

Beat Tracking of Musical Performances Using Low-Level Audio Features

William A. Sethares, Robin D. Morris, and James C. Sethares

Abstract—This paper presents and compares two methods of tracking the beat in musical performances, one based on a Bayesian decision framework and the other a gradient strategy. The techniques can be applied directly to a digitized performance (i.e., a soundfile) and do not require a musical score or a MIDI transcription. In both cases, the raw audio is first processed into a collection of “rhythm tracks” which represent the time evolution of various low-level features. The Bayesian approach chooses a set of parameters that represent the beat by modeling the rhythm tracks as a concatenation of random variables with a patterned structure of variances. The output of the estimator is a trio of parameters that represent the interval between beats, its change (derivative), and the position of the starting beat. Recursive (and potentially real time) approximations to the method are derived using particle filters, and their behavior is investigated via simulation on a variety of musical sources. The simpler method, which performs a gradient descent over a window of beats, tends to converge more slowly and to undulate about the desired answer. Several examples are presented that highlight both the strengths and weaknesses of the approaches.

Index Terms—Author, please supply your own keywords or send a blank e-mail to keywords@ieee.org to receive a list of suggested keywords..

I. INTRODUCTION

A COMMON human response to music is to “tap the foot” to the beat, to sway to the pulse, to wave the hands in time with the music. Underlying such mundane motions is an act of cognition that is not easily reproduced in a computer program or automated by machine. This beat tracking problem is important as a step in understanding how people process temporal information and has applications in the editing of audio/video data [5], in synchronization of visuals with audio, in audio information retrieval [33], in audio segmentation [38], and in error concealment [36].

Underlying the beat tracking algorithms is a method of data reduction that creates a collection of “rhythm tracks” which are intended to represent the rhythmic structure of the piece. Each track uses a different method of (pre)processing the audio by extracting different low-level audio features, and so provides a (somewhat) independent representation of the beat. The rhythm

tracks are described in detail in Section III and they may be modeled as a collection of normal random variables with changing variances: the variance is small when “between” the beats and large when “on” the beat.

The first algorithm for beat tracking exploits this simple stochastic model of the rhythm tracks using Bayesian methods as in Section IV-A. The tracks are examined to find the best \mathcal{T} (interval between successive beats), $\delta\mathcal{T}$ (rate of change of the interval), and τ (starting point or phase). Since each track represents a different realization of the underlying process (the audio), the technique attempts to combine the tracks to obtain optimal estimates. Using the “particle filter methods” of [12], Section IV-B demonstrates a recursive version that operates over successive blocks using the output distribution at one block as the prior distribution (initialization) of the next.

Because the Bayesian framework can be computationally complex, Section V explores a second algorithm for beat tracking that defines a cost function and invokes adaptive (gradient) methods for approximating the optimal values of \mathcal{T} and τ . The behavior of the resulting algorithms is explored in Section VI which compares the strengths and weaknesses of the two methods. The final section concludes the paper.

II. LITERATURE REVIEW

Listeners can easily identify complex periodicities such as the rhythms that normally occur in musical performances, even though these periodicities may be distributed over several interleaved time scales. The simplest such activity is to identify the pulse or beat of the music; yet even this is not easy to automate. An overview of the problem and a taxonomy of beat tracking methods can be found in [25], and a review of computational approaches for the modeling of rhythm is given in [16].

Traditional attempts to identify the metric structure of musical pieces such as [6], [23] and [30] often begin with a symbolic representation of the music: a musical score or a MIDI file transcription. This simplifies the rhythmic analysis since the pulse is inherent in the score, note onsets are clearly delineated, multiple voices cannot interact in unexpected ways, and the total amount of data to be analyzed is small compared to a CD-rate audio sampling of a performance of the same piece. The beat tracking of MIDI files has been explored extensively using: the (AI) beam search method [1], gradient methods [7], sets of competing oscillators [32], and probabilistic methods such as the Kalman filter [3], MCMC methods and particle filters [4], and a Bayesian belief network [22].

Many methods that deal directly with audio (such as [7], [25]) begin by locating interonset intervals (IOIs) using amplitude or energy profiles. When the location of IOIs is successful, the beat

Manuscript received March 11, 2003; revised January 7, 2004. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Michael Davies.

W. A. Sethares is with the Department of Electrical and Computer Engineering, University of Wisconsin-Madison, Madison, WI 53706-1691 USA.

R. D. Morris is with the RIACS, NASA Ames Research Center, Moffett Field, CA 94035-1000 USA.

J. C. Sethares is at 75 Lake Street, Cotuit MA.

Digital Object Identifier 10.1109/TSA.2004.841053

tracking can proceed analogously to when the input is a MIDI file. The pulse finding technique of [13] tracks the temporal position of quarter notes, half notes and measures, and incorporates a multi-agent expert system. Each agent predicts (using a windowed autocorrelation) when the next beat will occur based on onset time vectors that are derived from a frequency decomposition. Other methods [37] exploit some feature of the music such as the low-frequency bass and drum of much popular music.

Many psychoacoustically based models of the auditory system begin with a subband decomposition or bank of filters that divide the sound into a number of frequency regions. Such decompositions can be used as a first step in beat tracking, as shown by the spectrogram-like method of [29], the wavelet approach of [33]. Scheirer [26] argues that rhythmically important events are due to periodic fluctuations of the energy within various frequency bands and creates a beat tracking algorithm that uses a collection of comb filters to determine the frequency/period of the dominant beat. Similarly, [14] follows a subband decomposition with an autocorrelation method for periodicity determination.

III. CREATING “RHYTHM TRACKS”

There are many possible models for the pulse of a piece of music. One commonplace observation is that a large proportion of the most “significant audio events” occur on or near the beat, while events which are less significant rhythmically tend to occupy the space between the beats. One way to transform this observation into a concrete model is to suppose that the rhythm can be encoded into a sequence of random variables: times near the beat are highly likely to have significant energy (the random variables will have a large variance) while times between beats are likely to be unenergetic (the random variables will have small variance). We call such a model a *rhythm track* corresponding to a particular musical performance.

The Sections III-A–D provide four different methods of exploiting low-level audio features for the creation of rhythm tracks.

- 1) A time-domain energy method r_E
- 2) A frequency-domain method based on group delay r_{t_g}
- 3) A measure based on the center of the spectrum r_{SC}
- 4) A measure of the spectral dispersion r_{SD} .

Each of these provides a different meaning to the phrase “significant audio event.” This approach of utilizing low-level audio features is perhaps most similar to that of [15] which focuses on the problem of distinguishing duple from triple meter.

