# EXPLOITING GLOBAL FEATURES FOR TEMPO OCTAVE CORRECTION

*Hendrik Schreiber*

tagtraum industries incorporated
hs@tagtraum.com

*Meinard Müller*[*]

International Audio Laboratories Erlangen
meinard.mueller@audiolabs-erlangen.de

### ABSTRACT

Tempo estimation is a fundamental problem in music information retrieval. Most approaches attempt to solve two problems: first finding a dominant pulse and second correcting the metrical level of this pulse. The latter has also been dubbed fixing the octave error. We propose an algorithm for tempo estimation that addresses both problems mostly independently. While using a standard pulse detection technique, for octave error correction, we exploit a simple relationship between a single global feature, average spectral novelty, and listener perception of musical tempo. The proposed method is extremely simple. Nevertheless, it outperforms most existing tempo estimation methods and is on par with the best-performing ones. It thus exemplifies that a global feature-based approach can significantly improve tempo estimation.

***Index Terms*—** music information retrieval, tempo induction, rhythm analysis, audio signal processing

## 1. INTRODUCTION

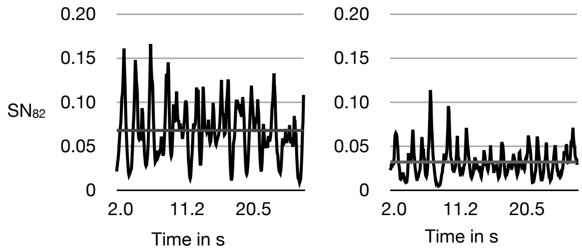
Describing its speed, *tempo* is one of the relevant descriptors for a piece of music. It can be defined as the number of times a listener "taps" a beat with his or her foot per time interval. As unit of measurement for tempo serves *beats per minute* (BPM). Automatic *tempo estimation/induction* is commonly used to estimate the general tempo of a musical piece. Unlike *beat tracking*, tempo estimation does not attempt to determine the exact location of individual beats. Tempo can be used for a wide variety of applications including music retrieval, score alignment, playlist generation, and DJ techniques like beatmixing. Because of its usefulness, the automatic extraction of tempo is a traditional task in Music Information Retrieval (MIR) and has received a lot of attention over the years [1, 2, 3]. Besides fluctuating tempi and other issues [4], one big problem in tempo estimation is the so-called *octave error*, i.e., results are fractions or multiples of the perceived tempo. Current algorithms are very accurate when ignoring the octave error, but accuracy decreases significantly when requiring the correct octave [5]. Therefore choosing the right octave or metrical level has been the subject of recent research and is the main focus of this paper. In [6] Hockman and Fujinaga provide a short overview of different approaches. Among them: limiting valid outputs to a single metrical level, picking a tempo closest to the mean of the expected distribution, using a hidden Markov model to model the temporal evolution of metrical sequences, and associating timbral characteristics to discrete BPM values. Hockman and Fujinaga themselves suggest classifying audio signals into the perceived tempo classes `slow` and `fast` using machine learning and a bag of global features. In their experiments they use Last.fm tags and YouTube playlists as ground truth, and achieve a remarkable accuracy for the popular genres Country, Jazz, Rap, R&B, and Rock. Ballroom genres and classical music were unfortunately not part of this study. Still, they show that algorithms can reliably classify an audio track as either `slow` or `fast`. The same is true for listeners [7]. Contrary to this, determining *one* exact tempo in BPM either via listeners or algorithms remains difficult—may even be impossible. The concept of *tempo ambiguity* [8] states that for some tracks listeners claim two different tempi, usually multiples of each other. In this context, Levy [7] points out that besides musical correctness, usefulness of an estimate should be taken into account: Even though a track has a tempo of 140 BPM as determined by expert listeners, it may be perceived as slow by many casual listeners. For them an estimate of 70 BPM may indeed be more useful. Consequently, one might argue that global, perceptual features of music—like slow vs. fast—should receive more attention when determining the "correct" tempo.

