Rhythm extraction from polyphonic symbolic music

Florence Levé, Richard Groult, Guillaume Arnaud, Cyril Séguin, Rémi Gaymay, Mathieu Giraud

RHYTHM EXTRACTION FROM POLYPHONIC SYMBOLIC MUSIC

Florence Levé, Richard Groult, Guillaume Arnaud, Cyril Séguin **Rémi Gaymay, Mathieu Giraud**
MIS, Université de Picardie Jules Verne, Amiens LIFL, Université Lille 1, CNRS

ABSTRACT

In this paper, we focus on the rhythmic component of symbolic music similarity, proposing several ways to extract a monophonic rhythmic signature from a symbolic polyphonic score. To go beyond the simple extraction of all time intervals between onsets (*noteson* extraction), we select notes according to their length (*short* and *long* extractions) or their intensities (*intensity*$^{+/-}$ extractions). Once the rhythm is extracted, we use dynamic programming to compare several sequences. We report results of analysis on the size of rhythm patterns that are specific to a unique piece, as well as experiments on similarity queries (ragtime music and Bach chorale variations). These results show that *long* and *intensity*$^{+}$ extractions are often good choices for rhythm extraction. Our conclusions are that, even from polyphonic symbolic music, rhythm alone can be enough to identify a piece or to perform pertinent music similarity queries, especially when using wise rhythm extractions.

1. INTRODUCTION

Music is composed from rhythm, pitches, and timbres, and music is played with expression and interpretation. Omitting some of these characteristics may seem unfair. Can the rhythm alone be representative of a song or a genre?

Small rhythmic patterns are essential for the balance of the music, and can be a way to identify a song. One may first think of some clichés: start of Beethoven *5th symphony*, drum pattern from *We will rock you* or Ravel's *Boléro*. More generally, Query By Tapping (QBT) studies, where the user taps on a microphone [10, 12], are able in some situations to identify a monophonic song. On a larger scale, musicologists have studied how rhythm, like tonality, can structure a piece at different levels [5, 16].

This article shows how simple extractions can, starting from a polyphony, build relevant monophonic signatures, being able to be used for the identification of songs or for the comparison of whole pieces.

In fact, most rhythm-only studies in Music Information Retrieval (MIR) concern *audio signal*. These techniques often rely in detection of auto-correlations in the signal. Some studies output descriptors [9, 15, 17] that can be used for further retrieval or classification. Several papers focus on applications of non-Western music [11, 13, 24].

There are other tools that *mix audio with symbolic data*, comparing audio signals against symbolic rhythmic pattern. For example, the QBT wave task of MIREX 2010 proposed the retrieval of monophonic MIDI files from wave input files. Some solutions involve local alignments [10]. Another problem is rhythm quantization, for example when aligning audio from music performances against symbolic data. This can be solved with probabilistic frameworks [2]. Tempo and beat detection are other situations where one extracts symbolic information from audio data [7, 18].

Some rhythm studies work purely on symbolic MIDI data, but where the input is not quantized [22], as in the QBT symbolic task in MIREX 2010. Again, challenges can come from quantization, tempo changing and expressive interpretations. Finally, on the side of *quantized symbolic music*, the Mongeau and Sankoff algorithm takes into account both pitches and rhythms [14]. Extensions concerning polyphony have been proposed [1]. Other symbolic MIR studies focus on rhythm [3, 4, 19–21].

However, as far as we know, a framework for rhythmic extraction from polyphonic symbolic music has never been proposed. Starting from a polyphonic symbolic piece, what are the pertinent ways to extract a monophonic rhythmic sequence? Section 2 presents comparison of rhythmic sequences through local alignment, Section 3 proposes different rhythm extractions, and Section 4 details evaluations of these extractions for the identification of musical pieces with exact pattern matching (Section 4.2) and on similarity queries between complete pieces (Sections 4.3 and 4.4).

2. RHYTHM COMPARISONS

2.1 Representation of monophonic rhythm sequences

For tempo-invariance, several studies on tempo or beat tracking on audio signal use relative encoding [10]. As we start from symbolic scores, we suppose here that the rhythms are already quantized on beats, and we will not study tempo and

meter parameters. If necessary, multiple queries handle the cases where the tempo is doubled or halved.

