ANALYZING AFRO-CUBAN RHYTHM USING ROTATION-AWARE DYNAMIC PROGRAMMING

ABSTRACT

The majority of existing research in Music Information Retrieval (MIR) has focused on either popular or classical music and frequently makes assumptions that do not generalize to other music cultures. We use the term Computational Ethnomusicology (CE) to describe the use of computer tools to assist the analysis and understanding of music cultures from around the world. Although existing MIR techniques can serve as a good starting point for CE, the design of effective tools can benefit from incorporating domain specific knowledge about the music style and culture of interest. In this paper we describe our realization of this approach in the context of studying Afro-Cuban rhythm. More specifically we show how computer analysis can help us characterize and appreciate the complexities of tracking tempo and analyzing micro-timing in this particular music style. A novel template-based method for tempo tracking in rhythmically complex Afro-Cuban music is proposed. Although our approach is domain-specific, we believe that the concepts and ideas used could also be used for studying other music cultures after some adaptation.

1 INTRODUCTION

The use of computers for automatic and semi-automatic rhythmic analysis of audio recordings can greatly enhance the study of music from cultures in which rhythm plays a dominant role. We present a set of techniques and tools designed for studying rhythm and timing on recordings of Afro-Cuban music with particular emphasis on “clave” which is a rhythmic pattern used for temporal organization. In order to visualize timing information we propose a novel graphical representation that can be generated by computer from signal analysis of audio recordings and from listeners’ annotations collected in real time. The proposed visualization is based on the idea of Bar Wrapping, which is the breaking and stacking of a linear time axis at a fixed metric location.

The techniques proposed in this paper have their origins in Music Information Retrieval (MIR) but have been adapted and extended in order to analyze the particular music culture studied. Unlike much of existing work in MIR in which the target user is an “average” music listener, the focus of this work is people who are “experts” in a particular music culture. Examples of the type of questions they would like to explore include: how do expert players differ from each other, and also from competent musicians who are not familiar with the particular style; are there consistent timing deviations for notes at different metric positions; how does tempo change over the course of a recording etc. Such questions have been frequently out of reach because it is tedious or impossible to explore without computer assistance.

Creating automatic tools for analyzing micro-timing and tempo variations for Afro-Cuban has been challenging. Existing beat-tracking tools either don’t provide the required functionality (for example only perform tempo tracking but don’t provide beat locations) or are simply not able to handle the rhythmic complexity of Afro-Cuban music because they make assumptions that are not always applicable, such as expecting more and louder notes on metrically “strong” beats. Finally the required precision for temporal analysis is much higher than typical MIR applications. These considerations have motivated us to design a beat tracker that utilizes domain-specific knowledge about cuban rhythms.

2 BACKGROUND

The proposed techniques fall under the general rubric of what has been termed Computational Ethnomusicology (CE), which refers to the design and usage of computer tools that can assist ethnomusicological research [17] . The use of technology to assist ethnomusicological research is not new. Cooper and Sapiro [6] provide a survey of early technologies used in ethnomusicology including such amazing contraptions as the array of 54 tuning forks each separated by 4 Hz from its neighbors created in 1834 by the German acoustician Johann Heinrich Scheibler. The need of MIR research to expand to other domains of than Western pop and classical music has been argued in Futrelle [11]. Retrieval based on rhythmic information has been explored in the context of Greek and African traditional music [1]. One of the most interesting current attempts in CE is the digitization of the sound archive of the Royal Museum for Central Africa [7].

Our focus in this paper is the analysis of music in which percussion plays an important role, specifically, Afro-Cuban music. One of the earliest attempts to use a computer to analyze audio recordings of percussive music is described in Schloss [16], and the complexities and detail of timing in music have been explored using a computational approach [4]. Beat tracking and tempo induction are active topics of research, although they have mostly focused on popular music styles. Overviews of these topics can be found in Gouyon et al. [12] and McKinney et al. [14]. Our work follows the Collins' suggestion [5] to build beat trackers that embody knowledge of specific musical styles.