Since the periodicities associated with musical pulse occur on a time scale between tenths of a second and a couple of seconds, the standard audio sampling rate of 44.1 kHz contains significant redundancies. The rhythm tracks are formed using overlapping (Hamming) windows (and applying the fast Fourier transform (FFT) for methods 2, 3, and 4). The data in each window is reduced to a single summary statistic, and so the rhythm tracks reduce the amount of data by a factor of 100 to 1000, depending on the window size and amount of overlap. Many other methods of creating rhythm tracks are possible: using other norms or distance measures, applying filters to the signals, etc. These four

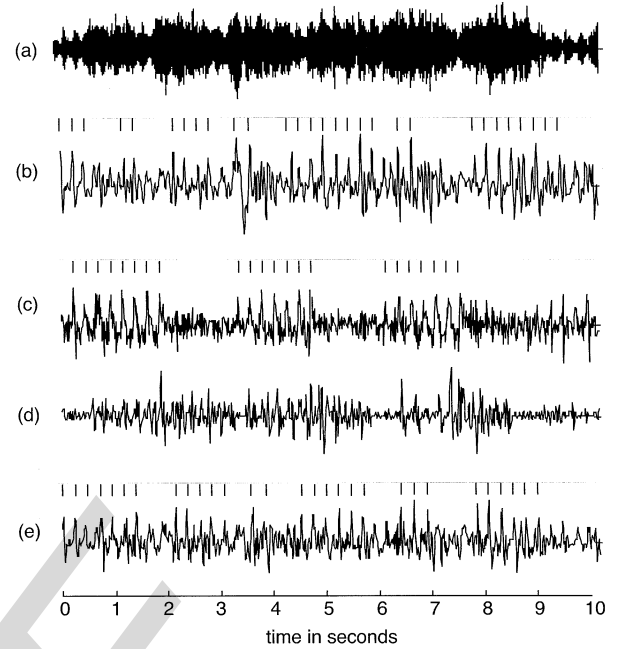


Fig. 1. Example of the various “rhythm tracks” applied to the first 10 s of a recording of Handel’s *Water Music*. (a) Audio waveform. (b) Energy method. (c) Group delay. (d) Change in the center of the spectrum. (e) Dispersion of spectral energy. Tick marks emphasize beat locations that are visually prominent.

are highlighted because they are easy to compute, convenient to describe, and each has some intuitive relevance to the task at hand.

An example of the various rhythm tracks is shown in Fig. 1, which compares the first 10 s of a recording of Handel’s *Water Music* with its various rhythm tracks. The audio waveform shown in part Fig. 1(a) contains 440 K data points (44.1 kHz times 10 s). Each of the rhythm tracks in Fig. 1(b)–(e) contains about 800 points (a window size of 2^{10} with an overlap of 2). The beats of the piece can be seen quite clearly in certain places in certain of the rhythm tracks, and these are annotated using tick marks. For instance, three strings of beats occur in Fig. 1(c): in the first 2 s, between 3.5 and 5 s, and between 6.5 and 8 s. Rhythm track Fig. 1(b) has regular spikes between 4 and 6 s, and again between 8 and 9.5 s. For this particular example, a combination of Fig. 1(b), (c), and (e) shows almost all the beats throughout the 10 s. In other examples, other combinations of the rhythm tracks may be more useful.

A. Energy Measure

The simplest of the rhythm tracks is particularly appropriate for audio in which the envelope of the sound clearly displays the beat. Let $x(t)$ represent the audio waveform, which is sampled at a constant interval to give the sequence $x[k]$. Group the sampled data into M overlapping segments each containing N consecutive terms. Let $x_n[k]$ represent the k th element (out of N) in the n th segment. The energy in the n th segment is

$$e[n] = \sum_{k=0}^{N-1} x_n^2[k], \quad n = 1, 2, \dots, M. \quad (1)$$

Then the terms of the “energy” rhythm track $r_E[n]$ are defined to be the change in (the derivative of) the $e[n]$. Though numerical

derivatives can be poorly conditioned, the action of the summing, combined with sensible overlapping ensures that the numerical problems do not overwhelm the data. An example is provided in Fig. 1(b).

B. Group Delay

The remaining methods operate in the frequency domain and share a common notation. With $x(t)$, $x_n[k]$, N , M as above, let $X_n[j]$ be the FFT of x_n . Each of the frequency-domain methods processes X_n in a different way to form the scalar value $r[n]$, and the sequence of such values (one for each set) forms the rhythm track sequence.

Structurally, the transform X_n consists of N complex numbers that are most commonly represented as magnitude and phase pairs, with the phase unwrapped (meaning that factors of 2π are added or subtracted so as to make the phase angle continuous across boundaries at integer multiples of $\pm\pi$). For many musical waveforms, the unwrapped phase lies close to a straight line. The slope t_g of this line defines the ‘‘group delay’’ method of creating rhythm tracks $r_{t_g}[n]$, which represents the slope of the unwrapped phase of the n th segment. An example is provided in Fig. 1(c). Appendix A shows how the slope t_g is proportional to a time-shifted version of the energy. It is not dependent on the total energy in the window, but rather on the distribution of the energy within the window.

C. Spectral Center

With notation inherited from the previous sections, the ‘‘spectral center’’ method of creating rhythm tracks locates the frequency f_c where half of the energy in the spectrum lies below f_c and half lies above. This is

$$sc[n] = \arg \min_{f_c} \left| \sum_{j=0}^{f_c} X_n^2[j] - \sum_{j=f_c}^{N-1} X_n^2[j] \right|. \quad (2)$$

The rhythm track value $r_{SC}[n]$ is then defined as the change in (i.e., the derivative of) $sc[n]$. The spectral center is sensitive to pitch changes and to changes in the distribution of energy such as might occur when different instruments enter or leave, or when one instrument changes registers. Like the group delay, it is insensitive to amplitude changes in the audio. A numerical example is provided in Fig. 1(d).

D. Spectral Dispersion

The spectral dispersion gives a measure of the spread of the spectrum about its center. Let

$$sd[n] = \sum_{j=0}^{N-1} X_n^2[j] |j - f_c| \quad (3)$$

define the spectral dispersion $sd[n]$ of the n th segment about the spectral center f_c . It weights energy at remote frequencies more than those close to the spectral center. The rhythm track $r_{SD}[n]$ is then defined as the change in (the derivative of) $sd[n]$. This provides a crude measure of how the spectral energy is distributed: small values mean that the energy is primarily concen-

trated near the center while large values mean that the energy is widely dispersed. For example, near the percussive attack of a violin the spectral dispersion is large, while it is small in the (relative) steady state between attacks. An example is provided in Fig. 1(e).

These are just four of the many possible low-level audio features that could be used in the creation of rhythm tracks. We have chosen to focus on these four because they appear to work well in the appointed task of beat tracking. While these four do not enjoy statistical independence, it is easy to see that they measure different features of the underlying audio stream since it is possible to create a sound for which any three of the rhythm tracks are (essentially) constant, but the fourth varies significantly. For example, an idealized trill on a violin has constant energy, constant dispersion, and constant group delay, but varying center. Similarly, if a short sinewave burst alternates with a white noise burst, they can be chosen so that the energy, group delay, and center remain the same but the dispersion varies widely. This is the sense in which the rhythm tracks provide an ‘‘independent’’ measure of the changes in the sound.

IV. BAYESIAN BEAT DETECTION

This section describes a principled approach to detecting the beats from the rhythm tracks. The approach begins by constructing a simple (and likely over-simplified) generative model of the probabilistic structure of the rhythm tracks in Section IV-A. The parameters of this model can be followed through time using a particle filter [8], [12], [34], as detailed in Section IV-B. The framework allows seamless and consistent integration of the information from multiple rhythm tracks into a single estimate of the beat timing.

A. A Model for the Rhythm Tracks

Inspection of the rhythm tracks in Fig. 1 reveals that, to a first approximation, they are composed primarily of large values at (or near) the beats and small values off the beat. The simplest possible model for the structure of the rhythm tracks is thus to say that they are made up of realizations of an independent Gaussian noise process, where the variance of the noise on the beat is larger than the variance off the beat.

Clearly this ignores much of the structure that is present. For example, the time-domain energy method often shows larger positive peaks compared with the negative ones. The tracks also often display oscillatory behavior, mirroring the intuitively obvious idea that the samples cannot be truly independent. However, the simple model is shown in the experiments to capture enough of the structure to allow reliable beat extraction in a variety of musical situations. While it is in principle possible to derive the distribution of the samples in the rhythm tracks from a probabilistic model of the original audio, this is too complex to result in a feasible algorithm. Also, it is no more obvious how to construct a model for the audio than for the rhythm tracks directly.