Rhythm can be represented in different ways. Here, we model each rhythm as a succession of *durations* between notes, i.e. inter-onset intervals measured in quarter notes or fractions of them (Figure 1).

Figure 1. The monophonic rhythm sequence (1, 0.5, 0.5, 2).

Thus, in this simple framework, there are no silences, since each note, except the last one, is considered until the beginning of the following note.

2.2 Monophonic rhythm comparison

Several rhythm comparisons have been proposed [21]. Here, we compare rhythms while aligning durations. Let $S(m, n)$ be the best score to locally align a rhythm sequence $x_1 \dots x_m$ to another one $y_1 \dots y_n$. This similarity score can be computed via a dynamic programming equation (Figure 2), by discarding the pitches in the Mongeau-Sankoff equation [14]. The alignment can then be retrieved through backtracking in the dynamic programming table.

$$S(a,b) = \max \begin{cases} S(a-1,b-1) + \delta(x_a, y_b) & \text{(match, substitution } s\text{)} \\ S(a-1,b) + \delta(x_a, \varnothing) & \text{(insertion } i\text{)} \\ S(a,b-1) + \delta(\varnothing, y_b) & \text{(deletion } d\text{)} \\ S(a-k,b-1) + \delta(\{x_{a-k+1} \dots x_a\}, y_b) & \text{(consolidation } c\text{)} \\ S(a-1,b-k) + \delta(x_a, \{y_{b-k+1} \dots y_b\}) & \text{(fragmentation } f\text{)} \\ 0 & \text{(local alignment)} \end{cases}$$

Figure 2. Dynamic programming equation for finding the score of the best local alignment between two monophonic rhythmic sequences $x_1 \dots x_a$ and $y_1 \dots y_b$. δ is the score function for each type of mutation. The complexity of computing $S(m, n)$ is $O(mnk)$, where k is the number of allowed consolidations and fragmentations.

There can be a *match* or a *substitution (s)* between two durations, an *insertion (i)* or a *deletion (d)* of a duration. The *consolidation (c)* operation consists in grouping several durations into a unique one, and the *fragmentation (f)* in splitting a duration into several ones (see Figure 3).

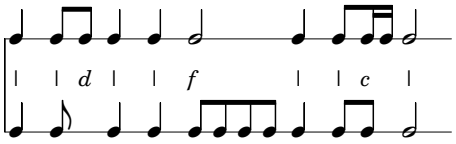

Figure 3. Alignment between two rhythm sequences.

Matches, consolidations and fragmentations respect the beats and the strong beats of the measure, whereas substitutions, insertions and deletions may alter the rhythm structure and should be more highly penalized. Scores will be evaluated in Section 4 where it is confirmed that, most of the time, the best results are obtained when taking into account consolidation and fragmentation operations.

3. RHYTHM EXTRACTION

How can we extract, from a polyphony, a monophonic rhythmic texture? In this section, we propose several rhythmic extractions. Figure 4 presents an example applying these extractions on the beginning of a chorale by J.-S. Bach.

The simplest extraction is to consider all onsets of the song, reducing the polyphony to a simple combined monophonic track. This "*noteson* extraction" extracts durations from the inter-onset intervals of all consecutive groups of notes. For each note or each group of notes played simultaneously, the considered duration is the time interval between the onset of the current group of notes and the following onset. Each group of notes is taken into account and is represented in the extracted rhythmic pattern. However, such a *noteson* extraction is not really representative of the polyphony: when several notes of different durations are played at the same time, there may be some notes that are more relevant than others.

In symbolic melody extraction, it has been proposed to select the highest (or the lowest) pitch from each group of notes [23]. Is it possible to have similar extractions when one considers the rhythms? The following paragraphs introduce several ideas on how to choose onsets and durations that are most representative in a polyphony. We will see in Section 4 that some of these extractions bring a noticeable improvement to the *noteson* extraction.

