Summary of Chapter 6 - Tempo and Beat Tracking

Adler, Tilman
21450674

Stumpf, Martin
21450903

12.01.2015

1 Introduction

This summary covers Chapter 6 of "Fundamentals of Music Processing" by Meinard Müller [**?**]. Its structure follows that of the chapter itself, i.e. beginning with feature extraction of note onsets by *energy*, *spectral information*, *phase* and *complex-domain information* respectively. We will then continue to Tempo Analysis by *Fourier Analysis*, *Autocorrelation* and *Predominant Local Pulse*. Finally we cover Beat Tracking and Adaptive Windowing.

2 Main Section

2.1 Introduction

Music tempo, while easily detected by humans, is not easily recognised by a computer. Not only can it be useful to have the tempo information itself. It can increase confidence in other parts of a detector, too.

The chapter at hand covers different ways to extract onsets (loosely speaking, the time a note is started to be played) as features and further analysis to extract tempo information and track a beat.

2.2 Feature extraction: Detection of Onsets

2.2.1 Energy based novelty function

The energy based approach uses the fact that with the onset of a note, the energy in the signal rises rapidly. This can be detected by first computing a *local energy function* from the signal. This results in a function describing the *energy distribution*. This function is then *derived*. The negative parts of the derivative are discarded (*half-wave rectification*). Small peaks which have been insignificant compared to large ones can be enhanced by *applying a logarithm*.

2.2.2 Spectral based novelty function

In energy based approaches noise can render small energy peaks invisible. Therefore a spectral based novelty is introduced. Here, a *Short Time Fourier Transform* (STFT) is applied to the signal to seperate frequencies.

The idea is, that a peak might still be detectable in a certain frequency band while being superimposed by noise in the others. To enhance small onsets logarithmic compression is applied to the spectrogram. Then the approach is similar to the energy based one. The frequency bins are *derived*, *half-wave rectified* and *summed up*. To reduce the baseline noise local average is computed and subtracted from the resulting function.

2.2.3 Phase based novelty function

Another approach to detect onsets is to use the phase of the STFT frequency coefficients. This phase does not change during a stable tone and changes rapidly and randomly during the noise-like sound (called transient) right after an onset.

An unstable phase can be detected by the *second-order derivative*. The absolut values of the derivative are then *summed up* across all frequency bins.

2.2.4 Complex-Domain novelty function

While the phase based approach only uses phase information, a change into the complex domain allows using the magnitude, too. At each timestep a expected value is *extrapolated* from the value of the previous magnitude and phase. An onset can be detected by *subtracting* the expected from the measured value and half-wave rectification.

2.3 Tempo Analysis

Based on the onsets as features tempo information about the musical piece can be derived. The chapter covers two approaches: a *Fourier Transform* based one and one using *autocorrelation*. Both yield a tempogram. A tempogram is similar to a spectrogram, but instead of musical tone frequencies it represents onset frequencies.

2.3.1 Fourier Tempogram

The tempogram in this approach is obtained by applying a STFT to a novelty function as described in 2.2. Similarly to overtones in music this yields "tempo harmonics" i.e. multiples of the detected onset frequency. The frequency axis in the tempogram now actually models frequencies of the onsets.

2.3.2 Autocorrelation Tempogram

Another way to retrieve the tempogram is by convolving the novelty function with itself in a local window. Regular onsets then overlap, yielding high values in the tempogram. Thus, strictly speaking the frequency axis in it now models *time-lag*. That is, it does not represent a frequency, but how many samples the function has to be shifted so that repetitions in it match themselves.

Therefore in order to be comparable to the Fourier-based approach the results have to be inverted and stretched across a logarithmic axis.

Another difference to the Fourier-based algorithm is that here, there are "tempo subharmonics" in the resulting tempogram. That is, for each base frequency there are multiple other frequencies with the lag value $l' = \frac{1}{n}l, n \in \mathbb{N}^+$.

2.3.3 Cyclic Tempogram

Analogous to chroma featuers in tone frequency analysis, one can define a *Cyclic Tempogram* to be a tempogram in which onset frequencies and their multiples ("tempo harmonics") are added together. This can be useful, for example, for music segmentation, if pitch cannot be used reliably.

As is the case in a chromagram, where the base tone is $A4 = 440\text{Hz}$, for a cyclic tempogram a base onset frequency τ has to be defined.

2.4 Beat and Pulse Tracking

Until now we have only detected the predominant onsets. The task which is tackled by *Beat Tracking* is to use the predominant tempo to reconstruct the actual beats. That includes those with a very light or nonexistent onset (pauses).

2.4.1 Predominant Local Pulse

To achieve this, the *Predominant Local Pulse* (PLP) function is created from the phase information included in the *Fourier Tempogram* as described in 2.3.1.

First, the frequency bin with the highest magnitude within a time-window is chosen. Then, the phase information of this bin is used to create a respective *windowed sinusoid*. All of those of all (overlapping) windows are *added up* and *half-wave rectified*.

One of the advantages of this method is, that it is pretty robust against changes in tempo. However where a sudden tempo change occurs the value of the PLP function becomes close to zero as badly aligned windowed sinusoids cancel each other out.

2.4.2 Beat Tracking by Dynamic Programming

In those cases where the beats of a song are strong and steady, i.e. there are no tempo changes, one does not need the flexibility of the PLP algorithm. Instead, using a rough given global tempo estimate, one can calculate the *estimated rough distance* between two beats.

Having fixed a beat one can now search for the optimal previous beat by utilizing a penalty function. This function is given the distance between the current and the previous beat as input. It then needs to penalize (positive and negative) deviations between the estimated rough distance and the given distance.

Finally, one can choose the set of length L of all beats, which maximizes the beat magnitudes Δ and at the same time minimizes the penalty functions P on all beats b_l in the set as follows:

$$\sum_{l=1}^{L} \Delta(b_l) - \lambda \sum_{l=2}^{L} P(b_l - b_{l-1}) \tag{1}$$

With λ being a arbitrary weight parameter.

This set can be computed very efficiently using dynamic programming.

2.4.3 Adaptive Windowing

The thusly detected beat information can be used to improve upon pitch detection by removing transient parts of a beat, which are typically pretty noisy, from the window. That means windows are no longer time dependent but beat dependent and not necessarily equally spaced and sized.

As the noisy part is removed the pitch detection becomes a lot clearer and optimally windows do not comprise more than one pitch.

3 Feedback

See lecture evaluation.