While Hockman and Fujinaga do not incorporate their classifier into a tempo estimation system, Peeters and Flocon-Cholet [9] successfully built such a system using a few selected features and GMM-Regression. Using the same features, it estimates both a perceptual tempo and a perceptual tempo class in one step. Contrary to this approach, Gkiokas et al. [10] use tempo classification to pick one of multiple possible solutions. Their classifier uses a support vector machine (SVM) trained with the same periodicity vectors that are also used to find tempo candidates. While Gkiokas et al. use a rather complex periodicity detection, employing constant Q-transforms and harmonic/percussive separation, Tzanetakis and Percival [5] chose to simplify state-of-the-art tempo estimation as much as possible, relying on an onset detection function, its autocorrelation, and its cross correlation with an idealized pulse train. Octave correction is achieved with a very simple heuristic.

But none of the mentioned systems uses simple global features for octave correction. Not quite an exception to this observation, but representing a step in a similar direction, Schuller et al. [11] exploit the fact that ballroom tempi are very genre-specific, by first performing a genre classification and then using its result to determine tempo and tempo octave. Because most of the ballroom genres are very much defined by a narrow BPM range, it is not clear, whether this approach could also work for genres with broader BPM ranges like Pop.

In this paper we combine a standard method for pulse detection with a simple method for tempo octave estimation based on a single global feature. Although we emphasize simplicity, we show that this combination can lead to convincing results. In Section 2, we start with deriving a global feature for rough tempo estimation, then, in Section 3, we describe the algorithm and how it estimates a dominant pulse, a tempo octave, and ultimately a BPM value. Section 4 evaluates the algorithm comparing it with other methods using a large

[*]The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and the Fraunhofer-Institut für Integrierte Schaltungen IIS.




(a) “Come Away With Me” (b) “The Rock Show”

**Fig. 1**. Spectral novelty $\mathrm{SN}_{82}$ of excerpts of Norah Jones’ slow “Come Away With Me”, and of Blink-182’s fast “The Rock Show”. The mean is shown as horizontal line.

dataset. Finally, in Section 5, we present our conclusions.

The code was implemented using the Java-based open source audio feature extraction framework *jipes*.[1] In the interest of reproducibility, we are making a binary version of the algorithm available at `http://bit.ly/H3ZonA`.

## 2. FEATURE SELECTION

Inspired by [6] we collected a number of global song features via the consumer application *beaTunes*.[2] For each song we retrieved Last.fm’s most popular tags and selected those songs associated with the tag `slow` or `fast`, but not both. For the genres Rock, Pop, Jazz, Alternative, Industrial, Heavy Metal, Soul, and Dance the dataset contained 8517 songs, 1296 (15.2%) of which labeled as `fast`. Because of the obvious imbalance between slow and fast songs also observed in [7], we grouped the data by genre, each group with an equal number of songs labeled as `slow` or `fast`. We did so by randomly removing songs of the overrepresented tempo class. Using the common global features (e.g. described in [12]) mean RMS, standard deviation of RMS, mean spectral centroid, mean relative spectral entropy, peak spectral fluctuation, mean spectral novelty, and mean spectral spread we classified the songs into `slow` and `fast` using one feature at a time. From these features, the *mean spectral novelty* (SNM) turned out to be the most successful one. Obviously, the selection of SNM is neither the result of an exhaustive search nor can classification based on one feature at a time be expected to be the best choice. Nevertheless, even a single, imperfect feature can suffice to show the merits of using global features for octave correction, which is the subject of this investigation.

$\mathrm{SNM}_L$ is calculated by first converting the signal to mono with a sample rate of 11025 Hz. Then we compute the spectra $X(t)$ with $t \in [0:T] := \{0, 1, 2, \ldots, T\}$ of 93 ms windows with $1/2$ overlap, by applying a Hamming window and then performing an STFT. From $X$ we build a self-similarity matrix $S$, using the cosine of the angle between two power spectral vectors $Y(t) = |X(t)|^2$ as similarity score. The novelty score $\mathrm{SN}_L$ is calculated with a square Gaussian checkerboard kernel $C_L$ with length $L = 64$, see [13]. Considering the given sample rate and window overlap, this is equivalent to a 2.97 s kernel. We choose to normalize the score $\mathrm{SN}_L$ by dividing by the sum of the absolute values of all kernel elements (Eq. 1). To obtain the mean $\mathrm{SNM}_L$ we average $\mathrm{SN}_L(t)$ for $t \in [L/2 : T - L/2]$.

