Extracting Patterns from Guitar Accompaniment Data: Some Experimental Results

Ernesto Trajano de Lima[1], **Søren Tjagvad Madsen**[2,3], **Márcio Dahia**[1], **Gerhard Widmer**[2,3], **Geber Ramalho**[1]

[1]Centro de Informática (CIn) – Universidade Federal de Pernambuco
Caixa Postal 7851– 50732-970 – Recife – Pernambuco – Brazil

[2]Austrian Research Institute for Artificial Intelligence (ÖFAI)
Freyung 6/6, A-1010, Vienna, Austria

[3]Department of Computational Perception – University of Linz, Austria

{etl,mlmd,glr}@cin.ufpe.br, soren@ofai.at, gerhard.widmer@jku.at

Abstract. *The analysis of expressive performance, an important research topic in Computer Music, is almost exclusively devoted to the study of western classical piano music. Instruments like the acoustic guitar and styles like Bossa Nova and Samba have almost never been studied, despite their harmonic and rhythmic richness. This paper describes some experimental results obtained with the extraction of rhythmic patterns from the guitar accompaniment of Bossa Nova songs. The performances, recorded with a MIDI guitar, were represented as strings and processed by three different string matchers. Results obtained showed that parts of a previously acquired catalogue of patterns could be found in the data set. Surprising results, such as very long patterns, also showed up.*

1. Introduction

It is common sense that playing music in the exact way as written in the score results in a mechanical and uninteresting succession of sounds. To make the written music interesting, the musician is required to make variations on various musical parameters, such as: local tempo (*accelerandi*, *ritardandi*, *rubato*); dynamics; notes articulation (*staccati*, ligatures, etc.); micro-silences between the notes, etc. [Widmer, 1998]. Several researchers stress the importance, as well as the the difficulties, of studying this phenomenon, also known as *expressive performance* [Sundberg et al., 1991, Desain et al., 2001, Widmer et al., 2003].

Studying the expressive performance phenomenon usually implies discovering some sort of systematic relationship within a piece and/or among pieces. These relationships can be described in many ways, from different points of view or levels of abstraction, and including various musical parameters. Examples of such relationships are rules like "*lengthen a note if it is followed by a longer note and if it is in a metrically weak position*" or "*stress a note by playing it louder if it is preceded by an upward melodic leap larger that a perfect fourth*" (for more examples see [Widmer, 2002]).

The research on expressive performance today is almost exclusively devoted to the western classical music composed for the piano. We are interested in the study of the *Música Popular Brasileira* (Brazilian Popular Music)—MPB, represented by artists like João Gilberto, Tom Jobim, Caetano Veloso, Gilberto Gil, etc., in particular the guitar music. There is, however, a fundamental difference between these two genres: While in western classical music there is one "official" notated version of musical pieces (the

score), in MPB it does not exist. What is usually available is the description of the chord grid only, and, sometimes, the score for the melody. So, in this genre (MPB), the rhythmic accompaniment must be determined by the performer himself.

It is known that the guitar accompaniment in styles like *Bossa Nova* and *Samba* is built by the concatenation of certain rhythmical patterns [Garcia, 1999, Sandroni, 2001]. There are, however, several aspects of the accompaniment construction that are only known by practitioners of these styles. Moreover, the knowledge about the accompaniment construction is mainly subjective. Due to this lack of formalized knowledge, there are many open questions such as:

- Are there rhythmic patterns that are preferred by a certain performer or required for a certain musical style? In which situations and in which frequency do they show up?
- Are there variations of these patterns? Is it possible to group these variations in meaningful way? Which variations (timing, dynamics, etc.) are acceptable within a pattern?
- Is it really the case that everything is a pattern, i.e., are there parts that are not recurrent?
- Is it possible to justify the choice of a pattern in terms of other musical features (melody, harmony, tempo, musical structure, style, etc.)?
- Is it possible to build a dictionary of patterns for a given player? Does this dictionary changes when the style changes (Bossa Nova and Samba, for instance)? Do different players have different dictionaries?
- Is it possible to build a grammar or a set of rules that is able to describe formally how the patterns are chained and/or the rhythmical transformations done by a given performer? If so, what are the relations between grammars from performers p_1 and p_2?

