EFFICIENT EXTRACTION OF CLOSED MOTIVIC PATTERNS IN MULTI-DIMENSIONAL SYMBOLIC REPRESENTATIONS OF MUSIC

Olivier Lartillot
University of Jyväskylä
PL 35(A)
40014 Jyväskylä, Finland
lartillo@campus.jyu.fi

ABSTRACT

An efficient model for discovering repeated patterns in symbolic representations of music is presented. Combinatorial redundancy inherent in the pattern discovery paradigm is usually filtered using global selective mechanisms, based on pattern frequency and length. The proposed approach is founded instead on the concept of closed pattern, and insures lossless compression through an adaptive selection of most specific descriptions in the multi-dimensional parametric space. A notion of cyclic pattern is introduced, enabling the filtering of another form of combinatorial redundancy provoked by successive repetitions of patterns. The use of cyclic patterns implies a necessary chronological scanning of the piece, and the addition of mechanisms formalising particular *Gestalt* principles. This study shows therefore that automated analysis of music cannot rely on simple mathematical or statistical approaches, but requires instead a complex and detailed modelling of the cognitive system ruling the listening processes. The resulting algorithm is able to offer for the first time compact and relevant motivic analyses of monodies, and may therefore be applied to automated indexing of symbolic music databases. Numerous additional mechanisms need to be added in order to consider all aspects of music expression, including polyphony and complex motivic transformations.

Keywords: Closed pattern discovery, Gallois connection, Formal Concept Analysis, cyclic pattern, cognitive modelling.

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 (Ruwet, 1987; Lerdahl and Jackendoff, 1983). The pattern discovery system described in this paper is applied uniquely to symbolic representation. A direct analysis on the signal level would arouse tremendous difficulties. 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. Indeed, computer researches on this subject hardly offer results close to listeners' or musicologists' expectations. Hence the pattern discovery task is too complex to be undertaken directly at the audio signal, and needs rather a prior transcription from the audio to the symbolic representations, in order to carry out the analysis on a conceptual level.

We previously showed that the pattern discovery task leads to a problem of combinatorial redundancy, which needs to be carefully controlled (Lartillot, 2004). We proposed therefore a heuristic based on maximally specific description of pattern classes and introduced a notion of implication relation between multi-dimensional pattern description. This new paper relates both heuristics to the concept of *closed pattern* (Zaki, 2005), and to to the subconcept-superconcept relation defined in the *Formal Concept Analysis* (FCA) theory (Ganter and Wille, 1999), stemming from the Gallois connection between pattern description and pattern class. The second part of the paper introduces the concept of cyclic patterns and presents an extension of the subconcept-superconcept relation to this new paradigm. This enables a simple modelling and efficient control of the complex structural configurations found in every musical piece, even simple ones. A less formalised description of the whole theory can be found in Lartillot (2005).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

©2005 Queen Mary, University of London

2 CLOSED PATTERN DISCOVERY

This section presents the basic problem of pattern discovery and introduces the notion of closed pattern.

2.1 Definitions

Let $S =< a_1 a_2 \ldots a_N >$ be a sequence of elements of some set $a_i \in A$. A *subsequence* $S_{i,l}$ of index $i \in [1, N]$ and of length $l \in [1, N + 1 - i]$ is the sequence :

$$S_{i,l} =< a_i a_{i+1} \ldots a_{i+l-1} > . \quad (1)$$

A sub-sequence $S_{i,k}$ is *included* in another sub-sequence $S_{j,l}$, noted $S_{i,k} \subset S_{j,l}$ when $j \leqslant i$ and $i + k \leqslant j + l$. In a first simple version, a *pattern* of *length* l, denoted $P \in \mathcal{P}(S)$, can be defined as a repeated sub-sequence:

$$P \in \mathcal{P}(S) \iff \exists (i,j) \in [1,N]^2, P = S_{i,l} = S_{j,l}. \tag{2}$$

The *support* of a pattern P, denoted $\sigma(P)$, is the number of occurrences of the pattern, i.e.

$$\sigma(P) = \left|\left\{i \in [1,N], S_{i,l} = P\right\}\right|. \tag{3}$$

2.2 Redundancy Filtering

The task of discovering repeated patterns leads to combinatorial problems. Indeed each pattern of length l, contains $\sum_{i=1}^{l} i = \frac{l(l-1)}{2} = O(l^2)$ sub-patterns. All these sub-patterns are explicitly discovered by any basic pattern discovery algorithm (Zaki, 2005). One common way to solve this problem consists in focusing on the *maximal patterns* P of the sequence S, denoted $P \in \mathcal{M}(S)$, which are patterns of S not included in any other pattern of S :

$$P \in \mathcal{M}(S) \iff \begin{cases} P \in \mathcal{P}(S) \\ \nexists Q \in \mathcal{P}(S), P \subset Q. \end{cases} \tag{4}$$

This heuristic enables a significant reduction of the number of discovered pattern, but leads also to a loss of information. Indeed, not all the sub-patterns may be immediately reconstructed knowing the maximal patterns. For instance, the grey sub-pattern in figure 1 is redundant as it can be simply retrieved as a suffix of the black pattern. However, in figure 2, the same grey sub-pattern is not redundant any more, because its support (4) is larger than the support of the black pattern (2).

