EXTRACTION OF METRICAL STRUCTURE FROM MUSIC RECORDINGS

Elio Quinton, Mark Sandler
Centre for Digital Music,
Queen Mary University of London
London, UK
e.quinton@qmul.ac.uk
m.sandler@qmul.ac.uk

Christopher Harte
Melodient Limited
UK
chris@melodient.com

ABSTRACT

Rhythm is a fundamental aspect of music and metrical structure is an important rhythm-related element. Several mid-level features encoding metrical structure information have been proposed in the literature, although the explicit extraction of this information is rarely considered. In this paper, we present a method to extract the full metrical structure from music recordings without the need for any prior knowledge. The algorithm is evaluated against expert annotations of metrical structure for the GTZAN dataset, each track being annotated multiple times. Inter-annotator agreement and the resulting upper bound on algorithm performance are evaluated. The proposed system reaches 93% of this upper limit and largely outperforms the baseline method.

1. INTRODUCTION

Rhythm is a fundamental aspect of music and extraction of its properties from audio is a wide field of research. In this paper, we focus on the metrical structure of music. The model we use characterises the metrical structure by the hierarchical organisation of the underlying metrical levels and is described in more details in section 2. This model takes inspiration from music theory works such as [1], [2] or [3].

Information about metrical structure has been approached in different ways in Music Information Retrieval research. Various mid-level features such as *beat spectrum* [4], *fluctuation pattern* [5], *inter-onset histograms* [6], or *periodicity spectra* [7] have been used to perform specific tasks such as tempo estimation and beat tracking [8, 9] or classification and similarity [10, 11, 12, 13, 14, 15]. These features usually represent information about periodicities present in the audio signal, which are related, but not necessarily equivalent, to the pulse rates of the metrical levels. In other words, some information about the metrical structure is implicitly encoded in these features which is then used to perform other tasks. Information about metrical structure is not directly extracted from these features.

On the other hand, there is a small body of work aiming at specifically extracting some metrical information. For example, Gouyon proposed a method to produce a dichotomy between duple and triple meter [16]. The Echo Nest API[1] offers as “meter” assessment an integer number that specifies “how many beats are in each bar”. Klapuri proposed a method to simultaneously extract three metrical levels that he describes as the “most important” ones [17]: the *tatum*, *tactus* and the *measure* levels. *Tatum* stems from ‘temporal atom’ and represents the shortest inter-onset interval present in the music. The *tactus* is typically defined as the rate at which listeners would tap along to the music. Tactus is also commonly associated with the *tempo* of a piece, although this view has been challenged [18]. The *measure* is defined as “[...] typically related to the harmonic change rate or to the length of a rhythmic pattern” [17]. In a similar fashion, Uhle proposed a method for estimation of tempo, “micro time” (relating the tatum period to tempo) and time signature [19]. Srinivasamurthy performed a study on the case of carnatic music [20], tracking the *sama* and *aksara* in order to characterise the tala cycle. The *aksara* is the smallest time unit of the cycle, so in that respect is analogous to the *tatum*. The *sama* is “the first aksara” of the cycle, that is to say the starting point of the cycle, which is analogous to the *measure* defined by Klapuri. Similar to the feature introduced by Peeters to perform rhythm classification in [10], Robine defines Meter Class Profiles [21] as vectors of thirteen dimensions representing the relative strength of pulses at rates related in a fixed set of integer ratios to the tempo (which is required as prior knowledge). As such, they can contain information about more than three metrical levels, but don’t explicitly extract such information. Moreover, their discriminative power is only evaluated on the basis of time signature classes, thereby neglecting a part of the metrical structure. Robine notes that some information of interest is overlooked by such a reduction and this is a shortcoming we aim to tackle in this paper. At the exception of Lartillot’s Matlab Toolbox [22], which we use as a baseline, none of these methods involve the direct extraction of the full metrical hierarchy.

The approach presented here aims at explicitly extracting the full metrical structure of a musical piece without requiring any prior knowledge. The structure adapts to the music and is therefore not limited in terms of number of metrical levels represented; their relationships only limited by the structural formalism described in section 2.

In order to evaluate the algorithm, we collected metrical structure annotations for the GTZAN dataset[2] from formally trained professional musicians. Each track of the dataset has been annotated by multiple annotators so that the inter-annotator disagreement and the resulting upper limit of achievable algorithm performance have been assessed [23].

