Musical Parallelism and Melodic Segmentation: A Computational Approach

Emilios Cambouropoulos
Aristotle University of Thessaloniki

Despite the consideration that musical parallelism is an important factor for musical segmentation, there have been relatively few systematic attempts to describe exactly how it affects grouping processes. The main problem is that musical parallelism itself is difficult to formalize. In this study, a computational model that extracts melodic patterns from a given melodic surface is presented. Following the assumption that the beginning and ending points of "significant" repeating musical patterns influence the segmentation of a musical surface, the discovered patterns are used as a means to determine probable segmentation points of the melody. "Significant" patterns are defined primarily in terms of frequency of occurrence and length of pattern. The special status of nonoverlapping, immediately repeating patterns is examined. All the discovered patterns merge into a single "pattern" segmentation profile that signifies points in the surface most likely to be perceived as points of segmentation. The effectiveness of the proposed melodic representations and algorithms is tested against a series of melodic surfaces illustrating both strengths and weaknesses of the approach.

Received March 27, 2004, accepted February 2, 2005

Music becomes intelligible to a great extent through self-reference, that is, through the relations of new musical passages to previously heard material. Structural repetition and similarity are crucial devices in establishing these relations. Similar musical entities are organized into musical categories including rhythmic and melodic motives, themes and variations, harmonic progression groups, and so on. However, musical similarity not only establishes relationships between different musical entities but also enables the definition of these entities by directly contributing to the segmentation of a musical surface into meaningful units.

Despite the importance of musical parallelism, even the most elaborate contemporary musical theories avoid tackling the problem of parallelism in a formal way. Theories that attempt to describe musical similarity systematically either restrict themselves to a well circumscribed and limited area of musical knowledge, for example, Ruwet's machine (Ruwet, 1987), or allow a fair amount of musical intuition to the analyst, for example, traditional thematic analysis (Reti's thematic processes; Reti, 1951), segmentation choices in pitch-class set theory (Forte, 1973), paradigmatic analysis (Nattiez, 1975, 1990). Lerdahl and Jackendoff (1983) acknowledge the importance of musical parallelism (parallelism rule GPR6) but admit that their "failure to flesh out the notion of parallelism is a serious gap in [their] attempt to formulate a fully explicit theory of musical understanding" (p. 53). Temperley, who has developed one of the most sophisticated computational models of musical cognition, admits that "despite the clear role of parallelism in meter, it would be very difficult to incorporate parallelism into a computational model. The program would have to search the music for patterns of melodic and rhythmic repetition. Since this seems to me a huge and complex problem, I am not addressing it formally in this book" (Temperley, 2001, p. 51). See, however, the next section for a proposal by Temperley and Bartlette (2002) that incorporates parallelism in a metric preference rule system.

Models of melodic segmentation are often based on local Gestalt-based factors that essentially identify points of local maximal change in various musical parameters, including IOIs (inter-onset intervals), pitch intervals, dynamic changes, and so on. Higher-level processes, however, play an important role as well. In this study, a central assumption is that similar musical patterns tend to be highlighted and perceived as units/wholes whose beginning and ending points influence the segmentation of a musical surface. The relation between musical parallelism and melodic segmentation is discussed more extensively in the section Segmentation and Parallelism.

The aim of this study is to examine the relationship between musical parallelism and segmentation via computational modeling. A computational model that

Music Perception volume 23, issue 3, pp. 249-267, issn 0730-7829, electronic issn 1533-8312

extracts melodic patterns from a given melodic surface is presented; following the assumption that the beginning and ending points of "significant" repeating musical patterns (primarily in terms of frequency of occurrence and length of pattern) influence the segmentation of a musical surface, the discovered patterns are used as a means to determine probable segmentation points of the melody. All the discovered patterns merge into a single "pattern" segmentation profile that signifies points in the surface most likely to be perceived as points of segmentation. The study focuses on a special type of repetition, referred to as formative repetition by D. Lidov (1979), that involves immediately repeating patterns that often diverge toward their endings, contain small variations, and may be transposed; the function of this type of repetition is to "form" motives and phrases.

This study does not provide a comprehensive standalone computer program for melodic segmentation; since the proposed model addresses only one specific segmentation factor (that relates to musical parallelism), testing it against a large melodic corpus without incorporating it first in a comprehensive segmentation model would be meaningless. The current study explores melodic surface representation issues and issues relating to the pattern extraction mechanism itself through the application of a series of different representations and algorithm variants on progressively "difficult" melodic parallelism examples. The main goal is neither to provide a comprehensive solution to the problem of melodic parallelism nor to simulate computationally the exact cognitive mechanisms involved, but rather to shed light on various aspects and to enable a better understanding of the problem.

Related Work on Pattern Extraction Techniques for Melodic Segmentation

Pattern-matching techniques have been employed in attempts to formalize musical similarity. Much of the research has focused on algorithms for comparing melodic sequences (i.e., finding the best possible alignment between two given melodic excerpts) or for melodic recognition (i.e., finding instances of a given melodic excerpt in a larger musical database). There have been, however, relatively few attempts to tackle the difficult issue of pattern extraction (i.e., extracting important patterns in one or more musical sequences). Overviews of the application of pattern-processing algorithms on musical strings can be found in Crawford et al. (1998), Rolland and Ganascia (1999), Cambouropoulos et al. (2001), and Meredith et al. (2002).

Several recent attempts to formalize pattern extraction and melodic segmentation are presented below. All these models are relevant for segmentation tasks in that they discover important musical patterns; however, only the last two models address melodic segmentation explicitly.[1]

Meredith et al. (2002) present an algorithm for discovering repeated patterns in multidimensional representations of polyphonic music. The proposed algorithm computes all the maximal repeated patterns in a multidimensional data set (e.g., all the maximal repeated patterns in a two-dimensional representation of polyphonic music where one axis represents time and the other pitch). The authors maintain that maximal repeated patterns tend to be musically important; however, they acknowledge that the algorithm discovers too many such patterns and that mechanisms for selecting a smaller set of salient patterns is necessary. (They propose some possible mechanisms but admit that further research is required to restrain the abundance of extracted patterns.) A handful of musical examples are chosen to show the potential of the algorithm.

Rolland (1999, 2001) introduces an approximate pattern extraction model that identifies all melodic passage pairs that are significantly similar (a similarity threshold is set in advance), then extracts actual patterns in terms of a set of instances that includes a prototype, and finally orders these patterns according to a prominence value based on factors such as frequency of occurrence and pattern length. The heavy combinatorial computation required is carried out in a computationally economic fashion using dynamic programming concepts. The model has been tested on a corpus of jazz melodies.

A computational model for melodic parallelism that affects the determination of metrical structure is introduced by Temperley and Bartlette (2002). This model calculates the "goodness" of beat intervals (i.e., time spans between beats) in terms of parallelism; this goodness value contributes, via the parallelism rule, to finding a preferred metrical structure. The model calculates "parallelism" values for all the possible pitch interval pairs in a melody (adjacent or further apart); these values depend primarily on whether the intervals are the same (diatonic intervals) or have the same or a different

[1] An interesting model that investigates melodic segmentation, parallelism, and metrical structure by Ahlbäck (2004) was published too late to be included in this study. Among others, the model has a component that performs "a segmentation of the melody by analysis of melodic parallelism and structural discontinuity" (p. 20).

contour. From these values, parallelism scores are computed for beat pairs that reflect the extent to which events in the vicinity of the first beat are "paralleled" in the vicinity of the second beat; these scores are used in the parallelism rule of the metrical structure preference rule system. It should be noted that the proposed model does not explicitly identify patterns; neither does it provide a segmentation of the melodic surface.

Ferrand and Nelson (2003) propose a memory-based model for melodic segmentation. Different classes of Markov models are used for acquiring melodic regularities and for determining the probabilities of sequences of symbols. The main assumption is that "segmentation boundaries are likely to occur close to accentuated changes in entropy" (p. 142), that is, points in the melody where the predictability associated with the occurrence of a musical event changes abruptly from low to high or from high to low (these points tend to coincide with the limits of recurring patterns). The proposed model "learns" from raw musical data, that is, it does not require a training data set with segmentation points annotated. The model is applied to Debussy's *Syrinx*, and the results are compared to empirical segmentation data for the same piece.

A different memory-based approach for melodic segmentation is presented by Bod (2002), which requires an annotated melodic data set in which segmentation boundaries have been manually identified in advance. By using the frequencies of occurrence of melodic fragments encountered in previous melodies, predictions are made for where boundaries might occur in new unseen melodies. The models presented by the author are tested against the Essen Folksong collection (training set of approximately 5,000 folk songs and test set of 1,000 folk songs) and yield more than 80% phrase detection accuracy.

