AN EFFICIENT ALGORITHM FOR MOTIVIC PATTERN EXTRACTION BASED ON A COGNITIVE MODELING

Olivier Lartillot
University of Jyväskylä
Department of Music
PL 35(A)
40014, Jyväskylä, Finland

ABSTRACT

This paper describes a computational model for discovering repeated patterns in symbolic representations of music. Patterns are discovered through an incremental adaptive identification along a multi-dimensional parametrical space. The difficulties of pattern discovery mainly come from combinatorial redundancies, that the human cognitive system is able to control efficiently. Our research attempts to reconstruct these principles. A specificity relation is defined between pattern descriptions, unifying suffix relation – between patterns – and inclusion relation – between multi-parametric pattern descriptions – and enabling a filtering of redundant descriptions. Successive repetitions of patterns imply another kind of combinatorial proliferation, which can be managed with cyclic patterns. By reconstructing these redundancy control mechanisms, musicology-relevant analyses can be automated. Current researches include the enrichment of the model, which was primarily dedicated to the analysis of monodies, with polyphony management mechanisms. The system may be used either for musicology researches, as an improvement of traditional analysis techniques, or for industrial application to automated analysis of musical databases.

1. INTRODUCTION

This paper is focused on automated description of symbolic music, and presents an efficient algorithm for discovering repeated patterns. Repeated patterns are structures easily perceived by listeners, experienced or not, and represent therefore one of the most salient characteristics of musical works, sometimes called *parallelism* [15]. Musicologists have tried to carry out very detailed motivic analysis [20]. But for all the tremendous amount of energy dedicated, complete and exhaustive analyses of complex pieces could hardly be achieved. Besides, the objective relevance of these analyses is not insured, since most of the analytic processes are carried out intuitively [6]. Attempts to formalize motivic analysis have been undertaken [22, 19]. It seems however that a precise control of the huge complexity of musical structures and discovery processes needs the help of computational models.

In this paper, patterns will be searched from symbolic representations of music rather than from signals or sound recordings. A pattern extraction task on the symbolic level, although theoretically simpler, remains extremely difficult to carry out, and its automation has not been achieved up to now. Computer researches on this subject [3, 4, 7, 8, 16, 21, 23] hardly offer results close to the expectations of listeners or musicologists. We have tried to find the underlying reasons of such difference, and discovered the existence of a problem of combinatorial redundancy, that come from the definition of the pattern discovery task. It is supposed that this redundancy is implicitly controlled by the cognitive system founding listening and analysis processes. We thus propose a cognitive modeling of such principles of redundancy control that actually offers results of significant quality.

2. AN INCREMENTAL MULTIDIMENSIONAL MOTIVIC IDENTIFICATION

2.1. The Musical Dimensions

Music is expressed along multiple parametric dimensions (Figure 1). We consider three different melodic dimensions: diatonic (*diat*, defined by the difference in scale degrees between successive note pitches), chromatic (*chro*, difference in semi-tones), and another based on gross contour (*cont*, sense of variation between successive note pitches). The rhythmic dimension, *ryth*, features note durations expressed relatively to metric unit.

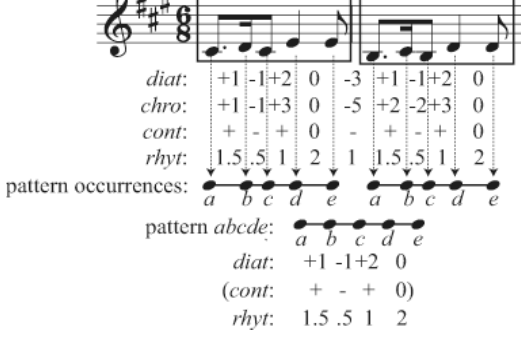

Figure 1. Melodico-rhythmic description. For a 6/8 metric, the metric unit is the 8th note. A sequential repetition of descriptions leads to the discovery of pattern *abcde*. Contour description, here redundant with diatonic description, may be removed.

These parameters describe each *interval* between successive notes, where interval is considered in a general sense, including the rhythmic dimension. When a succession of descriptions is repeated, each repetition is called an *occurrence* of the *pattern* described by this succession of descriptions. The pattern can be modeled as a chain of states, each successive state representing the successive note of each occurrence, and each successive transition representing the descriptions of the successive intervals (Figure 1). The set of all motives can be displayed as a prefix tree, since two motives with same prefix can be considered as two different continuations of this prefix.

2.2. Identification of similarities.

Patterns are generally not exactly repeated but transformed in multiple ways. These patterns should therefore be detected through an identification of their different occurrences beyond their apparent diversities. Current approaches follow two different strategies. One is based on *numerical similarity*, and tolerates a certain amount of dissimilarity between compared parameters [7, 21] (Figure 2, top). Reference cognitive studies [12], however, question this first strategy, asserting that similarity does not come from a minimization of numerical distance. They propose instead an alternative strategy based on *exact identification along multiple musical dimensions* of various specificity levels (Figure 2, bottom).

