Representation and Discovery of Multiple Viewpoint Patterns*

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

Christina Anagnostopoulou
Faculty of Music
University of Edinburgh
Edinburgh, Scotland
chrisa@dai.ed.ac.uk

Abstract

An important problem in computational music analysis is the representation and automated discovery of recurrent patterns. In this paper we present a new method for pattern representation and discovery in a large corpus of music. Using the formalism of multiple viewpoints, music is viewed as multiple streams of description derived from the basic surface representation. Patterns are discovered within viewpoint sequences derived from the corpus for selected viewpoints. A statistical method is used to restrict attention to only those patterns which occur much more frequently than expected, where expectation is based on a Markov model of viewpoint elements. The concept of the longest significant patterns in a corpus is introduced. The method presented in this paper is designed to rapidly enumerate all longest significant patterns within a large corpus. An application of the method to the Bach chorales is presented.

1 Introduction

The low entropy of music is due to the inherent structural constraints in a musical style, and repetition of both *intra*- and *inter*- opus musical material. An important problem in computational music analysis is the representation and automated discovery of recurrent musical patterns. Patterns can be used for abstraction and compact representation of a work (Smaill et al. 1993); as musical building blocks for the paradigmatic analysis of a work (Nattiez 1975); as fragments for motive-based algorithmic composition (Rolland and Ganascia 2000); as keys for content-based music retrieval (Hsu et al. 1998); and for the recognition and distinction of musical style or authorship (Westhead and Smaill 1993).

Repetition in music occurs not only as repetition of exact pitches and durations, or as mere transposition into a different key, but is often much more subtle. There can be repetitions within different musical parameters, such as specific intervals, melodic motion (contour), relative rhythmic values, middle or fundamental structure, harmonic progressions (implicit or explicit), register, dynamics series, pitch class sets, and so on. Approaches to pattern discovery in music analysis have so far concentrated on the similarity relationships within pitch or transposed passages rather than on recurrent patterns within deeper musical parameters. However, in a music analysis task of any kind, it would make more sense to be able to capture these recurrent patterns: they are more general, look at a deeper level of similarity within the musical corpus and make explicit exactly where the similarities between the patterns lie. Pattern discovery algorithms should be able to handle the comparison of a small set of pieces as well as the processing of a large corpus with hundreds or thousands of pieces. Finally, such methods should have some selection procedures for the results, removing uninteresting patterns from consideration.

The topic of this paper is the discovery of general patterns that span a substantial number of diverse works. Our pattern discovery technique is based on the music representation formalism of multiple viewpoints (Conklin and Witten 1995), presented in this paper, where each viewpoint models some musical parameter. In this way we are able to search for patterns within these parameters, or viewpoints, rather than patterns of notes taken directly from the music, and we can capture the exact level where similarity occurs in the music. Pattern discovery is performed by building a suffix tree data structure with all multiple viewpoint sequences derived from the corpus for chosen viewpoints. A subsequent step finds those patterns that occur in a specified minimum number of pieces and that satisfy a statistical significance criterion. A further filtering looks at all significant discovered patterns and selects the *longest significant patterns* within the set. This paper presents an application of the method to the Bach chorales.

*In *Proceedings of the 2001 International Computer Music Conference, International Computer Music Association, 479–485, 2001.*

2 Methods

2.1 Multiple viewpoints of music

The representation language we use for music is based on the formalism of multiple viewpoints (Conklin and Witten 1995). Viewpoints are functions, defined or selected by the music analyst, that operate on the basic representation. A viewpoint models some specific type of musical feature derived from the musical surface, for example, melodic contour, intervals, duration, or interval from a tonic referent pitch. A piece of music is therefore transformed into a higher level description derived from the basic surface representation. At the surface, a piece is represented as a set of sequences (tracks, voices) of events derived from a MIDI encoding: an event has a pitch, duration, and a start time. In addition to these event attributes, we assume some global attributes such as fermatas (used as phrase markers), key signature, and time signature.