Fig. 2 shows the structure of the model. The parameters can be divided into two sets: the *structural* parameters remain essentially constant through the piece (and are estimated off-line

from training data as in Section VI), while the *timing* parameters are the most interesting from the point of view of beat extraction.

The structural parameters:

- σ_1^2 is the “off the beat” variance;
- σ_2^2 is the “on the beat” variance;
- ω is the beatwidth, the variance of the width of each set of “on the beat” events. For simplicity, this is assumed to have Gaussian shape.

The timing parameters:

- τ is the time of the first beat;
- T is the period of the beat;
- δT is the rate of change of the beat period.

Given the signal \mathbf{r} (the rhythm track), Bayes theorem asserts that the probability of the parameters given the signal is proportional to the probability of the signal given the parameters multiplied by the prior distribution over the parameters. Thus

$$p(\tau, T, \delta T, \sigma_1, \sigma_2, \omega | \mathbf{r}) = p(\mathbf{r} | \tau, T, \delta T, \sigma_1, \sigma_2, \omega) \times p(\tau) p(T) p(\delta T) p(\sigma_1^2) p(\sigma_2^2) p(\omega^2)$$

where the priors are assumed independent.¹ Each of the prior probabilities on the right-hand side are fixed vis a vis the length of the data record, while the first term increases as a function of the length of the data. Accordingly, the first term dominates. Let $t_i = iT + \sum_{j=0}^i j\delta T + \tau$ be the time of the i th beat and let $\lambda_t = \sum_{i=-\infty}^{\infty} e^{-(t-t_i)^2/2\omega^2}$ be a sum of shifted Gaussian functions. The variance

$$\sigma_t^2 = \lambda_t \sigma_2^2 + (1 - \lambda_t) \sigma_1^2 \quad (4)$$

specifies the likelihood of the rhythm track model as

$$p(r_t | \tau, T, \delta T, \sigma_1, \sigma_2, \omega) = \mathcal{N}(0, \sigma_t^2) \quad (5)$$

where t is the (positive) time at which the sample r_t is observed, and where $\mathcal{N}(\cdot, \cdot)$ denotes the Gaussian distribution.

Because the rhythm track values r_t are assumed independent, the probability of a block of values $\{r_t, t_1 < t < t_2\}$ is simply the product of the probability of each value. Thus σ_t^2 is a combination of the variances on and off the beat, weighted by how far t is from the nearest (estimated) beat location. While this may appear noncausal, it is not because it only requires observations up to the current time. Also note that the summation in the definition of λ_t can in practice be limited to nearby values of j .

To see the relationship between experimentally derived rhythm tracks and the model, Fig. 3 shows an “artificial” rhythm track constructed from alternating small and large variance normally distributed random variables. Observe that qualitatively, this provides a reasonable model of the various rhythm tracks in Fig. 1.

B. Tracking Using Particle Filters

Divide each rhythm track into blocks, typically about 400 samples long. Collect the timing parameters, τ , T and δT into a state vector \mathbf{t} , and let $p(\mathbf{t}_{k-1} | \cdot)$ be the distribution over the parameters at block $k-1$. The goal of the (recursive) particle filter

¹In reality, the priors cannot be truly independent, for example, the structure of the model dictates that $\sigma_2 > \sigma_1$.

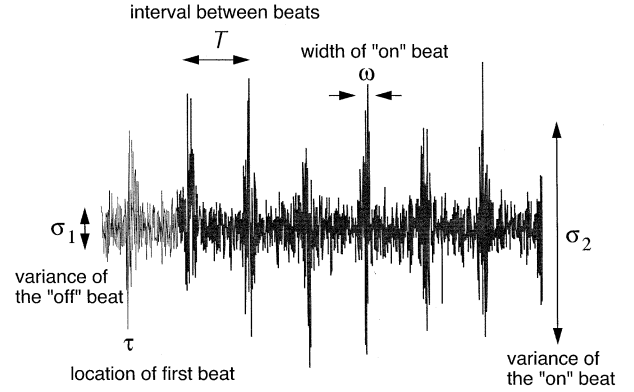


Fig. 2. Parameters of the rhythm track model are T , τ , ω , σ_1 , σ_2 and δT (not shown).

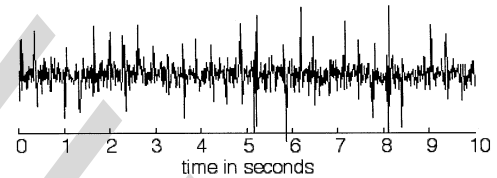


Fig. 3. An “artificial” rhythm track with a period of twenty-five samples: every group of 20 small variance $N(0, 1)$ random variables is followed by five large variance $N(0, 1.7)$ random variables. Observe the similarity between this and the experimental rhythm tracks of Fig. 1.

is to update this to estimate the distribution over the parameters at block k , that is, to estimate $p(\mathbf{t}_k | \cdot)$. For the linear Gaussian case, this can be optimally solved using the Kalman filter [17]. However, the timing parameters \mathbf{t} enter into (5) in a nonlinear manner, and so the Kalman filter is not directly applicable.

The tracking can be divided into two stages, *prediction* and *update*. Because the beat period does not remain precisely fixed (otherwise the tracking exercise would be pointless), knowledge of the timing parameters becomes less certain over the block and the distribution of \mathbf{t} becomes more diffuse. The update step incorporates the new block of data from the rhythm track, providing new information to lower the uncertainty and narrow the distribution.

The predictive phase details how \mathbf{t}_k is related to \mathbf{t}_{k-1} in the absence of new information. In general, this is a diffusion model

$$\mathbf{t}_k = f_{k-1}(\mathbf{t}_{k-1}, w_{k-1})$$

where f_{k-1} is some known function and w_{k-1} is a vector of random variables. The simplest form is to suppose that as time passes, the uncertainty in the parameters grows as in a random walk

$$\mathbf{t}_k = \mathbf{t}_{k-1} + w_{k-1}$$

where the elements of w_{k-1} have different variances that reflect prior information about how fast the particular parameter is likely to change. For beat tracking, these variances are dependent on the style of music; for instance, the expected change in T for a dance style would generally be much smaller than for a style with more rubato.

At block k , a noisy observation is made, giving the signal

$$r_k = h_k(\mathbf{t}_k, v_k)$$

where h_k is a measurement function and v_k is the noise. To implement the updates recursively requires expressing $p(\mathbf{t}_k|\mathbf{R}_k)$ in terms of $p(\mathbf{t}_{k-1}|\mathbf{R}_{k-1})$, where \mathbf{R}_{k-1} represents all the rhythm track samples up to block $k-1$.

This can be rewritten

$$p(\mathbf{t}_k|\mathbf{R}_{k-1}) = \int p(\mathbf{t}_k|\mathbf{t}_{k-1})p(\mathbf{t}_{k-1}|\mathbf{R}_{k-1})d\mathbf{t}_{k-1} \quad (6)$$

as the product of the predictive distribution $p(\mathbf{t}_k|\mathbf{t}_{k-1})$ (which can be calculated from the diffusion model) and the posterior distribution at time k (which can be initialized using the prior), then integrated over all possible values \mathbf{t}_{k-1} . Bayes theorem asserts that

$$p(\mathbf{t}_k|\mathbf{R}_k) = \frac{p(r_k|\mathbf{t}_k)p(\mathbf{t}_k|\mathbf{R}_{k-1})}{p(r_k|\mathbf{R}_{k-1})} \quad (7)$$

where the term $p(r_k|\mathbf{t}_k)$ is a simplification of $p(r_k|\mathbf{t}_k, \mathbf{R}_{k-1})$ as the current observations are conditionally independent of past observations given the current parameters. The numerator is the product of the likelihood at block k and the predictive prior, and the denominator can be expanded

$$p(r_k|\mathbf{R}_{k-1}) = \int p(r_k|\mathbf{t}_k)p(\mathbf{t}_k|\mathbf{R}_{k-1})d\mathbf{t}_k.$$

In a one-shot estimation, this normalization can be ignored because it is constant. In the recursive form, however, it changes at each iteration.