Several approaches to pattern discovery follow this second strategy of identification along different musical dimensions [2, 4, 16] and search for repetitions along each different dimension and product of dimensions. Nonetheless there exist patterns that are progressively constructed along variable successive musical dimensions. Such *heterogeneous* patterns cannot be identified by traditional approaches. For instance, each line of the score in Figure 3 contains a repetition of a same pattern: in the first half, both melodic and rhythmic dimensions are repeated whereas, in the second half, only the rhythmic dimension is repeated. The model presented in this paper is able to discover such heterogeneous patterns.

2.3. Incremental Pattern Construction

The basic principle of our algorithm, aimed at an exhaustive discovery of repeated patterns, is based on *associative memory*, i.e. the capacity of relating items that feature similar properties. The associative memory is modeled through hash tables associated with the different musical parameters (i.e. melodic and rhythmic dimensions). A first set of hash tables store the intervals of the piece with respect to their values along each different musical dimension. For instance, two tables (Figure 4, line *a*) store the intervals of the score according to their diatonic and rhythmic values. We can see in the melodic table that the first interval of each bar share same diatonic value *diat* = +1, and, in the rhythmic table, that they also share same rhythmic value *rhyt* = 1.5.

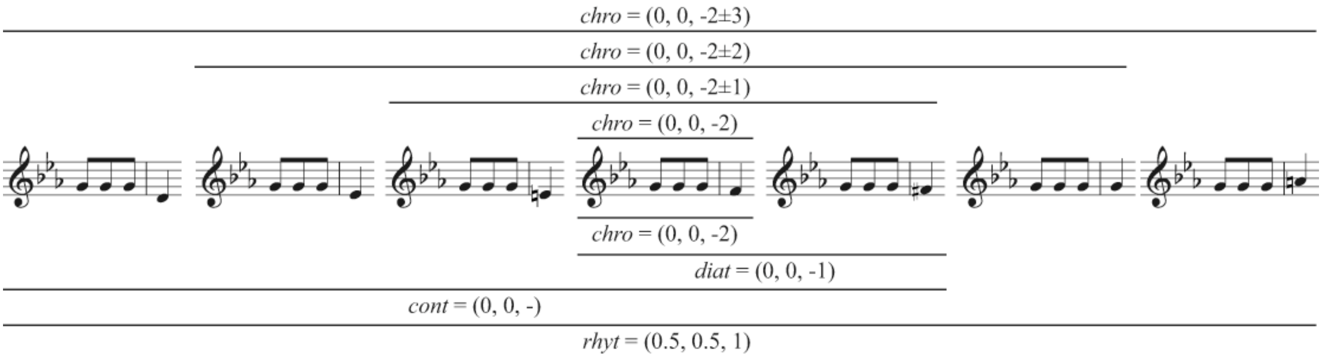

Figure 2. Two pattern classification methods: one (top) based on a numerical similarity distance, for instance, along the chromatic dimension *chro*, the other (bottom) based on exact identification along different musical dimensions.

Figure 3. Repetition of a heterogeneous pattern (squared): the first half is melodico-rhythmic (melodic and rhythmic descriptions are repeated) and the second half is simply rhythmic (only the rhythmic descriptions are repeated).

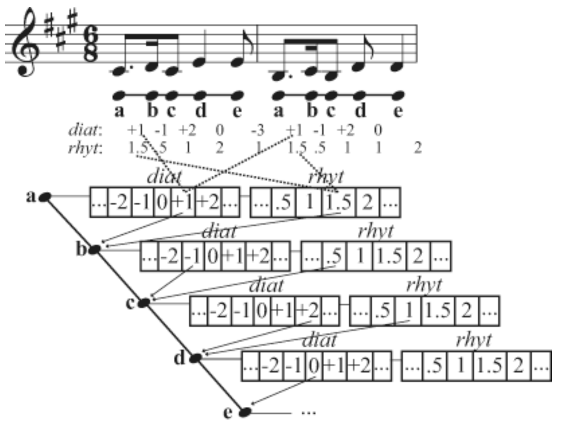

Figure 4. Progressive construction of pattern *abcde* and of its two occurrences, by storing the intervals in hash tables associated with each successive state of the pattern.

Intervals sharing a same value form occurrences of a pattern that represents this particular value. The pattern is represented as a child (for instance, *b*) of the root of the pattern tree (*a*). When a new pattern is created, new tables (at the right of node *b*) store all the possible intervals that *immediately follow* the occurrences of this pattern (here, *b*). When identities are detected in these new tables, a new pattern is created as an extension of the previous one (*c*, as an extension of *b*), and is represented as a child in the pattern tree, and so on. This algorithm enables a progressive discovery of the successive extensions of each pattern, either homogeneous or heterogeneous: the selection of musical dimensions defining each successive extension of a pattern may vary. For instance, in Figure 4, the last extension of pattern *abcde* is simply diatonic since the rhythm of the last interval in each occurrence is different. Additional constraints have been integrated in order to insure a minimal continuity along these variable successive musical dimensions.

