Beat Induction and Rhythm Recognition

Simon Dixon

Department of Computer Science, Flinders University of South Australia, GPO Box 2100, Adelaide SA 5001. `dixon@cs.flinders.edu.au`

Abstract. Most people have no difficulty in picking out the beat in a piece of music, and even if they cannot define what it is they have detected, they can tap their feet in time with the music. This ability is called *beat induction*. A more difficult task, which we call *rhythm recognition*, involves uncovering the hierarchical structure of the timing relationships in the music, at a higher level than the frequencies of vibration of individual notes and chords, but at a lower level than the musical form and phrase structure. In this paper, we present a bottom-up approach to the recognition and transcription of musical rhythms from acoustic signals. We assume no a priori high-level knowledge of the music such as the time signature or tempo, but attempt to derive this information from the timing patterns of detected note onsets. The problem of rhythm recognition is divided into three stages: firstly, finding the main beat or rhythmic pulse of the music; secondly, discovering the rhythmic structure, a hierarchical arrangement of units consisting of multiples and submultiples of beats; and thirdly, mapping the note onsets to musical timings and durations. We compare our methods with other work presented in the literature and with commercial products which purport to solve the same problems. This work forms part of an automatic music transcription system currently under development by the author.

1 Introduction: Human Rhythm Perception

Most people have no difficulty in picking out the beat[1] in a piece of music. Although we may be unable to give a precise definition of what it is that we have detected, we can clap or dance in time with this beat. In fact, when listening to music we often subconsciously tap a foot in time with the beat. This ability is called *beat induction*. On the other hand, understanding the rhythm of a piece of music is a more difficult task, which involves uncovering the hierarchical structure of the timing relationships in the music. These relationships exist at a higher level than the frequencies of vibration of individual notes and chords, but at a lower level than the musical form and phrase structure [10]. The process of extracting these relationships is called *rhythm recognition*.

Although we do not attempt to model or describe the cognitive mechanisms involved in human rhythm perception, we do note certain features which motivate an ambitious unsupervised approach to the problem. Firstly, human rhythm perception sets its own parameters; the tempo and the metrical structure are not specified explicitly at the beginning of a piece, and if they change suddenly

[1] Or, more precisely, the *tactus*.

during the piece, the perceptual system is able to adjust itself within seconds to the new listening framework. Secondly, it copes well with "noise" in the input, that is, deviations from precise timing are allowed, as are variations in speed, without disturbing the overall perception of the music. Thirdly, it is able to cope with a degree of syncopation, that is, sections of music where no audible event occurs at the time of the beat, but where events still occur at other times between the beats. Rather than ignoring such parts as erroneous, they are perceived as rhythmically interesting sections of the music.

In contrast with these capabilities, computer music software does not cope well with any of these situations. Commercial sequencing and transcription programs require the beat to be declared explicitly before the music is processed, so that all data can then be indexed relative to this given beat. Even recent research systems are limited by the fact that once they get out of synchronisation with the music, it is very difficult for them to recover and resume correct interpretation of the rhythmic structure [3]. The robustness of human perception is one feature which is extremely difficult to reproduce in a computer system.

In this paper, we present a bottom-up approach to the recognition and transcription of musical rhythms from acoustic signals. We assume no a priori high-level knowledge of the music such as the time signature or tempo, but attempt to derive this information from the timing patterns of detected note onsets. Our approach breaks the problem of rhythm recognition into three subproblems: finding the main beat, or rhythmic pulse, of the music; discovering the structure of the rhythm, that is, units containing multiples and submultiples of beats; and mapping the note onsets to musical timings and durations. We present a solution to the first two stages, and discuss how we intend to tackle the final stage of processing.

We conclude this section with a brief outline of this paper: section 2 describes our approach to onset detection using multiple resolution Fourier analysis; the following section contains our beat induction algorithm which maps inter-onset intervals into equivalence classes, keeping track of the relative frequency of each class; then, in section 4, we describe the structural information that can be drawn from the onsets times and the induced beat to solve the first two stages of the rhythm recognition problem; section 5 discusses quantisation, the mapping of absolute time to relative time (in musical notation); and we conclude the paper with a review of related work followed by a discussion of achievements and future research directions.