This method can be applied to the model of rhythm tracks by writing the predictive distribution as shown in the equation at the bottom of the page where σ_T^2 , $\sigma_{\delta T}^2$ and σ_τ^2 are the variances of the diffusions on each parameter, which are assumed independent. The likelihood is

$$p(r_k|\mathbf{t}_k) = N(0, \sigma_k^2) = e^{-(r_k^2/\sigma_k^2)} \quad (8)$$

where σ_k^2 is defined as in (4). Assuming an initial distribution $p(\mathbf{t}_1|\mathbf{R}_0)$ is available, these equations provide a formal solution to the estimate through time of the distribution of the timing parameters [2].

In practice, however, for even moderately complex distributions, the integrals in the above recursion are analytically intractable. Particle filters [8], [34] overcome these problems by approximating the (intractable) distributions with a set of values (the ‘‘particles’’) that have the same distribution, and then updating the particles over time. Estimates of quantities of interest (means, variances, etc.) are made directly from the sample set. More detailed presentations of the particle filter method can be found in [9] and [12].

Applied to the beat tracking problem, the particle filter algorithm can be written succinctly in three steps. The particles are a set of N random samples, $\mathbf{t}_k(i)$, $i = 1 \dots N$ distributed as $p(\mathbf{t}_{k-1}|\mathbf{R}_{k-1})$.

1) **Prediction:** Each sample is passed through the system model to obtain samples of

$$\mathbf{t}_k^\dagger(i) = \mathbf{t}_{k-1}(i) + w_{k-1}(i) \text{ for } i = 1, 2, \dots, N$$

which adds noise to each sample and simulates the diffusion portion of the procedure, where $w_{k-1}(i)$ is assumed to be a 3-dimensional Gaussian random variable with independent components. The variances of the three components depend on how much less certain the distribution becomes over the block.

2) **Update:** with the new block of rhythm track values, r_k , evaluate the likelihood for each particle using (5). Compute the normalized weights for the samples

$$q_i = \frac{p(r_k|\mathbf{t}_k^\dagger(i))}{\sum_i p(r_k|\mathbf{t}_k^\dagger(i))}.$$

3) **Resample:** Resample N times from the discrete distribution over the $\mathbf{t}_k^\dagger(i)$'s defined by the q_i 's to give samples distributed as $p(\mathbf{t}_k|\mathbf{R}_k)$.

To initialize the algorithm, draw N samples from the prior distribution $p(\tau, T, \delta T)$, which is taken as uniform over some reasonable range. If more information is available (as studies such as [19] suggest), then better initializations may be possible. A number of alternative resampling schemes [9], [10] with different numerical properties could be used in the final stage of the algorithm.

C. Multiple Rhythm Tracks

A major advantage of the Bayesian approach is its ability to incorporate information from multiple rhythm tracks. Assuming that the various rhythm tracks provide independent measurements of the underlying phenomenon (a not unreasonable assumption given that the tracks measure different aspects of the input signal), then the likelihood for a set of rhythm tracks is simply the product of the likelihood for each track. Thus, the algorithm for estimating the optimal beat times from a collection of four rhythm tracks is only four times the difficulty of estimating from a single-rhythm track.

V. GRADIENT ESTIMATION

This section defines a ‘‘cost’’ $J(\mathcal{T})$ that varies with the time interval \mathcal{T} between successive beats; \mathcal{T} that minimize the cost for particular rhythm tracks are good candidates for the duration of a beat of the corresponding music. It is not generally possible to minimize J directly, so the optimization must be approached via some kind of iterative method. This section details two possibilities, one relies on calculation of the gradient and the other approximates the gradient using only evaluations of J . Both are able to reasonably solve the optimization problem, and both are inherently recursive. This should reasonably allow the algorithm to track the beat as the beat duration changes over

$$p(\mathbf{t}_k|\mathbf{t}_{k-1}) = e^{-((\mathcal{T}_k - \mathcal{T}_{k-1})^2 / 2\sigma_T^2)} e^{-((\delta\mathcal{T}_k - \delta\mathcal{T}_{k-1})^2 / 2\sigma_{\delta T}^2)} e^{-((\tau_k - \tau_{k-1})^2 / 2\sigma_\tau^2)}$$

time. Indeed, examples demonstrate that this is so, at least as long as the changes occur slowly. A rate of change term analogous to $\delta\mathcal{T}$ in (5) is not used in the gradient algorithm.

To be concrete, suppose that $r(t)$ is the data in a rhythm track. Let $g(t)$ be a function that is large near the origin and that grows small as t deviates from the origin. The Gaussian function $g(t) = e^{-t^2/\sigma^2}$ is one possibility, where σ is chosen so that the ‘‘width’’ of $g(t)$ is narrower than the time span expected to occur between successive beats. Thus, the choice of σ in the gradient method is analogous to the beatwidth parameter ω in the Bayesian method. Let \mathcal{T}_k be the best estimate at time step k of the beat duration, and let τ_k be the time of the most recent beat. Then the sum $\tau_k + \mathcal{T}_k$ provides the best estimate of the next beat location. The cost is defined to be

$$J(\mathcal{T}_k) = \frac{1}{2} [\arg \max\{r(t) g(t - \tau_k - \mathcal{T}_k)\} - (\tau_k + \mathcal{T}_k)]^2. \quad (9)$$

To understand this, observe that $g(t - \tau_k - \mathcal{T}_k)$ is a (Gaussian) pulse with variance σ^2 and centered at $\tau_k + \mathcal{T}_k$, the estimated location of the next beat. The product $r(t)g(t - \tau_k - \mathcal{T}_k)$ weights the rhythm track so as to emphasize information near the expected beat and to attenuate data far from the expected beat. The $\arg \max\{\cdot\}$ function picks out the largest peak in the rhythm track near the expected beat location, and returns the location of the peak. This is (likely) the actual beat location. The difference between the $\arg \max$ (the likely location of the beat as given in the data) and $\tau_k + \mathcal{T}_k$ (the estimated location of the beat) is thus the basis of the cost. If the rhythm track was a regular succession of pulses and the estimates of τ and \mathcal{T} were accurate, then the cost would be zero. When the rhythm track is derived from a piece of music as described in Section III, then the cost can be used to continuously make better estimates of the beat duration.

One approach is to use a gradient descent strategy, which updates the current estimate at time k using the iteration

$$\mathcal{T}_{k+1} = \mathcal{T}_k - \mu \frac{\partial J(\mathcal{T}_k)}{\partial \mathcal{T}_k} \quad (10)$$

where μ is a (small) stepsize that determines how much the current estimates react to new information. Calculating this gradient is somewhat tricky due to the presence of the $\arg \max$ function, and details are given in Appendix B, where (10) is explicitly shown to be

$$\mathcal{T}_{k+1} = \mathcal{T}_k + \mu \frac{\sigma^2 [r(t_k^*)\dot{r}(t_k^*) - \dot{r}^2(t_k^*)] [t_k^* - \tau_k - \mathcal{T}_k]}{2r^2(t_k^*) + \sigma^2 [r(t_k^*)\dot{r}(t_k^*) - \dot{r}^2(t_k^*)]} \quad (11)$$

where

$$t_k^* = \arg \max\{r(t) g(t - \tau_k - \mathcal{T}_k)\} \quad (12)$$

is the value at which $r(t) g(t - \tau_k - \mathcal{T}_k)$ achieves its maximum.