3. COMBINATORIAL REDUNDANCY FILTERING

A running of the basic algorithm on musical examples, even simple, produces a huge number of patterns that do not correspond, for most of them, to actual perceived structures, and implies a combinatorial explosion. This is due first to the fact that our approach accepts a large number of possible configurations, such as heterogeneous patterns. But in fact all computational approaches face up to same combinatorial problems. The complexity is commonly reduced through a filtering of the results following global criteria, such as a selection of longest or most frequent patterns [2, 4, 16]. However, this filtering does not improve the perceptive relevance of the results, and may arbitrarily discard interesting patterns. Our study shows the underlying reasons of this phenomenon of combinatorial explosion and of perceptive irrelevance of the results. In fact, the repetition-based paradigm implicitly leads to this kind of redundancy, that the cognitive system of the listener is able to control. A computational modeling of the listening processes should therefore reconstruct these strategies. After thorough studies, we have been able to decompose this problem into several different distinct general sub-problems. They will be presented in the remainder of this section, with proposed mechanisms that are able to resolve each of them as simply as possible.

3.1. Maximally specific descriptions.

3.1.1. Suffix filtering.

Suffixes of patterns should not generally be represented explicitly as autonomous patterns, as this would introduce a significant amount of redundancy and lead to exponential complexity with respect to the size of pattern trees. However, a suffix should be explicitly represented whenever one of its occurrences is not a suffix of any occurrence of the whole pattern. For instance, pattern *aij*, in Figure 5, is simply a suffix of pattern *abcde*, whereas it includes, in Figure 6, occurrences that are not suffixes of occurrences of *abcde*. In this second case, pattern *aij* can be explicitly represented.

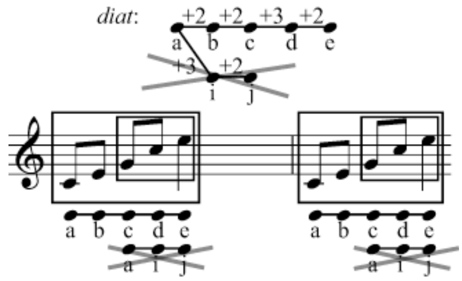

Figure 5. Pattern *aij* exists simply as a suffix of *abcde*, since both pattern classes are equal. Hence pattern *aij* should not be explicitly represented.

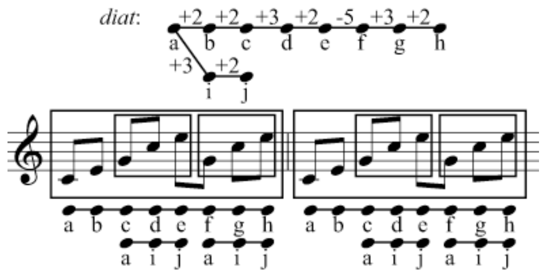

Figure 6. Pattern *aij* is not a simple suffix, of either *abcde* or *abcdefgh*, since their classes are strictly included in the class of pattern *aij*. Pattern *aij* should therefore be explicitly represented.

This principle can be formalized more precisely by defining the *class* of a pattern *m* as the set of its occurrences, and by introducing the notion of *inclusion relation* between pattern classes. The class of pattern *abcde* and the one of its suffix *aij* are considered, in the first case, as *equal* since each occurrence of pattern *aij* is a suffix of an occurrence of pattern *abcde*. In the second case, on the contrary, there exist occurrences of *aij* that are not suffixes of occurrences of pattern *abcde*. The class of pattern *abcde* is then considered as *strictly included* in the class of pattern *aij*. Therefore, for a pattern to be considered as an autonomous, its class should not be equal to the class of a pattern of which it is a suffix.

3.1.2. Specific description of patterns.

This heuristics can be applied to a more general relation of *specificity* between descriptions. Pattern *abcde* (Figure 7) features melodic and rhythmical descriptions, whereas pattern *afghi* only features its rhythmic part. Hence pattern *abcde* can been considered as more specific than pattern *afghi*, since its description contains more information. The less specific pattern *afghi* should not generally be explicitly represented if its class is equal to the class of the more specific pattern *abcde*. If on the contrary there exists an occurrence of pattern *afghi* that is not an occurrence of pattern *abcde* (Figure 7, right), then the pattern *afghi* can be explicitly represented. If we include the suffix relation into the specific relation – by stating that a suffix of a motif is less specific – then this new heuristics applies also to the filtering of suffixes described in the previous paragraph.

