Bridging the Audio-Symbolic Gap: The Discovery of Repeated Note Content Directly From Polyphonic Music Audio

Tom Collins[1], Sebastian Böck[1], Florian Krebs[1], and Gerhard Widmer[1]

[1] *Department of Computational Perception, Johannes Kepler University Linz, Austria*

Correspondence should be addressed to Tom Collins (tom.collins@jku.at)

ABSTRACT

Algorithms for the discovery of musical repetition have been developed in audio and symbolic domains more or less independently for over a decade. In this paper we combine algorithms for multiple F0 estimation, beat tracking, quantisation, and pattern discovery, so that for the first time, the note content of motifs, themes, and repeated sections can be discovered directly from polyphonic music audio. Testing on deadpan and expressive piano renditions of pieces, we compared pattern discovery performance against runs on symbolic representations of the same pieces. Comparing deadpan audio with deadpan-symbolic representations, establishment precision and recall fell by $\sim 25\%$, and by $\sim 50\%$ when comparing expressive audio with deadpan-symbolic representations. The music data and evaluation results establish a benchmark for future work that attempts to bridge the audio-symbolic gap.

1. INTRODUCTION

Traditionally in music information retrieval, structural analysis of audio has focused on *segmentation*—the construction and labelling of a set of non-overlapping time windows covering the duration of a song or piece [1]. Often segmentation involves calculating and summarising self-similarity matrices [2, 3]. Taking the excerpt by Wolfgang Amadeus Mozart (1756-1791) in Fig. 1A as an example, the single, linear segmentations in Figs. 1B–D are all plausible solutions. They capture aspects of the piece's structure, but no one segmentation covers all of the following:

- The motif P_1, bounded by a solid black line in Fig. 1A, occurs transposed as P_2, and was originally annotated by Schoenberg [4];
- R_1, also bounded by a solid black line, occurs a second time as R_2, and is another of Schoenberg's [4] annotations;
- The theme, according to Barlow and Morgenstern [5], appears bounded by a dashed line and is labelled Q_1. It reappears in the left hand in bars 18–22 (not shown);
- Bars 1–12 end with a repeat mark (see the arrow in Fig. 1A) and so form a repeated section. This is indicated by the dotted line and the labels S_1 (first time) and S_2 (second time).

Advancing beyond single linear segmentation, a recent paper [6] on the structural analysis of audio used scape plots [7] to show multiple segmentations and their relations simultaneously. Segments appeared nested within a large triangle that represented the entire song/piece. Still it was not possible to see/hear the note content of these segments, however. The current paper aims to take one step further, beyond rectangular [1] or nested [6] segments, by enabling the structural analysis of audio to take place at an unprecedented level of detail and meaning; that of note content.

The use of an *automated transcription system* will be key to realising such an aim. Transcription is a 'technique, learned by every music student, of taking aural dictation, in which it is necessary to listen accurately, to construe analytically, and to notate' [8]. The development of automated transcription systems involves several challenging problems, including the *estimation of multiple fundamental frequencies* (multiple F0 estimation) [9, 10, 11, 12] and the *quantisation of musical time*

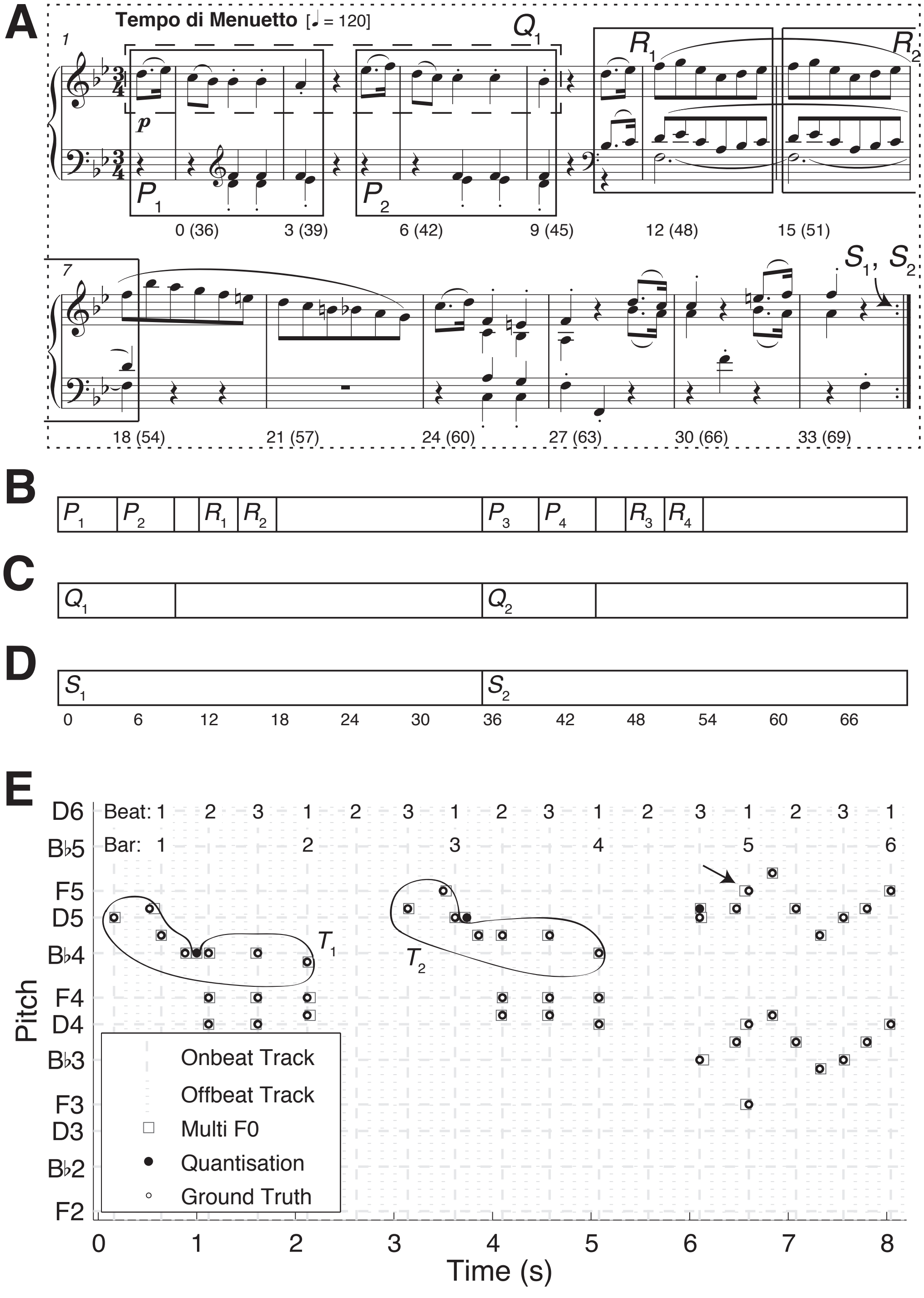

Fig. 1: (A) Bars 1–12 of the second movement from Piano Sonata no.4 in E♭ major K282 by Mozart, annotated with repeated patterns; (B-D) Three plausible, linear segmentations of the excerpt from Fig. 1A. Numbers below the staff in Fig. 1A and below the segmentation in Fig. 1D indicate crotchet beats, from zero for bar 1 beat 1; (E) Transcription of Friedrich Gulda's performance of bars 1–5, and an annotation T_1, T_2 of a pattern discovered by SIARCT-CFP.