2 Onset Detection

The input to the rhythm recognition system is a digitally sampled acoustic signal, stored in a single channel 16 bit linear pulse code modulated (PCM) format. Various sampling rates (8-48kHz) may be used. In the experiments described in this paper, solo jazz guitar CDs were used for the acoustic input, sampled at 8kHz. The loss of frequency range caused by using a low sampling rate did not affect the results of this work, but provided significant savings in memory and processing time.

The first stage in rhythm perception is the detection of the beginnings of notes, which are called *onsets*. (Similarly, the end of a note is called its *offset*.) Note onsets define the rhythm of a piece of music, whereas the offsets have almost no bearing on the perceived rhythm, except to alter the smoothness or connectedness of the notes. On percussive instruments (e.g. piano, guitar), it is not even possible to specify when a note offset occurs, as notes decay exponentially, so notes are often already inaudible before they are physically released. Also, it is rare for offsets to be performed as notated in music; in fact, many works for wind instruments or voice would not give the performers sufficient time to breathe if the offsets were played strictly as written!

We use a sliding window short-time Fourier transform to create a frequency-time representation of the music, and search each frequency band for sudden increases in amplitude which may correspond to note onsets. By using a very small window length for the transform, we are able to achieve a high degree of time resolution, at the expense of a corresponding loss in frequency resolution. For this paper we used a window size of 16ms (128 samples), which gives a sufficiently high time resolution (and certainly finer than human perception), but gives a frequency resolution of 62.5Hz, more than an order of magnitude too coarse for accurate note identification across all frequencies of interest. Since we are interested in music transcription, it is important that our approach to rhythm recognition does not preclude correct note identification. To obtain a high enough frequency resolution to distinguish notes to the nearest semitone, we perform another Fourier transform on the acoustic data, using a much larger time window (usually 100-200ms). The loss in time resolution by using the larger window does not damage our results, as we already have an accurate estimate of note onset times from the small window transform. A more detailed discussion of our approach to note identification can be found in [7].

To guard against false detections of onsets, we correlate possible onsets across frequency bands, noting that most musical notes begin with a transient of indeterminate pitch, which produces a peak across a large range of frequencies at the beginning of the note (see Figure 1). On the contrary, we do not get a high degree of correlation between the frequencies at which the onsets are detected and the frequencies of the notes to which the onsets belong, but the information from the transients is useful in discriminating true onsets from spurious peaks in the data. Another method that is used to filter out spurious peaks is by insisting that each note has a sufficiently long duration that its frequency can be determined by the auditory system. This condition is checked when the longer window Fourier transform is calculated. It is probable that this condition will have to be relaxed if the system is required to process music containing drum sounds.

The final stage that remains is to correlate the two transforms of the data, which is done by analysing each high resolution frequency band around the time of each detected onset. For peak detection, we use a search for local maxima