In more detail, a viewpoint is a partial function which associates a *viewpoint element* with sequences. The notation $[\tau]$ denotes the range of this function; the set of valid viewpoint elements for a viewpoint τ. For example, for the melodic interval viewpoint, the viewpoint elements are integers. A viewpoint is a partial function, meaning that it may be undefined at certain locations. For example, the melodic interval viewpoint is undefined for a sequence containing only one event (see Figure 1).

For every viewpoint a *viewpoint sequence* function transforms a sequence of basic events into a sequence of defined viewpoint elements. The viewpoint sequence function simply applies the viewpoint to every element in the sequence, retaining those elements where the viewpoint is defined. For example, for the melodic contour viewpoint, this function transforms a sequence of pitches into a sequence of contour indicators (see Figure 1).

A *linked viewpoint* is a combination of two or more viewpoints that models other viewpoints simultaneously. A link between viewpoints can be defined using the constructor $\otimes$. For a linked viewpoint $\tau = \tau_1 \otimes \ldots \otimes \tau_n$, and any sequence m, $\tau(m)$ is undefined if $\tau_i(m)$ is undefined for any component viewpoint, else it is the tuple $\langle \tau_1(m), \ldots, \tau_n(m) \rangle$. The set of viewpoint elements is therefore the cross product of all sets of component viewpoint elements:

$$[\tau_1 \otimes \ldots \otimes \tau_n] = [\tau_1] \times \ldots \times [\tau_n] \qquad (1)$$

For example, Figure 1 illustrates the linked viewpoint between melodic contour and duration. Note how the elements of this viewpoint are pairs of values, one from each component viewpoint.

A *threaded* viewpoint models the value of a *base viewpoint* at defined temporal or metric locations within a piece; for example, at the beginning of a bar or phrase or at every quarter note pulse. These defined locations are captured by a *test viewpoint*. A threaded viewpoint is defined only at locations where the test viewpoint is true. In this way, a viewpoint "threads" through a sequence, potentially ignoring non-adjacent surface events. Any viewpoint (even a linked viewpoint) can be used as a base viewpoint. The test viewpoint must have a Boolean (0 or 1) value. Given a base viewpoint τ and a test viewpoint θ, a threaded viewpoint can be defined using the constructor $\oslash$. The set of viewpoint elements for a threaded viewpoint is the cross product of the base viewpoint elements and the set of inter onset intervals:

$$[\tau \oslash \theta] = [\tau] \times [\mathtt{ioi}] \qquad (2)$$

The `ioi` viewpoint is the inter onset interval between two events; the difference between their start times.

For example, we can construct a viewpoint that measures the melodic interval between events that occur as the first event in a bar, or a viewpoint that threads through events that start on crotchet beats (see Figure 1).

2.2 Viewpoint patterns

A *viewpoint pattern* P_τ is a sequence of viewpoint elements for some viewpoint τ. A pattern *occurs* in a piece if it is contained in the viewpoint sequence for that piece (see Figure 2). The *length* of a pattern P is denoted $l(P)$. The *empty pattern* $\emptyset_\tau$ for a viewpoint τ has zero length and is defined to occur anywhere that the viewpoint is defined. Henceforth we omit the subscript from viewpoint patterns as the viewpoint should always be evident from context.

The *piece count* of a pattern is the number of pieces that a pattern occurs in. The *total count* of a pattern is its total number of occurrences, including repetitions within an individual piece.

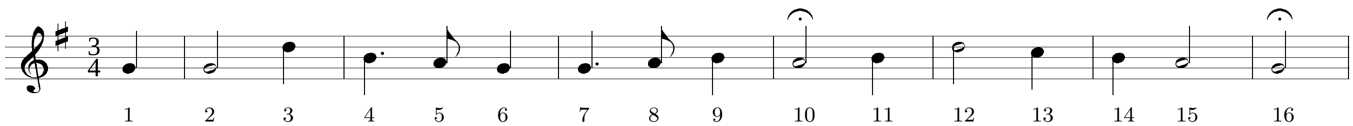
Pattern scoring. The potential musical significance of a viewpoint pattern P with respect to a data set is evaluated by comparing the piece count of P with how many times we expect it to occur if pieces in the data set were generated from a Markov model of viewpoint elements.

