DETECTING WESTERN RHYTHMIC DIVISIONS IN GLOBAL TEMPO ESTIMATION TASKS

Tyler Furrier
Northeastern University
Boston, MA
furrier.t@northeastern.edu

Anthony De Ritis
Northeastern University
Boston, MA
a.deritis@northeastern.edu

ABSTRACT

The process of automated tempo estimation has long been a hot property in the world of music information retrieval (MIR). Year by year, the field of MIR sees increasing data science development, efficiency, and accuracy. While math has grown more involved, the attention to cognitive research concerning this task does not seem to have been upheld and developed to the same degree. In hopes of an elegant algorithm that performs reliably in global tempo estimation, we examined common procedures among popular tempo extraction algorithms. Rather than using machine learning processes to compete in the accuracy arms race, we take an explainable approach inspired by previous research and aim to implement a simple set of procedures while still achieving accuracy. We propose a novel process inspired by the prevalent western theory of rhythm to interpret beat periodicity charts more holistically. This addition worked in eliminating tempo aliases or "red herrings" (incorrect but rhythmically related), while also pairing a maximal tempo decision with the specific metric hierarchy accompanying that maximal response. Our 'meter aware' approach resulted in improvement in global tempo estimation for modern songs when compared with its predecessor.

1. INTRODUCTION

Since the normalization of the computer and its applications, demand for music information retrieval algorithms has been bullishly increasing. One of the most prominently discussed and developed categories is tempo estimation and beat tracking. This may be attributed to the level of unknowns in the research around beat and meter perception, and the accompanying diversity in the array of algorithms in lieu.

The way we listen to music has changed drastically in the last ten years, with most of the listening being through streaming platforms such as Spotify and Apple Music. With it has come the increasing use of listening based recommendations and the necessity for computer processes to distinguish between different songs. However, most of these playlists and radios are either based on the listening activity of other users (behavioral data) or are handmade by an employee of the streaming service. One might hope for the possibility of intelligent algorithms that can work with features of a song that match the perception of a listener. In other words, if a computer can listen to music the way a human does, the way we interact with music can reach new heights. If reasonable accuracy is found within these processes, then they would offer more than just better music recommendations, they would lay the groundwork for more specific sorting of playlists and even the possibility for continuous mixes of songs that would otherwise justify the expensive price of hiring a skilled DJ. Unfortunately, even the most advanced models, for certain specific musical tasks, still diverge significantly from the accuracy and listening mechanisms that we operate naturally as humans.[8] In the interest of this greater goal, we explore the benefits of incorporating well researched theories of perception and music generally into a tempo estimation algorithm.

2. BASE MODEL

2.1 Overview

The basis of our experimentation works through a "vanilla" tempo extractor which evolves from previous algorithmic approaches.[1][3] Another evaluation from 2012 [1] looked at twenty-three competitive tempo extraction methods and was deemed useful. was examined look at twenty-three competitive tempo extraction methods. No single algorithm inferred tempo perfectly, each performed better in different categories. Of all estimation methods, each had the two primary steps of generating an Onset Strength Signal (OSS) and extracted some version of beat periodicity. In general, this step is used to find onsets or beat locations in the sound file. While some methods marked beat locations and others marked strength of onsets at each time point, the common denominator in each method is an observance of the change in magnitude of various frequency bands with time. The second portion present in all twenty-three methods was an estimation of the periodicity of the derived data after parsing the sound file in step one. For most methods, this step was either based on or implementing an autocorrelation of some form.[1]

It is worth mentioning that these surveys are over a decade old, and there are many other high performing tempo models. There are several other approaches that yield better results. However, the simplicity of the algorithm proposed by [3] allows for faster replication. A more important benefit to the algorithm's simplicity is the fact that (barring a small final step that we skipped) it does not involve machine learning approaches like the neural networks seen in the newer algorithms.[5] The goal of this paper is primarily experimental research into the efficacy of various intricacies in MIR processes. This research is by no means as formal or professional as the citations mentioned. There was a necessity to develop and test this implementation hastily. Most AI/ML steps involve a much larger time commitment in their development, as well as a significant loss of visibility/traceability/explainability, both of which are counterproductive to these experimentations.

