeed most) pitched

instruments have a transient onset which has much in common with percussive sounds. Percussive sounds are usually characterized by significant increases in signal energy (a 'transient') and methods for detecting this type of musical sound are relatively well developed. Harmonic change with little associated energy variation is much harder to reliably detect and has received less attention in the literature. Two recent studies of onset detection are Bello et al. [30] and Collins [95].

While the discussion below assumes that a hard detection decision is made as to whether an onset is present at a given location, the beat trackers discussed below which work on continuous detection functions also need to transform the raw audio into something more amenable. They also process the signal in ways similar to those described below but do not perform the step of making hard onset detection decisions, instead leaving this to the later beat-tracking process. The hard-decision onset detection method yields a set of discrete onset times, whereas the latter method results in a continuous function from which beat tracking is performed.

4.4.1 Transient Event Detection

Transient events, such as drum sounds or the start of notes with a significant energy change (e.g. piano, guitar), are easily detected by examining the signal envelope. A typical approach, which is an adaptation of methods used by a variety of other researchers [148], [392], [564], proceeds as follows: An energy envelope function $\mathrm{E}_j(t)$ is formed by summing the power of frequency components in the spectrogram for each time slice over the range required:

$$\mathrm{E}_j(n) = \sum_{k \in \kappa_j} |\mathsf{STFT}_x^{\mathrm{w}}(n,k)|^2, \tag{4.1}$$

where $\mathsf{STFT}_x^{\mathrm{w}}(k,n)$ is the short-time Fourier transform (STFT) of the signal $x(n)$ with rectangular window w centred at time n; k is the frequency index (see Chapter 2 for details). Usually analysis frames of about 20 ms are used in computing the energy envelope, with 50–75% overlap between successive frames. Different bands j can be used; for instance, low frequency information covering the range 20–200 Hz is useful to separate. Setting κ_j to the middle range of 200 Hz to 15 kHz covers the majority of the harmonic information; meanwhile, extending over 15–22.05 kHz (assuming a sample rate of 44.1 kHz), the upper band is often generally free from harmonic content but contains a clear indication of any strong transient information [444]. This is contrary to the opinion of Duxbury [164], who claimed that there is no useful information in this range. Many other ways to split the frequency spectrum have also been proposed. One common approach is to use 5–10 sub-bands that are distributed uniformly on a logarithmic frequency scale.

$\mathrm{E}_j(n)$ is not an ideal signal representation for detecting onsets. A potential approach for improving it uses a three-point linear regression to find $\mathrm{D}_j(n)$,

the gradient of $\mathrm{E}_j(n)$, and peaks in this function are detected. The linear regression fits a line $Y_i = a + bX_i + e_i$ to a set of N data pairs; we are only interested in the estimate of b which is given by $\hat{b} = (\sum_{i=1}^{N} X_i Y_i - N\bar{X}\bar{Y})/(\sum_{i=1}^{N} X_i^2 - N\bar{X}^2)$, where $\bar{X}$ and $\bar{Y}$ denote the average of X and Y, respectively. In the case here, X is the equi-spaced set of time indices n in $\mathrm{E}_j(n)$ and Y is the corresponding E_j. In the case where $N = 3$, this reduces to

$$\mathrm{D}_j(n) = \frac{\mathrm{E}_j(n+1) - \mathrm{E}_j(n-1)}{3}. \tag{4.2}$$

It should be noted that the commonly used technique of differencing the signal, where $\mathrm{D}_j(n) = \mathrm{E}_j(n) - \mathrm{E}_j(n-1)$, is simply linear regression with N set to 2. The linear regression approach, like that of Klapuri [347], aims to detect the start of the transient, rather than the moment it reaches its peak power.

$\mathrm{D}_j(n)$ is often called a detection function [30] and is a transformed and reduced signal representation. Subsequent processing needs to detect the onsets contained within this. This is usually done by simply selecting maxima in $\mathrm{D}_j(n)$ and discarding peaks which do not pass a series of tests. Low-energy peaks should be ignored (for instance by testing if they are less than two times the local 1.5-second average of E_j) and peaks can also be ignored if there is a higher-energy peak in the local vicinity[4] by using Dixon's timing criterion [148]. Thresholds and constants are usually heuristically determined and designed to give reasonable performance with a large range of styles. Figure 4.2 shows an example of a peak extraction method. When several sub-bands j are involved, the functions $\mathrm{D}_j(n)$ can be combined by half-wave rectifying and across-band summing before the peak-picking process [37], [347].

4.4.2 Pitched Event Detection

Detection of note starts where there is no associated energy transient (e.g. violins, choral music) has received less attention than the easier problem addressed above. Notable recent exceptions are Klapuri [349], who used very narrow frequency bands to detect changes in frequency; Laurent et al. [395], who used wavelets; Davy and Godsill [123], who took a support vector machine approach; Desobry et al. [143], who furthered Davy's research and also used kernel methods; and Abdallah and Plumbley [1], who used independent component analysis (ICA) to generate a 'surprise' measure followed by an HMM to perform reliable detection. Also, Bello et al. [31] utilized phase inconsistencies in a manner very similar to time reassignment and Duxbury et al. [164] proposed a spectral change distance measure adapted from the Euclidean measure which was then applied to adjacent spectrogram frames. Recently, Duxbury, Bello et al. [162], [163] have combined the previous two approaches into a single measure for detection of harmonic changes via either

[4]This is similar to the psychoacoustic masking thresholds found for humans [475], [694].

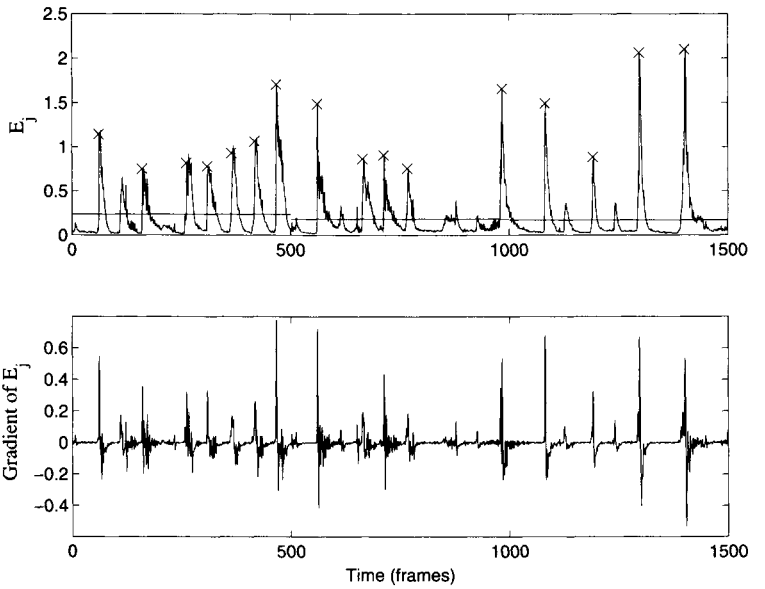


Fig. 4.2. Example of onset detection for transient events. The upper plot shows the energy-based detection function, $\mathrm{E}_j(n)$; also shown are horizontal lines giving the 1.5 s local average of the energy function and $\times$'s showing the detected onsets. The lower plot shows the gradient function $\mathrm{D}_j(n)$ from which peaks are found.

or both of phase inconsistency or spectral change. This method shows improvements over both individual approaches.