This paper presents some experiments that deal with some of these questions, focusing on the discovery of rhythmic patterns in Bossa Nova music. For this, two different performers played several songs on a MIDI guitar, which were processed in the form of strings. Based on previous work (the acquisition of a catalogue of Bossa Nova patterns [Dahia et al., 2004]), we tried to identify how much of this catalogue could be automatically found in the data we collected[1].

The remainder of the paper is organized as follows: In Section 2, we describe how the data was acquired and the representation we used. In Section 3, we introduce the algorithms we used. In Section 4, we present the experiment itself, as well as the results we obtained. Finally, in Section 5, we present some conclusions and future directions for this work.

2. Data Acquisition and Representation

For this experiment, two different players (from now on referred to as Player1 and Player2) recorded the accompaniment of some Bossa Nova songs on a MIDI guitar. Player1 performed the following songs: *Bim Bom*, *O Barquinho*, *Insensatez* (How Insensitive), *Garota de Ipanema* (Girl from Ipanema), *Só Danço Samba*, and *Wave*. From Player2 we recorded *A Felicidade*, *Chega de Saudade*, *Corcovado*, *Desafinado*, *Eu Sei Que Vou Te Amar*, *Samba de uma Nota Só*, *Garota de Ipanema*, *Só Danço Samba*, *Insensatez*, *Tarde em Itapoã*, and *Wave*. In the total, we collected 16 recordings (ca. 30 minutes

[1]The catalogue reflects the rhythmic patterns used by João Gilberto, the "inventor" of this style.

of music). It was requested the performers to play the songs according to a provided notation (the chord grid as notated by Chediak [Chediak, 1990]).

The acquired data was, however, not ready for usage. Probably due to technological restrictions, the resulting MIDI files were noisy (they contained several extra notes, i.e., notes the player did not play). So, we had to manually correct them, removing these extra events. After correction, the data was beat tracked at the eight note level using *BeatRoot* [Dixon, 2001], an interactive system that outputs the MIDI file beat tracked.

As we are interested in the discovery of rhythmic patterns, the exact pitches played (i.e., A, B, C♯, etc.) are not that relevant. Apart from the durations, a much more relevant abstraction is the right hand finger used by the player to pluck the string[2]. It becomes even more relevant when you find in the literature (e.g., in [Garcia, 1999]) that musicians usually describe the rhythmic patterns in terms of "baixo" or *bass* (events played with the thumb only) and "puxada" or *chord* (events played with some combinations of two or more fingers). The right hand fingering for each song was determined automatically. Given a MIDI file (beat tracked or not), the algorithm (as described in [Trajano et al., 2004]) outputs the fingering as depicted in Figure 1. Letters T, F, M and R represent, respectively, the thumb, fore, middle and ring fingers, crosses (+) represent the beats, and pipes (|) represent the measure bars. Each beat was equally divided by four, so each letter, cross or minus (–) represents the duration of a 32nd. Except for the last line, that represents exclusively the beats, each of the remaining lines represents one guitar string, ordered from higher to lower (i.e., first line represents the high E string, second line the B string, and so on until the low E string).

```
|-----------------|-----------------|--
|R---R-----R-----|R---R-----R-----|R-
|M---M-----M-----|M---M-----M-----|M-
|F---F-----F-----|F---F-----F-----|F-    [...]
|-----------------|-----------------|--
|T-------T-------|T-------T-------|T-
|+---+---+---+---|+---+---+---+---|+-
```

Figure 1: Right hand fingering for song *Insensatez*, as played by Player2

In order the reduce the complexity of the pattern extraction, this initial representation for the fingering was reduced to a one-dimensional string. This simplified string is formed by the alphabet $\Sigma = \{b, B, p, P, l, a, A, s, S, -, +, |\}$. The meaning of these symbols is the following:

- Uppercase letters stand for events that occur on-beat, while lowercase letters for off-beat events;
- Letter b stands for "bass", i.e., events played with the thumb only;
- Letter p stands for "chord" (sic), i.e., events that are played with some combinations of two or more of fingers F, M and R[3];
- Letter l also stands for "chord", but a chord whose duration goes beyond the measure it was played (it means that we make a difference between a chord that is completely within a single measure and a chord that starts in one measure and ends in the next one);
- Letter a stands for "all", i.e., b and p played together;
- Letter s stands for "single note", i.e., events that are played with only one of fingers F, M and R; and

[2]We assume the player is right handed.

