A Lightweight Multi-agent Musical Beat Tracking System

Simon Dixon

Austrian Research Institute for Artificial Intelligence,
Schottengasse 3, A-1010 Vienna, Austria.
simon@ai.univie.ac.at

Abstract. Beat tracking is what people do when they tap their feet in time to music. We present a software system which performs this task, processing music in a standard digital audio format and estimating the locations of musical beats. A time-domain algorithm detects salient acoustic events, and then a clustering algorithm groups the time intervals between events to obtain hypotheses about the current tempo. Multiple competing agents track these hypotheses throughout the music, with further agents being created at decision points. The output for each agent is a sequence of beat locations, which is evaluated for its closeness of fit to the data. This approach to beat tracking assumes no previous knowledge of the music such as the style, time signature or approximate tempo; all required information is derived from the audio data. The system has been tested with various styles of music (popular, jazz, and classical) and performs robustly, rarely making errors in popular music, and recovering quickly from errors in more complex styles of music, despite the fact that no high level musical knowledge is encoded in the system.

1 Introduction

Although most people can tap their feet in time with music, equivalent performance on a computer has proved remarkably difficult to achieve. One reason for this is that these systems have been based only on the codification of high level metrical knowledge. We show that such knowledge is secondary to the beat tracking task. Just as people can follow the beat of music without musical training and without previous knowledge of the particular piece of music, so can a computer program.

In this paper, we present a system which processes musical audio signals, estimating the tempo and determining the locations of musical beats. No specific assumptions are made about the music being analyzed, but the system performs robustly on various types of popular, jazz and classical music. We do not attempt to model or describe the cognitive mechanisms involved in human rhythm perception. However we do note certain features of human perception which motivate an ambitious unsupervised approach to the beat tracking problem, namely, that human rhythm perception is self-calibrating and copes well with both syncopation and noise in the input.

The subsequent sections of the paper contain a review of related work, then a description of the musical assumptions made in this work. The next sections present the algorithm for onset detection from raw audio data, followed by the algorithm for *tempo induction*, defined as the estimation of the time interval between successive occurrences of the main rhythmic pulse of the music. We then describe the multi-agent approach to *beat tracking*, which is the determination of beat locations (and therefore tempo fluctuations) in the light of the previous tempo estimations. The results from testing the system with various types of music are then presented and discussed, and the paper concludes with a description of applications of the beat tracking system.

2 Related Work

A substantial amount of research has been performed in the area of rhythm recognition by computer, including a demonstration of various beat tracking methods using a computer to control a shoe which tapped in time with the calculated beat of the music [5]. These systems are difficult to compare directly, as they make different assumptions about the input format, style and complexity of the music.

Much of the work in machine perception of rhythm has used MIDI files as input [18, 2, 14], which contain control information for a synthesizer rather than audio data. MIDI files consist of sequences of events, usually corresponding to pressing and releasing keys on a piano-style keyboard, plus an encoding of the time duration between successive events. Structural information such as the time signature and tempo can also be stored in MIDI files, but it is usually assumed that such information is not available to rhythm recognition programs.

Using MIDI files, the input is usually interpreted as a series of event times, ignoring the event duration, pitch, amplitude and chosen synthesizer voice. That is, each note is treated purely as an uninterpreted event. It is assumed that the other parameters do not provide *essential* rhythmic information, which in many circumstances is true. However, there is no doubt that these factors provide useful rhythmic cues; for example, more salient events tend to occur on stronger beats.

Notable work using MIDI file input is an emulation of human rhythm perception [18], 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 meter, as well as avoiding many of the implausible rhythmic interpretations produced by commercial systems.

Desain [2] compares two different approaches to modeling rhythm perception, the symbolic approach of Longuet-Higgins [16] and the connectionist approach of Desain and Honing [3]. Although this work only models one aspect of rhythm perception, the issue of quantization, and the results of the comparison are inconclusive, it does highlight the need to model expectancy, either explicitly or implicitly. Expectancy is a type of predictive modeling, particularly relevant to real time processing, as it provides a contextual framework in which subsequent rhythmic patterns can be interpreted with less ambiguity.