3.1.3. Specific description of pattern occurrences.

We have prolonged this attempt to optimize pattern descriptions by adding a principle of maximally specific descriptions of pattern occurrences: when a pattern occurrence is discovered (for instance, pattern *abcde*), all the occurrences of less specific patterns (for instance, pattern *afghi*) are not superposed on it, since they do not bring additional information, and can be directly deduced from the most specific pattern occurrence (*abcde*) and from the specificity relation (between *abcde* and *afghi*).

The less specific description should be taken into account implicitly though, because their extensions may sometimes lead to specific descriptions. For instance (Figure 8), groups 1 and 3 are occurrences of pattern *afghi*, and groups 3 and 4 are occurrences of pattern *abcde*. Since pattern *abcde* is more specific, the less specific pattern *afghi* does not need to be associated with group 4. However in order to detect groups 2 and 5 as occurrences of pattern *afghjkl*, it is necessary to implicitly consider group 4 as an occurrence of pattern *afghi*. Hence, even if pattern *afghi*, since less specific than *abcde*, was not explicitly associated with group 4, it had to be considered implicitly in order to construct pattern *afghjkl*. Implicit information is reconstituted through a traversal of the pattern network along specificity relations.

3.1.4. Generalization of patterns.

New patterns can be discovered as simple generalization of already known patterns. In bar 7 of Figure 9, the two first notes form an occurrence of pattern *ah*. The third note cannot however fulfill the known extension of pattern *ah* into pattern *ahi*, because the melodic description *diat* = 0 does not match here. However, as the rhythmic description *rhyt = 2* matches, a new extension *ahj* is discovered as a generalization of pattern *ahi*.

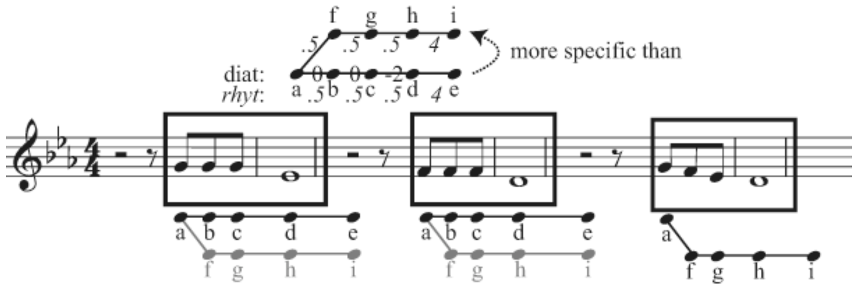

Figure 7. After analyzing the four first bars, both patterns *abcde* and *afghi* having same classes, only the more specific pattern *abcde* should be explicitly represented. The less specific pattern *afghi* will be represented once the third occurrence is discovered, as it is not an occurrence of the more specific pattern *abcde*.

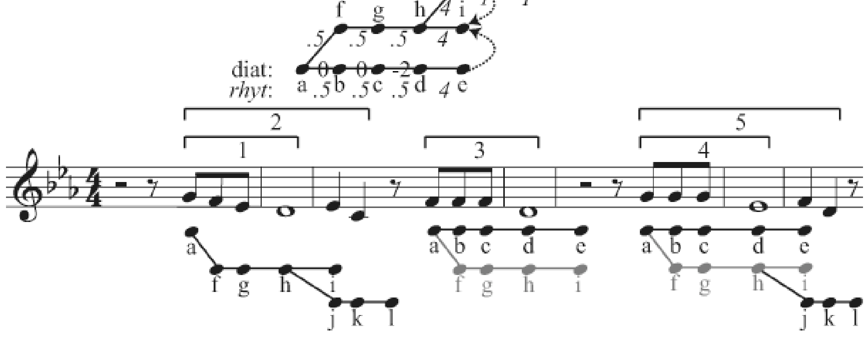

Figure 8. Group 4 can be simply considered as occurrence of pattern *abcde*. However, in order to detect group 5 as occurrence of pattern *afghjkl*, it is necessary to implicitly infer group 4 as occurrence of pattern *afghi* too.

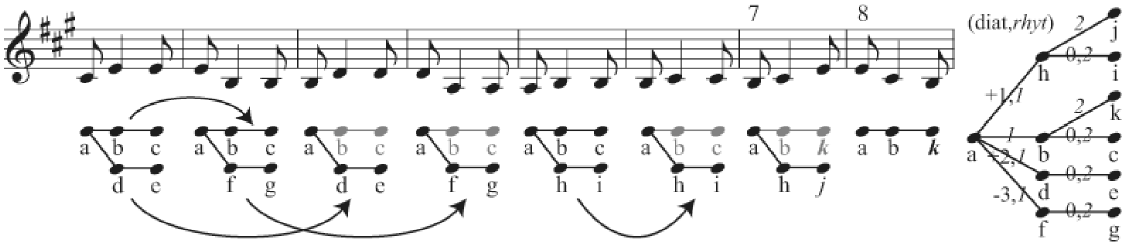

Figure 9. Progressive discovery of the pattern repetitions on the score (represented in left) and the resulting pattern tree (right). Pattern descriptions in gray are less specific than simultaneous descriptions in black, and should not therefore be represented explicitly. However, the generalization of pattern *ahi* (bar 6) to pattern *ahj* (bar 7) leads to the implicit generalization of pattern *abc* to pattern *abk*, than can therefore be immediately identified in bar 8.

