Détection du tempo et du rythme dans un morceau de musique

ENS - M1RI - Projet de traitement du signal (n°2)

2013

Contexte et objectifs

Le tempo d'un morceau de musique, exprimé en *pulsations par minute* ou en anglais *beat per minute (bpm)* correspond intuitivement à sa vitesse, ou encore au nombre de fois que l'on taperait du pied pour en accompagner le rythme. Son estimation est un problème important dans de nombreuses tâches de traitement du signal musical, comme la reconnaissance du genre musical ou le mix automatique. Le « rythme » est une notion plus complexe, caractérisé par des motifs à différentes échelles de temps.

L'objectif de ce projet est en premier lieu de mettre en œuvre un détecteur de tempo basé sur l'article joint. L'identification du rythme pourra (facultativement) être abordée en dernière partie. La lecture de l'article est un préalable. Le traitement proposé s'appuie essentiellement sur 3 étapes :

1. La réalisation d'un banc de filtres, permettant un traitement ultérieur en sous-bandes.
2. Dans chaque sous-bande, un traitement de détection d'attaque, composé essentiellement d'une extraction d'enveloppe, d'une dérivation et d'une décimation.
3. Dans chaque sous-bande, les signaux modifiés par l'étape précédente passent en parallèle dans une série de filtres résonnants (filtre en peigne de fréquence donnée). Les sorties d'un filtre résonnant donné sont sommées, et le tempo détecté est celui du filtre résonnant dont la sortie est d'énergie maximale.

Les signaux fournis se présentent sous 2 formes : les fichiers `signalxx.wav` sont des extraits musicaux sur lesquels on cherche à estimer le tempo (supposé constant) ; les fichiers `tapxx.wav` sont les fichiers de « tapping » correspondants (enregistrement de la pulsation tapée par un opérateur musicien écoutant simultanément le signal au casque) et pourront être utilisés pour les tests et pour évaluer la performance finale.

1 Banc de filtres

En préalable au traitement, le signal d'entrée passe dans un banc de filtres, résultant en un certain nombre de signaux de sous-bande dont le contenu fréquentiel est concentré dans une bande de fréquences étroite. Implémenter :

1. Le banc de filtres RII proposé par Scheirer.
2. Un banc de filtres TFD, par exemple à M =16 ou 32 voies, dont les réponses en fréquences vérifient la relation : $H_k(e^{2i\pi\nu}) = H(e^{2i\pi(\nu-k)/M})$. Le filtre prototype sera obtenu par la méthode de Remez, avec une bande de transition $[0.9\nu_c; 1.1\nu_c]$ (la fréquence de coupure ν_c sera déterminée en fonction du nombre de voies du filtre) et une atténuation de 40 dB.

2 Détection d'attaque

Dans chaque sous-bande, on implémentera la détection d'attaque proposée par Scheirer, en deux étapes :

1. Une détection d'enveloppe par convolution avec une demi-fenêtre de Hanning ;
2. Une dérivation simple (différences finies d'ordre 1) suivie d'une « rectification demi-onde » (les valeurs négatives sont mises à zéro).

On représentera les signaux à chaque étape de ce traitement sur les signaux de test. (Facultatif : comparer avec le traitement similaire effectué par le groupe 1 pour la détection d'attaque). Les signaux ainsi filtrés seront ensuite décimés comme proposé dans l'article.

3 Filtres résonateurs

Les signaux de sous-bande issus du traitement précédent sont ensuite mis en entrée d'une série de filtres résonateurs à différentes fréquences balayant les tempos possibles (typiquement entre 40 et 180 bpm). Implémenter ces filtres résonateurs, observez leur réponse en fréquence et illustrer (représenter leurs sorties pour différentes entrées).

On pourra s'aider des réponses aux questions suivantes :

— quelle est la relation entre T et le tempo exprimé en bpm ? (ne pas oublier l'étape de décimation)
— pourquoi ces filtres résonateurs sont appelés « filtres en peigne » ?
— montrer que les filtres résonateurs s'expriment comme le suréchantillonnage d'un filtre de référence $H_r(z)$
— en déduire la réponse d'un filtre H_T lorsqu'on lui applique en entrée un train d'impulsions de largeur T.

On achèvera la détection de tempo par la sommation des sorties des filtres dans chaque sous-bande et par la recherche de l'énergie maximale et on testera ces traitements sur un signal synthétique de test (train d'impulsions gaussiennes, par exemple) puis sur les signaux fournis. La vraie valeur (« groundtruth ») sera extraite des fichiers contenant un tapping ajouté manuellement.

4 Identification du rythme (facultatif)

Une fois le tempo détecté, on peut envisager d'identifier le type de rythme par une approche supervisée : on possède une base de motifs rythmiques étiquetés par genre pour procéder à des comparaisons (fichier `salsa.wav`, `rock.wav`, `reggae.wav`). Le but de cette partie est de chercher dans cette base le rythme le mieux aligné avec celui du morceau inconnu, en trois étapes :

1. Estimer le tempo des rythmes étiquetés.
2. Modifier l'échelle temporelle de ces rythmes pour que leur tempo soit similaire à celui qu'on a estimé sur le morceau cible. On procèdera par un simple changement de fréquence d'échantillonnage, la hauteur n'ayant ici aucune importance.
3. Utiliser l'algorithme appelé *dynamic time warping* (DTW) pour aligner le morceau cible sur chaque rythme étiqueté. On choisira judicieusement les signaux à aligner (ce n'est pas forcément le signal brut : on pourra utiliser des intermédiaires de calcul produit par le système de détection du tempo, choix à justifier). On décidera que le rythme du morceau cible est le rythme étiqueté pour lequel l'alignement DTW a été le moins coûteux.

Tempo and beat analysis of acoustic musical signals

Eric D. Scheirer[a)]
Machine Listening Group, E15-401D MIT Media Laboratory, Cambridge, Massachusetts 02139

(Received 27 December 1996; revised 26 August 1997; accepted 15 September 1997)

A method is presented for using a small number of bandpass filters and banks of parallel comb filters to analyze the tempo of, and extract the beat from, musical signals of arbitrary polyphonic complexity and containing arbitrary timbres. This analysis is performed causally, and can be used predictively to guess when beats will occur in the future. Results in a short validation experiment demonstrate that the performance of the algorithm is similar to the performance of human listeners in a variety of musical situations. Aspects of the algorithm are discussed in relation to previous high-level cognitive models of beat tracking. © *1998 Acoustical Society of America.* [S0001-4966(98)02801-X]

PACS numbers: 43.75.Yy, 43.75.St [WJS]

INTRODUCTION

Automatic extraction of rhythmic pulse from musical excerpts has been a topic of active research in recent years. Also called *beat-tracking* and *foot-tapping,* the goal is to construct a computational algorithm capable of extracting a symbolic representation which corresponds to the phenomenal experience of ''beat'' or ''pulse'' in a human listener.

''Rhythm'' as a musical concept is intuitive to understand, but somewhat difficult to define. Handel writes ''The experience of rhythm involves movement, regularity, grouping, and yet accentuation and differentiation'' (Handel, 1989, p. 384) and also stresses the importance of the phenomenalist point of view—there is no ''ground truth'' for rhythm to be found in simple measurements of an acoustic signal. The only ground truth is what human listeners agree to be the rhythmic aspects of the musical content of that signal.

As contrasted with ''rhythm'' in general, ''beat'' and ''pulse'' correspond only to ''the sense of equally spaced temporal units'' (Handel, 1989). Where ''meter'' and ''rhythm'' associate with qualities of grouping, hierarchy, and a strong/weak dichotomy, ''pulses'' in a piece of music are only periodic at a simple level. For our purposes, the *beat* of a piece of music is the sequence of equally spaced phenomenal impulses which define a tempo for the music. This paper is only concerned with beat and tempo. The grouping and strong/weak relationships which define rhythm and meter are not considered.

It is important to note that there is no simple relationship between polyphonic complexity—the number and timbres of notes played at a single time—in a piece of music, and its rhythmic complexity or pulse complexity. There are pieces and styles of music which are texturally and timbrally complex, but have straightforward, perceptually simple rhythms; and there also exist musics which deal in less complex textures but are more difficult to rhythmically understand and describe.

The former sorts of musical pieces, as contrasted with the latter sorts, have a ''strong beat,'' and it is with them that this paper is predominantly concerned. For these kinds of music, the rhythmic response of listeners is simple, immediate, and unambiguous, and every listener will agree on the rhythmic content. Rhythmically complex music is discussed toward the end of the paper.

Previous approaches

There is a large body of work originating in the music-psychology community which attempts to group musical *onsets* together into a rhythmic context; that is to say, to construct a model which subsumes multiple onsets separated in time into a rhythmic clock, ''hierarchy,'' grouping, or oscillatory model.

Povel and Essens presented research (Povel and Essens, 1985) on the association of ''internal clocks'' with temporal onset signals. They described an algorithm which could, given a set of inter-onset intervals as input, identify the clock which a listener would associate with such a sequence of intervals. Their research was particularly interested in the way that perceived accents lead to the internal clock. Although obviously related to music, their research purports to examine time intervals in general rather than being restricted to musical stimuli. Parncutt's recent work (Parncutt, 1994) extends this type of model to include a great deal of structural information about duration and phenomenal accent.