2.2 Onset Computation / Spectral Flux

We derive an onset strength signal (OSS) by computing a Discrete Fourier Transformation on every time window (turning down edges to reduce leaking) and then calculating the magnitude of an onset in any given time window as the accumulation of increase in magnitude for each frequency relative to the prior window.[3]

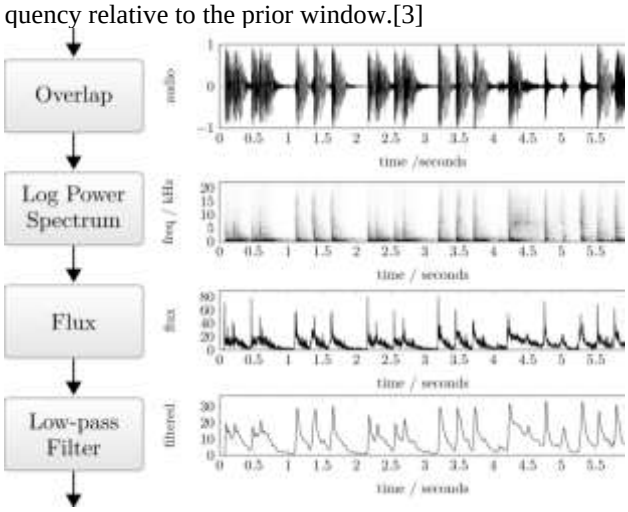

Figure 1. Dataflow diagram for OSS calculation from Percival[3]

Our base approach will look similar to the approach detailed by Percival and Tzanetakis as shown in figure 1. [3] However, certain implementation details were changed or omitted.

2.2.1 Time-Frequency Representation

A Short-Time Fourier Transform (STFT) was used to obtain time-frequency representations throughout tow size for the transformation was 1024 samples or 23.2 ms at our consistent sample rate of 44100 Hz. Notably, window sizes that are powers of two result in faster run time. The hop size, or time difference between each window was 126 samples (2.9 ms). These values were taken from the previously cited work from Percival, but the hop size was decreased from 128 to 126 in order to make the sample rate of the OSS an integer (350 Hz). Before transformation, each window was multiplied by a Hanning window function. This is common practice and reduces the impact of incomplete frequencies at the edge of the window from just outside. Further, magnitudes in each frequency bin are scaled logarithmically to achieve a log-power spectrogram. It is also worth noting that absolute values of each frequency bin's magnitude are used to ignore phase, although more complex notions of "phase-memory" could be worth testing.[9]

*2.2.2 Onset Computation / Spectral Flux**

$$Flux(n) = \sum_{k=1}^{N-1} max(L_P(k,n) - L_P(k,n-1), 0)$$

The above equation describes how we calculate spectral flex given L_P is the log-power spectrogram. Each sample in the flux is the sum of increase in magnitude across each frequency bin. Importantly, decreases in magnitude are ignored. This rule was taken from the Percival paper, which does not go into detail as to why decreases should be ignored.[3] However, it seems there are several possible explanations. One explanation is that a decrease in one frequency should not cancel out increase in another, for example a note stepping down should be considered an onset.

2.2.3 Low pass filter

At this point, the OSS will contain a fair amount of noise from slight changes that are not onsets. This will 'blur' any insights we extract from the OSS. Further, it will bias higher tempos. To remove the noise while not removing potential onsets or rhythmic content, a 14th-order FIR filter was applied with a cutoff of 7 Hz, since it is twice the frequency of 210 BPM which is the presumed maximum tempo for this task.

```
def low_pass_filter(signal, cutoff,
fs, order, plot: bool = False):
    from numpy import convolve
    from scipy.signal import firwin
    nyq = 0.5 * fs
    normal_cutoff = cutoff / nyq
    taps = firwin(
              order + 1,
              normal_cutoff,
              window="hamming")
    y = convolve(
              signal,
              taps,
              mode='same')
```

Figure 2. The python code used for applying the low-pass filter.

2.3 Autocorrelation

To identify tempo candidates, an autocorrelation[1] is applied to the OSS at various time lags. The inspiration for this algorithm performed an autocorrelation on various windows and then used complex methods to accumulate the final tempo across them. For the sake of runtime and speed of development, this was skipped, and the autocorrelation was performed on the entire OSS. By eliminating a complex step, we also remove a factor of variability and complexity in the performance results. With a simpler model, we can more easily and concretely draw conclusions from the effects of our additions in section three.