Allen and Dannenburg [1] describe a beat tracking system that uses beam search to consider multiple hypotheses of beat timing and placement. A heuristic evaluation function directs the search, preferring interpretations that have a "simple" musical structure and make "musical sense", although they do not define what they mean by these terms. They also do not describe the input format or any specific results.

An alternative approach uses a nonlinear oscillator to model the expectation created by detecting a regular pulse in the music [14]. A feedback loop controls the frequency of the oscillator so that it can track variations in the rhythm. This system performs quite robustly, but, like connectionist approaches, does not provide any intuition about the beat tracking process, and therefore would be difficult to extend using high level knowledge.

There has been only a small amount of rhythm research using audio input, beginning with the percussion transcription system of Schloss [19]. Onsets were detected as peaks in the slope of the amplitude envelope, where the envelope was defined to be equal to the maximum amplitude in each period of the high-pass filtered signal, and the period defined as the inverse of the lowest frequency expected to be present in the signal. The system was limited in that it required parameters to be set interactively, and it was evaluated only by resynthesis of the signal.

A more complete approach to beat tracking of acoustic signals was developed by Goto and Muraoka [10, 12, 13]. They developed two systems for following the beat of popular music in real time. The earlier system (BTS) used frequency histograms to find significant peaks in the low frequency regions, corresponding to the frequencies of the bass and snare drums, and then tracked these low frequency signals by matching patterns of onset times to a set of pre-stored drum beat patterns. This method was successful in tracking the beat of most of the popular songs on which it was tested. A later system allowed music without drums to be tracked by recognizing chord changes, assuming that significant harmonic changes occur at strong rhythmic positions. These systems required a powerful parallel computer in order to run in real time.

Commercial transcription and sequencing programs do not address the issues covered by these research systems. They generally require that the tempo and time signature are specified before the music is played, and the system then aligns each note with the nearest position on a metrical grid. Recent systems allow parameterization of this grid in terms of its resolution (the shortest allowed note length), and adjustment of restrictions on the complexity of rhythm that can be produced by the system. These systems often produce implausible rhythmic interpretations, and cannot be used in an unsupervised manner for anything but simple rhythms.

3 Musical Assumptions

Despite the large amount of research in time and rhythm in music, the beat tracking problem remains poorly defined. The reason is that the beat is a subjective property of performed music. Formal musical models tend to be based on the notational representation rather than performance [15, 17], and those which address performance timing do so from the point of view of generation and/or transformation of timing rather than extraction or explanation of performance data [4]. We follow [11] in evaluating the correctness of the beat tracking system relative to a subjective labelling of beat positions.

A theoretical definition of *beat* is "a perceived pulse marking off equal durational units" [9]; in practice, the durational units marked off by the onsets of notes on successive beats are only approximately equal. However, for a large amount of music, the subjective differences in perceived beat are minor, otherwise human activities such as ensemble playing and dancing would not be possible. In this study we restrict our attention to music which has such an agreed beat, and rely on the onset detection algorithm to choose the more salient events as possible beat locations.

4 Audio Processing

In this and the following sections, we describe the stages of processing performed by the beat tracking system. All of the software is written in C++ and runs on a Unix platform (Linux or Solaris). The complete processing of a song takes about 10 seconds on a current PC, making it viable for use in real time audio applications, although the software is not currently designed for real time use. The input to the system is a digitally sampled acoustic signal, such as is found on audio compact discs. The stereo compact disc data is converted to a single channel format by averaging the left and right channels, resulting in a single channel 16 bit linear pulse code modulated (PCM) format, with a sampling rate of 44.1kHz.

The aim of the initial signal processing stage is to detect *events* in the audio data, from which rhythmic information can be derived. For the purposes of this work, events correspond to note onsets, that is, the beginnings of musical notes, including percussive sounds. By ignoring note durations and offset times, we discard valuable information, but our results justify the present assumption, that there is sufficient information in note onsets to perform beat tracking.

A time-domain method similar to that of Schloss [19] is employed for onset detection. This method involves passing the signal through a simple high-pass filter, calculating the absolute sum of small overlapping windows of the signal, and then finding peaks in the slope of these window sums using a 4 point linear regression. Only the more salient event onsets are detected with the method, which is ideal for the subsequent task of tempo induction.