In section 2 we introduce the formalism used to describe the metrical structure. The extraction algorithm is described in section 3, the evaluation using the new annotations is described in

[1] http://developer.echonest.com/docs/v4

[2] The annotations are publicly available for download at the following address:
http://www.isophonics.net/content/metrical-structure-annotations-gtzan-dataset

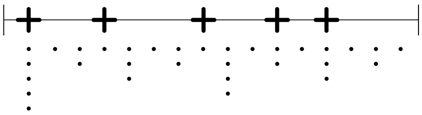

Figure 1: *A simple rumba clave rhythm pattern represented by the crosses. Each horizontal line of dots represents an underlying metrical level implied by the repetition of the pattern. Their hierarchical organisation is used to characterise the metrical structure*

section 4 and results presented and discussed in section 5.

2. FORMALISING THE METRICAL HIERARCHY

The metrical structure of a music piece will be characterised here by the hierarchical organisation of its underlying metrical levels. Figure 1 illustrates the derivation of metrical levels from an example rhythm pattern. Naturally, the underlying metrical levels structure is dependent on the rhythm content of a musical piece. Figure 2 shows a hierarchical representation of metrical structure for several examples. Each horizontal level of nodes on the tree accounts for one metrical level (index $i \in [0, L]$), which is associated with a frequency, or rate f_i measured in BPM (Beats Per Minute). The number of metrical levels necessary to represent the rhythm hierarchy of a piece of music is therefore $L+1$. These rates can be grouped in ascending order in a vector $M = (f_0, f_1, \cdots, f_L)$. Hierarchical relationships are defined by the number of child nodes $\lambda_i \in \mathbb{N}$ each level generates. This implies that $\lambda_i = \frac{f_i}{f_{i-1}}$. A sequence of frequency ratios $\Lambda = \langle \lambda_1, \cdots, \lambda_i, \cdots, \lambda_L \rangle$ is defined. It contains only hierarchical relationships between the metrical levels and therefore can be used for tempo-independent analysis. Retrieving M from Λ only requires the provision of one absolute point of reference, that is one metrical rate. For instance $M = f_0 \star \Lambda$, where the symbol $\star$ is used to represent the fact that the frequency f_0 can be recursively multiplied by the elements λ_i of Λ so that $f_i = f_0 \cdot \prod_{k=1}^{i} \lambda_k$, with $i > 0$.

Metrical hierarchy is related in musical stave notation terms to the time signature *and* the note values used in a composition. As an example, a musical piece using eighth notes in a $\frac{3}{4}$ time signature can be represented by Figure 2 (b), with metrical level $i-1$ being the bar level, level i the quarter note level (three quarter notes in one $\frac{3}{4}$ bar) and level $i+1$ being the eighth note level (quarter note divides into two eighth notes). Consider an example having this metrical structure and a quarter note rate of 150BPM, it would result in $M = (50, 150, 300)$ and $\Lambda = \langle 3, 2 \rangle$. If this piece had an additional layer of subdivision, such as sixteenth notes for example, it would result in $M = (50, 150, 300, 600)$, $\Lambda = \langle 3, 2, 2 \rangle$ and one more level of child nodes on a hierarchical tree representation. This example demonstrates that the information encoded by this representation of the metrical structure differs from the one encoded in a time signature notation. For instance, in the two example we just cited, both would easily be scored as $\frac{3}{4}$, but their metrical structure is different as a result of the use of an extra level of subdivision in the latter case.

3. FEATURE EXTRACTION ALGORITHM

Our extraction algorithm is performed on the audio recording of a piece of music and does not require any prior knowledge. The flowchart of the algorithm given in Figure 3 can be broken down into three processing steps. First, an onset detection function is computed from audio using the *superflux* method [24]. Then, we perform an analysis of the periodicities present in the musical signal, with the hypothesis that some of them will correspond to metrical level rates. Finally the metrical structure is estimated by peak-picking the periodicity spectrum. In this section, we describe the two latter stages.

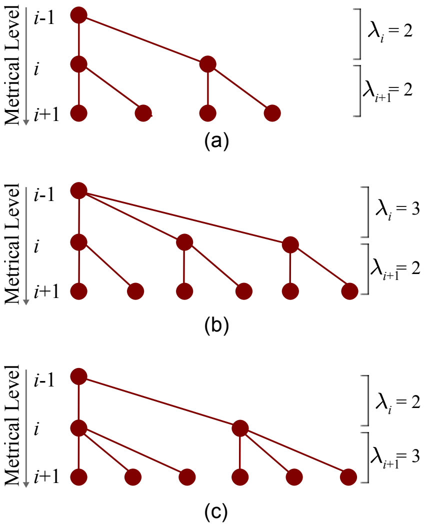

Figure 2: *Tree representation for metrical hierarchy. (a) A simple duple hierarchy dividing the lower level into two groups of two. (b) A simple triple hierarchy dividing the lower level into three groups of two. (c) A compound-duple hierarchy dividing the lower level into two groups of three.*