For sake of comparison, the first iteration takes the largest peak as the tempo prediction, while the second iteration proposes a more complex evaluation of the periodicity chart.

Since each bit in the OSS equates to ~2.9ms, initial implementations of our autocorrelation used linear interpolation to ensure exact lag times rather than rounding to the nearest bit and describing a tempo with up to 1.45ms different of a quarter note[2] than its annotation. For the sake of efficiency, the final version upsampled the OSS and then rounded the lag to the nearest integer number of bits to skip which sped up the process significantly.

3. IMPROVED METHODS/MODEL

3.1 Steps After Autocorrelation

As we noted from previous research and accounted for in our initial approach, octave ambiguity is the cause for most incorrect estimates. Octave ambiguity describes cases in which a tempo estimation is some integer or integer fraction of the annotated "ground truth" tempo where the estimation is tracking a different rhythmic division than the annotation.

Octave ambiguity is the tip of the iceberg, there are other aliases that are not just integers. Consider figure 3, there are many peaks besides the annotated tempo of 126 and the octave below: 63. For the set of songs we considered, all these peaks were rhythmically related to the ground truth tempo and its rhythmic content.

After computing values for each lag, there are various potential approaches. From a quick survey, the most promising seem to be a neural network [Schreiber 2018] or aggregation of tempo octaves. Again, for the sake of simplicity and explainability we are avoiding neural nets.

3.2 Avoiding Pulse Trains

They found success using a pulse train that outlined a typical simple meter, we looked past this step out of concerns for universality. This paper does not include direct proof that the pulse train is rigid in suitability, but it is fair to assume that it would improperly handle musical pieces that are compound in subdivision. There is no hope in solving just that incompatibility, even if differing pulse trains were used for various meters, there is still a necessity of perfect alignment with the rhythms present. A perfect alignment does not exist, there could always be unreliability in a mechanical pulse train given the human ability to hallucinate significant stretches in rhythmic onset timing to satisfy the mechanical perception.[6] If these points require a headache and a half to comprehend why a pulse train does not work, consider the following simpler point. A pulse train may consider songs as out of phase and weaker in tempo due to the beats falling consistently in between dominant beats when our perception is that of a strong tempo. For example, consider "Cotton-Eyed Joe" by The Chieftains[3] at 1:45 when much of the percussive weight falls on the 8th notes.[7]

3.3 Meter Accentuation

More importantly, the work that our base algorithm draws from addresses octave ambiguity but does not address the 2/3 alias and their approach seemed to be rigid to simple rhythmic division.[3] In place of a pulse train[4], we tried using the OSS autocorrelation peaks as indicators of various periodicities that may help confirm a tempo candidate. For example, consider the following figure:

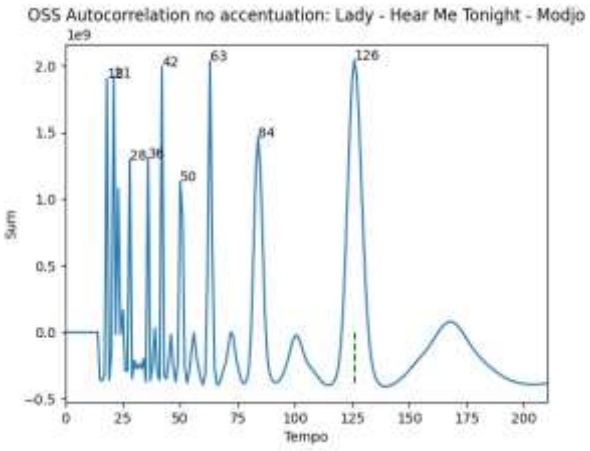

Figure 3: Magnitudes for autocorrelation at various integer tempos (BPM). Top 10 peaks are labeled with the integer tempo. A green dashed line denotes the manually annotated tempo.

Autocorrelating upon the OSS for each tempo will result in peaks pertaining to various periodicities of the song. For example, consider figure 3 showing periodicities for

[1] An autocorrelation is the correlation of signal with a delayed copy of itself, the magnitude in this case indicates the magnitude at which periodicity of that delay exists.

[2] I understand that the pulse is not always a quarter note, but for the sake of brevity and specificity we are calling the primary beat a quarter note

[3] https://open.spotify.com/track/2oWHbanLLXpg4L2yHoGxCt?si=a6101f43d6bf48cc

[4] Without a pulse train, autocorrelations do not need to be windowed since it was presumably needed for correcting the phase of the pulse train if moments of silence are present that reset the phase of the beat.