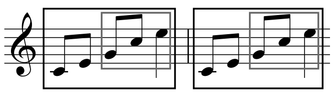

Figure 1: The grey sub-pattern is not a closed pattern: it is a simple suffix of the black pattern.

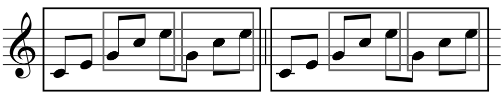

Figure 2: The grey sub-pattern is now closed: its support is bigger than the support of the black pattern.

A pattern P will be called *closed*, denoted $P \in \mathcal{C}(S)$, if and only if there exists no proper super-pattern Q of same support :

$$P \in \mathcal{C}(S) \iff \begin{cases} P \in \mathcal{P}(S) \\ \nexists Q \in \mathcal{P}(S), \begin{cases} P \subset Q \\ \sigma(P) = \sigma(Q). \end{cases} \end{cases} \tag{5}$$

The set of closed patterns offers a compact and lossless description of the musical piece.

2.3 An Incremental and Chronological Approach

This paper proposes an *incremental* and *chronological* approach to closed pattern discovery. Section 4 justifies this strategy, by showing the necessity to model successive repetitions of a same pattern as a traversal through one single cyclic pattern, and hence to consider a chronological approach of music. A detailed and illustrated description of the algorithm summarised below has been presented in (Lartillot, 2004).

2.3.1 Incremental approach

The successive prefixes of each pattern are discovered progressively: they are considered as successive *intermediary states* of a *pattern chain* (PC) whose final state represents the whole pattern. At each step of the progressive construction, all the occurrences of the last discovered prefix are considered. Identical continuations form new extensions of the prefix, represented as children of the current state. Since each state can accept several children, the set of all patterns form a tree, called *pattern tree* (PT). An example of PT can be seen in figure 4, above the score.

Similarly, pattern occurrences are also represented as chains – called pattern occurrence chains (POCs) – whose successive states represent the successive prefixes (see figure 4, below the score). Each state of a POC is related to its corresponding PC. As each pattern occurrence can feature several different possible continuations, the set of all pattern occurrences that are initiated by one note forms a tree, called pattern occurrence tree (POT). The root of each POT is associated to the root of the PT (node a in figure 4), which represents the simple concept of note, and is therefore called *note pattern*. Since all notes can potentially initiate a POT, they are all occurrences of the note pattern.

The inclusion relation between patterns may be decomposed as a product of two sub-relations: prefix and suffix relations. Any sub-pattern will then be considered as a prefix of a suffix of a pattern. The closure of a pattern P may then be assessed following these two relations.

1. A pattern P is *prefix-closed* if there does not exist pattern Q of which P is a prefix of same support. Since all the prefixes of patterns are displayed in PTs, non prefix-closed patterns are not discarded. The explicit representation of prefixes as intermediary states of pattern chains induces a mere linear complexity, as each pattern of length l is represented by l states.

2. The problem of close patterns selection will therefore be studied along suffix relations only. A pattern P is *suffix-closed* if there does not exist any pattern Q of which P is a suffix of same support.

2.3.2 Chronological approach

The pattern discovery process is chronological: the main routine of the algorithm consists in a single traversal of the sequence S, from the first element a_i to the last element a_N. Each new element a_i induces an update of the whole pattern tree.

For this purpose, hash-tables memorise all the possible *continuations* of P, that is, all the possible elements appearing just after each of its occurrence. If a previous occurrence of pattern P has been continued by an element identical to a_i, then an extension P' of pattern P may be discovered (if not already) and represented as one of its child. The suffix-closed condition should however apply: there should not exist a pattern Q' of which P' is a suffix of same support. During the chronological analysis, a pattern P' that was first considered as non-closed may become closed once discovering a new occurrence that is not an occurrence of the super-pattern Q' (an example will be shown in paragraph 3.3). Super-patterns Q' can easily be retrieved, as extensions of the corresponding super-patterns Q of P by the same element a_i. Not all the super-patterns Q need to be considered, but only those containing an occurrence concluded by previous element a_{i-1}.

3 MULTI-DIMENSIONAL CLOSED PATTERNS

In previous subsection, pattern was searched in sequences of elements $S =< a_1 a_2 \ldots a_N >, a_i \in A$. Music, on the contrary, is expressed in a multi-dimensional parametric space.

3.1 Multi-Parametric Description of Music

The model presented in this paper only analyses monodic sequences, i.e. successions of notes without superposition. Any monodic sequence can therefore be represented as previously :

$$S =< n_1 n_2 \ldots n_N >, n_i \in \mathcal{N}, \tag{6}$$

where $\mathcal{N}$ is the parametric space of notes. In the model, this note space is simply reduced to :

$$\mathcal{N} = \mathsf{diat} \times \mathsf{chro} \times \mathsf{rhyt} \tag{7}$$

where

- diat, or *diatonic pitch space*, represents pitches as positions in the implicit tonal scale. Diatonic transpositions will be detected in this space.
- chro, or *chromatic pitch space*, represents pitches as positions on the piano keyboard. Following the MIDI standard, with middle C is associated the value 60.
- rhyt, or *metrical space*, represents temporal positions in term of distance from the beginning of the musical sequence. The rhythmic unit of the metrical space is given by the time signature.