Desain and Honing have contributed many results to the computational modeling of beat-tracking. Their models (Desain and Honing, 1992; Desain, 1995) typically also begin with inter-onset intervals and associate a rhythmic pulse with the interval stream. However, unlike the Povel/Essens and Parncutt models, these models are *process models*—they process the input sequentially rather than all-at-once—a necessary aspect of a model of human rhythmic perception. Desain's ''(de)composable'' model calculates rhythmic expectations due to each of the possible inter-onset times in a rhythmic stream, and sums them to create an overall rhythmic expectation.

Large and Kolen have described a beat-tracking model (Large and Kolen, 1994) based on nonlinear oscillators. The model takes a stream of onsets as input, and uses a gradient-descent method to continually update the period and phase of an oscillator. In this manner, the oscillator is matched with

[a)]Electronic mail; eds@media.mit.edu

the input stream, and the resulting oscillation process seems to be a good match for the human perception of beat.

Longuet-Higgens and Lee have written many papers (for example, Longuet-Higgens and Lee, 1984) on the induction of rhythmic hierarchies from monophonic time sequences. They are more interested in the development of theories which describe the relationship of rhythm, meter, and phrasing than on the boot-strapping process which creates a tempo and beat percept. Tempo perception may be viewed as ''underlying'' their models.

These approaches, and others such as Rosenthal (1993) and Brown (1993), require that robust onset detection precede beat analysis, which entails an important restriction to their applicability. The models do not operate on acoustic signals, but on symbolic data such as event lists or MIDI. As the extraction of onsets from multitimbral, polyphonic music is itself a difficult problem, this is a serious restriction of any model which claims to treat human rhythm perception. There has been little attempt to merge these sorts of models with real-time acoustic pattern recognition to allow them to work with acoustic data.

More recently, there has been some research attempting to extract rhythm and/or pulse information directly from acoustic signals. Goto has demonstrated a system which combines both low-level ''bottom-up'' signal processing and high-level pattern matching and ''agent-based'' representations to beat-track and do simple rhythmic grouping for popular music (Goto, in press). His method extracts drum patterns from a signal and uses a template-matching model to determine the beat from the drum track. This system runs in real time on a parallel-processing computer and has been used to control interactive-graphics displays from ecological music signals. His description does not directly address the equivalent processing of signals without drums, but it seems that the required musical knowledge base would be much more difficult to acquire.

N. P. Todd's work (Todd, 1994) has described algorithms which detect onsets in monophonic music under certain timbral constraints, and then group these onsets in a rhythmic framework using a multi-scale smoothing model. The onset model used is a simple one based on leaky integration. The resulting ''rhythmogram'' representation conceives of pulse, and in some cases, meter and phrase, perception as a very low-level process arising directly from the time- and loudness-integration properties of the auditory periphery. The model as presented can be implemented in an incremental manner, but was only tested using toy examples (although, interestingly, a speech example was included).

All of the abovementioned research uses what has been described as a *transcriptive* metaphor for analysis (Scheirer, 1996). That is, the music is first segmented, or assumed to already be segmented, into notes, onsets, timbres, and so forth. Post-processing algorithms are then used to group rhythms and track beats. As high-quality polyphonic music transcription algorithms are still years in the future—the state-of-the-art systems cannot transcribe pieces more complex than four-voice piano music (Martin, 1996)—it seems logical for practical reasons to attempt to construct systems which can arrive at a musical understanding of a piece of music without going through a transcription step. Further, as the validity of the transcriptive metaphor as a framework for music perception has been challenged (Scheirer, 1996), it is scientifically appropriate as well.

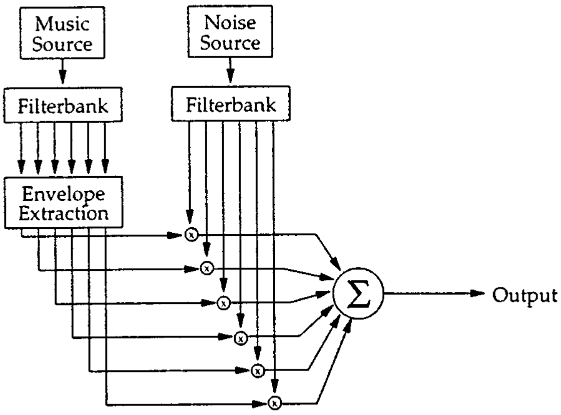

FIG. 1. Creating a ''modulated noise'' signal from a music signal. The output signal, for many sorts of frequency filterbanks, will have the same rhythmic percept as the input music signal, indicating that the amplitude envelopes of the bands are a sufficient representation for rhythmic analysis.

In the body of this paper, the following topics are discussed: psychoacoustic demonstrations which lead to processing simplifications for beat-tracking, the construction of the algorithms themselves, example results from test signals and ecological signals, a validation experiment which compares the behavior of the algorithm to that of human subjects, the relationship of this model to previous models of rhythm perception, and finally, conclusions about beat-tracking and rhythmic grouping and a description of future work to be pursued in these directions.

I. PSYCHOACOUSTIC SIMPLIFICATION

One of the key difficulties with the transcriptive models of rhythmic perception described above is the complexity of grouping harmonic partials together to form notes, and determining the onset times of those notes. Even if simplifying assumptions about the pitch and timbral content are made, identifying attack and release times is no easy task (Scheirer, in press).

However, it seems from a psychoacoustic demonstration on beat perception that certain kinds of signal manipulations and simplifications can be performed without affecting the perceived pulse content of a musical signal. Consider the signal flow network shown in Fig. 1.

An ''amplitude-modulated noise'' is constructed by signal by vocoding a white noise signal with the subband envelopes of a musical signal. This is accomplished by performing a frequency analysis of the music (processing through a filterbank of bandpass filters, perhaps, or grouping output from FFT bins together), and also of a white-noise signal from a pseudo-random generator. The amplitude of each band of the noise signal is modulated with the amplitude envelope of the corresponding band of the musical filterbank output, and the resulting noise signals are summed together to form an output signal.

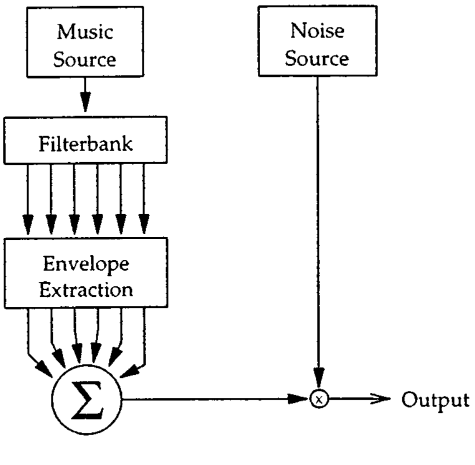

FIG. 2. A noise signal which does not have the same rhythmic characteristics as the musical input, indicating that the sum of the amplitude envelopes is not a sufficient representation for rhythm analysis. Certain types of non-linear combination by frequency channel are evidently present in the beat perception facility.

For many kinds of frequency filterbanks, the resulting noise signal has a rhythmic percept which is significantly the same as that of the original music signal. Even if there are very few, very broad bands (for example, four three-octave bands covering the audible spectrum), the pulse and meter characteristics of the original signal are instantly recognizable (sound example #1a) [Audio examples for this paper can be found on the author's WWW site at http://sound.media.mit.edu/eds/beat/].

Since the only thing preserved in this transformation is the amplitude envelopes of the filterbank outputs, it stands to reason that only this much information is necessary to extract pulse and meter from a musical signal; that is, algorithms for pulse extraction can be created which operate only on this much input data, and ''notes'' are not a necessary component for hearing rhythm. This is a vast reduction of input data size from the original signal. Shannon has reported a similar effect for the perception of speech (Shannon, 1995).

Certain other kinds of simplifications are not possible. For example, if only one band is used, or equivalently, the subband envelopes are linearly combined before modulating the noise (Fig. 2) (Vercoe, 1994) a listener can no longer perceive the rhythmic content of many signals (sound example #1b). Thus it seems that separating the signal into subbands and maintaining the subband envelopes separately is necessary to do accurate rhythmic processing.

Stated another way, the algorithm in Fig. 2 is a method for generating new signals whose representation under a filterbank-envelope-and-sum process is the same as a given piece of music. However, since these new signals often do not bear a perceptual equivalency with the originals, the filter-envelope-sum framework must be *inadequate* to represent data in the musical signal which is important for rhythmic understanding. This fact immediately leads to a psychoacoustic hypothesis regarding rhythmic perception: some sort of cross-band rhythmic integration, not simply summation across frequency bands, is performed by the auditory system.

A psychoacoustic experiment to examine the exact properties of filterbank-and-envelope manipulations which do not disturb rhythm perception is underway; in the meantime, it seems important that a rhythmic processing algorithm should treat frequency bands separately, combining results at the end, rather than attempting to perform beat-tracking on the sum of filterbank outputs.

II. DESCRIPTION OF ALGORITHM

The beat-tracking algorithm to be presented here bears most resemblance to the method of Large and Kolen (Large and Kolen, 1994) in that it uses a network of resonators to phase-lock with the beat of the signal and determine the frequency of the pulse. However, the particular method used here is somewhat different; the resonators are analytically much simpler than theirs, a bank of resonators is used rather than gradient descent, and more pre- and post-processing of the signal is necessary in order to accurately extract the desired information, as the present model operates on acoustic data rather than an event stream.