Large differences between observed and expected counts indicate a potentially interesting pattern. A pattern is given a *pattern score*, which represents the magnitude of this difference:

$$\frac{(\#(P) - E(P))^2}{E(P)} \qquad (3)$$

where $\#(P)$ is the observed total count and $E(P)$ is the expected total count (defined below) for the pattern P. The score for a pattern will increase with the difference between its observed and expected total count.

Expected count for a pattern. The expected total count for a pattern is the number of sites where the pattern could possibly occur multiplied by the probability of finding the pattern



Viewpoint	Sequence	Result
`st`	$\overline{e_7}$	36
`pitch`	$\overline{e_{12}}$	74
`int`	$\overline{e_1}$	undefined
`contour` $\otimes$ `dur`	$\overline{e_{12}}$	$\langle 1, 8 \rangle$
`int` $\otimes$ `ioi`	$\overline{e_1}$	undefined
`int` $\otimes$ `ioi`	$\overline{e_{13}}$	$\langle -2, 8 \rangle$
`st`	$\overline{e_7}$	$(8, 12, 20, 24, 30, 32, 36)$
`int` $\otimes$ `ioi`	$\overline{e_1}$	$()$
`int` $\otimes$ `ioi`	$\overline{e_7}$	$(\langle 0, 4\rangle, \langle 7, 8\rangle, \langle -3, 4\rangle, \langle -2, 6\rangle, \langle -2, 2\rangle, \langle 0, 4\rangle)$
`contour`	$\overline{e_{16}}$	$(0, 1, -1, -1, -1, 0, 1, 1, -1, 1, 1, -1, -1, -1, -1)$
`int` $\oslash$ `fb`	$\overline{e_{16}}$	$(\langle 4, 12\rangle, \langle -4, 12\rangle, \langle 2, 12\rangle, \langle 5, 12\rangle, \langle -3, 12\rangle, \langle -4, 12\rangle)$
`int` $\oslash$ `fph`	$\overline{e_{11}}$	$(\langle 4, 48\rangle)$
`int` $\oslash$ `isq`	$\overline{e_7}$	$(\langle 0, 4\rangle, \langle 7, 8\rangle, \langle -3, 4\rangle, \langle -4, 8\rangle, \langle 0, 4\rangle)$

Figure 1: A fragment from the chorale *Aus meines Herzens Grunde*, with some example applications of the viewpoint element function (top) and the viewpoint sequence function (bottom) for various viewpoints. Start time (`st`) is represented as semiquaver ticks from time 0, and pitch as a MIDI number. The first event in this fragment starts at tick 8. The notation $\overline{e_n}$ is an abbreviation for a sequence of events $(e_1, \ldots, e_n)$. The viewpoint `contour` refers to melodic contour, and `int` to melodic interval. The `ioi` viewpoint is the inter onset interval between two events. The test viewpoints `fb`, `fph`, and `isq`, used to construct threaded viewpoints, are true if an event is the first in a bar, first in a phrase, or on a crotchet beat, respectively.

Viewpoint	Pattern	Occurrences
`int`	$\emptyset$	$2, 3, \ldots, 16$
`int` $\otimes$ `ioi`	$\langle 0, 4 \rangle$	$2, 7$
`dur`	$(4, 3, 2, 4)$	$3, 6$
`int` $\oslash$ `fb`	$\emptyset$	$4, 7, 10, 12, 14, 16$
`contour` $\oslash$ `fb`	$(\langle 1, 12\rangle, \langle -1, 12\rangle)$	$2, 10$

Figure 2: Examples of viewpoint patterns for the chorale fragment in Figure 1. Occurrences refer to the event number of the first event in the fragment where the pattern occurs. Note the use of the empty pattern $\emptyset$ which is defined anywhere that the viewpoint is defined.

in a random viewpoint sequence. These two quantities are defined here.

