Perception-Based Musical Pattern Discovery

Olivier Lartillot

Ircam — Centre Georges-Pompidou
email: Olivier.Lartillot@ircam.fr

Abstract

A new general methodology for Musical Pattern Discovery is proposed, which tries to mimic the flow of cognitive and sub-cognitive inferences that are processed when hearing a piece of music. A brief survey shows the necessity to handle such perceptual heuristics and to specify perceptual constraints on discoverable structures. For instance, successive notes between patterns should verify a specific property of closeness. A musical pattern class is defined as a set of characteristics that are approximately shared by different pattern occurrences within the score. Moreover, pattern occurrence not only relies on internal sequence properties, but also on external context. Onto the score is build pattern occurrence chains which themselves interface with pattern class chains. Pattern classes may be inter-associated, in order to formalize relations of inclusion or repetition. The implemented algorithm is able to discover pertinent patterns, even when occurrences are, as in everyday music, translated, slightly distorted, slowed or fastened.

1 Introduction

Musical Pattern Discovery (MPD) is an emerging discipline, which aims at offering automated analyses of musical scores (Lartillot 2003). Lots of Musical Information Retrieval applications would highly benefit from the tools offered by MPD. Indeed, as MPD would automatically retrieve the content of score, new types of music browsing, listening, visualizing, etc., would be possible.

Nowadays, however, no computer system is able to complete in a pertinent way the demanding task of MPD. In this paper, we suggest a new approach that may offer an answer to fundamental problems arisen in this discipline, and that could therefore solve a certain number of difficulties.

We introduce a general methodology whose intuition stems from a mimicking of human perception. We will defend such a position through a critical overview of current approaches in MPD. This overview will lead us towards a characterization of musical pattern that takes into account the fundamental notions of repetition, sequence and similarity in the musical context of temporal perception. We then propose a complete computational model that attempts to follow such methodology, and that already offers promising results.

The remaining of the paper is structured as follows. A preliminary survey of MPD concepts and methods will enable a formalization of our approach. We then introduce the data architecture and the algorithmic principles of the proposed system. Finally, we illustrate the use of the system in various musical contexts.

2 Some Necessary Conditions For Musical Pattern Discovery

2.1 Pattern Characterization

The concept of musical pattern may be characterized following three main criteria:

Implicit knowledge. Pattern may result from implicit knowledge that cannot be obtained directly from the score, such as: expected phrase length or metric (Lerdahl and Jackendoff 1983). The trouble is, pattern perception cannot firmly rely on such theory-oriented implicit knowledge. Indeed, musical motives may be structured in an ambiguous way, through a breaking of these rigid rules, playing with the listener's expectations and conveying musical delightedness. A mere fugue may easily show that patterns (here themes, for instance) may appear at very unexpected metric places, and may feature a long temporal extension.

Local boundaries. Low-level structural properties of the musical surface may be obtained through local boundary detection (Cambouropoulos 1998). For instance, grouping boundaries may be introduced between entities that contrast one with the other according to their pitch, duration, intensity parameters, etc. Such heuristics may enable an understanding of metric phenomenon, for instance. However, such local segmentation does not contribute to the understanding to the idea of musical pattern itself. Indeed, a musical pattern is implicitly built through contrastive aggregation.

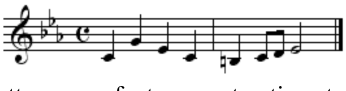

Fig. 1. A pattern may feature contrastive steps.

In figure 1, the first leap of interval of fifth, although triggering a contrastive idea, is the important element that characterizes the beginning of the pattern itself. A pattern is therefore not a conservation of sameness, but on the contrary a travel along differences. In fact, similar patterns features similar differences.

In the same time, it has to be recognized that local boundaries may provoke a breaking of sequencing. We have integrated such characteristic of music in our mechanism of stream perception inside polyphony (see paragraph 4.1).

Repetition. Finally, a musical pattern may be defined as a set of characteristic that is shared by several sets of notes throughout the score. These sets of notes are said to be *similar* in a certain sense. This may look as a particularly intricate definition of pattern, which could have been formalized more simply as: a set of notes that is approximately "repeated". But two difficulties should be taken into account: firstly, that the concept of pattern refers either to the characteristics, to a prototype of such characteristics, or to occurrences; secondly, that such concept of similarity has to be explicitly defined (see paragraph 2.3 and 2.4).