3 PROBLEM FORMULATION

The Cuban clave is a small collection of rhythms found embedded in virtually all Cuban music. Clave is a repeated syncopated rhythmic pattern that is often explicitly played, but often only implied; it is the essence of periodicity in Cuban music. An instrument also named clave (a pair of short sticks hit together) usually plays this repeating pattern. Clave is found mainly in two forms: rumba clave and son clave (see Figure 1). The study of timing requires estimating how tempo changes over time (what is called the tempo curve) as well as locating the individual "notes" of the clave. Ground truth data has been gathered by having an expert percussionist with Afro-Cuban experience tap the clave on a laptop's built-in microphone. Custom sample-accurate tap detection/logging software automatically timestamps the taps.

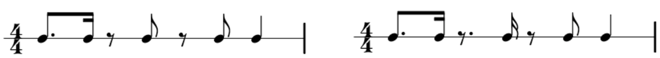

Figure 1. Son (left) and rumba (right) clave

Recordings of Afro-Cuban music challenge existing state-of-the-art beat-tracking algorithms. The two main reasons are 1) the complex and dense rhythm 2) the lack of regular approximately isochronous pulses. Figure 2 shows how two recent state-of-the-art beat-tracking systems are not precise enough to generate an accurate tempo curve for an Afro-Cuban recording. The ground truth tempo curve (calculated from the expert human clave taps) is shown at the top left. The two bottom plots show the tempo curves of BeatRoot [9] and a beat tracker using dynamic programming proposed by Ellis [10]. The top right tempo curve is the one detected by the method proposed in this paper. The method is specifically designed to take into account clave as the rhythmic backbone. The plots in the figure are shown in order to motivate the proposed approach. The comparison is not fair, as the other algorithms are more generally applicable and designed with different assumptions, but in any case it demonstrates the need for a domain-specific method to deal with these recordings.

4 DATA PREPARATION

It is common for Afro-Cuban songs to begin with just the sound of the clave for one or two repetitions to establish the initial tempo. However as other instruments (both percussive and pitched) and voices enter the mix the sound of the

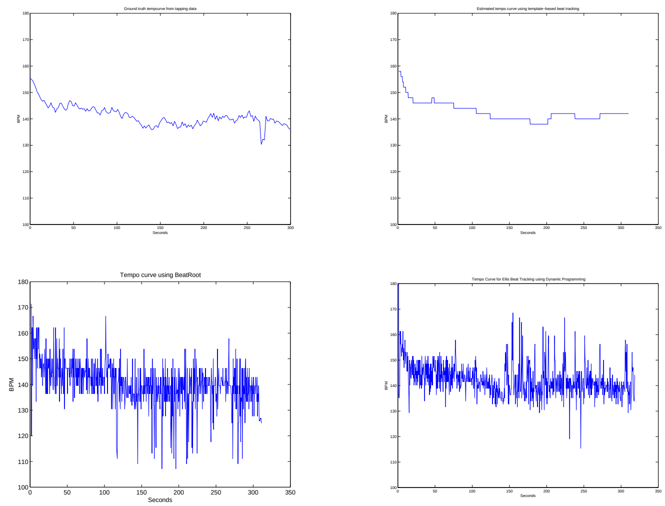

Figure 2. TempoCurves

clave tends to become masked. The first step of data preparation is to enhance the sound of the clave throughout the song using a matched filter approach. In addition onset detection is performed.

4.1 Clave enhancement using Matched-Filtering

A matched filter detects or enhances the presence of an *a priori* known signal within an unknown signal. Its impulse response is simply a time-reversed copy of the known signal, which in our case is the beginning portion of one isolated note of clave. The clave instrument affords very little timbral variety, in other words, every note of clave in a given recording sounds substantially like all the others, so a matched filter made from any single note (frequently easily obtained from the beginning of the song) will enhance the presence of the clave throughout the song and suppress the remaining signal. One free parameter is the filter order, i.e., the duration of the segment of the clave note; in each case we selected a "good" matched filter experimentally by listening to the output of different configurations. All the curves in Figure 2 and results in this paper have been calculated on audio signals output by matched filtering.