While this algorithm is fairly simple, it does require the calculation of the (numerical) derivatives \dot{r} and \ddot{r} . The update term can be factored as

$$[t_k^* - \tau_k - \mathcal{T}_k] \left[\frac{2r^2(t_k^*)}{2r^2(t_k^*) + \sigma^2 [r(t_k^*)\dot{r}(t_k^*) - \dot{r}^2(t_k^*)] - 1} \right].$$

Near a beat $\dot{r}(t_k^*) \approx 0$, $r(t_k^*)$ is large, and $\ddot{r}(t_k^*)$ is negative (since $r(t_k^*)$ is near a maximum). Hence, the term multiplying σ^2 in the denominator is negative, and the denominator is smaller than the numerator. Thus, the fraction is larger than 1, and so the update has the opposite sign from $t_k^* - \tau_k - \mathcal{T}_k$. Accordingly, the simpler algorithm

$$\mathcal{T}_{k+1} = \mathcal{T}_k - \mu [t_k^* - \tau_k - \mathcal{T}_k] \quad (13)$$

updates in the correct direction, and may be considered a reasonable simplification of (11). In fact, (13) has a easy intuitive interpretation: $\tau_k + \mathcal{T}_k$ is the predicted location of the next beat, while t_k^* is the where the beat actually occurs. The difference between the prediction and the measurement provides information to improve the estimate. The algorithm increases \mathcal{T} if the prediction was early, and decreases \mathcal{T} if the prediction was late. This update is analogous to one of the algorithms in [7], though the inputs are quite different since (13) operates on the rhythm tracks and without note onset information.

VI. EXAMPLES

This section provides a number of examples that show how the beat tracking algorithms function. In all cases, the output is a sequence of times that are intended to represent when ‘‘beats’’ occur: when listeners ‘‘tap their feet’’. To make this accessible, an audible burst of noise was superimposed over the music at the predicted time of each beat. By listening, it is clear when the algorithm has ‘‘caught the beat’’ and when it has failed. We encourage the reader to listen to the .mp3 examples from our website [35] to hear the algorithms in operation; graphs such as Figs. 4 and 5 are a meager substitute.

A. Using the Particle Filter

For the Bayesian particle filter of Section IV it was first necessary to estimate the structural parameters, σ_1 , σ_2 , and ω . Initial values were chosen by hand based on an inspection of the rhythm tracks. Using these values, the algorithm was run to extract the beats from several pieces. These results were then used to re-estimate the parameters using the entire rhythm tracks, and the values of the parameters from the different training tracks

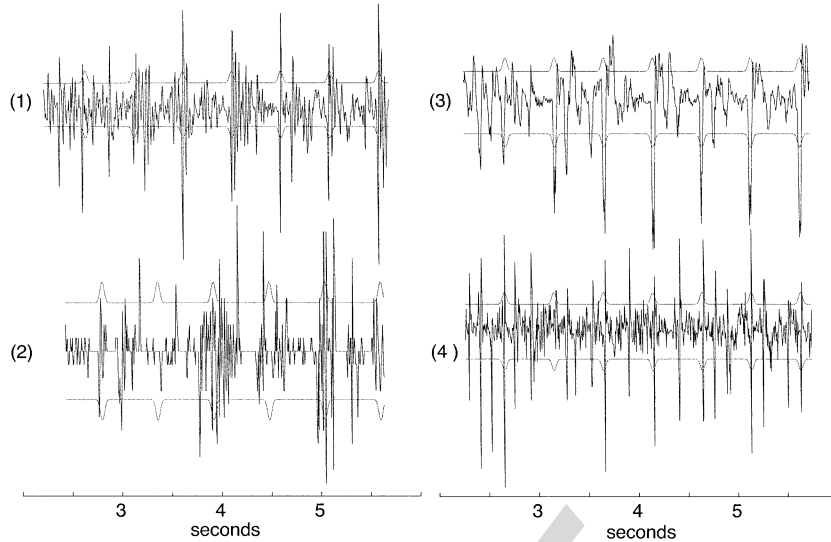


Fig. 4. The four rhythm tracks of “Pieces of Africa” by the Kronos quartet between 2 and 6 s. The estimated beat times [which correctly locate the beat in cases (1), (3), and (4)] are superimposed over each track.

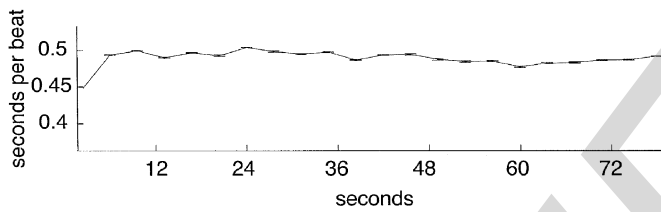


Fig. 5. The tempo of Kronos Quartet’s “Pieces of Africa” changes slightly over time and the tempo parameter \mathcal{T} follows.

were then averaged.² These values were then fixed when estimating the beats in subsequent audio tracks which were not part of the training set. The nominal values were $\omega = 0.02$, $\sigma_1^2 = 0.14$, and $\sigma_2^2 = 0.2$, while the initialization of \mathcal{T} was uniform in the range $[0.2, 0.4]$ s, $\delta\mathcal{T}$ was uniform $[-0.0001, 0.0001]$ and τ was uniform $[0, 0.2]$.

The particle filter was applied to a variety of different pieces in different musical styles including pop music (“Norwegian Wood” by the Beatles, “Mr Tambourine Man” by the Byrds), jazz (“Take Five” by David Brubek), classical (Scarlatti’s “Sonata K517 in D Minor”, “Water Music” by Handel), film (“Theme from James Bond”), folk (“The Boxer” by Simon and Garfunkel), country (“Angry Young Man” by Steve Earl), dance (“Lion Says” by Prince Buster and The Ska), and bluegrass (“Man of Constant Sorrow”). In all of these cases the algorithm located a regular lattice of times that correspond to times that a listener might tap the foot. The first 30 s of each of these can be heard at [35].

While the algorithm is running, plots such as Fig. 4 are generated. This shows the four rhythm tracks at the start (between 2 and 6 s) of “Pieces of Africa” by the Kronos quartet. The smooth curves with the bumps show the predicted tap times. Some of the rhythm tracks show the pulse nicely, and the algorithm aligns

²We scale the rhythm tracks so that they have approximately equal power. This allows use of one set of parameters for all rhythm tracks despite different physical units.

itself with this pulse. Rhythm track three provides the cleanest picture with large spikes at the beat locations and small deviations between. Similarly, the first rhythm track shows the beat locations but is quite noisy (and temporally correlated) between spikes. Rhythm track four shows spikes at most of the beat locations, but also has many spikes in other locations, many at twice the tap rate. Rhythm track two is unclear, and the lattice of times found by the algorithm when operating only on this track is unrelated to the real pulse of the piece. In operation, the algorithm derives a distribution of samples from all four rhythm tracks that is used to initialize the next block. The algorithm proceeds through the complete piece block by block.

The predicted times actively follow the music. For instance, over the first 1:14 s of Kronos Quartet’s “Pieces of Africa”, the tempo wavers somewhat, speeding up at around 60 s and then slowing back down. The algorithm is able to track such changes without problem, and the tempo parameter \mathcal{T} is plotted in Fig. 5. This example can also be heard at [35].