[13, 14, 15]. Research has tended to emphasise the former problem over the latter, with frame-based evaluation methods overlooking the quantisation problem entirely [10]. In this paper, algorithms for multiple F0 estimation and quantisation are integrated and evaluated as one, for the first time to our knowledge. Then, using an existing, symbolic method [16] for discovering motifs, themes, and repeated sections such as in Fig. 1A, we evaluate the extent to which meaningful repeated patterns can be identified in the transcriptions of both deadpan and expressive performances.

As most songs and pieces of music are not readily available in machine-readable symbolic formats, being able to broaden the scope of pattern discovery algorithms to handle audio input would constitute a significant step forward. With this aim in mind, the paper is arranged as follows: after a review outlining existing work in pattern discovery, we go on in Sec. 3 to describe our combination of multiple F0 estimation, beat-tracking, quantisation, and pattern discovery algorithms (please see the flow chart in Fig. 2). An evaluation of the system is reported in Sec. 4 for pieces from the Johannes Kepler University Patterns Development Database (JKUPDD); a purpose-built collection for the development and testing of pattern discovery algorithms [17]. The final section discusses the evaluation results and the prospects for applying our system to yet more complex audio signals.

2. REVIEW OF EXISTING WORK

This review focuses on symbolic pattern discovery, but we begin by acknowledging some existing methods for audio structural segmentation that utilise a transcription front-end [18, 19, 20]. As with the systems mentioned above [1, 2, 3], we find the presumed existence of some optimal, linear segmentation to be oversimple and problematic; there are three plausible segmentations of the excerpt from Mozart's K282 (Fig. 1A-D), and this example reflects the often-hierarchical nature of musical repetition.

It is helpful to delineate three classes of music representation to which we refer throughout the paper:

1. Deadpan symbolic (representation derived from the musical score, with metronomically exact timing);

2. Deadpan audio (audio files synthesised from deadpan-symbolic representations);

3. Performed audio (acoustic recordings of expressive human performances).

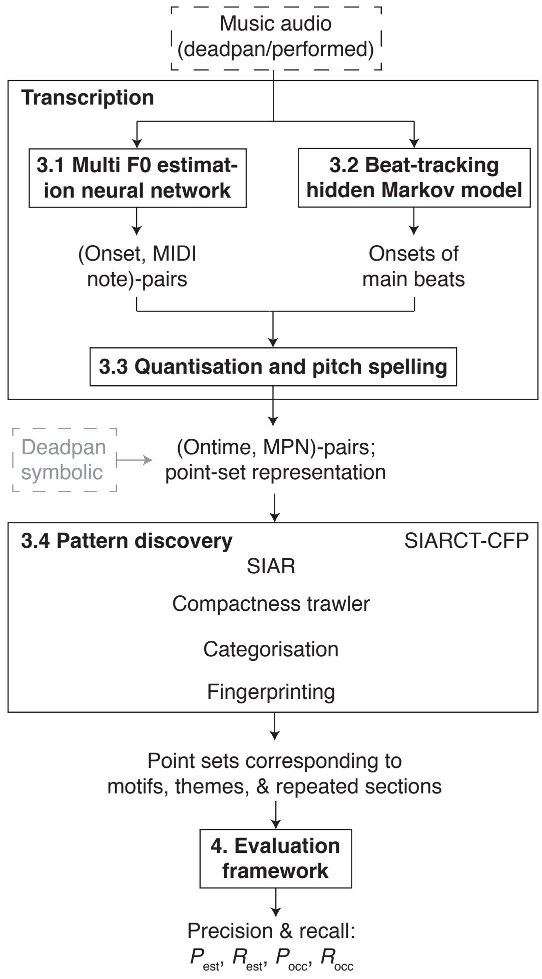
Music audio
(deadpan/performed)
Transcription
3.1 Multi F0 estimation neural network
3.2 Beat-tracking hidden Markov model
(Onset, MIDI note)-pairs
Onsets of main beats
3.3 Quantisation and pitch spelling
Deadpan symbolic
(Ontime, MPN)-pairs; point-set representation
3.4 Pattern discovery
SIARCT-CFP
SIAR
Compactness trawler
Categorisation
Fingerprinting
Point sets corresponding to motifs, themes, & repeated sections
4. Evaluation framework
Precision & recall:
P_{est}, R_{est}, P_{occ}, R_{occ}

Fig. 2: Flow chart depicting the combination of multiple F0 estimation, beat-tracking, quantisation, and pattern discovery algorithms. Input formats are indicated by dashed boxes, algorithms by solid boxes, and intermediary or output formats without boxes.

Not surprisingly, the majority of work on pattern discovery at the note level makes use of deadpan-symbolic representations [21, 16]. A piece is represented as a point set D, consisting of (ontime, MIDI note)-pairs, say. Two subsets $A, B \subset D$ that obey a translational relationship $B = A + \mathbf{c}$ might correspond to two occurrences of some motif, theme, or repeated section [21, 16]. Algorithms that operate on point sets like D, returning sets (or *patterns*) such as A, are called *geometric pattern discovery algorithms*. They are the main method for discovering repetition in unvoiced polyphonic music.

If one were to run a geometric pattern discovery algorithm on a symbolic representation containing expressive timing (e.g., a recording of a performer playing a MIDI instrument), this would cause problems because the exactness of the translations (such as $B = A + \mathbf{c}$) would be lost. While allowing for inexactness at this early processing stage is conceivable (e.g., the distance between each point $\mathbf{a} + \mathbf{c}$ and the corresponding $\mathbf{b}$ is always less than ε), it will be costly in terms of runtime and output size. As such, below we will quantise an incoming set of event times, in order to enable the application of an existing geometric pattern discovery algorithm.