Computational models presented above are not directly comparable as they have varying scopes and give special attention to different facets of musical parallelism and/or segmentation. Below are a few general comments that relate to the approach taken in this study:

1. Some models extract directly significant musical patterns from raw unstructured musical material, which may be considered a strength in that such models can be applied directly on large data sets of readily available music (e.g., MIDI files). However, it may be cognitively more plausible that some preprocessing of musical data is required for the pattern-processing mechanisms to be more efficient. For instance, with polyphonic music it is plausible that a listener organizes the musical surface into streams before—or at least concurrently with—discovering patterns. It is unlikely that patterns distributed across different streams can be perceived at all (Bregman, 1990).
2. Extracting patterns that embody drastic variations (e.g., ornamented or reduced patterns) directly from the musical surface is a task hampered with many difficulties. For instance, how much tolerance should be allowed for the approximate matching process? Where are the exact boundaries of two patterns that match approximately (i.e., how do we know that extra notes beyond the boundaries are not part of the pattern)? In addition to these concerns, computational complexity is also greater for approximate pattern-processing techniques. It seems more plausible that, first, simple pattern extraction may contribute to melodic segmentation and, second, more sophisticated pattern-matching techniques may be applied to the segmented surface.
3. Representation of the musical surface is a very important issue. Are diatonic intervals sufficient (many of the aforementioned models use this representation)? Should more abstract representations be employed, for instance, a step-leap representation? Should time patterns be taken into account? If yes, should IOIs be used? Or ratios? Or even more abstract representations? (A musical representation for pattern extraction tasks is proposed in the multiple-viewpoint representation by Conklin & Witten, 1995; Conklin & Anagnostopoulou, 2001.)
4. The use of previously learned musical schemata is clearly an important factor for segmentation (e.g., cadential schemata). However, the emergence of *melodic* patterns (e.g., themes, motives, etc.) is primarily linked to the unique structure of a particular musical piece. Linguistic-oriented approaches that attempt to learn patterns from previously seen pieces for predicting boundaries in new pieces seem less appropriate for music (except perhaps for musical corpora where there is strong inter-opus coherence). These approaches can be attractive for intra-opus applications (with the only reservation that very small data sets are not ideal for statistical approaches).
5. Evaluation of computational models for musical parallelism and/or segmentation is a difficult issue, as there exist no significant, authoritative, annotated data sets against which models can be tested. This problem hampers attempts to compare models against each other. Researchers use different musical data sets for evaluation. Test data sets are often small (sometimes just a handful of examples), but in this

case, a detailed qualitative evaluation is possible (at the expense of having selected biased data). At other times, large data sets are used, but quantitative results are difficult to judge (for instance, what does it mean that an algorithm extracted 500 significant patterns from a data set that contains thousands of notes? How many of these patterns are musically significant? How many significant patterns have been missed altogether?).

The approach in this article attempts to address some of these problems, but in no case does it provide a comprehensive solution. A computational model is proposed as a means to explore parallelism in relation to melodic segmentation. The proposed model cannot be tested on how well it performs melodic segmentation in general, nor can it be directly compared to other models as it is not a complete model for segmentation. (It has to be incorporated into a broader model of melodic segmentation.) Some possible advantages of the proposed algorithm are that it is simple to implement, fast in terms of computational complexity, and easy to experiment with (parameters can be altered and different melodic surface representations can be tried). Open questions for further investigation are discussed in the last section.

The proposed algorithm is applied to a small set of melodic examples with the following characteristics:

1. Boundaries due to parallelism are unambiguously defined (i.e., hardly any musician/music analyst would disagree on where the “correct” boundaries are).
2. Local Gestalt-based boundary detection models fail to identify these boundaries.
3. The examples illustrate progressively difficult yet clear pattern-matching problems.

It is easy to find many counterexamples for which the proposed model would fail (see the example in the last section). However, clearly presenting the strengths and limitations of a certain approach may contribute to a better understanding of the problem and lead to new, more robust and sophisticated models.

There is hardly any work of empirical research that directly examines the influence of parallelism on segmentation. Early work by Deutsch (1980) that bears on parallelism has shown that it was easier for listeners to learn and remember patterned melodies (which consisted of repeating three-note or four-note patterns) than unpatterned ones. Sloboda (1985) suggests that memory ability can be improved if items to be remembered can be linked or related together: “In music, such relations are, to a large extent, already present in the patterning and structure of a composition . . . [among others] economy of coding is achieved if repetitions can be identified and noted” (Sloboda, 1985, p. 190). The role of parallelism on memory will be discussed further in the section Segmentation and Parallelism.

The principle of similarity/difference underlies perceptual tasks including musical segmentation and categorization. For segmentation, it has been shown that cues at the musical surface, such as changes in register, timbre, dynamics, tempo, and so on, play a primary role in perceiving boundaries in both tonal and especially nontonal music (Lamont & Dibben, 2001; Lalitte et al., 2004; see also the overview for experimental work in local detail grouping factors—Frankland & Cohen, 2004). For categorization, structural similarity at the musical surface and/or reductions of it has been shown to influence the formation of motivic/thematic categories, especially through repeated hearings of the musical material (Pollard-Gott, 1983; Deliège, 1996, 2001). Other research suggests, however, that surface similarity, rather than “deeper” structural similarity, is the primary factor in categorization tasks (Lamont & Dibben, 2001; McAdams et al., 2004).

Some researchers acknowledge the importance of musical parallelism in segmentation tasks (for instance, Clarke & Krumhansl, 1990, identify the reiteration of musical material already heard as one of four characteristics contributing to segmentation in an experiment involving the perception of musical form). But this topic has not been examined in any detail in experimental studies. After presenting a recent detailed study that involved the quantification of Lerdahl and Jackendoff’s local grouping rules, Frankland and Cohen (2004) admit that “the current work could not be extended to Symmetry (GPR5) and Parallelism (GPR6) because these rules are not clearly defined,” and they assert that the lack of an explicit description of parallelism is unfortunate because “it is mainly Symmetry and Parallelism that serve as a link between the low-level rules (i.e., GPRs 2, 3) and the high-level analyses (i.e., Time-Span reduction and Prolongation Reduction)” (Frankland & Cohen, 2004, p. 538). The aim of the current study is to formalize aspects of parallelism that contribute to melodic segmentation so that a fully formalized theory of musical parallelism may become possible.

In the following sections, the issue of pattern extraction is first discussed, and an efficient pattern extraction algorithm is explained. Then the relationship between musical similarity and segmentation is examined, and a model that segments a melodic surface based on melodic pattern extraction is presented. Finally, a series of further improvements on the current model is suggested. Throughout the study several melodic examples illustrate the strengths and weaknesses of the overall approach. The current study is a continuation of the earlier research presented in Cambouropoulos (1998).

Pattern Extraction

Musical entities that constitute a musical pattern are often structured hierarchically, that is, some notes (or chords, etc.) are more prominent than others in metrical position, duration length, register, harmony, tonal hierarchies, and so on. What kind of pattern-processing techniques are most adequate for establishing similarities between structured strings like melodic passages?

To simplify for the sake of argument, we can suppose two main approaches:

1. Approximate pattern-processing techniques applied to the unstructured musical surface
2. Exact pattern-processing techniques applied to the musical surface and on a number of reduced versions that consist of structurally more prominent components

The first approach is based on the assumption that musical segments construed as being parallel (similar) will have *some* of their component elements identical (e.g., two instances of a melodic motive will have a "significant" amount of common notes or intervals but not necessarily all)—some approximate pattern-matching algorithms based on this approach are described in Bloch and Dannenberg (1985), Cope (1990), Stammen and Pennycook (1993), and Rolland (1999, 2001). The second approach is based on the assumption that parallel musical segments are necessarily identical in at least one parametric profile of the surface or reduction of it (e.g., two instances of a melodic motive will share an identical parametric profile at the surface or some higher level of abstraction, for instance, a pattern of metrically strong or tonally important notes/intervals and so on). Computational techniques based on this approach are described in Conklin and Anagnostopoulou (2001, 2006), Cambouropoulos (1998), and Hiraga (1997) see also technique proposed by Lartillot (2004) that allows extraction of mixed parametric patterns.

An exact pattern extraction algorithm will be presented below. It will be maintained that exact pattern-matching techniques at the musical surface (or a *slightly* reduced version of it) are sufficient for melodic *segmentation* tasks, which will be discussed in more detail in the Segmentation and Parallelism section.

An Exact Pattern Extraction Algorithm

An efficient algorithm that computes *all* the repetitions in a given string is described in Crochemore (1981); see also the description by Iliopoulos et al. (1996)—an informal description of Crochemore's algorithm is given in Appendix 1. For a given string of symbols (simple or complex), the matching process starts with the smallest pattern length (one element) and ends when the largest pattern match is found. This algorithm takes $O(n \cdot \log n)$ time where n is the length of the string; this is the fastest algorithm possible. This algorithm can be applied to as many parametric profiles considered necessary (e.g., pitch intervals, contour, durations, inter-onset intervals, dynamic intervals, implied harmony) for the melodic surface and/or reductions of it.

Selection Function

It is apparent that a procedure for the discovery of all identical melodic patterns for many melodic parametric strings will produce a great number of possible patterns, many of which would be considered counterintuitive and nonpertinent by a musician/analyst.

Rowe attaches a strength value to each pattern depending on its frequency of occurrence: "Each known pattern has an associated strength: the strength is an indication of the frequency with which the pattern has been encountered in recent invocations of the program" (Rowe, 1993, p. 248). Frequency of occurrence and pattern length, two properties of pattern significance, are balanced in the pattern score procedure proposed by Conklin and Anagnostopoulou (2001).

In line with the procedure proposed by Cambouropoulos (1998), a prominence value is attached to each of the discovered patterns based on the following factors: (a) prefer most frequently occurring patterns, (b) prefer longer patterns, (c) avoid overlapping. A *selection function* that calculates a numerical strength value for a single pattern according to these principles can be devised, for instance:

$$f(L, F, DOL) = L^a \cdot F^b / 10^{c \cdot DOL}$$

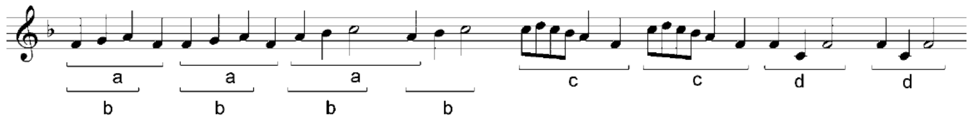

FIG. 1. *Frère Jacques*—most prominent pitch patterns extracted by the exact pattern induction algorithm and the *selection function* (applied only to the diatonic pitch profile).

where L = pattern length; F = frequency of occurrence for one pattern; DOL = degree of overlapping;[2] a, b, c = constants that give different prominence to the above principles (the following values have been used: $a = 1$, $b = 2$, $c = 3$).