[3]The terms "baixo" and "puxada" may explain more clearly the origin of letters b and p!

- Symbols $+$, $-$, and $|$ have the same meaning stated before.

Note that this reduction is also done by the musicians themselves, when they describe the rhythmic patterns they play as sequences of basses and chords ("baixos" and "puxadas"). Figure 2 depicts part of the fingering for song Insensatez. Above the thick black line is the fingering as output by the right hand fingering algorithm. Under it is the resulting simplified string.

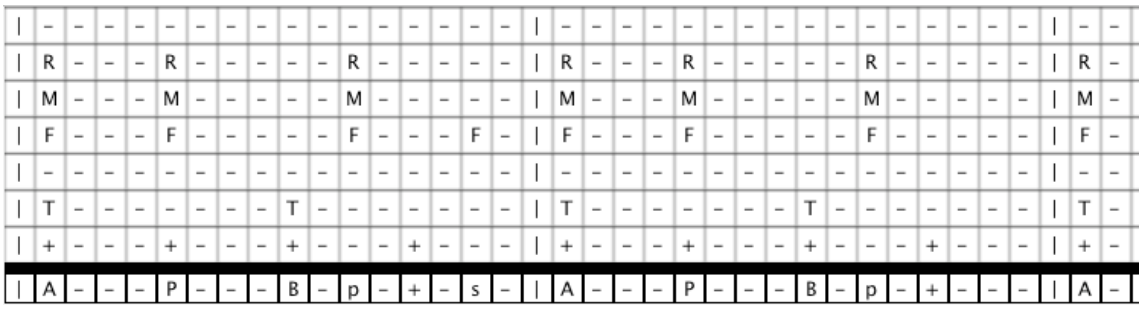

Figure 2: Fingering and 1-dim. string for song Insensatez, as played by Player1

3. Algorithms

For our experiment, we used three algorithms: Boyer–Moore [Boyer and Moore, 1977], FlExPat [Rolland, 2001] and SimilaritySegmenter [Madsen and Widmer, 2005]. The Boyer–Moore algorithm is an exact matcher that first preprocesses the pattern and only then tries to find occurrences of it in a text. Because it compares only a fraction of the characters in the original text, it can run in sub-linear time, and is the preferred method in most string matching applications [Gusfield, 1997].

FlExPat is an inexact string matching algorithm that was inspired by algorithms from the Computational Biology field, but that also incorporates results from previous research on musical similarity [Mongeau and Sankofff, 1990]. Given an input (here the simplified string previously described) and using the edit distance as its similarity measure[4], the algorithm outputs a collection of patterns, organized in classes. Each class has a prototype, that is the most representative pattern of the class, and several occurrences, possibly inexact, of this prototype. It is also possible for the user to provide the algorithm with some constraints, such as the maximal and minimal length of patterns, the similarity threshold, the maximum difference between two candidates to be compared, etc.

SimilaritySegmenter is an evolutionary algorithm that searches for similarities in data represented as a graph. The algorithm maintains a population of *Similarity Statements*—a guess that two subgraphs (substrings) of the same size are similar—initialized randomly. By doing crossover, mutation and selection, dynamically changing the size and position of the statements, the algorithhm can evolve the population to point at increasingly more similar strings. After a number of generations the evolutionary algorithm terminates, and the most fit statement is evaluated against a threshold, determining if the substrings are similar enough to be considered a pattern. In this case a deterministic search for more occurrences is performed, before a new evolutionary search for new patterns begins. The fitness evaluation (similarity measure) and threshold can be used to specify what types of similarities we want to search for and how much dissimilarity we will allow.

[4]Dynamic programming is used to compute it efficiently.

4. Experiment

In a previous work [Dahia et al., 2004], we used a catalogue containing 21 rhythmic patterns of Bossa Nova guitar music, acquired manually from João Gilberto's performances, to automatically generate the guitar accompaniment of some songs. These songs were, thereafter, judged by some specialists as being stylistically correct. Figure 3 depicts some of these patterns.