An alternative proposed by Hainsworth and Macleod [265] is the so-called modified Kullback–Leibler distance measure given by

$$\mathrm{d}_n(k) = \log_2\left(\frac{|\mathsf{STFT}_x^{\mathrm{w}}(n,k)|}{|\mathsf{STFT}_x^{\mathrm{w}}(n-1,k)|}\right), \tag{4.3}$$

$$\mathrm{d}_{\mathrm{MKL}}(n) = \sum_{k\in\mathcal{K},\mathrm{d}(k)>0} \mathrm{d}_n(k), \tag{4.4}$$

where $\mathsf{STFT}_x^{\mathrm{w}}(n,k)$ is the STFT computed with window w. The measure emphasizes positive energy change between successive frames and $\mathcal{K}$ defines the spectral range over which the distance is evaluated (30 Hz to 5 kHz is suggested as it represents the majority of clear harmonic information in the spectrum). Another advantage of this method is that it also takes into account any transient energy which happens to be present as a useful aid.

A window length of about 90 ms is sufficient to give good spectral resolution. To overcome frame to frame variation, histogramming of five frames (weighted backwards and forwards with a triangular function) before and after the potential change point was used and also a very short frame hop length (namely, 87.5% overlap) was chosen to increase time resolution.

Detection of the peaks in this measure is a separate problem and is discussed more fully in [265]. Figure 4.3 gives an example of detection using this method.

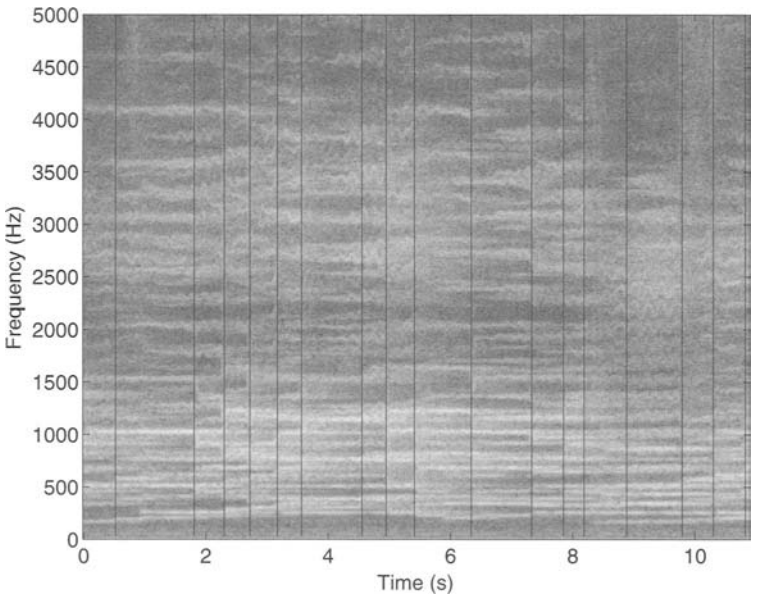


Fig. 4.3. Example of the output from the MKL harmonic change detection measure for an excerpt of Byrd's 4-Part Mass. Onsets were missed at 1 s and 5.9 s while the onset at 8.1 s is mis-estimated and should occur about 0.1 s later.

4.5 Rule-Based Approaches

We shall now discuss a number of broad methodologies for beat tracking in turn. Rule-based approaches were among the earliest used when computers were not capable of running complex algorithms. They tend to be simple and encode sensible music-theoretic rules. Tests were often done by hand and were limited to short examples. Often these did not even have expressive timing added to them and only aimed to extract the most likely pulse given the rhythmic pattern and tempo.

Steedman [607] produced one of the earliest computational models for rhythmic analysis of music. His input was symbolic and with a combination of musical structure recognition (especially melodic repetition) and psychologically motivated processing, he attempted to parse the rhythmic structure of Bach's 'Well Tempered Clavier' set of melodies. Similarly, Longuet-Higgins and Lee [418] proposed a series of psychologically motivated rules for finding

the beat and higher metrical levels from lists of onset times in a monophonic melody. The rules were never implemented by the authors in the original paper for more than five-bar examples, though there have since been several papers by Lee which are summarized by Desain and Honing [141].

Parncutt [497] developed a detailed model for salience or *phenomenal accent*, as he termed it, and used this to inform a beat induction algorithm. Also, he modelled medium tempo preference explicitly and combined these two in a model to predict the tactus for a series of repeated rhythms played at different speeds. Comparison to human preferences was good. Parncutt's focus was similar to that of Povel and Essens [529], while Eck [165] also produced a rule-based model which he compared to Povel and Essens and others.

Temperley and Sleator [622] also used a series of rules to parse MIDI streams for beat structure. They quoted Lerdahl and Jackendoff's generative theory of tonal music (GTTM) [404] as the starting point of their analysis, using the GTTM *event* rule (align beats with event onsets) and *length* rule (longer notes aligned with strong beats). Other rules such as *regularity* and a number based on harmonic content were also bought into play. The aim was to produce a full beat structure from the expressive MIDI input, and a good amount of success was achieved.[5]

4.6 Autocorrelation Methods

Autocorrelation is a method for finding periodicities in data and has hence been used in several studies. Without subsequent processing, it can only find tempo and not the beat phase.

The basic approach is to define an energy function $\mathrm{E}(n)$ to which local autocorrelation is then applied (in frames of length T_w, centred at time n):

$$\mathrm{r}(n,i) = \sum_{u=-(T_\mathrm{w}/2)+1}^{T_\mathrm{w}/2} \mathrm{E}(n+u)\mathrm{E}(n+u-i). \tag{4.5}$$

The value of i which maximizes $\mathrm{r}(n,i)$ should correspond to the period-length of a metrical level. This will often be the beat, but it is possible that if the tatum is strong that autocorrelation will pick this instead.

Tzanetakis et al. [631], [633] included a series of rhythmic features in their algorithm for classification of musical genre. While not specifically extracting a beat, it performs a function similar to beat analysis. Their method was based upon the wavelet transform, followed by rectification, normalization, and summation over different bands before using autocorrelation to extract periodicity. Local autocorrelation functions were then histogrammed over the entire piece to extract a set of features for further use; these tend to show more coherence for rock pieces than for classical music.