5 Tempo Induction

The tempo induction section of the system determines a set of hypotheses about the tempo of a given section of music, which may be expressed in beats per minute (BPM) or in seconds, as the *inter-beat interval* (IBI). The algorithm, described further in [6,7], is based on clustering of *inter-onset intervals* (IOI's). In the literature, an IOI is defined as the time between the onsets of two successive events, but we extend the definition to include times between onsets of pairs of events that are separated by intervening event onsets. All possible pairs of onsets that occur within 2.5 seconds of each other are grouped by the clustering algorithm. Figure 1 shows clustering for five events (A, B, C, D, E) into intervals of similar size. For example, cluster C1 contains the intervals AB, BC and DE, while cluster C2 contains AC and CD. Each cluster is identified by its average interval size.

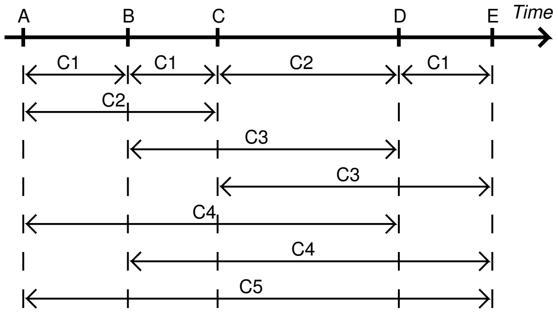

Fig. 1. Clustering of inter-onset intervals

After grouping IOI's into clusters, a score is calculated for each cluster, based on the number of IOI's in the cluster. The highly ranked clusters usually correspond to the beat or small integer multiples or fractions of the beat. For example, supposing that C2 represents the IBI, then C1 represents half of the IBI and C4 represents double the IBI. Each cluster's score is increased for each other cluster to which it is related by a small integer ratio, and a final ranking of the inter-beat interval hypotheses is determined.

In previous work [7], it was found that the correct tempo can be induced from a 5-10 second excerpt of the music with 90% reliability, and by using multiple (or longer) excerpts, the reliability quickly approaches 100%. In this work, it does not matter if the initial estimate is correct, as multiple hypotheses are checked in the beat tracking stage, so that an error in tempo induction can be corrected at a later time.

6 Beat Tracking Agents

The tempo induction algorithm computes the inter-beat interval, that is, the time between successive beats, but does not calculate the location of the beat. In Figure 1, the clustering might determine that C2 represents the inter-beat interval, but it does not reveal whether events A, C and D are beat locations or whether B and E are beat locations. By analogy with wave theory, we could say that it calculates the *frequency* but not the *phase* of the beat.

The phase is calculated by employing an agent-based architecture to examine multiple hypotheses simultaneously throughout the music. The agents are characterized by their *state* and *history*. The state is the agent's current hypothesis of the beat frequency and phase, and the history is the sequence of beat locations selected so far by the agent. Each agent is evaluated on the basis of its history, with higher scores being awarded for greater regularity in the spacing between events, greater salience of chosen events, and fewer gaps in the sequence.

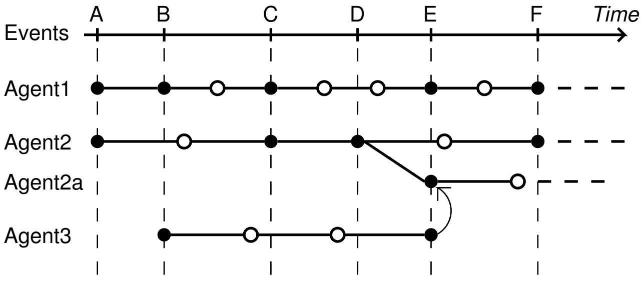

Fig. 2. Beat tracking agents (see text for details)

Initially, a number of agents are created for each of the tempo hypotheses from the tempo induction stage; for each tempo, one agent is created for each of the first few events in the piece, with its phase set to 0 at the time of the event onset. A simplified example is shown in Figure 2, where there are 2 tempo hypotheses, and two starting locations, events A and B. Agents 1 and 2 start with the same phase, but different tempo, while Agent 3 starts with a different phase, but the same initial tempo as Agent 2. Note there is no need to start an agent with the tempo of Agent 1 and phase of Agent 3, since Agent 1 covers event B itself.

