Representation and Discovery of Vertical Patterns in Music*

Darrell Conklin

Computational Biology
ZymoGenetics, Inc.
Seattle, USA
conklin@zgi.com

Abstract. The automated discovery of recurrent patterns in music is a fundamental task in computational music analysis. This paper describes a new method for discovering patterns in the vertical and horizontal dimensions of polyphonic music. A formal representation of music objects is used to structure the musical surface, and several ideas for viewing pieces as successions of vertical structures are examined. A knowledge representation method is used to view pieces as sequences of relationships between music objects, and a pattern discovery algorithm is applied using this view of the Bach chorale harmonizations to find significant recurrent patterns. The method finds a small set of vertical patterns that occur in a large number of pieces in the corpus. Most of these patterns represent specific voice leading formulae within cadences.

1 Introduction

The discovery of repeated structure in data is a fundamental form of inductive inference. In music, the discovery of repeated patterns within a single piece is a precursor to revealing its construction in terms of basic structures such as themes, motifs, and paradigmatic types [1]. The discovery of patterns that occur across many pieces in a style can yield signature motifs that can be used in the synthesis [2] and classification [3] of new pieces in the style. Regardless of the analytical task, computational methods for automated pattern discovery in music can be guided by two basic principles. First, to detect common structure, they should not be affected by typical musical transformations such as transposition and simple melodic elaboration. Second, to be of practical use, they should attempt to limit their output to patterns that have musical or statistical significance [4, 5].

The computational analysis of polyphonic music presents unique challenges not encountered in the study of isolated melody. Individual voices, while moving separately through the horizontal or melodic dimension, create vertical sonorities by temporal overlap with notes in other voices. While each individual voice

* in C. Anagnostopoulou, M. Ferrand, and A. Smaill (Eds.): *Music and Artificial Intelligence: Proc. ICMAI 2002*, LNAI 2445, pp. 32-42, Springer-Verlag, 2002.

follows general principles of voice leading, sequences of vertical structures outline a coherent harmonic motion. To adequately model polyphonic music, a knowledge representation method must allow for efficient retrieval, manipulation, and reasoning about concurrent and successive music objects.

Most approaches to automated pattern discovery in music have restricted their attention to an isolated melodic line of a single piece [6–9]. This paper will report on an abstract representation method for music in which both melodic and harmonic structures of polyphonic music can be described. A new representation method, extending recent work on *viewpoints* in music [4], is used for the description of relationships between successive vertical structures. A pattern discovery algorithm is used to find significant recurrent vertical structures in multiple pieces. Initial results of this technique with the Bach chorales are presented. The representation and discovery method produces several significant vertical patterns that occur in a large number of pieces in the corpus.

The paper concludes with a discussion of the strengths and limitations of the approach, and ideas for future research.

2 Methods

2.1 Music Objects

In our ontology of music we consider three types of musical objects: notes, simultaneities, and sequences. All music objects in a piece acquire an *onset time* and *duration*; these temporal attributes will be assumed present throughout the paper and will not be explicitly notated. The *Note* is the basic primitive music object. The *Sim* and *Seq* object types are polymorphic and can contain any other type of music object as components, including not only $Note$ objects, but (recursively) other Sim and Seq objects.

The primitive type *Note* is defined as a tuple of pitch class and octave, or alternatively, as an integer Midi number. Music objects are then defined by the recursive algebraic data type M:

$$\begin{aligned} M ::= Note \;|\; Seq(M) \;|\; Sim(M) \\ join : Seq(X) \times X \rightarrow Seq(X) \\ layer : Sim(X) \times X \rightarrow Sim(X) \end{aligned} \tag{1}$$

Informally, objects of type $Seq(X)$ are constructed by joining together music objects of type X, which do not overlap in time. Objects of type $Sim(X)$ are constructed by layering objects of type X, all of which have the same onset time.

