Real Time Beat Tracking using Novelty Curve

Team ID: 24746
Undergraduate members: Yash Sanghvi, Mehul Shah, Arunabh Ghosh and Krishna Reddy
Postgraduate member: Ashish Goyal
Supervisor: Prof. V. Rajbabu

***Abstract*— An alternative approach to track beats of a musical recording in real time is proposed here. Chunks of audio are obtained at a constant sampling rate and a corresponding subsampled function known as novelty curve is generated which is designed such that it peaks on note onsets and sudden bursts in energy. Using the novelty curve, tempo for a section of audio is estimated and the beat instants are detected and changed dynamically.**

I. INTRODUCTION

Automatic beat tracking of musical excerpts is an active area of research. The goal is to construct a computational algorithm capable of extracting the beat instants from a musical excerpt in real time and mimics the human *foot tapping* as closely as possible. Our algorithm can be broken down into three simple steps. Firstly, from the real time audio which is extracted one chunk at a time, we generate a novelty curve which is designed such that it peaks at note onsets and percussion instrument hits. This novelty curve is highly subsampled and is further processed directly to detect the beat instants. Secondly, we determine the tempo i.e. beat period from the novelty curve. Thirdly, the fixed number of beat instants of known periodicity are aligned in future time instants which are subject to change as time progresses.

The paper is divided into five sections. Section II deals with the choice of novelty curve and how it is generated. Section III would explain how the the tempo is estimated from the novelty curve. Section IV explains how the beat is aligned with respect to time and how they are subject to change as time progresses. Section V contains details of how the algorithm was implemented on an embedded device.

II. NOVELTY CURVE

Novelty curve is a mid level, subsampled representation of the the audio input. An audio novelty curve for this application should yield peaks at all note onsets and sudden energy bursts caused by strikes of percussion instruments. This curve is used directly in further processing to determine periodicity and beat alignment and hence the choice of novelty curve is essential to the beat tracking application. We chose the complex spectral method to obtain the novelty curve which can be described as follows.

A. Complex Spectral Method

This method combines both the phase and magnitude information of the spectrum of chunks of audio to determine the novelty curve. Let the chunk of audio of size N at time instant n be given by $\tau[n]$. We take the STFT of the audio chunk, $X(n, k)$ and let the phase of the STFT coefficient be given by $\phi(n, k) = arg\{X(n, k)\}$. It is expected for a stationary sinusoidal wave that the phase varies linearly over time and the magnitude to remain same over the chunks. Using these assumptions, the predicted FFT of the chunk at instant n $\hat{X}(n, k)$ is given by

$$X(\hat{n}, k) = |X(n-1, k)| e^{j\phi(\hat{n}, k)} \tag{1}$$

where

$$\phi(\hat{n}, k) = \phi(n-1, k) + (\phi(n-1, k) - \phi(n-2, k)) \tag{2}$$

. Thus the novelty curve at the instant n is given by the sum

$$O[n] = \sum_{k=0}^{N/2} |\Gamma(n, k) - \hat{\Gamma}(n, k)| \tag{3}$$

In a final step, we subtract the moving average of the novelty curve $\overline{O}[n]$ from $O[n]$and clamp the value to 0 if it is negative, i.e.

$$O[n] \leftarrow max(O[n] - \overline{O}[n], 0) \tag{4}$$

We have used chunks of size N = 1024 samples, sampled at a rate of 44.1kHz with 50% overlap between two chunks. Thus the sampling rate of the novelty curve is given by

$$f_s = \frac{44100}{1024 * 0.5} = 86.12Hz$$

Hz. As a final step, we linearly interpolate the novelty curve by a factor of 4 to obtain better accuracy during tempo estimation and beat alignment.

III. TEMPO ESTIMATION

We have now obtained the novelty curve form the audio chunks and we will use it to determine the tempo estimate of the latest T = 4 seconds of audio obtained. While determining the tempo we impose the constraint of *tempo coherence* i.e. there can be no sudden large jumps in the estimate of tempo. How this constraint is imposed is explained further in the section.

Firstly, we find out the autocorrelation sequence of the novelty curve. Then we find out the peaks in the sequence in the range $\frac{f_s}{3.5}$ and f_s as these values correspond to the tempo range 60 bpm to 210 bpm which is the general range of tempo for most of the musical pieces. The autocorrelation sequence maxima is estimated as the beat instant of the musical piece.

To impose tempo coherence, we keep track of the previous maxima of the autocorrelation sequence and search for the maxima only in a range +/- 5 samples around the previous maxima. Also we multiply the autocorrelation sequence with a gaussian window centered at the previous maxima and with a very small variance ($\approx f_s/8$).

