Automated Musical Tempo Estimation with Neural-Network-Based Onset Detection

Nathan Stephenson

TJHSST Computer Systems Lab, 2020–2021

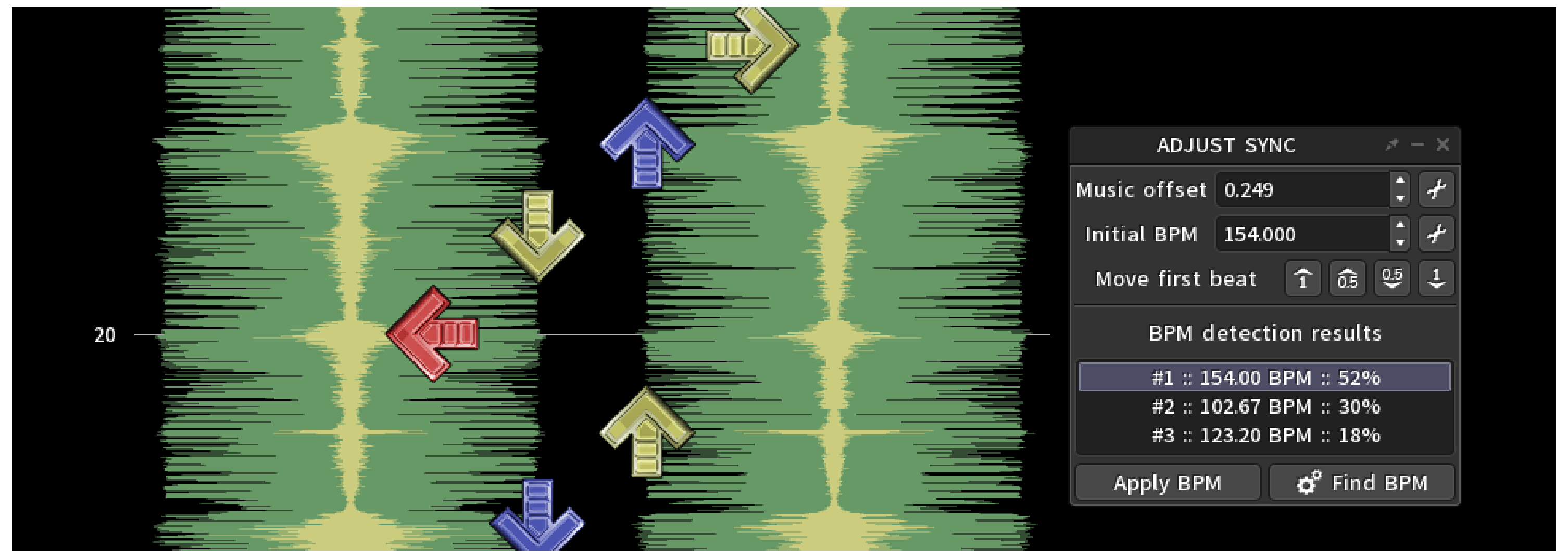
Introduction

Tempo detection is an important technology for the synchronization of anything to music. Knowing the rate at which beats occur per minute makes it significantly more convenient to time events in music-based games and videos, and an automated method to find the beats per minute (BPM) of an audio track is especially important when precision beyond the human ear is necessary and when BPM needs to be calculated for more than a few songs, as manually determining the BPM of a track using a metronome and trial and error is tedious. This project explores if musical tempo can be determined with high accuracy through automated means. My method utilizes the onsets of a song, which are the starts of sounds or musical notes. In this paper, various tempo detection methods as well as various onset detection methods are assessed to determine which are the best fit for precision and accuracy.

Background

A fairly unknown report was written by van de Wetering [1] on tempo detection, intended for the purpose of synchronizing music to rhythm games. Rhythm games require the player to play accurately to the music, and as such the game needs to be able to determine if the player's actions match with the music, making it important that the tempo used in the game is accurate. I found van de Wetering's method in a program he wrote called ArrowVortex (`https://arrowvortex.ddrnl.com/index.html`), which allows one to create rhythm game "charts," meaning synchronized instructions and patterns.