Throughout this paper, square brackets [ ] will be used to denote a sequence. A sequence $[o_1, \ldots, o_n]$ of joined music objects will be abbreviated as $\overline{o_n}$. Angle brackets $\langle\ \rangle$ will be used to denote a simultaneity.

This music ontology provides a natural way to describe polyphonic music in a hierarchical fashion. Naturally, similar music ontologies have appeared elsewhere, for example in the algebraic data type of Hudak et al. [10] and in the music structures approach of Balaban [11]. The OpenMusic environment also uses a similar ontology with simple elements, sequences, and superpositions as objects [12]. It appears that the various instances of this ontology differ mainly in the temporal overlapping restrictions imposed on objects, whether recursive embedding of objects is permitted, and whether rests are primitive objects.

2.2 Structuring and Partitioning of Pieces

A polyphonic piece of music is encoded at the basic level by a $Sim(Seq(Note))$ object; that is, by layered voices similar to a Midi multiple track encoding of a piece. For example, the Bach chorale fragment in Figure 1 would be encoded as four simultaneous sequences (octave encoding not shown):

$$\begin{array}{l} \langle\ [A, D, A, B, A, G, F\sharp, F\sharp], \\ \ \ \ [D, D, D, D, D, C\sharp, D, D], \\ \ \ \ [F\sharp, F\sharp, G, A, G, F\sharp, E, D, E, D, D], \\ \ \ \ [D, B, F\sharp, G, A, A, D, D]\ \rangle \end{array}$$

where $\langle\ \rangle$ denote a *Sim* object and [ ] denote a *Seq* object.

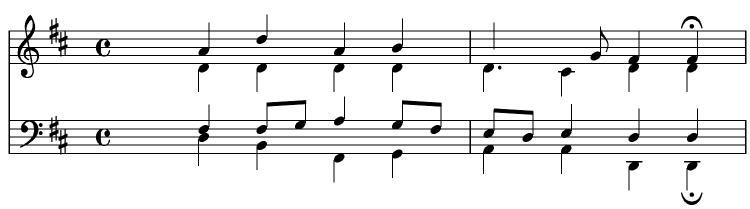

Fig. 1. from Bach, chorale BWV 262.

The pattern discovery algorithm used in this paper rapidly finds recurrent patterns in abstract sequences derived from the basic encodings of multiple pieces. Therefore, as a prelude to finding vertical patterns, it is necessary to restructure or partition the basic encoding of a piece from a simultaneity of *Seq* objects to a sequence of *Sim* objects. The music object data type is well-suited to this task because it allows the musical surface to be structured in many alternative ways.

There are two basic ways to restructure a piece into a sequence of *Sim* objects. The *natural* partitioning simply creates a new object whenever all voices have the same onset time. For example, the second bar of Figure 1 contains three natural $Sim(Seq(Note))$ objects, in the sequence:

$$\begin{array}{l} [\ \langle [A, G], [D, C\sharp], [E, D, E], [A, A] \rangle, \\ \ \ \ \langle [F\sharp], [D], [D], [D] \rangle, \\ \ \ \ \langle [F\sharp], [D], [D], [D] \rangle\ ] \end{array}$$

Though sufficient for simple species counterpoint, this example illustrates that the natural partitioning technique will under-partition more complex polyphony because common onset times might not be equivalent to the harmonic rhythm of the piece. A standard solution to this problem is to perform a *full expansion* of the piece, wherein pitches are duplicated at every unique onset time. For example, Figure 2b illustrates the expansion of the fragment in Figure 2a. This expansion would be represented by $Sim(Note)$ objects, in the sequence:

$$[\ \langle B, D, B, G\rangle, \langle B, E, C, G\rangle, \ldots, \langle A, F\sharp, C, A\rangle\ ]$$

A piece can now effectively be treated as a sequence of regular simultaneities. However, full expansion might over-partition a piece because simultaneities can be created even when there is no change in harmony. Pardo and Birmingham [13] consider as a solution a subsequent re-grouping of expanded partitions to correspond to predicted harmonic units. In the present method, the expanded piece can either be modeled directly or sampled at regular metric intervals using threaded viewpoints. This technique will be described below.