This "repetition"-oriented criterion of pattern seems to remain the most relevant one, since music motives are classically defined in this way (Nattiez 1990).

2.2 Set Paradigm Vs. String Paradigm

In previous definition, a pattern is a "repeated" set of notes. A pattern may in fact be either a general set of notes (Wiggins, Lemström and Meredith 2002) or a sequence, that is: a succession of notes, where successive notes in the sequence are successive notes in the score.

If pattern is not constrained to be a sequence, temporal distance between nearest notes within this set may be arbitrarily large. We previously claimed the necessity to enable such a general view of pattern, in order to catch slow melodies accompanied by fast arpeggios for instance (Lartillot 2002b). However, even if limitations are set on temporal distance between successive notes, such a loose formalization leads to computational ineffectiveness — since lots of absurd patterns may be found — and cognitive incoherencies (Hofstadter 2002). It appears that musical listening capabilities, such as the one previously described, may be explained in a more efficient way through the idea of parallel sequences (see paragraph 4.1).

2.3 Musical Similarity

The idea of successiveness should not be considered in a rigid way, in order to enable deletions of notes or insertions of new ones in the pattern. Dynamic programming (Rolland 1999) is the most classical way to handle such operations. But music features other kinds of sequence transformation, such as passing notes or appoggiaturas, which should be also considered.

Now patterns may be subject to other kinds of transformation. Simply transposed patterns may be detected by considering each pattern in its own transposition reference. For example, if patterns are described not with absolute pitch, but with relative pitch whose reference is the absolute pitch of the first note of the pattern, then such descriptions of transposed patterns are exactly identical.

In the same way, slower and faster patterns may be considered as identical one with the other if a relative temporal representation is considered. For this purpose, instead of considering the temporal interval between successive notes, the quotient between current temporal interval and first temporal interval is considered.

But real music features much more complex transformations. In particular, pitch and temporal distortions may appear locally inside patterns. To handle such plasticity, more relative viewpoints of the pattern may be considered, such as the contour representation in particular. However, such a crude representation is so loose that non-pertinent repetitions may also be detected.

In fact, when considering such local distortions, there exist no *viewpoint* (Conklin and Anagnostopoulou 2001) sufficiently loose for finding an exact repetition but in the same time sufficiently detailed for avoiding non-pertinent inferences. Therefore approximate repetition has to be tentatively inferred, to be *induced* (Lartillot 2002a) from rough phenomenon, even if risks have to be taken.

2.4 Incremental Inference of Similarity

Global analyses of the score, such as naïve statistics, not taking into consideration the incremental expression of music, fail to catch the essential temporal aspect. Classical pattern analyses, by explicitly considering prefixes and suffixes may offer better results. Dynamic programming successfully highlights the necessity to consider a progressive and chronological scanning of patterns.

Lots of musical phenomenon deeply relies on the fact that music is progressively perceived, and that the listener itself progressively infers new knowledge about what he is currently hearing. Therefore, music listening should be considered as a kind of progressive reasoning. That is why some configurations are not detected and therefore not pertinent, simply because they cannot be caught during progressive listening.

Hence, pattern cannot be defined solely along internal description, but also along external criteria, or *context*. It is senseless, therefore, to measure the similarity between sequences out of their context.

Patterns of figure 2 may be considered as two occurrences of a same abstraction, because of the intrinsic similarity, only if this example was the actual score itself. When these patterns are included inside a real score, their similarity should be inferred only if they share a similar context.

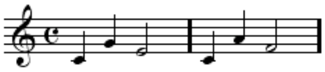

Fig. 2. The similarity of patterns cannot be measured outside of their actual context.

The incremental and logical thinking that builds human perception of music is ruled by fundamental principles, which are necessary for insuring a coherent process. For example, every time a sequence is considered as an occurrence of a pattern, every suffix could themselves be considered as occurrences of other pattern class, for simple mathematical reasons. But cognitively speaking, such inferences are not pertinent, since they do not correspond to inference human makes when listening to music. This is due to the fact that the first longest pattern was sufficient to explain the phenomenon, and that further inferences of suffixes would only infringe a clear analysis of the score. That is why suffix of pattern should not be explicitly represented (Dannenberg 2002).