The main loop of the beat tracking section passes each event to each agent, which compares the event's onset time with the predicted beat location. The agents have two windows of tolerance, an inner window, within which the agent is sure that the event corresponds to the predicted beat location, and an outer window, within which the agent is unsure if the event should be accepted as a beat location. If the event falls in the inner window, it is added to the agent's history, and the agent's tempo and phase hypotheses are updated. In the case that the event falls in the outer window around the predicted location, the agent creates a clone which accepts the event as a beat location, while the current agent rejects the event. In Figure 2, Agent 2 creates Agent 2a when event E falls in its outer window of expected beat locations. This guards against the current beat location being lost due to a rogue event, whilst also allowing for moderate deviations in tempo and phase to occur. When an accepted event is more than one beat from the previous beat location determined by the agent, the missing beats are filled in by interpolation (shown by hollow circles in the figure).

As the agents track the beats, a confidence value is maintained for each agent. This value is increased each time an event is accepted as a beat location. The amount of increase depends on the salience of the event and its proximity to the predicted beat location. The salience is measured in terms of the amplitude of the event onset, calculated on a logarithmic scale. The value is then reduced according to the difference between the predicted and actual beat locations, and then added to the agent's confidence value. The final confidence value for an agent is calculated by reducing the confidence for each beat which had to be interpolated, and normalizing the result so that the agents with a faster tempo are not advantaged by their greater number of beats.

It often occurs that two or more agents come to the same conclusion about the current tempo and beat phase. Since these are the only variables that determine the agent's state (and therefore its future behavior), it is computationally advantageous to remove all but one of these agreeing agents, retaining only the agent with the highest confidence (that is, with the best scoring history). In Figure 2, Agent 3 is terminated when its state coincides with that of Agent 2a, as indicated by the arrow at event E. Removal of duplicate agents is performed after each event is processed. Agents are also checked for currentness, and are removed if they are unable to find any event corresponding to a predicted beat for some fixed length of time.

After the last event is processed, the highest scoring current agent is selected, and its history is output as the beat tracking "solution". It is also possible to view a trace of the agents and their scores at each event during processing. For aural testing (and demonstrations) of the system, the music can also be played back or saved to file with a click track added to it, that is, a percussion track indicating the positions of beats as detected by the system. In the following section, testing methodology is discussed, and the results of beat tracking with various styles of music are presented.

7 Results and Discussion

Informal testing was performed by listening to the music with synthesized percussion strokes (e.g. cow-bell) played at the beat locations computed by the system. With this method, it is easy to check that the tempo estimation and tracking are approximately correct, but it is not a very precise form of testing. It is also very time-consuming if used repeatedly to test the effects of small adjustments to the system. However, aural testing provides intuition about the situations in which the beat tracker fails, which is useful for determining which aspects of the system require further work.

Fig. 3. Beat tracking test details and results

Song Title	Artist	Style	Date	Tempo range	Time sign.	Tracking Results
I Don't Remember A Thing	Paul Kelly and the Coloured Girls	Pop/rock	1987	139-142	4/4	100%
Dumb Things	” ”	Pop/rock	1987	151-154	4/4	100%
Untouchable	” ”	Pop/rock	1987	145-146	4/4	100%
Superstition	Stevie Wonder	Motown	1972	96-104	4/4	96%
You Are The Sunshine of My Life	Stevie Wonder	Motown	1972	127-136	4/4	92%
On A Night Like This	Bob Dylan	Country	1974	136-140	4/4	79%
Rosa Moreña	João Gilberto Trio	Bossa nova	1964	128-134	4/4	95%
Michelle	Béla Fleck and the Flecktones	Jazz swing	1991	180-193	3/4	92%
Jitterbug Waltz	James Morrison	Jazz waltz	1990	155-175	3/4	77%
Piano Sonata in C 3rd movt. 1st sect.	Wolfgang Mozart	Classical	1775	120-150	2/4	90%

In Figure 3, we present the results for beat tracking of songs representing various styles of music. The rightmost column of the table indicates the percentage of beat positions which were calculated correctly by the beat tracking system, a measure of the performance of the system. This does not indicate the nature of the errors made during beat tracking. For each of the songs listed, the tempo was estimated correctly. Therefore the results column can be considered to be the percentage of the song for which the phase was tracked correctly.

