METER IDENTIFICATION OF MIDI USING PATTERN DETECTION

Andrew McLeod
University of Edinburgh
`A.McLeod-5@sms.ed.ac.uk`

ABSTRACT

Meter identification is an important component to any Automatic Music Transcription (AMT) program. Much of the existing work on this, especially recently, has concentrated on looking at the salience of the beats of a performance. This work shows that pattern detection may also be helpful when performing meter identification. Specifically, a simple dictionary-based pattern detection model is introduced, along with some preliminary results. The model was evaluated on a corpus of computer-generated MIDI data of the 48 fugues from Bach's Well-Tempered Clavier, and it has chosen the correct meter as one of its top hypotheses for 41 (83%) of them, an improvement of Steedman's 37 (77%) on the same corpus [3].

1. INTRODUCTION

Meter identification is the organisation of the beats of a given musical performance into a tree structure, in which each node represents a single note value. The children of each node divide its note into some number of equal-length notes (usually two or three), where every node at a given level must have an equal value. For example, the metrical structure of a single **3/4** bar, down to the eight note level, is shown in Figure 1.

The task is an integral component of Automatic Music Transcription (AMT), specifically when attempting to identify the time signature of a given performance, since every time signature corresponds to exactly one metrical structure (although there is some ambiguity in the other direction, since it cannot be known whether a given note is a quarter note or an eight note, for example). The metrical structure must be properly aligned in phase with the underlying musical performance so that the root of the tree corresponds with a logical musical segment, often a bar. Further than just AMT, the method we use here for meter identification could have direct relevance to existing work on pattern detection and melody identification.

Existing work is presented in Section 2, and the new method introduced here is described in Section 3. Preliminary results are shown in Section 4, and conclusions and ideas for future work are presented in Section 5.

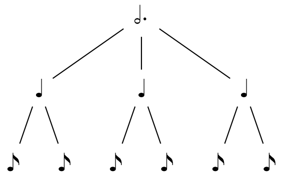

Figure 1. The metrical structure of a **3/4** bar.

2. EXISTING WORK

Some existing work tries to detect a performance's meter by looking for a regular pattern of beat length deviations. Cemgil et al. [1], for example, argue that longer beats are more likely to fall on rhythmically accented notes, which are in turn more likely to occur on metrically stressed beats (those which occur at the beginnings of nodes closer to the root of the metrical structure).

In addition to beat length deviations, which are only explicit in live performances, some rhythmic cues which can aid in meter identification are evident from both the score and live performances. Longuet-Higgins and Steedman [2] first explored this idea and found that certain rhythmic patterns are more likely to occur at certain levels of the tree structure, and can therefore be used to infer the meter of a song. Steedman [3] continued the work, noting that repeated patterns of notes are often found at identical levels of the beat hierarchy. Even with the promise shown in their work, few subsequent meter identification programs have incorporated similar rhythmic or melodic patterns.

3. PROPOSED METHOD

The method proposed here is designed to be run on MIDI input, and performs meter identification by looking for repeated patterns within a single voice based on a dictionary of rhythmic and melodic pattern types. Each pattern consists of a sequence of one or more of three basic pattern constructs: (1) a duration match, (2) an exact match, and (3) a rhythmic match. A duration match construct matches any two segments of MIDI of identical length, no matter the underlying notes. An exact match construct matches any two segments of MIDI containing only notes of identical pitch and value. A rhythmic match construct matches two segments of MIDI which contain notes of identical value and an optional interval constraint of either (1) identical (either both ascending or both descending) or (2) opposite (one ascending and one descending), each within a 50% semitone error.

Attribution: Andrew McLeod. "Meter Identification of MIDI Using Pattern Detection", Extended Abstracts for the Late-Breaking Demo Session of the 16th International Society for Music Information Retrieval Conference, 2015.

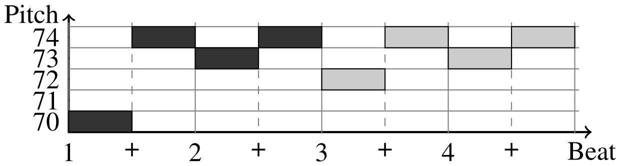

Figure 2. An example of a match of the pattern which contains a 0.5 beat duration match construct followed by a 1.5 beat exact match construct. The different shades represent the two different parts of the underlying MIDI which have matched.

