Beat Tracking with Dynamic Programming

Daniel P.W. Ellis
LabROSA, Dept. of Electrical Engineering
Columbia University, New York NY 10027 USA
dpwe@ee.columbia.edu

Abstract

There are many applications for which we would like to be able to track the 'beat' of a piece of recorded music – analogous to a listener's foot-tapping. This paper describes our beat-tracking system, which operates by first estimating a global tempo (via autocorrelation of an 'onset strength' signal), then using dynamic programming to find the best sequence of beat times through the whole piece that both places beats on moments of high 'onset strength', as well as maintaining a spacing between beats that agrees with the global tempo. This system has been submitted to the 2006 MIREX Audio Tempo Extraction and Audio Beat Tracking competitions.

Keywords: Tempo Extraction, Beat Tracking, Autocorrelation, Dynamic Programming

1. Introduction

Finding the beats in a musical recording is an interesting challenge and can form the basis of a number of applications, such as automatic accompaniment, transcription, computer-assisted audio editing, and music similarity. In this paper we describe our beat tracking system, which was in fact developed as part of our cover song detection system (since beat-synchronous features are a good way to normalize away tempo variations between different versions of a song) [1].

Evaluating systems for beat tracking (and hence tempo extraction) is complicated by the fact that different 'levels' in a hierarchy of beats may be considered as the main beat by different listeners. For MIREX, this problem has been neatly solved by collecting actual human tapping data for the test database [3]. Our system has been tuned with the 20 training samples released for the MIREX-06 tempo and beat evaluation. Each sample consists of 30 s of audio (from a range of styles and genres) along with the tap-instants of 40 different subjects who were played the samples. There are usually two different beat periods represented in the user data, where one is 2 or 3 times faster than the other. Beat trackers are evaluated by their ability to match the *entirety* of the subjective ground truth data, which means in practice choosing one tempo, and accepting that matches will not be high for subjects who choose a different tempo.

The following section describes our system, then section 3 reports performance on the evaluation data.

2. System Overview

This section describes each module in our system.

2.1. Onset Strength Signal

The first stage of processing is to convert the audio into a one-dimensional function of time at a lower sampling rate that reflects the strength of onsets (beats) at each time. We based this on the front-end on the one described in [2]. A log-magnitude 40-channel Mel-frequency spectrogram is calculated for 8 kHz downsampled mono versions of the original recording with a 32 ms window and 4 ms hop between frames. The first-order difference along time in each frequency channel is half-wave rectified (to leave only onset information) then summed across frequency. This 'onset strength' envelope is high-pass filtered with a 3 dB point at 0.01 rad/samp to remove d.c. offset (corresponding to global gain variations in the original signal, prior to the log operation).

2.2. Tempo Estimation

The onset strength for the entire signal is autocorrelated out to a maximum lag of 4 s (i.e. 1000 samples at our 4 ms sampling period). This raw autocorrelation is then scaled by a window to capture the intrinsic bias of listeners towards a particular range of tempi; in this way, the multiple peaks typical of the autocorrelation of a period signal can be resolved to a single dominant peak. Our window is a Gaussian on a log-time axis, and is characterized by its center (the BPM at which it is largest), and its half-width (the sigma of the Gaussian, in units of octaves on the BPM scale, since the axis is in fact logarithmic). We tuned these parameters by hand to give the best agreement with the subjective data provided for the MIREX competition; the best center was at 120 BPM (agreeing with perceptual results for the preferred tapping rate of subjects), and the width was 1.4 octaves (i.e. the window has fallen to 60.7% of its peak value at 323 BPM and at 44.6 BPM). The lag corresponding to the largest value in this autocorrelation was reported as the strongest tempo.