2.5 Selection

As many patterns may be found, pertinent patterns are considered as those that feature a highest defined score (Cambouropoulos 1998). Such selecting mechanism is a classical and efficient way to extract important knowledge. It should be remarked, however, that this global selection, although enabling a quick characterization of a piece, infringes a thorough understanding of the complete score. We would like to retrieve also little detail at particular places, that may be of high relevance, and that may be taken into account by an active listening. The only necessary condition for a pattern to be considered as pertinent is that its score (here a degree of activation) has to exceed a certain minimal threshold. Therefore, to pattern *selection* we would prefer the concept of pattern *detection.*

As the result of such analysis is very complex, an interface should enable a browsing inside the analysis space or across the score. As such interface is not available now, our current implementation has to feature selecting operations.

Finally, since the process of pattern discovery proceeds itself through explicit characterization, there is no need to characterize *a posteriori* the patterns that have been discovered.

3 Data Representation

3.1 Pattern Class And Occurrence

The fact that several sequences are considered as similar in a certain sense means that they all belong to a same abstraction, which may be considered as a pattern class. These sequences are therefore occurrences of the pattern class. In this way, any new sequence sharing the same similarity will simply be considered as a new occurrence of this pattern class. The pattern class is not represented by a single prototype, but by all its occurrences that are effectively linked to it.

3.2 Pattern Class Chain

According to the incremental characteristic of music perception, patterns are progressively discovered, interval by interval, from initial interval to whole pattern. Pattern classes have to be represented following this cognitive constraint. In particular, all possible prefixes of a pattern may be considered as a pattern, uncompleted indeed, but still a pattern. Therefore, prefix of pattern classes are pattern classes. The set of all prefixes of a pattern class, ordered by growing order of pattern length, is a chain, called pattern class chain (PCC).

Progressively, for each new note of the pattern that is being discovered, a new pattern class is added as a continuation of the previous pattern class. In this way, if a prefix of a discovered pattern class appears later, it will simply be considered as a new occurrence of the intermediary pattern class associated to this pattern.

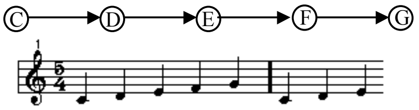

Fig. 3. A PCC, an occurrence of the whole PCC (first bar), and an occurrence of one prefix (second bar).

3.3 Pattern Occurrence Chain

For each pattern occurrence, an additional interface should associate the sequence of notes inside the score that constitutes the occurrence with the pattern class. Such interface may also be described as a chain — called pattern occurrence chain — where each successive state within the POC represents at the same time a note in the sequence and a state in the PCC.

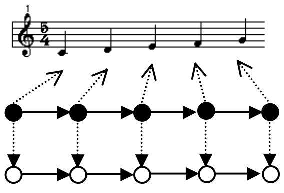

Fig. 4. The POC (black circles) interfaces notes in the score with the corresponding PCC (white circles).

3.4 Pattern Associations

The idea of segmentation may implicitly and dangerously suggest that score features only one level of pattern representation. On the contrary, patterns of different lengths may coexist and there may be inclusion or intersection relationships between them.

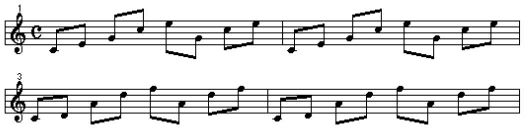

Fig. 5. Beginning of Bach's *Prelude* in *C*.

In figure 5, the main 8-note pattern is itself structured into an exact repetition of 3-note patterns. Each occurrence of the 8-note pattern, though constantly varied, carefully verifies the inner repetition of two sub-patterns. And each 8-note pattern is itself repeated twice.

Thus pattern cannot simply be characterized through an enumeration of similar intervals. The inner description as explained below should be made explicit too, and should be inferred by the machine.

We propose to represent such relationship between pattern and sub-pattern as follows. If occurrences of a pattern class feature a particular sub-pattern, a new POC, representing this sub-pattern, is linked to the PCC of the pattern itself. With such linking inside PCCs, a new association network is build between pattern classes. This high-level organization may help the recognition of basic pattern occurrences. In this way, expectations are generated by the system during the analysis: when a new occurrence of the pattern is discovered, sub-patterns are also *expected* (Meyer 1956).