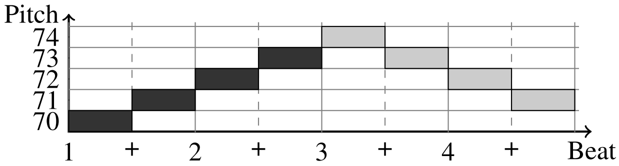

Figure 3. An example of a match of the pattern which contains only a 2 beat rhythmic match construct with opposite interval matching. The different shades represent the two different parts of the underlying MIDI which have matched.

When combining these constructs into patterns, each construct is assigned a minimum and maximum length, between which the length of any matching MIDI segment must fall. A construct's length can be based on the match-length of any preceding construct in its pattern. For example, a pattern could consist of a one to three beat duration match followed by an exact match of three times the length of that duration match. In the case of a rhythmic match construct, the interval match type must be chosen as well. A pattern as a whole is said to match a segment of MIDI if its construct matches are immediately consecutive.

Figure 2 contains an example of MIDI matching a pattern consisting of a duration match construct followed by an exact match construct, while Figure 3 contains an example of MIDI matching a pattern containing only a rhythmic match construct with opposite interval matching. In each figure, the two consecutive matches are differentiated by their shades. In the rhythmic match, note that the intervals between notes in the first match are all -1, while they are $+1$ in the second match.

To identify a performance's metrical structure based on all of its pattern matches, the matches are combined together and possible metrical trees are ranked by the number of pattern matches which it fits. A metrical tree fits a pattern match if and only if that match would fall exactly within a single node of the tree. In this step, two meters with the same structure but different phases (indicating that the same basic meter has been identified, but with bars on different beats), are considered entirely different.

4. PRELIMINARY RESULTS

Rank	Accuracy
Top 1	45.8%
Top 3	68.8%
Top 5	79.1%
Top 10	85.4%

Table 1. The percentage of the fugues whose meter's correct structure was found somewhere within the given ranking of metrical structures hypotheses.

Here, a simple dictionary of only four patterns was used: (1) An exact match construct of 4 to 48 beats; (2) a rhythmic match construct of 4 to 48 beats, and with an identical interval match; (3) a duration match construct of 2 to 4 beats, followed by an exact match construct 1 to 7 times as long; and (4) a duration match construct of 2 to 4 beats, followed by a rhythmic match construct 2 to 4 times as long, and with an identical interval match.

The pattern detection and meter identification processes were evaluated on a corpus of MIDI data of the 48 fugues from Bach's Well-Tempered Clavier. The MIDI files are computer-generated, and the notes are already separated into the voices suggested by the original score. Additionally, they are beat-aligned, and the MIDI files contain some tempo information already, which we use to track the location of every 32nd note in beat identification.

Even with such a simple dictionary of patterns, and a very simple method of ranking tree structures, we achieve some promising results, as shown in Table 1. Compared to Steedman's [3] results of 77.1%, the top 1 result here is lower; however, even with such a small dictionary of patterns, the accuracy already surpasses Steedman's within the top 5 hypotheses, and its performance should only continue to improve as a larger dictionary is used.

5. CONCLUSION

In continuing this work, it would be more musical to define interval matching differently for different interval types (e.g. major and minor thirds), rather than allowing for a 50% semitone error on any interval. Additionally, it would be useful to learn some simple patterns from input MIDI data. Then, the different patterns could be weighted so that those which more often match the correct tree structure would have more influence over the resulting metrical structure. It would also be useful to extend this work to live MIDI performance data, a task which would require both a voice separator and a beat detector as preprocessing, and the ability to recover from errors generated by those processes.

6. REFERENCES

[1] Ali Taylan Cemgil, Peter Desain, and Bert Kappen. Rhythm Quantization for Transcription. *Computer Music Journal*, 24(2):60–76, June 2000.

[2] H. C. Longuet-Higgins and Mark J. Steedman. On Interpreting Bach. *Machine Intelligence*, 6:221–241, 1971.

[3] M J Steedman. The perception of musical rhythm and metre. *Perception*, 6(5):555–69, January 1977.