The pattern discovery task cannot be directly applied to this note sequence S, because each successive note is related to a distinct metrical position and is therefore distinct. Even if the metrical space is discarded, neither rhythmic patterns nor transposed melodic patterns may be discovered. Following common practice, the musical sequences will be modelled as a succession of intervals between successive notes :

$$S = (\overrightarrow{n_1 n_2} \succ \overrightarrow{n_2 n_3} \succ \ldots \succ \overrightarrow{n_{N-1} n_N}). \tag{8}$$

An interval $\overrightarrow{n_i n_{i+1}} \in \overrightarrow{\mathcal{N}}$ is a vector between two points of the note space $\mathcal{N}$

$$\begin{aligned} n_i &= (\mathsf{diat} = d_i, \mathsf{chro} = c_i, \mathsf{rhyt} = t_i) \\ n_{i+1} &= (\mathsf{diat} = d_{i+1}, \mathsf{chro} = c_{i+1}, \mathsf{rhyt} = t_{i+1}) \end{aligned} \tag{9}$$

and can therefore be described by the three coordinates :

$$\overrightarrow{n_i n_{i+1}} = \begin{pmatrix} \mathsf{diat}(\overrightarrow{n_i n_{i+1}}) \\ \mathsf{chro}(\overrightarrow{n_i n_{i+1}}) \\ \mathsf{rhyt}(\overrightarrow{n_i n_{i+1}}) \end{pmatrix} = \begin{pmatrix} d_{i+1} - d_i \\ c_{i+1} - c_i \\ t_{i+1} - t_i \end{pmatrix}. \tag{10}$$

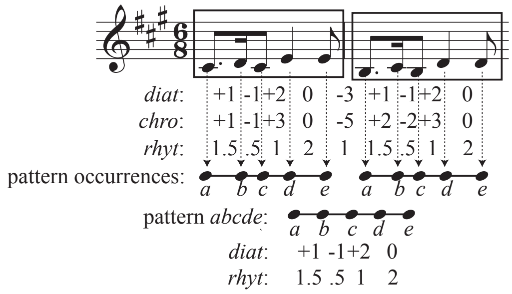

Figure 3: Multi-dimensional description of a musical sequence, which contains two occurrences of a pattern $abcde$.

3.2 Formal *Context* Representation of Patterns

We will represent musical patterns with the help of a conceptual framework that defines *objects* associated with different kinds of *attributes* (Ganter and Wille, 1999). These attributes consist not only of the different musical dimensions, but also of the different sub-patterns and super-patterns. Following the incremental and chronological approach explained in paragraph 2.3, we can restrict our study of the inclusion relations between patterns to suffix relations. In this respect the objects of the pattern descriptions are the successive notes of the musical sequence forming the set $\mathcal{N}(S)$. Each note $n_i \in \mathcal{N}(S)$ relates to a specific *temporal context*, defined by *the part of the musical sequence concluded by this note* n_i.

Each note n_i is described firstly by the different musical characteristics of the preceding interval: $\overrightarrow{n_{i-1} n_i}$.

$$\mathcal{D}^{0,p}_{\mathsf{desc}}(n_i) \; : \; \begin{aligned} &\mathsf{desc}(\overrightarrow{n_{i-1} n_i}) = p, \\ &\mathsf{desc} \in \{\mathsf{diat}, \mathsf{chro}, \mathsf{rhyt}\}, p \in \mathsf{desc}. \end{aligned} \tag{11}$$

Each note n_i is also described by the musical characteristics of the previous intervals:

$$\mathcal{D}^{j,p}_{\mathsf{desc}}(n_i) \; : \; \begin{aligned} &\mathsf{desc}(\overrightarrow{n_{i-j-1} n_{i-j}}) = p, \\ &\mathsf{desc} \in \{\mathsf{diat}, \mathsf{chro}, \mathsf{rhyt}\}, p \in \mathsf{desc}. \end{aligned} \tag{12}$$

Then the pattern description of the sequence S may be expressed as a *formal context* $(\mathcal{N}(S), \mathcal{D}, I)$ where :

- the set of *objects* is $\mathcal{N}(S)$: the set of notes in S,
- the set of *attributes* is $\mathcal{D}$: the set of elementary musical descriptions defined by equations 11 and 12,

- and I is the binary relation between $\mathcal{N}(S)$ and $\mathcal{D}$, called *incidence*, defined by:

$$(n_i, \delta) \in I \iff \delta(n_i) \text{ is true.} \qquad (13)$$

The *derived description* C' of a set of notes $C \subset \mathcal{N}(S)$ is defined as the common description of all these notes:

$$C' = \{\delta \in \mathcal{D} \mid \forall n \in A, (n, \delta) \in I\}. \qquad (14)$$

The notes in C are therefore occurrences of a same pattern, which is maximally described by C'.