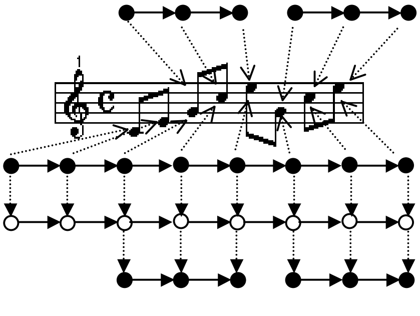

Fig. 6. On the first bar of Bach's Prelude may be build a POC (below the score) that is associated to the 8-note PCC (white circles), and two POC for the 3-note PCC (over the score). These 3-note patterns, since they are repeated on different 8-note patterns, are represented directly on the 8-note PCC with two additional POCs (at the bottom).

3.5 Pattern Repetition

If a pattern is repeated successively several times, occurrences are themselves elementary objects that forms a meta-pattern. In particular, local intervals may be considered between such successive occurrences. Such a successive repetition of occurrences of a same pattern class may simply be represented by extending each pattern with the first note of the following pattern, and by associating to this added note the POC of this same pattern class.

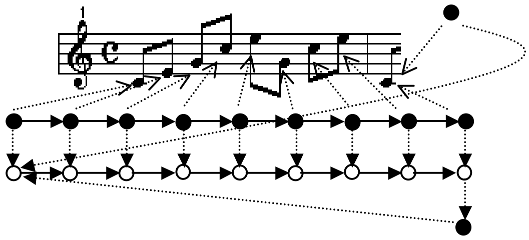

Fig. 6. When a pattern is repeated more than twice, the last note of the pattern is linked to the first note through an additional POC on the PCC.

Such a mechanism is not as arbitrary as it may appear. When perceiving such successive repetitions of occurrences of a same 8-note pattern, we actually perceive 9-note pattern, where last note is in the same time the first note of a new pattern. Thanks to this mechanism, every time we perceive a whole occurrence of the 9-note pattern, we then expect a new occurrence again.

4 Algorithms

4.1 Decomposing Polyphony Into Streams

In paragraph 2.2, we explained that a pattern should be considered as a sequence. This means that for each note of the pattern, its preceding note inside the pattern should be one of its *previous* notes, where *previous* is defined as following.

Let n be current note. Let $m1$, $m2$, … be the set of simultaneous notes that precedes n but in the same time are temporally closest to n. Let mi be the note that has the closest pitch compared to n. Then mi is a *previous note* of n.

Now if there exist past notes $p1$, $p2$, … whose pitch is *particularly closer* (defined below) to the pitch of n than any other note already known to be close to n, then the pi that is the temporally closest note is also a previous note. This last operation is repeated as long as necessary.

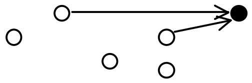

Fig. 7. Current note (in black) and past note (in white). Some past notes are defined as previous note (with an array).

Now the pitch of pi is *particularly closer* to the pitch of n than any other close notes when the quotient between the pitch distance between pi and n and the pitch distance between any close note and n is below a certain threshold.

In this way, a current note may accept several possible previous notes. Such interval between any previous note and current note, which is the building element of pattern construction, will be called *local interval*. Thanks to this decomposing of polyphony into streams, overlapping patterns may be discovered. See figure 10 for an example of this method.

4.2 Pattern Class Discovery

In this section, we will show how our system is able to detect new pattern classes, that is, new abstractions. As told previously, a pattern is defined as an approximately (or exactly) repeated sequence. So pattern will be discovered only if a similarity relationship is inferred between a current sequence and a past one. Past sequence has to be *recalled* because of its similarity with current sequence. The trouble is: current sequence does not already exist as a sequence if repetition itself is not already detected.

In our previous work (Lartillot 2002b), we alleviated the task by imposing a constraint, which can be expressed as follows: for a new pattern repetition to be detected, the repetition of each single interval of the patterns has to be explicitly and progressively discovered. In particular, the similarity between the first interval of each patterns has to be inferred before inferring the similarity of the remaining of the pattern. The trouble is, such a constraint can hardly be satisfied. Nevertheless, we will show first how we implemented our previous approach. A generalization of this algorithm will then be proposed, that can overcome previous limitation.

First Approach. First, every local interval has to be memorized in an associative memory that is able to retrieve any interval similar to a query. For this purpose, a hash-table associates for each interval parameter the set of its occurrences within the score. Now if the hash-table shows a similarity between current local interval *i1* and an old local interval *i1'*, a new pattern class is inferred (unless already discovered) associated to this single interval. Then if there exists any similarity between an interval *i2* that follows previous local interval *i1*, and an interval *i2'* that follows previous old local interval *i1'*, then a new pattern class extends previous pattern class. And so on.