3.1 Considering length of notes: *long*, *short*

Focusing on the rhythm information, the first idea is to take into account the effective lengths of notes. At a given onset, for a note or a group of notes played simultaneously:

- in the *long* extraction, all events occurring during the length of the longest note are ignored. For example, as there is a quarter on the first onset of Figure 4, the second onset (eighth, tenor voice) is ignored;

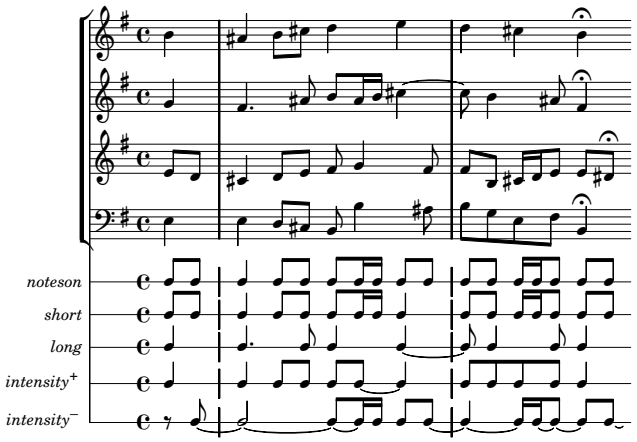

Figure 4. Rhythm extraction on the beginning of the Bach chorale BWV 278.

- similarly, for the *short* extraction, all events occurring during the length of the shortest note are ignored. This extraction is often very close to the *noteson* extraction.

In both cases, as some onsets may be skipped, the considered duration is the time interval between the onset of the current group of notes and the following onset that is not ignored. Most of the time, the *short* extraction is not very different from the *noteson*, whereas the *long* extraction brings significant gains in similarity queries (see Section 4).

3.2 Considering intensity of onsets: *intensity*$^{+/-}$

The second idea is to consider a filter on the number of notes at the same event, keeping only onsets with at least k notes (*intensity*$^{+}$) or strictly less than k notes (*intensity*$^{-}$), where the threshold k is chosen relative to the global intensity of the piece. The considered durations are then the time intervals between consecutive filtered groups. Figure 4 shows an example with $k = 3$. This extraction is the closest to what can be done on audio signals with peak detection.

4. RESULTS AND EVALUATION

4.1 Protocol

Starting from a database of about 7000 MIDI files (including 501 classical, 527 jazz/latin, 5457 pop/rock), we selected the quantized files by a simple heuristic (40 % of onsets on beat, eighth or eighth tuplet). We thus kept 5900 MIDI files from Western music, sorted into different genres (including 204 classical, 419 jazz/latin, 4924 pop/rock). When applicable, we removed the drum track (MIDI channel 10) to avoid our rhythm extractions containing too many sequences of eighth notes, since drums often have a repetitive structure in popular Western music. Then, for each rhythm extraction presented in the previous section, we extracted all database files. For each file, the *intensity*$^{+/-}$ threshold k was choosen as the median value between all intensities. For this, we used the Python framework `music21` [6].

Our first results are on *Exact Song identification* (Section 4.2). We tried to identify a song by a pattern of several consecutive durations taken from a rhythm extraction, and looked for the occurrences of this pattern in all the songs of the database.

We then tried to determinate if these rhythm extractions are pertinent to detect similarities. We tested two particular cases, *Ragtime* (Section 4.3) and *Bach chorales variations* (Section 4.4). Both are challenging for our extraction methods, because they present difficulties concerning polyphony and rhythm: Ragtime has a very repetitive rhythm on the left hand but a very free right hand, and Bach chorales have rhythmic differences between their different versions.

4.2 Exact Song Identification

In this section, we look for patterns of consecutive notes that are exactly matched in only one file among the whole database. For each rhythm extraction and for each length between 5 and 50, we randomly selected 200 distinct patterns appearing in the files of our database. We then searched for each of these patterns in all the 5900 files (Figure 5).

