PERCEPTION-BASED MUSICAL PATTERN DISCOVERY: WHY AND HOW

Olivier Lartillot

Ircam — Centre Georges-Pompidou
Musical Representations Team, Paris
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 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 [7]. 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:

2.1.1. Implicit knowledge

Pattern may result from implicit knowledge that cannot be obtained directly from the score, such as: expected phrase length or metric [8]. 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.

2.1.2. Local boundaries

Low-level structural properties of the musical surface may be obtained through local boundary detection [2]. 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.
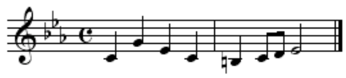
Fig. 1. A pattern may feature contrastive steps.

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.

2.1.3. 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.

This "repetition"-oriented criterion of pattern seems to remain the most relevant one, since music motives are classically defined in this way [10].

2.2. 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 [11] 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* [3] 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* [5] from rough phenomenon, even if risks have to be taken.

2.3. 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.

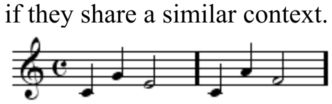

Fig. 2. The similarity of patterns cannot be measured outside of their actual context.

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 [4].

2.4. Selection

As many patterns may be found, pertinent patterns are considered as those that feature a highest defined score [2]. 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.

We propose therefore to model a pattern as a chain of states, where each successive state is associated to each successive note of the pattern. In this way, a pattern class is a chain of states — called pattern class chain (PCC) — where each state represents the shared characteristic of the associated note, and each state transition represents the shared characteristics of the associated transition between two successive notes.

Moreover, when a pattern class is being discovered, it is impossible to decide whether the discovery process is now achieved or whether, on the contrary, following notes — that have not been heard yet — will extend the pattern. That is why each prefix of the pattern class may be considered as a temporary PCC, until it is extended further.

3.3. Pattern Occurrence Chain

Each pattern occurrence is also a conceptual representation that interfaces the considered pattern class with the sequence of notes inside the score that constitutes the occurrence. Such interface may also be described as a chain — called pattern occurrence chain (POC) — where each successive state within the POC interfaces a note in the sequence with its corresponding state in the associated PCC.

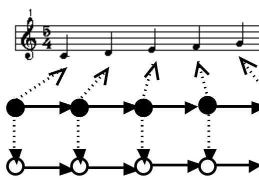

Fig. 3. The POC (black circles) interfaces notes in the score with the corresponding PCC (white circles).

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.
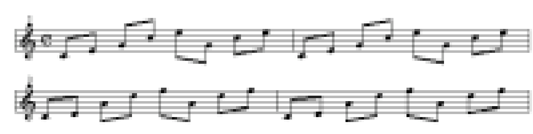
Fig. 4. Beginning of Bach's *Prelude* in *C*.

In figure 4, 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* [9].

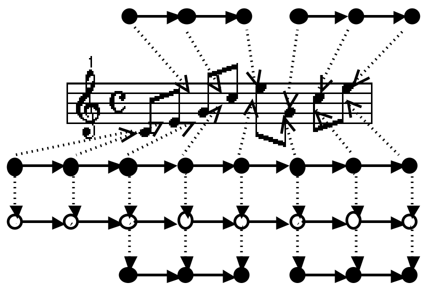

Fig. 5. First bar of Bach's *Prelude*, a POC (below the score) associated to the 8-note PCC (white circles), and two POC for the 3-note PCC (over the score). These 3-note patterns are represented directly on the 8-note PCC with two additional POCs (at the bottom).

3.5. Pattern Repetition

Music usually features successive repetitions of a same pattern class. If no new mechanisms were added, the system would consider each possible concatenation of these successive patterns as a new pattern. These inferences, not corresponding to human judgments and leading to combinatory explosion, should be forbidden. This fundamental issue has in fact never been taken into consideration.