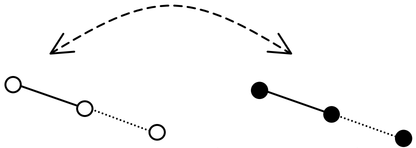

Fig. 8. In first approach, similarity of single intervals (solid lines) has to be inferred. This similarity can be extended to following intervals (dotted lines).

If *i1* and *i1'* have to be identical for being considered as similar, then pattern featuring a slight distortion on its first interval will not be detected. Therefore, a looser comparison between *i1* and *i1'* should be tolerated. But in this case, lots of non-pertinent little patterns will be inferred too. Moreover, with such approach it is not possible to detect patterns with different speed, since *i1* and *i1'* should have similar interonset value.

Second Approach. We propose to improve our first approach as follows. If current interval *i1* is particularly similar to an old local interval *i1'*, then a pattern class is inferred as previously. If, on the contrary, this similarity is not very high, previous local intervals *i0* that precede *i1* are considered, and compared to previous local intervals *i0'* that precede *i1'*. If the sequence *i0-i1* is considered as similar to the sequence *i0'-i1'*, then a pattern class is inferred, that consist of this succession of two intervals. In this way, a pattern may be detected even if its first interval was not a sufficient clue.

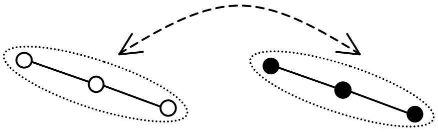

Fig. 9. In second approach, similarity may be inferred between sequences of intervals.

Now such approach may be immediately generalized to *n* intervals instead of 2.

Pattern Class Extension. Once a new pattern class has been discovered, its extension is an easier task. Indeed, the new local interval that extends the discovered new pattern just have to be compared to possible continuations of the discovered old pattern, instead of comparing it to all possible intervals through the hash-table. Indeed, thanks to the previous pattern class initiation, two or more similar contexts have been discovered in the score. Pattern extension just consists of a deeper analysis of found contexts.

4.3 Pattern Occurrence Discovery

The discovery of a new occurrence of an already discovered pattern class follows the two steps of Pattern Class Discovery — namely: pattern initiation and pattern extension. But this time, no pattern class have to be inferred.

5 Results

This model has been implemented as a library of *Open Music*, a musical representation software developed at Ircam. This new library called *OMkanthus* is able to proceed to analysis of MIDI files. In version 0.1, these results are displayed as a list of texts that is not easy to understand. That is why this library is provided with some basic tools for selecting and displaying longest patterns, most frequent patterns, or most pertinent patterns, where pertinence is a product of length and frequency. Here are some results.

When asking the pattern classes achieving the highest degree of pertinence in the beginning of a piano reduction of the *Allegro* of Beethoven's *Fifth Symphony*, we obtain all occurrences of the 4-note pattern (line #4 of figure 11). The prefixes of this pattern are also represented in this figure in lines #1 to #3. This additional information, stemming from intermediate inference during the progressive pattern perception and therefore not pertinent as analytical results, should not therefore be made explicit. This bad behavior is not due to our analytical system, however, but to the roughness of the simple tools we have designed for result selection for display.

When asking the longest pattern, we obtain this time 12-note pattern that consists of repetition of three occurrences of the 4-note pattern. Pattern found line #1 of figure 12 is very pertinent, whereas pattern found line #2 should not have been inferred. Here is one bad behavior of current version 0.1 of our algorithm.

When asking the pattern classes achieving the highest degree of pertinence in the beginning of the melody of the *Rondo* of Mozart's *Sonata* in *A* major K331 "*Alla Turca*", we obtain the 5-note pattern, line #2 of figure 13, and the long theme itself, line #1. More precisely, the long theme stops just when chords appears, simply because version 0.1 of *OMkanthus* cannot discover chords. Here also, very short patterns (like simple interval or chord) are displayed, showing the imperfections of our selection method.

When asking the pattern classes achieving the highest degree of pertinence in the beginning of Bach's *Prelude* in *C*, BWV 846, we obtain the 8-note pattern (line #4 of figure 14), the 16-note pattern (which consists in fact of the repetition of two 8-note patterns, line #6), and 2-note patterns that are repeated inside the 8-note pattern itself. When considering the different occurrences of these 2-note patterns, we notice that lots of non-pertinent occurrences are found in the end of the example. This comes from a heuristic stating that the more a pattern is repeated, the more approximate patterns may be tolerated. Such heuristic shows here its limitation.