The *derived class* D' of a complex description $D \subset \mathcal{D}$ is dually defined as the set of notes complying with this description:

$$D' = \{n \in \mathcal{N}(S) \mid \forall \delta \in D, (n, \delta) \in I\}. \qquad (15)$$

The pattern discovery task consists in finding exhaustive class D' sharing a same description D. The trouble is, lots of different descriptions D_i may lead to same classes D'_i.

3.3 Formal *Concept* Representation of Patterns

The *derivators operations* defined by equation 14 and 15 establish a Gallois connection between the power set lattices on $\mathcal{N}(S)$ and $\mathcal{D}$. The Gallois connection leads to a dual isomorphism between two closure systems, whose elements, called *formal concepts* of the formal context $(\mathcal{S}(S), \mathcal{D}, I)$ corresponds exactly to the *close patterns* $P = (C, D)$, verifying:

$$C \subset \mathcal{N}(S), D \subset \mathcal{D}, C' = D, \text{ and } D' = C. \qquad (16)$$

For a close pattern $P = (C, D)$, C is called the *extent* of D and D the *intent* of C. We may simply call C and D respectively the *class* and the *description* of P.

Hence, for a set of patterns $P_i = (D'_i, D_i)$ of same class $D'_i = C$, the close pattern $P = (C, D)$ is described using the derived operator C' defined in equation 14: it contains all the elementary descriptions common to all notes of the class C. In other words, closed patterns are described as precisely as possible. However, as listeners tend to perceive only repetition of connex sub-sequences, only the descriptions of the longest set of contiguous intervals ($\mathcal{D}^j \succ \mathcal{D}^{j-1} \succ \ldots \succ \mathcal{D}^0$) leading to the context note should be selected. Older descriptions $\mathcal{D}^{j+k}$ should be discarded if there is no description $\mathcal{D}^{j+1}$ associated to step $j+1$. We have proposed a further continuity constraint, that seems to correspond more deeply to listeners perception, stating that contiguous intervals should be described by same dimensions:

$$\begin{array}{l} \forall i \in [1, j], \exists \mathsf{desc} \in \{\mathsf{diat}, \mathsf{chro}, \mathsf{rhyt}\}, \\ \exists (p, p') \in \mathsf{desc}^2, \mathcal{D}^{j,p}_{\mathsf{desc}} \text{ and } \mathcal{D}^{j-1,p'}_{\mathsf{desc}} \end{array} \qquad (17)$$

Closed patterns, as formal concepts, are naturally ordered by the *subconcept-superconcept* relation defined by

$$(C_1, D_1) < (C_2, D_2) \iff C_1 \subset C_2 \;\; (\iff D_2 \subset D_1). \qquad (18)$$

(C_1, D_1) may therefore be considered as *more specific* than (C_2, D_2).

The incremental and chronological pattern discovery methodology presented in section 2.3 may be generalised using this multi-parametric definition of closed pattern. For instance, pattern $a \succ b \succ c \succ d \succ e$ (more simply denoted e), in figure 4, features melodic and rhythmical descriptions:

$$\left(\begin{array}{c} \mathsf{diat} = 0 \\ \mathsf{rhyt} = .5 \end{array} \succ \begin{array}{c} \mathsf{diat} = 0 \\ \mathsf{rhyt} = .5 \end{array} \succ \begin{array}{c} \mathsf{diat} = -2 \\ \mathsf{rhyt} = .5 \end{array} \succ \mathsf{rhyt} = 4 \right)$$

whereas pattern $a \succ f \succ g \succ h \succ i$ (or i) only features its rhythmic part:

$$(\mathsf{rhyt} = .5 \succ \mathsf{rhyt} = .5 \succ \mathsf{rhyt} = .5 \succ \mathsf{rhyt} = 4).$$

Hence pattern e is more specific than pattern i. When only the two first occurrences are analysed, as both patterns have same support, only the more specific pattern e should be explicitly represented. But the less specific pattern i will be represented once the last occurrence is discovered, since it is not an occurrence of the more specific pattern e.

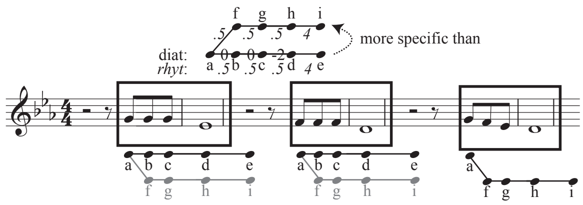

Figure 4: The rhythmic pattern $afghi$ is less specific than the melodico-rhythmic pattern $abcde$.