A rhythmic pulse is described in terms of a frequency and phase component, just as for a periodic sound waveform; the frequency of the pulse in a rhythmic musical signal is the tempo or rate of the rhythm, and the phase of the pulse indicates where the ''downbeat'' of the rhythm occurs. That is, the times at which a pulse occurs can be defined to have zero phase, and thus the points in time exactly in-between pulses have phase of π radians, etc. It is important to note that while human pitch recognition is only sensitive to signal phase under certain unusual conditions, rhythmic response is crucially a phased phenomenon—tapping on the beat is not at all the same as tapping against the beat, or slightly ahead of or behind the beat, even if the frequency of tapping is accurate.

Figure 3 shows an overall view of the tempo-analysis algorithm as a signal flow network. The functionality will be briefly described, and then more details given piece-by-piece in the following sections. The algorithms here were developed empirically; however, in Sec. V their relationship to existing models of rhythm perception is discussed.

As the signal comes in, a filterbank is used to divide it into six bands. For each of these subbands, the amplitude envelope is calculated and the derivative taken. Each of the envelope derivatives is passed on to another filterbank of *tuned resonators*; in each resonator filterbank, one of the resonators will phase-lock, the one for which the resonant frequency matches the rate of periodic modulation of the envelope derivative.

The outputs of the resonators are examined to see which ones are exhibiting phase-locked behavior, and this information is tabulated for each of the bandpass channels. These tabulations are summed across the frequency filterbank to arrive at the frequency (tempo) estimate for the signal, and reference back to the peak phase points in the phase-locked resonators determines the phase of the signal.

A. Frequency analysis and envelope extraction

As discussed in Sec. I, envelopes extracted from a small number of broad frequency channels are sufficient information to rhythmically analyze a musical signal, at least for

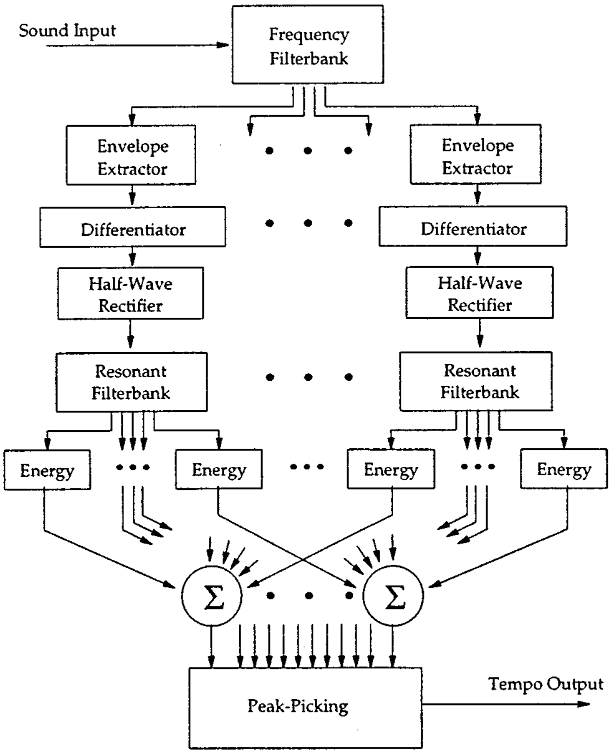

FIG. 3. Schematic view of the processing algorithm. See text for details.

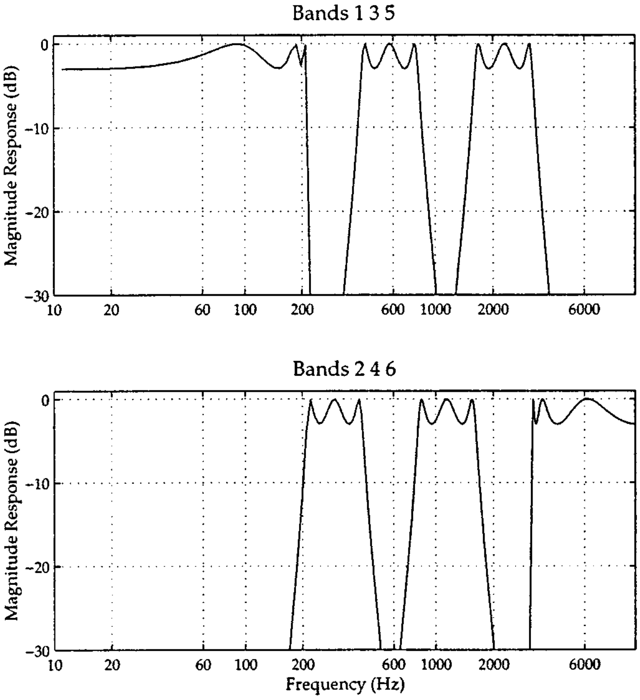

FIG. 4. Magnitude response of the frequency filterbank used in the system, plotted in two pieces for clarity. The upper plot shows the first, third, and fifth bands; the lower, the second, fourth, and sixth. Each filter is a sixth-order elliptic filter, with 3 dB of passband ripple and 40 dB of stopband rejection.

human listeners. Further, empirical studies of the use of various filterbanks with this algorithm have demonstrated that the algorithm is not particularly sensitive to the particular bands or implementations used; it is expected that psychoacoustic investigation into rhythmic perception of amplitude-modulated noise signals created with the various vocoder filterbanks would confirm that the same is true of human rhythmic perception.

The filterbank implementation in the algorithm has six bands; each band has sharp cutoffs and covers roughly a one-octave range. The lowest band is a low-pass filter with cutoff at 200 Hz; the next four bands are bandpass, with cutoffs at 200 and 400 Hz, 400 and 800 Hz, 800 and 1600 Hz, and 1600 and 3200 Hz. The highest band is high pass, with cutoff frequency at 3200 Hz. Each filter is implemented using a sixth-order elliptic filter, with 3 dB of ripple in the passband and 40 dB of rejection in the stopband. Figure 4 shows the magnitude responses of these filters.

The envelope is extracted from each band of the filtered signal through a rectify-and-smooth method. The rectified filterbank outputs are convolved with a 200-ms half-Hanning (raised cosine) window. This window has a discontinuity at time $t=0$, then slopes smoothly away to 0 at 200 ms. It has a low-pass characteristic, with a cutoff frequency at about 10 Hz (''frequency'' in this case referring to envelope spectra, not waveform spectra), where it has a -15 dB response, and 6-dB/octave smooth rolloff thereafter.

The window's discontinuity in time means that it has nonlinear phase response; it passes slow envelope frequencies with much more delay than rapid ones. High frequencies, above 20 Hz, are passed with approximately zero delay; 0 Hz is delayed about 59 ms and 7 Hz advanced about 14 ms. Thus there is a maximum blur of about 73 ms between these envelope frequencies.

This window performs energy integration in a way similar to that in the auditory system, emphasizing the most recent inputs but masking rapid modulation; Todd (1992) examines the use of temporal integration filters which are directly constructed from known psychoacoustic properties. After this smoothing, the envelope can be decimated for further analysis; the next stages of processing operate on the decimated band envelopes sampled at 200 Hz. There is little energy left in the envelope spectra at this frequency, but it aids the phase-estimation process (see below) to maintain a certain precision of oversampled envelope resolution.

After calculating the envelope, the first-order difference function is calculated and half-wave rectified; this rectified difference signal will be examined for periodic modulation. The derivative-of-envelope function performs a type of onset filtering process (see, for example, Smith's work on difference-of-Gaussian functions for onset segmentations Smith, 1994) but the explicit segmentation, thresholding, or peak-peaking of the differenced envelope is not attempted. The subsequent modulation detectors in the algorithm are sensitive, similar to the sensitivity of autocorrelation, to ''imperfections'' in an onset track. The half-wave rectified envelope difference avoids this pitfall by having broader (in time) response to perceptual attacks in the input signal. This process might be considered similar to detecting onset points in the signal bands, and then broadening them via low-pass filtering.