It should be remarked that such pattern repetition is a special case of pattern association. If each pattern is extended with the first note of the succeeding pattern, then this last note of such extended pattern may be associated to the first note of the same pattern class. This means that, in the extended pattern class, the last state is linked to the first state. The idea of pattern cycling is therefore explicitly represented.

The first note of each new occurrence, as soon as it appears, is immediately associated to a new pattern occurrence chain. An additional mechanism prevents any pattern, whose first note is also the last note of another occurrence of the same pattern, to be extended further.

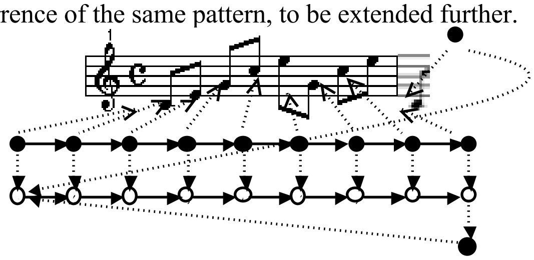

Fig. 6. When a pattern is repeated more than twice, the last note of the pattern is linked to the first note through an additional POC on the PCC.

Such a mechanism is not as arbitrary as it may appear. When perceiving such successive repetitions of occurrences of a same 8-note pattern, as soon as the last note of an occurrence is detected, the first note of a new occurrence is expected. This expected transition between these two notes may therefore be represented as an extension of the basic 8-note pattern into a 9-note pattern, where the last note is also the first note of a new occurrence.

4. ALGORITHMS

4.1. 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 [6], 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.

4.1.1. 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.

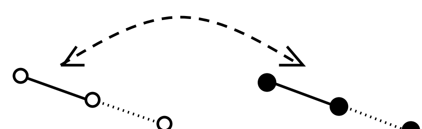

Fig. 7. In the first approach, similarity of single intervals (solid lines) has to be inferred. This similarity can be extended to following intervals (dotted lines).

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.

4.1.2. 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.

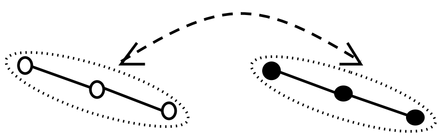

Fig. 8. In the second approach, similarity may be inferred between sequences of intervals.

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

4.1.3. 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.

4.2. 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.

4.2.1. Pattern Occurrence Initiation

When a similarity has been discovered between two different contexts, and before deciding to create any new pattern class, we have to make sure that such context does not already exists in the beginning of one of the set of all discovered pattern classes (or more simply to the set of pattern classes associated to the past context). If there does exist such pattern class, a new pattern occurrence simply associates the new discovered context with the retrieved pattern class.

4.2.2. Pattern Occurrence Extension

Since the beginning of currently discovered pattern occurrence is already associated to a pattern class, each of its successive candidate continuations may simply be compared to the successive continuations along the pattern class. In this case, current pattern occurrence does not need to be compared to old occurrences. The similarity threshold here is even greater than for pattern class discovery, since the expectation is explicitly represented within the pattern class.

4.3. Interval Distances

These local similarities are progressively built from single intervals. Distances are computed first between single intervals, then between succession of intervals, or patterns.

Let n_1, n_2, n_3, n_4 be four notes whose respective pitches are p_1, p_2, p_3, p_4 and respective durations d_1, d_2, d_3, d_4.

We propose, in a first approach, to formalize the perceptual distance between two intervals (n_1, n_2) and (n_3, n_4) as a weighted product of a pitch distance and a duration distance:

$$D((n_1, n_2), (n_3, n_4)) = (\text{abs}\,[(p_2 - p_1) - (p_4 - p_3)] + 1) \times (\max\,[d_1/\,d_3,\, d_3/\,d_1])^{0.7}$$

Here, only the duration of the first note of each interval is taken into consideration, since this duration is also the temporal distance between the successive notes. We may also remark that in music, pitches are subtracted, whereas duration are divided.