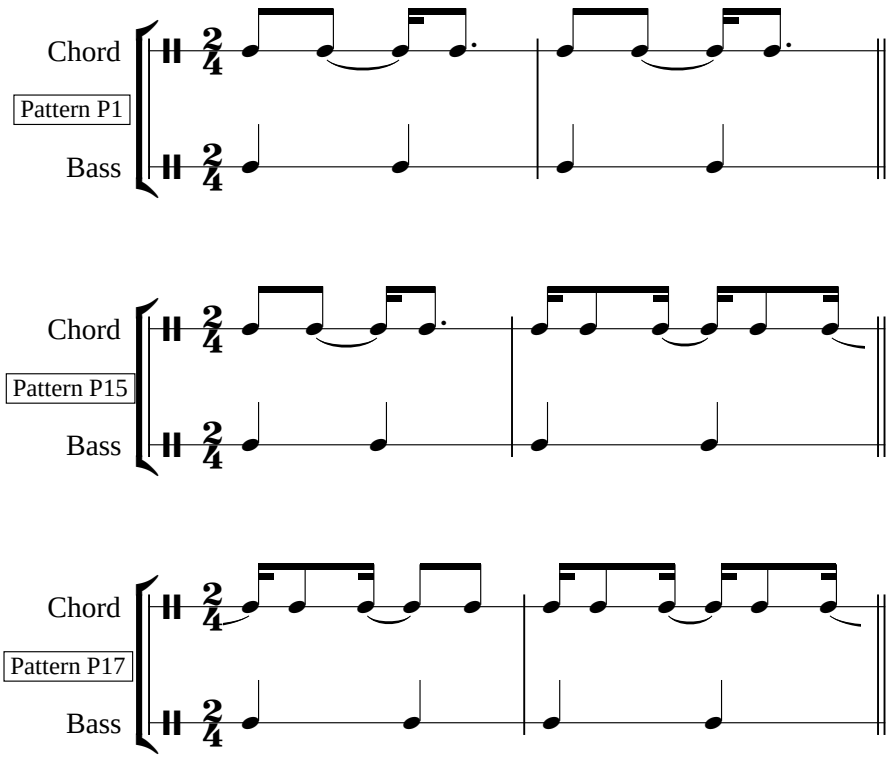

Figure 3: Examples of rhythmic patterns from the catalogue

As one can imagine, the manual acquisition of such a catalogue is tiresome. It can even be said that, in practice, it can not be done if you want to have the catalogues of many players[5]. So, the acquisition of catalogues of this kind must be done automatically (or semi-automatically, at least). The main objective of our experiment is to identify at which extent this Bossa Nova catalogue could be automatically found in the data we collected. This will give us hints on algorithms that could be applied to the acquisition of rhythmic catalogues for other styles, as Samba, for instance.

We rewrote the catalogue in the form of the simplified string previously described and used the Boyer–Moore algorithm to test occurrences of each pattern in the data set. From the 21 patterns in the catalogue, the algorithm could only find one pattern (P1) in the whole data set, appearing only in three of the songs. Table 1 summarizes the results. Each entry in the table represents the number of occurrences pf P1.

The main question now is the following: is there any flaw in the methodology we used? Since João Gilberto is the inventor of the Bossa Nova guitar style, we expected to find a more representative number of the patterns in the data set. Besides, by hearing the songs it is possible to notice that more patterns from the catalogue are used. The problem confirms that an exact matcher like the Boyer–Moore is not able to cope with any sort of deviations that are usually done by the performers (as in any expressive performance, in

[5]See, for instance, the work of Owens [Owens, 1974]. It took him 16 years to build a similar catalogue for Charlie Parker *only*.

Song	Player1	Player2
Desafinado	0	1
Garota de Ipanema	0	6
Insensatez	4	29

Table 1: Number of occurrences of pattern P1

fact). The most usual of these deviations are small anticipations and delays of events. As an example, part of a pattern that is transcribed as `A---P---B-p-+---` can be played slightly different from notated (and it usually is!) such as in `A---P---Bp--+---` (an anticipation of the last p) or as in `A---P---B--p+---` (a delay of the same p event). Another possible modification is playing the same pattern as `A---P---B-p-+-s-`, i.e., adding a single note at the end. Figure 4 depicts these variations (the "s" above the last 16th represents a single note).