4.2 Onset detection

Onset detection aims at finding the starting time of musical events (e.g. notes, chords, drum events) in an audio signal. However, polyphonic music poses an increased challenge since nominally simultaneous notes might be spread over tens of milliseconds, turning the definition of onsets ambiguous. Similarly, it is hard to define a precise onset time for sounds with slow attacks.

In a recent tutorial article, Dixon revisited the problem of onset detection [8], where a number of onset detection algorithms were reviewed and compared on two datasets. This study was itself based on a previous article in which a theoretical and empirical comparison of several state-of-the-art onset detection approaches is presented [3]. Following the findings and results in [8], the approach used in this work is based on the use of the spectral flux as the onset detection function, defined as:

$$SF(n) = \sum_{k=0}^{N/2} H(|X(n,k)| - |X(n-1,k)|) \qquad (1)$$

where $H(x) = \frac{x+|x|}{2}$ is the half-wave rectifier function, $X(n,k)$ represents the k-th frequency bin of the n-th frame of the power magnitude (in dB) of the short time Fourier Transform, and N is the corresponding Hamming window size. For the experiments performed in this work a window size of 46 ms (i.e. $N = 2048$ at a sampling rate $f_s = 44100$ Hz) and a hop size of about 11ms (i.e. 512 samples at $f_s = 44100$ Hz) are used. The onsets are subsequently detected from the spectral flux values by a causal peak-picking algorithm that attempts to find local maxima as follows. A peak at time $t = \frac{nH}{f_s}$ is selected as an onset if it fulfills the following conditions:

1. $SF(n) \geq SF(k) \;\; \forall k \; : n - w \leq k \leq n + w$
2. $SF(n) > \frac{\sum_{k=n-mw}^{n+w} SF(k)}{mw+w+1} \times thres + \delta$

where $w = 6$ is the size of the window used to find a local maximum, $m = 4$ is a multiplier so that the mean is calculated over a larger range before the peak, $thres = 2.0$ is a threshold relative to the local mean that a peak must reach in order to be sufficiently prominent to be selected as an onset, and $\delta = 10^{-20}$ is a residual value to avoid false detections on silent regions of the signal. All these parameter values were derived from preliminary experiments using a collection of music signals with varying onset characteristics.

In order to reduce the false detection rate, the onset detection function $SF(n)$ is filtered using a Butterworth filter:

$$H(z) = \frac{0.1173 + 0.2347z^{-1} + 0.1174z^{-2}}{1 - 0.8252z^{-1} + 0.2946z^{-2}} \qquad (2)$$

In order to avoid phase distortion (which would shift the detected onset time away from the $SF(n)$ peak) the signal is filtered in both the forward and reverse directions. The result has precisely zero-phase distortion, the magnitude is the square of the filter's magnitude response, and the filter order is double the order of the filter specified in Equation 2.

5 TEMPLATE-BASED TEMPO TRACKING

We propose a new method to deal with the challenges of beat tracking in Afro-Cuban music. The main idea is to use domain specific knowledge, in this case the clave pattern, directly to guide the tracking. The method consists of the following four basic steps: 1) Consider each detected onset time as a potential note of the clave pattern. 2) Exhaustively consider every possible tempo (and clave rotation) at each onset by correlating each of a set of clave-pattern templates against an onset strength envelope signal beginning at each detected onset. 3) Interpret each correlation amount as a score for the corresponding tempo (and clave rotation) hypothesis. 4) Connect the local tempo and phase estimates to provide a smooth tempo curve and deal with errors in onset detection, using dynamic programming.

The idea of using dynamic programming for beat tracking was proposed by Laroche [13], where an onset function was compared to a predefined envelope spanning multiple beats that incorporated expectations concerning how a particular tempo is realized in terms of strong and weak beats; dynamic programming efficiently enforced continuity in both beat spacing and tempo. Peeters [15] developed this idea, again allowing for tempo variation and matching of envelope patterns against templates. An approach assuming constant tempo that allows a simpler formulation at the cost of more limited scope has been described by Ellis [10].