6 Conclusion and Future Works

In this paper, we have proposed a new approach that attempt to answer to some fundamental problems arisen by current researches in MPD, through the elaboration of a new general computational model founded on explicit principles and aiming at mimicking human capacities for music perception and understanding.

The proposed model and implementation are still in an early phase, showing numerous limitations. With such approach, many questions arise but few are answered yet. Nevertheless, one interest of this general methodology is to propose a framework in order to make explicit some assumptions shared by MPD community.

It would be impossible to list all the difficulties that have to be solved now. Maybe the hardest part is to make these difficulties explicit. Some further improvements include chord pattern discovery, comparison of sub-patterns associated to a pattern (inferring the similarity between sub-pattern themselves, comparing the relative pitch and temporal distance between sub-patterns).

Then an interface has to be designed, enabling a browsing inside the score and the discovered structures. In a long term, such approach may try to go beyond pattern and catch higher-level concepts. Would a system be able to retrieve music theory?

Acknowledgments

My doctorate project is supervised by Emmanuel Saint-James (LIP6, Paris 6) and Gérard Assayag (Musical Representation Team, Ircam). Some ideas arise from stimulating discussion with my colleague Benoit Meudic.

References

Cambouropoulos, E. 1998. "Towards a General Computational Theory of Musical Structure." Ph. D. thesis, University of Edinburgh.

Conklin, D. and C. Anagnostopoulou. 2001. "Representation and Discovery of Mutiple Viewpoint Patterns." *Proceedings of the International Computer Music Conference*. International Computer Music Association, pp. 479-485.

Dannenberg, R. 2002. "Listening to "Naima": An Automated Structural Analysis of Music from Recorded Audio". *Proceedings of the International Computer Music Conference*. International Computer Music Association, pp. 28-34.

Hofstadter, D. 2002. Personal communication.

Lartillot, O. 2002a. "Musical Analysis by Computer Following Cognitive Model of Induction of Analogies." *Proceedings of the International Computer Music Conference*. International Computer Music Association, pp. 20-27.

Lartillot, O. 2002b. "Generalized Musical Pattern Discovery by Analogy from Local Viewpoints." *Discovery Science*. Berlin: Springer-Verlag.

Lartillot, O. 2003. "Perception-Based Advanced Description of Abstract Musical Content." *Digital Media Processing for Multimedia Interactive Services*. Singapore: World Scientific Press, to appear.

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

Meyer, L. 1956. *Emotion and Meaning in Music*. The University of Chicago Press.

Nattiez, J.-J. 1990. *Music and discourse: towards a semiology of music*. Princeton: Princeton University Press.

Rolland, P. Y. 1999. "Discovering Patterns in Musical Sequences." *Journal of New Music Research* 28(4):334-350.

Wiggins, G., K. Lemström and D. Meredith. 2002. "SIA(M)ESE: An Algorithm for Transposition Invariant, Polyphonic Content-Based Music Retrieval." *Proceedings of the International Conference on Music Information Retrieval*. Ircam-Centre Pompidou, pp. 283-284.

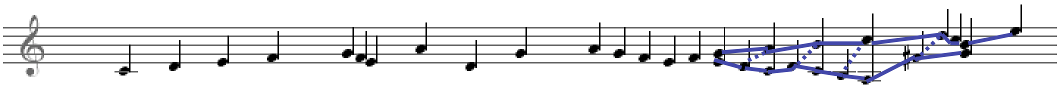

Fig. 10. Local relationships between notes in the beginning of the *Fugue* in *C* Major, BWV 846.

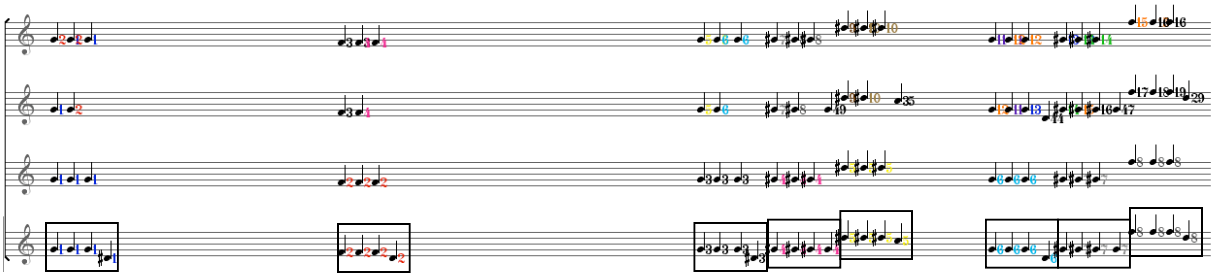

Fig. 11. Most pertinent patterns of the beginning of piano reduction of the *Allegro* of Beethoven's *Fifth Symphony*.