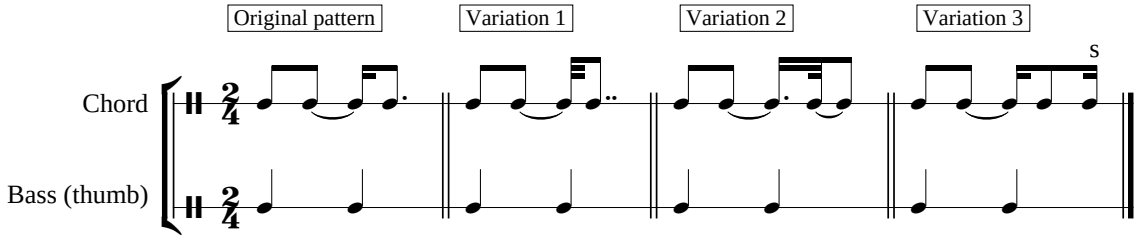

Figure 4: Some variations of part of a pattern

The main problem here is that an exact matcher, such as the Boyer–Moore algorithm, isn't suitable for the task, due to these kind of variations. In order to discover patterns, we must use algorithms that can perform the so called *inexact string matching*. For that, we used FlExPat and SimilaritySegmenter.

The results were in general good. If we take into account the small modifications done by the performers, this time more patterns from the catalogue were found in the data set[6].

The configuration we used for FlExPat was the following:

- $m_{min} = 17$ (minimal pattern length);
- $m_{max} = 34$ (maximal pattern length);
- similarity threshold: 0.75 (normalized values);

The lengths m_{min} and m_{max} correspond to patterns varying from one to two measures. It does not mean, however, that the patterns necessarily start at the beginning of the measure. There is no way to specify such a constraint in FlExPat. On average, each song has 48 classes of patterns (smallest number of classes was 40 and greatest number was 78), which is an acceptable number.

In song Garota de Ipanema, FlExPat identified the following pattern for Player2: `|A---P---B-p-+---|P---P---Bp--+l`. The corresponding pattern in the catalogue is `|A---P---B-p-+---|A---P---B-p-+-l-`. Looking closer at the modifications done by the performer, it is possible to notice that he usually substitutes the `A` by `P` in the second measure of the pattern, and that he anticipates both final chords in this same measure. Figure 5 depicts these variations.

[6]Note that now, with the inexact matching, there is a difference in our methodology: while with the Boyer–Moore algorithm we tested the existence of the patterns of the catalogue directly in the data set, now we must first induce patterns and only then look and see if the induced patterns have some resemblance with the patterns in the catalogue.

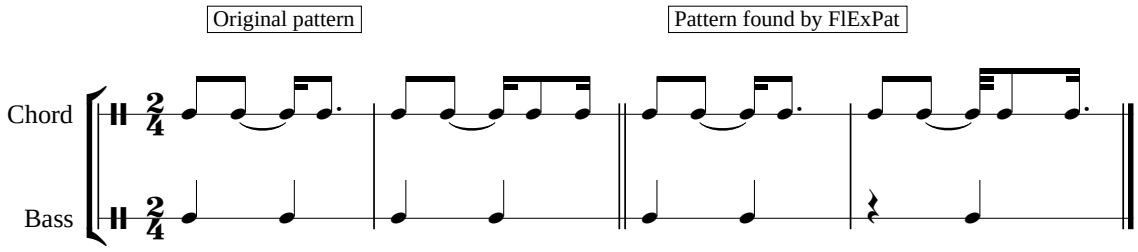

Figure 5: Pattern found by FlExPat

In song Wave, as played by Player1, FlExPat identified as the prototype of class 18 the following pattern: `|A---P---B-p-+l--|B---P---Bp--+p-`. The corresponding pattern in the catalogue is `|A---P---B-p-+-l-|B---P---B-p-+-l-`. As in the previous case, the modifications done by the performer are mainly anticipations.

Table 2 summarizes the results we obtained for FlExPat.