An evaluation framework for pattern discovery algorithms exists, based on robust metrics called *establishment precision* and *establishment recall*, and *occurrence precision* and *occurrence recall* [17]. The *establishment* metrics measure the extent to which an algorithm can discover at least one occurrence of each ground truth repeated pattern (the algorithm may miss several or all other occurrences and still score highly). For instance, an algorithm that returns P_1 from Fig. 1A but misses subsequent occurrences P_2, P_3, and P_4 will still be rewarded. The *occurrence* metrics, on the other hand, assess the extent to which an algorithm finds all occurrences of each ground truth pattern, given that it has discovered one occurrence. For instance, an algorithm that returns all occurrences P_i, Q_j, and R_k from Fig. 1A but misses S_1 and S_2 entirely will still be rewarded.

3. TRANSCRIPTION & DISCOVERY SYSTEM

A flow chart for our transcription and pattern discovery system is shown in Fig. 2. Previous to this paper, discovery of musical structure at the note level relied on a deadpan-symbolic representation, indicated in Fig. 2 by the dashed, grey box. By the end of this section, we will have described how pattern discovery at the note level can be achieved using *audio* as input (deadpan or performed). Parameter choices for the various components—multiple F0 estimation [9], beat-tracking [13], quantisation, and pattern discovery [16]—have for the most part been established in previous work, and so will not be discussed in minute detail here.

We begin with an example of the transcription system in action, given in Fig. 1E. Multiple F0 and onset estimation give the data plotted as grey squares (throughout the term *onset* is used to mean the time in seconds in $\mathbb{R}$ at which an audio event such as a performed note is estimated to begin). The beat-tracking algorithm provides estimates of the times in the audio at which the main beats (or *onbeats*) occur, indicated in Fig. 1E by the vertical dashed lines. The interval between two onbeats is further subdivided by a fixed set of irreducible fractions, giving time estimates for certain *offbeats*, indicated by the vertical dotted lines. Now we have a bijection between (1) a set of on- and offbeat times and (2) symbolic ontimes (the term *ontime* is used to mean the time in $\mathbb{Q}$ measured in crotchet beats at which a note begins, according to a deadpan-symbolic representation of the piece). *Quantisation*—the process of attributing ontimes (and thence bar, beat, and sub-beat numbers) to a set of onsets detected in an audio file or some other continuous-time representation [14]—can now occur by moving an estimated (onset, MIDI note)-pair to the closest on- or offbeat time, and then using the bijection to derive its ontime. For instance, please see the arrow and filled black circles in Fig. 1E: those black circles also containing a white dot correspond to true positives. To make it easy to refer back to the terms *onset*, *onbeat*, *offbeat*, and *ontime*, working definitions are provided in Table 1.

3.1. Multiple F0 Estimation

As shown in the flow chart in Fig. 2, we use a system to convert audio signals into a piano-roll-like representation [9], e.g., (onset, MIDI note)-pairs (please see the grey squares in Fig. 1E). The system uses a neural network consisting of three bidirectional hidden layers with 88 long short-term memory units (LSTM) each [22], to simultaneously detect the onsets and the MIDI numbers of played notes. Bidirectionality and LSTM are intended to increase the network's ability to model the temporal context surrounding a given input value. This is particularly important for note detection, to capture characteristic envelopes accompanying not only the attack phase but also sustain, decay, and release phases of played notes.

The network operates on an input vector of 366 elements, consisting of the logarithmic magnitudes of semitone-filtered spectrograms and their first-order time derivatives, sampled from the 44.1 kHz audio signal at a constant frame rate of 100 fps. The spectrograms are obtained by two parallel short-time Fourier transforms (STFT) with window lengths of 46 and 186 ms, affording both a good temporal and frequency resolution. This enables the system to report even the lowest played notes with a high temporal precision. The network has a regression output layer consisting of 88 units, one for each

Term	Definition
Onset	Estimated time (s) in $\mathbb{R}$ at which an audio event begins.
Onbeat	Time (s) in $\mathbb{R}$ at which a main beat occurs, estimated by beat tracking.
Offbeat	Time (s) in $\mathbb{R}$ at which a simple subdivision of the main beat occurs, estimated by interpolation of beat-tracker output.
Ontime	Time in $\mathbb{Q}$ measured in crotchet beats at which a note begins, according to a deadpan-symbolic representation.

Table 1: Working definitions of onset, onbeat, offbeat, and ontime.

MIDI number. Compared to other state-of-the-art systems using one-versus-all classification [12], the regression output layer has been shown to be superior at distinguishing between fundamental and partial frequency content [9]. The activation function for each note is the output of each of these units, representing the probability that a MIDI note number is sounding at a certain time.

Note onsets and MIDI numbers are determined from the activation functions by smoothing and thresholding on a per-note basis, using a training set consisting of real recordings and synthesised versions of piano music from a range of sources [9, 11, 12], comprising a total of ~1.5 M notes. In the current paper a higher threshold than in the original implementation was chosen, with a value of .45 providing a compromise between false positives and false negatives. The reduction of false positives was of primary concern here, to minimise the size of the input to the pattern discovery algorithm.

3.2. Beat Tracking

The basic idea is to take the audio recording as an observed variable y, and try to infer the hidden variables x of bar position, tempo, and metre, based on a hidden Markov model (HMM) [13, 23, 24]. We track beats using bar position, and agree with Gouyon et al. [25] that: (1) tracking main beats and then subdividing is preferable to direct quantisation; (2) beat tracking based on a continuous feature calculated from audio is preferable to beat tracking based on estimated note onsets (which are available to us via the system for multiple F0 estimation). The joint probability distribution of a sequence of hidden states $x_{1:K}$ and observed states $y_{1:K}$ in our HMM factorises as

$$P(y_{1:K}, x_{1:K}) = P(x_1) \prod_{k=2}^{K} P(x_k|x_{k-1}) P(y_k|x_k). \quad (1)$$

The state space of the hidden variables x is the Cartesian product of three discrete sub-state spaces with the following hidden variables: the position inside a bar represented by equidistant metric positions $m \in \{1,2,...,M\}$, the tempo $n \in \{1,2,...,N\}$ representing integer multiples of $\frac{\Delta m}{\Delta t}$ where $\Delta t = 20\,\text{ms}$ is the audio frame length, and the metre $r \in \{\frac{3}{4}, \frac{4}{4}\}$. Thus, a state in this state space is notated $x_k = [m_k, n_k, r_k]$. To aid inference in this large state space (the total number of states is $M \times N \times R = 1920 \times 35 \times 2 = 134,400$), the system is restricted to three possible state transitions for each state and time frame k, as modeled by the transition probabilities for the tempo variable n:

$$P(n_k|n_{k-1}) = \begin{cases} 1 - p_n, & n_k = n_{k-1}, \\ \frac{p_n}{2}, & n_k = n_{k-1} + 1, \\ \frac{p_n}{2}, & n_k = n_{k-1} - 1, \end{cases} \quad (2)$$

where $p_n = .02$ is the probability of a tempo change. In reality, it could be that the tempo varies more than this from one audio frame to the next, but the restriction is applied in the interests of computational feasibility.