[1] `http://www.tagtraum.com/jipes`

[2] `http://www.beatunes.com/`

| Genre | Songs | Correct | $\mathrm{SNM}_{64}$ Threshold |
|---|---|---|---|
| Rock | 1706 | 87.86% | 0.034 |
| Pop | 262 | 88.17% | 0.036 |
| Jazz | 330 | 86.06% | 0.040 |
| Alternative | 72 | 81.94% | 0.034 |
| Industrial | 68 | 80.88% | 0.023 |
| Heavy Metal | 54 | 83.33% | 0.032 |
| Soul | 36 | 77.78% | 0.032 |
| Dance | 36 | 83.33% | 0.027 |
| All | 2564 | 85.02% | 0.036 |

**Table 1**: Classification results for different genres using $\mathrm{SNM}_{64}$ and a threshold that was calculated using a decision tree. The labels `slow` and `fast` obtained from Last.fm served as ground truth.

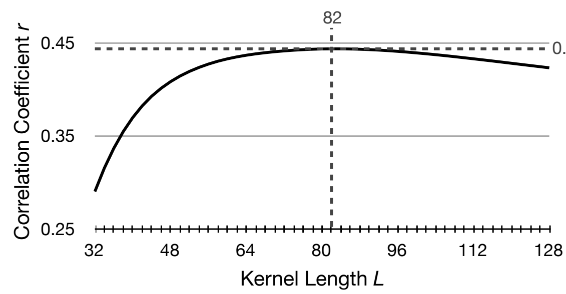



**Fig. 2**. Strength of linear relationship between $\mathrm{SNM}_L$ and ground truth BPM of songs in GTZAN measured with correlation coefficient $r$ depending on kernel length $L$. A maximum of $r = 0.44$ is reached at $L = 82$.

$$\mathrm{SN}_L(t) = \frac{\sum_{m=-L/2}^{L/2-1} \sum_{n=-L/2}^{L/2-1} C_L(m,n) \cdot S(t+m, t+n)}{\sum_{m=-L/2}^{L/2-1} \sum_{n=-L/2}^{L/2-1} |C_L(m,n)|} \qquad (1)$$

To illustrate $\mathrm{SNM}_L$, Fig. 1 shows spectral novelty values computed with a 3.81 s long kernel for a slow and a fast song. It seems surprising that $\mathrm{SNM}_L$ tends to be larger for slower songs than for faster songs. We conjecture that this is the case, because faster songs tend to have more spectral fluctuations than slower songs. With regard to the chosen parameter settings, these fluctuations appear more like noise and less significant compared to the fewer but relatively clear novelty peaks occurring in slower songs. This may explain why $\mathrm{SNM}_L$ is larger in the latter case.

With an overall correct slow/fast classification rate of 85% (Table 1), $\mathrm{SNM}_{64}$ can obviously help estimating the perceived tempo of music. But since our dataset did only contain few values for genres other than Rock, Pop, and Jazz, the validity of this statement is clearly limited to those genres. Also, a perceived slow tempo does not guarantee a certain BPM range. For example, a Viennese Waltz may be perceived as slow, but its tempo is typically 174 to 180 BPM. Furthermore, the kernel length $L = 64$ was chosen before the relationship to the perceived tempo class was discovered.

Therefore we investigated the relationship between SNM and the ground truth of the tempo annotated GTZAN genres dataset [14]. GTZAN consists of 1000 songs, 100 from each genre. As a simple measure of relationship we computed Pearson’s correlation coefficient $r$ between ground truth BPM and $\mathrm{SNM}_L$ for the kernel lengths

| Genre | $r$ | MAE | RMSE |
|---|---|---|---|
| Blues | 0.22 | 22.56 | 27.50 |
| Classical | *0.16* | 22.13 | 27.92 |
| Country | 0.42 | 17.18 | 21.46 |
| Disco | 0.30 | 12.33 | 17.35 |
| Hiphop | 0.34 | 7.51 | 11.00 |
| Jazz | 0.60 | 15.35 | 19.44 |
| Metal | 0.29 | 20.95 | 24.44 |
| Pop | **0.61** | 12.37 | 15.32 |
| Reggae | 0.22 | 13.41 | 17.48 |
| Rock | 0.23 | 20.54 | 24.25 |
| All | 0.44 | 17.50 | 21.86 |

**Table 2**: Genre-specific correlation $r$ between GTZAN ground truth BPM and $\mathrm{SNM}_{82}$ along with mean absolute errors (MAE) and root mean squared errors (RMSE) in BPM for genre-specific linear regressions.