The competition also requires a second tempo, which is scored against the second-most popular tempo observed in

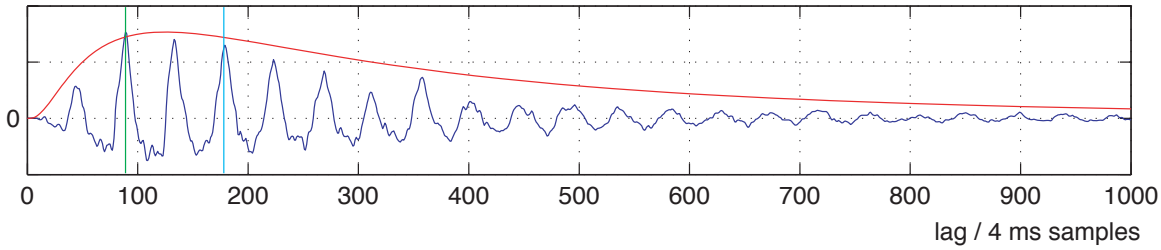

Figure 1. Autocorrelation of the 30 s Bragg excerpt. The log-time Gaussian weighting window is shown overlaid, and the primary (89 samples = 356 ms = 168.5 BPM) and secondary (178 samples = 712 ms = 84.3 BPM) periods are shown by vertical lines.

the subject studies. To find this, we searched the autocorrelation peaks closest to 0.33, 0.5, 2, and 3 times the strongest tempo. Whichever of these was largest was reported as the secondary tempo. The weight of the stronger tempo, also requested for the evaluation, was simply taken as the value of the largest autocorrelation peak divided by the sum of the peaks at both reported tempi. This value does not affect the evaluation metric as defined, so no effort was made to match it more closely to the ground-truth values in the training set.

Figure 1 shows the autocorrelation for example 2 of the training set, a 30 s excerpt from "New England" by Billy Bragg (vocals and guitar only).

2.3. Beat Tracking

The best BPM is passed to the beat tracking module, which attempts to find a sequence of beat times that all correspond to large values in the onset waveform, The onset signal is first smoothed by convolving with a Gaussian window whose half-width is 1/32 of the specified beat period. Then the best cumulative score is found for beat sequences ending at every possible time sample. This is done efficiently with dynamic programming: for each time point, a search is done over a range 0.5 to 2 beat periods into the past. The best cumulative score at each time in that window is scaled by a 'transition weight', which is another log-time gaussian, centered on the ideal time (one beat into the past), and with a width specified as a parameter of the system – a narrower width makes it harder for any beat to deviate far from the specified target period. The largest scaled value is chosen as the best predecessor beat for the current time, and added to the current onset signal value to give the best cumulative score for this time. The time of the preceding beat is also recorded. At the end of the excerpt, the best cumulative score within a couple of beats of the end is chosen, then traced back through all the preceding-time records to get the entire sequence of beats that gave rise to that best score.

In order to keep a balance between past scores and local match, the best score at the preceding beat is actually scaled by a constant a little smaller than 1 before being added to the current beat's score. This constant is a second parameter to the system: the smaller it is, the more weight is placed on achieving a good local match versus choosing a good history. This is a second parameter to the algorithm.

Figure 2 shows an example of the beats found in the first 15 s of the Bragg excerpt. The advantage of dynamic programming is that it effectively searches all possible sets of beat instants, since it is guaranteed to find the best-scoring sequence up to any point. This allows the best global beat sequence to be found, even if it involves some locally-poor matching, for instances beats that occur during silence or uninflected sustained notes.

3. Evaluation

The MIREX-06 audio tempo and beat evaluations make available 20 training excerpts, for which ground-truth tempi and beat times are given, as described above. The principal evaluation metrics are also defined. Thus we were able to optimize some of the tuning parameters of our system to maximize performance on this set.

For tempo extraction, the middle tempo of 120 BPM and the window width of 1.4 octaves were chosen this way. With these settings, using the weighted both-tempo matching score defined for the evaluation (which rewards identifying either or both of the two tempo levels in proportion to their observed prevalence among subjects) our system achieves a score of 77% correct.