The bar position m_k at time frame k is defined deterministically by

$$m_k = [(m_{k-1} + n_{k-1} - 1) \bmod (M \cdot r_{k-1})] + 1, \quad (3)$$

and the metre is assumed to be constant throughout a piece ($r_k = r_{k-1}$).

The observed states y are represented by a modified onset feature, the *LogFiltSpecFlux* [26], which is computed for two frequency bands (below 250 Hz and above 250 Hz). Considering the onsets in the bass band separately was found to produce higher downbeat and metre detection accuracy [13]. The observation likelihood $P(y_k|m_k, r_k)$ is modeled by a Gaussian mixture model (GMM) with $I = 2$ components as

$$P(y_k|m_k, r_k) = \sum_{i=1}^{I} w_{m_k,r_k,i} \cdot \mathcal{N}(y; \mu_{m_k,r_k,i}, \Sigma_{m_k,r_k,i}), \quad (4)$$

where $\mu_{m_k,r_k,i}$ is the mean vector, $\Sigma_{m_k,r_k,i}$ is the covariance matrix, and $w_{m_k,r_k,i}$ is the mixture weight of component i of the GMM. The parameters of the observation model were trained on pop, rock, and ballroom dance music datasets from [15, 27, 13]. We did not retrain the model for the current application to piano music. Finally, the sequence of hidden states with the maximum a posteriori probability (MAP) $P(m_{1:K}, n_{1:K}, r_{1:K}|y_{1:K})$ is obtained using the Viterbi algorithm [24]. Once the MAP-state sequence is computed, the set of beat times are obtained by interpolating $m_{1:K}^{MAP}$ at the corresponding bar positions.

3.3. Quantisation and Pitch Spelling

The ontimes in a deadpan-symbolic representation of a piece of music are in almost all cases expressible as $\tau = \alpha + \beta$, where $\alpha \in \mathbb{N}$ and $\beta \in F_n$, where n is small and F_n is the Farey series of order n ('the ascending series of irreducible fractions between 0 and 1 whose denominators do not exceed n' [28, p. 23]). For example, $F_4 = \{\frac{0}{1}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{1}{1}\}$ will suffice here. A solution to the quantisation problem consists of a function $f : \mathbb{R} \to \mathbb{N} \oplus F_n$ for mapping an observed onset $t \in \mathbb{R}$ to an ontime $\tau \in \mathbb{N} \oplus F_n$.

As shown in Fig. 2, we use the beat-tracking algorithm to estimate the time in seconds at which crotchet beats (in $\mathbb{N}$) should occur in input audio, and, by interpolation, we estimate the time at which a select number of offbeats (members of $\mathbb{N} \oplus F_4$) should occur. Labelling this set of estimated times T, we have a bijection between T and $\mathbb{N} \oplus F_4$. For an observed onset $t \in \mathbb{R}$, the quantisation function f maps t to the closest member of T, and thence to the corresponding ontime in $\tau \in \mathbb{N} \oplus F_4$.

The final step prior to pattern discovery is to apply a pitch-spelling algorithm to the quantised (ontime, MIDI note)-pairs. Pitch spelling makes it possible to use a numeric representation of *height on the staff* (also called *morphetic pitch number* or MPN, see [21]) rather than MIDI note number. This is preferable because many patterns are translationally exact in the former domain but not the latter. For example, the beginning of P_1 and P_2 in Fig. 1A have MPNs $68, 69, 67$ and $69, 70, 68$ respectively (translationally exact), and MIDI numbers $74, 75, 72$ and $75, 77, 74$ (not translationally exact). Our pitch-spelling component is straightforward, consisting of an estimate of the key of the entire piece, followed by assignment of the most likely pitch class given the key [7].

3.4. Pattern Discovery

For pattern discovery, we use the Structure Induction Algorithm for r superdiagonals and Compactness Trawler, with Categorisation and FingerPrinting (SIARCT-CFP) [16]. It begins with a point-set representation D of a piece, here using (ontime, MPN)-pairs as indicated in Fig. 2. SIAR calculates maximal subsets U_h of D such that $U_h + \mathbf{v}$ is also a subset of D for some vector $\mathbf{v}$. It does so *without* calculating all pairwise differences $(\mathbf{d} - \mathbf{e})$ where $\mathbf{d}, \mathbf{e} \in D$, and is therefore preferable to more exhaustive algorithms [21]. Letting the output of SIAR be $\mathscr{U} = \{U_1, U_2, \ldots, U_H\}$, the Compactness Trawler (CT, [29]) iterates over each member of $\mathscr{U}$, measuring the proportion of contemporaneous notes that are in successively larger regions of U_h (called *compactness* [21]), returning only those subsets that have compactness greater than some threshold a and contain at least b points. There is also a parameter for the definition of a region in the point set—lexicographic or convex hull. Typically, the output of this process, $\mathscr{V} = \{V_1, V_2, \ldots, V_J\}$ is far smaller than $\mathscr{U} = \{U_1, U_2, \ldots, U_H\}$, i.e., $J \ll H$. Members of $\mathscr{V}$ are pruned further by removing a set V_j that has too few unique pitch classes (< 3) and/or that is too short ($\text{ontime}_{\text{end}} - \text{ontime}_1 < 2$ crotchet beats).

It is still possible that members of $\mathscr{V}$ are musically very similar to one another, so a Categorisation process (the 'C' of 'CFP') is used to reduce redunancy. We calculate the symbolic musical similarity of select pairs from $\mathscr{V}$, returning only one of V_j and $V_{j'}$ in the event that the similarity $s(V_j, V_{j'})$ is greater than some threshold c [16]. A perceptually validated model for rating musical importance [30] is used to choose between V_j and $V_{j'}$, as well as to avoid calculation of all pairwise similarities. The output of categorisation is a further reduced set $\mathscr{W} = \{W_1, W_2, \ldots, W_L\}$, where a lower similarity threshold c reduces the number L of output sets, and vice versa. Finally, the symbolic FingerPrinting (FP) method [31, 16] is applied to each W_l to identify more or less exact occurrences of W_l in D. Thus, the output of SIARCT-CFP is a set $\mathscr{X} = \big\{\{X_{l,1}, X_{l,2}, \ldots, X_{l,o_l}\} : l = \{1, 2, \ldots, L\}\big\}$ of L musical patterns, where $X_{l,1}$ is a point set of (ontime, MPN)-pairs corresponding to the prototypical occurrence of some motif/theme/section, and $X_{l,2}, X_{l,3}, \ldots, X_{l,o_l}$ are further occurrences varying in exactness.