For every pattern discovered by the above pattern induction algorithm, a value is calculated by the selection function. The patterns that score highest should be the most significant (see Figure 1).

Segmentation and Parallelism

Segmentation of a musical surface is a central part of musical analysis; an initial selected segmentation can seriously affect subsequent analysis as a great number of intersegment musical structures are excluded a priori. The most commonly acknowledged (and perhaps most prominent) factors in musical segmentation relate to the perception of local discontinuities of the surface (e.g., a longer note between shorter ones or larger pitch interval between smaller intervals, etc.); one such model is the *Local Boundary Detection Model* (*LBDM*) proposed by Cambouropoulos (1998, 2001a)—see brief description in Appendix 2. Higher-level processes, however, also affect the segmentation of a musical surface. Perhaps the most important of these higher-level mechanisms is *musical similarity*, that is, similar musical patterns tend to be highlighted and perceived as units/wholes whose beginning and ending points influence the segmentation of a musical surface. For instance, a model for determining local boundaries would select the interval between the third and fourth notes of *Frère Jacques* as a local boundary (larger pitch interval between smaller ones), whereas a boundary between the fourth and fifth notes appears because of melodic repetition.

[2]DOL is defined as the number of elements shared by some patterns divided by the number of all elements in those patterns, or more precisely: $DOL = (T-U)/U$, where T is the total number of elements in all the instances discovered for a pattern ($T = F \cdot L$), and U is the number of elements in the union set of all the instances discovered for a pattern (this definition allows DOL to be in some cases greater than 100%).

General Assumptions

This study's focus is primarily a special case of melodic similarity, namely immediate repetition of melodic passages. These repeating passages often diverge toward their endings and contain small variations, and the repeated passage may be transposed. David Lidov (1979) calls this kind of repetition *formative repetition*. Its function is to establish or to "form" motives and phrases. This study assumes that it involves fundamental pattern discovery processes primarily at the melodic surface (not reductions of the surface) and is essentially independent of more abstract learned idiom-specific schemata (e.g., harmony, tonality, meter). This kind of melodic similarity is omnipresent in music.

From a cognitive point of view, elaborate pattern extraction processes are more likely to be applied to relatively short melodic excerpts due to the heavy computation involved. This activity is usually more intense at the beginning of a musical piece/section where new musical materials are introduced and established. Once a number of such musical ideas have been extracted, links to further new instances (varied or not) can be made more efficiently: once a pattern has been extracted from a local context, it is placed in long-term memory (i.e., it is learned); when the pattern is encountered again, later in the musical surface, it is recognized and used for further parsing of the surface.

The proposal here is that pattern extraction takes place primarily within a short temporal window, and it assists chunking the melodic input into meaningful units, thus expanding the storage capacity of short-term memory. Repetition expands the mnemonic capabilities of short-term memory (7 ± 2 different elements proposed by Miller, 1956) in the sense that more elements/chunks can be held by short-term memory. According to Snyder, "a pattern that fits within the time limits of short-term memory can have repetition of elements, and hence have more actual events than seven, perhaps up to a limit of 25" (Snyder, 2000, p. 50). In this sense, we can imagine a short temporal window sliding over the sequence of musical events; pattern extraction algorithms enable repeated patterns to be found within this window, which, in turn, assist with the segmentation and efficient encoding of the musical surface. The size of the window can reach the limit of the perceptual present (up to 10–12 seconds; see Snyder, 2000, p. 50) or even become as long as 30 seconds, according to Levinson's idea of "quasi-hearing" (Levinson, 1997). In this study, the pattern discovery algorithm is applied to short melodic sequences that can be considered to fit into the short perceptual window suggested above; the possibility of applying this algorithm to longer sequences is discussed further in the last section.

Pattern similarity assists metrical induction (e.g., Temperley & Bartlette, 2002). However, meter assists musical segmentation (e.g., Temperley, 2001), which enables further pattern processing of the segments. It is asserted in this study that pattern induction contributes significantly to the establishment of metrical structure by means of segmentation, especially at the beginning of a musical work or section where new musical material is introduced. Once meter is established it can assist further segmentation of the musical surface (assuming that metrical and grouping structures are coextensive; Lerdahl & Jackendoff, 1983). See the last section for more on the relation between meter and parallelism.

It is assumed that similarity processes for melodic segmentation tasks are confined essentially to the melodic surface in contrast to melodic categorization tasks (i.e., creating motivic/thematic categories *after* segments have been defined), which require similarity measurements at deeper levels of musical structure as well (see Cambouropoulos & Widmer, 2000; Cambouropoulos, 2001b, for a computational model of melodic categorization). Because extracting patterns at reduced versions of the melodic surface would result in ambiguous segmentations, as it would not be possible to define exactly where the boundaries of the repeated patterns should be placed (since notes are missing from the reduced version). This problem, in some sense, defeats the point of using pattern extraction at reduced versions of the surface for melodic segmentation. Of course, musical similarity appears in many guises at deeper levels of musical structure, but in these cases this sort of abstract similarity is not the most crucial factor in segmentation tasks; other factors, like Gestalt-based local boundary detection factors or learned schemata (e.g., harmonic cadences), are responsible for segmenting the surface and only then are more sophisticated comparisons of segments made possible at more abstract levels of description.

The present musical examples for testing the proposed algorithms have been selected because the segmentation process for these cases relies primarily on melodic parallelism and not on local detail grouping factors (local Gestalt-based factors provide clearly incorrect boundaries). These two segmentation components (i.e., local Gestalt-based factors and parallelism) commonly reinforce each other, but for the sake of clarity, examples that illustrate a conflict between the two approaches and a clear predominance of the parallelism factor have been selected. Also noteworthy, these melodic figures represent the *melodic surface* that is presented as input to the algorithms. It is assumed that the melodic surface does not include explicit metric information (i.e., the listener does not have direct access to such information); to stress this point, bar lines have been omitted from all examples.

In this study, the pattern extraction algorithm is applied to parametric profiles of the melodic surface for pitch intervals (diatonic intervals, a step-leap representation, and some further, more refined representations) and for inter-onset intervals (IOI ratios). One significant objective is to discover which of these parameters (or combination of them) is more appropriate for the segmentation task and to show how a "balanced" representation that is neither too specific nor too general may yield better results in more cases. However, the issue of representation is examined primarily to show its importance and how better representations can be devised rather than to propose a "best" solution.

The PAT Algorithm

The pattern extraction model described in the section Pattern Extraction, which consists of the exact pattern extraction algorithm and selection function, provides a means of discovering "significant" melodic patterns. There is still a need for further processing leading to a "good" description of the surface (in terms of exhaustiveness, economy, simplicity, etc.). It is likely that some instances of the selected pitch patterns should be dropped, or a combination of patterns that rate slightly lower than the top rating patterns may give a better description of the musical surface (for instance, in Figure 1, each pitch pattern, *a* or *b*, cannot explain the melodic structure—some instances of each of these patterns should be dropped and a combination of the two selected, namely *a-a-b-b*).

To overcome this problem, a simple method has been devised (see Table 1).

In the melodic example of *Frère Jacques* (Figure 2), the pattern boundary strength profile (*PAT*) has been calculated by applying the pattern extraction model to the diatonic pitch interval profile: notice the strong pattern boundaries at the points indicated by asterisks where no local boundaries are detected by *LBDM* or other local detail grouping models.

TABLE 1. The *PAT* Algorithm—construction of the pattern boundary strength profile.

A pattern extraction algorithm is applied to one (or more) parametric sequences of the melodic surface as required. No pattern is disregarded, but each pattern (both the beginning and ending of pattern) contributes to each possible boundary of the melodic sequence by a value that is proportional to its selection function value. That is, for each point in the melodic surface all the patterns are found that have one of their edges falling at that point and all their selection function values are summed. This way a pattern boundary strength profile is created (normalized from 0 to 1). It is hypothesized that points in the surface for which local maxima appear are more likely to be perceived as boundaries because of musical similarity.

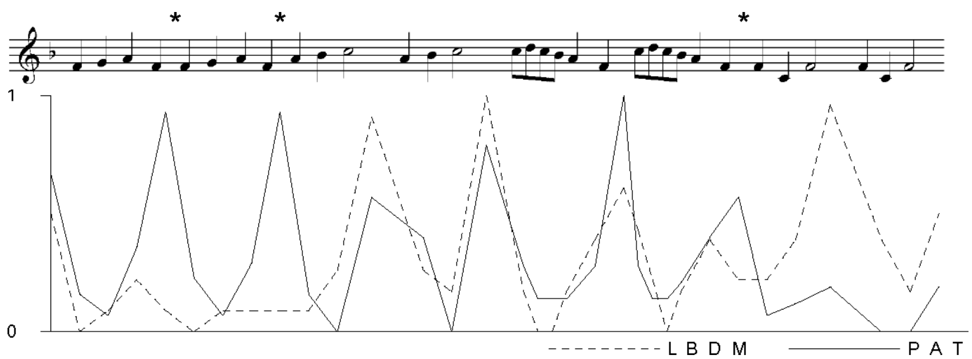

FIG. 2. *Frère Jacques*—Segmentation profile according to the Local Boundary Detection Model (*LBDM*) and the Pattern Boundary Detection Model (*PAT*) for the *diatonic pitch interval profile*; local maxima indicate positions that may be considered as points of segmentation (NB: strong pattern boundaries are detected at the points indicated by asterisks where no local boundaries are discovered by *LBDM*).