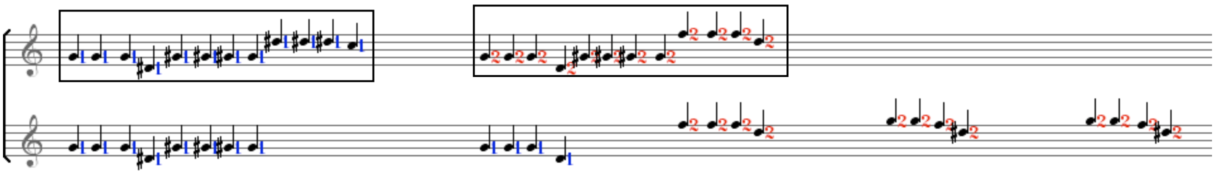

Fig. 12. Longest patterns of the beginning of piano reduction of the *Allegro* of Beethoven's *Fifth Symphony*.

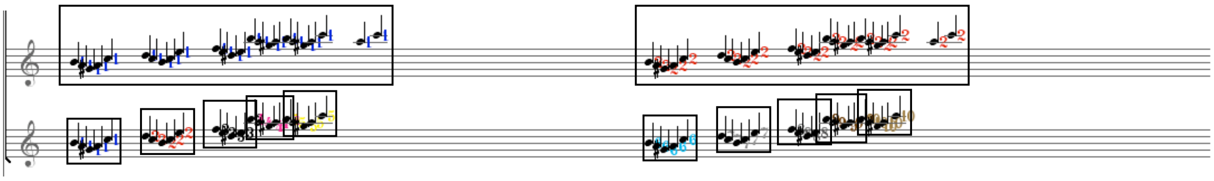

Fig. 13. Most pertinent patterns of the beginning of the melody of the *Rondo* of Mozart's *Sonata* in *A* major K331 "*Alla Turca*".

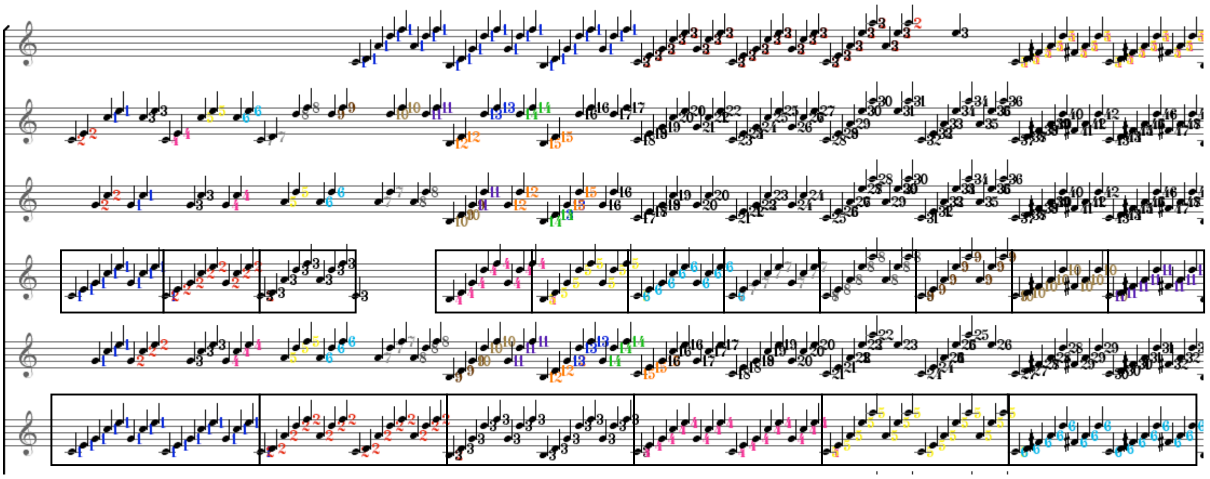

Fig. 14. Most pertinent patterns of the beginning of *Prelude* in *C Major*, BWV 846.