5.1 Clave pattern templates

At the core of our method is the idea of using entire rhythmic patterns (templates) for beat tracking rather than individual beats. First we construct a template for each possible tempo. We take the ideal note onset times in units of beats (e.g., for rumba clave, the list 0, 0.75, 1.75, 2.5, 3) and multiply them by the duration of a beat at each tempo, giving ideal note onset times in seconds. We center a Gaussian envelope on each ideal note onset time to form the template. The standard deviation (i.e., width) of these Gaussians is a free parameter of this method. Initial results with a constant width revealed a bias towards higher tempi, so widths are specified in units of beats, i.e., we scale the width linearly with tempo. Better results were obtained by making each template contain multiple repetitions of the clave, e.g., three complete patterns. Figure 3 shows a visual representation of the template rotations for all considered tempi.

With a 5-note clave pattern, any given note played by the clave could be the 1st, 2nd, 3rd 4th or 5th note of the pattern. Therefore we make templates for all "rotations" of the clave i.e for the repeating pattern as started from any of the five notes. For example, the rotation 0 of rumba clave is [0, 0.75, 1.75, 2.5, 3], and rotation 1 (starting from the second note) is [0.75 1.75 2.5 3 4] - 0.75 = [0 1 1.7 2.25 3.25]. Time 0 always refers to the onset time of the current note.

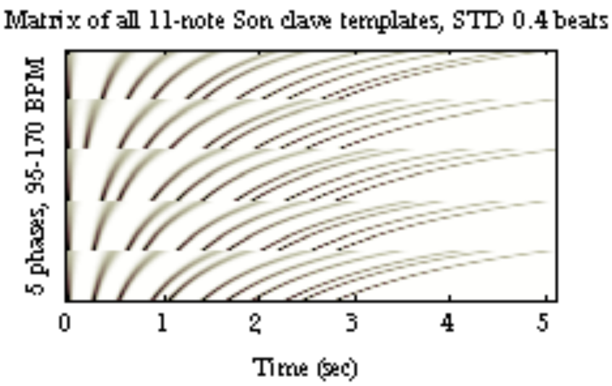

Figure 3. Clave Templates for all rotations and tempi

We correlate these templates with segments of an onset strength envelope (in our case, simply the total energy in each 1024-sample window of the matched filter output) beginning at the time of each detected onset. We interpret the correlation amount between the onset strength signal $O(t)$ and a template $T_{j,k}(t)$ with tempo j and rotation k as the strength of the hypothesis that the given onset is the given note of clave at the given tempo. Figure 4 depicts this process for some tempi and rotations and the corresponding scores. We exhaustively compute these correlations for every tempo j (e.g,. from 95 to 170 BPM in 1 BPM increments), for all five rotations of the clave pattern k , for every detected onset i at time t_i to produce a *score grid*:

$$score(i,j,k) = \sum_{t=0}^{LT_{j,k}-1} T_{j,k}(t) \times O(t_i + t) \quad (3)$$

where $LT_{j,k}$ is the length of template $T_{j,k}$.

5.2 Rotation-blind dynamic programming

It is trivial to look at a given onset, pick the tempo and rotation with the highest score, and call that the short-term tempo estimate. However, due to the presence of noise, inevitable onset detector errors, and the matched filter's far-from-perfect powers of auditory source separation, simply connecting these short-term tempo estimates does not produce a usable estimate of the tempo curve. Better results can be achieved by explicitly discouraging large tempo changes. We use dynamic programming [2] as an efficient means to estimate the best tempo path (i.e., time-varying tempo). In the next section we will consider the rotations of the template; for now let the "rotation-blind" score be:

$$scoreRB(i,j) = max(score(i,j,k)) \; k : 1..5 \quad (4)$$

We convert each score $scoreRB$ to a cost $C_{i,j}$ with a linear remapping so that the highest score maps to cost 0 and the lowest score maps to cost 1. We define a *path* P as a sequence of tempo estimates (one per onset), so that $P(i)$ is P's estimate of the tempo at time t_i. Our algorithm minimizes the *path cost* PC of the length n path P:

$$PC(P) = \sum_{i=0:n-1} C_{i,P(i)} + \sum_{i=0:n-2} F(P(i), P(i+1)) \quad (5)$$

where $F(tempo_1, tempo_2)$ is a "tempo discontinuity cost function" expressing the undesirability of sudden changes in tempo. Currently F is simply the absolute difference of the two tempi. Dynamic programming can efficiently find the lowest-cost path from the first onset to the last because the optimal path up to any tempo at time t_i depends only on the optimal paths up to time t_{i-1}. We record both the cost $PC(i,j)$ and the previous tempo $Previous(i,j)$ for the best path up to any given onset i and tempo j.

5.3 Rotation-aware dynamic programming

Now we will extend the above algorithm to consider rotation, i.e., our belief about which note of clave corresponds to each onset. Now our cost function $C_{i,j,k}$ is also a function of the rotation k. Our path tells us both the tempo $P_{tempo}(i)$ at time t_i and also the rotation $P_{rot}(i)$, so we must keep track of both previous $Previous_{tempo}(i,j)$ and $Previous_{rot}(i,j)$ (corresponding to the best path up to i and j). Furthermore, considering rotation will also give us a principled way for the path to skip over "bad" onsets, so instead of assuming that every path reaches onset i by way of onset $i-1$ we must also keep track of $Previous_{onset}(i,j)$.

The key improvement in this algorithm is the handling of rotation. Rotation (which indexes the notes in the clave pattern) is converted to *phase*, the proportion (from 0 to 1) of the way from one downbeat to the next. (So the phases for the notes of rumba clave are [0, 0.1875, 0.4375, 0.625, 0.75]). The key idea is predicting what the phase of the next note "should be": Given phase ϕ_1 and tempo j_1 for onset i_1, a candidate tempo j_2 for onset i_2, and the time between onsets $\Delta T = t_2 - t_1$, and assuming linear interpolation of tempo during the (short) time between these nearby onsets, we can use the fact that tempo (beat frequency) is the derivative of phase to estimate the phase $\hat{\phi}_2$:

$$\hat{\phi}_2 = \phi_1 + \Delta T \times ((j_1 + j_2)/2)/4 \times 60 \quad (6)$$

The 4×60 converts from BPM to bars per second.

Now we can add an extra term to our cost function to express the difference between the predicted phase $\hat{\phi}_2$ and the actual phase ϕ_2 corresponding to the rotation of whatever template we're considering for the onset at time t_2 (being careful to take this difference modulo 1, so that, e.g., the difference between 0.01 and .98 is only 0.03, not 0.97). We'll call this phase distance the "phase residual" R, and add the term $\alpha * R$ to our cost function.

	LP	CB	CH	PD	LPWT	PDWT
RB	40	26	22.9	63.1	39.96	62.7
RA	1.75	11	1.54	3.10	2.043	57.9

Table 1. RMS (in BPM) results for tempo curve estimation

Now let's consider how to handle "false" detected onsets, i.e., onsets that are not actually notes of clave. For onset n, we consider not just onset $n-1$ as the previous onset, but every onset i with $t_i > t_n - K$, i.e., every onset within K seconds before onset n, where K is set heuristically to 1.5 times the largest time between notes of clave (one beat) at the slowest tempo. We introduce a "skipped onset cost" β and include $\beta \times (n-i-2)$ in the path cost when the path goes from onset i to onset n.