Consider a pattern P of length $l(P)$. In a single piece, there are $l(P) - 1$ positions where the pattern cannot possibly occur, because it would extend past the end of the piece. Therefore, in a data set of n pieces, there are $n(l(P) - 1)$ positions where the pattern cannot possibly occur. It follows that there are $\#(\emptyset) - n(l(P) - 1)$ positions where the pattern P might occur in the data set. The expected number of occurrences $E(P)$ of a pattern P in the data set is therefore

$$E(P) = p(P) \times (\#(\emptyset) - n(l(P) - 1)) \tag{4}$$

That is, the number of times we expect to see a pattern P is the probability $p(P)$ of the pattern multiplied by the number of positions where the pattern could occur. Probabilities of viewpoint patterns are computed using a blended zero- and first-order Markov model of viewpoint elements seen in the corpus.

Statistical significance. It is useful to report a p-value for a pattern; the probability that an equal or greater pattern score could arise within random viewpoint sequences. Patterns with high p-values will frequently occur in random viewpoint sequences and therefore are not interesting.

An exponential probability distribution is used to model pattern scores. This p-value of a pattern must be adjusted to reflect that fact that we are evaluating its significance not in isolation but within all patterns found in the corpus. This is called a Bonferroni adjustment, and reflects the probability of finding a pattern of equal or greater score within all patterns tested. Given a particular pattern score, an adjusted p-value is computed by multiplying it by an adjustment factor which is simply the total number of patterns evaluated for significance.

Pattern discovery algorithm. The pattern discovery algorithm (Figure 3) employs a suffix tree data structure, which compactly stores all suffixes and substrings occurring within a data string. The algorithm proceeds as follows. First, for a viewpoint selected by the analyst, every piece is transformed into a viewpoint sequence. Then, every suffix of this viewpoint sequence is incorporated into the suffix tree. This suffix tree is scanned to produce the set of all patterns that occur within at least k pieces (we use $k = 10$ for most of our results). The size of this set is the adjustment factor used to compute pattern p-values. The statistical significance of each pattern in this set is evaluated, and insignificant patterns are discarded.

Longest significant patterns. The output from the algorithm above can include many patterns that are significant yet contained within longer significant patterns. To handle this effect, we place all significant patterns into a *subsumption taxonomy* (Woods 1991). This is a directed graph where nodes represent patterns and links represent pattern containment (subsumption). In a sense, the subsumption taxonomy can be viewed as an expansion of the suffix tree; nodes become explicit patterns rather than viewpoint elements.

Following the construction of the subsumption taxonomy, the longest significant patterns are found at the leafs (nodes subsuming no other nodes) of the data structure.

(a) For a selected viewpoint, transform all pieces in the corpus into viewpoint sequences.

(b) Incorporate the viewpoint sequence for every piece into a suffix tree.

(c) Search the suffix tree, building the set of all patterns occurring in at least k pieces.

(d) Compute a p-value for each pattern, discarding insignificant patterns.

(e) Build a subsumption taxonomy from all remaining significant patterns.

(f) Report the leafs of this structure as the longest significant patterns in the corpus.

Figure 3: The multiple viewpoint pattern discovery algorithm.

voice	events	
	total	average
soprano	9226	50
alto	11361	61
tenor	11570	63
bass	11809	64

Table 1: The event composition of 185 Bach chorales; the total number of events in a voice, and the average number in each voice in a chorale.

The chorale data set. This study uses 185 Bach chorales, comprising a total of about 40000 events (Table 1). Sections annotated by a repeat are not expanded. Even so, this data set has some redundancy, in the form of some transposed chorale melodies and transposed reuse of phrase material. For this study we do not attempt to remove this redundancy.