[5]Source code for Temperley's method is available in [596].

Foote and Uchihashi [194] used the principle of audio self-similarity to examine rhythmic structure. The assumption was that within the space of a single sub-beat, the sound is approximately constant and therefore the spectrum will have high similarity. They therefore defined a similarity measure as the normalized scalar product (computed over the frequencies k) of the magnitude spectra of frames at times n_i and n_j

$$\mathrm{d}_{\mathrm{Foote}}(n_i, n_j) = \frac{\langle |\mathsf{STFT}_x^{\mathrm{w}}(n_i, k)|, |\mathsf{STFT}_x^{\mathrm{w}}(n_j, k)| \rangle}{\|\mathsf{STFT}_x^{\mathrm{w}}(n_i, k)\| \, \|\mathsf{STFT}_x^{\mathrm{w}}(n_j, k)\|}, \tag{4.6}$$

where w is some window. This produced a two-dimensional plot of similarity between any two frames of the audio signal, which was then autocorrelatively analysed for tempo hypotheses using

$$\mathrm{B}(n_i, n_j) = \sum_{i', j'} \mathrm{d}_{\mathrm{Foote}}(n_{i'}, n_{j'}) \mathrm{d}_{\mathrm{Foote}}(n_i + n_{i'}, n_j + n_{j'}). \tag{4.7}$$

This was extended to be time varying, hence producing their 'beat spectrogram', which was a plot of the local tempo hypothesis versus time.

Other autocorrelation approaches include Mayor [445], who presented a somewhat heuristic approach to audio beat tracking: a simple multiple hypothesis algorithm was maintained which operated on his so-called BPM spectrogram, BPM referring to beats per minute. Also Paulus and Klapuri's method [503] for audio beat analysis utilized an autocorrelation-like function (based on de Cheveigné's fundamental frequency estimation algorithm [135]), which was then Fourier transformed to find the tatum. Higher-level metrical structures were inferred with probability distributions based on accent information derived using the tatum level. This was then used as part of an algorithm to measure the similarity of acoustic rhythmic patterns. Brown [55] used her narrowed autocorrelation method to examine the pulse in musical scores. Davies and Plumbley [118] and Alonso et al. [15], [16] have also produced autocorrelation-based beat trackers.

4.7 Oscillating Filter Approaches

There are two distinct approaches using oscillating filters: In the first, an adaptive oscillator is excited by an input signal and, hopefully, the oscillator will resonate at the frequency of the beat. The second method uses a bank of resonators at fixed frequencies which are exposed to the signal and the filter with the maximum response is picked for the tempo. Beat location can be calculated by examining the phase of the oscillator. This method is particularly suited to causal analysis.

The first, single-filter approach is typified by Large [389], [390], who used a single non-linear oscillator with adaptive parameters for the phase, frequency, and update rate, though these were initialized to the correct settings by hand.

The observed signal is a set of impulses $s(n) = 1$ when there is an onset event and $s(n) = 0$ otherwise. The oscillator is given by

$$o(n) = 1 + \tanh \alpha(\cos 2\pi\phi(n) - 1), \tag{4.8}$$

where $o(n)$ defines an output waveform with pulses at beat locations with width tuned by α; see Fig. 4.4. The phase is given by

$$\phi(n) = \frac{n - n_i}{p}, \tag{4.9}$$

where n_i is the location of the previous beat and p is the period of oscillation (tempo). Crucially, the single oscillator in (4.8) is assumed not to have a fixed period or phase and updates are calculated every time an onset event is observed in $s(n)$ using

$$\Delta n_i = \eta_1 s(n) \tfrac{p}{2\pi} \operatorname{sech}^2\{\alpha(\cos 2\pi\phi(n) - 1)\} \sin 2\pi\phi(n), \tag{4.10}$$

$$\Delta p = \eta_2 s(n) \tfrac{p}{2\pi} \operatorname{sech}^2\{\alpha(\cos 2\pi\phi(n) - 1)\} \sin 2\pi\phi(n), \tag{4.11}$$

where η_1 and η_2 are 'coupling strength' parameters. The update equations enable the estimation of the unknown parameters p and n_i. Marolt [433], however, points out that oscillators can be relatively slow to converge because they adapt only once per observation.

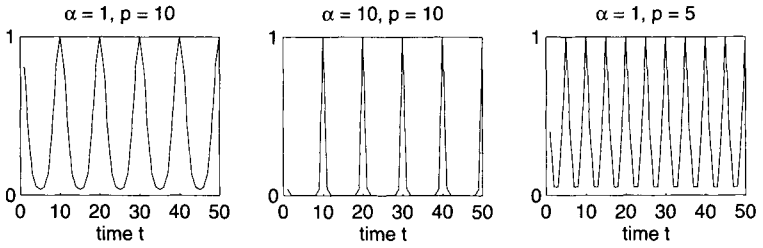


Fig. 4.4. Example output signals $o(t)$ generated using (4.8) for various values of α and p.

Large's test data was a series of impulses derived from expressive MIDI performances and the aim was to track the pulse through the example. An extra level of complexity which allowed the system to continue following the beat was to have a second oscillator 180° out of phase which could take over control from the first if confidence dropped below a certain threshold.

McAuley [450] presented a similar adaptive oscillator model to that of Large and indeed compared and contrasted the two models. Similarly, Toiviainen [626] extended Large's model to have short- and long-term adaption mechanisms. The former was designed to cope with local timing deviations while the latter followed tempo changes. It was tested on expressive MIDI

performances. Another variation is that of Eck [166], who used Fitzhugh–Nagumo oscillators (models of neural action) linked by Heaviside coupling functions into networks. His focus was to reproduce the downbeat extraction of Povel and Essens [529] from synthetic onset data. Various authors [166], [389] have also suggested that adaptive filters have neurological plausibility and this is their motivation for its use.

The second approach is typified by Scheirer [564], who produced one of the first systems for beat tracking of musical audio. The difference compared with Large's method is that Scheirer's method implemented a bank of comb filters at different fixed feedback delays and searched for the one which resonated best with the input signal at any given time. It should be noted that the bank responds in a comb-like manner with multiples and subdivisions of the tempo also showing resonation to the signal. Scheirer implemented 150 filters logarithmically spaced between 60 bpm and 240 bpm, where bpm stands for 'beats per minute'. The input audio signal was treated in six sub-bands to find rectified power envelopes as a function of time. Each sub-band was processed by a separate comb-filter bank before the outputs were summed and the oscillator with the greatest response picked as the current tempo. Phase was also considered so as to generate a tapping signal corresponding to the tactus.