3.4 Optimal score description

In order to reduce the space complexity of the pattern representation, and also to simplify as much as possible the pattern description of the musical score, to each closed pattern $P = (C, D)$ will be associated a *specific class* $SC(P)$ which consists of the set of occurrences that are not included into classes of more specific patterns:

$$SC(P) = C \setminus \bigcup_{(C', D') < (C, D)} C'. \qquad (19)$$

Reversely, the general pattern class can be retrieved by unifying the specific class with the union of the specific classes of all the more specific patterns :

$$C = SC(P) \cup \bigcup_{P' < P} SC(P'). \qquad (20)$$

During the chronological analysis of the musical score, only the specific classes are constructed. But each time a specific pattern occurrence is discovered, all the less specific patterns need to be recalled by the algorithm, because their extensions may lead to the discovery of new specific patterns. For instance, in figure 5, groups 1 and 3 are occurrences of pattern h, and groups 3 and 4 are occurrences of pattern d. Since pattern d is more specific, the less specific pattern h does not need to be associated with group 4 (that is why it is represented in grey). However in order to detect groups 2 and 5 as occurrences of pattern l, it is necessary to implicitly consider group 4 as an occurrence of pattern h. Hence, even if pattern h, since less specific than d, was not explicitly associated with group 4, it had to be considered implicitly in order to construct pattern l. Implicit information is reconstituted through a traversal of the pattern network along the subconcept-superconcept relations.

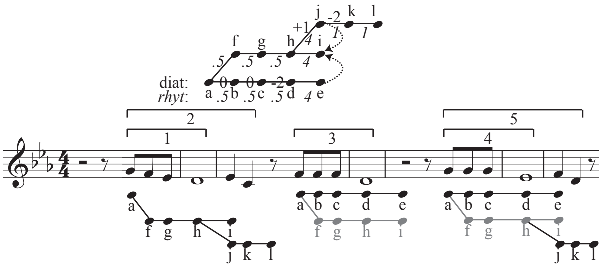

Figure 5: Group 4 can be simply considered as occurrence of pattern d. However, in order to detect group 5 as occurrence of pattern l, it is necessary to implicitly infer group 4 as occurrence of pattern h too.

3.5 Generalization of patterns

New patterns can be discovered through a simple generalisation of already known patterns. In bar 7 of figure 6, the two first notes form an occurrence of pattern h described by $\mathsf{diat} = +1$ and $\mathsf{rhyt} = 1$. The third note cannot however be associated with the known extension of pattern h into pattern i with description $\mathsf{diat} = 0$ and $\mathsf{rhyt} = 2$, because the melodic description $\mathsf{diat} = 0$ does not match here. However, as the rhythmic description $\mathsf{rhyt} = 2$ matches, a new extension j is discovered as a generalisation of pattern i.

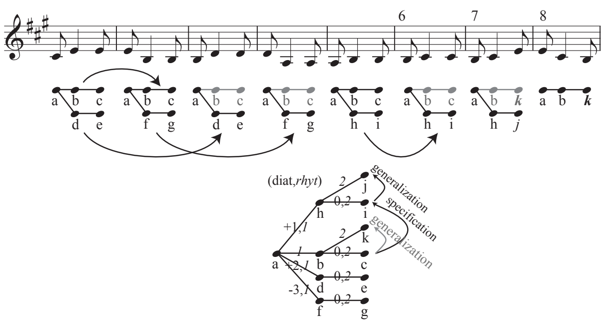

Figure 6: Progressive discovery of the pattern repetitions on the score and the resulting pattern tree (below the score). Pattern descriptions in grey are less specific than simultaneous descriptions in black.

The less specific patterns, although usually not explicitly represented in the analysis, should be updated if necessary. In particular, when a generalisation of a pattern is discovered, the generalisation of all its more general patterns should also be considered. For instance, as i has been generalised into j, c should be generalised into k in the same way. In this way, the analysis of the next bar (8) consists simply in recognising the already known general pattern k.

4 CYCLIC PATTERNS

In this section, we present another important factor of redundancy that, contrary to closed patterns, has not been studied in current general algorithmic researches.

4.1 Redundancy Due to Successive Repetitions

Combinatory explosion can be caused by successive repetitions of a same pattern (for instance the pattern $a \succ b \succ c$ in figure 7, of description $(\mathsf{rhyt} = 1 \succ \mathsf{rhyt} = 2)$). As each occurrence is followed by the beginning of a new occurrence, each pattern can be extended – leading to pattern d of description $(\mathsf{rhyt} = 1 \succ \mathsf{rhyt} = 2 \succ \mathsf{rhyt} = 1)$ – by a new interval whose description $(\mathsf{rhyt} = 1)$ is identical to the description of the first interval of the same pattern (i.e., pattern b). This extension can be prolonged recursively (into e, f, etc.), leading to a combinatorial explosion of patterns that are not perceived due to their complex intertwining.

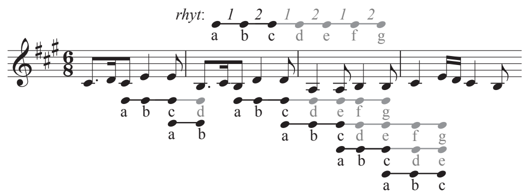

Figure 7: Multiple successive repetitions of pattern abc form a complex intertwining of non-perceived structures.

4.2 Cyclic Patterns

The graph representation (figure 7) shows that the last state of each occurrence of pattern c is superimposed on 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 8). The initial acyclic pattern $a \succ b \succ c$ leads therefore to a cyclic pattern that oscillates between two states $b' \circlearrowright c'$, corresponding to rhythmic values 1 and 2. Indeed, when listening to the remainder of the rhythmic sequence, we actually perceive this progressive oscillation between these two states b' and c'. Hence this cycle-based modelling explains a common listening strategy, and resolves the problem of combinatorial redundancy.