The less specific patterns, although usually not explicitly represented in the analysis, should be updated if necessary. In particular, when a generalization of a pattern is discovered, the generalization of all its more general patterns should also be considered. For instance, as *ahi* has been generalized into *ahj*, it should also be inferred that *abc* is generalized into *abk* in the same way. Hence the analysis of the next bar (8) consists simply in recognizing this general pattern *abk* already known.

The implementation of these principles is carried out through detailed algorithms that traverse the pattern network in a decreasing order of specificity. This mechanism insures an optimal pattern description, through lossless compression of the description size following a redundancy filtering. The optimization enables to offer clear results and to limit the combinatorial complexity of the process.

3.2. A cycle-based modeling of repeated patterns.

Combinatorial explosion can also be caused by successive repetitions of a same pattern (for instance the rhythmic pattern in figure 10). As each occurrence is followed by the beginning of a new occurrence, each pattern can be extended (leading to pattern *abcd*) by a new interval whose description is identical to the description of the first interval of the same pattern (i.e., pattern *ab*). This extension can be prolonged recursively (into *abcde*, *abcdef*, etc.), leading to a combinatorial explosion of patterns that are not perceived due to their complex intertwining [2].

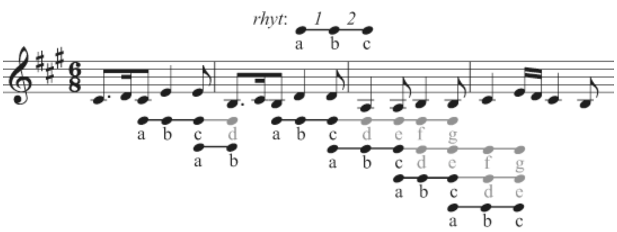

Figure 10. Multiple successive repetitions of pattern *abc* logically leads to its extension into pattern *abcd*, *abcde*, etc. that form a complex intertwining of non-perceived structures.

3.2.1. The perception of cycles.

Our representation (Figure 10) shows that the last state of each occurrence of pattern *abc* is superposed to the first state of the following occurrence. Listeners seem to tend to fusion these two states, and to perceive a loop from the last state (*c*) to the first state (*a*) (Figure 11). The initial acyclic pattern *abc* leads therefore to a cyclic pattern that oscillates between two states *a'* and *b'*, corresponding to rhythmic values 1 and 2. Indeed, when listening to the remainder of the musical sequence, we actually perceive this progressive oscillation between these two states *a'* and *b'*, that correspond to the two rhythmic values. Hence this cycle-based modeling seems to explain a common listening strategy, and resolve the problem of combinatorial redundancy.

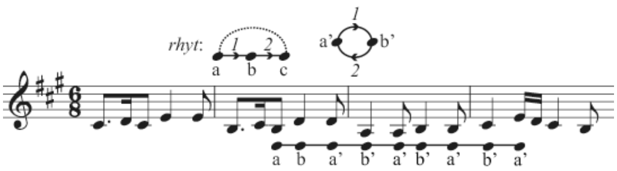

Figure 11. The listening of successive repetitions of pattern *abc* leads to the induction of its cyclicity, hence to an oscillation between states *a'* and *b'*.

This phenomenon of successive repetition, although very frequent in musical expression, has been scarcely studied [2, 5]. The combinatorial explosion generated by the phenomenon is reduced by selecting, once the analyses is done, the patterns featuring minimal temporal overlapping between occurrences [2]. The trouble is, as the selection is inferred globally, lots of interesting and relevant patterns are discarded. Our considering of local configurations enables a more precise filtering. Besides combinatorial redundancy remained problematic since the filtering was done after the actual analysis phase.

3.2.2. General and specific cycles.

The specificity relation needs to be applied to cyclic patterns too: a cyclic pattern *C* would be considered as more specific than another cyclic pattern *D* when the sequence of description of pattern *D* is included in the sequence of description of pattern *C*. Here too, this concept of specificity plays a pivotal role in music perception and enables a sound algorithmic processing of music. In Figure 12, the seven first notes of the cycle form a simple progression of cycle *a'b'* simply composed of a succession of two rhythmic values of 16^{th} and 8^{th} (*rhyt = 1* and *2*), the second interval also associated with a unison interval (*diat* = 0). Then a more specific cycle *c'd'* includes an ascending interval (*diat* = +1), and is generalized after four notes to cycle *e'f'* that does not feature the unison interval any more. Moreover, following the rule of generalization of generalized patterns explained in paragraph 3.1.4, the more general cycle *a'b'* too needs to be generalized to a cycle *g'h'* discarding the unison interval. These different cycles are actually perceived by the listener. Moreover, the integration of this phenomenon into the model helps insuring the relevance of the results and avoiding numerous unwanted combinatorial redundancies.