Note that the maxima of the autocorrelation sequence is not directly taken as the period of the beat instants. Firstly we check whether a third of the autocorrelation maxima or a quarter of the autocorrelation maxima has higher periodicity by making the following comparision

$$M = max[A(\tau/4), A(\tau/3)] \quad (5)$$

where $A(n)$ represents the autocorrelation value of the index n and τ represents the autocorrelation sequenct maxima. If $A(\tau/4)$ is greater we proceed to find out whether the τ or its double is closer to $f_s/2$ (which corresponds to 120bpm). Similarly if $A(\tau/3)$ is greater we proceed to find out whether 1.33τ or 0.67τ is closer to $f_s/2$. Whichever factor is closer to $f_s/2$ is decided as the beat period for that window.

The idea behind the process is to find out a factor of the autocorrelation maxima which lies closer to 120bpm as the human tapping signal have tendency to be periodic around this value.

The tempo estimation process is done after every 0.4 seconds in the above manner using a window of the last 4 seconds of the novelty curve to interpolate and find out the autocorrelation sequence from.

IV. Beat Alignment

The main idea behind beat alignment is to choose a starting point in the past i.e. a note onset which may correspond to a beat instant and then use the tempo estimate to make predictions about the beat instants in the future.

Firsly, we take the same frame used to measure autocorrelation sequence in the last step and find the maxima of the novelty curve from its last quarter. Since we are looking at an instant from the past, we can safely assume that a beat instant was declared in its temporal neighborhood. We look for the closest declared beat instant and define the quantity χ as the ratio of the time between the closest beat instant and novelty curve maxima and the estimated beat period for that window. If χ is less than 0.1, we place the 20 beat instants placed at time equal to beat period apart and starting from the note onsets with exponentially decreasing confidence weights. Otherwise, if χ is greater than 0.1, we start placing the beat instants from the closest beat instant itself.

After each beat instant is declared we apply a correction method such that two beat instants which are too close to be distinguishable are not declared twice. We take the latest declared beat instant and take the time interval +/- 0.45τ around it, and place the beat instant at the weighted average of the interval with weights equal to the confidence weight of the instants of the interval and rest of the confidence weights of the rest of the instants are made to zero.

V. Implementation on Embedded Device

We implemented the algorithm on Raspberry Pi (3rd Generation) which has 1.2GHz 64-bit quad core ARMv8 CPU and 1Gb RAM. The algorithm was implemented in Python along with heavy use of numpy library to vectorize the code and hence decrease the running time. The links for the the output of the algorithm on Raspberry Pi on an easy and tough musical piece are as follows:

- `https://www.youtube.com/watch?v=k9B6H-6BzUo`
- `https://www.youtube.com/watch?v=a1sCO_B9X7k`

We used the song 'Like a Stone' by Audioslave as an easy musical piece because of its constant tempo and simple 4/4 meter. For a hard piece, we chose one of the challenge pieces (challenge piece number 7, team ID: 24690).

For creative output, we designed an audio-visual metronome with 4 LEDs changing colour at every beat instant and a faint buzzer to indicate the end/start of every measure.

References

[1] Daniel P.W. Ellis, "Beat Tracking by Dynamic Programming," Journal of New Music Research, 36 (1), 51-60, 2007.

[2] Peter Grosche and Meinard Mller, Extracting predominant local pulse in- formation from music recordings, PIEEE Transactions on Audio, Speech, and Language Processing, 19(6), 1688-1701, 2011.

[3] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies and M. B. Sandler, "A Tutorial on Onset Detection in Music Signals," in IEEE Transactions on Speech and Audio Processing, vol. 13, no. 5, pp. 1035-1047, Sept. 2005.

[4] Allen, Paul E., and Roger B. Dannenberg, "Tracking musical beats in real time," Proceedings of the International Computer Music Conference. Vol. 1990. International Computer Music Association San Francisco, CA, 1990.

[5] A. P. Klapuri, A. J. Eronen and J. T. Astola, "Analysis of the meter of acoustic musical signals," in IEEE Transactions on Audio, Speech, and Language Processing, vol. 14, no. 1, pp. 342-355, Jan. 2006.

[6] Stark, Adam M., Matthew EP Davies, and Mark D. Plumbley, "Real-time beat-synchronous analysis of musical audio," Proceedings of the 12th Int. Conference on Digital Audio Effects, Como, Italy. 2009.

[7] A. Klapuri, M. Davy and S. Hainsworth, "Beat Tracking in Musical Metre Analysis," Signal Processing Methods for Music Transcription, Springer, 2006, ch. 4, pp. 101-129.