The model worked with considerable success, although there remained the problem of a 2–3 second burn-in period needed to stabilize the filters, and also a propensity for the algorithm to switch between tracking the tactus and its subdivisions/multiples since Scheirer did not explicitly address the stability of the beat estimate. Klapuri [349] (see below) capitalized on the latter observation in his method, using a bank of comb-filter resonators as the initial processing method for his system. McKinney and Moelants [452] also found a resonator method for tempo extraction to outperform histogramming and autocorrelation approaches.

4.8 Histogramming Methods

Several approaches have focused on audio beat tracking using histogramming of inter-onset intervals. First, the signal is analysed to extract onsets before the subsequent processing takes place. This was discussed above in Section 4.4. Differences between successive onsets can be used (first-order intervals), though it is more productive to also use the differences between onsets that are further apart (all-order intervals). The motivation for this is that often the successive onsets define the tatum pulse rather than the tactus, which can be better found using onsets spaced further apart. Histogramming has similarities to the autocorrelation approaches of Section 4.6, though with a discrete input rather than the continuous signal used for autocorrelation.

There are various methods of performing the histogramming operation; defining the set of calculated inter-onset intervals (IOIs), denoted o_i, $i = 1, 2, \ldots$, one can follow Seppänen [572] and divide the IOI time axis into J

bins and count the number of IOIs which fall in each: $h(j) = \text{count}(i, |o_i - u(j + 0.5)| < 0.5u)$ where u is the width of a bin. In contrast, Gouyon et al. [247] and Hainsworth [263] treat the IOI data as a set of Dirac delta functions and convolve this with a suitable shape function (e.g. a Gaussian). The resulting function generates a smoothly varying histogram. This is defined as $h(j) = \sum_i o_i * \mathcal{N}(j)$, where $*$ denotes convolution and $\mathcal{N}(j)$ is a suitable Gaussian function (low variance is desirable). Peaks can then be identified and the maximum taken as the tempo. Alternatively, Dixon [148] gives pseudocode for an IOI histogram clustering scheme.

Seppänen [572] produced an archetypal histogramming method. After an onset detection stage, he first extracted tatums via an inter-onset interval histogramming method. He then extracted a large number of features (intended to measure the musical onset salience) with the tatum signal informing the locations for analysis. These features were then used as the input to an algorithm based on pattern recognition techniques to derive higher metrical levels including the pulse and bar lines. Seppänen [573] gives further details of the tatum analysis part of the algorithm. The final thing to note is that the method was the first to be tested on a statistically significant audio database (around three hundred examples, with an average length of about one minute).

Gouyon et al. [247] applied a process of onset detection to musical audio followed by inter-onset interval histogramming to produce a beat spectrum. The highest peak (which invariably corresponded to the tatum) was then chosen as the 'tick'. This was then used to attempt drum sound labelling in audio signals consisting solely of drums [245], to modify the amount of swing in audio samples [244], and to investigate reliable measures for higher beat level discrimination (i.e., to determine whether the beat divided into groups of two or three) [246]. Other histogramming methods include Wang and Vilermo [661], Uhle and Herre [635], and Jensen and Andersen [318], all of which present variations on the general approach and use the results for different applications.

4.9 Multiple Agent Approaches

Multiple agent methods are a computer science architecture. While there is a great deal of variation in the actual implementation and often the finer details are left unreported, the basic philosophy is to have a number of agents or hypotheses which track independently; these maintain an expectation of the underlying beat process and are scored with their match to the data. Low-scoring agents are killed while high-scoring ones may be branched to cover differing local hypotheses. At the end of the signal, the agent with the highest score wins and is chosen. Older multiple agent architectures include the influential model of Allen and Dannenberg [14] and Rosenthal [547]. The

two most notable multiple agent architectures are those of Goto and that of Dixon.

Goto has produced a number of papers on audio beat tracking of which [221], [238], [240] are a good summary. His first method centred on a multiple agent architecture where there were fourteen transient onset finders with slightly varying parameters, each of which fed a pair of tempo hypothesis agents (one of which was at double the tempo of the other). A manager then selected the most reliable pulse hypothesis as the tempo at that instant, thereby making the algorithm causal. Expected drum patterns as a strong prior source of information were used and tempo was tracked at one sub-beat level (twice the speed) as well as the pulse in order to increase robustness.

This method worked well for audio signals with drums but failed on other types of music. Thus, he expanded the original scheme to include chord change detection [240], each hypothesis maintaining a separate segmentation scheme and comparing chords before and after a beat boundary.

Dixon [148] has also investigated beat tracking both for MIDI and audio, with the aim of outputting a sequence of beat times. The algorithm performed well with a MIDI input, and with the addition of an energy envelope onset detection algorithm, it could also be used for audio (though with lower performance). The approach was based upon maintaining a number of hypotheses which extended themselves by predicting beat times using the past tempo trajectory, scored themselves on musical salience, and updated the (local) tempo estimate given the latest observation. The tempo update was a function of the time coherence of the onset, while the salience measure included pitch and chord functions where the MIDI data was available. Hypotheses could be branched if onsets fell inside an outer window of tolerance, the new hypothesis assuming that the onset was erroneous and maintaining an unadjusted tempo. Initialization was by analysis of the inter-onset interval histogram. Dixon has also used his beat tracker to aid the classification of ballroom dance samples by extracting rhythmic profiles [149].

4.10 Probabilistic Models

Probabilistic approaches can have similarities to multiple agent architectures in that the models underlying each can be very similar. However, while the latter use a number of discrete agents which assess themselves in isolation, probabilistic models maintain distributions of all parameters and use these to arrive at the best hypothesis. Thus, there is an explicit, underlying model specified for the rhythm process, the parameters of which are then estimated by the algorithm. This allows the use of standard estimation procedures such as the Kalman filter [41], Markov chain Monte Carlo (MCMC) methods [208], or sequential Monte Carlo (particle filtering) algorithms [22] (see Chapter 2 for an overview of these methods).

This section will concentrate on some of the models developed rather than details of the estimation procedures which are used to evaluate the final answer, as these can often be interchangeable (a point made by Cemgil, who used a variety of estimation algorithms with the same model [77]).

Again, the various methods can be broken down into two general groups: those that work with a set of MIDI onsets (or equivalently a set of onsets extracted from an audio sample) and those that work to directly model a continuous detection function[6] computed from the original signal.

4.10.1 Discrete Onset Models

Those who have worked on the problem include Cemgil et al. [77], who worked with MIDI signals, and Hainsworth [263], who used Cemgil's algorithm as a starting point for use with audio signals.