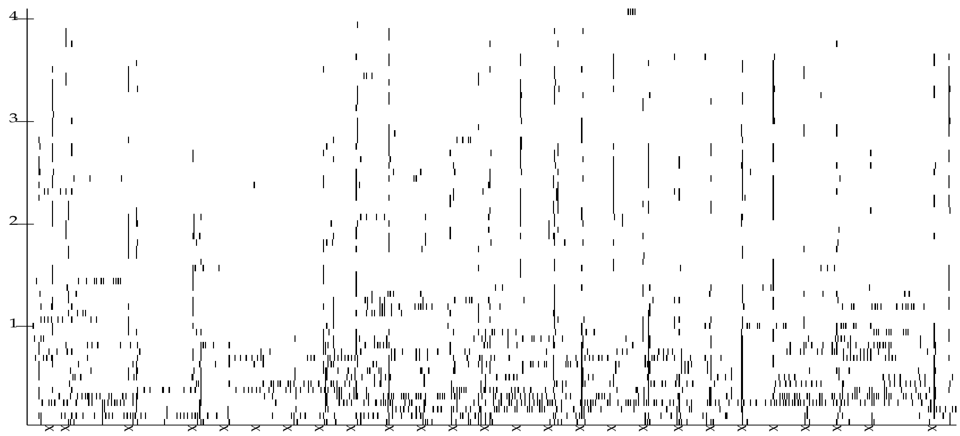

Fig. 1. Amplitude peaks in short window spectrogram; vertical axis is frequency in kHz; horizontal axis is time with note onsets marked by (x).

within a small band of frequencies, (usually one or two bands either side), and also require that the amplitude is above a minimum threshold level. The resulting peaks are used as a second check for false detections, and any onset which doesn't have a corresponding note to attach to is deleted at this stage.

It is difficult to evaluate the results for onset detection separately, as it is intricately related to our note identification algorithm. We note that in a small test (24 notes) where the data was analysed manually, there were two false positives detected, both of which corresponded to notes ending abruptly, causing a spike in the frequency-time distribution. These onsets are easily eliminated when it is found that there is no corresponding frequency components which begin or increase suddenly around these times. There were also three missed notes, all of which had small peaks but were below threshold. These are harder to correct, because they are surrounded by notes of much higher amplitude, so that the note identification stage will also have the same difficulties in detecting the onset.

The onset detection method may miss detecting valid onsets in cases where the initial transient is not as pronounced, such as with electronically produced sounds, or notes which are produced via a gradual change from another note. To date, this possible source of problems has not caused any missed detections, but we flag it for further investigation.

We note that by analysing the data at two independent levels of resolution, we are performing a similar type of analysis as occurs in the ear, which has mechanisms for detecting sound at both high and low resolutions of frequency (the inner and outer hair cells on the Organ of Corti). It is assumed that the brain correlates these data to produce a clearer overall picture of the properties of the source sound. The approach of performing spectral analysis at multiple resolutions is also a standard technique in engineering.

3 Beat Induction

Once the onsets have been detected and spurious peaks removed, we analyse the time durations between near pairs of notes. Using results from psychoacoustic research, we are able to use limits on the accuracy of production and perception of timing information in music to set parameters for this analysis. It is known that in musical performances, deviations of up to 30-40ms from the timing indicated in the score are not uncommon, and often go unnoticed by listeners [15]. This allows us to construct classes of inter-onset intervals (IOI's) which are sufficiently similar to be perceived as the same duration. These classes are characterised by the average duration of their members, and new members are added if their durations are close enough to this average. Closeness is defined in absolute terms by the constant *MaxError* (see algorithm below). If an IOI does not fit into any existing class, a new class is created.

Algorithm: Generate_Classes

For each pair of onset times t_1, t_2 (with $t_1 < t_2$)
    If $t_2 - t_1 < D$ (maximum distance between intervals)
        Let $I = t_2 - t_1$
        Find class C_n such that $|Average(C_n) - I|$ is minimum
        If $|Average(C_n) - I| < MaxError$ then
            $C_n := C_n \cup \{I\}$
        Else
            Create new class $C_m := \{I\}$
        End If
    End If
End For

Note that the process of adding an IOI to a class automatically adjusts the average of the members, so that the class boundaries are not rigid, but may drift over a period of time. For this reason it is important that these classes are not constructed over too long a time interval, or else variations in the tempo will corrupt the accuracy of the results. That is, we are looking to develop a local model of the tempo, and use that to determine the local structure. Having done this, the next local area may be calculated, with the added constraint of continuity, that is, the local tempo will not normally change between areas.

For each equivalence class, we calculate a score based on the number of intervals in the class and the agreement of the durations of the intervals. This gives us a ranking of classes, most of which are integer multiples or submultiples of the beat. Each score is adjusted to reflect the scores of other intervals which are related in this way, and a final best estimate of the period of the beat is determined.