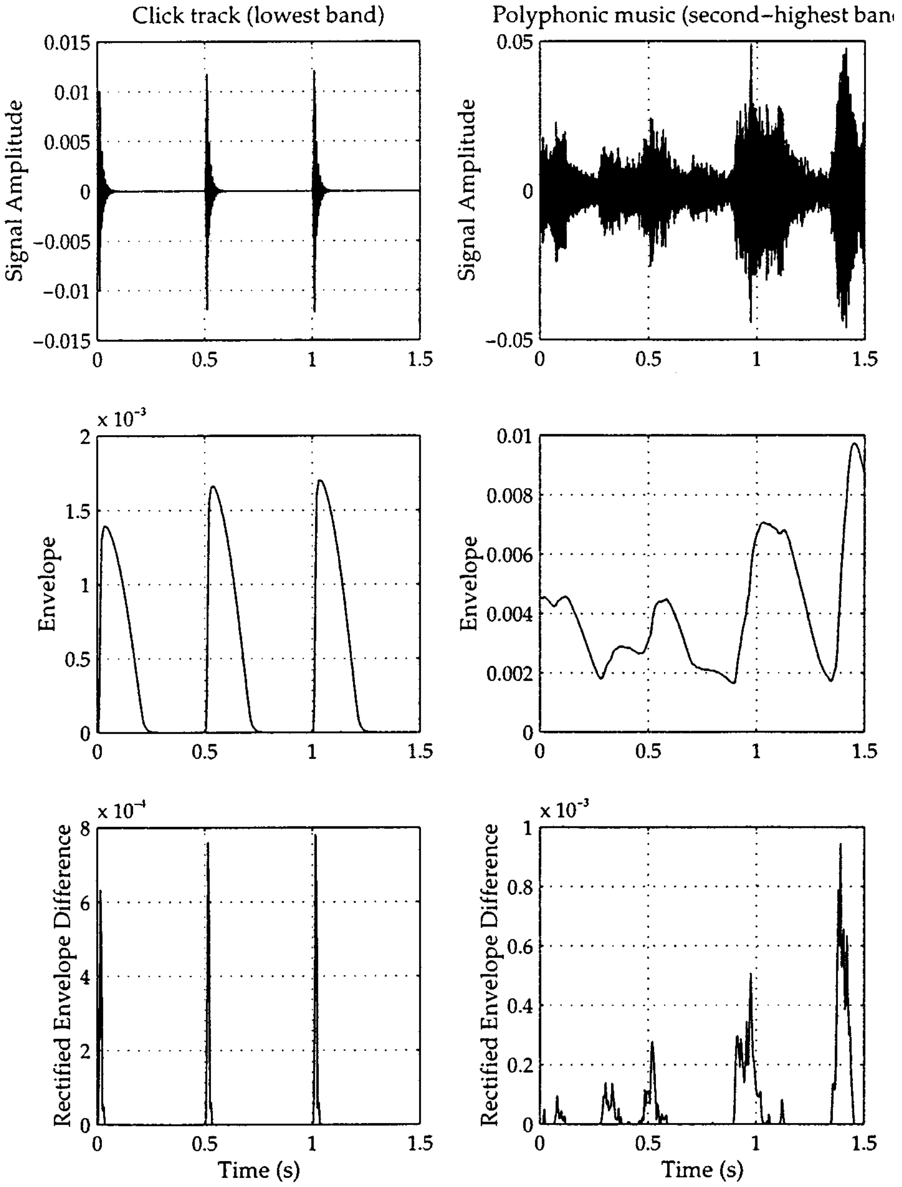

FIG. 5. Envelope extraction process, for a 2-Hz click track (left) and a polyphonic music example (right). The top panels show the audio waveforms; the middle panels, the envelopes; and the bottom, the half-wave rectified difference of envelopes. The lowest filterbank band is shown for the click track, the second-highest for the music. See text for details on algorithms.

Figure 5 shows the envelope extraction process for one frequency band in each of two signals, a 2-Hz click track and a polyphonic music example. The lowest band is shown for the click track, and the second highest for the music track.

B. Resonators and tempo analysis

After the envelope has been extracted and processed for each channel, a filterbank of comb filter resonators is used to determine the tempo of the signal. While comb filters are often used in reverberators and other sorts of audio signal processing, they also have properties which make them suitable for acting as resonators in the phase-locking pulse extraction process.

In particular, if we stimulate a comb filter with delay T and gain α with a right-sided pulse train of height A and period κ, we get reinforcement (resonance) if $T=\kappa$. Let x_t and y_t be the input and output signals at time t; the equation of the filter is then $y_t=\alpha y_{t-T}+(1-\alpha)x_t$, and

$$y_0=(1-\alpha)A$$

$$y_\kappa=\alpha(1-\alpha)A+(1-\alpha)A=(1-\alpha)A(1+\alpha)$$

$$y_{2\kappa}=(1-\alpha)A(\alpha^2+\alpha+1)$$

$$\vdots$$

$$y_{n\kappa}=(1-\alpha)A\left(\sum_{i=0}^{n}\alpha^i\right).$$

And so $\lim_{n\to\infty}y_{n\kappa}=[(1-\alpha)A]/(1-\alpha)=A$.

On the other hand, if $T\neq\kappa$, the convergence is to a smaller value. Let λ be the least common multiple (common period) of T and κ; there is only reinforcement every T/λ periods, and by a similar logic as the above,

$$\lim_{n\to\infty} y_{n\lambda} = \frac{(1-\alpha)A}{1-\alpha^{T/\lambda}},$$

and since $|\alpha|<1$ if the filter is to be stable, and $T/\lambda \geqslant 1$,

$$1-\alpha^{T/\lambda} \geqslant 1-\alpha.$$

So a filter with delay matching (or evenly dividing) the period of a pulse train will have larger (more energetic) output than a filter with mismatched delay.

We can see that this is true for any periodic signal by doing the analysis in the frequency domain. The comb filter with delay T and gain α has magnitude response

$$|H(e^{j\omega})| = \left|\frac{1-\alpha}{1-\alpha e^{-j\omega T}}\right|,$$

which has local maxima wherever $\alpha e^{-j\omega T}$ gets close to 1, i.e., at the Tth roots of unity, which can be expressed as

$$e^{-j2\pi n/T}, \quad 0 \leqslant n < T.$$

Using Fourier's theorem we know that these frequency-domain points are exactly those at which a periodic signal of period T has energy. Thus the comb filter with delay T will respond more strongly to a signal with period T than any other, since the response peaks in the filter line up with the frequency distribution of energy in the signal.

For each envelope channel of the frequency filterbank, a filterbank of comb filters is implemented, in which the delays vary by channel and cover the range of possible pulse frequencies to track. The output of these resonator filterbanks is summed across frequency subbands. By examining the energy output from each resonance channel of the summed resonator filterbanks, the strongest periodic component of the signal may be determined. The frequency of the resonator with the maximum energy output is selected as the tempo of the signal.

The α parameter for each comb filter is set differently, so that each filter has equivalent half-energy time. That is, a comb filter of period T has an exponential curve shaping its impulse response. This curve reaches half-energy output at the time t when $\alpha^{T/t}=0.5$. Thus α is set separately for each resonator, at $\alpha=0.5^{t/T}$. A half-energy time of 1500–2000 ms seems to give results most like human perception.

Figure 6 shows the summed filterbank output for a 2-Hz pulse train and for a polyphonic music example. The horizontal axis is labeled with ''metronome marking'' in beats per minute; this is a direct mapping of the delay of the corresponding comb filter. That is, for the 2-Hz power envelope signal, a feedback delay of 100 samples corresponds to a 500-ms resonance period, or a tempo of 120 bpm.

In the pulse train plot in Fig. 6, a clear, large peak occurs at 120 bpm, and additional smaller peaks at tempi which bear a simple harmonic relationship (3::2 or 4::5, for example) to the main peak. In the music plot, there are two peaks, which correspond to the tempi of the quarter note and half note in this piece. If the width of the upper plot were extended, a similar peak at 60 bpm would be visible.

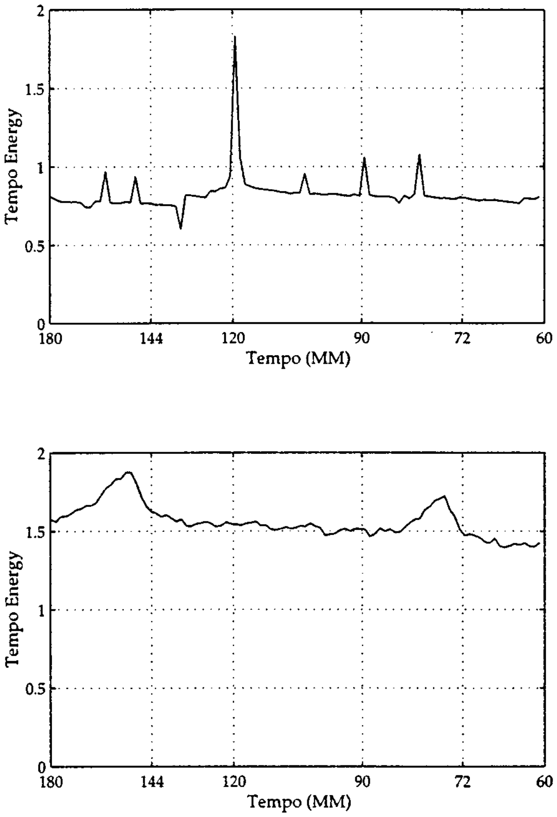

FIG. 6. Tempo estimates, after tracking 5 s of a 2-Hz click track (top) and of a polyphonic music example (bottom). The x-axes are labeled in beats per minute, that is, 120 MM=2 Hz. The polyphonic music shows more overall energy, but the tempo is still seen clearly as peaks in the curve.

C. Phase determination

It is relatively simple to extract the phase of the signal once its tempo is known, by examining the output of the resonators directly, or even better, by examining the internal state of the delays of these filters. The implementations of the comb filters for the resonator filterbank have lattices of delay-and-hold stages. The vector w of delays can be interpreted at a particular point in time as the ''predicted output'' of that resonator; that is, the next n samples of envelope output which the filter would generate in response to zero input.