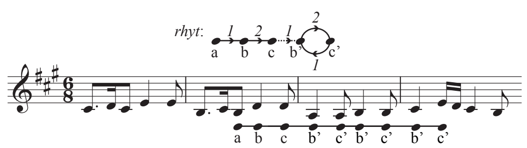

Figure 8: The successive repetitions of pattern $a \succ b \succ c$ lead to an oscillation between states $b' \circlearrowright c'$.

This cyclic PC $b' \circlearrowright c'$ is considered as a continuation of the original acyclic PC $a \succ b \succ c$ (figure 8) . Indeed, the first repetition of the rhythmic period is not perceived as a period as such but rather as a simple pattern: its successive notes are simply linked to the progressive states a, b and c of the acyclic PC. On the contrary, the following notes extends the POC, which cannot therefore be associated to the acyclic PC anymore, and are therefore linked to the successive states of the PC (b' and c'). The whole periodic sequence constitutes then a single POC representing the traversal of the acyclic PC followed by the cyclic PC.

It can be remarked also that, by property of the cyclic PC, no segmentation is explicitly represented between successive repetitions. The periodic sequence in figure 8 can be considered as a succession of periods quaver-crochet, or on the contrary crochet-quaver. Listeners may be inclined to segment at any phase of the cyclic PC, or not to segment at all.

This additional concept immediately solves the redundancy problem. Indeed, the unique POC that is progressively extended is more specific than its suffixes, which cannot therefore be extended any more.

This phenomenon of successive repetition, although very frequent in musical expression, has been rarely studied. Cambouropoulos (1998) proposed to control the combinatorial explosion by selecting, once the analyses completed, patterns featuring minimal temporal overlapping between occurrences. The trouble is, as the selection is inferred globally, relevant patterns may be discarded. Besides combinatorial redundancy remains problematic since the filtering is carried out after the actual analysis phase. Our focus on local configurations enables a more precise filtering.

4.3 General and Specific Cycles

The application of this concept on the multidimensional musical space requires a generalisation of specificity relations, defined in previous section, to cyclic patterns. A cyclic pattern C is 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.

In Figure 9, the seven first notes of the cycle oscillate around the cyclic PC $b' \circlearrowright c'$ described by:

$$\mathsf{rhyt} = 1 \circlearrowright \begin{array}{c}\mathsf{rhyt} = 2 \\ \mathsf{diat} = 0\end{array}.$$

Then appears a more specific cycle $d' \circlearrowright e'$ described by

$$\begin{array}{c}\mathsf{rhyt} = 1 \\ \mathsf{diat} = +1\end{array} \circlearrowright \begin{array}{c}\mathsf{rhyt} = 2 \\ \mathsf{diat} = 0\end{array}$$

and is generalised after four notes to cycle $d'' \circlearrowright f'$ that does not feature the unison interval any more:

$$\begin{array}{c}\mathsf{rhyt} = 1 \\ \mathsf{diat} = +1\end{array} \circlearrowright \mathsf{rhyt} = 2.$$

Moreover, following the rule of generalisation of generalised patterns explained in paragraph 3.5, the more general cycle $b' - c'$ too needs to be generalised into a cycle

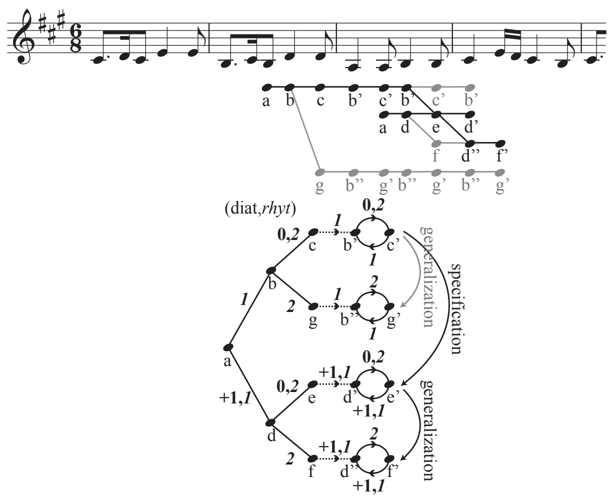

Figure 9: More detailed analysis of the perceived cyclic configurations.

$b'' - g'$ where the unison interval has been discarded:

$$\mathsf{rhyt} = 1 \circlearrowright \mathsf{rhyt} = 2.$$

These different cycles seem to be perceptible by listeners. Moreover, the integration of this phenomenon into the model helps insuring the relevance of the results and avoiding numerous unwanted combinatorial redundancies.