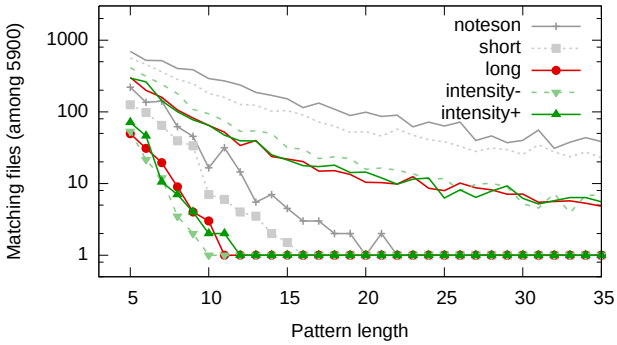

Figure 5. Number of matching files for patterns between length 5 and 35. Curves with points indicate median values, whereas other curves indicate average values.

We see that as soon as the length grows, the patterns are very specific. For lengths 10, 15 and 20, the number of patterns (over 200) matching one unique file is as follows:

Extraction	10 notes	15 notes	20 notes
noteson	49	85	107
short	58	100	124
long	85	150	168
intensity^{+}	91	135	158
intensity^{-}	109	137	165

We notice that the *long* and *intensity*$^{+/-}$ extractions are more specific than *noteson*. From 12 notes, the median values of Figure 5 are equal to 1 except for *noteson* and *short* extractions. In more than 70% of these queries, 15 notes are sufficient to retrieve a unique file.

The results for average values are disturbed by a few patterns that match a high number of files. Figure 6 displays some noteworthy patterns with 10 notes. Most of the time, the patterns appearing very frequently are repetitions of the same note, such as pattern (a). With *long* extraction, 174 files contain 30 consecutive quarters, and 538 files contain 30 consecutive eighths. As these numbers further increase with *noteson* (and *short*) extractions, this explains why the *long* extraction can be more specific.

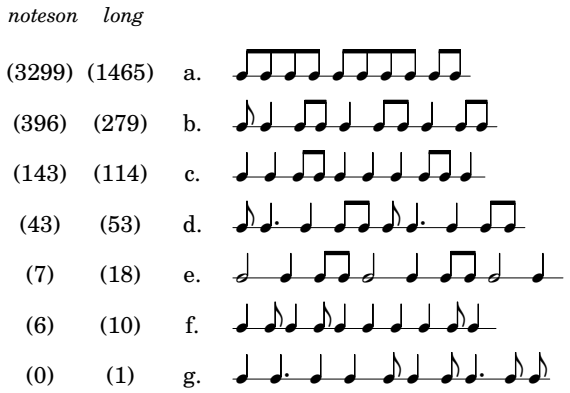

Figure 6. Some patterns with 10 durations, with the number of matching files in *noteson* and *long* extractions.

The number of occurrences of each pattern is mostly determined by its musical relevance. For example, in a pattern with three durations, (d) appears more often than (g), which is quite a difficult rhythm. In the same way, among patterns with only quarters and eighths, (b) and (c) can be found more often than (f). We also notice that patterns with longer durations, even repetitive ones such as pattern (e), generally appear in general less frequently than those containing shorter durations.

4.3 Similarities in Ragtime

In this section and the following, we use the similarity score computation explained in Section 2.2. Ragtime music, one of the precursors of Jazz music, has a strict tempo maintained by the pianist's left hand and a typical swing created by a syncopated melody in the right hand.

For this investigation, we gathered 17 ragtime files. Then we compared some of these ragtime files against a set of files comprising the 17 ragtime files and randomly selected files of the database. We tested several scores functions: always $+1$ for a match, and -10, -5, -2, -1, $-1/2$ or $-1/3$ for an error. We further tested no penalty for consolidation and fragmentation (c/f).

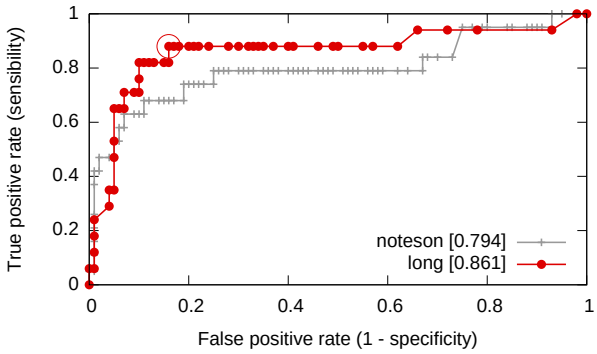

Figure 7. Best ROC Curves, with associated AUC, for retrieving 17 Ragtime pieces from the query *A Ragtime Nightmare*, by Tom Turpin, in a set of 100 files.