In the evaluation, SIARCT-CFP was run twice on each point-set representation, once with parameters known to favour the discovery of motifs,[1] and again with param-

[1] In the symbolic domain, $a = 9/10$, $b = 4$, region = convex hull, $c = 1/3$; for audio, $a = 4/5$ to handle noisier data.

eters aimed toward the discovery of repeated sections.[2] Outputs of each run were concatenated for evaluation. Please see [16] for more details.

4. EVALUATION

The main purpose of this section is to evaluate the performance of pattern discovery algorithms on automatically transcribed audio, compared with symbolic representations of the same music. First, however, it is necessary to quantify and interpret the performance of our transcription system. In the past this has been done using a frame-based approach [10], which is adequate for evaluating the multiple F0 component of the system, but overlooks the beat-tracking and quantisation components. Here we evaluate the performance of the beat tracker in isolation, then we provide an holistic evaluation of the transcription system (beat tracker, quantisation, and multiple F0 components), and finally we evaluate the pattern discovery algorithms on transcribed audio, compared with symbolic representations.

4.1. The Dataset

The evaluation makes use of five movements or pieces from the JKUPDD, containing a total of 26 motifs, themes, and repeated sections [17]. They are:

1. Fugue in A minor BWV889 by Johann Sebastian Bach (1685–1750);

2. The third movement (Scherzo) from Piano Sonata no.1 in F minor op.2 no.1 by Ludwig van Beethoven (1770–1827);

3. Mazurka in B♭ minor op.24 no.4 by Frédéric Chopin (1810–1849);

4. 'The silver swan' (1612) by Orlando Gibbons (1583–1625);

5. The second movement (Menuetto) from Piano Sonata no.4 in E♭ major K282 by Mozart.

The deadpan-symbolic representations are derived from KernScores (http://kern.ccarh.org/), and the ground-truth patterns are based on the annotations of three musicological works [4, 5, 32]. We created deadpan-audio versions of the pieces by synthesising the symbolic representations in Logic Express 9.0.0 at an appropriate tempo with the Yamaha Piano Hall sample. The deadpan-audio files were transcribed using the process explained in Sec. 3, as were performed-audio files for each piece.[3] The first author annotated onbeat times in the performed-audio files, independently of the other authors, for the purposes of evaluating the beat tracker.

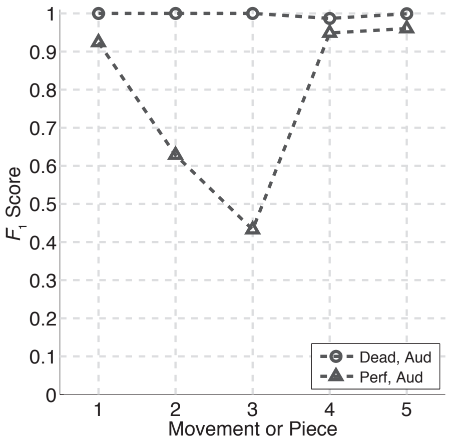

Fig. 3: F_1 score for the beat tracker on deadpan (circles) and performed (triangles) audio for five pieces from JKUPDD [17].

4.2. Evaluation of the Beat Tracker

For each ground-truth onbeat time, we determine if the beat tracker estimates an onbeat time within a ±70 ms window. If so, this is counted as a true positive (TP), and if not, this is counted as a false negative (FN). Any remaining onbeat times estimated by the beat tracker are counted as false positives (FP). The precision is then $P = \mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$, the recall $R = \mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$, and the F_1 score plotted in Fig. 3 is $F_1 = 2PR/(P+R)$. F_1 score is a standard metric in the Audio Beat Tracking task of the Music Information Retrieval Evaluation EXchange (MIREX).[4] (This is also the general identity for F_1 used in other evaluations below.)

As indicated by the circular markers in Fig. 3, Krebs et al.'s [13] algorithm provides perfect (or near-perfect) beat-tracking solutions for deadpan-audio versions of the pieces in our dataset. The algorithm also has strong results for performed-audio versions of pieces 1

[2] In the symbolic domain, $a = 1$, $b = 50$, region = lexicographic, $c = 4/5$; for audio, again $a = 4/5$.

[3] The performers of the five pieces were, respectively, Glenn Gould (Sony, 1993), Friedrich Gulda (Amadeo, 1992), Arthur Rubinstein (RCA Victor Europe / BMG, 1991), the first author, and again Gulda (Deutsche Grammophon, 2006).

[4] http://www.music-ir.org/mirex/wiki/2013:Audio_Beat_Tracking

(J. S. Bach), 4 (Gibbons), and 5 (Mozart), but weaker results for performances of pieces 2 (Beethoven) and 3 (Chopin). This algorithm was highly competitive with other state-of-the-art systems in the most recent MIREX, so its weaker performance on these two pieces is a reliable gauge of the current state of the art, rather than being indicative of a defective approach. Auditioning the results, it was apparent that the beat tracker vacillated between crotchet beats and alternate crotchet beats in piece 2 (Beethoven), leading to poor recall. For piece 3 (Chopin), the amount of rubato employed made accurate tracking very difficult.

4.3. Holistic Evaluation of the Transcriptions

When an algorithm outputs a perfect beat-tracking solution for an audio file, which is the case for Krebs et al.'s [13] algorithm across deadpan audio versions of our dataset, the precision of a whole transcription system (i.e., multiple F0 estimation, beat tracking, and quantisation) can be calculated by $|Y \cap Z|/|Z|$, and the recall by $|Y \cap Z|/|Y|$, where Y is the set of (ontime, MIDI note)-pairs for a ground truth, symbolic representation, and Z is the set of transcribed (ontime, MIDI note)-pairs. If, however, a beat-tracking solution contains false positives or false negatives, there will be shift and/or scale errors in the transcription, relative to the ground truth. With regards shift errors, a metric is required that sums the number of correctly transcribed notes within each shifted segment, weighted punitively by the number of shifts, so that a transcription containing fewer false beats and therefore fewer shifts will score better than a transcription with more shift errors. We use two metrics to fulfil this requirement, called *transcription precision* and *transcription recall*. Their full definition is deferred to the Appendix, so as not to impede the reporting of results.