The PAT Algorithm (Revised)

The above example consists only of exact full repetitions, although it is not a usual case. A frequently encountered situation occurs when two patterns diverge toward their ends (see examples in Figures 3, 4, and 5). Lerdahl and Jackendoff have incorporated this intuition in their parallelism grouping preference rule GPR6. This rule "says specifically that parallel passages should be analyzed as forming parallel parts of groups than entire groups. It is stated in this way to deal with the common situation in which groups begin in parallel fashion and diverge somewhere in the middle, often in order for the second group to make a cadential formula. (More rarely, parallelism occurs at ends of groups.)" (Lerdahl & Jackendoff, 1983, p. 51). Ahlbäck maintains that "grouping by similarity is start-oriented, since similarity in a temporal context is recognized through recurrence; repetition of what is already heard which promotes identification by start" (Ahlbäck, 2004, p. 251). Empirical research by Deliège (2001) supports the claim that beginnings of patterns play a special role in pattern recognition: "Pattern recognition was thus made on the basis of this very beginning [of melodic sequences], and subjects did not pay attention to what happened afterward" (Deliège, 2001, p. 400).

In general, the beginning of melodic patterns is paramount in discovering parallel passages. This intuition has been incorporated into the current model by making a very simple modification to the method described in Table 1: *only the beginnings of patterns contribute to the strength of the pattern boundary profile.*

In the examples of Figures 3, 4, and 5, the revised *PAT* model detects correctly the beginning of the repeated phrases. (The initial *PAT* model inserts spurious peaks at the endings of the exactly repeating parts of the phrases.) For the *Chorale St. Antoni* it should be noted that the repeated phrases are five (i.e., 3 + 2) bars long, which is very unusual; Lerdahl and Jackendoff (1983, p. 206) take this five-bar grouping structure for granted (no systematic procedure for detecting it is given), but the revised *PAT* algorithm correctly identifies the beginning of the second phrase.

Representation of the Melodic Surface

The pattern boundary detection model, as described to this point, can discover repeating patterns in the diatonic pitch interval domain that may or may not diverge toward their endings (patterns may be transposed). What happens if some intervals are not exactly the same (as, for instance, the first intervals of the repeating

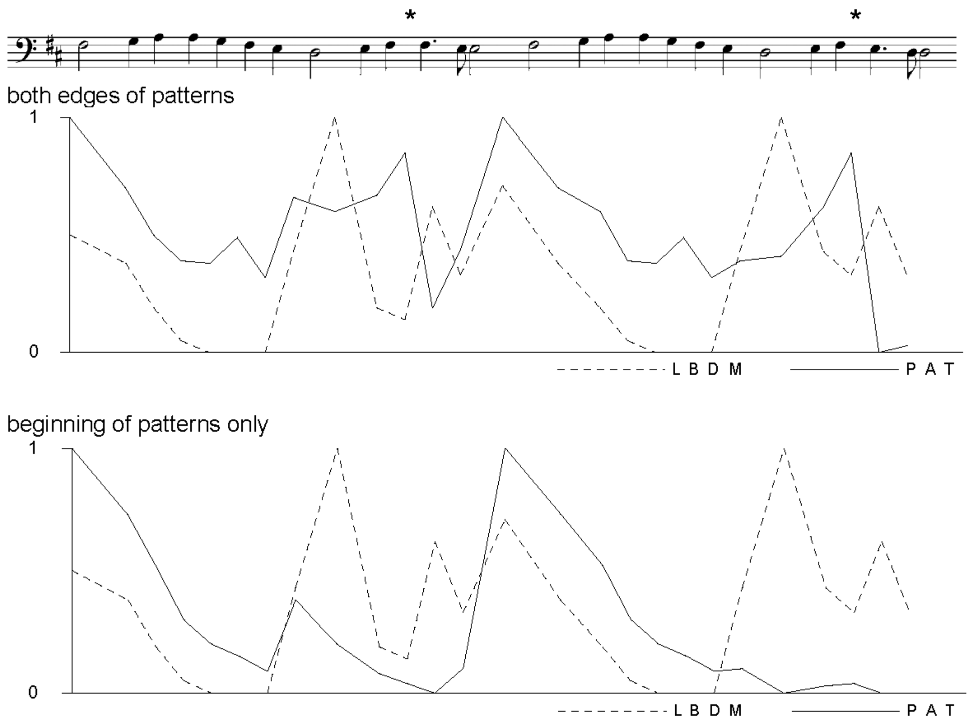

FIG. 3. Beginning of the finale theme from Beethoven's Ninth Symphony. Segmentation profile according to the Local Boundary Detection Model (*LBDM*) and the Pattern Boundary Detection Model (*PAT*) for the diatonic pitch interval profile. The strong pattern boundaries that indicate the end points of the exactly repeating parts of the two phrases (indicated by asterisks) are eliminated in the version of the model that takes into account only the beginnings of patterns.

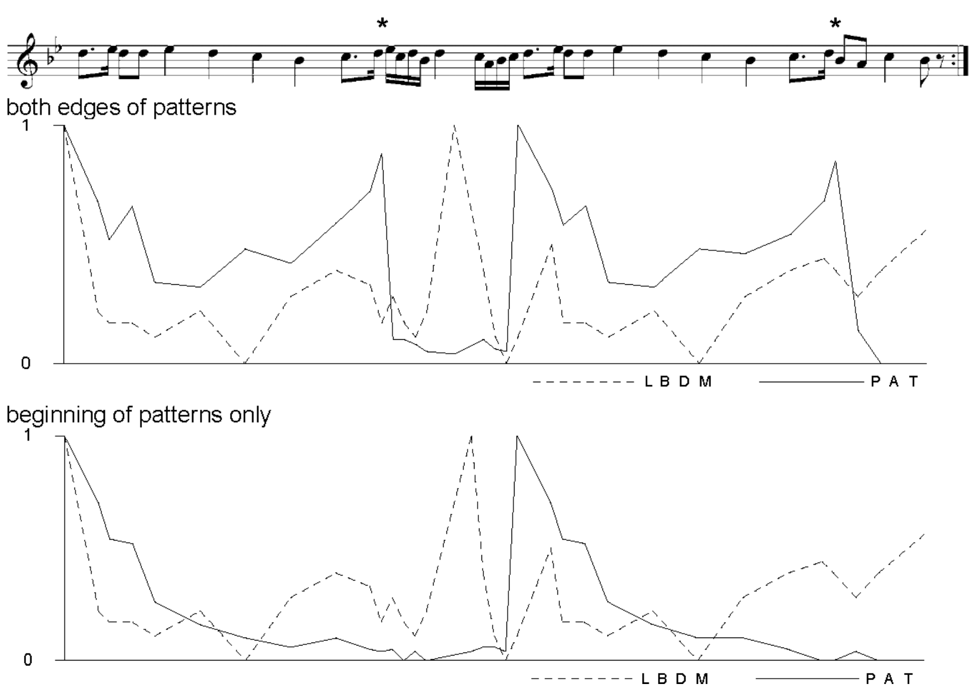

FIG. 4. *Chorale St. Antoni* (arranged by Brahms in his "Haydn Variations," op. 56). Segmentation profile according to *LBDM* and the Pattern Boundary Detection Model (*PAT*) for the diatonic pitch interval profile. NB: the strong pattern boundaries that indicate the end points of the exactly repeating parts of the two phrases (indicated by asterisks) are eliminated in the version of the model that takes into account only the beginnings of patterns.

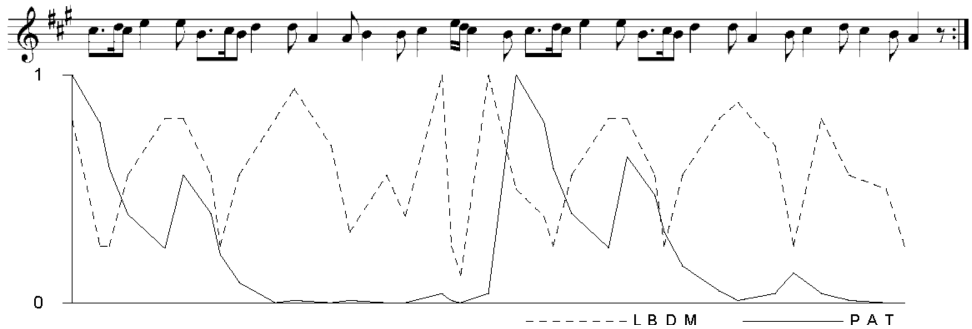

FIG. 5. Opening melody of Mozart's A-Major Sonata, K. 331. Segmentation profile according to *LBDM* and the Pattern Boundary Detection Model (*PAT*) for the diatonic pitch interval profile (beginning of patterns only). The *PAT* model correctly detects the beginning of the repeated phrase (*LBDM* fails) and also indicates the beginnings of the smaller one-bar length motives.