The sum of the delay vectors over the frequency channels for the resonators corresponding to the tempo determined in the frequency extraction process are examined. The peak of this prediction vector is the estimate of when the next beat will arrive in the input, and the ratio $\omega=2\pi(t_n-t)/T$, where t_n is the time of the next predicted beat, t the current time, and T the period of the resonator, is the phase ω of the tempo being tracked. The phase and period may thus be used to estimate beat times as far into the future as desired.

The implementation of the model performs the phase analysis every 25 ms and integrates evidence between frames in order to predict beats. Since re-estimation occurs multiple times between beats, the results from each phase analysis can be used to confirm the current prediction and adjust it as

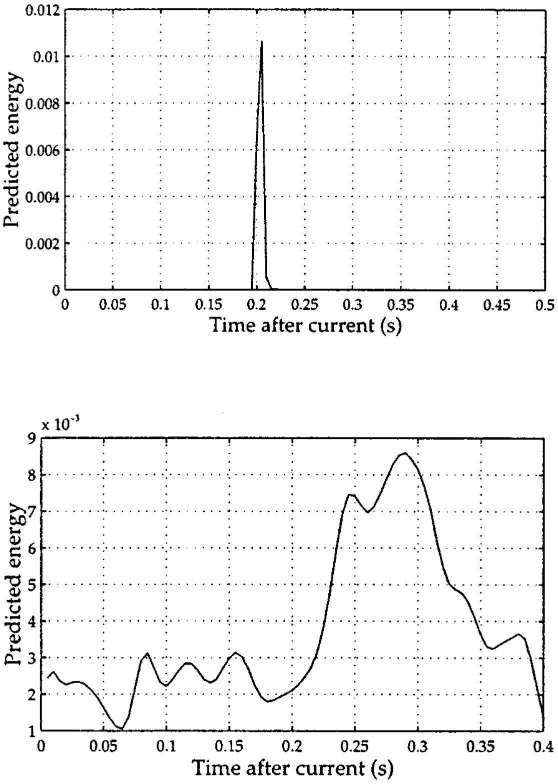

FIG. 7. Phase estimates, after tracking 5 s of a 2-Hz click track (top) and a polyphonic music example (bottom). The x-axis in each case covers the next full period of the resonator tracking the tempo, and the peak of the curve shows where the next beat is predicted to occur: about 210 ms in the future for the upper case, and 290 ms for the lower.

needed. Currently, this prediction/adjustment is done in an *ad hoc* manner, requiring only that several successive frames make the same beat prediction within a certain tolerance, and average all of these estimates to arrive at the final prediction. This stage is the appropriate one for the inclusion of high-level information, nondeterministic elements, or more sophisticated rhythmic modeling; see Sec. VI.

Figure 7 shows the phase peaks for a 2-Hz pulse train, and for a polyphonic music example. In the upper plot, as the tempo is 120 bpm, the x-axis covers the next half-second of time; and for the lower plot, the estimated tempo is 149 bpm (see Fig. 6), so one period is approximately 400 ms.

D. Comparison with autocorrelation methods

There is a certain analytical similarity between this bank-of-comb-filters approach and previous autocorrelation methods for finding tempo. Insofar as both are ways of detecting periodic energy modulations in a signal, they are performing similar calculations. However, there are several advantages to expressing these operations as multiple comb filters over expressing them as autocorrelation.

Predominantly, comb filtering implicitly encodes aspects of rhythmic hierarchy, where autocorrelation does not. That is, a comb filter tuned to a certain tempo τ has peak response to stimuli at tempo τ, but also lesser response to stimuli with tempi at multiples $(2\tau,3\tau)$, fractions $(\tau/2,\tau/3)$, and simple rational relationships $(3/2\tau,3/4\tau$, etc). The autocorrelation only has this shared response for fractional tempi, not multiples or rationally related tempi. An autocorrelation model asserts that a click track at 60 bpm gives no sense of tempo at 120 bpm, which seems intuitively wrong. The comb filter model asserts instead, that there is such a sense, but a reduced one when compared to a click track to 120 bpm.

These responses can be understood if we imagine building an autocorrelation filter at some lag, versus a comb filter at that same delay, in an FIR manner (that is, to unroll the usual IIR expression of the comb filter into an infinitely long ''FIR'' filter). The autocorrelation requires only a single tap on a delay line, since it only compares ''one cycle back'' in time. The comb filter requires an infinite number of taps, since it compares (with less and less weight) infinitely far back in time.

Autocorrelation methods are zero phase, which means that some other method of determining signal phase must be used. The comb filtering method shown here is phase preserving, and so provides a way of simultaneously extracting tempo and phase, as discussed in the previous section. The fact that the tempo and phase representations arise together gives us additional advantages in constructing higher-level processing algorithms treating the output of the beat-tracker.

One advantage of autocorrelation schemes is that they are more efficient in memory usage than banks of comb filters, as the various lags can all access the same delay line—which is why the autocorrelation is zero phase—whereas each comb filter must maintain a delay line of its own. In return for the extra memory usage, the comb filters provide estimates of output energy at each phase angle of each lag, where the autocorrelation accumulates it and only presents the summary.

Ultimately, it is representationally satisfying to have the frequency and phase of the signal explicitly encoded in the processing units of the algorithm. In an autocorrelation methodology, the rhythmic oscillations of the signal are only represented as post-processed summary results; whereas in the comb filtering method, the filter states themselves explicitly represent the rhythmic content—that is, there is an element of the processing network which phase-locks to and oscillates in synchrony with the signal.

III. IMPLEMENTATION AND COMPLEXITY

The algorithms described above have been implemented in $C++$ code; the resulting program causally processes audio files captured from compact disks or other audio recordings, or coming in via a live microphone input. In this section, the parameters available for controlling the speed and accuracy of the program are described.

A. Program parameters

The current implementation of the system has a number of parameters which can be used to control the accuracy/speed relationship of the algorithms. The program will run in real time on a very fast desktop workstation such as a DEC Alpha, depending on the settings of these parameters and the sampling rate of the incoming audio stream. It is also clear, due to the highly parallel structure of Fig. 3, that the algorithm could efficiently make use of a multiple-processor architecture. This has not yet been accomplished, however.

There are four major areas where the performance and accuracy of the system can be tuned, and control over three of them has been implemented. The algorithm has been tested for audio at sampling rates from 8 KHz to 44.1 KHz and gives roughly equivalent qualitative performance in all of these.

1. Frequency filterbank

As discussed in Sec. II, there is a fair amount of latitude in choosing a frequency filterbank for decomposing the incoming audio stream without affecting human rhythmic perception, and the speed of the system will vary a great deal with the complexity of these filters (since there is a fair CPU load for implementing high-order filters in real time on high-bandwidth audio), and their number (since for each of the frequency channels, a full resonator filterbank structure is implemented).

The performance of the beat-tracking program using filterbanks other than the six-channel sixth-order IIR filterbank described above has not been tested.

2. Envelope sampling rate

The decimation rate of the channel envelopes affects the speed and performance of the system. There are two major implications for using a slow envelope sampling rate: (1) there are many resonator frequencies which cannot be represented accurately with integer delays in the comb filters; and (2) the phase extraction can only be performed with accuracy equal to the envelope sampling rate, since the vector of delays has the same sampling rate.

In tradeoff to this, using a fast sampling rate for the envelopes entails a lot of work in the comb filtering, since the number of multiplies in each comb filter varies proportionately to this rate. Empirical testing over a variety of musical examples suggests that the envelopes should be sampled at least 100 Hz or so for best performance.

3. Number of resonators per frequency channel

The amount of computing incorporated in tracking and analysis of the comb filter resonators varies directly with their number. If too few resonators are used, however, a problem develops with sampling the tempo spectrum too sparsely. That is, since each resonator is attempting to phase-lock to one particular frequency (not to a range of frequencies), if there is no resonator tuned close to the tempo of a particular signal, that signal cannot be accurately tracked.

Also affecting this sparsity consideration is the range of resonator frequencies to be tracked. The wider the range of tempi to track, the sparser a fixed number of resonators will spread over that range.

Good results have been generated using a bank of 150 resonators for each channel, covering a logarithmically spaced range of frequencies from 60 bpm (1 Hz) to 240 bpm (3 Hz).

4. Analysis frame rate

In this particular implementation, a higher-level averaging scheme is used to decide where (at what times) to deduce beats in the input signal. That is, for each analysis frame, the phases of the resonators are examined; the evidence here suggests future beat locations. These suggestions are combined over multiple analysis frames; when several frames in a row point to the same future beat location, evidence accumulates for that time, and a beat is actually assigned there.

Thus the frequency with which the procedure of examining and summing the outputs and internal states of the resonators is executed has a strong effect upon the performance and speed of the program. Good results can be obtained if the analysis frame rate is at least 15 Hz.

Real-time performance cannot be obtained with the parameter values shown above; on an Alpha 3000 using highly optimized filtering and analysis code, with the envelope rate set to 75 Hz, 50 resonators per subband, and frames of beat predictions analyzed every 10 Hz, the required performance for real-time operation on 22-KHz input is reached. This real-time performance includes reading the sound file from disk and playing it back with short noise bursts added to highlight the beats. At this level of accuracy, the algorithm still performs acceptably well on some, but not all, musical examples.