3.1. Periodicity analysis

In order to perform the periodicity analysis, we rely on the approach introduced by Peeters [25]. Two rhythmograms are calculated in parallel using 12s Hann windows so that low periodicity rates are represented with good resolution and 0.36s hop size in order to maintain good time resolution; the first one, $\mathcal{R}_F(t, f)$ (with t representing time and f frequency), computed using a Fourier transform and the second one, $\mathcal{R}_A(t, f)$, using an autocorrelation function (ACF) with lags converted to a frequency scale. Given the dataset that will be used to carry the evaluation (cf. section 4.2), a certain metrical consistency in the music tracks is assumed. Therefore, the Fourier transform and autocorrelation function based rhythmograms, $\mathcal{R}_F(t, f)$ and $\mathcal{R}_A(t, f)$ respectively, can be summarised in average spectra $\Omega_F(f)$ and $\Omega_A(f)$ by summing frames as given in Equation 1.

$$\begin{aligned} \Omega_F(f) &= \sum_t \mathcal{R}_F(t, f) \\ \Omega_A(f) &= \sum_t \mathcal{R}_A(t, f) \end{aligned} \qquad (1)$$

These time-frequency transformations have the property to highlight the periodicities present in the signal, but also harmonics related to these periodicities. In particular, the spectrum produced using a Fourier transform of a periodic signal contains a series of higher harmonics while the ACF of the same signal would similarly contain a series of sub-harmonics. A strong hypothesis for the work presented here is that the periodicities contained in the onset

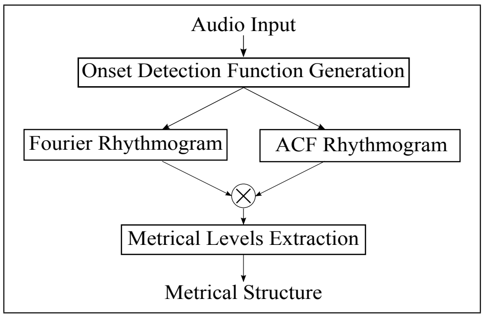

Figure 3: *The feature extraction algorithm is divided in three major steps: computing an onset detection function, performing a periodicity analysis by combining two rhythmograms and finally extracting the metrical structure from the result.*

detection function carry metrical structure information. However, harmonics of these periodicities are artefacts of the mathematical decomposition, which do not represent the periodicities initially present in the signal and therefore do not represent the metrical structure. A composite spectrum $\Omega_C(f)$ is produced by calculating the Hadamard product[3] of the spectra $\Omega_F(f)$ and $\Omega_A(f)$, previously resampled to a common frequency axis with 0.1 BPM resolution, and normalising the result:

$$\Omega_C(f) = \frac{\left(\Omega_A(f) \circ \Omega_F(f)\right)}{\max\limits_f \left(\Omega_A(f) \circ \Omega_F(f)\right)} \tag{2}$$

This approach aims at cancelling out the sub and higher harmonics so that only the periodicities present in the onset detection function remain in the composite spectrum $\Omega_C(f)$ because they are common to the two spectra $\Omega_F(f)$ and $\Omega_A(f)$. Figure 4 illustrates the effect of this approach on an example from the GTZAN dataset.

This track-level configuration is adopted because it suits the dataset used here. However, in a more general setting, the multiplication can be performed for every rhythmogram frame (or group of frames), and therefore capture the temporal evolution of the metrical structure.

3.2. Peak-picking algorithm

As stated earlier, our hypothesis is that metrical levels are represented by periodicities in the onset detection function, and therefore show up as peaks in the spectrum $\Omega_C(f)$. However, experience has shown that not necessarily all the peaks in $\Omega_C(f)$ are related to metrical levels. As a consequence, the metrical structure will be estimated in three steps: peak-picking $\Omega_C(f)$, generating one or more metrical structure candidates and then choosing the one that best fits the data.

First, a simple algorithm detecting local maxima if an element is larger than both of its neighbours is employed to find all the peaks in $\Omega_C(f)$. Only the peaks higher than a given threshold (0.005) are kept.