More precise testing was performed by comparison of results with manually calculated beat positions. The audio files were examined with standard digital audio editing software (GoldWave), and beat locations were determined for a number of beats on which clear percussive events occurred. These locations were then used to interpolate the locations of the intervening beats.

The first 3 songs are standard modern pop/rock, characterized by a very steady tempo, which is clearly defined by simple and salient drum patterns, similar to the data used in the early audio beat tracking work of Goto [10]. In the production of this style of music, it is common practice for each instrument to be recorded separately, using a metronome track for synchronization. In this case one expects the performed beat to be very regular, with only small deviations from mechanical regularity. The beat tracking system made no errors on these songs.

The next style examined was Motown/Soul, characterized by more syncopation, greater tempo fluctuations (5-10% in these examples), and more freedom to anticipate or lag behind the beat. This created a greater difficulty for the beat tracking system, which made errors of phase several times, but recovered quickly to track almost all of each song correctly.

The Bob Dylan song was more difficult again, because of his idiosyncratic style of singing and playing against the rhythmic context. Although the beat is reasonably clear to a human listener, the drums are not prominent, and there is a much lower correlation between the conceptual beat and the actual musical events than in the other styles. There were two main sections in which the beat tracking system lost synchronization and tracked the off-beats (i.e., it continued at the correct tempo but half a beat out of phase), but again the system recovered to the correct phase.

The next test involved a live bossa nova performance with syncopated guitar and vocals, and very little percussion to indicate beat positions. The song was tracked correctly except in few passages where it went out of phase, but the error was corrected within a few beats.

The two jazz pieces were chosen for their particularly complex, syncopated rhythms, which are difficult for humans to follow. These pieces also provided examples of a different time signature, swing eighth notes, and greater tempo variation. In both cases, the highest scoring agent was able to track the majority of the piece correctly, but encountered phase errors in some parts.

Finally, a classical piece was tested, the first section of the third movement of Mozart's Piano Sonata in C major (KV279). The system lost synchronization several times, tracking the off-beats rather than the beats, due to large tempo variations and the system's lack of musical knowledge for distinguishing between beats and off-beats.

It is interesting to note that phase errors were also encountered for simple pop music in an earlier version of the system. With the salience calculation removed, the system tracks the whole of *I Don't Remember A Thing* at half a beat out of phase. For pop music, the salience of events differentiates the beat from the offbeat at most points in the music. This may not be true in jazz, where the offbeat is often accentuated for long periods of time, so the system requires an alternative way of choosing the correct path through the data.

Note that the beat tracking system is not equipped with musical knowledge — no notion of off-beats or expected rhythmic patterns has been programmed into it. Its apparent musical intelligence comes from patterns *in the data*, without any high-level knowledge or reasoning. Apart from the advantage of simplicity, a further advantage is that the system is quite robust, and generalizes well to different styles of music, as long as there is a salient beat. In order to disambiguate complex or ambiguous rhythmic patterns, the system will need sources of musical knowledge other than timing of events; these are not presently available to it. In current work [8], we are examining a specialization of the system for solo piano music which incorporates a level of musical and stylistic knowledge with the aim of extracting the score from performance data.

8 Conclusion

We have described a beat tracking system which analyses acoustic data, detects the salient note onsets, determines possible inter-beat intervals and then employs multiple agents to find a sequence of events which represents the beat of the music. The system successfully tracks the beat in most popular music, but makes some phase errors, mainly when presented with extremely complex rhythms or music with large tempo deviations. Even in these situations, the performance is quite robust, with the system recovering from its errors and resuming correct tracking after a short period.

Unlike previous audio beat tracking systems which required a large parallel computer [13], our system has modest requirements, processing an average length song in about 10 seconds on a current personal computer, leaving sufficient resources for real time applications based on the beat tracking system.

One such application is an automatic disc jockey (DJ), which plays a list of songs, cross-fading between the songs so that the beats of successive songs are synchronized (beat-mixing). This has been implemented as part of the beat tracking system. Another application is that of a score extraction system. This application uses MIDI input rather than audio, and the system's job is to make "musical sense" of the performed rhythm. The nature of this problem is different, in that we seek a musical explanation for every event, whereas the current system ignores events which are determined not to lie on the beat. MIDI input also facilitates the use of other knowledge from the data, such as duration, pitch, repeated melodic patterns and musical voice. In recent work, such information has proved useful in calculating a more detailed salience value for musical events [8]. It is still an open problem how such details can be extracted reliably directly from audio data. A further application of beat tracking, and one which requires reliable recognition of pitch and duration of notes, is an automatic music transcription system, that is, a system which produces musical scores directly from audio data. Such a system requires a beat tracker as one component.