"Lady – Hear Me Tonight" by Modjo[5]. These subdivision peaks point to the rhythmic scheme of the piece or at least explain peaks as related due to frequent rhythms besides the main beat, allowing for more involved techniques to compare tempo candidates in the context of rhythmic division.

Many incorrect estimations were caused by the peak at 2/3 of the ground truth, corresponding to 3 eighth notes[6]. It is true that this alias or red herring might be accounted for by adding the magnitudes of tempos that signify some multiple of a subdivision of a beat to the magnitude of that tempo. In the case of "Lady – Hear Me Tonight", the magnitude of 129 BPM would be the sum of several lag cross correlations, including 84 BPM.

Testing this approach relies on the background knowledge of a song's subdivision. Since most current evaluation datasets do not contain annotations of meter, large scale benchmarking of an approach like this might be hindered or introduce tediousness for the researcher. This is further evidence towards conclusions that the current state of tempo estimation evaluation is not aptly suited to the complexity of human beat perception as well as the specificity of use case domain.[5]

Before explaining each peak for the song "Lady – Hear Me Tonight", it is important to preface that the autocorrelation sampling goes by each integer, so I conveniently call some numbers close enough. Presumably, if the same figures were to be generated from a tighter sampling of autocorrelations (finer level of granularity), the peaks would be exactly the mentioned decimal and consequentially higher in magnitude.

In the context of 126 bpm and a simple meter, the following tempos are explained as follows. Tempos highlighted in green are top 10 peaks.

126.00: Quarter note (126 * 1)
84.00: 3 eighth notes (126 * 2 / 3)
63.00: 2 Quarter notes (126 * 1 / 2)
50.40: 5 eighth notes (126 * 2 / 5)
42.00: 3 Quarter notes (126 * 1 / 3)
36.00: 7 eighth notes (126 * 2 / 7)
31.50: 4 Quarter notes
28.00: 9 eighth notes (126 * 2 / 9)
25.20: 5 Quarter notes (126 / 5)
22.90: 11 eighth notes (126 * 2 / 11)
21.00: 6 Quarter notes (126 / 6)
18.00: 7 Quarter notes (126 / 7)

Values such as 31.5 are presumably missing because the nearest autocorrelation values are 31 and 32 which are significantly different

With this observation, we propose a new approach where:

$$Original(t) = AC(60/t) \mid 15 < t < 840$$

We gather autocorrelations for a larger range to account for 16th notes at 210 bpm[7] and a 4-quarter-note lag at 60 bpm. The low-pass filter cutoff was increased accordingly.

$$S(t,d) = \{s,\quad s \cdot d,\quad s \cdot d \cdot 2\}$$

where t is the tempo, s is the time lag for one beat at that tempo, and d the division (either 2 or 3)

$$TempoAgg(t,d) = \sum_{t_l \in S(t,d)} \sum_{n=1}^{t_l \div 15} Original(t_l/n)$$

$$T = \max_{50 \le t \le 210} \left(\max_{2 \le d \le 3} TempoAgg(t,d) \right)$$

In less mathematical terms, we are counting the magnitude of some tempo t as the sum of n = 1, 2, 3, … quarter notes until that tempo is below the range of autocorrelation data. The same is conducted with the tempo of each subdivision. So, for simple meter, the magnitude becomes the sum of all autocorrelation lags in units of 1 beat, ½ beat, and ¼ beat. It is important to note that this *initial attempt* is not perfect, any required lag values that were not an integer were determined through linear interpolation.

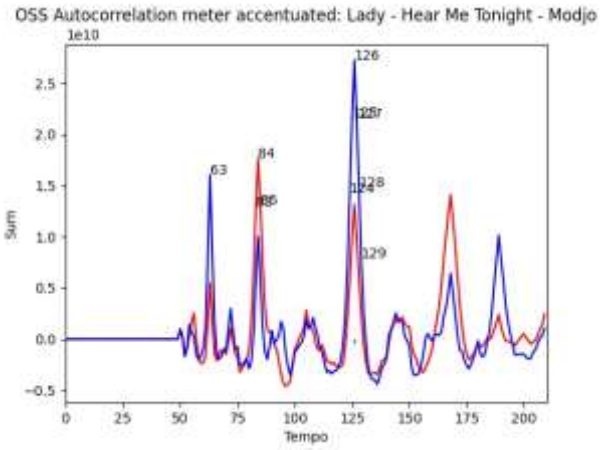

Figure 4: Division accentuated magnitudes at various integer tempos (BPM). Red is compound division and blue is simple.