Depending on the range of the initial timing parameter \mathcal{T} , the algorithm would sometimes lock onto a beat that was twice the speed or half the speed of the nominal tap rate. In one particular case (“Jupiter” by Jewel) it was able to lock onto either twice or half (depending on the initialization), but not the anticipated “quarter-note” itself. In another case “Lion Says” by Prince Buster and The Ska, the algorithm could lock onto the “on-beat,” the “off-beat,” or onto twice the nominal tap rate, depending on the initialization of \mathcal{T} . These can also be heard at the website. Given that reasonable people can disagree by factors of two on the appropriate tempo (one clapping hands or tapping feet at twice the rate of the other), and that some people tend to clap hands on the on-beat, while others do so on the off-beat, such effects should be expected. The faster rate is sometimes called the “tatum” while the slower is called the “beat.” Thus the algorithm cannot distinguish the tatum from the beat without further high-level information. Of more concern was the rare case that locked onto two equally spaced taps for each three beats.

In order to explore the behavior of the method further, we used the gnutella file sharing network [11] to locate twenty-six versions of the "Maple Leaf Rag" by Scott Joplin. About half were piano renditions, the instrument it was originally composed for. Other versions were performed on solo guitar, banjo, or marimba. Renditions were performed in diverse styles: a klezmer version, a bluegrass version, Sidney Bechet's big band version, one by the Canadian brass ensemble, and an orchestral version from the film "The Sting". In 14 of the twenty six versions the beat was correctly located using the default values in the algorithm. Another eight were correctly identified by increasing or decreasing the ranges of the initial time \mathcal{T} , the maximum allowable rate of change $\delta\mathcal{T}$, or the time window over which the algorithm operated. The remaining four are apparently beyond the present capabilities of the algorithm.

Two of the four failures were piano performances, one was a jazz band rendition, and one used primarily synthesized sounds. The likely reasons for failure are as diverse as the renditions. In one (performed by Hyman) all is well until about at about 2:04 when there are a series of drastic tempo shifts from which the algorithm never recovers. In another (performed by Tommy Dorsey's band) the algorithm synchronizes and loses synchrony repeatedly as the instrumentation changes. This is likely a fault of the rhythm tracks not being consistent enough. The synthesized version is bathed in reverb, and this likely decreases the accuracy with which the rhythm tracks can mirror the underlying beat of the piece.

B. Using the Gradient Algorithm

The promise of the gradient approach is its low numerical complexity. That the gradient is in principle capable of solving the beat tracking problem is indicated in Fig. 6 which shows 70 different runs of the algorithm applied to the "Theme from James Bond". Each run initializes the algorithm at a different starting value between 20 and 90 samples (between 0.11 and 0.52 s). In many cases, the algorithm converges nicely. Observe that initializations between 35 and 45 converge to the eighth-note beat at 0.23 s per beat, while initializations between 75 and 85 converge to the quarter-note beat at about 0.46 s. Other initializations do not converge, at least over the minute analyzed.

When first applying the algorithm, it was necessary to run through the rhythm tracks many times to achieve convergence. By optimizing the parameters of the algorithm (stepsize, number of beats examined in each iteration, etc) it was possible to speed converge to within 30 or 40 beats (in this case, 15 to 20 s). What is hard to see because of the scale of the vertical axis in Fig. 6 is that even after the convergence, the estimates of the beat times continue to oscillate above and below the correct value. This can be easily heard as alternately rushing the beat and dragging behind. The problem is that increasing the speed of convergence also increases the sensitivity.

In order to make the gradient algorithm comparable to the Bayesian approach, the basic iteration (10) needed to be expanded to use information from multiple \mathcal{T} intervals simultaneously and to use information from multiple rhythm tracks simultaneously. Both of these generalizations are straightforward in the sense that predictions of the beat locations and deviations can follow the same methods as in (10)–(13) whether predicting

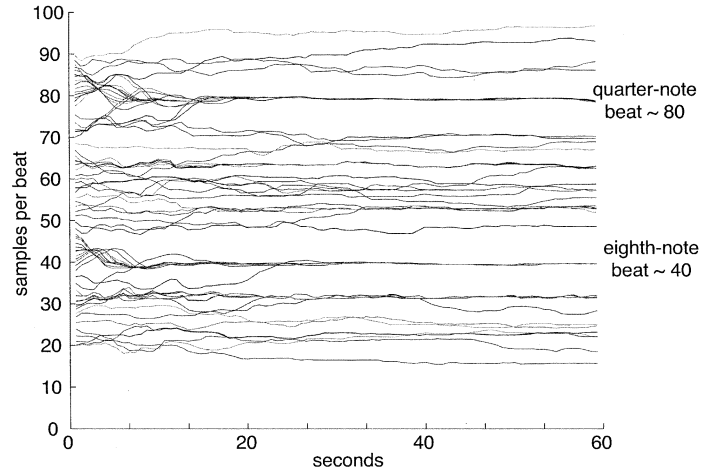


Fig. 6. Estimates of the beat period for the "Theme from James Bond" using the gradient algorithm. Depending on the initial value it may converge to the eighth-note beat at about 40 samples per period (about 0.23 s) or to the quarter-note beat near 80 samples (about 0.46 s).

one beat or n beats into the future and whether predicting from a single rhythm track or many. What is new is that there must be a way of combining the multiple estimates. We tried several methods including averaging the updates from all n beats and all rhythm tracks, using the median of this value, and weighting the estimates. The most successful of the schemes (used to generate the examples such as Fig. 6) weighted each estimate in proportion to $r(t_k^*)g(t_k^* - \tau_k - \mathcal{T}_k)$ [using the notation from (12)] since this places more emphasis on estimates which are "almost" right.

Overall, the results of the gradient algorithm were disappointing. By hand tuning the windows and stepsizes, and using proper initialization, it can often find and track the beat. But these likely represent an unacceptable level of user interaction. Since the Bayesian algorithm converges rapidly within a few beats, it can be used to initialize the gradient algorithm, effectively removing the initial undulations in the timing estimates. Of the 26 versions of the "Maple Leaf Rag", this combined algorithm was able to successfully complete only eight: significantly fewer than the particle filter alone.

VII. CONCLUSIONS

The analysis of musical rhythms is a complex task. As noted in [18], "even the most innocuous sequences of note values permit an unlimited number of rhythmic interpretations." Most proposals for beat tracking and rhythm finding algorithms operate on "interonset" intervals (for instance [6], [7], [13]), which presupposes either a priori knowledge of note onsets (such as are provided by MIDI) or their accurate detection. Our method, by ignoring the "notes" of the piece bypasses (or ignores) this element of rhythmic interpretation. This is both a strength and a weakness. Without a score, the detection of "notes" is a non-trivial task, and errors such as missing notes (or falsely detecting notes that are not actually present) can bias the detected beats. Since our method does not detect notes it cannot make such mistakes. The price, of course, is that the explanatory power of a note-based approach remains unexploited. Thus the beat tracking techniques of this paper are more methods of signal

processing at the level of sound waveforms than of symbol manipulation at the note level.

Scheirer [26] creates a signal that consists of noisy pulses derived from the amplitude envelope of audio passed through a collection of “critical band” filters. Interestingly, much of the rhythmic feel of the piece can be heard in the artificial noisy signal. Thus Scheirer suggests that a psychological theory of the perception of meter need not operate at the level of notes. To the extent that our method is capable of finding pulses within certain musical performances, our results support the conclusion that the “note” and “interonset” levels of interpretation are not a necessary component of rhythmic detection.

Though our method does not attempt to decode pitches (which are closely tied to a note level representation), it is not insensitive to frequency information since this is incorporated indirectly into the various rhythm tracks. This allows the timbre (or spectrum) of the sounds to influence the search for appropriate periodicities in a way that is lost if only energy or interonset interval encoding is used.