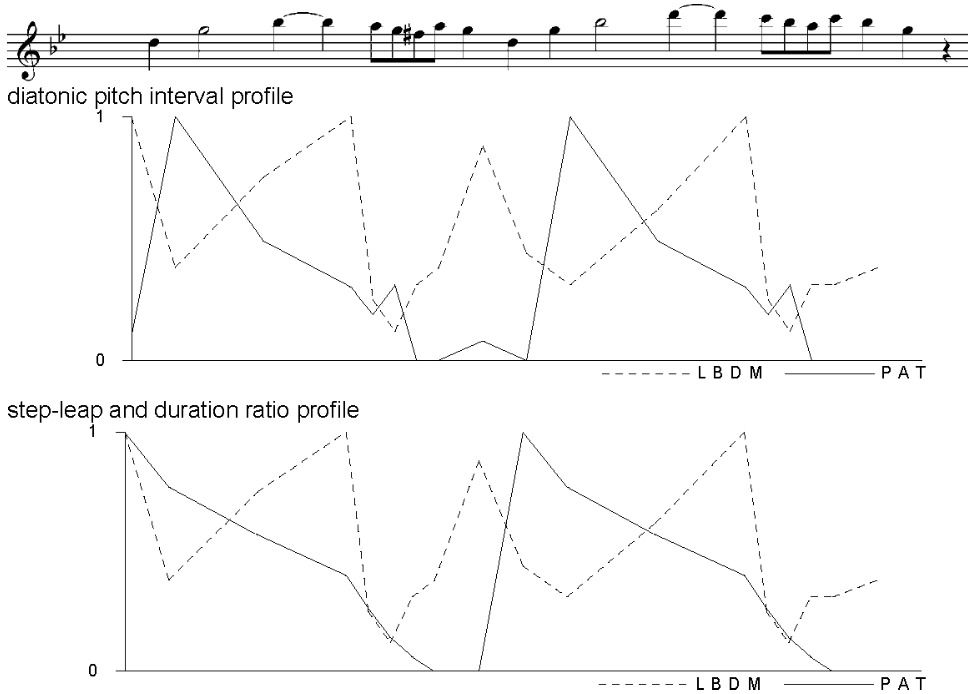

FIG. 6. Theme from Mozart's G-Minor Symphony, K. 550, movement III. Segmentation profile according to *LBDM* and the Pattern Boundary Detection Model (*PAT*), first, for the diatonic pitch interval profile and, second, for the combined step-leap and duration ratio profile. The diatonic pitch interval matching fails as the first interval of the repeating phrase is a third interval rather than a fourth interval. The combined step-leap and duration ratio encoding enables the correct segmentation of the two phrases; local boundaries are not capable of providing a correct segmentation.

phrases in Figure 6)? How can rhythmic information also be taken into account?

A more abstract representation for pitch intervals may be useful, such as a step-leap profile, especially if coupled with duration information. The step-leap encoding consists of five distinct symbols (+step, +leap, -step, -leap, same)—a rather too limited alphabet. If it is combined with duration symbols (or duration ratios), the alphabet becomes rich enough to capture all the necessary information so that the pattern boundary detection model may operate effectively. In this encoding, each interval of a melody is represented as a tuple (step-leap interval, duration ratio). This further adjustment to the model enables it to segment correctly more difficult cases as those in Figures 6 and 7, giving correct results also for the previous examples presented in this article.

A Variant of the PAT Algorithm for Further Flexibility

As mentioned above, approximation can be introduced into an exact pattern-matching process by using a more abstract representation at the level of the initial string of

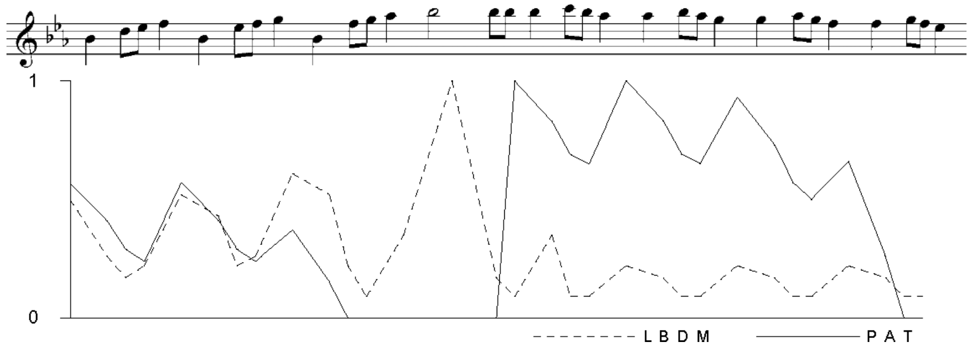

FIG. 7. Opening melody of Chopin's Waltz, op. 18. Segmentation profile according to *LBDM* and the Pattern Boundary Detection Model (*PAT*) for the combined step-leap and duration ratio profile. *PAT* correctly finds the motivic structure of this melody, especially in the second half where local detection models are not successful.

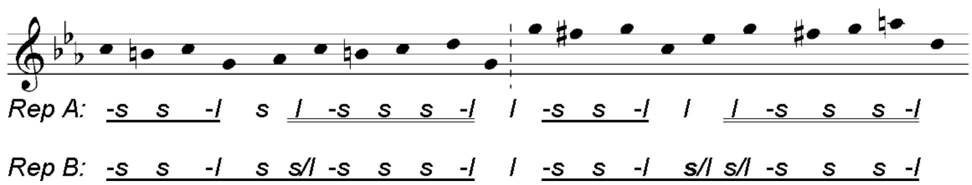

FIG. 8. The *step-leap* representation allows the extraction of two patterns repeating twice each (single- and double-line underlined patterns in representation A). The proposed representation that allows overlapping of pitch categories—in this case, a third interval can be a member of either *step* or *leap* (*s/l*)—allows the matching of the second half of the sequence to the first half (see representation B).

symbols. For instance, a pitch interval representation like the *step/leap* representation (or even *step/small-leap/medium-leap/large-leap*, etc.) allows different size leaps to be matched. One problem, however, is that the abstract categories in the representation have sharp boundaries, and no instance may belong to more than one category; this way, borderline members can never be matched to other "similar" members of other categories (e.g., a third interval as a member of *leap* can never be matched to a second interval, which is a *step*).

Consider, for instance, the sequence of pitch intervals in Figure 8. The *step-leap* representation allows the extraction of the two different underlined patterns (see representation A in Figure 8). A musician, however, would consider the second half of the sequence as a (near-exact) repetition of the first half (the pitches of this example are taken from Bach's *Well-Tempered Clavier*, Book I, Fugue in D# Minor; see Figure 12). This match can be achieved only if the first third interval in the second half of the pitch sequence can be matched with the corresponding second interval of the first half.

An abstract symbolic representation can become more flexible in terms of category gradedness and membership if instances are allowed to be members of more than one category. In the following examples, a third interval is allowed to be an instance of either *step* or *leap* (*s/l*)—see representation B in Figure 8. The alternative abstraction (step or leap) that allows the longest patterns to emerge is selected. (The first third interval of the melody's second half is taken to be a member of *step* and is thus matched to the corresponding second interval of the first half, as this gives a longer melodic repetition.)

The case where a second and a third interval should be considered similar is not simply a rare exception in music, but a common phenomenon, especially when themes appear in their dominant key (see, for instance, the tonal answers of almost half of Bach's fugue themes from the two books of the *Well-Tempered Clavier*). See Figures 10, 11, and 12 for selected examples (NB: Bach fugue themes and their tonal answers are presented as belonging to the same auditory stream; this is not musically correct but is cognitively plausible—a streaming algorithm could generate tentative streaming options including ones presented in the examples).

The problem set forth in this section can be solved by matching techniques that measure the distance between pitch numbers; however, in some cases the ability to use symbols rather than numbers is crucial to represent a musical sequence.

For the sake of testing the proposed more flexible representation on the examples of this study, the exact pattern-matching algorithm (Appendix 1) that extracts all repeating patterns was adjusted to cope with alternative

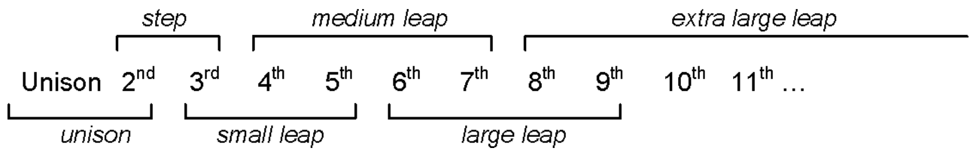

FIG. 9. A possible abstract representation for pitch intervals. In this representation overlap between categories is allowed; this is an arbitrary proposal to show the possibility of overlapping categories—further research is required to define a more cognitively plausible scheme (see proposal for seven partially overlapping classes by Lemström and Laine (1998)). Such a representation might be more powerful than the more standard step-leap or contour representations as it allows rather high discriminability between intervals and also significant flexibility. It was tested on all the examples in this article giving correct results.

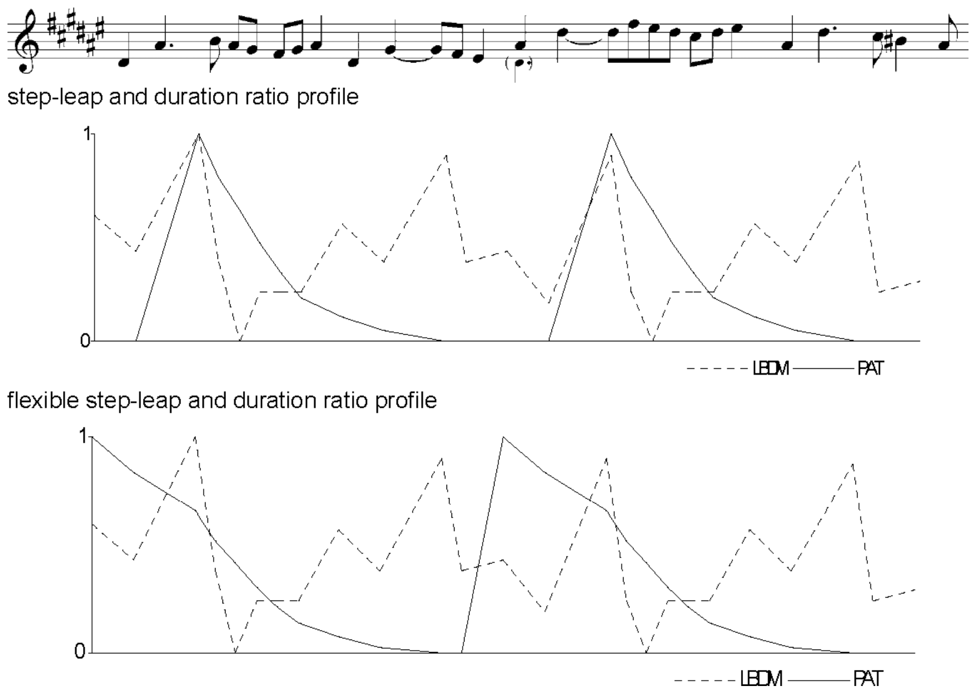

FIG. 10. Upper voice (theme and tonal answer as one melodic "stream") from the opening of Bach's *Well-Tempered Clavier*, Book I, Fugue in D# Minor. Segmentation profile according to *LBDM* and the Pattern Boundary Detection Model (*PAT*), first, for the combined step-leap and duration ratio profile and, second, for the same representation that allows additionally a third interval to be a member of either *step* or *leap* (in this case the repeated pattern is correctly identified).

symbols for elements of the initial string.[3] This variant is not efficient, but it gives correct results for the short test melodies here. An efficient algorithm for a similar pattern extraction problem has been recently developed using *don't care* symbols for elements that may belong to two categories—for example, a * symbol signifies an upward step or leap (Cambouropoulos et al., 2005). Further research, however, is required to incorporate this efficient algorithm in the proposed model.