3.4 Results at a Glance

The max magnitude for 84 bpm is in red, indicating a higher match for compound meter, which makes sense because there are 3 eighth notes in between each pulse. Importantly, the chart seems to be closer to the way we rhythmically interpret it, with the octave error (63 bpm) being present but not as strong as 126, and peaks in between being explained by misinterpretation of meter. At a first glance, it might look like a much prettier graph revolving around the tempo, compared to the previous sparse graph, potentially indicating an effective abstract representation.

However, the fact that a compound division at 84 doesn't line up with each measure could be accounted for. In other words, the red peak at 84 should not be that high if measure

[5] https://open.spotify.com/track/49X0LAl6faAusYq02PRAY6?si=1014bf8f705243e0

[6] For example, take a listen to Fake id (125bpm) at 83.333 bpm: https://drive.google.com/file/d/1sKtEq2xOWqLGGssggiqgzlPwTqjjanbu/view?usp=sharing

[7] This is a mistake, the max subdivision for 16th notes at 210 should account for compound meter where it is 210 * 3 * 2 = 1260, not 210 * 2 * 2 = 840.

long steps are being accounted for, since that tempo rarely lines up with the measure starts. This could potentially be an indicator of the OSS being inadequate for comparing measure similarities. Also, it could be a symptom of general biases and susceptibility to "over-averaging" which is discussed in the next steps section.

4. RESULTS

4.1 Evaluation

We ran both versions against a **dataset[8] of 50 fixed tempo songs** which I manually annotated and checked for tempo. Most of these songs were EDM/dance/"party music", and all of them were popular western songs. Most of the songs were simple meter, but there was a good sample of compound metered songs.

Similar to the work from Percival, we considered a type 1 and type 2 accuracy. Type 1 describes when the estimated tempo is close enough to the ground truth tempo. Type 2, describes whether any *octave* of the estimation is close enough to the ground truth tempo, Type 2 multiplies the estimation by 1/3, ½, 1, 2, and 3. However, while the predecessor to this paper used a 4% margin of accuracy, we lowered it to 2% based on updated research into the noticeable difference, insight to the problem statement surrounding this task, and my own personal expectations for performance on this type of data.[3, 5]

Base:
TYPE 1 ACCURACY: 0.78 | 39 / 50
TYPE 2 ACCURACY: 0.94 | 47 / 50
Meter Accentuated:
TYPE 1 ACCURACY: 0.92 | 46 / 50
TYPE 2 ACCURACY: 0.98 | 49 / 50

The song dataset consisted mostly of popular dance music with rhythmic content that aligns with typical western theory, and often rhythmic consistencies within the genre. As of right now, there is no saying whether comparable results can be found on **less popular music or non-western music**. Thus, much more work needs to be done before these additions can be deemed universally beneficial.

5. NEXT STEPS/IMPACT/DISCUSSION

5.1 Improving the Meter Accentuation

5.1.1 Increasing autocorrelation granularity

This is just an initial approach, and there are urgent cautionary modifications that might allow the algorithm to reach its full potential. For starters, the autocorrelation needs much more than integer precision to accurately capture magnitudes of related tempos. Even linearly interpolating the nearby integers introduces a randomness, with more entropy at lower tempos (which there are more off in the current approach).

[8] https://open.spotify.com/playlist/5UuPb1Dyb59lkBQr1u6PKK?si=99ae7c7dc3664f95

5.1.2 Removing biases towards faster tempos

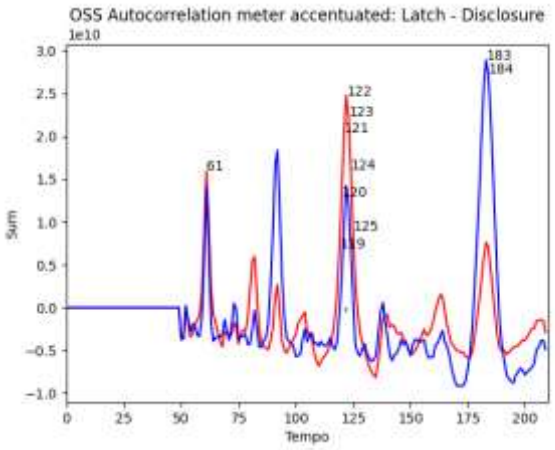

Figure 5: Division accentuated magnitudes at various integer tempos (BPM) for "Latch" by Disclosure. Red is compound division and blue is simple.