Figure 7 shows ROC Curves for *A Ragtime Nightmare*. A ROC Curve [8] plots sensibility (capacity to find true positives) and specificity (capacity to eliminate false positives) over a range of thresholds, giving a way to ascertain the performance of a classifier that outputs a ranked list of results. Here one curve represents one rhythm extraction with one score function. For each score function, we computed the true positive and the false positive rates according to all different thresholds. The *long* extraction, used with scores $+1$ for a match and -1 for all errors, gives here very good results: for example, the circled point on Figure 7 corresponds to 0.88 sensitivity and 0.84 specificity with a threshold of 45 (i.e. requiring at least 45 matches).

Considering the whole curve, the performance of such a classifier can be measured with the AUC (Area Under ROC Curve). Averaging on 9 different queries, the best set of scores for each extraction is as follows:

Extraction	Scores			Mean AUC
	$s/i/d$	c/f	match	
noteson	-5	0	$+1$	0.711
short	-1	-1	$+1$	0.670
long	-1	0	$+1$	**0.815**
intensity^{+}	$-1/3$	0	$+1$	0.622
intensity^{-}	-1	-1	$+1$	0.697

Most of the time, the matching sequences are long sequences of eighths, similar to pattern (a) of Figure 6. If such patterns are frequent in *noteson* database files (see previous section), their presence in *long* files is more frequent in Ragtime than in other musical styles. For example, pattern (a) is found in 76 % of Ragtime *long* extractions, compared to only 25 % of the whole database.

Indeed, in ragtime scores, the right hand is very swift and implies a lot of syncopations, while the left hand is bet-

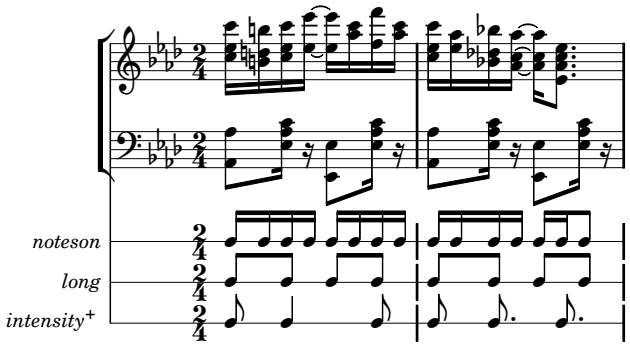

Figure 8. *Possum Rag* (1907), by Geraldine Dobyns.

ter structured. Here the syncopations are not taken into account in the *long* extraction, and the left hand (often made of eighths, as in Figure 8) is preserved during *long* extractions.

Finally, *intensity*$^+$ does not give good results here (unlike Bach Chorales, see next Section). In fact, *intensity*$^+$ extraction keeps the syncopation of the piece, as accents in the melody often involve chords that will pass through the *intensity*$^+$ filter (Figure 8, last note of *intensity*$^+$).

4.4 Similarities in Bach Chorales Variations

Several Bach chorales are variations of each other, sharing an exact or very similar melody. Such chorales present mainly variations in their four-part harmony, leading to differences in their subsequent rhythm extractions (Figure 9).

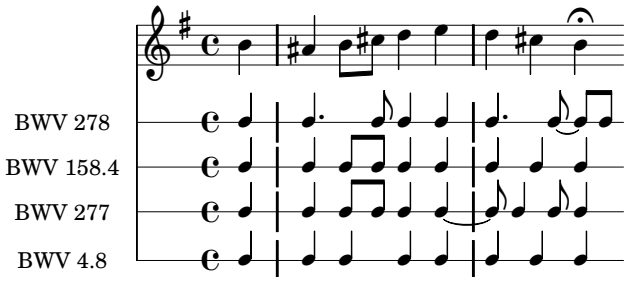

Figure 9. Extraction of *long* rhythm sequences from different variations of the start of the chorale *Christ lag in Todesbanden*. The differences between variations are due to differences in the rhythms of the four-part harmonies.