Figure 4 gives values of *transcription precision* and *transription recall* for deadpan- and performed-audio versions of the dataset pieces. Paralleling the results of the beat-tracking algorithm, the transcription system works well for deadpan audio, with consistently high values of recall. False-positives are more of a problem in deadpan audio than in performed audio, as indicated by the switching of dotted and solid lines between circle and triangle markers in Fig. 4. This is likely because volume levels in the deadpan-audio representations were uniform for prominent and accompanimental material alike, leading to more false-positive doubled octaves than when accompanimental material was played softly by human performers. The transcription of Rubinstein's performance of piece 3 (Chopin's Mazurka) proved particularly difficult, but this was to be expected given the rubato employed in playing mazurkas.

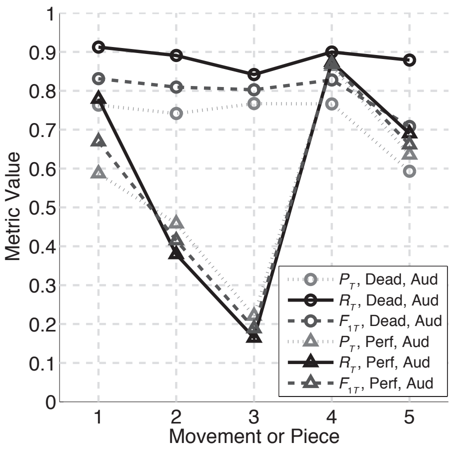

Fig. 4: Evaluation of the transcription system on deadpan (circles) and performed (triangles) audio for five pieces from JKUPDD [17].

4.4. Evaluating the Pattern Discovery Algorithm in Symbolic and Audio Domains

The main component of the evaluation involved running the pattern discovery algorithm SIARCT-CFP [16] on deadpan-symbolic, deadpan-audio, and performed-audio representations of the five pieces. The algorithm output for each piece and each representation was assessed using *establishment recall*, *establishment precision*, *occurrence recall*, and *occurrence precision* [17], as outlined in Sec. 2. To recap, the *establishment* metrics measure the extent to which an algorithm can discover at least one occurrence of each ground truth repeated pattern; the *occurrence* metrics assess the extent to which an algorithm finds all occurrences of each ground truth pattern, given that it has discovered one occurrence. The results are shown in Figs. 5A–D respectively. For the deadpan-symbolic representation, the establishment recall is high (Fig. 5A, solid line), as are the occurrence metrics for a similarity threshold of .75 (Figs. 5C and 5D, solid lines with circle markers). The strength of establishment recall suggests that SIARCT-CFP is effective at discovering at least one occurrence of each ground-truth motif/theme/section in deadpan-symbolic representations. The strength of the occurrence metrics suggests that SIARCT-CFP returns the relevant exact and inexact occurrences of discovered patterns. The precision of the pattern discovery algorithm is in most need of attention, with only one in three patterns being relevant in the best-case scenario (Fig. 5B, solid line, piece 4, Gibbons).

For deadpan-audio representations, performance of the discovery algorithm fell by approximately 25% compared with that of the algorithm applied to symbolic representations of the same pieces (c.f. means of .780 and .609 give a 21.9% decrease in Fig. 5A, .215 and .149 give a 30.8% decrease in Fig. 5B, .747 and .519 give a 30.5% decrease in Fig. 5C, and .783 and .629 give a 19.6% decrease in Fig. 5D, resulting in a mean percentage decrease of 25.7%). A corresponding comparison for the performed-audio representations revealed an approximate 50% drop in performance compared with the performance of the algorithm applied to symbolic representations. Piece 4 (Gibbons) is above this trend in both deadpan- and performed-audio versions, whereas the performed-audio version of Piece 3 (Chopin) pulls down the mean. These pattern discovery results parallel the quality of the initial audio-symbolic transcription, where transcription of both deadpan and performed versions of piece 4 (Gibbons) was strongest, and transcription of the performed version of piece 3 (Chopin) was weakest (c.f. triangles in Figs. 4 and 5A).

5. DISCUSSION

In this study, the first on structural segmentation of polyphonic audio to gain direct access to note content, we have found a drop in metrics of $\sim 25\%$ for the pattern discovery algorithm SIARCT-CFP [16] applied to deadpan-audio representations, compared with deadpan-symbolic representations. In the majority of cases, the algorithm is capable of establishing the existence of motifs, themes, and repeated sections, as well as returning more or less exact instances of patterns. There was a further drop in the metrics, comparing performed-audio representations with deadpan-symbolic representations, with poor transcription of one piece (Chopin's mazurka) being particularly influential here. An example of the type of pattern discovered by SIARCT-CFP for Gulda's performance of the Mozart movement is shown as T_1, T_2 in Fig. 1E. This is a successful discovery, closely related to the ground truth motif annotated as P_1, P_2 in Fig. 1A. By showing the note content, we are able to pinpoint which notes are part of the pattern, going beyond existing work on segmentation [1, 2, 3, 6, 18, 19, 20]. The evaluation dataset is of modest size, due to the time and expertise required to convert music analysts' annotations into a machine-readable ground truth, but our intention is that these music data and evaluations can act as a benchmark for future work attempting to bridge the audio-symbolic gap.[5]

For piece 4 (Gibbons), whose transcription from deadpan audio scored $F_{1,T} = .828$, the establishment recall fell by only .013 $(= .612 - .599)$ between deadpan-symbolic and deadpan-audio versions (Fig. 5A). This result might suggest that perfect transcription is not a prerequisite for the discovery of meaningful patterns in polyphonic music audio. For piece 1 (J. S. Bach), however, with a similar transcription $F_{1,T} = .831$, the fall-off in establishment recall is greater $(.267 = .857 - .590)$. One explanation is that SIARCT-CFP seems to perform best for repeated sections and themes, and less well for motifs. If motifs dominate in the ground truth for a piece, the fall-off in recall may be greater. Compared to music analysts' annotations, where often motifs are nested within themes, SIARCT-CFP parses the entire piece to discover motifs. This parsing of entire piece for motifs is also a reason behind the poor establishment precision in Fig. 5B, and is a strategy that may be revised in the future.

The current paper is one of several in the MIR literature that attempt to combine components designed for specific purposes (here, beat tracking and multiple F0 estimation) in order to undertake some further task (here, pattern discovery). Other examples include [33], where music/voice separation is achieved following extraction of repeated patterns, and [34], where beat tracking and multiple F0 estimation are incorporated in one unified system. As we continue to close the audio-symbolic gap, methods that combine existing components are likely to become more commonplace. It is timely, therefore, to ask whether it is sufficient to arrange such components in a feed-forward, bottom-up manner, or if there is some advantage to be gained by allowing both bottom-up and top-down communication between components. Above, the quality of the beat-tracking solution was a key determinant in the success of the pattern discovery algorithm, so in this respect a bottom-up system is vulnerable to propagation of error from earlier components. Future work should consider whether transcription and pattern discovery can be mutually beneficial, with information about discovered patterns helping to identify errors in transcription, which in turn could improve the discovery of motifs, themes, and repeated sections.