For beat tracking, we tuned the transition window width and forgetting factor to maximize the evaluation score on the training data. The optimal window width weighted the extreme values in the preceding beat window (at 0.5 and 2 periods earlier) both at $0.17\times$ the central peak. The best forgetting factor decayed the history by a factor of 0.8 at each step. With these settings, and using the defined primary metric, our system scored 56.6% correct. Note, however, that 100% is not obtainable due to inconsistencies between the ground-truth listeners – no single beat tracker output can agree well with both of the two levels of beats typically found among the training data.

Since beats tracked at a slower tempo run the risk of being one half-cycle out of phase (i.e. picking beats 2 and 4 instead of 1 and 3), we thought that always using the faster of the two identified tempo (i.e. picking beats 1, 2, 3, and 4)

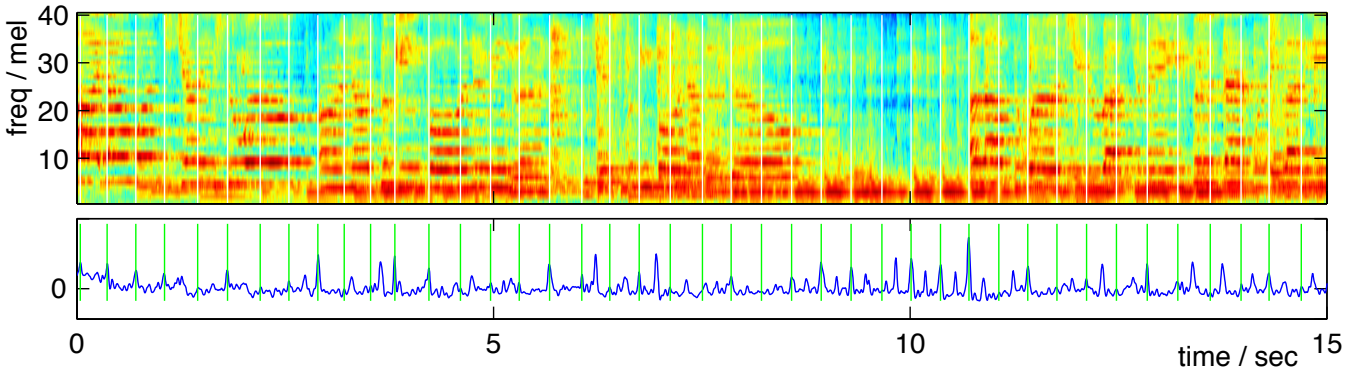

Figure 2. Excerpt showing the Mel-scale spectrogram (top pane), and the smoothed onset strength evelope (lower pane) for the first 15 s of the Bragg excerpt. Chosen beats are shown as vertical divisions. Notice the extensive syncopation (strong onsets midway between perceived beats).

might be a safer option. However, we found we got better scores by using the primary BPM value (i.e. the largest peak in the weighted correlation), whether or not it was faster than the alernative.

4. Conclusions

Using a relatively simple onset detection scheme, and assuming more-or-less fixed tempo throughout a piece, we find simple autocorrelation, suitably weighted to simulate a perceptual bias, does well at predicting perceived tempo. Beat tracking that uses dynamic programming to search all possible beat sequences does well, even when there are voids where beats have to be filled in, since the location of future as well as past beats can affect their position. Future work includes modifying the beat tracker to take account of slow but systematic changes in tempo, and perhaps a system that extracts multiple

Acknowledgments

This work was supported by the Columbia Academic Quality Fund, and by the National Science Foundation (NSF) under Grant No. IIS-0238301. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the NSF. We are very grateful to Martin McKinney and colleagues for collecting and making available the ground-truth data for this evaluation.

References

[1] D. P. W. Ellis. Identifying 'cover songs' with beat-synchronous chroma features. In *MIREX-06 Abstracts*, 2006.

[2] T. Jehan. *Creating Music by Listening*. PhD thesis, MIT Media Lab, Cambridge, MA, 2005.

[3] M. F. McKinney and D. Moelants. Extracting the perceptual tempo from music. In *Proc. Int. Conf. on Music Info. Retr. ISMIR-04*, pages 146–149, Barcelona, 2004.