The crux of the method is to define a model for the sequential update of a tempo process. This is evaluated at discrete intervals which correspond to note onsets. The tempo process has two elements: the first defines the tempo and phase of the beat process. The second is a random process which proposes notations for the rhythm given the tempo and phase. A simple example of this is that, given a tempo, the time between onsets could either be notated as a quaver or a crotchet, one speeding the tempo up and the other requiring it to slow down. The probabilistic model will propose both and see which is more likely, given the past data (and future if allowed).

The model naturally falls into the framework for jump-Markov linear systems where the basic equations for update of the beat process are given by

$$\boldsymbol{\theta}_n = \boldsymbol{\Phi}_n(\gamma_n)\boldsymbol{\theta}_{n-1} + \mathbf{v}_n, \tag{4.12}$$

$$s_n = \mathbf{H}_n\boldsymbol{\theta}_n + \epsilon_n. \tag{4.13}$$

$\{s_n\}$ is the set of observed onset times, while $\boldsymbol{\theta}_n$ is the tempo process at iteration (observed onset) n and can be expanded as

$$\boldsymbol{\theta}_n = \begin{bmatrix} \rho_n \\ \Delta_n \end{bmatrix}. \tag{4.14}$$

ρ_n is the predicted time of the n^{th} observation s_n, and Δ_n is the beat period in seconds, i.e. $\Delta_n = 60/p_n$ where p_n is the tempo in beats per minute. $\boldsymbol{\Phi}_n(\gamma_n)$ is the state update matrix, $\mathbf{H}_n = \begin{bmatrix} 1 & 0 \end{bmatrix}$ is the observation model matrix, and $\mathbf{v}_n$ and ϵ_n are noise terms; these will be described in turn.

The principal problem is one of quantization—deciding to which beat or sub-beat in the score an onset should be assigned. To solve this, the idealized

[6] Strictly speaking, it will be pseudo-continuous due to sampling.

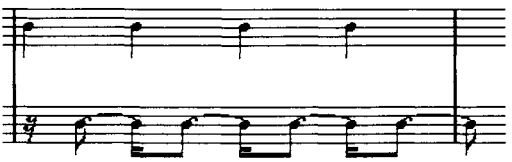


Fig. 4.5. Figure showing two identical isochronous rhythms. The top rhythm is much more likely in a musical notation context than the lower.

(quantized) number of beats between onsets is encoded as the random jump parameter, γ_n, in $\boldsymbol{\Phi}_n(\gamma_n)$,

$$\boldsymbol{\Phi}_n(\gamma_n) = \begin{bmatrix} 1 & \gamma_n \\ 0 & 1 \end{bmatrix}, \tag{4.15}$$

$$\gamma_n = c_n - c_{n-1}. \tag{4.16}$$

While the state transition matrix is dependent upon γ_n, this is a difference term between two absolute locations, c_n and c_{n-1}. c_n is the unknown quantized number of beats between the start of the sample and the n^{th} observed onset. It is this absolute location which is important and the prior on c_n becomes critical in determining the performance characteristics. This can be elucidated by considering a simple isochronous set of onsets—if absolute score location is unimportant, then the model has no way of preferring aligning them to be on the beat over placing them on, say, the first semiquaver of each beat. This is demonstrated in Fig. 4.5. Cemgil [77] broke a single beat into subdivisions of two and used a prior related to the number of significant digits in the binary expansion of the quantized location. In MIDI signals there are no spurious onset observations and the onset times are accurate. In audio signals, however, the event detection process introduces errors both in localization accuracy and in generating completely spurious events. Thus, Cemgil's prior is not rich enough; also, it cannot cope with compound time, triplet figures, or swing. To overcome this, Hainsworth [263] broke down notated beats into 24 sub-beat locations, $c_n = \{1/24, 2/24, \ldots, 24/24, 25/24, \ldots\}$, and a prior was assigned to the fractional part of c_n,

$$\mathsf{p}(c_n) \propto \exp\left(-\lambda \log_2 \{\underline{c}_n\}\right), \tag{4.17}$$

where $\underline{c}_n$ is the denominator of the fraction of c_n when expressed in its most reduced form; i.e., $d(3/24) = 8$, $d(36/24) = 2$, etc. λ is a scale parameter determining the sensitivity of the prior. This is shown graphically in Fig. 4.6. The prior is improper (i.e., it does not sum to unity), which is why $\mathsf{p}(c_n)$ is only expressed as a proportionality. The integer part of c_n increases as the number of beats processed increases. As a result of this, γ_n is always strictly positive; it will be less than 1 if a sub-beat interval is observed, but if there is more than one beat between observed, onsets, γ_n will be greater than 1.

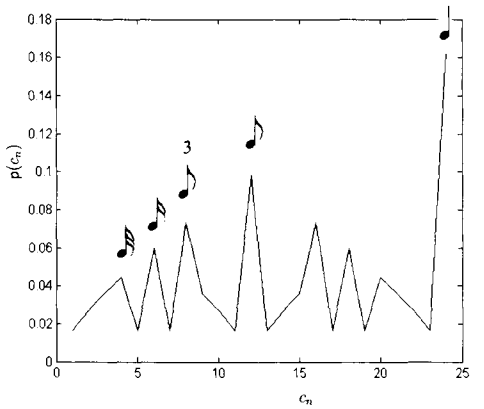


Fig. 4.6. Graphical description of the prior upon c_n. The horizontal axis is the sub-beat location from 1 to 24, while the associated probability $\mathsf{p}(c_n)$ is shown on the vertical axis.

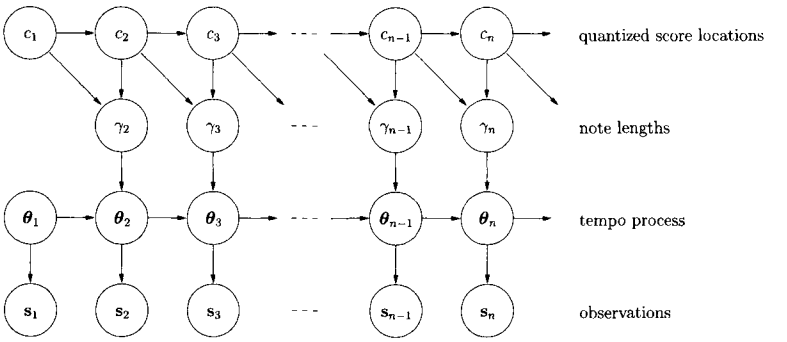


Fig. 4.7. Directed acyclic graph of the jump-Markov linear system beat model. The dependence between c_n and γ_n is deterministic, while other dependencies are stochastic.

The tempo process has an initial prior, $\mathsf{p}(\boldsymbol{\theta}_0)$, associated with it. For the purposes of a general beat-tracking algorithm, it is assumed that the likely tempo range is 60 bpm to 200 bpm and that the prior is uniform within this range.