B. Behavior tuning

In addition to controlling the tradeoff between program speed and accuracy, the behavior of the algorithm can be tuned with the α parameters in the comb filters. These parameters can be viewed as controlling whether to value old information (the beat signal extracted so far) or new information (the incoming envelopes) more highly. Thus if α is large (close to unity), the algorithm tends to ''lock on'' to a beat, and follow that tempo regardless of the new envelope information. On the other hand, if α is small, the beat-track can be easily perturbed by changes in the periodicity of the incoming signal. Manipulating these parameters for the comb filter structure is computationally similar to manipulating the windowing function of a narrowed autocorrelation.

Higher-level or domain-specific knowledge could be used to set this parameter based on previous information. For example, in rock or pop music, the beat is usually quite steady, so a high value for α would be appropriate; while for classical music, particularly styles including many tempo changes, a smaller value would be more optimal.

IV. VALIDATION

It is somewhat of a difficult proposition to evaluate the construction of an ecological beat-tracking model, for there are few results in the literature dealing with listeners' tempo responses to actual musical excerpts. Most psychophysical research has dealt primarily with special cases consisting of simple tones in unusual temporal relationships, which will typically be more difficult to track than ''real music'' for a listener. Conversely, most beat-tracking systems have been

TABLE I. Performance of the beat-tracking algorithm, summarized by musical genre. Results were auditioned and classified into groups by qualitative success level. ‘‘Urban’’ styles include rap, funk, and R & B music; ‘‘Quiet’’ includes muzak and an ‘‘easy-listening’’ example. All sounds are available via the WWW.

Genre	No. of cases	Correct	Partial	Wrong
Rock	17	13	3	1
Country	3	3	0	0
Urban	9	7	1	1
Latin	5	3	2	0
Classical	9	4	4	1
Jazz	8	3	1	4
Quiet	3	2	0	1
Reggae	2	2	0	0
Non-Western	4	4	0	0
Total	60	41	11	8

evaluated intuitively, by using a small number of test cases (whether acoustic or MIDI-based) and checking that the algorithm ‘‘works right.’’

In this section, the performance of the algorithm is evaluated in both qualitative and quantitative manners. Results are provided on the qualitative performance for 60 ecological music excerpts, with sound examples publicly available for listening. Results are also provided from a short validation pilot experiment which was conducted to confirm that the performance of the algorithm is like the performance of human listeners.

A. Qualitative performance

Examples of many different types of music have been tested with the implemented algorithm, using a short application which reads a sound sample off of disk, causally beat-tracks it, and writes a new sound file with clicks (short noise bursts) added to the signal where beats are predicted to occur. A selection of these sound files is available for listening via the World Wide Web (‘‘results’’ page), and the results are summarized below. The wide set of input data contains 60 examples, each 15 s long, of a number of different musical genres. Rock, jazz, funk, reggae, classical, ‘‘easy-listening,’’ dance, and various non-Western music are represented in the data set and can be tracked properly. Some of the examples have drums, some do not; some have vocals, some do not. Five of the examples would be judged by human listeners to have no ‘‘beat.’’ Table I summarizes the results by musical genre, and some qualitative descriptions of typical results are provided below.

Forty-one of 60 samples (68%) have been qualitatively classified as being tracked accurately, and another 11 (18%) as being tracked somewhat accurately. This accuracy percentage is not directly comparable to that reported for other systems, because the data set used here is more difficult. All of the ‘‘easy’’ cases of rock-and-roll with drums keeping a straightforward beat were tracked correctly; and five of the eight examples not tracked accurately are said by human listeners to have no ‘‘beat’’ to begin with. It is premature to interpret these results as indicative of consistent genre-to-genre differences in accuracy; there are too few examples and the within-genre differences in accuracy too great.

For the cases which track correctly, there is a startup period between 2 and 8 s long during which the resonant filters have not yet built up an accurate picture of the signal. After this period, for most signals, the algorithm has settled down and begun to track the signal accurately, placing the clicks in the same locations a human listener would. Examining some of the other, incorrectly tracked examples, is instructive and highlights some of the deficiencies of this method.

Examples #1, #2, and #57 are all up-tempo jazz cases in which human listeners do perceive a strong beat, but no beat is ever extracted by the system. In these three cases, the beat is described by syncopated instrumental lines and complex drum patterns. That is, there is not actually very much energy modulating at the frequency which is the perceptual beat tempo for humans. Human listeners have a great ability to induce ‘‘apparent’’ frequencies from complicated modulation sequences. For these examples, the algorithm is not able to find a pulse frequency, and so the beat output is more-or-less random.

The same is apparent in example #37, which is a pop tune that has a ‘‘mixed’’ or ‘‘clave’’ beat—the beat is not even, but subdivided into oddly spaced groups. Each two measures, containing 16 eighth notes between them, are divided into a 3-3-3-3-2-2 pattern. A human listener has no trouble understanding the relationship between this pattern and a more common 4-4-4-4 pattern, but the algorithm seems to assume that the groups of three are the basic beat, and then get confused when the pattern doesn’t come out right.

Among the examples judged as being tracked with some accuracy, but not entirely correctly, the most common problem is phase shifting. For example, in example #16, a jazz piano trio, the beat estimate is correct on the frequency, but switches back and forth between assigning beats to the ‘‘upbeat’’ or the ‘‘downbeat.’’ Although this behavior is not unlike some human jazz listeners, a human would likely be more consistent in deciding where to place the beat. This behavior could be easily corrected by adding a small amount of high-level knowledge to the beat-tracking system.

Similar to this, in example #7, a rhythm and blues tune, the algorithm is uncertain about assigning the beat to the quarter-note pulse or to the eighth-note pulse, and so switches back and forth between them. A human listener might also suffer from similar confusion, but would likely make an arbitrary decision and then stay with it unless the music changed radically.

Other than these two sorts of confusions for certain rhythmically complex musics, the algorithm seems to perform quite successfully at tracking the musical beats.

1. Tempo modulation

As Todd correctly points out (Todd, 1994), to be an accurate model of human rhythm perception (and, of course, to be maximally useful as a music analysis tool), a beat-

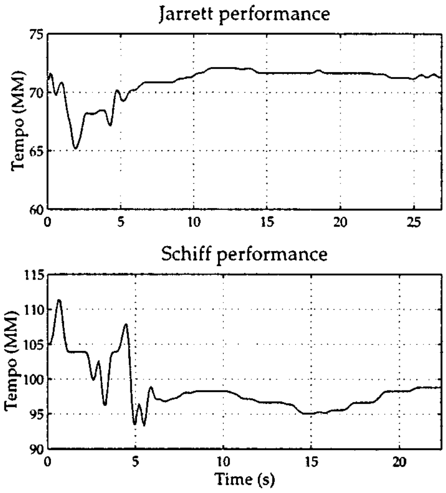

FIG. 8. ‘‘Tempo curve’’ for two performances of the same piece of music. Each tempo track has a short startup period during which the tempo estimation is unstable; after that there are clear differences in the two performances. The timescales are slightly different to make the performance scales align (the same musical excerpt is used in both cases).

tracking system must be robust under expressive tempo modulation. The algorithm described here is able to follow many types of tempo modulations; this is effected in the signal processing network by simply examining, over time, the resonator producing the most energetic output. That is, when the tempo of a signal modulates, the response of the resonator corresponding to the old tempo will die away, and that of the resonator corresponding to the new tempo will gain.

Figure 8 shows ‘‘tempo curves’’ (Desain and Honing, 1992) for two expressively modulated performances of a piece of music (Keith Jarrett and Andras Schiff performances, of the beginning of the G-minor fugue from book I of Bach’s Bach’s *Well-Tempered Clavier* [sound example 3]). The algorithm is quite sensitive to the variations in tempo over time.

B. Validation experiment

A short validation experiment has been conducted to confirm the qualitative results given in the previous section. This experiment was not intended to highlight important psychoacoustic effects in beat perception, but only to test whether the beat-tracking algorithm performs generally like a human listener.

1. Subjects

Five adult listeners, all graduate students and staff members at the MIT Media Laboratory, participated in the experiment. All were experienced musicians with normal hearing.

2. Overview of procedure

Subjects listened to seven musical examples, drawn from different musical genres, through headphones. They indicated their understanding of the beat in the music by tapping along with the music on a computer keyboard.

3. Materials

Seven musical excerpts from the above set were used. Each was digitally sampled from an FM radio tuner to produce a monophonic 22-KHz sound file, 15 s long. A computer interface was created on a DEC Alpha workstation with which the musical excerpts were presented to subjects at a comfortable listening level over AKG-K240M headphones.

The musical excerpts were as follows: a Latin-pop song at moderately fast tempo (#10), a jazz piano trio at fast tempo (#17), a ‘‘classic rock’’ song at moderately slow tempo (#20), an excerpt from a Mozart symphony at moderate tempo (#40), an ‘‘alternative rock’’ song at moderately slow tempo (#45), and a piano etude with varying tempo (#56).

A click track ‘‘step function’’ was also created for the experiment, in which 10-ms white noise bursts were presented at a tempo of 120 bpm (interonset time of 500 ms) for 6 s, then at a tempo of 144 bpm (interonset time of 417 ms) for 4.6 s, then again at 120 bpm for 6 more s. This stimulus is used to evaluate the response of human listeners and the beat-tracking algorithm to sudden changes in tempo.