Secondly, from this list of peaks, the biggest is selected and its abscissa in $\Omega_C(f)$ is labeled $f_{\max}$ (located around 200BPM in the example of Figure 4). This represents the rate containing the most energy in the spectrum and is therefore assumed to represent a salient metrical level. The metrical structure estimation is not sensitive to the choice $f_{\max}$ (i.e. it can equally correspond to any metrical level), however picking the most energetic rate minimises the likelihood of deriving $f_{\max}$ from a spurious peak and therefore maximises the robustness of the system in that respect. By implication the rates of all other metrical levels f_j should be related to $f_{\max}$ by integer ratios (cf. section 2). Then, the abscissa f_j of the j^{th} peak in $\Omega_C(f)$ is compared to $f_{\max}$ and is kept as a candidate level if, and only if, it satisfies one of the following conditions

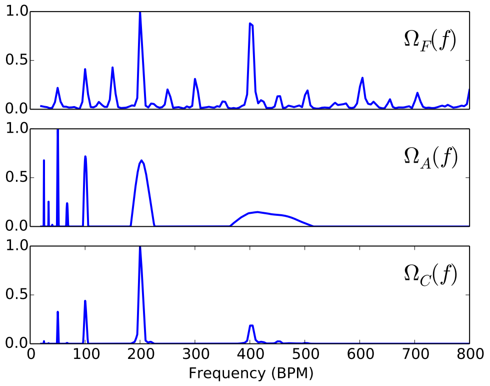

Figure 4: *Example periodicity spectra for the track blues.00053. Respectively from top to bottom, Fourier transform based, $\Omega_F(f)$, autocorrelation function based, $\Omega_A(f)$ and the result of their multiplication, $\Omega_C(f)$. Most of the harmonics in the Fourier and ACF spectra are rejected from $\Omega_C(f)$.*

$$\exists n \in \mathbb{N} : \begin{cases} \frac{f_j}{f_{\max}} = n & \text{if} \quad f_j > f_{\max} \\ \frac{f_{\max}}{f_j} = n & \text{if} \quad f_j < f_{\max} \end{cases} \tag{3}$$

otherwise it is rejected. Finding all the peaks that are integer ratios of $f_{\max}$ is not sufficient to guarantee that they form a hierarchy consistent with the model introduced in section 2, however. The rate of each metrical level and its immediate neighbour must be related by an integer ratio λ_i too.

As a consequence, the last peak-picking step is as follows: starting with $f_{\max}$, iterative comparison of metrical level candidates f_j is performed upwards (comparison with candidates with higher rates) and downwards (comparison with candidates with lower rates). For that purpose, the procedure described by algorithm 1 is applied repeatedly to each candidate until the list of candidates is exhausted. Algorithm 1 applies for the upwards case. The downward case algorithm is easily obtained by symmetry. For each candidate f_j the algorithm considers its two nearest neighbours and appends the successful candidates to the metrical structure, rejects the others and creates additional metrical structure candidates if necessary.

Lines 1 to 3 filter out metrical level candidates not related in integer ratio to f_j. Once an integer ratio $\frac{f_q}{f_j}$ with $q > j$ is found, the second nearest neighbour f_{q+1} is taken in account. A special case occurs when $\frac{f_{q+1}}{f_j}$ is an integer ratio but $\frac{f_{q+1}}{f_q}$ is not. This means that the metrical level f_j could equally be subdivided in levels f_q or f_{q+1} whereas these two levels can't coexist in the

[3] An element by element multiplication denoted as $\circ$

Algorithm 1 Peak-picking kernel: $\mathcal{K}(f_j, M)$

```
Require: $f_j$ is the level under analysis and $M$, the metrical structure candidates
 1: while $\frac{f_{j+1}}{f_j} \notin \mathbb{N}$ do
 2:     $f_{j+1} \leftarrow f_{j+2}$
 3: end while
 4: $f_q \leftarrow f_{j+1}$
 5: if $\frac{f_{q+1}}{f_j} \in \mathbb{N}$ then
 6:     if $\frac{f_{q+1}}{f_q} \notin \mathbb{N}$ then
 7:         $M_1 \leftarrow M$
 8:         $M_2 \leftarrow M$
 9:         $f_j \leftarrow f_q$
10:         $(f_j, M_1) \leftarrow \mathcal{K}(f_j, M_1)$ {call peak-picking kernel}
11:         $M \leftarrow (M, M_1)$
12:         $f_j \leftarrow f_{q+1}$
13:         $(f_j, M_2) \leftarrow \mathcal{K}(f_j, M_2)$ {call peak-picking kernel}
14:         $M \leftarrow (M, M_2)$
15:     else
16:         append $f_{j+1}$ to $M$
17:         $f_j \leftarrow f_{j+1}$
18:     end if
19: else
20:     append $f_{j+1}$ to $M$
21:     $f_j \leftarrow f_{j+1}$
22: end if
23: return $f_j, M$
```

same metrical hierarchy. In such a situation, two parallel hierarchy candidates are generated (lines 7 and 8) and constructed independently by calling two new instances of the peak-picking kernel (lines 9 to 14). Unless this condition is entered, f_{j+1} is appended to the metrical structure, the index of level under analysis is incremented (lines 17 and 21), and the peak-picking kernel called again.