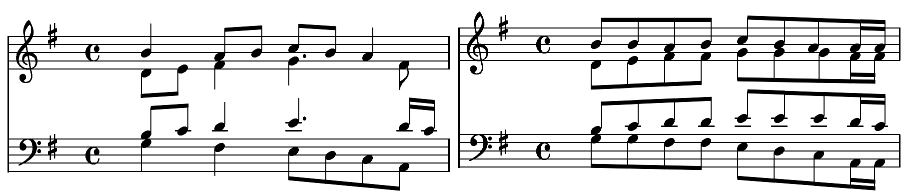

Fig. 2. from Bach, chorale BWV 260. a) original, b) expanded.

Full expansion of complex polyphony to reveal vertical structures for music generation is performed by Lartillot et al. [14] and Alamkan et al. [15]. Lemström and Tarhio [16] use full expansion in the context of polyphonic music information retrieval. Metric sampling of vertical structures for music generation is also performed by Ponsford and Wiggins [17]. The Humdrum toolkit [18] has a "ditto" function that performs full expansion.

2.3 Typed Viewpoints

A *viewpoint* is a mathematical function that computes features of music objects in a sequence, including features that denote relationships between objects within the sequence. Earlier work on viewpoints [19, 4] was restricted to sequences of notes. With music objects defined as above, it is possible to extend the capability of viewpoints to model sequences of any type of music object.

A *viewpoint of type* X maps objects of type $Seq(X)$ to *viewpoint elements.* For example, the twelve-tone melodic interval viewpoint is of type $Note$, and has integers as viewpoint elements. The pitch class interval viewpoint (interval modulo 12) is of type $Note$, and has integers between 0 and 11 as viewpoint elements. Features of melodic segments [8, 20] might be described by viewpoints of type $Seq(Note)$ with lists as viewpoint elements. Later in the paper we show how to construct viewpoints of type $Sim(X)$ from base viewpoints of type X.

The semantics of a viewpoint is that a viewpoint element models the attribute of an object or the relation between an object and the sequence of objects that precede it in a piece. More precisely, for a viewpoint τ, if

$$\tau(join(c, o)) = v \tag{2}$$

we infer that the music object o has the attribute value v in the context of the sequence c. This construction highlights three points; first, that a *relation* between music objects can be encoded as an *attribute* of the latest object. For example, the melodic interval between o and the last object in c can be encoded as an attribute of o. A single viewpoint encodes only one such relationship, because τ is a function. Second, the complete context preceding an object, and not only the immediately preceding object, is used to compute its attribute. This is to allow viewpoints to model relations between any number of preceding objects. Third, a viewpoint cannot look ahead in the sequence; the viewpoint element of an object depends on preceding objects in the sequence. This is because viewpoints were initially designed for left-to-right music prediction and generation [19].

2.4 Viewpoint Sequences

Given a sequence $\overline{o_n}$ of music objects of some type X, and a viewpoint τ of the same type X, the *viewpoint sequence* for $\overline{o_n}$ is a list of defined viewpoint elements which is the application of τ to each initial segment of $\overline{o_n}$:

$$[\tau(\overline{o_1}), \tau(\overline{o_2}), \ldots, \tau(\overline{o_n})] \tag{3}$$

For example, for the melodic interval viewpoint, the viewpoint sequence for the soprano line $[A, D, A, B, A, G, F\sharp, F\sharp]$ of Figure 1 is

$$[5, -5, 2, -2, -2, -1, 0] \tag{4}$$

Viewpoints can be seen as a method for handling sparse musical data. A viewpoint sequence is more abstract than the musical surface, and therefore significant recurrent patterns are more likely to be revealed within viewpoint sequences than within encodings of musical surfaces. In this sense, viewpoints are analogous to *class-based models* [21] of natural language, which view sentences as sequences of abstract word features. In a similar way, a viewpoint encodes notes by abstract equivalence classes rather than by pitch and duration.