For this investigation, we considered a collection of 404 Bach chorales transcribed by `www.jsbchorales.net` and available in the `music21` corpus [6]. We selected 5 chorales that have multiple versions: *Christ lag in Todesbanden* (5 versions, including a perfect duplicate), *Wer nun den lieben Gott* (6 versions), *Wie nach einer Wasserquelle* (6 versions), *Herzlich tut mich verlangen* (9 versions), and *O Welt, ich muss dich lassen* (9 versions).

For each chorale, we used one version to query against the set of all other 403 chorales, trying to retrieve the most similar results. A ROC curve with BWV 278 as a query is shown in Figure 10. For example, with *intensity*$^+$ extraction and scores -1 for $s/i/d$, 0 for c/f, and $+1$ for a match, the circled point corresponds to a threshold of 26, with 0.80 sensitivity and 0.90 specificity. Averaging on all 5 chorales, the best set of scores for each extraction is as follows:

Extraction	Scores			Mean AUC
	$s/i/d$	c/f	match	
noteson	-1	0	$+1$	0.769
short	-1	0	$+1$	0.781
long	-5	-5	$+1$	**0.871**
intensity$^+$	-1	0	$+1$	**0.880**
intensity$^-$	-5	0	$+1$	0.619

Even if the *noteson* extractions already gives good results, *long* and *intensity*$^+$ bring noteworthy improvements. Most of the time, the best scores correspond to alignments between 8 and 11 measures, spanning a large part of the chorales. We thus managed to align almost globally one chorale and its variations. We further checked that there is not a bias on total length: for example, BWV 278 has a length of exactly 64 quarters, as do 15% of all the chorales, but the score distribution is about the same in these chorales than in the other ones.

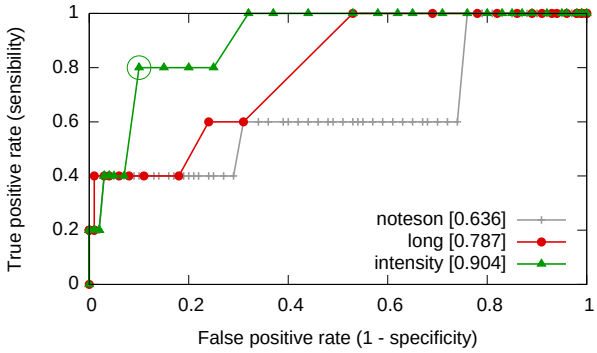

Figure 10. Best ROC Curves, with associated AUC, for retrieving all 5 versions of *Christ lag in Todesbanden* from BWV 278 in a set of 404 chorales.

5. DISCUSSION

In all our experiments, we showed that several methods are more specific than a simple *noteson* extraction (or than the similar *short* extraction). The *intensity*$^-$ extraction could provide the most specific patterns used as signature (see Figure 5), but is not appropriate to be used in similarity queries. The *long* and *intensity*$^+$ extractions give good results in the identification of a song, but also in similarity queries inside a genre or variations of a music.

It remains to measure what is really lost by discarding pitch information: our perspectives include the comparison of our rhythm extractions with others involving melody detection or drum part analysis.

Acknowledgements. *The authors thank the anonymous referees for their valuable comments. They are also indebted to Dr. Amy Glen who kindly read and corrected this paper.*

6. REFERENCES

[1] Julien Allali, Pascal Ferraro, Pierre Hanna, Costas Iliopoulos, and Matthias Robine. Toward a general framework for polyphonic comparison. *Fundamenta Informaticae*, 97:331–346, 2009.

[2] A. T. Cemgil, P. Desain, and H. J. Kappen. Rhythm Quantization for Transcription. *Computer Music Journal*, 24:2:60–76, 2000.