32 to 128 and found the maximum of $r = 0.44$ at $L = 82$, covering 3.81 s (Fig. 2). The low correlation coefficient indicates that this is not a strong linear relationship—at least not for the whole collection. In fact, the results in Table 2 suggest, that the relationship between SNM and BPM is genre-dependent. With $r = 0.61$ and $r = 0.60$ it is very promising for Pop and Jazz, and less so for Blues, Classical, or Reggae with $r = 0.22$, $r = 0.16$, and $r = 0.22$, respectively. But considering that we just want to estimate the tempo octave rather than the precise BPM, mean absolute errors (MAE) of less than 23 BPM and root mean squared errors (RMSE) of less than 28 BPM for genre-specific linear regressions (Table 2), make a linear model suitable enough for our purposes.

## 3. ALGORITHM

We are dividing the problem of tempo estimation into three separate tasks: 1) computing a dominant pulse while largely ignoring the tempo octave, 2) determining a rough estimate of the perceived tempo and thus the tempo octave, and 3) combining the two results in a meaningful way.

### 3.1. Estimating the Dominant Pulse

To estimate the dominant pulse, we follow the general idea of standard approaches measuring changes in the power spectrum $Y$, see [15, 16]. The power for each bin $k$ at time $t$ is given by $Y(t, k)$, its positive logarithmic power $Y_{\ln}(t, k) := \ln(1000 \cdot Y(t, k) + 1)$, and its frequency by $F(k)$. We define the onset strength function (or novelty curve) $O(t)$ as the sum of the bandwise differences between the logarithmic powers $Y_{\ln}(t, k)$ and $Y_{\ln}(t-1, k)$ for those $k$ where the frequency $F(k) \in [30, 720]$ Hz and $Y(t, k)$ is greater than $\alpha Y(t-1, k)$ [17]:

$$\begin{aligned} I(t,k) &= \begin{cases} 1, & \text{if } Y(t,k) > \alpha Y(t-1,k) \\ & \text{and } F(k) \in [30, 720], \\ 0, & \text{otherwise,} \end{cases} \\ O(t) &= \sum_k \big(Y_{\ln}(t,k) - Y_{\ln}(t-1,k)\big) \cdot I(t,k). \end{aligned} \tag{2}$$

The factor $\alpha = 1.76$ was introduced to disregard small increases in loudness and thus to reduce noise in the onset strength signal. Just like the frequency range, its value was found experimentally.

$O(t)$ is transformed using a DFT with length 8192. At the given sample rate, this length ensures a resolution of 0.156 BPM. The peaks of the resulting beat spectrum $B$ represent the strength of BPM values in the signal [18]. But they do not take into account harmonics, i.e., the fact that a 30 BPM peak usually implies a 60 BPM peak [19, 20]. Therefore we derive an enhanced beat spectrum $B_E$, which boosts frequencies that are supported by certain harmonics:

$$B_E(k) = \sum_{i=0}^{2} |B(k \cdot 2^i)| \tag{3}$$

Similar to computing a spectral sum [21] or an enhanced beat histogram [5], $B_E$ incorporates harmonics by simply adding to each bin the magnitudes of the bins corresponding to two times and to four times of its own frequency.[3] Because most popular Western music is in $4/4$-time, and most octave errors are by factor of two [2], we purposefully leave out the third harmonic. This allows $B_E$ to better match a wrong, but strong tempo octave. To calculate the estimated dominant pulse $T$, we determine the highest value of $B_E$ and finally convert its associated frequency to BPM:

$$T = F(\operatorname*{argmax}_k B_E(k)) \cdot 60 \tag{4}$$

### 3.2. Estimating the Tempo Octave

Since we have found a somewhat linear relationship between SNM and BPM, all we have to do to estimate the rough tempo TO, is to find the kernel length $L$ that leads to $\mathrm{SNM}_L$ values that correlate well with a training ground truth and then perform a linear regression. Because the time complexity of computing $\mathrm{SNM}_L$ is quadratic, we prefer smaller $L$. We found that the value determined for GTZAN, $L = 82$, represents a good tradeoff between correlation and runtime behavior. To compute the linear regression with WEKA [22], we use the combined five datasets [7, 9, 3, 23, 14] also used in [5], but a ground truth improved by Percival. The resulting regression for the rough perceived tempo estimate TO is given by:

$$\mathrm{TO} = -851.144 \cdot \mathrm{SNM}_{82} + 137.623 \tag{5}$$

### 3.3. Combining Tempo and Tempo Octave

As mentioned above, most octave errors are by factor of two [2]. Therefore, to compute the final tempo $T_{\text{final}}$, we divide/multiply $T$ with/by two until it is closest to TO. In other words, $T_{\text{final}} = 2^i \cdot T$ with $i \in \mathbb{Z}$ such that $0.75 \cdot \mathrm{TO} < 2^i \cdot T < 1.5 \cdot \mathrm{TO}$.

## 4. EVALUATION

The proposed method `schr1` was compared with the best performing algorithms [24, 25, 26, 27] [4,5] discussed in [5] and a baseline method `schr0` using the same five datasets, with the aforementioned improved ground truth.[6] The baseline `schr0` consists of just the pulse estimation part described above, but without the $\mathrm{SNM}_{82}$-based octave correction. As measures of accuracy we employed *Accuracy1*, the percentage of estimates that are within 4% of the ground truth tempo, and *Accuracy2*, the percentage of estimates that are within 4% of a multiple of $1/3$, $1/2$, 2, 3 times the ground truth

[3] The actual implementation differs slightly to take the discrete nature of the DFT into account.

[4] zplane [aufTAKT] V3, `http://www.beat-tracking.com/`

[5] Dev build v3.2, `http://developer.echonest.com/`

[6] Therefore the results are not identical to [5].

| Dataset | Songs | schr1 | schr0 | marsyas | gkiokas | zplane | echonest | ibt | qm_vamp |
|---|---|---|---|---|---|---|---|---|---|
| ACM MIRUM | 1410 | **76.1** | 70.3- | 71.6- | 72.6- | 70.2- | 73.8 | 63.0- | 63.9- |
| ISMIR04 Songs | 465 | **73.1** | 61.9- | 58.5- | 56.8- | 56.1- | 57.0- | 46.7- | 42.8- |
| Ballroom | 698 | 66.3 | 65.5 | 63.3 | 62.9- | **66.5** | 56.6- | 63.8 | 65.3 |
| Hainsworth | 222 | 70.7 | 68.9 | 66.7 | 64.4 | 69.8 | 66.7 | **72.5** | **72.5** |
| GTZAN Genres | 1000 | **77.0** | 69.2- | 74.6 | 71.1- | 68.5- | 67.8- | 60.4- | 57.9- |
| Dataset Average | 759 | **72.7** | 67.2 | 66.9 | 65.6 | 66.2 | 64.4 | 61.3 | 60.5 |
| Combined Datasets | 3795 | **73.9** | 68.0- | 69.0- | 68.0- | 67.3- | 66.6- | 61.0- | 60.5- |

(a) *Accuracy1*

| Dataset | Songs | schr1 | schr0 | marsyas | gkiokas | zplane | echonest | ibt | qm_vamp |
|---|---|---|---|---|---|---|---|---|---|
| ACM MIRUM | 1410 | 96.0 | 96.5 | 96.0 | **97.8+** | 93.8- | 92.8- | 92.8- | 92.3- |
| ISMIR04 Songs | 465 | 91.8 | **92.0** | 83.2- | 90.8 | 82.4- | 78.5- | 76.8- | 77.9- |
| Ballroom | 698 | 96.3 | 97.6 | 91.6- | **97.7** | 94.4 | 86.1- | 89.8- | 87.8- |
| Hainsworth | 222 | **86.9** | 84.7 | 82.0 | 84.7 | 82.4 | 85.6 | 82.0 | 83.8 |
| GTZAN Genres | 1000 | 92.6 | 92.6 | 90.8 | **92.9** | 88.6- | 86.7- | 86.2- | 85.8- |
| Dataset Average | 759 | 92.7 | 92.7 | 88.7 | **92.8** | 88.3 | 85.9 | 85.5 | 85.5 |
| Combined Datasets | 3795 | 94.1 | 94.4 | 91.4- | **94.9** | 90.5- | 87.8- | 87.9- | 87.5- |