Pattern	Player1	Player2	Pattern	Player1	Player2
P1	*yes*	*yes*	P12	no	no
P2	*yes*	no	P13	no	no
P3	no	no	P14	no	no
P4	*yes*	no	P15	*yes*	no
P5	no	no	P16	no	no
P6	no	no	P17	no	no
P7	no	no	P18	no	no
P8	no	no	P19	no	no
P9	*yes*	*yes*	P20	no	no
P10	*yes*	*yes*	P21	no	no
P11	no	no	/	/	/

Table 2: Occurrences of patterns in data set (FlExPat)

Although FlExPat found several interesting patterns in the data set, there were some problems with the results we obtained. The main problem we found was that the extracted patterns, many times, start from the middle of a measure, such as in `--B---B---+---|P---B---B---+---|P` (FlExPat also found the expected pattern, `|P---B---B---+---|P---B---B---+--`). This brings some problems to the evaluation of the results: due to the great number of patterns with this structural malformation, it becomes difficult to validate the patterns. This problem is even bigger when most of a class is formed by such patterns (it happened many times, unfortunately).

For the SimilaritySegmenter, we allowed a maximal number of 2 mismatches per measure. We also specified some structural constraints, namely that each pattern should start at the beginning of the measure (first character should be a |) or that it should start at the first l before a measure bar. That is, patterns should have one of the following two structures:

- $| < pattern >$ or
- $l * | < pattern >$, where $*$ is a sequence, possibly empty, of only minuses ($-$).

Table 3 summarizes the results the obtained for SimilaritySegmenter.

Due to the possibility of specifying structural constraints, the results were much easier to evaluate (as compared to FlExPat). One surprising result was that the algorithm

Pattern	Player1	Player2	Pattern	Player1	Player2
P1	*yes*	*yes*	P12	no	no
P2	no	no	P13	no	no
P3	no	no	P14	*yes*	no
P4	no	no	P15	no	*yes*
P5	no	no	P16	no	no
P6	no	no	P17	no	no
P7	no	no	P18	no	no
P8	no	no	P19	no	no
P9	*yes*	*yes*	P20	no	no
P10	no	*yes*	P21	no	no
P11	no	no	/	/	/

Table 3: Occurrences of patterns in data set (SimilaritySegmenter)

was able to find rather long patterns. In song Insensatez performed by Player2, for instance, the algorithm found three occurrences of the following pattern (136 characters long, i.e., 8 measures):

```
|A---P---B-p-+---|A---P---B-p-+---|A---P---B-p-+---
|A---P---B-p-+---|A---P---B-p-+---|A---P---B-p-+---
|A---P---B-p-+---|A---P---B-p-+---
```

Although it is formed by the same cell (`|A---P---B-p-+---`), it is very significant that the algorithm could find such a long pattern. It may have structural implications: Player2 may have used this sequence during some specific part of the song. Looking closer at the data, we notice that, in this specific case, Player2 used this pattern as the accompaniment of most part of the theme. Other example of a long pattern is the following one:

```
|B---P---B-p-+-s-|A---P---B-s-+-l-|B-p-+-p-B-p-+-l-
|B-p-+-p-B---P---
```

This pattern is 68 characters long, occurred twice, and was also played by Player2, this time in song Tarde em Itapoã,

The algorithm also found some garbage, such as, `|B-` or `|A---+-`, but the main question that arose during the experiment was due to the algorithm's inherent non-determinism: if a pattern in the catalogue wasn't found is it because it was really not played or is it because the algorithm didn't run long enough? This seems to be case of P10 for Player1, which was found several times by FlExPat.

5. Final Remarks and Future Work

This paper described an experiment that dealt with extraction of rhythmic patterns from Bossa Nova songs. Sixteen beat tracked MIDI files, representing the recording of several songs by two different players, were represented as a string and thereafter processed by three different string matching algorithms. The objective was to identify in the data set patterns from a previously acquired catalogue.

First results showed that, as expected, an exact matcher was not able to cope with the variations of the patterns done by the players, as well as, with some single notes they play between chords. With the use of inexact matchers the results were much more

interesting: several patterns from the catalogue could be found in the data set. We also obtained some surprising results, such as the long patterns found by SimilaritySegmenter.

There are, however, several points for improvement. The first one, and may be most important one, is the representation. The representation we used is, indeed, simple. Attributes like tempo, structural or harmonic information are not represented. The more attentive reader may have noticed that the actual duration of the events is not represented. We just used the onset information, which turned out to work for this experiment. To further investigate the particularities of the patterns, however, we surely need to represent appropriately the duration of each event.

The evaluation of the algorithms' results was done in an *ad hoc* manner: results were compared, one by one, to the patterns in the catalogue. This procedure takes too much time, is error prone, and, therefore, must be improved. We plan to implement a tool to help us with this task.