[3] J. C. C. Chen and A. L. P. Chen. Query by rhythm: An approach for song retrieval in music databases. In *Proceedings of the Workshop on Research Issues in Database Engineering*, RIDE '98, pages 139–, 1998.

[4] Manolis Christodoulakis, Costas S. Iliopoulos, Mohammad Sohel Rahman, and William F. Smyth. Identifying rhythms in musical texts. *Int. J. Found. Comput. Sci.*, 19(1):37–51, 2008.

[5] Grosvenor Cooper and Leonard B. Meyer. *The Rhythmic Structure of Music*. University of Chicago Press, 1960.

[6] Michael Scott Cuthbert and Christopher Ariza. music21: A toolkit for computer-aided musicology and symbolic music data. In *Int. Society for Music Information Retrieval Conf. (ISMIR 2010)*, 2010.

[7] Simon Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30(1):39–58, 2001.

[8] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[9] Matthias Gruhne, Christian Dittmar, and Daniel Gaertner. Improving rhythmic similarity computation by beat histogram transformations. In *Int. Society for Music Information Retrieval Conf. (ISMIR 2009)*, 2009.

[10] Pierre Hanna and Matthias Robine. Query by tapping system based on alignment algorithm. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2009)*, pages 1881–1884, 2009.

[11] Andre Holzapfel and Yannis Stylianou. Rhythmic similarity in traditional turkish music. In *Int. Society for Music Information Retrieval Conf. (ISMIR 2009)*, 2009.

[12] Jyh-Shing Jang, Hong-Ru Lee, and Chia-Hui Yeh. Query by Tapping: a new paradigm for content-based music retrieval from acoustic input. In *Advances in Multimedia Information Processing (PCM 2001), LNCS 2195*, pages 590–597, 2001.

[13] Kristoffer Jensen, Jieping Xu, and Martin Zachariasen. Rhythm-based segmentation of popular chinese music. In *Int. Society for Music Information Retrieval Conf. (ISMIR 2005)*, 2005.

[14] Marcel Mongeau and David Sankoff. Comparaison of musical sequences. *Computer and the Humanities*, 24:161–175, 1990.

[15] Geoffroy Peeters. Rhythm classification using spectral rhythm patterns. In *Int. Society for Music Information Retrieval Conf. (ISMIR 2005)*, 2005.

[16] Marc Rigaudière. *La théorie musicale germanique du XIXe siècle et l'idée de cohérence*. 2009.

[17] Matthias Robine, Pierre Hanna, and Mathieu Lagrange. Meter class profiles for music similarity and retrieval. In *Int. Society for Music Information Retrieval Conf. (ISMIR 2009)*, 2009.

[18] Klaus Seyerlehner, Gerhard Widmer, and Dominik Schnitzer. From rhythm patterns to perceived tempo. In *Int. Society for Music Information Retrieval Conf. (ISMIR 2007)*, 2007.

[19] Eric Thul and Godfried Toussaint. Rhythm complexity measures: A comparison of mathematical models of human perception and performance. In *Int. Society for Music Information Retrieval Conf. (ISMIR 2008)*, 2008.

[20] Godfried Toussaint. The geometry of musical rhythm. In *Japan Conf. on Discrete and Computational Geometry (JCDCG 2004), LNCS 3472*, pages 198–212, 2005.

[21] Godfried T. Toussaint. A comparison of rhythmic similarity measures. In *Int. Society for Music Information Retrieval Conf. (ISMIR 2004)*, 2004.

[22] Ernesto Trajano de Lima and Geber Ramalho. On rhythmic pattern extraction in Bossa Nova music. In *Int. Society for Music Information Retrieval Conf. (ISMIR 2008)*, 2008.

[23] Alexandra L. Uitdenbogerd. *Music Information Retrieval Technology*. PhD thesis, RMIT University, Melbourne, Victoria, Australia, 2002.

[24] Matthew Wright, W. Andrew Schloss, and George Tzanetakis. Analyzing afro-cuban rhythms using rotation-aware clave template matching with dynamic programming. In *Int. Society for Music Information Retrieval Conf. (ISMIR 2008)*, 2008.