A musical expert (the author) assigned exact beat times to each excerpt by listening repeatedly and placing ‘‘click’’ sounds in the perceptually appropriate positions. This task was different than the tapping task in which the subjects participated; the expert listened repeatedly to each stimulus, placing beats, listening to results, and adjusting the beat position if necessary. It is considered to be more accurate and robust than the real-time tapping task, although there is little literature on humans performing either of these sorts of judgments [see Drake *et al.* (1997) and Parncutt (1994) for two other ‘‘tapping tasks’’]. The expert labeling was conducted separately from the tapping experiment, the expert did not know the results of the experiment or the algorithm execution, and the subjects were not presented with the expert data. The resulting ‘‘ground truth’’ beat times are used for the evaluation of results, below.

4. Detailed procedure

Subjects were seated in front of the computer terminal and instructed in the task: they were to listen to short musical examples and tap along with them using the space bar on the keyboard. They were instructed to tap at whatever tempo felt appropriate to the musical excerpt, but to attempt to tap in equal intervals (a pilot experiment revealed that some subjects like to ‘‘drum along’’ in rhythmic or even syncopated patterns with the music if they are not instructed otherwise). They listened to a 120-bpm click-track as a training sample to indicate they understood the procedure, and then proceeded with each of the seven experimental trials.

All seven trials were run in the same sequence for each listener, in a single block. The experiment was not counterbalanced based on an assumption that there is little training effect in this task. After each trial, the subject was instructed by the interface to press a key different than the space bar to continue to the next trial. The entire experiment took approximately 5 min per subject. The computer interface recorded the time of each tap, accurate to approximately 10 ms, and saved the times to a disk file for analysis.

Finally, the beat-tracking algorithm was executed on each of these seven stimuli to produce beat times as estimated by the model described in the previous sections. These beat times were saved to a disk file and analyzed for comparison with the human beat times. The algorithm parameters were adjusted to give optimum performance for this set of trials, but not changed from trial-to-trial.

5. Dependent measures

The human and algorithmic beat-tracks were analyzed in two ways. First, the beat placements were compared to the ideal placements as judged by the expert listener; then, the regularity of tapping was assessed by examining the variance of interonset times.

To compare the beat placements, a matching comparison was conducted. Each beat placed by a human subject or by the beat-tracking model was matched with the closest (in time) comparison beat in the expert beat-track. Initially, only the beats actually placed by the expert were used, but since some subjects and the algorithm tapped twice as fast as the expert on some examples, beats were allowed to be matched to the midpoint between expert beats. The root-mean-square deviations of the subject's taps from the expert's taps were collected for each subject and trial, averaging across taps within a trial.

This rms deviation is a measure of how close the tapper came to the ''ideal'' beat locations. If it is very low, all of the tapper's placements were very close to expert judgments; if high, the tapper's placements were randomly distributed compared to the expert judgments.

This measure leaves open an important aspect of beat-tracking, which is regularity. As described in the qualitative results, the algorithm sometimes demonstrates unusual behavior by switching from one tempo to another, or from off-the-beat to on-the-beat, in the middle of a trial. To evaluate the regularity of tapping, the variance of interonset interval was calculated for each trial-by-subject, each trial by the model, and each trial by the expert. Note that, as described above, the human subjects were explicitly encouraged to tap regularly.

Again, the expert's behavior is taken as ideal; if the variance is larger for some tapper than for the expert, it indicates that the tapping was irregular relative to the expert. If the variance is smaller, it indicates that the tapping was more regular than the expert (not necessarily a positive aspect in the case of changing tempi). Irregularity generally arises in this data from leaving out beats, each occurrence of which adds an inter-onset interval twice as large as the rest, increasing the variance.

6. Results and discussion

The beat-placement comparison is shown in Fig. 9. Results indicate that the performance of the algorithm in placing beats in logical locations was at least comparable to the human subjects tested for all the musical cases; in four of the seven cases, the model was the most or second-most accurate tapper. This indicates that whenever a beat position was chosen by the algorithm, the position was very close to the ideal beat position as determined by the expert judgment.

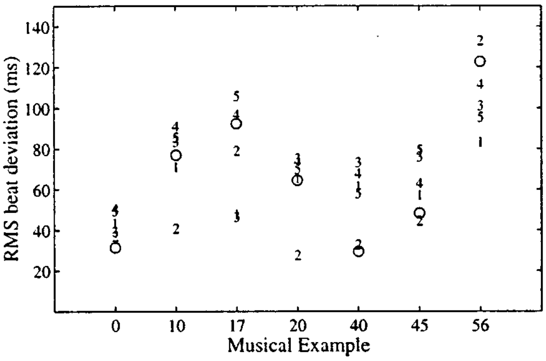

FIG. 9. Scatter plot of human (subj. number) and model (O) beat position accuracy for each of the seven experimental trials. Trial '0' corresponds to the click-track step function. Each point measures how accurate that subject was, relative to the expert, in placing beats in time. The expert judgments are at zero variance for each column. For each trial, the algorithm beat position was at least comparable to the performance of the human subjects. Overall, the algorithm performance showed a highly significant positive correlation with the human subject performance [$r=0.814$; $p(\mathrm{df}=5) < 0.015$].

The regularity comparison is shown in Fig. 10. Results here indicate that the algorithm was as regular as a human listener for five of the seven trials, and less consistent for two of the trials. In one case, it and several of the human subjects were more consistent than the expert. More *post hoc* analysis is necessary to understand why the algorithm performance is irregular in these trials; preliminary results suggest that these two stimuli have relatively slow onsets carrying the beat (violins in one case, electronically gated drum sounds in the other).

These two results are consistent with the qualitative re-

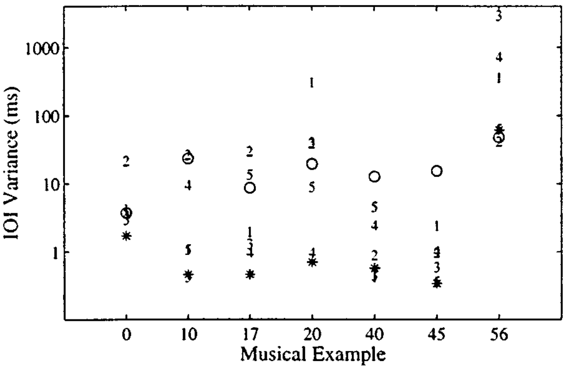

FIG. 10. Scatter plot of human (subj. number), model (O), and expert (*) IOI variances for each of the seven experimental trials. Trial ''0'' corresponds to the click-track step function. Each point shows the regularity of tapping of a subject for one trial; large values represent less regular tapping. For trials #40 and #45, the algorithm was not as consistent in tapping as a human listener. Overall, the algorithm performance showed a highly significant positive correlation with the human subject performance, and both the algorithm and the human subjects showed highly significant positive correlations with the expert judgement [$r=0.889$, $r=0.863$, $r=0.995$, respectively; $p(\mathrm{df}=5) < 0.01$ in each case].

sults described above. When the algorithm chooses to place a beat, it does so with great accuracy and musical relevance; however, for certain musical excerpts, it is somewhat inconsistent in its tapping regularity. That is, for these examples, it drops beats or shifts phase more often than a human listener. This is not a bad result, because it is exactly this inconsistency which could best be addressed by including high-level information in the model (such as simply including instructions to ''try to tap regularly'').

V. DISCUSSION

In previous sections, the construction of a beat-tracking system has been approached from a largely empirical perspective. However, it is also valuable to compare the resulting algorithm to previous work on pulse perception in humans.

A. Processing level

Perhaps the most obvious difference between the method presented here and much of the previous work on beat-tracking is that this algorithm knows almost nothing about musical timbre, genres, or even notes or onsets. This approach to tempo analysis might be called a ''perceptual model'' of tempo, to contrast it with cognitive structuralist models.

That is to say, in models such as Povel and Essens (1985), Desain (1995), or Goto (in press), there are two stages of processing represented (the first is implicit in the Povel/Essen and Desain models). The first stage processes the acoustic stream, classifying the various pieces of sound into onsets and time intervals, separating the streams of sound, and understanding the accent structure and timbre of various components. Then, the second stage places these events in relationship to each other in order to determine the tempo and phase of the signal.

In contrast to this, the model presented here agrees with the viewpoint of Todd (1994), in which tempo and rhythm are low-level ''perceptual judgments'' about sound, with little cognition or memory required for processing. This viewpoint is intuitively appealing for at least one major reason, which is that certain features of tempo and beat are processed in non-attended auditory streams. Music listeners, even nonmusicians, often have the experience of conducting a conversation and suddenly realizing that they have been tapping their foot to background music. If the foot-tapping process requires cognitive structuring of the input data, it seems likely that other cognitive hearing tasks such as speech-understanding would interfere.

The finding of Levitin and Cook (1996) that there is a great ability for listeners to learn and remember absolute musical tempo implies that tempo is a simple, low-level perceptual quality. The body of initial work on rhythm perception in non-human animals (for example, Hulse *et al.*, 1984) would seem to imply similar conclusions.

The resemblance between the algorithm as drawn in Fig. 3 and modern models of pitch hearing is striking. Both models contain frequency-decomposition front ends followed by temporal integration. This comparison is explored in depth in Scheirer (1997) and leads to the question of whether pitch and tempo perception might be related auditory phenomena.