Examples of the application of the new version of the *PAT* algorithm are given in Figures 10, 11, and 12. This new version of the pattern extraction algorithm makes it possible to adopt more sophisticated representations of the melodic surface that allow overlapping among abstract categories (e.g., the third interval being either *step* or *leap*, or a more "sophisticated" pitch interval representation like that shown in Figure 9).

Additional Examples

The *PAT* algorithm was tested against the empirical data obtained in a segmentation experiment conducted by Koniari et al. (2001). In the specific experiment (one of the two segmentation experiments in this study),

[3]The main difference between the algorithm variant implemented here and the algorithm described by Crochemore (1981) is that at each level of the algorithm (i.e., for the start-sets corresponding to the different lengths of patterns) start-sets that are subsets of other larger start-sets have to be deleted.

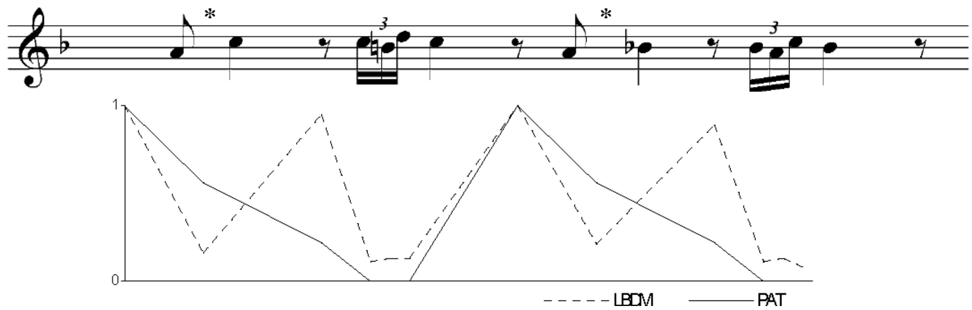

FIG. 11. Opening melody from Beethoven's Piano Sonata, op. 10, no. 2. Segmentation profile according to *LBDM* and the Pattern Boundary Detection Model (*PAT*) variant for the combined step-leap and duration ratio profile that allows additionally a third interval to be a member of either *step* or *leap* (in this case the repeated pattern is correctly identified—the two intervals indicated by the asterisks are matched).

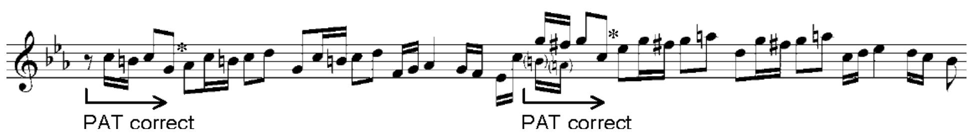

FIG. 12. Upper-voice "stream" (theme and tonal answer) from the opening of Bach's *Well-Tempered Clavier*, Book I, Fugue in C Minor. The Pattern Boundary Detection Model (*PAT*) variant correctly detects the beginning of the repetition (tonal answer) for the combined step-leap and duration ratio profile that allows additionally a third interval to be a member of either *step* or *leap* (NB: the two intervals indicated by the asterisks are matched).

FIG. 13. Rondo, finale from the Sonatina No. 2 in C Major, by Anton Diabelli; numbers indicate the segmentation points presented in Koniari et al. (2001).

children listened to the rondo from the finale of the Sonatina No. 2 in C Major by Anton Diabelli (Figure 13) and indicated positions of punctuation (referred to as "segmentations") by pressing the space bar of a keyboard; a familiarization factor was introduced by allowing one group of children to listen to the piece one time and another group three times before doing the task. The results show that a maximum of 14 segmentations were given for this piece, but all of these were not necessarily marked by each listener. (White columns in Figure 14 indicate the average number of segmentations given by children musicians and nonmusicians for the two different familiarization conditions—the average values have been normalized from 0 to 1 to be comparable to the output values of the *PAT* model.) "It is worth noting that all the segmentations that were recorded corresponded to the main articulations of the piece, as they would appear in a classical morphological analysis: that is, as ends of musical phrases and motifs" (Koniari et al., 2001, p. 313).

Despite the fact that Koniari et al. do not explicitly focus on the role parallelism plays in segmentation, it is apparent that repetitions and variations clearly contribute to the understanding of the musical work and to the way listeners segment it. (Expressive performance also plays a role—one wonders if listeners would give the same segmentations while listening to a mechanical performance without any articulations.)

The melody of Diabelli's Rondo—only the sequence of notes—was given as input to the *PAT* algorithm (the accompaniment was omitted as the algorithm can be applied only to melodies). The algorithm produced 14 peaks that coincide with the listeners' segmentations (the algorithm gives one additional strong segmentation point at the very beginning of the piece but misses the boundary at the end of the piece as it accounts for only the beginnings of patterns)—see black columns in Figure 14. Not only are all the "correct" boundaries discovered by the model but the main segmentation positions at the middle of the Rondo (segmentation 7 in Figure 14) and its subperiods (segmentations 4 and 11) come out relatively stronger in accordance with the experimental results and the morphological analysis. The strength values of the segmentations, however, are quite different from the experimental values, especially in relation to the smaller phrases and subphrases; this is partly due to the fact that the algorithm does not explicitly account for musical symmetry and hierarchy.

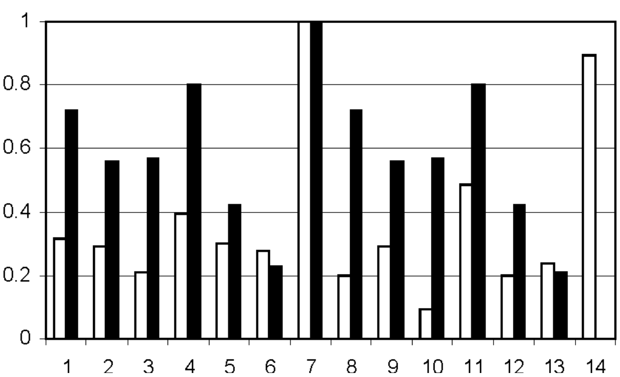

FIG. 14. Segmentation results on a Rondo by Diabelli, by children-listeners (white columns) and by the *PAT* algorithm (black columns)—segmentation points are indicated in the horizontal axis and segmentation "strengths" in the vertical axis (see text for details).

As mentioned earlier, parallelism affects metrical structure, and the reverse. What happens, however, if a piece of music does not have metrical structure? The *PAT* algorithm has been applied to the opening melody of Mussorgsky's *Pictures at an Exhibition*, Promenade (Figure 15). One can see that the exact repetition determines a clear boundary in the middle of this melodic excerpt; the *PAT* algorithm identifies it correctly. In this case, the melody has a clear tactus but not a higher-level regular metrical structure (see Figure 16). Segmentation models that rely on metrical structure would have a problem in this and other cases of nonmetrical music. The relationship between segmentation, parallelism, and metrical structure requires further investigation (see also the next section).

Further Improvements and Conclusion

In this study, the computational attempt for capturing melodic similarity with a view to achieving melodic segmentation is still a long way from providing a robust,

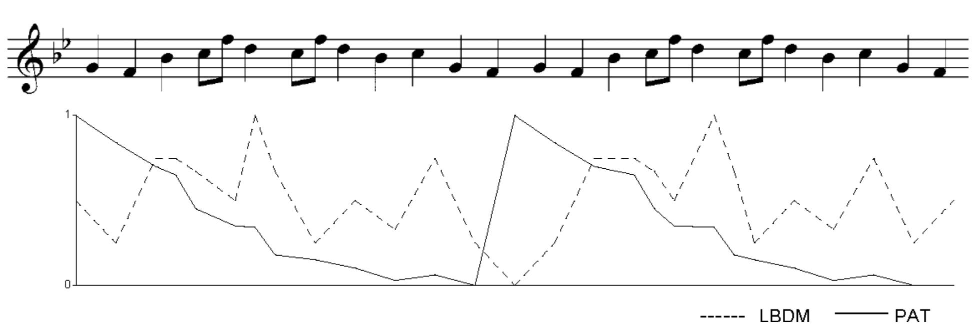

FIG. 15. The opening melody of Mussorgsky's *Pictures at an Exhibition*, Promenade. The Pattern Boundary Detection Model (*PAT*) correctly detects the beginning of the repetition.

FIG. 16. The opening melody of Mussorgsky's *Pictures at an Exhibition*, Promenade. Score including bar lines and time signature indications as notated by the composer.