3 Results

About 20 viewpoints were encoded; most of them pertain to melodic and rhythmic aspects of the chorales. Several viewpoints are test viewpoints that are used mainly for linking and threading with other viewpoints. We have also encoded some viewpoints which model harmonic or vertical structures. An extended set of viewpoints and results will appear in a longer

	number of patterns			
k	total	significant	longest	
2	11305	7263	275	(16.9)
5	1285	237	81	(7.1)
10	535	96	33	(6.1)
25	174	40	15	(5.1)
50	75	23	9	(4.6)
100	28	9	6	(3.7)

Table 2: Numbers of patterns found within the soprano line of 185 Bach chorales, using the melodic interval viewpoint. The second column refers to the number of raw, unfiltered patterns occurring in at least k pieces. The third column refers to the the number of statistically significant patterns at a p-value of 0.01. The last column refers to the number of patterns remaining at the leafs of the subsumption taxonomy. The average length of the longest patterns is indicated in brackets.

paper. Here, the adjusted p-value cutoff for patterns was set to 0.01. In all experiments, unless specified otherwise, the parameter k (the minimum number of pieces a pattern must occur in) was set to 10.

In this section we present some results obtained with the pattern discovery algorithm. Our most interesting results came from the linked and the threaded viewpoints, where we identified patterns that captured deeper structure of the music. The patterns presented in Figure 4 are among the highest scoring patterns that were discovered for the particular viewpoints.

General behavior of the algorithm. Table 2 illustrates the behavior of the algorithm as a function of the parameter k (the minimum number of pieces a pattern must occur in). At lower values, the method discovers many patterns. The filtering effectiveness (from total patterns to longest significant patterns) can be as high as 98%. At $k = 2$ many long patterns are found; most of these long patterns arise from redundancy with the corpus. As k increases, the patterns found are shorter, as they are required to occur in more pieces. Even with $k = 100$, some significant patterns are found.

Melodic intervals in soprano and alto voices. For the soprano line, 33 longest significant melodic interval patterns were found. For the alto line, 29 were found. Referring to example 4(a), it is quite common in the soprano to have stepwise movement. The initial rising fourth suggests a harmonic progression of V-I, and it is likely that the I is on the strong beat.

Example 4(b) demonstrates a familiar feature of alto lines: flat melodic lines that serve mainly to fill in the harmony. The semitone movement suggests a leading note to tonic succession. A leading note is restricted: it can usually rise up to the

Figure 4: Some multiple viewpoint patterns discovered in the 185 Bach chorales. Each block illustrates an instance of a pattern, with its viewpoint, total count, piece count, voice, and pattern. An event is presented as a semibreve if its duration is not determined by the viewpoint pattern. The viewpoint `pcint` is the pitch class interval viewpoint. The dots between the crotchets in the threaded viewpoint signify that quavers or even semiquavers may occur between the indicated pitches. All viewpoint patterns are invariant under transposition.

tonic, or occasionally drop a third (in the middle voices). Here we have an example of both.

Melodic contour. A single significant melodic contour pattern, example 4(c), was discovered by our algorithm. This is a long line, spanning 12 events, that occurs within 10 pieces. It is of interest that the interval pattern of 4(a) is a specific instance of a portion of this contour pattern.

Pitch class intervals for bass. For the bass line, we used a pitch class interval viewpoint. A total of 57 longest significant pitch class interval patterns was found. Pattern 4(d) is an example of harmonic movement — the end of the segment shows a potential perfect cadence.

A well known common pattern in the bass line of the chorales is a chromatic stepwise movement, which is mentioned in most Bach chorale composition books. In example 4(e) we have found a pattern for this phenomenon. This pattern occurs 19 times, within 12 different pieces.