2.5 Viewpoint Patterns

A *viewpoint pattern* is a viewpoint sequence fragment encountered in a corpus of pieces. Whereas music objects and viewpoint sequences represent individual, concrete entities, viewpoint patterns represent abstractions, or concepts. A pattern *occurs* in a piece if it is contained in the viewpoint sequence for that piece. The *coverage* of a set of patterns is the number of pieces containing one or more patterns in the set. The *p-value* of a pattern that occurs n times in a corpus is the probability that the pattern can be found n or more times in a random corpus of the same size (see [4] and [5] for details on p-value computation). A *significant* pattern has a low p-value (say, < 0.01).

A viewpoint pattern P *subsumes* another viewpoint pattern P', written $P \succeq P'$, if P is contained in P'. For example, for the melodic interval viewpoint,

$$[2, -2] \succeq [5, -5, 2, -2, -1, 0] \tag{5}$$

If $P \succeq P'$, then the coverage of P is always greater than the coverage of P'. Given a set of significant patterns, a *shortest significant pattern* is subsumed by no other pattern in the set, and a *longest significant pattern* subsumes no other pattern in the set.

Shortest and longest significant patterns are useful in different analytical situations. Shortest significant patterns are interesting because they are the most general significant patterns found in a corpus (by definition, any shorter pattern would not be significant) and can be found in many pieces, including those not in the analysis corpus. Longest significant patterns can highlight extensive similarity within a single piece or a few pieces. In the study of the Bach chorales reported below, shortest significant patterns are found within a large corpus.

The pattern discovery algorithm used below in Results employs a suffix tree data structure [22] to rapidly enumerate recurrent patterns in a viewpoint sequence representation of the corpus [4]. Pattern p-values are computed for all recurrent patterns and the longest and shortest significant patterns are then revealed by performing pattern subsumption tests.

2.6 Constructing Vertical Viewpoints

A powerful feature of viewpoints as a representation formalism for music is that complex viewpoints can be constructed from simpler ones using viewpoint constructors. For example, two viewpoints might be linked together to produce a new one that computes pairs of viewpoint elements; or a viewpoint might be threaded through a piece at defined metric locations [4]. In this section a new constructor that creates vertical viewpoints is informally described.

There are many possible viewpoints for $Sim(Note)$ objects, many drawing from atonal music theory research. A basic requirement for the analysis of multiple pieces is that a viewpoint τ is transposition invariant, meaning that $\tau(X) = \tau(T(X))$ for any transposition T of the notes embedded in a sequence X of music objects. Though the pitch class set is not invariant, various types of interval functions [23] are invariant. The harmonic function of a chord is invariant and abstract in that it ignores voicing. Figured bass notation is invariant provided that the bass note is expressed as a relation to a tonic rather than as an absolute pitch class. Various types of interval vectors are invariant; for example, intervals between a bass note and all other voices, or between all pairs of objects in a simultaneity [18]. Finally, we mention as candidates formal groups of transformations of triads including Parallel (major to minor mode) and Relative (major to relative minor) [24].

With the exception of figured bass notation, the viewpoints presented above do not retain information about voice leading. Encoding voice leading information is possible with viewpoints. A viewpoint of type X can be "lifted" to form a new (vertical) viewpoint of type $Sim(X)$. This new viewpoint represents relations between all objects in a simultaneity and preceding objects in the same voice. Figure 3 illustrates an example of this technique for the melodic interval viewpoint.

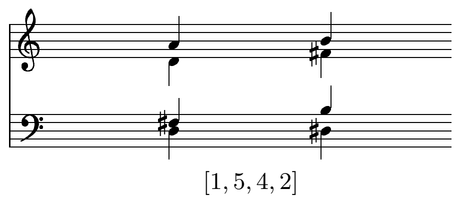

Fig. 3. The melodic interval viewpoint, of type $Note$, is lifted to a new viewpoint of type $Sim(Note)$. The new viewpoint returns a tuple of intervals between all voices.