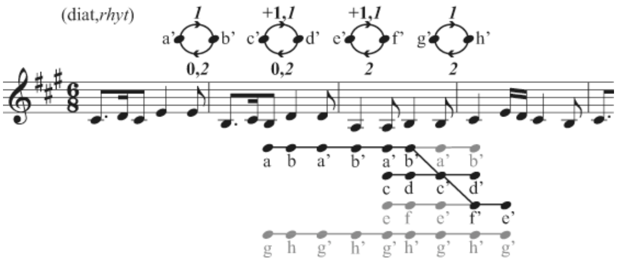

Figure 12. Actually, complex cyclic patterns are perceived: a first oscillation between two rhythmic values with a unison interval (*a'b'*), then, more specifically, the repetition of an ascending interval (*c'd'*). The cycle is then generalized due to the absence of the unison (*e'f'*) and cycle *a'b'* is also generalized, for the same reason, to *g'h'*.

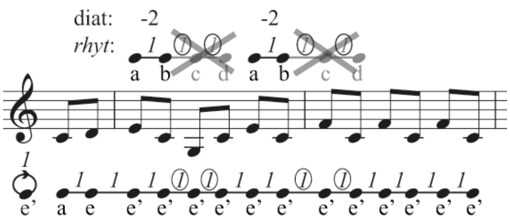

Figure 13. Pattern *ab* is a specific *figure* above a *background* generated by the cyclic pattern *e'*. Following the *Gestalt* rule of figure against ground, the figure cannot be extended by a description that is identical to the background.

3.2.3. *Figure/ground* Gestalt *rule.*

Another phenomenon of combinatorial redundancy appears when a specific pattern is superposed several times to a less specific cyclic pattern (Figure 13). The more specific pattern could logically be extended by the successive extensions of the cyclic pattern (leading to patterns *abc*, *abcd*, etc.). This phenomenon, which may appear frequently in a musical piece, would lead to another combinatorial proliferation of redundant structures if not correctly controlled by relevant mechanisms. In fact, these extensions are not actually perceived by the listener because of a general *Gestalt* rule of figure against ground: the more specific pattern (*ab*) constitutes a *figure*, above a *background* generated by the cyclic pattern. Following this rule, the figure cannot be extended (into *abc*) by a description that can be simply identified to a background extension. The rule allows a selective control of redundant extensions.

4. RESULTS SENSIBLY CLOSE TO PERCEIVED STRUCTURES

This model was first developed as a library of *OpenMusic* [1]. A new version will be included in the next version 2.0 of *MIDItoolbox* [13]. The model can analyze monodic musical pieces (i.e., constituted by a series of non-superposed notes) and highlight the discovered patterns on a score.

4.1. Some examples.

Thanks to the complex modeling of listening strategies, the automated analysis system is able to offer a clear pattern description of simple musical pieces, corresponding mostly to actually perceived structures.

4.1.1. *Mozart* Sonata in A *K 331.*

For instance, the analysis of the first theme of Mozart *Sonata in A* K 331 (Figure 14) shows the basic pattern (a), repeated and transposed, and the 4-measure long phrase (b) repeated twice. However, a slight rhythmic transformation at the end of the first occurrence does not allow a discovery of the whole phrase (b'). Is also shown the successive repetition of a simple rhythm featuring an eight-note and a fourth-note, that lead to a cyclic pattern (c). A detailed description of the cyclic pattern, discovered by the system, has been presented in Figure 12. Interestingly enough, the same algorithm is able to discover 4-measures long phrases and simple successions of ascending (e) and decreasing (d and f) conjunct intervals.

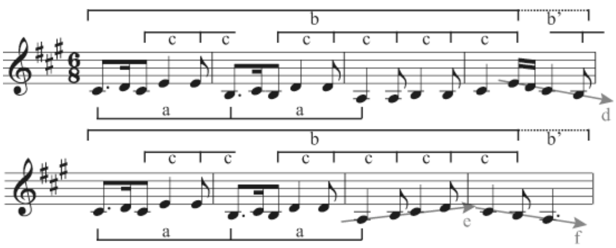

Figure 14. Automated analysis of first theme of Mozart *Sonata in A* K 331.

4.1.2. *Geisslerlied.*

Figure 15 presents the resulting analysis of a medieval song called *Geisslerlied* that the linguistic Nicolas Ruwet, in one of the first and most famous attempt to model motivic analysis [22], proposed as a first application of his method. Our model is the first computational system able to offer a relevant and compact analysis of this piece.