Accuracy should be as close to 100% as possible for as many songs as possible in order to properly synchronize anything to music, or else the rhythm will slowly diverge. Van de Wetering's solution seems to show accurate results under the assumption that the tempo is constant in the music (which is true of most modern, professionally produced music) and that there are relatively sharp changes in the waveform (making slow orchestral works a bad fit, for example).



Music offset 0.249
Initial BPM 154.000

#1 :: 154.00 BPM :: 52%
#2 :: 102.67 BPM :: 30%
#3 :: 123.20 BPM :: 18%

Figure 1: ArrowVortex and its sync feature.

Methods

A selection of 30-second samples of music were taken from Spotify for reference. The artists were selected mostly through Billboard's Top Artists of the 2010s list, though some extra artists were also added into the selection. The nature of the dataset suggests that the majority of the music being tested falls under pop or rap music, but genres such as country and EDM are also incorporated.

Using van de Wetering's tempo detection process as a base, I compare various onset detection methods to see which works the best with the implementation. Nine algorithms will be taken from the `aubio` Python library, and three algorithms from the `madmom` Python library, all of which from the latter are neural-network-based. Spotify, `aubio`, and `madmom` also have their own tempo detection methods which I will evaluate as well.

Results

Method[1]	MSE[2]	Error rate	Insignificant errors	Half-BPM detections
BLSTM	13.813	70/773 (9.06%)	198	2
Complex	17.344	48/773 (6.21%)	124	1
CNN	8.424	56/773 (7.24%)	197	3
Energy	54.041	129/773 (16.7%)	148	10
HFC	44.041	68/773 (8.80%)	120	2
LL (RNN)	14.153	40/773 (5.17%)	152	0
MKL	16.176	55/773 (7.12%)	115	2
KL	27.468	77/773 (9.96%)	140	2
PB	262.734	288/773 (37.3%)	158	33
SD	36.647	61/773 (7.89%)	135	6
SF	33.291	47/773 (6.08%)	123	1
`aubio`	638.627	772/772 (100%)	0	130
`madmom`	151.536	686/772 (88.9%)	3	381
Spotify	116.462	388/772 (50.3%)	376	264

[1] Italicized methods are neural-network-based.

[2] The mode is used as ground truth.

[3] Highlighted cells indicate the best value in a column. Insignificant errors are not highlighted because the best candidates in the column received the most significant errors.

Table 1: Comparison of various onset and tempo detection methods on the Spotify dataset.

Analysis

Notably, none of the other tempo detection methods succeded in reaching comparability to van de Wetering's implementation. The overwhelming difference in error rate shows how important it is to introduce a working tempo detection system and open it for public use so that it is easier to find the correct values without doing the process by hand. Tempo detection for `madmom` and `aubio` was also notably slow; it took approximately an hour to process all the songs (1.79 seconds per 30-second sample on average for `madmom` and `aubio` combined) while it took about 20 minutes to process the onsets and tempo for all 12 different onset methods using van de Wetering's method. In terms of performance, van de Wetering's method comparatively worked quickly.

For precision, it seems that all of the neural-network-based onset detection algorithms are the best when detecting tempo. However, very slight differences in error are common and rounding is recommended to ensure that they work well. In terms of reliability and small error rates, the **Complex Domain, Spectral Flux and LL algorithms** shined.

In terms of performance, machine-learning-based methods are naturally slower and more resource-intensive, though all of them work faster than real-time on most devices. Ultimately, Complex Domain and Spectral Flux seem to stand out as the best non-machine-learning based algorithms, and from preliminary testing it would seem that Complex Domain is more precise in its estimations, making that and the LL algorithm the most well-rounded algorithms out of the bunch, especially since both of them can be used in real-time as well.

Conclusion

From looking at the previous data, it is clear that the current state-of-the-art tempo estimation is not perfect. Ideally the error rate should be as close to 0% as possible, as projects which require synchronization to music need accurate tempo values. The next steps would be to pinpoint the weaknesses of each of the methods and create a large dataset with lots of variety as well as human-confirmed BPM values to reference. Using data from all sorts of musical genres would be ideal. Overall, I am pleased with the progress in this project as it is extremely useful to me and many others to be able to synchronize videos, games, and various media automatically to music without having to go through lengthy trial-and-error processes.

References

[1] B. van de Wetering, *Non-causal beat tracking for rhythm games*, Mar. 2016.