Consider figure 5, where the song is compound, and the ground truth tempo is 122 bpm. This was the only song that the algorithm did not estimate correctly for type 2 accuracy. The false tempo returned was 183 or 3/2 * 122 and a perception of two 8th notes in simple meter. There are a couple speculations as to what causes this error, the first of which is a bias towards faster tempos intrinsic in the algorithm design. For multiples of each beat division, a faster tempo will be able to fit more beat increments before it is too slow and out of the autocorrelation range. For the "Latch" example, magnitude aggregation for 183 will reach 12 quarter notes before dropping under 15 bpm. On the other hand, 122 will only fit 8 quarter notes and then be out of range. Importantly, those 3 magnitude additions come along with 6 more from the first subdivision and 12 from the second. This potential overemphasis on smaller subdivisions is elaborated upon in section 5.1.5. There may also be room for investigation into whether there is a slight bias towards compound meters as well due to the number of magnitude additions from subdivision being 3 to just 2 in simple meter.

A possible solution to these biases would be to take average magnitudes across each lag summation.

5.1.3 Measure blurring.

While averaging across aggregated peaks may alleviate the explicit bias, there could be a problem we will call "measure blurring". It is entirely possible that many weak magnitudes across various quarter-note-length lags could give a tempo candidate an edge over the ground truth when the ground truth has a large magnitude at the lag of the length of quarter notes in each actual measure, but small magnitudes everywhere else. In this sense, the information pertaining to measure length is *blurred* and average magnitude for each measure length becomes the deciding factor.

5.1.4 Separating Candidates by Measure Length

To avoid decreases in reliability due to the theorized measure blurring, tempo candidates could be separated by measure length rather than just separating by subdivision. It would be important to remember that this change might require the dataset to be not only fixed tempo but also fixed measure length, or alternatively use windowed autocorrelations to dynamically switch the max candidate to the proper meter. In fact, this concern could realistically apply to a lesser extent already with the separation of subdivision context.

Investigating ideal aggregation of measure length magnitudes may be worthwhile as well. For example, you want to take the max measure, but what if the measure if "out of phase" from the next? Presumably, you could aggregate multiples of measure and ½ of the multiple should be less significant or potentially "out of phase". If a so-called measure had one variation and followed by another with a different variation, and then back to the first… it would make sense to call those first two a single measure. The preconceived notion of complexity in identify measure length becomes more confident when suspicions regarding OSS clarity are raised in section 5.2, since the likelihood of discriminating measure-based repetition against reference level repetition by using the OSS, which was an already scrutinized method due to not having frequency data, becomes even more extremely unlikely.

5.1.5 Re-weighing subdivisions

To address a larger emphasis on shorter rhythmic divisions, it may be worth giving less weight to 16th note lags to account for there being more. In general, it makes sense to test what significance the magnitude of something like a 16th note lag plays relative to a quarter-note lag, experimenting with various weights.

This paper only introduces the ideated approach in its infancy. With this concern in mind and the others mentioned, there are many configurations to experiment and benchmark with in hopes of feeling out necessity levels for each proposed update. Further, testing for interactions when including multiple at a time might help reveal how each change is functioning, any exclusivity, or complimenting traits. Hence, current time constraints do not allow for further development without showing a limited perspective which might be biased towards whichever changes did make it in time.

5.2 Improving The OSS

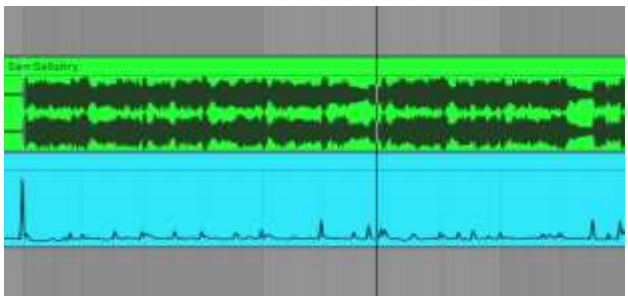

Figure 5: OSS for "Assumptions – Sam Gellaitry" at ~2:30

There are many questions to be raised surrounding the adequacy of the OSS. To be fair, developing accurate representations of onsets or even frequencies is a can of worms. However, this sections details observations that were made and are worth sharing for the sake of improvement, if not as evidence of a necessity for renovation in the OSS department.