The algorithm described above was tested on a collection of samples of solo jazz guitar music (acoustic and electric). These were chosen for their difficultly for beat induction, because of the high level of syncopation and because the performers take liberties with the rhythm. In order to deal with changing tempos, it is essential that the beat induction algorithm works on small amounts of data, so we tested the algorithm with samples of only 5 seconds to test whether the system can find the local tempo at any point in a piece of music. Our results for the first 24 5-second segments of one piece of music are presented below, with a comparison to an autocorrelation function (see section 6). Each row represents a time interval expressed as a fraction or multiple of the correct beat. The table entries represent the number of samples which were assigned this time interval as their beat.

Interval (beats)	Autocorrelation	Our algorithm
$\frac{1}{3}$	1	1
$\frac{1}{2}$	1	0
$\frac{2}{3}$	2	0
1	12	22
$\frac{4}{3}$	1	1
2	3	0
other	4	0

In the two cases where our algorithm failed to choose the correct beat, it chose the correct beat as the second and third most likely choices respectively. Using the assumption of continuity of tempo between adjacent time frames, these errors would be corrected easily.

4 Deriving Structural Information

The ranked estimates discussed in the previous section give us more than just an estimate of the period of the beat. They also provide us with structural information in the form of the multiples and submultiples of beats which occur commonly in the music. The system recognises that pieces in simple time have a highly ranked submultiple at half the beat, and pieces in compound time have submultiples at one third and two thirds of the beat. This information has not yet been integrated into the higher level rhythm recognition process.

We also expect that when examining longer samples of music, highly ranked multiples should occur at multiples of the beat corresponding to the length of the bar. Using this information, the hierarchical structure of the rhythm can be extracted without necessarily modelling expected rhythmic patterns. The next stage of the system will incorporate a hierarchy of rhythmic decomposition patterns, sorted by complexity, so that the system is steered towards the more plausible interpretations.

Another method of obtaining higher level structure is by autocorrelation [2], but this relies on precise timing relationships between notes which are not present in our work until after the quantisation stage. Brown's work does not examine performance data, but calculates the autocorrelation function on the IOI's as written in the score. The results in section 3 show that autocorrelation is not suitable for extracting the rhythmic structure of performance data.

5 Quantisation

Once the beat and metre have been established, the next step is to express each of the durations in terms of these parameters. The simplest way to do this is to choose a limit of resolution (a time quantum) and round each onset time to the nearest whole number of quanta. This approach produces musical nonsense, as it fails to distinguish between likely and unlikely rhythmic patterns (Figure 2). (Most commercial music software uses this technique.)

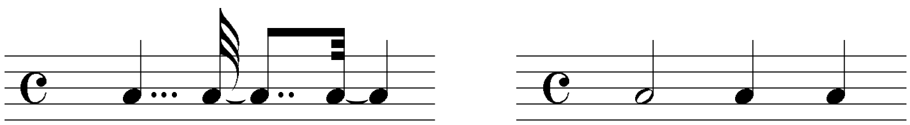

Fig. 2. Left: an implausible rhythm; Right: a similar but more plausible rhythm

The issue here is not one of accuracy — the interpretation on the left of Figure 2 may be a more accurate rendition of a performance than the rhythm on the right. But due to the difficulty for human performers to reproduce such rhythms accurately, it is a matter of musical convention that such rhythms are not notated. Also, the left hand rhythm is likely to have been produced by a performer playing the right hand rhythm and either inadvertently modifying the tempo, or anticipating the beat deliberately for expressive effect.

To solve this problem, we must create a model of the type of rhythmic patterns we expect. One approach is to order the patterns by the degree of syncopation, and then use the quantisation process to minimise the syncopation whilst also minimising the amount that times are adjusted by the quantisation process. This approach is similar to that of Longuet-Higgins and Lee [12], and assuming that the measure of syncopation corresponds to the complexity of expressing the timing information, it also relates to the approach of Tanguiane [16].