At the end of this stage, hierarchy candidates have been generated, and are represented by their vector M. Finally, for each hierarchy candidate, each one of the metrical levels f_i is associated with a weight $w_i = \Omega_C(f_i)$ stored in $W = (w_0, w_1, \cdots, w_L)$. Each hierarchy candidate is graded by the sum of the weights of its metrical levels $\Theta = \sum\limits_i w_i$. The hierarchy with the biggest cumulated weight Θ is considered as the most salient, and is therefore chosen as the hierarchy that best fits the data. As an example for the track *disco.00045*, for which the various periodicity spectra were given in Figure 4, the metrical hierarchy extracted is $M = (30.7, 61.5, 124.5, 245, 490.1)$ and $W = (0.05, 0.6, 1.0, 0.7, 0.9)$. Considering a quarter note at 124.5 BPM, the vector M represents a metrical structure exclusively based on duple subdivisions that would easily be scored in $\frac{4}{4}$, in which case the 490.1 BPM rate would represent sixteenth notes and the 30.7 BPM rate would represent the bar level.

3.3. Limitations

The metrical structure model used here is fit for representation of any sort of isosynchronous metrical structure. However, it does not enable representation of non-isosynchronous groupings. Consider a meter featuring a cycle of 5 units of a given metrical level grouped in threes and twos notated 3+2 (Dave Brubeck's Take Five is an example of such grouping). In our model the 3+2 grouping would not be accounted for. Nevertheless, the 5 ratio between the cycle and the metrical level used as a base for grouping fits in the model thus accounting for a meter "in five". Expanding the model to include non-isosynchronous metrical groupings representation is an avenue for future work, in which case the technical implementation might need to be adapted accordingly. Fourier transforms are probably not the best formalism to represent non-isosynchronous groupings because they decompose the signal on a basis of sine wave functions, which are intrinsically isosynchronous.

4. ALGORITHM EVALUATION

4.1. Evaluation metrics

Evaluation of the metrical structure extraction algorithm is performed on the GTZAN dataset as follows. For each track, a pairwise comparison of every level of the metrical hierarchy of the annotation (AN) and the extracted feature (EF) is performed. The metrical level rates from a vector M of size N are converted to a logarithmic scale. A binary matrix $\mathcal{M}$ of size $N_{AN} \times N_{EF}$ storing the matching information between extracted feature and annotation is built with each element $\mathcal{M}_{ij}$ defined as:

$$\mathcal{M}_{ij} = \begin{cases} 1 & \text{if } |f_i^{AN} - f_j^{EF}| < \xi \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Consequently, each match between an annotation and an extracted metrical level is associated with the value 1 while mismatches are associated with 0. A tolerance ξ is applied to account for the variability of human rating; its value set at 15% of the annotated value.

In this context, a false negative would be characterised by a row of zeros in the matrix $\mathcal{M}$ because they correspond to levels being present in the annotation but not in the extracted feature. Likewise, a false positive would be characterised by a column of zeros. The number of true positives is obtained by summing all the coefficients $\mathcal{M}_{ij}$ of the matrix. Finally, standard information retrieval system metrics are applied. For each track, Precision, Recall and F-measure are calculated, measuring the performance of the system on each track. Average values of these scores across all tracks of the dataset are then calculated.

An example of such metrics is given below. It corresponds to the evaluation of the extracted metrical structure against one annotation for the track *rock.00029*.

$$\mathcal{M} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

In this case, there are four true positives, i.e. four levels matching, indicated by the ones, one false negative indicated by the last row of zeros and no false positive as there is no column of zeros. It results in Precision=1.0, Recall=0.80 and F-measure=0.89.