(b) *Accuracy2*

**Table 3**: Tempo results for (a) *Accuracy1* and (b) *Accuracy2* in percent. The $+$ and $-$ signs indicate a statistically significant difference between an algorithm and schr1. Bold numbers mark the best-performing algorithm(s) for a dataset. "Dataset Average" is the mean of the algorithms' results for each dataset. "Combined Datasets" is the accuracy over all datasets. schr0 is schr1 without octave correction.

tempo. We tested for statistical significance with McNemar's test and a significance value of $p < 0.01$, see [2]. Table 3 shows the results computed with data kindly made available by Tzanetakis and Percival. For *Accuracy1*, schr1 performs either as well or better, often significantly, than all other algorithms. In particular, *Accuracy1* for the combined datasets is with 73.9% significantly higher. For *Accuracy2*, schr1 reaches values similar to the best performing algorithm gkiokas. With an *Accuracy2* of 94.1% for the combined datasets, schr1 performs significantly better than all other algorithms except the much more complex gkiokas and the baseline method schr0.

In Table 4, we have analyzed the errors of the various algorithms with regard to *Accuracy1*. For example, marsyas [5] scores lower than schr1 since there are slightly more tempo confusions by a factor of two (9.6% compared to 8.5%) and also for factors/quotients beyond three (8.6% compared to 5.9% in other). Furthermore, gkiokas [24] has a relatively large percentage (14.6%) for tempo confusions by factor of two—something they addressed for ballroom genres in [10]. Summarizing, while schr1 has the fewest octave errors of any of the tested systems, tempo confusions by factor or fraction of two remain the biggest challenge for the best performing systems.

| Algorithm | ×1/2 | ×2 | ×1/3 | ×3 | other |
|---|---|---|---|---|---|
| schr1 | 11.7 | 8.5 | 0.0 | 0.1 | 5.9 |
| schr0 | 10.8 | 14.6 | 0.4 | 0.7 | 5.6 |
| marsyas | 11.7 | 9.6 | 0.7 | 0.5 | 8.6 |
| gkiokas | 10.4 | 14.6 | 1.3 | 0.5 | 5.1 |
| zplane | 8.9 | 13.9 | 0.0 | 0.4 | 9.5 |
| echonest | 8.2 | 12.2 | 0.6 | 0.3 | 12.2 |
| ibt | 6.8 | 19.6 | 0.0 | 0.6 | 12.1 |
| qm_vamp | 4.7 | 21.4 | 0.0 | 0.8 | 12.5 |

**Table 4**: Percentages of the reported results for the combined datasets that are equal to a certain integer multiple or fraction of the ground truth (4% tolerance). Base data for third party algorithms obtained courtesy of Tzanetakis and Percival.

## 5. CONCLUSIONS

We have presented a very simple and effective tempo estimation algorithm that combines standard pulse detection with a continuous tempo octave estimation using the single global feature $\mathrm{SNM}_{82}$. Broad experimental evaluation shows that our method performs as well or significantly better than other state-of-the-art algorithms for a large, mixed-genre dataset. This indicates that perceptual global features can play an important role in tempo octave estimation. In the future, we plan to evaluate other global features to improve genre-specific tempo octave estimation.


## 6. ACKNOWLEDGEMENTS

We would like to thank George Tzanetakis and Graham Percival for sharing their detailed results.


## 7. REFERENCES


[1] Eric D. Scheirer, "Tempo and beat analysis of acoustical musical signals," *Journal of the Acoustical Society of America*, vol. 103, no. 1, pp. 588–601, 1998.

[2] Jose R. Zapata and Emilia Gómez, "Comparative evaluation and combination of audio tempo estimation approaches," in *42nd AES Conference on Semantic Audio*, Ilmenau, Germany, 2011.

[3] Fabien Gouyon, Anssi P. Klapuri, Simon Dixon, Miguel Alonso, George Tzanetakis, Christian Uhle, and Pedro Cano, "An experimental comparison of audio tempo induction algorithms," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1832–1844, 2006.