So far, the model for tempo evolution and proposing a set of onset times has been considered. Finally, the observation model must be specified. s_n is the n^{th} observed onset time and therefore corresponds to the ρ_n in $\boldsymbol{\theta}_n$. Thus, $\mathbf{H}_n = \begin{bmatrix} 1 & 0 \end{bmatrix}$. The state evolution error, $\mathbf{v}_n$, and observation error, ϵ_n, are given suitable distributions—usually for mathematical convenience, these are zero-mean Gaussians with appropriate covariances [26]. The overall model can be summarized by a directed acyclic graph (DAG) as shown in Fig. 4.7. It should be noted that even spurious onsets are assigned a score location.

When working with real-world audio signals, more information than just the onset times can be extracted from the signal, and this can aid the analysis of the rhythm. The most obvious example is the amplitude of onsets while others include a measure of chordal change and other 'salience' features as postulated by Parncutt [497] and Lerdahl and Jackendoff [404]. Hainsworth [263] utilized these in his model as a separate jump-Markov linear system for amplitude and a zero-order Markov model for salience (here, the salience is only a function of the current state and has no sequential dependency). There has been little research into appropriate measures of salience for extracting accents in music; other than the papers mentioned above, Seppänen [573] and Klapuri [349] also proposed features which perform this function.

Given the above system, various estimation procedures exist. Cemgil [77] described the implementation of MCMC methods as well as particle filters to estimate the *maximum a posteriori* (MAP) estimate for the rhythm process, while Hainsworth [263] utilized particle filters to find the MAP estimate for the posterior of interest given by $p(c_{1:n}, \boldsymbol{\theta}_{1:n}, \alpha_{1:n} | s_{1:n}, a_{1:n}, \mathcal{S}_{1:n})$, where $\alpha_{1:n}$ was the underlying amplitude process observed as $a_{1:n}$, and $\mathcal{S}_{1:n}$ was the observed set of saliences. Full details can be found in either of the publications.

Other similar methods include an earlier approach of Cemgil's [79] where what he termed the 'tempogram' (which convolved a Gaussian function with the onset time vector and then used a localized tempo basis-function[7] to extract a measure of tempo strength over time) was tracked with a Kalman filter [41] to find the path of maximum tempo smoothness.

Raphael's methods [537] were based around hidden Markov models where a triple-layered dependency structure was used: quantized beat locations informed a tempo process which in turn informed an observation layer. The Markov transitions were learned between states from training data, and then the rhythmic parse evaluated in a sequential manner to decide which was the most likely tempo/beat hypothesis. This was tested on both MIDI and audio (after onset detection) and success was good on the limited number of examples, though manual correction from time to time was permitted.

Laroche [392], [393] used a maximum likelihood framework to search for the set of tempo parameters which best fit an audio data sample. The input was processed by typical energy envelope difference methods to extract a list of onset times. Inter-onset times (which are phase independent) were then used to provide likelihoods for the 2-D search space with discretized tempo and swing as the two axes. This algorithm has been included in commercially available Creative sound modules for several years. Lang and de Freitas [387] presented a very similar algorithm to that of Laroche but used a continuous signal representation and a slightly more complex estimation procedure.

[7] The tempo basis function was defined as a set of weighted Dirac functions $\psi(t; \tau, \omega) = \sum_{i=-\infty}^{\infty} \alpha_i \delta_{\tau + i\,2^{\omega}}(t)$ at a delay of τ and spaced with frequency (and hence tempo) given by ω.

Hainsworth also presented a second algorithm which is essentially a reformulation of the above but using Brownian motion relations as a base [266]. It was not as successful as the above model. Others include Takeda et al. [617] and Lam and Godsill [386].

4.10.2 Continuous Signal Representations

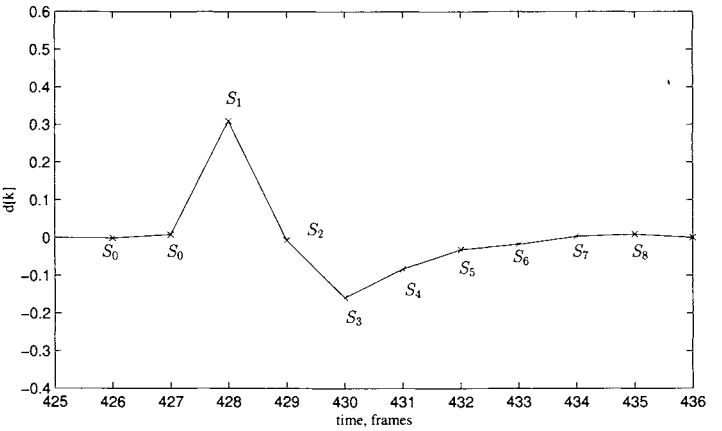
The second approach to tracking the beat with stochastic models uses a detection function and attempts to model this directly instead of extracting onsets first. As such it must have all the elements of the above models, including a tempo process and a model for the likelihood of an onset being present at any given beat or sub-beat location; however, it must also have a model for the signal itself and what is expected at an onset and between these.

Hainsworth [263] proposed a method using particle filters whereby the tempo was modelled as a constant velocity process similar to the one described above and which proposed onsets in a generative manner at likely sub-beat locations. The signal detection function modelled was a differenced energy waveform, utilizing high-frequency information, very similar to $\mathrm{D}_j(t)$ shown in the lower plot of Fig. 4.2.

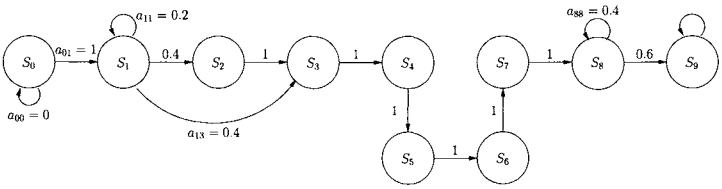
Onset locations can clearly be seen in this signal representation, and on close examination all onsets have a very similar evolution in time which can be well modelled by a hidden Markov model (HMM; see Chapter 2 for a definition). This is performing the task of onset detection. The model used is shown in Fig. 4.8 with each state having a different output distribution (also termed likelihood). For mathematical convenience, these are Gaussians with differing means and variances but sufficiently separated so that the output distribution of state S_1 does not significantly overlap with that of S_0 or S_2, etc. This defines a generative model for the signal—by generative, it is meant that by using a random number generator and the specified distributions, a process with the same statistical properties as the original signal can be generated.

A naive scheme simply generates proposals from the prior distributions, but the Viterbi algorithm (see [654] and Chapter 2) can be used to find the best path through the HMM and also its probability, which simplifies the calculation needed once an onset is hypothesized. The model worked well on the small number of examples tried but required the expected sub-beat structure to be specified by hand for robust performance.