As another example, the pitch class interval viewpoint (interval modulo 12), threaded on quarter note beats, and lifted, when applied to the expanded fragment of Figure 2b yields the viewpoint sequence

$$[[11, 3, 4, 10], [10, 2, 1, 3], [8, 0, 0, 9]] \tag{6}$$

representing the vertical pitch class interval between the $Sim(Note)$ objects at beats 2, 3, and 4 of the fragment. Viewpoints can be threaded using any other time span; for the Bach chorales a quarter note time span is meaningful because it corresponds to harmonic rhythm.

Though the examples above used melodic and pitch class interval viewpoints, the vertical viewpoint construction technique can be applied to any other base viewpoint, for example, scale degree of a note, diatonic intervals, and melodic contour. Furthermore, the technique is not restricted to $Note$ viewpoints, and can be applied to any type of music object and appropriately typed base viewpoint.

3 Results

The typed viewpoint scheme described above in Methods was applied to 185 Bach four-part chorales. Pieces were expanded, and viewed using the pitch class interval viewpoint, threaded at every quarter note beat, and lifted. There are approximately 9000 vertical structures in this data set. The pattern discovery algorithm discussed in Methods was used to find all shortest significant patterns.

A total of 32 shortest significant patterns was found (see Table 1). The coverage of this set was 142 pieces; therefore, the majority of pieces in the corpus contain at least one occurrence of one of the 32 patterns. Table 1 also presents a harmonic analysis of instances of the pattern. For these significant patterns two lifted voice leading intervals suffice to fully specify the chord progression.

	Pattern	p-value	coverage	function
1	[[2,10,11,0],[5,9,8,10]]	≈ 0	36	$\text{ii}_5^6 - \text{V} - \text{I}$
2	[[2,5,11,0],[5,2,8,10]]	5e-14	19	$\text{ii}_5^6 - \text{V} - \text{I}$
3	[[2,11,11,0],[5,8,8,10]]	2e-11	19	$\text{ii}_5^6 - \text{V} - \text{i}$
4	[[2,11,10,0],[5,8,9,10]]	2e-10	16	$\text{ii}_5^6 - \text{V} - \text{I}$
5	[[4,0,0,11],[0,0,11,0]]	2e-06	15	$\text{i}^6 - \text{V}_4^5 - \text{V}_3^5$
6	[[0,0,11,0],[5,9,8,10]]	9e-06	20	$\text{V}_4^5 - \text{V}_3^5 - \text{I}$
7	[[2,6,11,0],[5,1,8,10]]	4e-05	7	$\text{ii}_5^6 - \text{V} - \text{i}$
8	[[3,0,0,0],[4,0,0,11]]	2e-05	12	$\text{i} - \text{i}^6 - \text{V}_4^5$
9	[[5,0,2,10],[2,11,10,0]]	2e-05	10	$\text{I} - \text{ii}_5^6 - \text{V}$
10	[[9,11,0,2],[0,10,0,1]]	2e-05	6	$\text{i}^6 - (\text{V}^6) - \text{i}$

Table 1. The top 10 shortest significant vertical pitch class interval patterns found in the Bach chorales. The harmonic function of instances of the pattern is indicated. Brackets enclose a chord that contains an accented non-harmonic tone.