Other famous musical examples have been analyzed, such as the beginning of Beethoven's *Fifth Symphony*, other Mozart *Sonatas*, or Arabic modal improvisations, leading to interesting results. Neither these levels of precision or of perceptive relevance have been achieved before. Others pattern discovery systems would indeed include a numerous set of redundant patterns such as suffixes or redundant extensions. This shows the necessity of mechanisms of adaptive redundancy filtering such as those proposed in this paper. However the analyses remain significantly restricted, as numerous aspects of musical expression have not been taken into account yet.

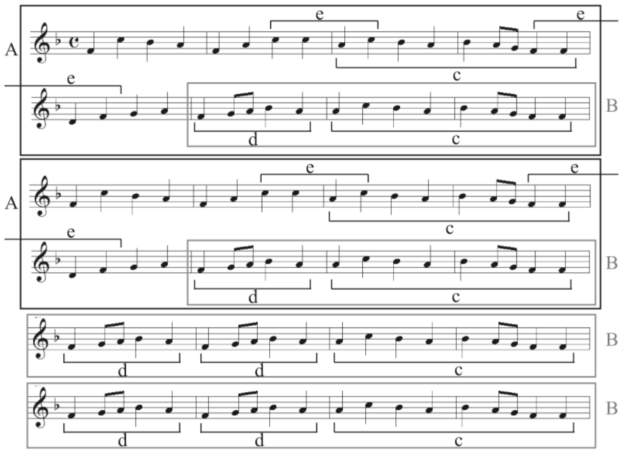

Figure 15. Analysis of a *Geisslerlied*.

4.2. Algorithm complexity.

The algorithm complexity may be expressed along two dimensions. First, through the complexity of discovered structures: proliferation of redundant patterns, for instance, would lead to combinatorial explosion, since each new structure needs proper processes assessing their interrelationships with the other structures, and inferring their possible extensions. Hence a maximally compact description insures in the same time the clarity and relevance of the results and the limitation of combinatorial explosion. Complexity may be considered also with respect to the technical implementation of the modeling. Our proposed algorithms remain yet in a draft version and are for the time being only partially optimized. Yet the modeling has been conceived with the constant objective to minimize computational costs. Hence the identification of similar descriptions, as shown in paragraph 2.3, is based on hash tables, which insure optimal time complexity.

4.3. Future works

Thanks to this perceptive mimicry, the model offers promising results. Yet bad behaviors need to be controlled, and a large scope of musical expression – such as polyphony – has not been taken into account yet.

4.3.1. *Integrating* Gestalt *segmentation mechanisms*

The structures currently found are based solely on pattern repetitions. Should be added another mechanism based on merging of notes closed in time or pitch domain, and, reversely, on segmentation between distant notes, following *Gestalt* rules of proximity and similarity [15, 3]. Although this rule plays a significant role in the perception of large-scale musical structures, there is no common agreement on its application to detailed structure, because it highly depends on the subjective choice of musical parameters used for the segmentations [9]. In our approach, we propose to investigate the competitive/collaborative interrelations between the two rules of pattern discovery and *Gestalt* segmentation. For instance, a pattern repetition may be masked due to an important temporal gap within one of the occurrences.

4.3.2. *Detecting musical transformations*

Our model is able to detect not only exact repetitions of patterns, but also partial repetitions along particular musical dimensions, leading to the discovery of interesting structures. Yet other aspects of musical transformations should be considered too, such as the local insertion or deletion of notes. Solutions have been proposed [8, 18, 21], primarily based on dynamic programming and edit distance, allowing optimal alignment between similar notes of each occurrence. We expect our methodology to offer new solutions to these problems, funded on complex modeling of detailed cognitive strategies.

4.3.3. *Polyphonic pattern discovery*

Our approach is limited to the detection of repeated *monodic* patterns (i.e. sequences of successive notes) within a monodic musical piece. Music, in general, is *polyphonic*: it can contain simultaneous notes forming chords, in particular, and simultaneous monodic lines forming different voices. Researches have been carried out in this domain [4, 11, 16, 17], mainly focused on the discovery of repeated exact patterns along different pre-specified dimensions. In our approach, we are currently developing rules of automated discovery of melodic lines inside polyphonic sets of notes (or *stream segregation*), based on cognitive heuristics. Our study will then focus on the interactions between pattern discovery and stream segregation. We will then extend the scope by considering pattern of chords, which will need a formalizing of a general concept of interval between successive chords. Could then be considered also the general problem of discovery of patterns composed of segments [5, 10, 14].