The ideal approach must allow the user to specify at some level how the rhythm should be interpreted by the system, that is to specify the user's preference on the tradeoff between accuracy and simplicity. In this way, the system could be used by composers seeking a faster way of producing musical scores, who are more likely to prefer the simpler representation, but also by musicologists studying performance data, who are more likely to want a precise representation of expressive timing variations. Pressing [13] describes an interactive system where the user specifies a menu of allowed syncopations, which provides a precise method of specifying what is meant by the concept of "minimal" syncopation.

6 Related Work

A substantial amount of research has been performed in the area of rhythm recognition by computer, culminating in a display at the 1994 International Computer Music Conference in which a variety of methods were demonstrated by attaching the computer to a shoe which tapped in time with the calculated beat of the music [6]. Many of these methods are difficult to compare, as they rely on different forms of input, or make different assumptions about the nature of the music being analysed, or are limited by a subset of a complete rhythmic language, or rely on a high level of supervision (interaction with the user).

Much of the work in machine perception of rhythm has used MIDI files as input [14, 4, 9], which contain control information but no direct acoustic information. A MIDI file consists of a chronologically ordered sequence of events, such as keys being pressed or released, and timing information in the form of time durations between each pair of events. (Type 1 MIDI files also allow for the encoding of structural information such as the time signature and tempo, but most research in this area presumes that this information is not available to the rhythm recognition program.)

Work using MIDI files views the input as a list of inter-onset intervals (IOI's), and does not attach any rhythmic significance to differences in pitch or volume of notes. That is, each note is treated purely as an uninterpreted event. This ignores potentially useful information which could be gained by separating parts using principles of auditory streaming [1]. This form of input also sidesteps the first difficult issue in rhythm recognition, which is the detection of onsets of notes. Nevertheless, the ideas are still useful, as they can be applied to subsequent stages of analysis.

Notable work using MIDI file input is the emulation of human rhythm perception of [14] which produces multiple hypotheses of possible hierarchical structures in the timing, assigning a score to each hypothesis, corresponding to the likelihood that a human listener would choose that interpretation of the rhythm. This technique gives the system the ability to adjust to changes in tempo and metre, as well as avoiding many of the implausible rhythmic interpretations produced by commercial systems.

A similar approach is advocated by Tanguiane [16], who uses Kolmogorov complexity as the measure of the likelihood of a particular interpretation, with the least complex interpretations being favoured. He presents an information-theoretic account of human perception, and argues that many of the "rules" of music composition and perception can be explained in information-theoretic terms.

Desain [4] compares two different approaches to modelling rhythm perception, the symbolic approach of Longuet-Higgins [11] and the connectionist approach of Desain and Honing [5]. Although this work only models one aspect of rhythm perception, the issue of quantisation, and the results of the comparison do not provide a definitive preference for one style over the other, it does highlight the need to model expectancy, either explicitly or implicitly. Expectancy, as described in the work cited above, is a type of predictive modelling which is particularly relevant to real-time processing as it provides a contextual framework in which subsequent rhythmic patterns can be interpreted with less ambiguity.

An alternative approach uses a nonlinear oscillator to model the expectation created by detecting a regular pulse in the music [9]. A feedback loop controls the frequency of the oscillator so that it can track variations in the rhythm. This system performs quite robustly, but due to its intricate mathematics it does not correspond to any intuitive notion of perception, and in this sense is very similar to connectionist approaches.

One of the few attempts to analyse rhythm in acoustic signals is the Beat Tracking System (BTS) of Goto and Muraoka [8]. This system is designed to follow the beat of pop music in real time (one of the shoe-tapping programs mentioned above). BTS uses frequency histograms to find significant peaks in the low frequency regions, which are presumed to correspond to the bass and snare drums, and then tracks these frequency regions by matching timing patterns in these bands to a small set of pre-stored drum beat patterns. It was found that this method was successful for 42 of the 44 popular songs on which it was tested. Clearly this method could not be used in other styles of music where drums do not keep the beat, or where the rhythmic patterns are less predictable.