Pattern 1 of Table 1 is the most statistically significant shortest pattern found. Figure 4 presents a sample of various instances of this pattern, which outlines a standard $\text{ii}_5^6 - \text{V} - \text{I}$ cadential formula. It is apparent that a wide diversity of melodic diminution is applied to this pattern, and that the method has succeeded in identifying a deeper structure than is evident in the musical surface alone. Passing tones, escape tones, anticipations and consonant skips are encountered. Instances of this pattern have a two beat repetition of the soprano $\hat{2}$ followed by $\hat{1}$. The leading tone in the alto voice falls by skip to the $\hat{5}$ in the final I. The motion of the bass from $\text{V} - \text{I}$ is either ascending or descending; the pitch class interval viewpoint is invariant to interval direction. The pattern occurs in both 3/4 and 4/4 meters. It is evident that expansion of the pieces was necessary in order to reveal this pattern. Furthermore, the pitch class interval viewpoint was necessary, and the pattern could not be revealed in its generality by searching for transposed pitch repetition.

A full analysis of the discovered vertical patterns will appear elsewhere; here we point out that several patterns are variants of others, for example, patterns

Fig. 4. A selection of instances of the $\mathrm{ii}_5^6 - \mathrm{V} - \mathrm{I}$ cadential formula represented by pattern 1 from Table 1. Clockwise from upper left: from Bach, chorales BWV 329, 361, 356, 379, 380, 355. The complete bars containing the pattern are notated.

1 and 4 simply swap the alto and tenor voices. Instances of patterns 1, 2 and 4 are alternative voicings of the $\mathrm{ii}_5^6 - \mathrm{V} - \mathrm{I}$ formula. Patterns 3 and 7 are minor mode versions of this cadential formula. If desired, viewpoints could be easily designed that are invariant to these effects of voicing and mode. Other patterns, for example, 5 and 8, overlap and are joined into a longer pattern simply by instructing the algorithm to seek longest rather than shortest significant patterns (results of longest significant pattern discovery not shown).

4 Discussion

A knowledge representation method for music should be able to express common abstract musical patterns. This paper has used a music representation method that allows a piece to be alternatively structured to reveal both horizontal and vertical patterns. The two dimensions of polyphonic music were coupled by constructing a lifted interval viewpoint that takes voice leading into account. Though several interesting patterns were found, a more complete study of different vertical viewpoints should be done. It is expected that viewpoints achieving an appropriate level of abstraction will give rise to extensive significant patterns.

A shortcoming of this technique for describing relationships between vertical objects is that a polyphonic piece must initially be structured as a set of monophonic voices. This precludes its use on free-form Midi data, where a single track can contain concurrent events, and where there may be no natural way to divide the track into voices. Nevertheless, this is a limitation only of lifted vertical viewpoints, and other vertical viewpoints suggested in this paper could be applied to free-form Midi data.

Related work on polyphonic music has produced other interesting ideas for representing vertical structures. Meredith et al. [25] describe a representation in which notes are encoded as multidimensional vectors, one dimension for each feature of a note, and repeated patterns within a piece are geometric translations of vectors. Farbood and Schoner [26] use harmonic intervals as one component of a model for predicting first species counterpoint to a cantus firmus line. As discussed above in Methods, several groups have considered techniques for partitioning pieces into vertical objects.

A weakness of the expansion and sampling technique for partitioning polyphonic music into vertical structures is that it will unavoidably sample accented non-harmonic tones. For example, whenever a suspension occurs, after expansion the duplicated suspended note will become a non-harmonic tone. The vertical structures containing such non-harmonic tones might in fact be recurrent, but a deeper structure could probably be discovered if the non-harmonic tones were resolved prior to pattern discovery [17, 27]. An interesting approach to this problem would be to first perform a limited Schenkerian-style reduction of the musical surface, followed by expansion and pattern discovery. To implement this idea effectively using current pattern discovery methods, the reduction process should produce only one deterministic reduction of a piece.