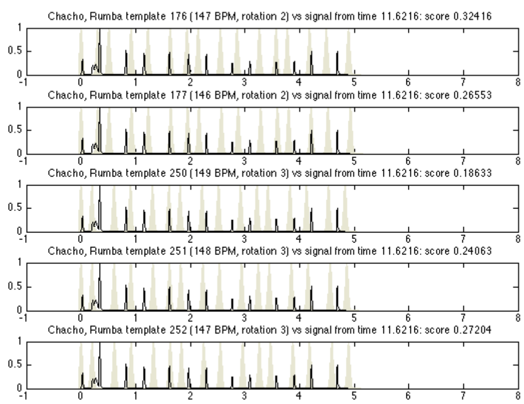

Figure 4. Matching different templates to onset strength envelope

Table 1 shows the Root-mean-square (RMS) error between the ground truth tempocurve and the tempocurves estimated by the rotation-blind (RB) and rotation-aware (RA) configurations of our method. In all cases the rotation-aware significantly outperforms the rotation-blind method (which usually tracks correctly only parts of the tempo curve). The first three recordings (LP, CB, CH) have rumba-clave and the fourth piece (PD) has son-clave. [1] The last two columns show the results when using the "wrong" template. Essentially when the template is not correct the matching cost of the beat path is much higher and the tempo curve estimation is wrong. Figure 5 shows the score grid for the rotation-blind (top) and rotation-aware (bottom) configurations overlaid with the estimated and ground truth tempocurves.

[1] LP: La Polemica, Rumba Caliente 88, Los Munequitos de Matanzas
CB: Canter Bueno, El Callejon De Los Rumberos, Yoruba Andabo
CH: Chacho, Los Munequitos de Matanzas, Cuba: I am time
PD: Popurrit De Sones Orientales

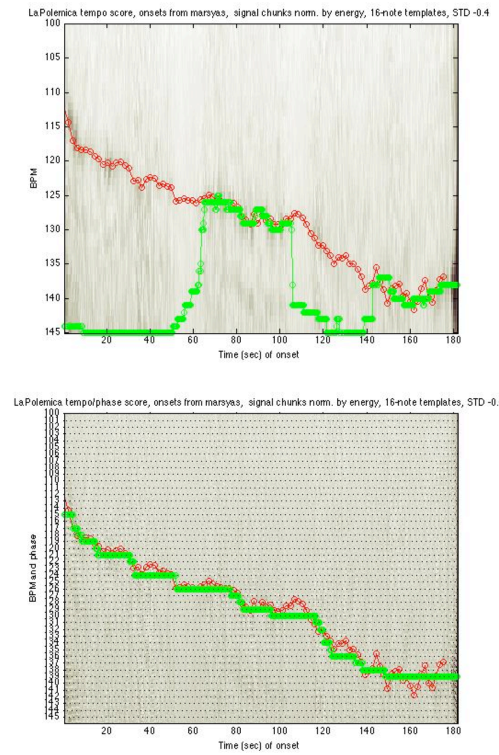

Figure 5. Rotation-blind (top) and rotation-aware (bottom) beat tracking

6 BAR-WRAPPING VISUALIZATION

A clave performance typically consists of about 625-1000 "notes". Simply plotting each point along a linear time axis would require either excessive width, or would make the figure too small to see anything; this motivates bar wrapping. Conceptually, we start by marking each event time (in this case, each detected onset) on a linear time axis. If we imagine this time axis as a strip of magnetic tape holding our recording, then metaphorically we cut the tape just before each downbeat, so that we have 200 short pieces of tape, which we then stack vertically, so that time reads from left to right along each row, and then down to the next row, like text in languages such as English. Each of these "strips" is then stretched horizontally to fill the figure width, adding a tempo curve along the right side to show the original duration of each bar. Figure 6 depicts the times of our detected onsets for *LP* with this technique. The straight lines show the theoretical clave locations. By looking at the figure one can notice that the 5th clave note is consistently slightly later than the theoretical location. This would be hard to notice without precise estimation of the tempocurve.