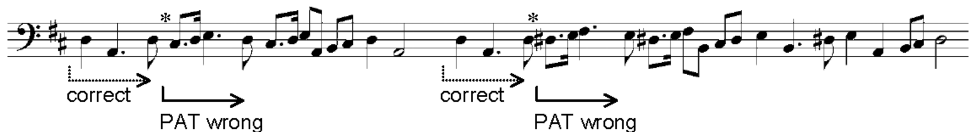

FIG. 17. Theme from Schubert's Symphony in B Minor, "Unfinished," D. 759. None of the versions of the Pattern Boundary Detection Model (*PAT*) described in this article can correctly detect the beginning of the repetition, as the intervals, indicated by asterisks, have different directions (ascending–descending).

flexible, and general model of melodic parallelism. Yet the model shows potential; further research is necessary to improve the model and to evaluate it on a much larger scale.

Further investigation is required for finding the most adequate way(s) to represent melodies so that patterns can best be extracted. I have proposed that a linked step-leap interval and duration ratio representation is better than either representation alone or the most commonly used diatonic pitch interval representation. The step-leap representation was enhanced by allowing overlapping between the *step* and *leap* categories; other more general representations than the diatonic pitch interval representation and more specific ones than the step-leap representation are possible (e.g., see Figure 9). Such representations are also possible for the duration/IOI domain, which requires further exploration. Additionally, some limited reduction of the surface may be necessary, such as consolidation of repeating notes. A good representation is paramount in devising pattern extraction models that are more general and that can cope with a larger number of cases. However, there will always be cases for which a representation is inadequate. See, for instance, in Figure 17 the example for which the proposed representation is not appropriate—there is a mismatch in interval direction at the points indicated by asterisks. (An approximate pattern-matching algorithm, however, can cope with this case.) A single representation and a single pattern extraction algorithm will probably never be sufficient for all cases; a combination of representations and algorithms may be required. Yet it is interesting to take a certain methodology to its limits to discover its shortcomings.

The boundaries discovered by the pattern boundary detection model may complement the segmentation given by the model *LBDM* in defining a total boundary strength profile. The total boundary strength profile can be calculated as a weighted average of the local boundary and pattern boundary strength profiles even though more sophisticated methods for combining the two should be explored. The local maxima in the total boundary strength profile can be used as a guide for the final segmentation of the musical surface.

Can the proposed model be applied to long melodic sequences? The answer is positive in terms of the algorithm employed (there is no limit on the length of the input melodic sequence), prompting another question: would this be of value or at least useful? From a cognitive point of view, computationally intense pattern extraction processes are likely to be applied to relatively short melodic excerpts. (Extracted patterns can then be used in different more economic pattern-processing strategies.) In terms of formative repetition (i.e., immediate near-exact repetition), musical similarity is contained within relatively short melodic passages. Obviously, *PAT* can be applied to long melodic sequences using a shifting overlapping windowing technique whereby the analysis is done gradually for relatively short melodic fragments. Alternatively, if the algorithm is primarily aimed at modeling musical analytic tasks, the pattern extraction process can be applied to a long melodic sequence, but the selection process has to be modified to give additional emphasis on recency (i.e., immediacy of repetition). A pattern that repeats often in a musical piece does not necessarily imply more significance for melodic segmentation than an equal-length pattern that repeats just twice in immediate succession. (If the *PAT* model is applied to such a melodic sequence, the pattern that repeats twice would have very small boundary peaks compared to the one repeated many times.) Additional study is required to establish the most appropriate means for the proposed model's application on long melodic sequences.

Another study might implement an online version of the pattern extraction algorithm. (Crochemore's algorithm is inherently an off-line algorithm.) This dynamic algorithm would be closer to the way listeners perceive patterns as these build gradually during listening. However, one should note that a pattern-relating boundary can appear only in retrospect. That is, only after a repetition has started to unfold in time can one realize that its beginning appeared a few moments earlier; real-time segmentation based on the discovery of patterns is not possible. In this respect, the implementation of the above sliding window pattern extraction technique may be a relatively good candidate for the exploration of the cognitive processes that relate to perception of boundaries due to parallelism.

The metrical structure of a musical work can play an important role in establishing an overall final segmentation. Temperley (2001) explicitly incorporates in his metrical structure model a preference rule according to which strong beats are located near the beginnings of groups. In this sense, if metrical structure is known, segmentation points can be determined at parallel points of the metrical structure. (Almost every example in this study could be correctly segmented according to this rule.) However, here it is assumed that metrical structure is *not* known. It is hypothesized that, at the beginning of a musical work or at points where new musical materials are introduced, a listener attempts to segment the musical surface based on local detail grouping rules and by using pattern extraction methods (not metrical structure). Once this result is achieved, metrical structure can be induced and, in turn, can be used to facilitate further segmentation processes. A model of parallelism, as proposed here, cannot provide a final segmentation of a melody on its own. This model, however, can discover significant positions of strong pattern boundaries, especially at the beginning of a piece, which can assist in selecting a certain metrical structure; the induced metrical structure, in turn, can reinforce relatively weaker segmentation boundaries and assist in breaking down a melody into smaller groups.

Overall, the methods and results in this article provide information in an attempt to address the difficult issue of musical parallelism and its links to melodic segmentation. The examples against which the proposed algorithm was tested were known to pose serious problems for local detail grouping algorithms; additionally these examples contain increasing difficulties regarding melodic similarity. The proposed model is quite successful in tackling all of these problems, but it requires future experimentation and development for its integration in a comprehensive model of melodic segmentation.[4]

Author Note

Address correspondence to: Emilios Cambouropoulos, Department of Music Studies, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece. E-MAIL *emilios@mus.auth.gr*

References

AHLBÄCK, S. (2004). *Melody beyond notes: A study of melodic cognition*. Ph.D. thesis, Göteborgs Universitet, Sweden.

BLOCH, J. J., & DANNENBERG, R. B. (1985). Real-time computer accompaniment of keyboard performances. In *Proceedings of the International Computer Music Conference (ICMC85)* (pp. 232-235). San Francisco, CA.

BOD, R. (2002). Memory-based models of melodic analysis: Challenging the gestalt principles. *Journal of New Music Research, 31*, 27-36.

BREGMAN, A. S. (1990). *Auditory scene analysis*. Cambridge, MA: MIT Press.

CAMBOUROPOULOS, E. (1998). *Towards a general computational theory of musical structure*. Ph.D. thesis, University of Edinburgh, U.K. http://users.auth.gr/~emilios

CAMBOUROPOULOS, E. (2001a). The local boundary detection model (LBDM) and its application in the study of expressive timing. In *Proceedings of the International Computer Music Conference (ICMC01)* (pp. 232-235). Havana, Cuba.

CAMBOUROPOULOS, E. (2001b). Melodic cue abstraction, similarity and category formation: A formal model. *Music Perception, 18*, 347-370.

CAMBOUROPOULOS, E., CRAWFORD, T., & ILIOPOULOS, C. S. (2001). Pattern processing in melodic sequences: Challenges, caveats and prospects. *Computers in the Humanities, 35*, 9-21.

CAMBOUROPOULOS, E., CROCHEMORE, M., ILIOPOULOS, C. S., MOHAMED, M., & SAGOT, M.-F. (2005). A pattern extraction algorithm for abstract melodic representations that allow partial overlapping of intervallic categories. In *Proceedings of the International Symposium on Music Information Retrieval ISMIR 2005*. (pp. 167-174) Queen Mary, University of London.

CAMBOUROPOULOS, E., & WIDMER, G. (2000). Automated motivic analysis via melodic clustering. *Journal of New Music Research, 29*, 303-318.

CLARKE, E., & KRUMHANSL, C. L. (1990). Perceiving musical time. *Music Perception, 7*, 213-251.

CONKLIN, D., & ANAGNOSTOPOULOU, C. (2001). Representation and discovery of multiple viewpoint patterns. In *Proceedings of the International Computer Music Conference (ICMC01)* (pp. 479-485). Havana, Cuba.

CONKLIN, D., & ANAGNOSTOPOULOU, C. (2006). Segmental pattern discovery in music. *INFORMS Journal of Computing*, Vol. 18, No. 3 (pp. numbers forthcoming).

[4]I would like to thank Costas Tsougras for suggesting the melodic examples by J. S. Bach (Figures 10, 12) and Mussorgsky (Figure 16), as well as David Temperley and two anonymous reviewers for their very useful comments on an earlier version of this manuscript.

CONKLIN, D., & WITTEN, I. H. (1995). Multiple viewpoint systems for music prediction. *Journal of New Music Research, 24*, 51-73.

COPE, D. (1990). Pattern-matching as an engine for the computer simulation of musical style. In *Proceedings of the International Computer Music Conference* (pp. 288-289). Glasgow.

CRAWFORD, T., ILIOPOULOS, C. S., & RAMAN, R. (1998). String matching techniques for musical similarity and melodic recognition. *Computing in Musicology, 11*, 71-100.

CROCHEMORE, M. (1981). An optimal algorithm for computing the repetitions in a word. *Information Processing Letters, 12*, 244-250.

DELIÈGE, I. (1996). Cue abstraction as a component of categorization processes in music listening. *Psychology of Music, 24*, 131-156.

DELIÈGE, I. (2001). Prototype effects in music listening: An empirical approach to the notion of imprint. *Music Perception, 18*, 371-407.

DEUTSCH, D. (1980). The processing of structured and unstructured tonal sequences. *Perception and Psychophysics, 28*, 381-389.

EEROLA, T., JÄRVINEN, T., LOUHIVUORI, J., & TOIVIAINEN, P. (2001). Statistical features and perceived similarity in folk melodies. *Music Perception, 18*, 275-296.

FERRAND, M., & NELSON, P. (2003). Unsupervised learning of melodic segmentation: A memory-based approach. In *Proceedings of the 5th Triennial ESCOM Conference* (pp. 141-144). Hanover, Germany.

FORTE, A. (1973). *The structure of atonal music*. New Haven, CT: Yale University Press.