FlExPat's problem (structural malformation) must be examined more carefully. Instead of a problem, it can mean another thing. Several patterns of the catalogue have a common substructure (i.e., subparts of these patterns are equal). It may be the case that the second measure of patterns whose second measure are equal, are frequently concatenated with patterns whose first measure are similar. So, it may be the case that these concatenations are so typical that they are "promoted" to patterns by the algorithm. This must be further investigated.

Another point for further investigation is the possibility of combining the strengths of the algorithms into a single procedure. Whereas FlExPat performs a complete search for patterns, SimilaritySegmenter is able to satisfy some quite specific structural constraints about them. It may be interesting to try to combine such particularities into a single method.

Finally, we already have collected some other songs, namely some Sambas recorded by three different players. We plan, as soon as we have prepared the data, to perform a similar experiment on this data set.

6. Acknowledgments

The first author is supported by a research grant from CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico). The research by the Austrian Research Institute for Artificial Intelligence was supported by the Austrian *Fonds zur Förderung der Wissenschaftlichen Forschung (FWF)* under grant no. Y99-START. The Austrian Research Institute for Artificial Intelligence acknowledges financial support by the Austrian Federal Ministries for Education, Science and Culture (BMBWK) and for Transport, Innovation and Technology (BMVIT).

References

Boyer, R. S. and Moore, J. S. (1977). A fast string searching algorithm. *Commun. ACM*, 20(10):762–772.

Chediak, A., editor (1990). *Songbook: Bossa Nova*, volume 1–5. Lumiar Editora, Rio de Janeiro.

Dahia, M., Santana, H., Trajano, E., Sandroni, C., Ramalho, G., and Cabral, G. (2004). Using patterns to generate rhythmic accompaniment for guitar. In *Proc. of Sound and Music Computing (SMC'04)*, pages 111–115.

Desain, P., Honing, H., and Timmers, R. (2001). Music performance panel: NICI/MMM position statement. In *MOSART Workshop on Current Research Directions in Computer Music*, Barcelona, Spain.

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

Garcia, W. (1999). *Bim Bom: A Contradição sem Conflitos de João Gilberto*. Editora Guerra e Paz.

Gusfield, D. (1997). *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press.

Madsen, S. T. and Widmer, G. (2005). Evolutionary search for musical parallelism. In *Proc. of the EvoWorkshops2005*, Lecture Notes in Computer Science 3449, pages 488–497. Springer Verlag.

Mongeau, M. and Sankofff, D. (1990). Comparision of musical sequences. *Computer and the Humanities*, 24:161–175.

Owens, T. (1974). *Charlie Parker: Techniques of Improvisation, 2 vols.* PhD thesis, Univeristy of Los Angeles California.

Rolland, P.-Y. (2001). FlExPat: Flexible extraction of sequential patterns. In Cercone, N., Lin, T. Y., and Wu, X., editors, *Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM'01)*, pages 481–488. IEEE Computer Society.

Sandroni, C. (2001). *Feitiço Decente: Transformações do Samba no Rio de Janeiro (1917–1933)*. Jorge Zahar Editor, Rio de Janeiro.

Sundberg, J., Friberg, A., and Frydén, L. (1991). Common secrets of musicians and listeners: an analysis-by-synthesis study of musical performance. In Howell, P., West, R., and Cross, I., editors, *Representing Musical Structure*, pages 161–197. Academic Press.

Trajano, E., Dahia, M., Santana, H., and Ramalho, G. (2004). Automatic discovery of right hand fingering in guitar accompaniment. In *Proceedings of the International Computer Music Conference (ICMC'04)*, pages 722–725.

Widmer, G. (1998). Applications of machine learning to music research: Empirical investigations into the phenomenon of musical expression. In Michalski, R. S., Bratko, I., and Kubat, M., editors, *Machine Learning, Data Mining and Knowledge Discovery: Methods and Applications*. Wiley & Sons, Chichester (UK).

Widmer, G. (2002). Machine discoveries: A few simple, robust local expression principles. *Journal of New Music Research*, 31(1):27–50.

Widmer, G., Dixon, S., Goebl, W., Pampalk, E., and Tobudic, A. (2003). In search of the Horowitz factor. *AI Magazine*, 24(3):111–130.