Cemgil and Kappen [4] compare two algorithms for the beat tracking of MIDI data and conclude that the particle filter methods outperform iterative methods such as simulated annealing and iterative improvement. This parallels our results that the particle filter outperforms the gradient method. One interpretation for the disparity is that the gradient algorithm has no way to exploit the probabilistic structure of the rhythm tracks.

The particle filter beat tracking method is generally successful at identifying the initial tempo parameters and at following tempo changes. One mode of failure is when the tempo changes too rapidly for the algorithm to track, as might occur in a piece with extreme rubato. A principled approach to the handling of abrupt rapid changes would be to include a small probability of a radical change in the parameters of the particle filter. Perhaps the most common mode of failure is when the rhythm tracks fail to have the hypothesized structure (rather than a failure of the algorithm in identifying the structure when it exists). Thus a promising area for research is the search for better rhythm tracks. There are many possibilities: rhythm tracks could be created from a subband decomposition, from other distance measures in either frequency or time, or using probabilistic methods. What is needed is a way of evaluating the efficacy of a candidate rhythm track. Also at issue is the question of how many rhythm tracks can be used simultaneously. In principle, there is no limit as long as they remain “independent.” Given a way of evaluating the usefulness of the rhythm tracks and a precise meaning of independence, it may be possible to approach the question of how many degrees of freedom exist in the underlying rhythmic process.

APPENDIX A ANALYSIS OF THE GROUP DELAY

For many musical waveforms, the unwrapped phase of the Fourier Transform often lies very close to a line. This appendix

investigates the meaning of this observation and interprets the slope t_g of this line in terms of the concentration of energy at a time (shift) proportional to t_g .

Given a signal $x(t)$, its complex valued Fourier Transform $X(\omega)$ can be written in terms of its magnitude and phase as

$$X(\omega) = m(\omega)e^{j\theta(\omega)}$$

where $j = \sqrt{-1}$. Consider the related transform $\hat{X}(\omega) = m(\omega)e^{j0} = m(\omega)$ which is obtained from $X(\omega)$ by setting all the phase values to zero. Denote the corresponding time waveform $\hat{x}(t)$.

First, we show that $\hat{x}(t)$ attains its largest value at $t = 0$

$$\begin{aligned} |\hat{x}(t)| &= \left| \frac{1}{2\pi} \int_{-\infty}^{\infty} m(\omega)e^{j\omega t} d\omega \right| \\ &\leq \frac{1}{2\pi} \int_{-\infty}^{\infty} |m(\omega)| |e^{j\omega t}| d\omega. \end{aligned}$$

But $m(\omega)$ is already positive, and $|e^{j\omega t}| = 1 = e^{j\omega 0}$, and hence

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} m(\omega)e^{j\omega 0} d\omega = \hat{x}(0).$$

Thus, the procedure of removing all phase information from a signal changes it substantively, so that the largest value occurs at zero. In general, the bulk of the energy in $\hat{x}(t)$ is concentrated near $t = 0$.

The observation that the group delay parameter t_g is (nearly) constant is equivalent to requiring that

$$t_g = \frac{\theta(\omega)}{\omega}$$

be constant. Hence, t_g is the slope of $\theta(\omega) = t_g\omega$, and $x(t)$ can be written in terms of the transform $X(\omega)$ as

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} m(\omega)e^{j\theta(\omega)}e^{j\omega t} d\omega.$$

When the phase is nearly linear, this can be approximated by

$$\approx \frac{1}{2\pi} \int_{-\infty}^{\infty} (m(\omega)e^{j\omega t_g}) e^{j\omega t} d\omega.$$

By the “modulation” property of the Fourier transform [20], this is precisely $\hat{x}(t + t_g)$, where $\hat{x}(t)$ is the zero phase version of $x(t)$.

Accordingly, the process of replacing an approximately linear phase $x(t)$ with an exact linear phase $x_{t_g}(t)$ is the same as translating the zero phase $\hat{x}(t)$ version of the signal t_g units in time. Since $\hat{x}(t)$ has its energy concentrated near $t = 0$, the exact linear phase signal $x_{t_g}(t)$ has its energy concentrated near $t = t_g$.

Thus, when applying the FFT to a windowed version of a continuous signal, t_g provides an estimate of where the energy is concentrated. It is not dependent of the total energy in the

window, but rather on the distribution of the energy within the window.

One way to view this approximation is to observe that the transform of $\hat{x}(t + t_g)$ is

$$\begin{aligned} e^{jt_g\omega} \hat{X}(\omega) &= m(\omega) e^{j\theta(\omega)} e^{j(t_g\omega - \theta(\omega))} \\ &= \hat{X}(\omega) \cdot 1(\omega) e^{j(t_g\omega - \theta(\omega))} \end{aligned}$$

where $1(\omega)$ represents the magnitude of an all pass filter (which is unity for all frequencies) and $e^{j(t_g\omega - \theta(\omega))}$ represents its phase, which is equal to the difference between the original phase of $X(\omega)$ and the phase of $\hat{X}(\omega)$.

APPENDIX B

DERIVATION OF THE GRADIENT ALGORITHM

This appendix calculates the gradient of the cost function (9) with respect to \mathcal{T}_k in order to put (10) into implementable form. For simplicity of exposition, the dependence of terms on k is suppressed, and the recommended Gaussian form for $g(t)$ is assumed. Applying the chain rule to (9) gives

$$\begin{aligned} \frac{\partial J(\mathcal{T})}{\partial \mathcal{T}} &= [\operatorname{argmax}\{r(t)g(t - \tau - \mathcal{T})\} - (\tau + \mathcal{T})] \\ &\quad \times \left[\frac{\partial \operatorname{argmax}\{r(t)g(t - \tau - \mathcal{T})\}}{\partial \mathcal{T}} - 1 \right]. \end{aligned}$$

This is a straightforward calculation except for the derivative of the argmax function. Let $t^* = \operatorname{argmax}\{r(t)g(t - \tau - \mathcal{T})\}$ be the value at which $r(t)g(t - \tau - \mathcal{T})$ achieves its maximum. The analysis proceeds by writing t^* explicitly as a function of \mathcal{T} , and then applying the inverse function theorem to express \mathcal{T} as a function of t^* , at which point the desired derivative can be computed.

Observe that $r(t)g(t - \tau - \mathcal{T}) = r(t)e^{-(t - \tau - \mathcal{T})^2 / \sigma^2}$ achieves its maximum when its derivative is zero, that is, when

$$\begin{aligned} \dot{r}(t)e^{-(t - \tau - \mathcal{T})^2 / \sigma^2} + r(t) \\ \times \left[2 \frac{(t - \tau - \mathcal{T})}{\sigma^2} e^{-(t - \tau - \mathcal{T})^2 / \sigma^2} \right] = 0. \end{aligned}$$

Since $e^{(\cdot)}$ is never zero, this maximum occurs at values of $t = t^*$ for which

$$\dot{r}(t^*) + 2 \frac{(t^* - \tau - \mathcal{T})}{\sigma^2} r(t^*) = 0.$$

This can be solved for \mathcal{T} as

$$\mathcal{T} \equiv f(t^*) = t^* - \tau + \frac{\sigma^2 \dot{r}(t^*)}{2r(t^*)} \quad (14)$$

which expresses \mathcal{T} as a function of t^* . However, calculation of the gradient requires an expression for t^* as a function of \mathcal{T} . If the inverse function is denoted $t^* = h(\mathcal{T})$, then the Inverse Function Theorem [24] expresses the derivative $\dot{h}(\mathcal{T})$ in terms of $\dot{f}(t^*)$ as

$$\dot{h}(\mathcal{T}) = \frac{1}{\dot{f}(h(\mathcal{T}))}.$$

From (14), the derivative of $f(t)$ with respect to t^* is

$$\dot{f}(t^*) = 1 + \frac{\sigma^2 [r(t^*)\ddot{r}(t^*) - \dot{r}^2(t^*)]}{2r^2(t^*)}$$

and so

$$\dot{h}(\mathcal{T}) = \frac{1}{\dot{f}(t^*)} = \frac{2r^2(t^*)}{2r^2(t^*) + \sigma^2 [r(t^*)\ddot{r}(t^*) - \dot{r}^2(t^*)]}.$$