Our solution for multiple F0 estimation generalises well to various piano sounds, but it remains to be seen if a

[5] Please see www.tomcollinsresearch.net for supporting material.

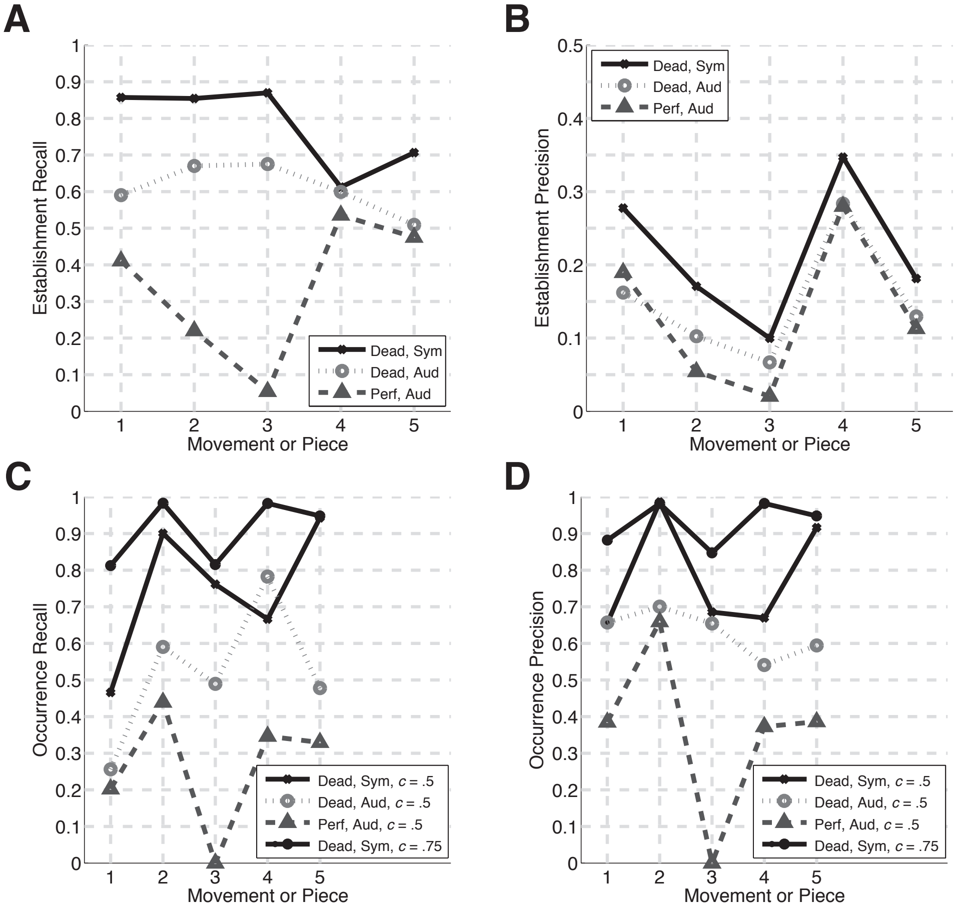

Fig. 5: Evaluation metrics for the SIARCT-CFP pattern discovery algorithm, run on symbolic and transcribed-audio representations of movements/pieces by J. S. Bach, Beethoven, Chopin, Gibbons, and Mozart.

retrained network can generalise to audio material for other or multiple instruments. We are keen to investigate this matter, as the type of nested patterns shown in Fig. 1A are certainly not restricted to classical music. For instance, the excerpt $3'04''$–$3'36''$ of ‘Milk’ from the album *Aka Shake Heartbreak* (2004) by the Kings of Leon consists of a repeating bass pattern lasting four bars, during which time there are four occurrences of a pattern in the lead guitar. While we accept that structural segmentation has an appealing simplicity, and hence utility in music information retrieval, we argue that the annotations made possible in this paper are more accurate and meaningful, depicting multiple musical structures simultaneously, and affording direct access to note content.

6. ACKNOWLEDGMENTS

We would like to thank the three anonymous reviewers for their helpful comments. Thank you to Andreas Arzt and Filip Korzeniowski for technical assistance, and for help with recording the piano performance of Gibbons’ ‘The silver swan’. We also benefited from the use of KernScores. This work was supported by Austrian Science Fund (FWF; grants TRP 109 and Z159) and the EU Seventh Framework Programme FP7 / 2007-2013 through the project PHENICX (grant no. 601166).

7. REFERENCES

[1] R. J. Weiss and J. P. Bello, “Identifying repeated patterns in music using sparse convolutive non-

negative matrix factorization," Proceedings of the International Symposium on Music Information Retrieval, Utrecht, The Netherlands, 2010, 123–128

[2] O. Nieto, E. J. Humphrey and J. P. Bello, "Compressing music recordings into audio summaries," Proc ISMIR, Porto, Portugal, 2012, 313–318

[3] J. Foote, "Automatic audio segmentation using a measure of audio novelty," IEEE International Conference on Multimedia and Expo, New York, NY, 2000 452–455

[4] A. Schoenberg, *Fundamentals of musical composition* Publ., London, UK, 1967; Faber and Faber

[5] H. Barlow and S. Morgenstern, *A dictionary of musical themes* Publ., New York, NY, 1948; Crown Publishers

[6] M. Müller and N. Jiang, "A scape plot representation for visualizing repetitive structures of music recordings," Proc ISMIR, Porto, Portugal, 2012 97–102

[7] C. S. Sapp, "Visual hierarchical key analysis," in ACM Computers in Entertainment **3** (2005), no. 4, 1–19

[8] B. D. Kernfeld, "Transcription," in *The new Grove dictionary of jazz* Publ., New York, NY, 2002; Grove

[9] S. Böck and M. Schedl, "Polyphonic piano note transcription with recurrent neural networks," Proc International Conference on Acoustics, Speech, and Signal Processing, Kyoto, Japan, 2012

[10] M. P. Ryynänen and A. Klapuri, "Polyphonic music transcription using note event modeling," Proc IEEE Workshop on Applications of Signal Processing to Audio, New Paltz, NY, 2005, 319–322

[11] V. Emiya, R. Badeau and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," in IEEE Transactions on Audio, Speech, and Language Processing **18** (2010), no. 6, 1643–1654