4.4. Priority Rules

It appears that the pattern class and pattern occurrence discovery routines have to be called in a very precise order. In particular, if the pattern class discovery step is systematically processed before the pattern occurrence discovery step, then new classes — that should have been in fact identified with already discovery classes — may be inferred. That is why new notes have to be compared to known pattern first.

Since previous notes have already been linked to POCs, such priority rules may be stated as follows : for each of these transitory POCs, first check whether the new note can be considered as its next step; else, check whether the new note can be considered as a new extension of the POC.

Moreover, for similar reasons, POCs have to be considered in a decreasing order of length.

5 RESULTS

This model has been implemented as a library of *Open Music* [1], 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 Bach's *Prelude in C*, we obtain the 8-note pattern (figure 9, first line) and 3-note patterns that are repeated inside the 8-note pattern itself (second line). Such interesting result, discovered by a system that has no access to any prior cultural knowledge, has never been obtained by any previous algorithm.

When asking the most pertinent pattern classes in the beginning of the *Fifth Symphony* by Beethoven, we obtain the 4-note pattern (figure 10a-b, second line) and the encapsulation of 3 times the 4-note patterns (figure 10b, third line). Note however that a non-pertinent occurrence has been found (dotted lined). Non-pertinent encapsulation of 2 times the 4-note patterns [first line] has also been considered.

When asking the most pertinent pattern classes in the beginning of the *Rondo alla turca* by Mozart, we obtain the 5-note pattern (figure 11, second line). The algorithm is not well prepared to face polyphonic pattern. That is why we observe some failures in pattern detection (figure 11, first and fourth lines). Finally, the long pattern (fifth line) pertinently encapsulates the repetitions of the 5-note pattern, but cannot then integrate the following chords, due to the polyphony limitations.

Such results may appear very simple and evident, compared to the complexity of the conceptual framework that has been developed for this purpose. But it has to be remarked that such results are very interesting for Artificial Intelligence researches, and also for the understanding of the cognitive mechanisms of music perception. Moreover, now that the algorithm performance starts corroborating with expecting behavior, a little refinement of the model should offer more subtle results. An important study has to focus on the optimization of the algorithms, as current version, easily entering into combinatory explosion, cannot handle more than several hundred of notes.

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).

There may be, inside patterns, "enclaves" of foreign notes not really belonging to these patterns. Patterns may also features transitory states, such as passing notes or appoggiatura. More generally, patterns may be included inside a polyphonic flow. If all this flow is represented like a single totally ordered sequence, patterns representations, here also, feature enclaves. Such problem has already been tackled [12], but uniquely for exact repetition. Chords should also be taken into consideration and patterns of chords should be discovered.

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?

5. 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. Up-to-date information about this project is available at: *www.ircam.fr/equipes/repmus/lartillot*

6. REFERENCES

[1] Assayag, G., C. Rueda, M. Laurson, C. Agon, and O. Delerue, 1999. "Computer Assisted Composition at Ircam : PatchWork and OpenMusic," *Computer Music Journal* 23(3):59-72.

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

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

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

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

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

[7] Lartillot, O. and E. Saint-James. to appear. "Automating Motivic Analysis Through Perceptual Rules." *Computing in Musicology* 13.
[8] Lerdahl, F. and R. Jackendoff. 1983. *A Generative Theory of Tonal Music*. Cambridge, Massachusetts: MIT Press.
[9] Meyer, L. 1956. *Emotion and Meaning in Music*. The University of Chicago Press.
[10] Nattiez, J.-J. 1990. *Music and discourse: towards a semiology of music*. Princeton: Princeton University Press.
[11] Rolland, P. Y. 1999. "Discovering Patterns in Musical Sequences." *Journal of New Music Research* 28(4):334-350.
[12] 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.

