ECS7006 Coursework 1 – Beat Tracking

Claudiu Andrei - 180332423

1 Introduction

The task to complete in this assignment revolves around the idea of implementing and exploring rhythm annotating/beat tracking methodologies. The goal is to produce an output that maps the tactus, known as the primary metric level for rhythm, for a given Ballroom Dancing style track. The proposed solution is built on the 2007 work by P.W. Ellis in the "Journal of New Music Research" [2], which makes use of the dynamic programming approach. Additionally, this solution references the textbook used for this module [3] which is used to set a boundary for certain concepts covered through this report.

2 Background

Before attempting to resolve the problem of beat tracking it is important to define the meaning of the term and what it entails. It is widely accepted that the term rhythm can be referred to as the perceived composition of: *tempo* (in its most basic definition the pace at which a passage of music shall be played), *timing* (musical notation to specify how many beats or pulses are contained in each measure or bar), *pulse* (a series of uniformly spaced beats which can be measured in bpm) and *metre* (refers to the regularly recurring patterns and accents such as bars and beats). The concept of beat is derived from the pulse, as this is what allows to identify the tactus, or rate at which one would tap along with the music. During the last couple of decades, a lot of progress has been made, with increasingly more accurate approaches and technologies used involving deep learning methodologies [1].

Since in music information retrieval work the rhythm is usually derived from onset times, the task of onset detection is trivial for being able to estimate other musical features as well as metrical and rhythm properties. Approaching the onset detection problems requires having pre-defined functions known as novelty functions to which post-processing steps are applied. Methods which achieve this goal include energy -based, phase-based, complex domain and spectral based onset detection [3].

3 Implementation

This report explores the implementation of rhythm tracking techniques as presented in [2], a system driven by an objective function seeking maximization of the onset strength at every hypothesized beat time. Here the onset strength is to be derived from source with some system and the consistency of the inter-onset-interval by assuming a pre-existing constant tempo. Assumptions on the positions of the strongest notes onsets must be made such that these go along with a constant tempo. The system is developed using Python and takes advantage of the libraries Numpy and Librosa.

3.1 Onset Detection Function

The onset detection function is implemented as per mentioned procedure in [3], where the STFT of the source is calculated with window size of 1024, sample size of 512 and sampling rate of 22050 Hz. Following the procedure, a logarithmic compression with γ factor of 100 is applied, preceding a first derivative calculation and a half-wave rectification step, taking care to exclude drops in energy. The sum across all frequencies is then calculated and a local averaging with window size M = 10 performed. The final step of the onset detection function is a simple normalization step.

3.2 Estimating Tempo

An existing pre-estimated tempo is assumed for this solution, which is achieved thanks to the automation provided by the function librosa.bea.tempo provided by the Librosa library. This passes the onset detection function defined in the previous section as an argument, resulting in a BPM estimation.

3.3 Beat Estimation with Dynamic Programming

The onset detection function and tempo estimation sections previously described have been defined so to follow the Ellis implementation, where the generated beat sequence corresponds to patterns present in the onsets. The following objecting function expresses this precisely:

$$C\left(\{t_i\}\right) = \sum_{i=1}^{N} O\left(t_i\right) + \alpha \sum_{i=2}^{N} F\left(t_i - t_{i-1}, \tau_p\right)$$

We subtract a consistency function F(Δt, τ) weighted by a consistency factor α

$$F(\Delta t, \tau) = -\left(\log \frac{\Delta t}{\tau}\right)^2$$

$$C^*(t) = O(t) + \max_{\tau=0 \ldots t} \left\{\alpha F\left(t-\tau, \tau_p\right) + C^*(\tau)\right\}$$

$$P^*(t) = \arg \max_{\tau=0 \ldots t} \left\{\alpha F\left(t-\tau, \tau_p\right) + C^*(\tau)\right\}$$

To achieve the optimization of the objective function C* and P* are calculated for each step t. The largest value of C* represent the beat instant tN. The sequence { t_i } (optimal beat sequence) is given after the back tracking step, searching backwards through P*, while the preceding beat time t_{N-1} is P*(t_N). The dynamic programming allows to break the exponential in N problem into sub-problems, resulting into a linear-time execution. The loop is executed until the start of the signal is reached and P*(t) = 0. The last step comprises outputting the beat sequence with the most optimal core, composed of onset detection indexes leading to time stamps.

3.4 Downbeat Estimation and Metre Detection

For the following step after determining a beat sequence, an array with multiple hypotheses for both the measure (either 3/4 or 4/4) and the beginning of the first beat within the measure is created. In order for this to be obtained a subset of beat sequence indexes is extracted, for which the values are extracted every forth or third step, beginning from the 1st, 2nd, 3rd or 4th position for a 4/4 measure or 1st, 2nd and 3rd for a 4/4 measure, resulting in 7 possible indices. This is used to extract a subset of the novelty function for each hypothesis. For each of these the mean is calculated and the highest value is returned, producing a subset of beat sequence matching the hypothesis best.

4 Results

Results were evaluated using a ballroom dataset, consisting of 20 examples from the data set for each genre. The algorithm has generally performed well, excelling with the genre ChaChaCha and performing worse on Samba examples. In certain scenarios, downbeat tracking did not perform well, however when listening to the generated clicks, no particular errors appear evident in the tracking of bars. When comparing the data of the annotated set and the estimated one it is evident that in most cases the approach seems to be successful in tracking the beat of the source, however some less than optimal results are achieved for some genres such as Rumba-American.

5 Conclusions and limitations

The main limitation of the beat tracking procedure is its dependency on a single, predefined tempo $\hat{\tau}$. Using a small weighting parameter λ, the procedure may yield good beat tracking results even in the presence of local deviations from the ideal beat period $\hat{\delta}$. However, the presented procedure is not designed for handling music with slowly varying tempo or abrupt changes in tempo. Despite these limitations, the simplicity and efficiency of the dynamic programming approach to beat tracking makes it an attractive choice for many types of music [3]. One disadvantage of downbeat estimating is that it presupposes that the first estimated beats have a high probability of being accurately anticipated, which in most situations turns out to be a false hypothesis. Due to the peculiarities of the ballroom dance data set utilised in this assignment, the algorithm is hard programmed to only consider 3/4 or 4/4 examples.

References

[1] Matthew E.P. Davies and Sebastian Böck. "Temporal Convolutional Networks for Musical Audio Beat Tracking". In: *European Signal Processing Conference*. 2019.

[2] Daniel P.W. Ellis. "Beat tracking by dynamic programming". In: *Journal of New Music Research* 36.1 (2007), pp. 51–60. issn: 09298215. doi: 10.1080/09298210701653344.

[3] Meinard Müller. *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. 1st. Springer Publishing Company, Incorporated, 2015. isbn: 3319219448.

[4] Julius Richter. "Style-Specific Beat Tracking with Deep Neural Networks". Master Thesis. 2019.