Linked interval and duration. For linked viewpoints, we were able to combine different parameters to see how these are related in the music. For example, for a linked viewpoint between melodic intervals and duration, we found 23 longest significant patterns. Figure 4(f) is an example of a leap in the soprano followed by rapid stepwise movement of the opposite direction, presumably to counterbalance potential singing mistakes in a congregation.

Threaded viewpoints. An example of a threaded viewpoint is one which describes the melodic interval at crotchet beats. A total of 335 patterns was discovered, and after the subsumption filter 17 remained. Figure 4(g) shows one threaded pattern. Occurrences of this pattern can include quavers or semiquavers in between the crotchets. These can be passing notes (at the last interval), consonant skips, échapées, cambiatas, and suspensions.

Threaded patterns are a step closer to the metric reduction or deeper structure of a work in the Schenkerian sense (Forte and Gilbert 1982). However, a metric reduction is more complicated than the process of the threaded viewpoint: for example, in case of a suspension, the harmonic note might not be on a crotchet beat. In that case, applying pattern discovery to a metric reduction of the score would yield better results.

4 Discussion

Pattern discovery in music is a difficult problem. Making truly new discoveries is rare, but computational techniques can contribute. This paper has presented a new formalism for describing musical patterns and a new algorithm for discovering them. The computational approach employed is to look for patterns which occur much more frequently than expected. The use of p-values for patterns can separate the truly significant patterns from statistical background. These significant patterns can be explored further by the music analyst.

There have been various approaches to automatic pattern discovery in music. Most approaches focus their analysis on a single piece for patterns (Cambouropoulos 1998; Hsu et al. 1998; Meredith et al. 2001) and are not directly applicable to the analysis of a large number of pieces. Though an artificial single piece might be constructed for these methods by joining several pieces together, since they have running times of a polynomial order in the length of the piece they may not be practical for the analysis of hundreds or thousands of pieces.

An approach that can naturally find patterns in two pieces is known as dynamic programming (Mongeau and Sankoff 1990). In this technique musical similarity is encoded into a distance function, pairs of transposed melodies are compared, and the common pattern is the trace of aligned elements. Iterations of this pairwise comparison are necessary in order to find patterns occurring within more than two sequences. By contrast, in our approach, knowledge is encoded into discrete modules, the viewpoints. Patterns are found not in a surface representation but rather in a deeper transformed representation. Rather than looking at similarity or partial similarity in the score, we shift the problem into the representation level, and look for identity. An identity at one or more viewpoints results from a similarity (of varying degree) in the music. Furthermore, since we seek identities within a transformed representation, our algorithm is computationally efficient and will find all of the patterns in a corpus.

Pattern discovery algorithms can produce voluminous output which must be filtered for both statistical and musical significance. This is usually done by preferring the longest, most frequent patterns. However, the properties of pattern length and frequency are inversely related, because longer patterns tend to occur less frequently. Balancing these two properties in a single measure is the essence of evaluating a pattern. Cambouropoulos (1998) uses a function of the three variables of pattern length, pattern frequency, and pattern overlap to rank patterns. The parameterization of the equation involving these three variables is performed manually by the investigator. Hsu et al. (1998) calibrate a minimum acceptable pattern length by running the method on synthetic random melodies.

It can be demonstrated that the pattern score of Equation 3 balances the two properties of frequency and length in a single measure, and avoids the need for pattern length threshold specification. It follows from Equation 3 that for a pattern P, if its frequency $\#(P)$ remains constant while its length increases, its expected frequency $E(P)$ will decrease while its score will increase. On the other hand, if its length remains constant while its frequency $\#(P)$ increases, its score will increase because $E(P)$ will remain constant.

In Nattiez' (1975) two paradigmatic analyses of Debussy's

Syrinx, we observe the need for longest significant patterns (first analysis) and most general patterns (second analysis). Both types of patterns are useful for further music analysis. Nattiez permits pattern overlapping in exceptional circumstances, when it is felt that both patterns are equally important and belong to different classes. In our method, pattern overlapping is allowed only when the overlapping patterns are not covered by a longer significant pattern.