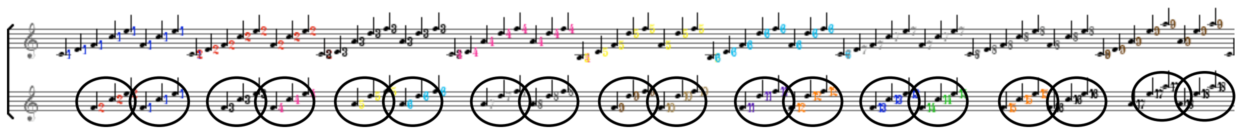

Fig. 9. Most pertinent patterns of the beginning of *Prelude* in *C* Major, BWV 846, by Bach.

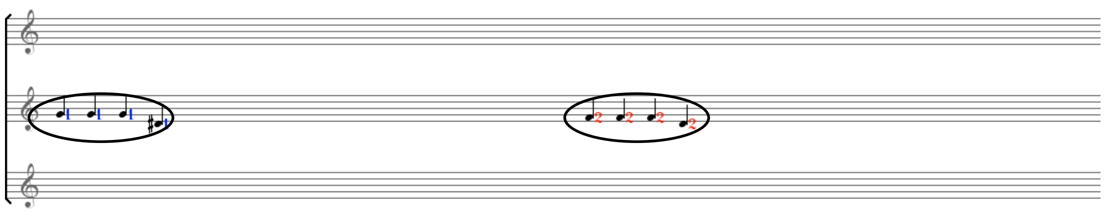

Fig. 10a. Most pertinent patterns of the beginning of the *Fifth Symphony* by Beethoven.

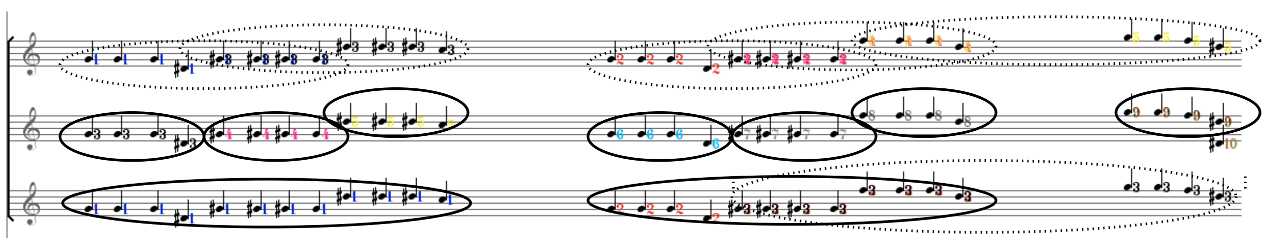

Fig. 10b. continued.

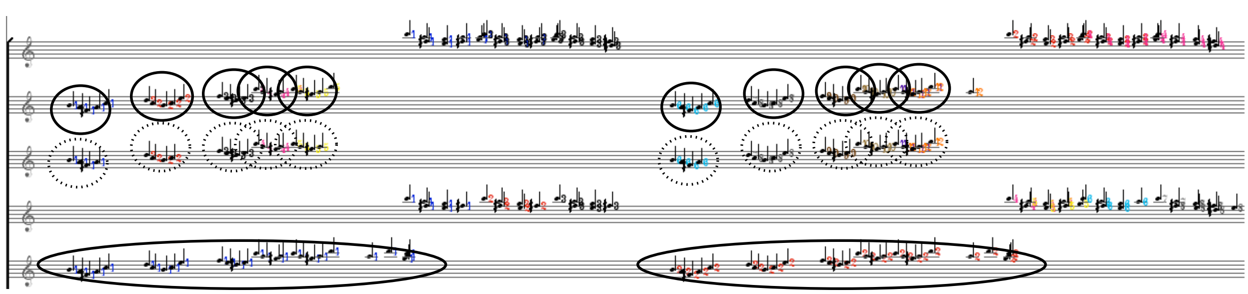

Fig. 11. Most pertinent patterns of the beginning of the *Rondo alla turca* by Mozart.