Rotation-aware dynamic programming is used to find the downbeat times An explicit downbeat estimate occurs whenever the best path includes a template at rotation 0. But there

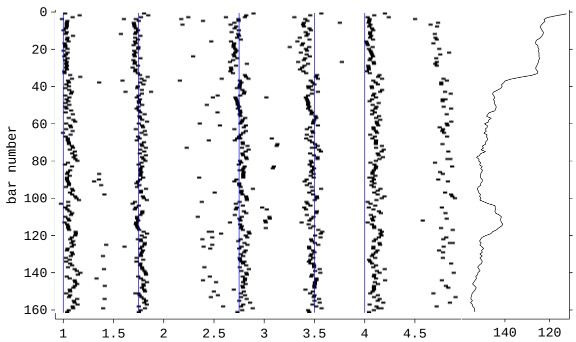

Figure 6. Bar-wrapping visualization

might not be a detected onset at the time of a downbeat, so we must also consider implicit downbeats, where the current onset's rotation is not 0 but it is lower than the rotation of the previous onset in the best path. The phase is interpolated to estimate the downbeat time that "must have occurred" between the two onsets.

7 CONCLUSIONS

We proposed a beat-tracking method specifically designed for Afro-Cuban music. The key idea is to use the clave as a template for beat tracking using a dynamic programming approach. The method is able to track timing correctly in challenging recordings for which beat trackers designed for popular music fail. Using the estimated tempocurve it is possible to visualize microtiming patterns using the technique of bar-wrapping. There are many future work directions. Rhythmic analysis can be used to categorize recordings into different styles and possibly identify particular artists of even percussionists. We also plan to apply the method to more recordings and continue working with ethnomusicologists and performers interested in exploring timing. It is our belief that our template-based rotation-aware formulation can also be applied to popular music by utilizing different standard drum patterns as templates. We are interested in collaborating with other researchers who would like to explore this possibility. All the code implementing the method can be obtained by emailing the authors.

8 REFERENCES

[1] I. Antonopoulos et al. Music retrieval by rhythmic similarity applied on greek and african traditional music. In *Proc. Int. Conf. on Music Information Retrieval(ISMIR)*, 2007.

[2] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[3] J.P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M.B. Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1047, September 2005.

[4] J. Bilmes. Timing is of the essence: Perceptual and computational techniques for representing, learning and reproducing timing in percussive rhythm. Master's thesis, Massachusetts Institute of Technology, 1993.

[5] N. Collins. Towards a style-specific basis for computational beat tracking. In *Int. Conf. on Music Perception and Cognition*, 2006.

[6] D. Cooper and I. Sapiro. Ethnomusicology in the laboratory: From the tonometer to the digital melograph. *Ethnomusicology Forum*, 15(2):301–313, 2006.

[7] O. Cornelis et al. Problems and opportunities of applying data and audio mining techniques to ethnic music. *Journal of Intangible Heritage*, (in press), 2007.

[8] S. Dixon. Onset detection revisited. In *Proc. International Conference on Digital Audio Effects (DAFx)*, Montreal, Canada, 2006.

[9] S. Dixon. Evaluation of audio beat tracking system beatroot. *Journal of New Music Research*, 36(1), 2007.

[10] D. Ellis. Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1), 2007.

[11] J. Futrelle and S. Downie. Interdisciplinary communities and research issues in music information retrieval. In *Proc. Int. Conf. on Music Information Retrieval(ISMIR)*, 2002.

[12] F. Gouyon et al. An experimental comparison of audio tempo induction algorithms. *IEEE Trans. on Audio Speech and Language Processing*, 14(5), 2006.

[13] J. Laroche. Efficient tempo and beat tracking in audio recordings. *Journal of the Audio Engineering Society*, 51(4):226–233, 2003.

[14] M.F. McKinney et al. Evaluation of audio beat tracking and music tempo extraction algorithms. *Journal of New Music Research*, 36(1), 2007.

[15] G. Peeters. Template-based estimation of time-varying tempo. *EURASIP Journal of Advances in Signal Processing*, 2007.

[16] A. Schloss. *On the Automatic Transcription of Percussive Music: From Acoustic Signal to High-Level Analysis*. PhD thesis, Stanford University, 1985.

[17] G. Tzanetakis, A. Kapur, W.A. Schloss, and M. Wright. Computational ethnomusicology. *Journal of Interdisciplinary Music Studies*, 1(2):1–24, 2007.