4.2. Evaluation Dataset

We have produced expert annotations for the GTZAN dataset [26], which is composed of 1000 music excerpts of 30 seconds duration grouped in 10 genres (with 100 tracks in each group). This dataset covers a range of metrical structures, although simple duple type

Table 1: *System configurations ('methods') under evaluation defined by three parameters: the periodicity spectrum used 'PS', the activation of the second peak-picking step 'PF', and the activation of the peak-picking kernel 'PPK'. Results are presented for each method as well as for the baseline method [22] as Precision, Recall, F-measure and Performance Relative to the Upper Limit (PRUL) scores.*

	PS	PF	PPK	Precision	Recall	F-measure	PRUL
Method 1	$\Omega_C(f)$	on	on	**0.83**	**0.84**	**0.82**	**93.2%**
Method 2	$\Omega_C(f)$	on	off	0.61	0.86	0.68	77.3%
Method 3	$\Omega_C(f)$	off	off	0.51	0.96	0.64	72.7%
Method 4	$\Omega_A(f)$	on	on	0.86	0.77	0.80	90.9%
Method 5	$\Omega_A(f)$	on	off	0.70	0.79	0.72	81.8%
Method 6	$\Omega_A(f)$	off	off	0.43	0.95	0.58	65.9%
Lartillot [22]	-	-	-	0.36	0.55	0.43	48.9%

of meter (typically scored in $\frac{4}{4}$) accounts for a large part of the distribution. Considering the short length of the tracks, it is assumed that the metrical structure is relatively constant throughout the excerpts and consequently only an overall annotation at the track level was produced. This assumption proves right in the vast majority of the cases. The annotators were presented with one, randomly picked track from the dataset at a time and asked to annotate the rate (measured in BPM) of every metrical level they could hear in the music. They could achieve this either by filling in the BPM value directly or by tapping along to automatically measure this rate.

In order to provide an estimation of the reliability of the annotations, the dataset has been entirely annotated by multiple experts. Flexer showed that inter-rater disagreement results in an upper limit for the performance possibly achievable by an algorithm [23]. In our case, for every track, each pair of annotators is considered and the level of inter-annotator agreement is measured using the metrics introduced in section 4.1. Instead of comparing annotation data (AN) and an extracted feature (EF), annotations produced by one annotator are compared with annotations produced by another. The F-measure is used as a figure of merit to assess the agreement for each track, 1 meaning perfect agreement (annotators have annotated a structure that contains exactly the same metrical levels) and 0 meaning complete disagreement (nothing in common in their annotations). The average F-measure obtained across the dataset is then 0.88. This reflects a high level of inter-annotator agreement on average while setting the upper limit of average F-measure possibly achievable by an algorithm on this dataset [23]. In the following, for each track, the extracted feature is evaluated against all the annotations available ; from which are calculated the average values presented below.

4.3. Baseline method

The *mirmetre()* function from the mirtoolbox[4] [22] has been used as a baseline. The metrical structure estimation proposed in [22] comprises three steps that are very similar to the ones in the method presented in this paper. First of all, an onset detection function is processed using a spectral flux method. Secondly an analysis of the periodicities present in this onset detection curve is performed by calculating an ACF rhythmogram (labeled "autocorrelogram" in the original publication). Finally, the metrical structure is estimated from the ACF rhythmogram. In our experiment, we set the window length and hop size identical to the values used for the algorithm described in section 3. All other parameters were set to default values. The metrical structure is returned in the form of a list of metrical level pulse rates. For each metrical level rate, an average value for the entire duration of the track is used for the evaluation.

5. RESULTS

5.1. Experiment

In order to assess the usefulness of the different elements of the algorithm, the evaluation is repeated several times leaving some elements out. The role of three elements is investigated in particular. Firstly, the periodicity spectrum (PS) used either $\Omega_C(f)$, which results from the multiplication of the ACF and Fourier transform-based rhythmograms (cf. section 3), or $\Omega_A(f)$ in which case no multiplication is performed and the metrical structure extraction is performed directly on $\Omega_A(f)$. This enables comparison with the baseline method. Secondly, the peak filtering step described by Equation 3 and labeled 'PF' can be turned on and off. Finally the metrical hierarchy-constrained peak-picking step involving the peak-picking kernel $\mathcal{K}$ of algorithm1 can also be turned on and off and is referred to as PPK. The system configurations under evaluation are given in Table 1 and labeled as 'methods'. Method 1 corresponds to the complete system, as presented in section 3.

5.2. Results and discussion

For all methods under evaluation, we present in Table 1 the results as average precision, recall and F-measure scores for the entire dataset. Only the metrical level rates in the range 30-800BPM are considered for evaluation. The 30BPM lower limit is chosen because periodicity spectra (in particular $\Omega_F(f)$) tend to be very noisy in the 0-25BPM range. The 800BPM limit loosely corresponds to the fastest rate playable by virtuoso musicians[5]. We also calculate the Performance relative to the upper limit (PRUL) implied by the inter-annotator disagreement established in subsection 4.2 to an F-measure of 0.88. Consequently, for each method we have PRUL $= \frac{100 \cdot x}{0.88}$ where x is the corresponding average F-measure.