FRANKLAND, B. W., & COHEN, A. J. (2004). Parsing of melody: Quantification and testing of the local grouping rules of Lerdahl and Jackendoff's *A Generative Theory of Tonal Music. Music Perception, 21*, 499-543.

HIRAGA, Y. (1997). Structural recognition of music by pattern-matching. In *Proceedings of the International Computer Music Conference (ICMC97)* (pp. 426-429). Thessaloniki, Greece.

ILIOPOULOS, C. S., MOORE, D. W. G., & PARK, K. (1996). Covering a string. *Algorithmica, 16*, 288-297.

KONIARI, D., PREDAZZER, S., & MÉLEN, M. (2001). Categorization and schematization processes used in music perception by 10- to 11-year-old children. *Music Perception, 18*, 297-324.

LALITTE, P., BIGAND, E., POULIN-CHARRONNAT, B., MCADAMS, S., DELBÉ, C., & D'ADAMO, D. (2004). The perceptual structure of thematic materials in *The Angel of Death. Music Perception, 22*, 265-296.

LAMONT, A., & DIBBEN, N. (2001). Motivic structure and the perception of similarity. *Music Perception, 18*, 245-274.

LARTILLOT, O. (2004). A musical pattern discovery system founded on a modelling of listening strategies. *Computer Music Journal*, 28(3), 53-67.

LEMSTRÖM, K., & LAINE, P. (1998). Musical information retrieval using musical parameters. In Proceedings of the International Computer Music Conference (ICMC98), (pp. 341-348). Ann Arbour, Michigan.

LERDAHL, F., & JACKENDOFF, R. (1983). *A generative theory of tonal music*. Cambridge, MA: MIT Press.

LEVINSON, J. (1997). *Music in the moment*. Ithaca, NY: Cornell University Press.

LIDOV, D. (1979). Structure and function in musical repetition. *Journal of the Canadian Association of University Schools of Music, 8*, 1-32.

MCADAMS, S., VIEILARRD, S., HOUIX, O., & REYNOLDS, R. (2004). Perception of musical similarity among contemporary thematic materials in two instrumentations. *Music Perception, 22*, 207-237.

MEEK, C., & BIRMINGHAM, W. P. (2001). Thematic extractor. In *Proceedings of the International Symposium on Music Information Retrieval ISMIR 2001* (pp. 119-128). University of Indiana, Bloomington.

MEREDITH, D., LEMSTRÖM, K., & WIGGINS, G. A. (2002). Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research, 31*, 321-345.

MILLER, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review, 63*, 81-97.

NATTIEZ, J-J. (1975). *Fondements d'une sémiologie de la musique*. Paris: Union Générale d'Editions.

NATTIEZ, J-J. (1990). *Music and discourse: Towards a semiology of music*. Princeton, NJ: Princeton University Press.

POLLARD-GOTT, L. (1983). Emergence of thematic concepts in repeated listening to music. *Cognitive Psychology, 15*, 66-94.

RETI, R. (1951). *The thematic processes in music*. New York: Macmillan.

ROLLAND, P. Y. (1999). Discovering patterns in musical sequences. *Journal of New Music Research, 28*, 334-350.

ROLLAND, P. Y. (2001). FlExPat: Flexible extraction of sequential patterns. In *Proceedings of the IEEE International Conference on Data Mining (IEEE ICDM'01)* (pp. 481-488). San Jose, CA.

ROLLAND, P. Y., & GANASCIA, J. G. (1999). Musical pattern extraction and similarity assessment. In E. Miranda (Ed.), *Readings in music and artificial intelligence* (pp. 115-144). London: Gordon & Breach–Harwood Academic Publishers.

ROWE, R. (1993). *Interactive music systems: Machine listening and composing*. Cambridge, MA: MIT Press.

RUWET, N. (1987). Methods of analysis in musicology. *Music Analysis, 6*, 4-39.

SLOBODA, J. A. (1985). *The musical mind*. Oxford: Clarendon Press. Snyder, B. (2000). *Music and memory: An introduction*. Cambridge, MA: MIT Press.

STAMMEN, D. R., & PENNYCOOK, B. (1993). Real-time recognition of melodic fragments using the dynamic timewarp algorithm. In *Proceedings of the International Computer Music Conference (ICMC'93)* (pp. 232-235).

TEMPERLEY, D. (2001). *The cognition of basic musical structures*. Cambridge, MA: MIT Press.

TEMPERLEY, D., & BARTLETTE, C. (2002). Parallelism as a factor in metrical structure. *Music Perception, 20*, 117-149.

APPENDIX 1

Informal description of Crochemore's pattern extraction algorithm

Let's assume we have a string of symbols (e.g., letters of the alphabet, pitches, pitch-duration tuples, etc.). Each symbol in the string has a corresponding start position in the string (e.g., the start position of the 3rd symbol is 3). The start positions are split into start-sets where each start-set contains the start positions of all the occurrences of each symbol. In the next step, each start-set is split into subsets in relation to itself and the other start-sets of the same level (for instance, in Figure 18, start-set *a*={1,3,7,9,12,16} is split in relation to *b*={2,4,8,11,13}, into {1,3,7,12} & {9,16} because each start position {1,3,7,12} has a corresponding start position in start-set b that is greater by 1, which essentially means that pattern ab occurs in these start positions: *ab*={1,3,7,12}.) This procedure repeats for each level of pattern lengths until the largest possible recurring pattern is found; then the algorithm stops. The algorithm employs a technique to reduce the aforementioned procedure from $O(n^2)$ to $O(n \cdot logn)$: each start-set need only be split in relation to all the other start-sets with the exception of the largest start-set (e.g., for the three symbol alphabet {a,b,c} of Figure 18, if the start-set for pattern *a* has been split in relation to start-sets *b* and *c* giving *ab* and *ac* start-sets it need not be split in relation to itself—being the largest start-set—as the remaining start positions correspond obviously to pattern *aa*)—in the case of a binary alphabet the technique is also called the "smaller-half trick" whereas for a larger alphabet the "larger-part trick".

A formal description of Crochemore's algorithm is presented in (Crochemore 1981); see also description by Iliopoulos et al. (1996).

APPENDIX 2

The Local Boundary Detection Model (LBDM) is based on the two following rules:

Change Rule (CR): Boundary strengths proportional to the degree of change between two consecutive intervals are introduced on either of the two intervals (if both intervals are identical no boundary is suggested).

Proximity Rule (PR): If two consecutive intervals are different, the boundary introduced on the larger interval is proportionally stronger.

The Change Rule can be implemented by a degree-of-change function (see suggestion below). *The Proximity Rule* can be implemented simply by multiplying the degree-of-change value with the absolute value of each pitch/time/dynamic interval. This way, not only relatively greater neighboring intervals get proportionally higher values but also greater intervals get higher values in absolute terms.

In the description of the algorithm below only the pitch, *IOI* and *rest* parametric profiles of a melody are mentioned. It is possible, however, to construct profiles for dynamic intervals (e.g., velocity differences) or for harmonic 'intervals' (distances between successive chords) or any other relevant parameter.

start position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
string	a	b	a	b	c	c	a	b	a	c	b	a	b	c	c	a

1	a={1,3,7,9,12,16}	b={2,4,8,11,13}	c={5,6,10,14,15}		
2	ab={1,3,7,12}	ba={2,8,11}	bc={4,13}	ca={6,15}	cc={5,14}
3	aba={1,7}	abc={3,12}	bab={2,11}	bcc={4,13}	cca={5,14}
4	abcc={3,12}	babc={2,11}	bcca={4,13}		
5	abcca={3,12}	babcc={2,11}			
6	babcca={2,11}				

FIG. 18. An example of the application of Crochemore's exact pattern extraction algorithm on a string of symbols from alphabet {a,b,c}. (Numbers on the left column indicate pattern lengths—the equal sign indicates the corresponding start-set for each pattern—patterns that occur only once are not reported).

The LBDM algorithm

A melodic sequence is converted into a number of independent parametric interval profiles P_k for the parameters: *pitch* (pitch intervals), *ioi* (interonset intervals) and *rest* (rests—calculated as the interval between current onset with previous offset). Pitch intervals can be measured in semitones, and time intervals (for IOIs and rests) in milliseconds or quantized numerical duration values. Upper thresholds for the maximum allowed intervals should be set, such as the whole-note duration for IOIs and rests and the octave for pitch intervals; intervals that exceed the threshold are truncated to the maximum value.

A parametric profile P_k is represented as a sequence of n intervals of size x_i:

$P_k = [x_1, x_2, \dots x_n]$ where: $k \in \{pitch, ioi, rest\}$, $x_i \geq 0$ and $i \in \{1,2,\dots n\}$

The degree of change r between two successive interval values x_i and x_{i+1} is given by:

$$r_{i,i+1} = \frac{|x_i - x_{i+1}|}{x_i + x_{i+1}} \qquad \text{iff } x_i + x_{i+1} \neq 0 \text{ and } x_i, x_{i+1} \geq 0$$

$$r_{i,i+1} = 0 \qquad \text{iff } x_i = x_{i+1} = 0$$

(N.B. A small value should be added to the size of all intervals, such as 1 semitone to pitch intervals, so as to avoid irregularities introduced by intervals of size 0).

The strength of the boundary s_i for interval x_i is affected by both the degree of change to the preceding and following intervals, and is given by the function:

$$s_i = x_i \cdot (r_{i-1,i} + r_{i,i+1})$$

For each parameter k, sequence $S_k = [s_1, s_2, \dots s_n]$ is calculated, and normalized in the range: $[0, 1]$.

The overall local boundary strength profile for a given melody is a weighted average of the individual strength sequences S_k (weights used in current experiments: w_{pitch}=0.25, w_{ioi}=0.50, w_{rest}=0.25). Local peaks in this overall strength sequence indicate local boundaries.