4.4. Cognitive and industrial applications.

This study shows that musical patterns results from a network of interdependent mechanisms, which need to be carefully modeled within a conceptual network. The different processes take the form of basic operators that applies to each successive phase of the construction of the structure. The stability of the whole system depends on the good definition of each elementary operator: a little default may lead to general chaotic behavior and combinatorial explosion, whose original cause may sometimes be quite hard to discover. Because of the paramount difficulty to control the whole mechanisms insuring the relevance of discovered patterns, we assume that a modeling able to offer satisfying results may present a certain analogy with the actual cognitive processes that govern the human listening activity. The resulting model will hence be offered to cognitive validations and improvements with the help of experimental psychology.

The automated discovery of repeated patterns may lead to interesting applications. A new kind of similarity distance between musical pieces may be defined, based on these pattern descriptions. A music database could then be browsed using this pattern-based similarity distance: from a given musical piece, a user may then find pieces that features similar patterns. The automated pattern description of musical database may also enable an improvement of pattern matching algorithm: when a user looks for a specific pattern in a music database, the search, that should initially be undertaken throughout the whole database, can be reduced to the set of characteristic patterns that have been discovered during the initial analysis.

These researches are carried out in the University of Jyväskylä in close collaboration between computer science, cognitive psychology, musicology and ethnomusicology, in particular within the context of a collaborative project with Stephen McAdams (CIRMMT, Montreal), Mondher Ayari and Gérard Assayag (Ircam) funded by the French Academy (CNRS).

5. REFERENCES

[1] Assayag, G. et al. "Computer Assisted Composition at Ircam: From Patchwork to Openmusic", *Computer Music Journal*, 23(3), 1999.

[2] Cambouropoulos, E. *Towards a General Computational Theory of Musical Structure*. PhD thesis, University of Edinburgh, 1998.

[3] Cambouropoulos, E. "Musical Parallelism and Melodic Segmentation: A Computational Approach", *Music Perception*, to appear.

[4] Conklin, D., and C. Anagnostopoulou. "Representation and Discovery of Multiple Viewpoint Patterns", *Proceedings of the International Computer Music Conference*, San Francisco, USA, 2001.

[5] Conklin, D., and C. Anagnostopoulou. "Segmental Pattern Discovery in Music", *INFORMS Journal of Computing*, to appear.

[6] Cook, N. *A Guide to Musical Analysis*. London, Dent, 1987.

[7] Cope, D. *Computer and Musical Style*. Oxford University Press., 1991.

[8] Dannenberg, R. and N. Hu. "Pattern Discovery Techniques for Music Audio." *Proceedings of the International Conference on Music Information Retrieval*, Paris, France, 2002.

[9] Deliège, I. "Grouping Conditions in Listening to Music: An Approach to Lerdahl and Jackendoff's Grouping Preference Rules". *Music Perception*, 4(4), 1987.

[10] Deutsch, D. and J. Feroe. "The Internal Representation of Pitch Sequences in Tonal Music". *Psychological Review*, 88(6), 1981.

[11] Dovey, M.J. : "A technique for "regular expression" style searching in polyphonic music", *Proceedings of the 2nd International Conference on Music Information Retrieval*, 2001.

[12] Dowling, W.J., and D.L. Harwood. *Music Cognition*. Academic Press, London, 1986.

[13] Eerola, T., and P. Toiviainen. "MIR In Matlab: The MIDI Toolbox." *Proceedings of the International Conference on Music Information Retrieval*, Barcelona, Spain, 2004.

[14] Lartillot, O., and E. Saint-James. "Automating Motivic Analysis through the Application of Perceptual Rules", *Music Query: Methods, Strategies, and User Studies* (*Computing in Musicology* 13). MIT Press, to appear.

[15] Lerdahl, F., and R. Jackendoff. *A Generative Theory of Tonal Music*. MIT Press, 1983.

[16] Meredith, D., K. Lemström and G. Wiggins. "Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music" *Journal of New Music Research*, 31(4), 2002.

[17] Meudic, B. and E. Saint-James. "Automatic Extraction of Approximate Repetitions in Polyphonic MIDI Files Based on Perceptive Criteria". in U.K. Will (ed.), *Computer Music Modelling and Retrieval,* Springer-Verlag, 2004.

[18] Mongeau, M. and D. Sankoff. "Comparison of Musical Sequences" *Computer and the Humanities*, 24, 1990.

[19] Nattiez, J.-J.. *Music and Discourse: Towards a Semiology of Music*. Princeton, Princeton University Press, 1990.

[20] Reti, R. *The Thematic Process in Music*. New York, Macmillan, 1951.

[21] Rolland, P.-Y. "Discovering Patterns in Musical Sequences", *Journal of New Music Research*, 28(4), 1999.

[22] Ruwet, N. "Methods of Analysis in Musicology", Music Analysis 6/1-2, 1987, 4-39, translated from « Méthodes d'analyse en musicologie », *Revue belge de musicologie,* 20, pp. 65-90, 1966.

[23] Temperley, D. *The Cognition of Basic Musical Structures*. MIT Press, 1988.