To illustrate concerns, figure 5 shows the OSS for "Assumptions" by Sam Gellaitry which can be heard and seen in video form at the following link: https://drive.google.com/file/d/1AScZtlduFB9ZhMk1KVGx6ydwmb0gAsdM/view?usp=sharing

Immediately, inconsistencies can be noticed between extremely large onset markings at certain points that do not seem to match with the perceived onset strength. These onsets are disproportionately high *only* when multiple frequency bands play together.

One explanation is that optimized libraries use common aggregation functions like max or mean, which cause windows where magnitude increases in many frequency bins to have extremely high onset markings and others to be relatively skewed downward by zeros. An adequate model accounting for perceived magnitude when multiple onsets occur across different frequency ranges is much more complex than these simple magnitude sums.

One optimistic outlook is that rapid increase in concurrent onset windows could be due to a scaling issue, where some unit of log-power is being added and creating a multiplicative effect rather than additive effect.

Even if the onset magnitude is properly scaled, there could still be artificial onset strength when the method is combining frequencies that physically are not close enough to combine or interact.

Once again, this is a can of worms, lots of research needs to be read before claims in this paper can be made about adequately combining onsets across frequencies.

Yet, it is clear that the can should be opened or other representations should be investigated before measure long lags are taken on songs that do not repeat identically like the production heavy music tested in this paper.

Finally, if other research has not raised the concern, it is worth questioning whether slight misalignment in timing might discard the perceived unity of instruments and percussion playing in the same beat. Experimentation could involve looking for improvement in an approach that will re-window the OSS to align with beats and aggregate errors in timing to the perceived rhythmic location. If this type of method does not help the OSS, it at least would help in pooling various rhythmic excerpts into fewer categories and aid in efficient data science campaigns.

5.3 Utilizing Meter Insight for for Real Uses

5.3.1 Dualistic Relationship with Beat Tracking

The idea that global tempo estimation can aid in beat tracking tasks is hinted at but has yet to be implemented as an all-in-one computational process.[5]

Even in a scope limited to western semantics of rhythm, human processes of beat tracking rely on classification of rhythms that do not fall perfectly 'on beat' due to performance timing. Research shows that these rhythmic classifications are subjective and probabilistic. Human annotations cannot be accounted for by a simple model like rounding to the nearest rhythmic point with a certain precision. However, the same research found that the probabilities of each rhythmic interpretation vary greatly when preceded by simple or compound meter.[6]

If an accurate estimation of tempo can be reliably output for a given set of music, that prediction can assist the beat tracking processes for the same set of music. In beat tracking adds a metric of confidence to the candidate tempo. This is an untested utilization, meaning the possibility of improved results cannot be written off.

This would allow things like automated quantization. Even at its surface level, automated quantization (beat-matching, to use DJ jargon) becomes a realistic vision. Reliable beat matching might enable seamless transitions in combination with higher-entropy algorithms like neural networks that pick transitions with high harmonic and rhythmic compatibility.

5.3.2 Using Hierarchical Meter Detection

Even if human level beat tracking is realistic for a desired set of songs, it is not sufficient for identifying measure locations or significance and roles of certain beats. In other words, two songs can overlay their primary beats perfectly and still be out of sync. One example is a 4/4 song against another where the first quarter note is aligned with the second. A more prevalent example might be a ¾ song over a 6/8 meter[9].

Our proposed addition takes subdivision into consideration, but fully discriminating predictions by western meter was not attempted and a more diverse dataset would be necessary to test meter estimation. It is important to note that meter can be subjective, most likely more so than tempo. It is important to clarify that meter estimation algorithms are just that, estimative models rather than reliable algorithms.

While several features might be capable or even necessary for this type of inference, meter provides context that is best explained by unique human perception of both phrasing and emphasis along with cognitive processes of rhythm interpretation that are not yet aptly understood. With an accurate prediction of the subdivision alone, some of these complexities can begin to unravel[10].