Gathering terms together shows that

$$\begin{aligned} \frac{\partial J(\mathcal{T})}{\partial \mathcal{T}} &= [t^* - \tau - \mathcal{T}] [\dot{h}(\mathcal{T}) - 1] \\ &= \frac{-\sigma^2 [r(t^*)\ddot{r}(t^*) - \dot{r}^2(t^*)] [t^* - \tau - \mathcal{T}]}{2r^2(t^*) + \sigma^2 [r(t^*)\ddot{r}(t^*) - \dot{r}^2(t^*)]} \end{aligned}$$

which gives the desired form for the algorithm update.

REFERENCES

- [1] P. E. Allen and R. B. Dannenberg, "Tracking musical beats in real time," in *Proc. Int. Computer Music Conf.*, Glasgow, Glasgow, Scotland, U.K., Sep. 1990.
- [2] J. M. Bernardo and A. F. M. Smith, *Bayesian Theory*. New York: Wiley, 2001.
- [3] A. T. Cemgil, B. Kappen, P. Desain, and H. Honing, "On tempo tracking: Tempogram representation and Kalman filtering," *J. New Music Res.*, vol. 28, no. 4, 2001.
- [4] A. T. Cemgil and B. Kappen, "Monte Carlo methods for tempo tracking and rhythm quantization," *J. Artific. Intell. Res.*, vol. 18, pp. 45–81, 2003.
- [5] D. Cliff, "Hang the DJ: Automatic Sequencing and Seamless Mixing of Dance Music Tracks," Hewlett Packard TR HPL-2000.104, 2000.
- [6] P. Desain, "A (de)composable theory of rhythm perception," *Music Perception*, vol. 9, no. 4, pp. 439–454, Summer 1992.
- [7] S. Dixon, "Automatic extraction of tempo and beat from expressive performances," *J. New Music Research*, vol. 30, no. 1, 2001.
- [8] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.
- [9] A. Doucet, N. de Freitas, and N. Gordon, "An introduction to sequential Monte Carlo methods," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds. New York: Springer-Verlag, 2001.
- [10] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-blackwellised particle filtering for dynamic Bayesian networks," *Proc. Uncertainty in Artificial Intelligence*, Jun. 2000.
- [11] [Online]. Available: <http://www.gnutella.com>
- [12] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *Proc. Inst. Elect. Eng. F*, vol. 140, no. 2, pp. 107–113, Apr. 1993.
- [13] M. Goto, "An audio-based real-time beat tracking system for music with or without drum-sounds," *J. New Music Res.*, vol. 30, no. 2, pp. 159–171, 2001.
- [14] F. Gouyon and P. Herrera, "A beat induction method for musical audio signals," in *Proc. 4th WIAMIS-Special Session on Audio Segmentation and Digital Music*, London, U.K., 2003.
- [15] —, "Determination of the meter of musical audio signals: Seeking recurrences in beat segment descriptors," in *Proc. AES, 114th Conv.*, Amsterdam, The Netherlands, Mar. 2003.
- [16] F. Gouyon and B. Meudic, "Toward rhythmic content processing of musical signals: Fostering complementary approaches," *J. New Music Res.*, vol. 32, no. 1, pp. 159–171, 2003.
- [17] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1991.
- [18] H. C. Longuet-Higgins and C. S. Lee, "The rhythmic interpretation of monophonic music," *Music Perception*, vol. 1, no. 4, pp. 424–441, Summer 1984.
- [19] L. van Noorden and D. Moelants, "Resonance in the perception of musical pulse," *J. New Music Res.*, vol. 28, no. 1, Mar. 1999.
- [20] Porat, *Digital Signal Processing*. New York: Wiley, 1997.
- [21] D. J. Povel and P. Essens, "Perception of temporal patterns," *Music Perception*, vol. 2, no. 4, pp. 411–440, Summer 1985.

- [22] C. Raphael, "Automated rhythm transcription," in *Proc. Int. Symp. Music Information Retrieval*, Bloomington, IN, Oct. 2001.
- [23] D. Rosenthal, "Emulation of rhythm perception," *Comput. Music J.*, vol. 16, no. 1, Spring 1992.
- [24] H. L. Royden, *Real Analysis*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1988.
- [25] J. Seppänen, "Computational Models of Musical Meter Recognition," M.S. thesis, Tampere Univ. Tech., Tampere, Finland, 2001.
- [26] E. D. Scheirer, "Tempo and beat analysis of acoustic musical signals," *J. Acoust. Soc. Amer.*, vol. 103, no. 1, pp. 588–601, Jan. 1998.
- [27] W. A. Sethares, *Tuning, Timbre, Spectrum, Scale*. New York: Springer-Verlag, 1997.
- [28] W. A. Sethares and T. Staley, "The periodicity transform," *IEEE Trans. Signal Processing*, vol. 47, no. 11, pp. 2953–2964, Nov. 1999.
- [29] —, "Meter and periodicity in musical performance," *J. New Music Res.*, vol. 30, no. 2, Jun. 2001.
- [30] M. J. Steedman, "The perception of musical rhythm and meter," *Perception*, vol. 6, pp. 555–569, 1977.
- [31] N. P. M. Todd, D. J. O'Boyle, and C. S. Lee, "A sensory-motor theory of rhythm, time perception and beat induction," *J. New Music Res.*, vol. 28, no. 1, pp. 5–28, 1999.
- [32] P. Toiviainen, "Modeling the perception of meter with competing subharmonic oscillators," in *Proc. 3rd Triennial ESCOM Conf.*, Uppsala, Sweden, 1997.
- [33] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Processing*, vol. 10, no. 5, pp. 293–302, Jul. 2002.
- [34] J. Vermaak, C. Andrieu, A. Doucet, and S. J. Godsill, "Particle methods for Bayesian modeling and enhancement of speech signals," *IEEE Trans. Speech Audio Processing*, vol. 10, no. 3, pp. 173–185, Mar. 2002.
- [35]
- [36] Y. Wang, "A beat-pattern based error concealment scheme for music delivery with burst packet loss," in *Int. Conf. Multimedia and Expo.*, 2001.
- [37] E. Wold, T. Blum, D. Keislar, and J. Wheaton, "Content-based classification, search and retrieval of audio," *IEEE Trans. Multimedia*, vol. 3, no. 3, pp. 27–36, 1996.
- [38] T. Zhang and J. Kuo, "Audio content analysis for online audiovisual data segmentation and classification," *IEEE Trans. Speech Audio Processing*, vol. 9, no. 4, pp. 441 457–441 457, May 2001.



William A. Sethares **AUTHOR: PLS. SUPPLY BIOGRAPHY ELECTRONICALLY BY EMAIL TO J.SINAY@IEEE.ORG**



Robin D. Morris **AUTHOR: PLS. SUPPLY BIOGRAPHY ELECTRONICALLY BY EMAIL TO J.SINAY@IEEE.ORG**



James C. Sethares **AUTHOR: PLS. SUPPLY BIOGRAPHY ELECTRONICALLY BY EMAIL TO J.SINAY@IEEE.ORG**