Comparing the results of Method 1 and 2 clearly shows that constraining the peak-picking algorithm with a musically meaningful model for metrical structure (via the activation of step PPK) results in a substantial increase in performance (0.14 points of F-measure score). This is primarily achieved by increasing preci-

[4] version 1.6.1

[5] Work on music perception such as [2] mention an upper threshold around 100ms (600BPM), but virtuoso playing involves metrical rates around 600BPM and slightly above. Consequently, we increased this limit to 800BPM to leave some headroom.

sion score at the expense of a very small decrease of recall, which means that the PPK step effectively helps picking peaks that correspond to metrical level rates with a very little rate of error. Comparison of methods 2 and 3 reveals that the peak filtering step PF only brings a small improvement, and therefore is not sufficient to extract a meaningful metrical structure on its own. A similar trend emerges from comparison of methods 4, 5 and 6.

Methods 3 and 6 both have the PF and PPK steps deactivated; only the first raw peak-picking step is active (cf. section 3). The evaluation of method 3 enables an assessment of the metrical information captured by $\Omega_C(f)$, from which tempo estimation was performed in [25]. Methods 3 and 6 exhibit similar performance in terms of recall with very high scores (0.96 and 0.95 respectively), which is to be expected because all the peaks present in $\Omega_C(f)$ are still considered at this stage. It means that almost all the metrical level rates are captured as peaks in the periodicity spectra. This was a hypothesis for the design of the extraction process and is validated by the present result. In addition, method 3 scores higher than method 6 in terms of precision. Once again this result is consistent with the assumption that irrelevant peaks would be rejected by the multiplication of $\Omega_A(f)$ and $\Omega_F(f)$. However, the rather low precision (0.51) also demonstrates that $\Omega_C(f)$ does not only contain peaks relating to metrical level rates. From the higher performance reached by method 1, we can conclude that peak-picking strategy materialised by steps PF and PPK is essential to perform accurate metrical structure extraction.

Lartillot's baseline method should be compared with methods 4, 5 and 6, as they all use ACF to estimate periodicities of the onset detection function. In all cases, the baseline method is outperformed. Given that the onset detection function and periodicity estimation used in the baseline method are not largely different from the algorithm presented in this paper, the difference probably resides mostly in the metrical structure estimation steps. As a consequence, it corroborates the idea that the peak-picking of the periodicity spectra is a difficult and sensitive, yet crucial step. Lartillot's peak picking is achieved using some heuristics that are not strongly rooted in music theory whereas our constraining of the metrical structure estimation with a musicologically motivated model proves to be instrumental in achieving an optimal level of agreement with human experts. Method 1, which involves the use of all the processing stages, delivers the best overall performance and achieves the highest F-measure reaching 93.2% of the upper limit imposed by inter-rater disagreement.

6. CONCLUSIONS

We have presented a method for explicit extraction of the full metrical structure from music recordings without the need for any prior knowledge. The extraction process is constrained by a model rooted in music theory, which proves to be a critical step in achieving high performance. The algorithm is evaluated against newly produced annotations of the GTZAN dataset. Taking in account inter-annotator disagreement, we find that our system reaches 93% of maximum achievable accuracy, and largely outperforms the baseline method.

It has been shown that using metrical structure information can help improve beat tracking [27]. The method we have introduced in this paper conforms with expert human judgment and we envision that it could be useful in informing other MIR tasks such as beat-tracking, downbeat estimation and transcription. Moreover, there is evidence that the metrical structure plays an important role in perception of musical pace [2]: "Differences in surface rhythm and metrical structure do interfere with judgments of tempo [in this context meaning how fast the music feels], even if two passages have the same beat rate". The metrical structure as a feature can therefore be useful in the assessment of pace and related tasks. For instance, it could have applications such as automatic music sequencing, music database navigation, or mashup creation and complement systems such as [28].

7. ACKNOWLEDGMENTS

This work is supported by the UK Engineering and Physical Sciences Research Council (EPSRC) and Omnifone Ltd.

8. REFERENCES

[1] Fred Lerdahl and Ray S. Jackendoff, *A Generative Theory of Tonal Music*, MIT Press, 1983.

[2] Justin London, *Hearing in time*, Oxford University Press, 2012.

[3] Chunyang Song, *Syncopation: Unifying Music Theory and Perception*, Ph.D. thesis, Queen Mary University of London, 2014.

[4] Jonathan Foote and Shingo Uchihashi, "The Beat Spectrum: A New Approach To Rhythm Analysis.," in *ICME*, 2001.