References

1. N. Cook. *A Guide to Musical Analysis*. Oxford University Press, 1987.
2. D. Cope. *Computers and Musical Style*. A-R Editions, 1991.
3. M. Westhead and A. Smaill. Automatic characterisation of musical style. In *Music Education: An AI Approach*, pages 157–170. Springer-Verlag, 1993.
4. D. Conklin and C. Anagnostopoulou. Representation and discovery of multiple viewpoint patterns. In *Proceedings of the International Computer Music Conference*, pages 479–485, Havana, Cuba, 2001. International Computer Music Association.
5. D. Huron. What is a musical feature? Forte's analysis of Brahms's Opus 51, No. 1, revisited. *Music Theory Online*, 7(4), July 2001.
6. P-Y. Rolland and J-G. Ganascia. Musical pattern extraction and similarity assessment. In E. Miranda, editor, *Readings in Music and Artificial Intelligence*, chapter 7, pages 115–144. Harwood Academic Publishers, 2000.
7. J-L. Hsu, C-C. Liu, and A. Chen. Discovering nontrivial repeating patterns in music data. *IEEE Transactions on Multimedia*, 3(3):311–325, September 2001.
8. E. Cambouropoulos. *Towards a General Computational Theory of Musical Structure*. PhD thesis, University of Edinburgh, 1998.
9. A. Smaill, G. Wiggins, and M. Harris. Hierarchical music representation for composition and analysis. *Computers and the Humanties*, 93:7–17, 1993.
10. P. Hudak, T. Makucevich, S. Gadde, and B. Whong. Haskore music notation — an algebra of music. *Journal of Functional Programming*, 6(3):465–483, May 1996.
11. M. Balaban. The music structures approach in knowledge representation for music processing. *Computer Music Journal*, 20(2):96–111, 1996.

12. C. Agon, G. Assayag, O. Delerue, and C. Rueda. Objects, time and constraints in OpenMusic. In *Proceedings of the International Computer Music Conference*, Ann Arbor, Michigan, 1998. International Computer Music Association.
13. B. Pardo and W. Birmingham. Automated partitioning of tonal music. In *Conference of the Florida Artificial Intelligence Research Society*, May 2000.
14. O. Lartillot, S. Dubnov, G. Assayag, and G. Bejerano. Automatic modeling of music style. In *Proceedings of the International Computer Music Conference*, pages 447–453, Havana, Cuba, 2001. International Computer Music Association.
15. C. Alamkan, W. Birmingham, and M. Simoni. Stylistic structures: an initial investigation of the stochastic generation of music. Technical Report CSD-TR-395-99, Electrical engineering and computer science department, University of Michigan, 1999.
16. K. Lemström and J. Tarhio. Searching monophonic patterns within polyphonic sources. In *Proc. Content-based multimedia information access (RIAO 2000)*, pages 1261–1279, 2000.
17. D. Ponsford, G. Wiggins, and C. Mellish. Statistical learning of harmonic movement. *Journal of New Music Research*, 28(2), 1999.
18. D. Huron. *Music Research using Humdrum: A user's guide*.
19. D. Conklin and I. Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1):51–73, 1995.
20. K. Höthker, D. Hörnel, and C. Anagnostopoulou. Investigating the influence of representations and algorithms in music classification. *Computers and the Humanities*, 35:65–79, 2001.
21. P. Brown, V. Della Pietra, P. deSouza, J. Lai, and R. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.
22. D. Gusfield. *Algorithms on strings, trees, and sequences.* Cambridge University Press, 1997.
23. D. Lewin. Intervallic relations between two collections of notes. *Journal of Music Theory*, 3(2):298–301, 1959.
24. D. Lewin. *Generalized Musical Intervals and Transformations.* Yale University Press, 1987.
25. D. Meredith, G. Wiggins, and K. Lemström. Pattern induction and matching in polyphonic music and other multidimensional datasets. In *Proc. Conference on Systemics, Cybernetics and Informatics*, volume X, pages 61–66, 2001.
26. M. Farbood and B. Schoner. Analysis and synthesis of Palestrina-style counterpoint using Markov chains. In *Proceedings of the International Computer Music Conference*, pages 471–474, Havana, Cuba, 2001. International Computer Music Association.
27. A. Marsden. Representing melodic patterns as networks of elaborations. *Computers and the Humanities*, 35:37–54, 2001.