[4] Peter Grosche, Meinard Müller, and Craig Stuart Sapp, "What makes beat tracking difficult? A case study on Chopin Mazurkas," in *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR)*, Utrecht, The Netherlands, 2010, pp. 649–654.

[5] George Tzanetakis and Graham Percival, "An effective, simple tempo estimation method based on self-similarity and regularity," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, 2013.

[6] Jason Hockman and Ichiro Fujinaga, "Fast vs slow: Learning tempo octaves from user data.," in *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR)*, Utrecht, The Netherlands, 2010, pp. 231–236.

[7] Mark Levy, "Improving perceptual tempo estimation with crowd-sourced annotations.," in *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*. 2011, pp. 317–322, University of Miami.

[8] Martin F McKinney and Dirk Moelants, "Deviations from the resonance theory of tempo induction," in *Proc. Conference on Interdisciplinary Musicology*, Graz, Austria, 2004.

[9] Geoffroy Peeters and Joachim Flocon-Cholet, "Perceptual tempo estimation using GMM-regression," in *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies*, New York, NY, USA, 2012, MIRUM '12, pp. 45–50, ACM.

[10] Aggelos Gkiokas, Vassilios Katsouros, and George Carayannis, "Reducing tempo octave errors by periodicity vector coding and svm learning.," in *Proceedings of the 13th International Conference on Music Information Retrieval (ISMIR)*. 2012, pp. 301–306, FEUP Edições.

[11] Björn Schuller, Florian Eyben, and Gerhard Rigoll, "Tango or waltz?: Putting ballroom dance style into tempo detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2008, pp. 12, 2008.

[12] Olivier Lartillot and Petri Toiviainen, "MIR in Matlab (II): A toolbox for musical feature extraction from audio," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Vienna, Austria, 2007, pp. 127–130.

[13] Jonathan Foote, "Automatic audio segmentation using a measure of audio novelty," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, New York, NY, USA, 2000, pp. 452–455.

[14] George Tzanetakis and Perry Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[15] Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B. Sandler, "A tutorial on onset detection in music signals," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 1035–1047, 2005.

[16] Peter Grosche and Meinard Müller, "Extracting predominant local pulse information from music recordings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 6, pp. 1688–1701, 2011.

[17] Anssi P. Klapuri, "Sound onset detection by applying psychoacoustic knowledge," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Washington, DC, USA, 1999, pp. 3089–3092.

[18] Jonathan Foote and Shingo Uchihashi, "The beat spectrum: A new approach to rhythm analysis," in *Proceedings of the International Conference on Multimedia and Expo (ICME)*, Los Alamitos, CA, USA, 2001.

[19] Geoffroy Peeters, "Template-based estimation of time-varying tempo," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, pp. 158–158, 2007.

[20] Peter Grosche, Meinard Müller, and Frank Kurth, "Cyclic tempogram – a mid-level tempo representation for music signals," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Dallas, Texas, USA, 2010, pp. 5522 – 5525.

[21] M. Alonso, B. David, and G. Richard, "A study of tempo tracking algorithms from polyphonic music signals," in *In 4-th COST 276 Workshop*, 2003.

[22] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten, "The weka data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009.

[23] Stephen Webley Hainsworth, *Techniques for the Automated Analysis of Musical Audio*, Ph.D. thesis, University of Cambridge, UK, September 2004.

[24] Aggelos Gkiokas, Vassilis Katsouros, George Carayannis, and Themos Stafylakis, "Music Tempo Estimation and Beat Tracking by Applying Source Separation and Metrical Relations," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Kyoto, Japan, March 2012.

[25] João Lobato Oliveira, Matthew E. P. Davies, Fabien Gouyon, and Luís Paulo Reis, "Beat tracking for multiple applications: A multi-agent system architecture with state recovery," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 20, no. 10, pp. 2696–2706, 2012.

[26] João Lobato Oliveira, Fabien Gouyon, Luis Gustavo Martins, and Luís Paulo Reis, "Ibt: A real-time tempo and beat tracking system," in *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR)*, 2010, pp. 291–296.

[27] Matthew E. P. Davies and Mark D. Plumbley, "Context-dependent beat tracking of musical audio.," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 15, no. 3, pp. 1009–1020, 2007.