An approach to musical style recognition (Westhead and Smaill 1993) and generation (Cope 1987) is to use a catalog of *signatures* that cover instances of the style. An interesting application of our method is to produce the *most general consistent patterns* occurring within a musical corpus. Consistency can be defined with respect to positive and negative examples of the style. The most general consistent patterns will be more useful than the longest significant patterns for the task of style recognition, as they are more frequent in the musical corpus and much more likely to occur in new, unseen examples of the style. For the task of style generation, general patterns are less likely to be recognizable as fragments from the pieces used to define the patterns.

Patterns are statistically significant if they are surprising with respect to a background model. Therefore, the closer the background model is to the style under consideration, the more subtle and interesting the discovered patterns will be. In this study we have used a fairly primitive Markov model as a background model. Alternatives to this are to parameterize the Markov model on another style, or on another voice within the corpus.

Although Bach chorales have traditionally been treated as exemplary harmonic sequences of a homophonic texture, our results show that voice-leading techniques are just as important as in the other works of J.S.Bach. Our model is especially suitable for teaching purposes in that it can contribute information on Bach chorale composition by the production of significant sequential patterns of the various viewpoints.

In summary, this paper has presented a new approach to pattern representation and discovery which is particularly well suited to various music analysis purposes. Based on the multiple viewpoint formalism, it produces explicit viewpoint patterns rather than similarity judgments of note patterns. Music is transformed into viewpoint sequences. An efficient suffix tree data structure is used to rapidly discover all patterns. A statistical method is used to restrict attention to only those patterns which occur much more frequently than expected. The significant patterns are organized into a subsumption taxonomy, and the longest significant patterns in a corpus are found at the leafs of the structure. The method presented here can be used to rapidly enumerate all patterns within a large corpus.

Future work will include application of the pattern discovery method to harmonic aspects of music, and a more extensive analysis of patterns discovered for multiple viewpoints. The interactions between melodic and vertical viewpoints will be used to provide further interesting insights to the corpus of the Bach chorales.

References

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

Conklin, D. and I. Witten (1995). Multiple viewpoint systems for music prediction. *International Journal of New Music Research 24*(1), 51–73.

Cope, D. (1987). An expert system for computer–assisted composition. *Computer Music Journal 11*(4), 30–46.

Forte, A. and S. Gilbert (1982). *Introduction to Schenkerian Analysis*. W.W. Norton.

Hsu, J.-L., C.-C. Liu, and A. Chen (1998). Efficient repeating pattern finding in music databases. In *Proc Int Conf on Information and Knowledge Management*, pp. 281–288. Association of Computing Machinery.

Meredith, D., G. Wiggins, and K. Lemström (2001). Pattern induction and matching in polyphonic music and other multidimensional datasets. In *Proc SCI 2001: Conference on Systemics,Cybernetics and Informatics*.

Mongeau, M. and D. Sankoff (1990). Comparison of musical sequences. *Computers and the Humanities 24*, 161–175.

Nattiez, J.-J. (1975). *Fondements d'une Sémiologie de la Musique*. Union Générale d'Editions.

Rolland, P.-Y. and J.-G. Ganascia (2000). Musical pattern extraction and similarity assessment. In E. Miranda (Ed.), *Readings in Music and Artificial Intelligence*, Chapter 7, pp. 115–144. Harwood Academic Publishers.

Smaill, A., G. Wiggins, and M. Harris (1993). Hierarchical music representation for composition and analysis. *Computers and the Humanties 93*, 7–17.

Westhead, M. and A. Smaill (1993). Automatic characterisation of musical style. In *Music Education: an AI approach*, pp. 157–170. Springer-Verlag.

Woods, W. (1991). Understanding subsumption and taxonomy: A framework for progress. In J. F. Sowa (Ed.), *Principles of Semantic Networks*, pp. 45–94. Morgan–Kaufmann.