Studies such as that of Povel and Essens (1985) have demonstrated convincingly that beat perception may be explained with a model in which a perceptual clock is aligned with the accent structure of the input. A clock model is fully compatible with the method proposed here; it seems natural and intuitive to posit such an internal clock. However, the Povel and Essens model of clock induction, and similarly the Parncutt model, relies heavily on structural qualities of the input, such as a sophisticated model of temporal accent, to function.

Todd has argued that such phenomena do not need to be modeled cognitively, but rather can be explained as natural emergent qualities of known psychoacoustic properties of masking and temporal integration. This model agrees here as well, for it has demonstrated empirically that musical signals can be accurately beat-tracked without any such factors explicitly taken into account. However, a more thorough evaluation of this model would include testing it on the unusual and difficult sequences tested in the course of developing accent models, to determine if changes to weighting factors or integration constants need to be made in order to replicate these psychophysical effects.

B. Prediction and retrospection

Desain's recent work on beat-tracking has included valuable discussion of the role of prediction and retrospection in rhythmic understanding. Clearly, prediction is a crucial factor in an accurate model of human rhythm perception, as simply to synchronize motor motion (like foot-tapping) with an auditory stream requires prediction. There is a pleasing symmetry between Desain's ''complex expectancy'' curves and the phase-prediction vectors extracted here from the comb filter delay lines (as in Fig. 7).

Desain, citing Jones and Boltz (1989), draws attention to the utility of considering prediction and retrospection to be similar aspects of a single process. ''Retrospection'' refers to the manner in which new stimulus material affects the memory of previous events. Although there is no retrospection included in the model—remembrance would seem to be an inherently cognitive process—the phase-prediction curves could be used as input for this process as well.

When evaluating this model, it is important to keep in mind the complexity of introspection on musical phenomena. Although after-the-fact, listeners have made a rhythmic model of the very beginning of a musical phrase, it is clear that this model must have arisen via retrospection, for there is not enough information in the signal alone to form it progressively. Simply because a listener feels that he ''understands'' the rhythm of the beginning of a musical segment does not mean that the beginning itself contains sufficient information to allow such understanding.

C. Tempo versus rhythm

The effects which are not explained with this model are those related to grouping of stimuli into a rhythmic hierarchy. There are many known effects in this area, ranging from the low-level, such as Povel and Okkerman's work on perceived accents in nonaccented sequences (Povel and Okkerman, 1981) to very broad theories of generative rhythmic modeling such as the well-known Lerdahl and Jackendoff work (Lerdahl and Jackendoff, 1983).

This model is compatible with and complementary to the bulk of this research, since most of the theories assume that a temporal framework has already been created. Synthesis of a model which operates from an acoustic source and one which includes musical assumptions and explanation should be possible, and would then represent a very robust theory of rhythmic understanding.

However, the model presented here should not be taken as attempting to explain rhythm perception as well as tempo; the viewpoint is rather that these processes are to some extent separable and may be addressed and modeled independently.

VI. CONCLUSION AND FUTURE WORK

An algorithm has been described which can successfully beat-track digital audio representing music of many different types. The music does not have to contain drums or any other specific timbres, and it does not have to conform to any predetermined set of musical templates. The beat-tracking procedure can be run in real-time on an advanced desktop workstation.

There are still aspects of the algorithm which are inadequately tested and understood. For example, would it be equally accurate but more efficient with a different filterbank, or could it be made more accurate in this way? What would be the implications of using a different temporal integration function, with different or more psychoacoustically accurate properties? What about using an entirely perceptually motived front end? These questions are unfortunately still unaddressed.

Errors still made by the algorithm are typically due to the inability to understand beat relationships at various tempi; that is, a human listener intuitively understands the way eighth-note patterns group to form quarter-note and half-note patterns, and while some processing of this sort is done implicitly in the resonators due to phase-locking at harmonic ratios, it would clearly make the algorithm more robust to have an explicit model of this sort of rhythmic grouping.

Perhaps the way to build a system that can track complicated beat patterns is to construct it in two layers. The lower layer would be a simple perceptual beat extraction system as described here, which finds the level at which the pulse is evenly divided in time. Then, a higher-level grouping model selects and processes the beats to form an model of the rhythmic hierarchy present in the signal, based on pattern-recognition detection of accent structures and instrumental beat patterns. Building a system in this manner would allow us to leverage much of the existing work in cognitive rhythm models to apply to the analysis of digital audio as well as symbolically represented music.

ACKNOWLEDGMENTS

Thanks are due the Interval Research Corporation for their support of this work. Also, Masataka Goto of Waseda University provided valuable discussions on rhythm-tracking and research on which to build. Dan Levitin provided the suggestion to base experimental validation on an expert listener. The suggestions of two anonymous reviewers improved the scope and quality of the paper immensely. As always, Professor Barry Vercoe and the Machine Listening Group at the Media Lab, especially Bill Gardner and Dan Ellis (now at ICSI-Berkeley), have been essential through their comments, critiques, and discussions.

Brown, J. C. (**1993**). ‘‘Determination of the meter of musical scores by autocorrelation,’’ J. Acoust. Soc. Am. **94**, 1953–1957.

Desain, P. (**1995**). ‘‘A (de)composable theory of rhythm perception,’’ Music Perception **9**, 439–454.

Desain, P., and Honing, H. (**1992**). *Music, Mind, and Machine: Studies in Computer Music, Music Cognition, and Artificial Intelligence* (Thesis Publishers, Amsterdam).

Drake, C., Penel, A., Bigand, E., and Stefan, L. (**1997**). ‘‘Tapping in time with musical and mechanical sequences,’’ in *Proceedings of the 1997 European Society for Cognition of Music*.

Goto, M., and Muraoka, Y. (**in press**). ‘‘Music understanding at the beat level: Real-time beat tracking for audio signals,’’ in *Readings in Computational Auditory Scene Analysis*, edited by D. Rosenthal and H. Okuno (Erlbaum, Mahwah, NJ).

Handel, S. (**1989**). *Listening* (MIT, Cambridge, MA).

Hulse, S. H., Humpal, J., and Cynx, J. (**1984**). ‘‘Discrimination and generalization of rhythmic and arrhythmic sound patterns by European starlings (*Sturnus vulgaris*),’’ Music Perception **1**, 442–464.

Jones, M. R., and Boltz, M. (**1989**). ‘‘Dynamic attending and responses to time,’’ Psychol. Rev. **96**, 459–491.

Large, E., and Kolen, J. F. (**1994**). ‘‘Resonance and the perception of musical meter,’’ Connection Science **6**, 177–208.

Lerdahl, F., and Jackendoff, R. (**1983**). *A Generative Theory of Tonal Music* (MIT, Cambridge, MA).

Levitin, D. J., and Cook, P. R. (**1996**). ‘‘Memory for musical tempo: Additional evidence that auditory memory is absolute,’’ Percept. Psychophys. **58**, 927–935.

Longuet-Higgens, H. C., and Lee, C. S. (**1984**). ‘‘The rhythmic interpretation of monophonic music,’’ Music Perception **1**, 424–441.

Martin, K. (**1996**). ‘‘Automatic transcription of simple polyphonic music,’’ J. Acoust. Soc. Am. **100**, 2813A.

Parncutt, R. (**1994**). ‘‘A perceptual model of pulse salience and metrical accent in musical rhythms,’’ Music Perception **11**, 409–464.

Povel, D.-J., and Essens, P. (**1985**). ‘‘Perception of temporal patterns,’’ Music Perception **2**, 411–440.

Povel, D.-J., and Okkenman, H. (**1981**). ‘‘Accents in equitone sequences,’’ Percept. Psychophys. **30**, 565–572.

Rosenthal, D. (**1992**). ‘‘Machine rhythm: Computer emulation of human rhythm perception,’’ Ph.D. thesis, MIT Media Laboratory.

Scheirer, E. (**1996**). ‘‘Bregman's chimerae: Music perception as auditory scene analysis,’’ in *Proceedings of the 1996 International Conference on Music Perception and Cognition*, Montreal.

Scheirer, E. (**1997**). ‘‘Pulse tracking with a pitch tracker,’’ in *Proceedings of the 1997 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Mohonk, NY.

Scheirer, E. (**in press**). ‘‘Using musical knowledge to extract expressive performance from audio recordings,’’ in *Readings in Computational Auditory Scene Analysis*, edited by H. Okuno and D. Rosenthal (Erlbaum, Mahwah, NJ).

Shannon, R. V., Zeng, F.-G., Wyngoski, J., Kamath, V., and Ekelid, M. (**1995**). ‘‘Speech recognition with primarily temporal cues,’’ Science **270**, 303–304.

Smith, L. S. (**1994**). ‘‘Sound segmentation using onsets and offsets,’’ J. New Music Res. **23**, 11–23.

Todd, N. P. McA. (**1994**), ‘‘The auditory ‘primal sketch’: A multiscale model of rhythmic grouping,’’ J. New Music Res. **23**, 25–70.

Vercoe, B. L. (**1994**). ‘‘Perceptually-based music pattern recognition and response,’’ in *Proceedings of the 1994 International Conference on Music Perception and Cognition*.