The use of manual beat tracking for evaluation of the system limits the amount of testing that can be performed, but is necessary when analyzing performed music. Unlike speech-related research, music recognition research currently has no large corpora of tagged data for thorough testing and comparison of results (or, for that matter, the use of probabilistic search techniques). In further work we plan to perform a study of beat tracking in synthetically generated music, where the variations in tempo and onset times can be controlled precisely, and performance can be evaluated automatically.

9 Acknowledgements

This research is part of the project Y99-INF, sponsored by the Austrian Federal Ministry of Education, Science and Culture in the form of a START Research Prize and support to the Austrian Research Institute for Artificial Intelligence. The author wishes to thank Emilios Cambouropoulos and Gerhard Widmer for suggestions and comments on an earlier version of this paper.

References

1. P.E. Allen and R.B. Dannenburg. Tracking musical beats in real time. In *Proceedings of the International Computer Music Conference*, pages 140–143. International Computer Music Association, San Francisco CA, 1990.
2. P. Desain. A connectionist and a traditional AI quantizer, symbolic versus subsymbolic models of rhythm perception. *Contemporary Music Review*, Volume 9, pages 239–254, 1993.
3. P. Desain and H. Honing. Quantization of musical time: A connectionist approach. *Computer Music Journal*, Volume 13, Number 3, 1989.
4. P. Desain and H. Honing. Towards a calculus for expressive timing in music. *Computers in Music Research*, Volume 3, pages 43–120, 1991.
5. 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.
6. S. Dixon. Beat induction and rhythm recognition. In *Proceedings of the Australian Joint Conference on Artificial Intelligence*, pages 311–320, 1997.
7. S. Dixon. A beat tracking system for audio signals. In *Proceedings of the Diderot Forum on Mathematics and Music*, pages 101–110. Austrian Computer Society, 1999.
8. S. Dixon and E. Cambouropoulos. Beat tracking with musical knowledge. In *ECAI 2000: Proceedings of the 14th European Conference on Artificial Intelligence.* IOS Press, 2000. To appear.
9. W.J. Dowling and D.L. Harwood. *Music Cognition.* Academic Press, 1986.
10. 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.
11. M. Goto and Y. Muraoka. Issues in evaluating beat tracking systems. In *Issues in AI and Music – Evaluation and Assessment: Proceedings of the IJCAI'97 Workshop on AI and Music.* International Joint Conference on Artificial Intelligence, 1997.
12. M. Goto and Y. Muraoka. Real-time rhythm tracking for drumless audio signals – chord change detection for musical decisions. In *Proceedings of the IJCAI'97 Workshop on Computational Auditory Scene Analysis.* International Joint Conference on Artificial Intelligence, 1997.
13. M. Goto and Y. Muraoka. An audio-based real-time beat tracking system and its applications. In *Proceedings of the International Computer Music Conference.* Computer Music Association, San Francisco CA, 1998.
14. E.W. Large. Beat tracking with a nonlinear oscillator. In *Proceedings of the IJCAI'95 Workshop on Artificial Intelligence and Music.* International Joint Conference on Artificial Intelligence, 1995.
15. F. Lerdahl and R. Jackendoff. *A Generative Theory of Tonal Music.* MIT Press, 1983.
16. H.C. Longuet-Higgins. *Mental Processes.* MIT Press, 1987.
17. H.C. Longuet-Higgins and C.S. Lee. The perception of musical rhythms. *Perception*, Volume 11, pages 115–128, 1982.
18. D. Rosenthal. Emulation of human rhythm perception. *Computer Music Journal*, Volume 16, Number 1, pages 64–76, 1992.
19. W.A. Schloss. *On the Automatic Transcription of Percussive Music: From Acoustic Signal to High Level Analysis.* Ph.D. thesis, Stanford University, CCRMA, 1985.