In comparison, Sethares et al. [577] proposed four filtered signals (time domain energy, spectral centroid, spectral dispersion, and one looking at group delay) which were then simply modelled as Gaussian noise with a higher variance at beat locations compared to between them. Looking back at Fig. 4.2, it can clearly be seen where the variance of the generative noise process used to model the signal would be higher. A model similar to those above was used and a particle filter environment chosen for the estimation procedure. The



(a) Data with states superimposed.



(b) Directed acyclic graph of HMM model.

Fig. 4.8. HMM for beat-tracking algorithm with Viterbi decoding included. States S_5, S_6, and S_7 are functionally equivalent to S_4, and S_8 is equivalent to S_0. The null state, S_9, has no observation associated with it, therefore making transition to it highly unattractive.

model did not explicitly include a model for sub-beats but seemed to function well on the data presented.

A somewhat different method for tracking the beat through music was presented by Klapuri et al. [348], [349]. A four-dimensional observation vector (as a function of time) was generated by applying a similar method to that of Scheirer [564] to generate resonator outputs but using different frequency bands and a different method for extracting the energy signal which also captures harmonic onsets. A measure of salience, dependent upon the normalized instantaneous energies of the comb-filter resonators, was also attached to this.

A problem with Scheirer's method was that it was prone to switch between different tempo hypotheses (usually doubling or halving), and Klapuri addressed this using an HMM to impose some smoothness to the tempo evolution. He proposed a joint density for the estimation of the period-lengths of the tatum, tactus, and measure level processes, applying a combination of sensible priors and dependencies learned from data. The phase of the tatum

and tactus pulse were estimated to maximize the observed salience at beats. In estimating the phase of the super-beat (measure) structure, a key assumption made was the expectation of two simple beat patterns which occur frequently in so-called 4/4 time. While this should considerably aid performance with music in this time signature, performance in the super-beat estimation was degraded for examples with a ternary metre (e.g. 3/4). Nevertheless, the algorithm was tested on a significant database and was successful. A comparison is presented below.

4.11 Comparison of Algorithms

If the focus is restricted to beat tracking in musical audio signals, then the methods discussed above in Sections 4.5 to 4.10 have various strengths and weaknesses. This section will highlight them and then present a comparison of several methods.

Rule-based approaches have never been applied to audio and have solely been used to code sensible but simple music theoretic rules in order to model music psychology expectations. The reason that they have never been used on audio signals is possibly because they are not easily expanded to cope with erroneous data and hence would perform poorly on the inexact data produced by onset detection algorithms.

Autocorrelative and histogram methods have much in common; they are both methods of obtaining a tempo profile, the difference being that autocorrelation works with a sampled signal while histogramming works with discrete onset times. They are therefore useful for finding the tempo but are not immediately applicable to extracting the beat phase (this is a secondary task).

Adaptive oscillators are particularly suited to causal operation and have some psychoacoustic justification [390]. However, they have not been applied to audio signals. This may be because the update routines required on adaptive single filters are not easily adaptable to real data or possibly because they are not well able to cope with sub-beats. Many of the systems also required manual initiation to set the correct tempo and phase. Comb-filters as implemented by Scheirer [564] and used by Klapuri et al. [349] have been applied to audio signals.

This leaves multiple agent approaches and probabilistic, model-based methods. These two bear some significant similarities, but the latter delimits the underlying assumptions from estimation procedures whereas they are intermixed by multiple agent methods. This makes the adaption and optimization of the probabilistic models easier, though reasonable success has been reported with both approaches.

4.11.1 Tests

There has been a move in recent years towards testing algorithms with a large database of audio samples collated from all genres and usually from standard,

commercially available sources. This was begun with Seppänen [572] with a database of 330 audio samples, while Klapuri [349] used 478. A comparison was also undertaken by Gouyon et al. [248] into tempo induction from audio signals using a large dataset of 3199 examples from three databases and is currently the most extensive.

The comparison below used a hand-labelled database of 222 samples of around one minute divided into six categories: rock/pop, dance, jazz, folk, classical, and choral. The tempos were limited to the range 60–200 bpm with the exception of the choral samples. Several examples exhibited significant rubato, 8 had a *rallantando* (slowing down), and 4 had a sudden tempo change. Forty-two also had varying amounts of swing added. Full details of the database can be found in [263].

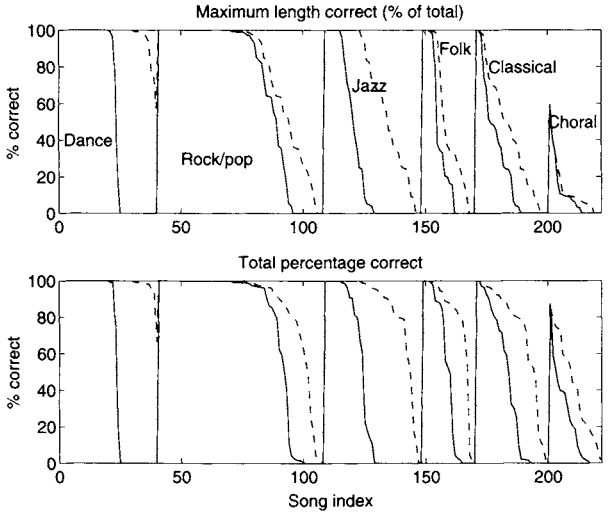
Another problem is how to evaluate the performance of a beat-tracking algorithm. As of this writing, no study has yet made a serious attempt to notate the complete rhythm and idealized score locations of every onset present in the audio sample;[8] rather the assessment has been limited to 'tapping in time' to the sample and producing an output of beat times that agrees with those of trained human musicians.

Klapuri [349] gives two criteria, which are adopted here, to judge the performance of an algorithm on a particular example. The first is 'continuous length' (C-L), by which it is meant the longest continually correctly tracked segment, expressed as a percentage of the whole. Thus, a single error in the middle of a piece gives a C-L result of 50%. Another, looser criterion is simply the total percentage of the whole which is correctly tracked (defined as 'TOT' from now on). Here, both are expressed as percentages of the manually detected beats which are correctly tracked, rather than of the time stretches these represent. Using Klapuri's definitions once again, a beat is determined to be correctly tracked if the phase is within $\pm$ 15% and the tempo period is correct to within $\pm$ 10%.