[12] G. E. Poliner and D. P. W. Ellis, "A discriminative model for polyphonic piano transcription," in EURASIP Journal on Advances in Signal Processing **2007** (2007), no. 1, 9 pages

[13] F. Krebs, S. Böck and G. Widmer, "Rhythmic pattern modeling for beat and downbeat tracking in musical audio," Proc ISMIR, Curitiba, Brazil, 2013, 227–232

[14] A. T. Cemgil and B. Kappen, "Monte Carlo methods for tempo tracking and rhythm quantization," in Journal of Artificial Intelligence Research **18** (2003), no. 1, 259–273

[15] S. Hainsworth and M. D. Macleod, "Particle filtering applied to musical tempo tracking," in EURASIP Journal on Applied Signal Processing **2004** (2004), no. 15, 2385–2395

[16] T. Collins, A. Arzt, S. Flossmann and G. Widmer, "SIARCT-CFP: improving precision and the discovery of inexact musical patterns in point-set representations," Proc ISMIR, Curitiba, Brazil, 2013, 549–554

[17] T. Collins, "Discovery of repeated themes and sections," Retrieved 4th May 2013, from http://www.music-ir.org/mirex/wiki/2013:Discovery_of_Repeated_Themes_%26_Sections

[18] R. Dannenburg and H. Ning, "Pattern discovery techniques for music audio," in Journal of New Music Research **32** (2003), no. 2, 153–163

[19] J. Paulus and A. Klapuri, "Music structure analysis by finding repeated parts," Proc Audio and Music Computing for Multimedia, Santa Barbara, CA, 2006, 59–68

[20] J. C. Ross, T. P. Vinutha and P. Rao, "Detecting melodic motifs from audio for Hindustani classical music," Proc ISMIR, Porto, Portugal, 2012, 193–198

[21] D. Meredith, K. Lemström and G. A. Wiggins, "Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music," in Journal of New Music Research **31** (2002), no. 4, 321–345

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," in Neural Computation **9** (1997), no. 8, 1736–1780

[23] N. Whiteley, A. T. Cemgil and S. Godsill, "Bayesian modelling of temporal structure in musical audio," Proc ISMIR, Victoria, Canada, 2006, 29–34

[24] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in Proceedings of the IEEE **77** (1989), no. 2, 257–286

[25] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle and P. Cano, "An experimental comparison of audio tempo induction algorithms," in IEEE Transactions on Audio, Speech, and Language Processing **14** (2006), no. 5, 1832–1844

[26] S. Böck, F. Krebs and M. Schedl, "Evaluating the online capabilities of onset detection methods," Proc ISMIR, Porto, Portugal, 2012, 49–54

[27] S. Böck and M. Schedl, "Enhanced beat tracking with context-aware neural networks," Proc International Conference on Digital Audio Effects, Paris, France, 2011

[28] G. H. Hardy and E. M. Wright, *An introduction to the theory of numbers* Publ., Oxford, UK, 1979; Clarendon Press

[29] T. Collins, J. Thurlow, R. Laney, A. Willis and P. H. Garthwaite, "A comparative evaluation of algorithms for discovering translational patterns in baroque keyboard works," Proc ISMIR, Utrecht, The Netherlands, 2010, 3–8

[30] T. Collins, R. Laney, A. Willis and P. H. Garthwaite, "Modeling pattern importance in Chopin's mazurkas," in Music Perception **28** (2011), no. 4, 387–414

[31] A. Arzt, S. Böck and G. Widmer, "Fast identification of piece and score position via symbolic fingerprinting," Proc ISMIR, Porto, Portugal, 2012, 433–438

[32] S. Bruhn, *J.S. Bach's Well-Tempered Clavier: in-depth analysis and interpretation* Publ., Hong Kong, 1993; Mainer International

[33] Z. Rafii and B. Pardo, "REpeating Pattern Extraction Technique (REPET): a simple method for music/voice separation," in IEEE Transactions on Audio, Speech, and Language Processing **21** (2013), no. 1, 73-84

[34] K. Ochiai, H. Kameoka and S. Sagayama, "Explicit beat structure modeling for non-negative matrix factorization-based multipitch analysis," Proc International Conference on Acoustics, Speech, and Signal Processing, Kyoto, Japan, 2012, 133–136.

APPENDIX

Let Y be the set of (ontime, MIDI note)-pairs for a ground truth, symbolic representation, and Z be the set of transcribed (ontime, MIDI note)-pairs.

1. Let $\mathbf{v}_1$ be the most frequent vector in the array $\Delta(Y,Z) = (\mathbf{y}_i - \mathbf{z}_j)_{i \in I, j \in J}$, where $I = \{1, 2, \ldots, |Y|\}$ indexes the set Y, $J = \{1, 2, \ldots, |Z|\}$ indexes Z, and $\mathbf{v}_1(2) = 0$, meaning these are matches between elements of Y and Z for correctly estimated MIDI numbers. Suppose $\mathbf{v}_1$ occurs k_1 times in $\Delta(Y,Z)$.
2. Remove all indices from I and J that are associated with instances of the most frequent vector, as well as the corresponding elements of $\Delta(Y,Z)$. Now (and in general) let $\mathbf{v}_n$ be the most frequent vector in the new version of $\Delta(Y,Z)$, still requiring $\mathbf{v}_n(2) = 0$. Suppose $\mathbf{v}_n$ occurs k_n times in $\Delta(Y,Z)$.
3. Repeat step 2 until, on the Nth occasion, either I or J are empty, or no $\mathbf{v}_n$ exists satisfying $\mathbf{v}_n(2) = 0$.

The *transcription precision* and *transcription recall* are

$$P_T(Y,Z) = \frac{1}{|Z|} \sum_{n=1}^{N} f(n) k_n, \tag{5}$$

$$R_T(Y,Z) = \frac{1}{|Y|} \sum_{n=1}^{N} f(n) k_n, \tag{6}$$

where $f(n)$ is a punitive weighting function, and $f(n) = 1/n$ will be used here. Another weighting strategy would be to weight punitively by the absolute size of the shift error for each segment, $|\mathbf{v}_n(1)|$, giving a function such as $f(n) = \min\{1, 1/|\mathbf{v}_n(1)|\}$. The F_1 measure is calculated in the usual way. Precision and recall for a transcription containing no shift errors may be marginally greater than $|Y \cap Z|/|Z|$ and $|Y \cap Z|/|Y|$ respectively. This is because $f(n)k_1 = 1 \cdot |Y \cap Z|$, and the summands are negligible for $n > 1$, especially when weighted by $f(n)$. Scaling errors are considered a more serious problem, and no attempt is made to reward solutions containing multiple segments that are transcribed correctly up to different scale factors.