4.4 The Figure/Ground Rule

Another kind of redundancy appears when occurrences of a pattern – such as the melodico-rhythmic pattern c in figure 10, described by $\mathsf{diat} = -2$ and $\mathsf{rhyt} = 1$ – are superposed to a cyclic pattern (b'), such that the pattern (c) is more specific than the cycle period (b', simply rhythmic: $\mathsf{rhyt} = 1$). In this case, the intervals that follow these occurrences are identical, since they are related to the same state (b') of the cyclic pattern. Logically the pattern could be extended following the successive extensions of the cyclic patterns (leading to patterns d, e, etc.). This phenomenon, which may frequently appear in a musical piece, would lead to another combinatorial proliferation of redundant structures if not correctly controlled by relevant mechanisms. On the contrary, following the *Gestalt* Figure/Ground rule, listeners tend to perceive the pattern c as a specific figure that emerges above the periodic background. Following this rule, the figure cannot be extended (into d) by a description that can be simply identified with the background extension.

5 RESULTS

This model was first developed as a library of *OpenMusic*, called *OMkanthus*. A new version will be included in the next version 2.0 of *MIDItoolbox* (Eerola and Toiviainen, 2004), a *Matlab* toolbox dedicated to music analysis. The model can analyse monodic musical pieces (i.e., constituted by a series of non-superposed notes) and highlight the discovered patterns on a score.

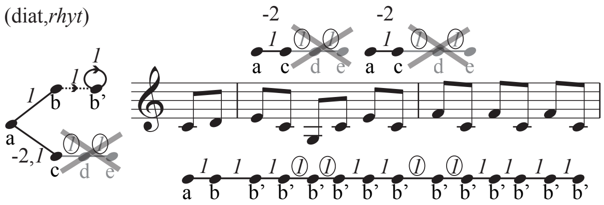

Figure 10: Pattern c is a specific figure, above a background generated by the cyclic pattern b'.

5.1 Experiments

The model has been tested with different musical sequences taken from several musical genres (classical music, pop, jazz, etc.). Table 1 shows some results. The experiment has been undertaken with version 0.6.8 of *OMkanthus* on a 1-GHz PowerMac G4. A musicologist expert has validated the analyses. The proportion of patterns considered as relevant is displayed in the table.

The analysis of a medieval song called *Geisslerlied* (Ruwet, 1987) – sometimes used as a reference test for formalised motivic analysis – gave quite relevant results. The analysis has been actually carried out on a slight simplification of the actual piece presented by Ruwet, excluding local motivic variations out of reach of the current modelling.

The melodico-rhythmic analysis of the French song *Au clair de la lune* posed problems: 21 patterns were discovered from a 44-note long sequence. This is due to the fact that the successive steps of progressive generalisation or specification of cycles are currently modelled using distinct intermediary cyclic patterns. The inference of these redundant cyclic patterns will be avoided in further works.

The algorithm has been successfully applied on a melodic analysis of a complete two-voice *Invention* by J.S. Bach. Figure 11 shows the analysis of the 21 first bars. The repetition of ascending quarter notes in bars 3 and 4 has not been detected because the contour dimension was not considered in the experiment. The cyclic patterns are represented by graduated lines, the graduation representing each return of one possible phase. As explained in section 4.2, no preference is given by the model between different possible phases of the same cycle. The rhythmic analysis of the piece, on the contrary, failed, due to the alternation of sequences of either quarter notes or 8th notes, which will require a formalisation through hierarchical pattern chains (where successive states of higher-level patterns are linked to distinct lower-level patterns).

The analysis of The Beatles' *Obla Di Obla Da* melody shows 14 relevant pattern classes, representing the chorus, verses, phrases and motives inside each of these structures. The 4 irrelevant patterns are redundant patterns subsumed by the 14 relevant ones.

In all these pieces, some patterns are considered as irrelevant because they cannot be perceived as such by listeners. Additional mechanisms should be added to prevent these irrelevant inferences, based on short-term memory, top-down mechanisms, etc.

5.2 About Algorithm Complexity

The algorithm complexity may be expressed first in terms of discovered structures: proliferation of redundant patterns, for instance, would lead to combinatorial explosion, since each new structure needs proper processes assessing its interrelationships with other structures, and inferring 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. Concerning technical implementation, the prototype needs further optimisations. Yet the modelling has been conceived with a view to minimising computational costs. Hence the identification of similar descriptions is based on hash tables, as explained in paragraph 2.3.2, which reduce time complexity.

The overall computational modelling results in a complex system formed by a large number of highly dependent mechanisms. Without a real synthetic vision of the whole system, no general assessment of the global complexity of the modelling has been achieved yet. The complete rebuilding of the modelling currently undertaken should enable a better awareness and control of complexity.

6 CURRENT RESEARCHES

The structures currently found are based solely on pattern repetitions. Segmentation rules based on *Gestalt* principles of proximity and similarity (Lerdahl and Jackendoff, 1983) (Cambouropoulos, 1998) need to be added. 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 (Deliège, 1987). The study will focus in particular on the competitive/collaborative interrelations between the two mechanisms, in particular the masking effect of local disjunction on pattern discovery.