Commercial transcription and sequencing programs do not even address the issues covered in these research systems. The tempo and time signature must be explicitly entered before the music is played, and the system then aligns each note with the nearest position on a metrical grid. This grid is computed from the given beat and a resolution limit which is defined to be the shortest allowed note length. Thus these systems often produce implausible rhythmic interpretations, and cannot be used in an unsupervised manner for anything but the simplest rhythms.

7 Discussion and Future Work

It is clear that rhythm identification is not a trivial task, despite the apparent ease with which many humans are able to perform the task. We have described a rhythm recognition process which analyses acoustic data, detecting a sequence of note onsets, and then discovers patterns in the intervals between the onsets, from which the most likely beat is induced.

The rhythm detection system has not yet been completed, but the early stages show promising results. The onset detection and beat induction algorithms have been tested on a number of examples of jazz guitar music, in which the rhythm is quite complex, and have produced correct results for the induced beat. It is expected that similar results shall be obtainable with other instruments and mixed instrument music, as the system does not rely on any features which are peculiar to the guitar, nor to stringed instruments or plucked instruments in general.

The next stage of using structural information for quantisation is currently being addressed, and we are examining some of the techniques discussed in the previous section. A comparison of our approach with connectionist techniques has also begun. The precise direction of future work depends somewhat on the results of these investigations and implementations, with the end goal being an automatic music transcription system.

Other research currently underway includes dynamic modelling of parameters of musical instruments and voices using multidimensional feature spaces, as well as psychoacoustic results from auditory scene analysis, with the aim of separating music into parts in a similar way that a human listener is able to attend to the various instruments playing in an ensemble.

References

1. A.S. Bregman. *Auditory Scene Analysis: The Perceptual Organisation of Sound.* Bradford, MIT Press, 1990.
2. J.C. Brown. Determination of the meter of musical scores by autocorrelation. *Journal of the Acoustical Society of America*, 94(4):1953–1957, 1993.
3. R.B. Dannenberg. Recent work in real-time music understanding by computer. *Proceedings of the International Symposium on Music, Language, Speech and Brain*, 1991.
4. P. Desain. A connectionist and a traditional AI quantizer: Symbolic versus sub-symbolic models of rhythm perception. *Contemporary Music Review*, 9:239–254, 1993.
5. P. Desain and H. Honing. Quantization of musical time: A connectionist approach. *Computer Music Journal*, 13(3), 1989.
6. P. Desain and H. Honing. Foot-tapping: a brief introduction to beat induction. In *Proceedings of the International Computer Music Conference*, pages 78–79. Computer Music Association, San Francisco CA, 1994.
7. S.E. Dixon. Multiphonic note identification. *Australian Computer Science Communications*, 18(1):318–323, 1996.
8. M. Goto and Y. Muraoka. A real-time beat tracking system for audio signals. In *Proceedings of the International Computer Music Conference*. Computer Music Association, San Francisco CA, 1995.
9. E.W. Large. Beat tracking with a nonlinear oscillator. In *Proceedings of the IJCAI Workshop on Artificial Intelligence and Music*, 1995.
10. F. Lerdahl and R. Jackendoff. *A Generative Theory of Tonal Music.* MIT Press, 1983.
11. H.C. Longuet-Higgins. *Mental Processes.* MIT Press, 1987.
12. H.C. Longuet-Higgins and C. Lee. The rhythmic interpretation of monophonic music. *Music Perception*, 1:424–441, 1994.
13. J. Pressing. Personal communication. 1996.
14. D. Rosenthal. Emulation of human rhythm perception. *Computer Music Journal*, 16(1):64–76, 1992.
15. J. Sundberg. *The Science of Musical Sounds.* Academic Press, 1991.
16. A.S. Tanguiane. *Artificial Perception and Music Recognition.* Springer-Verlag, 1993.