Once again, one component of the algorithm can compound synergistically with others to yield more confident and accurate results. To start, many tempo aliases or false candidates are described by rhythmic patterns and are accounted for once the tempos are evaluated in the context of meter.

For applications like recommendation systems or beat matching, it is beneficial that a process is describing the rhythmic character of a phrase, since that description is more abstract and is anticipated to be more effective than comparing two raw sets autocorrelation data and undoubtedly more efficient than utilizing the raw OSS or raw signal. This makes sense, you don't smack a compound meter song on top of a simple meter song.

Again, interpretations of rhythm are not bound to simple rounding methods. A rhythm can be interpreted in many different ways and played in many different ways. This bending and stretching of time in the human mechanisms that perform and perceive tempo is greatly affected by a bias to any pre-exposed meter.[6] Depending on our beat tracking approach, a meter estimation may also help detect beats with confidence. This of course rolls back to helping confirm or retest previous tempo and meter estimations, allowing a cycle again that may continuously develop information extraction until convergence. Although our task does not involve explicit notation of rhythms (does involve beat tracking), grouping rhythms as their notation may be an efficient method of pooling and feature extraction that best matches human mechanisms.

Acknowledgments

It's important that I give a huge thanks to Tony De Ritis for his mentoring and guidance which helped me to better understand the complexities of this task as well as his help in managing the time constraints. Discussions around this task and MIR in general can be a deep rabbit hole, when I put a bit too much on my plate, his guidance allowed me to keep my ambitions realistic for the time frame and organize many different thoughts and ideas.

I would also like to acknowledge my previous professors who have helped me acquire the knowledge and skills necessary to come up with many of the thoughts in this paper. Whether their effective teaching style gave the foundational knowledge required to understand this discussion, or they taught completely different subjects but unknowingly encouraged me to approach music, academics, and research topics with confidence, curiosity, and excitement… they can take a large amount of credit for the way I have grown as a student and more generally a thinker. My musical learning experiences, especially in but also beyond Northeastern, have allowed me to enjoy new

[9] As stated before, more complex approaches may write off these cases anyways based on their harmonic mismatch. Regardless of the potential efficiency of the mysterious proposals, detecting meter is sure to save time.

[10] It's like playing sudoku, when you know a square can be one of two numbers and the implications of one option either confirm or deny that option. Or more simply, once you know one square, many others are clear and so on. Maybe solitaire is a better example though.

sides of music and I am able to joyfully jog my brain around these topics for hours. If you are one of these professors who have made an impact, you know… thank you from the bottom of my heart! By making the lives of students like me more enjoyable, effective, and purposeful, you are making the world a better place to exponential effects! <3

6. REFERENCES

[1] J. Zapata & E . Gómez, "*Comparative evaluation and combination of audio tempo estimation approaches"*, in *Proc. AES Conf. Semantic Audio*,Jul.2011

[2] L. van Noorden and D. Moelants, "Resonance in the perception of musical pulse," *Journal of new music research*, vol. 28, no. 1, pp. 43–66, 1999, doi: 10.1076/jnmr.28.1.43.3122.

[3] G. Percival and G. Tzanetakis, "Streamlined tempo estimation based on autocorrelation and cross-correlation with pulses," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 22, no. 12, pp. 1765–1776, 2014, doi: 10.1109/TASLP.2014.2348916.

[4] M. Talbot-Smith, *Audio Engineer's reference book*. 1999.

[5] H. Schreiber, J. Urbanoand M. Müller, "Music Tempo Estimation: Are We Done Yet?", *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, p. 111–125, 2020.DOI: https://doi.org/10.5334/tismir.43

[6] Desain, P., & Honing, H. (2003). The Formation of Rhythmic Categories and Metric Priming. Perception, 32(3), 341-365. https://doi.org/10.1068/p3370

[7] T. Chieftains, R. Skaggs, Cotton-Eyed Joe. RCA Victor, 1992.

[8] Feather, J., Leclerc, G., Mądry, A. *et al.* Model metamers reveal divergent invariances between biological and artificial neural networks. *Nat Neurosci* **26**, 2017–2034 (2023). https://doi.org/10.1038/s41593-023-01442-0

[9] K. Aczél and S. Iváncsy, "Sound separation of polyphonic music using instrument prints," 2007 15th European Signal Processing Conference, Poznan, Poland, 2007, pp. 931-935.