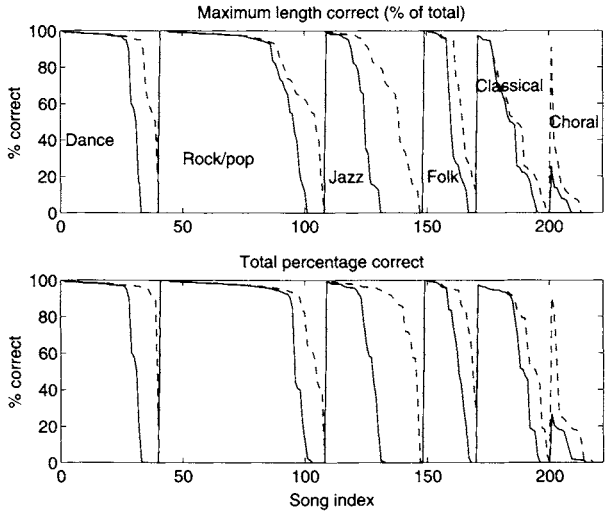
Here, the trackers[9] of Scheirer [564], Klapuri [349], and Hainsworth [263] are compared and the results are shown in Table 4.2. The columns under 'Raw' are base results according to the above criteria; however, it is sometimes found that the beat tracker tracks something which is not the predefined beat but is a plausible alternative. Usually, this is half the correct tempo (in the case of fast samples) or double (for particularly slow examples). When swing is encountered, it is occasionally possible for the trackers to even track at one and a half times the tempo (i.e., tracking three to every two correct beats). Doubling or halving of tempo is psychologically plausible and hence acceptable; however the errors encountered with swing are not. The second set of columns compares results once doubling and halving of tempo are allowed. Performance on individual genres is shown graphically in Fig. 4.9 for Hainsworth's and Klapuri's algorithms.

[8]The closest is probably Goto and Muraoka [234].

[9]The beat trackers tested were all the original authors' own.



(a) Hainsworth's results



(b) Klapuri's results

Fig. 4.9. Graphical display of the results for Hainsworth's (top) and Klapuri's beat tracker. The solid line is the raw result while the dashed line is the 'allowed' result. Note that ordering is strictly by performance for each genre under any particular criteria.

Table 4.2. Comparison of results on the database. The three beat trackers use audio adata as inputs.

	Raw		Allowed	
	C-L (%)	TOT (%)	C-L (%)	TOT (%)
Hainsworth	45.1	52.3	65.5	80.4
Scheirer	23.8	38.9	29.8	48.5
Klapuri	55.9	61.4	71.2	80.9

It can be seen that Klapuri's model performs the best in terms of raw results and continuous tracking, while the performance of Hainsworth when considering total number of beats with allowed tempo mistakes is about equivalent. Klapuri's method performs better than Hainsworth's with rock/pop and dance, though it fails somewhat with jazz. Hainsworth's outperforms Klapuri's on choral music, probably because of the onset detection algorithm used by Hainsworth (described above in Section 4.4), which gives superior performance for these choral samples.

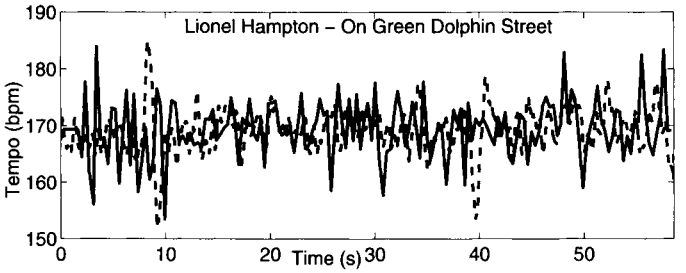
Both Klapuri's and Hainsworth's models significantly outperform Scheirer's. Klapuri [349] compared his model to Scheirer's and also Dixon's [148] modified MIDI beat-tracker. Seppänen [572] reported that his program was less successful than Scheirer's, tested on a large database that was a subset of Klapuri's. Also, on the related issue of tempo induction, the comparison by Gouyon et al. [243] showed that Klapuri's method performed the best at this task.

Finally, performance of one of the stochastic models which uses a signal representation is shown on a single example in Fig. 4.10. This shows Hainsworth's second stochastic model (described above in Section 4.10.2) with a swing example. The model is very successful at extracting onsets and is good at tempo tracking. The limitation is that the expected sub-beat structure has to be specified in advance. Thus, the model cannot be considered pan-genre.

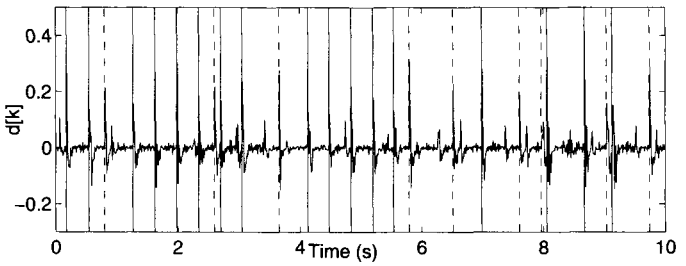
4.12 Conclusions

This chapter has discussed a number of differing approaches to the generic task of 'beat tracking'. Under this catch-all term, there are actually a number of possible goals, from replicating human tempo preference to a full labelling of every onset as to its correct quantized score location. Recent methods have aimed to extract the correct tempo and beat phase from audio signals ('tapping in time to the music').

Current methods such as Klapuri's [349] or Hainsworth's [263], [266] are, starting to achieve a reasonable level of success over databases of significant size and complexity. However, they are less successful on certain genres such as jazz (where part of the appeal of the style is its rhythmic complexity)



(a) Tempo profile.



(b) Onset detection process.

Fig. 4.10. Output of Hainsworth's second stochastic beat tracker (see Section 4.10.2) for a swing example. a) shows tracked tempo (dashed) and hand-labelled tempo (solid); b) shows the onset detection process for the first 10 seconds with solid vertical lines denoting detected beats and dashed vertical lines showing the detected swung quavers.

and classical music (which is prone to radical rhythmic evolution and also has fewer easily extractable beat cues). Classical music particularly seems to require pitch analysis in order to extract reliable beat cues. Thus, while the aim is obviously to have a generic beat tracker which works equally well with all genres, it is likely that in the short term, style-specific cues will have to be added. Klapuri [353] and Goto [221] both apply knowledge of typical drum patterns in popular music to their algorithms. Dixon [149] goes a step further and uses rhythmic energy patterns extracted from audio samples to aid classification of ballroom dance examples, a process which could easily be reversed to aid beat tracking.

In addition to better modelling specific styles and the rhythmic expectations therein, the second area for expansion is to look at better signal representations for extracting the cues needed to perform beat tracking. Rock and pop music with its drum-heavy style is easily processed using energy measures; classical music is much harder to process and only relatively recently have methods been applied to extract note changes where there is little transient energy. These will need to be improved.

In conclusion, the field of beat tracking or rhythmic analysis is one area of musical audio processing where some significant success has been achieved and there is much to build upon. However, there is also room for improvement and further accomplishments.