Musical transformations should be considered, such as local insertion or deletion of notes. Solutions have been proposed (Rolland, 1999) based on optimal alignments between approximate repetitions using dynamic programming and edit distances. We are developing algorithms that automatically discover, from the rough surface level of musical sequences, musical transformations revealing the sequence of pivotal notes forming the deep structure of these sequences. These mechanisms induce new connections between non-successive notes, transforming the *syntagmatic chain* of the original musical sequence into a complex *syntagmatic graph*. The direct application of the pattern discovery algorithm on this syntagmatic graph will enable the detection of approximate repetitions and require the introduction of further perceptual heuristics in order to select precisely the relevant results.

Our approach is limited to the detection of repeated monodic patterns. Music in general is polyphonic, where simultaneous notes form chords and parallel voices. Researches have been carried out in this domain (Meredith et al., 2002), focused on the discovery of exact repetitions along different separate dimensions. Our model will be

Table 1: Results of analyses, either melodic (M) or melodico-rythmic (M+R), performed by *OMkanthus* 0.6.8.

Musical sequence		**Analysis**	**Pattern classes**			**Computation**
Name	**Notes**	**type**	**Discovered**	**Relevant**	**Success**	**time**
Geisslerlied, slightly simplified	108	M	6	5	83%	2.2 sec.
Au clair de la lune (folk song)	44	M+R	21	5	24%	5.6 sec.
Bach, *Invention in D minor*, BWV 775	283	M	49	34	69%	37.6 sec.
Mozart, *Sonata in A*, K331 first theme, first half, melody	36	M+R	14	10	71%	0.8 sec.
The Beatles, *Obla Di Obla Da*	390	M	14	10	71%	28.1 sec.

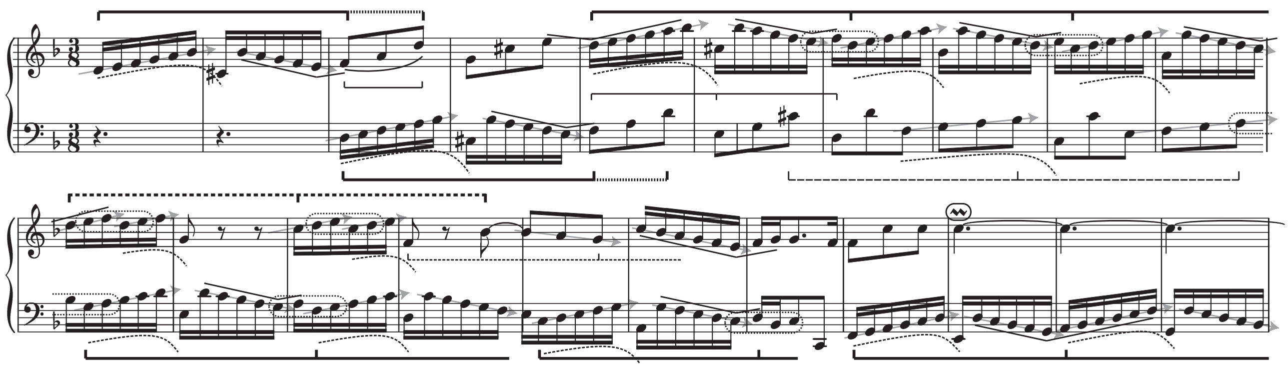

Figure 11: Automated motivic analysis of J.S. Bach's *Invention in D minor* BWV 775, 21 first bars. The occurrences of each pattern class are designated in a distinct way.

generalised to polyphony following the syntagmatic graph principle. We are developing algorithms that construct, from polyphonies, syntagmatic chains representing distinct monodic streams. These chains may be intertwined, forming complex graphs along which the pattern discovery algorithm will be applied. Pattern of chords may also be considered in future works.

The automated discovery of repeated patterns can be applied to automated indexing of musical content in symbolic music databases. This approach may be generalised later to audio databases, once robust and general tools for automated transcription of musical sound into symbolic scores will be available. A new kind of similarity distance between musical pieces may be defined, based on these pattern descriptions, offering new ways of browsing inside a music database using pattern-based similarity distance.

ACKNOWLEDGEMENTS

I would like to thank Michèle Sebag, Marc Damez and the reviewers for their invaluable advice.

REFERENCES

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

I. Deliège. Grouping conditions in listening to music: An approach to lerdahl and jackendoffs grouping preference rules. *Music Perception*, 4(4):325–350, 1987.

T. Eerola and P. Toiviainen. Mir in matlab: The midi toolbox. In *Proceedings of the International Conference on Music Information Retrieval*, 2004.

B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag, 1999.

O. Lartillot. A multi-parametric and redundancy-filtering approach to pattern identification. In *Proceedings of the International Conference on Music Information Retrieval*, 2004.

O. Lartillot. Efficient extraction of closed motivic patterns in multi-dimensional symbolic representations of music. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*. IEEE Computer Society Press, 2005.

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

D. Meredith, K. Lemström, and G. Wiggins. Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, 31(4):321–345, 2002.

P.-Y. Rolland. Discovering patterns in musical sequences. *Journal of New Music Research*, 28:334–350, 1999.

N. Ruwet. Methods of analysis in musicology. *Music Analysis*, 6(1-2):4–39, 1987.

M. Zaki. Efficient algorithms for mining closed itemsets and their lattice structure. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):462–478, 2005.