[5] Elias Pampalk, Andreas Rauber, and Dieter Merkl, "Content-based organization and visualization of music archives," in *Proceedings of the tenth ACM international conference on Multimedia*, 2002, pp. 570–579.

[6] Simon Dixon, Elias Pampalk, and Gerhard Widmer, "Classification of dance music by periodicity patterns.," in *ISMIR*, 2003.

[7] Andre Holzapfel and Yannis Stylianou, "Scale transform in rhythmic similarity of music," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 1, pp. 176–185, 2011.

[8] Masataka Goto, "An audio-based real-time beat tracking system for music with or without drum-sounds," *Journal of New Music Research*, vol. 30, no. 2, pp. 159–171, 2001.

[9] Simon Dixon, "Evaluation of the audio beat tracking system beatroot," *Journal of New Music Research*, vol. 36, no. 1, pp. 39–50, 2007.

[10] Geoffroy Peeters, "Rhythm Classification Using Spectral Rhythm Patterns.," in *ISMIR*, 2005, pp. 644–647.

[11] Jonathan Foote, Matthew L. Cooper, and Unjung Nam, "Audio Retrieval by Rhythmic Similarity.," in *ISMIR*, 2002.

[12] Matthias Gruhne, Christian Dittmar, and Daniel Gaertner, "Improving Rhythmic Similarity Computation by Beat Histogram Transformations.," in *ISMIR*, 2009, pp. 177–182.

[13] Maria Panteli, Niels Bogaards, and Aline Honingh, "Modeling Rhythm Simirlarity For Electronic Dance Music," in *International Society for Music Information Retrieval Conference*, 2014.

[14] Leigh Smith, *Rhythmic similarity using metrical profile matching*, Ann Arbor, MI: MPublishing, University of Michigan Library, 2010.

[15] Jouni Paulus and Anssi Klapuri, "Measuring the similarity of Rhythmic Patterns," in *ISMIR*, 2002.

[16] Fabien Gouyon and Perfecto Herrera, "Determination of the meter of musical audio signals: Seeking recurrences in beat segment descriptors," in *Audio Engineering Society Convention 114*. 2003, Audio Engineering Society.

[17] Anssi P. Klapuri, Antti J. Eronen, and Jaakko T. Astola, "Analysis of the meter of acoustic musical signals," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 1, pp. 342–355, 2006.

[18] Justin London, "Tactus $\neq$ Tempo: Some Dissociations Between Attentional Focus, Motor Behavior, and Tempo Judgment," *Empirical Musicology Review*, vol. 6, no. 1, pp. 43–55, Jan. 2011.

[19] Christian Uhle and Juergen Herre, "Estimation of tempo, micro time and time signature from percussive music," in *Proc. Int. Conference on Digital Audio Effects (DAFx)*, 2003.

[20] Ajay Srinivasamurthy and Xavier Serra, "A supervised approach to hierarchical metrical cycle tracking from audio music recordings," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. 2014, pp. 5217–5221, IEEE.

[21] Matthias Robine, Pierre Hanna, and Mathieu Lagrange, "Meter Class Profiles for Music Similarity and Retrieval.," in *ISMIR*, 2009, pp. 639–644.

[22] Olivier Lartillot, Donato Cereghetti, Kim Eliard, Wiebke J. Trost, Marc-Andre Rappaz, and Didier Grandjean, "Estimating tempo and metrical features by tracking the whole metrical hierarchy," in *Proceedings of the 3rd International Conference on Music & Emotion (ICME3), Jyvaskyla, Finland, 11th-15th June 2013. Geoff Luck & Olivier Brabant (Eds.)*. 2013, University of Jyvaskyla, Department of Music.

[23] Arthur Flexer, "On inter-rater agreement in audio music similarity," in *International Society for Music Information Retrieval Conference*, 2014.

[24] Sebastian Bock and Gerhard Widmer, "Maximum filter vibrato suppression for onset detection," in *Proc. of the 16th Int. Conf. on Digital Audio Effects (DAFx). Maynooth, Ireland (Sept 2013)*, 2013.

[25] Geoffroy Peeters, "Time variable tempo detection and beat marking," in *Proceedings of the ICMC*, 2005.

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

[27] Norberto Degara, Antonio Pena, Matthew EP Davies, and Mark D. Plumbley, "Note onset detection using rhythmic structure," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. 2010, pp. 5526–5529, IEEE.

[28] Matthew EP Davies, Philippe Hamel, Kazutomo Yoshii, and Misako Goto, "AutoMashUpper: automatic creation of multi-song music mashups," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 12, pp. 1726–1737, 2014.