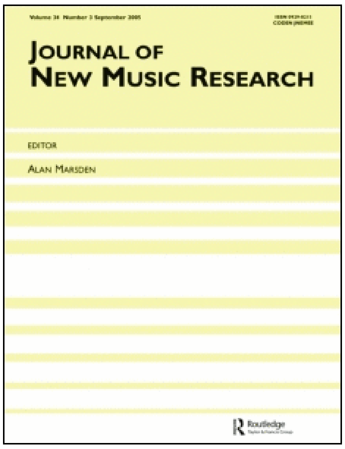


Journal of New Music Research
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713817838

In Search of Harmony: A Constraint-Based Approach to the Task of Harmony Retrieval

Tamar Berman[a]
[a] University of Illinois at Urbana-Champaign, USA

To cite this Article Berman, Tamar(2007) 'In Search of Harmony: A Constraint-Based Approach to the Task of Harmony Retrieval', Journal of New Music Research, 36: 4, 301 — 312

To link to this Article: DOI: 10.1080/09298210801927820

URL: http://dx.doi.org/10.1080/09298210801927820

Journal of New Music Research
2007, Vol. 36, No. 4, pp. 301 – 312

Routledge
Taylor & Francis Group

In Search of Harmony: A Constraint-Based Approach to the Task of Harmony Retrieval

Tamar Berman

University of Illinois at Urbana-Champaign, USA

Abstract

This paper describes a system prototype of a search engine for harmony sequences. The system, which was successfully tested on midi sequences of music by W.A. Mozart, has several distinguishing features. These include: (a) a user-friendly interface for specification of the sequence through pitch and time constraints, (b) support for the retrieval of sequences comprised of complex events containing both melodic and harmonic features, which may be interspersed by non-sequence data, and (c) pitches are stored as categorized distances from an anchor point, resulting in immediate, online calculation of tonal functions. A method for the representation and retrieval of harmony sequences will be described. This method makes use of the music theoretical concepts of schema and style structure, and has been successfully applied to the tasks of schema retrieval, similarity retrieval, theme retrieval and data mining.

1. Introduction

The task of musical pattern matching has been extensively studied, and is the focus of a substantial body of research in Computational Musicology as well as Music Information Retrieval (MIR). The problem of retrieving instances of a query melody has been solved in many ways: N-grams (Downie and Nelson 2000), Markov models (Birmingham et al., 2001) and string matching techniques (McNab et al., 1996) have all been successfully applied to this task. However, the parallel problem of harmony sequence retrieval is not yet fully solved, due to the complexity of the underlying issues and the difficulty in extending melody search techniques to this multi-dimensional problem. Works such as Huron's Humdrum toolkit (1999), Doraisamy and Ruger (2003), Pickens (2004) have made significant contributions to tackling this problem, yet the solutions available today are not yet on par with those that exist for melody search. For example, melody search systems that operate on symbolic data do not have a serious problem in ascertaining that a match exists at the individual event level: recognizing a particular pitch or interval in the data is usually a straightforward task, whereas determining that a certain tonal function is present in polyphonic music can sometimes be a perplexing task, even for skilled human listeners.

Another body of research looks at the problem of music similarity, which is the retrieval of musical pieces that are similar to a given query. Several different types of similarity have been studied. For example, Foote (1997), Logan and Salomon (2001), Pampalk (2006) and others have described methods for audio-based similarity assessment that are based on timbre descriptors. Here, pieces are compared at a "macro" level (i.e. are the pieces overall similar to each other), resulting in classifications that have considerable overlap with genre classifications.

Mardirossian and Chew (2006) describe a method for similarity assessment of the tonal behaviour of musical pieces based on key histograms and average time spent in each key, which in turn is based on previous work by Tzanetakis et al. (2002). This study differs from the present one in that it deals with similarity at the musical piece level rather than phrase level.

Recently, a number of researchers have addressed the problem of cover song identification: Ellis and

Correspondence: Tamar Berman, Graduate School of Library and Information Science, University of Illinois at Urbana-Champaign, 501 E. Daniel St., Champaign, IL 61820 USA. E-mail: tamar@berman.tk

DOI: 10.1080/09298210801927820 © 2007 Taylor & Francis

Poliner (2007), Serrà and Gómez (2007), Bello (2007) and Marolt (2006) describe systems that have the capability of retrieving different versions of the same song from audio data. This type of retrieval addresses phrase-level similarity, and is similar to the melody search systems in the sense that the retrieved sequences are ideally identical to the query, with a possibility of slight mismatches. This is expressed, for example, in the minimum-edit-distance criteria used by many of these systems. While this type of search is useful for many purposes, the model described here approaches the similarity problem from a different perspective.

By allowing the user to specify which features of the original phrase need to be preserved in the retrieved sequences, the system allows for pieces that are quite different in their details yet share important features to be considered similar. The system poses a strict requirement that the user-specified features be present, but is flexible regarding the other contents of the phrase. Metaphorically stated, this system is interested in retrieving "sibling" phrases and not only "clones" of the query sequence.

This paper will present a representation and retrieval method for musical sequences. In this method, pitches are stored as categorized distances from an anchor point, and sequences are described as constraints on pitch and time. In addition, by evoking the music-theoretical concepts of schema and style structure it will be shown that the problems of harmony retrieval and phrase-level similarity can both be greatly assisted by schema theory. We will therefore proceed with a definition of these terms.

In his book "Explaining Music", music theorist, philosopher and historian Leonard Meyer writes:

> "Archetypical patterns and traditional schemata are the classes – the rules of the game . . . in terms of which particular musical events are perceived and comprehended. No melody, however original and inventive, is an exception to this principle"
>
> (Meyer 1973, p. 213)

In "*Beyond Schenkerism*" music theorist Eugene Narmour discusses similar concepts:

> "in order to discover the system of significant traits which defines the stylistic language, the style analyst will abstract from the repertory at hand a lexicon of *style forms* . . . style forms may be defined as those parametric entities which achieve enough closure so we can understand their functional coherence without reference to the specific . . . contexts from which they come . . . In order to avoid creating an ocean of lifeless facts . . . the style analyst will attempt to restore the syntactic function of style forms by arranging them in various specific contexts according to their statistically most common occurrences. The contexts which result from such arrangement can be called *style structures*"
>
> (Narmour 1977, pp. 173–174)

Schemas and style structures are essentially the building blocks of musical style. The reader who is familiar with object theory may conceive of the schemas as the "classes" of the style; the instances of each such class are the individual musical passages created by a composer writing in the style.

The representation model described here is designed to facilitate the retrieval of style structures and schemas. It has the capability of accepting a parametric definition of constraints on pitch and time, defined by a "style analyst", and retrieving instances of the structure from a music database. Interestingly and not surprisingly, the same mechanism can be used to retrieve musical passages that are similar to each other, due to a shared structure defined by the constraints. Stated differently, the constraints define the musical class, so all musical passages that satisfy the constraints and therefore belong to the class will be similar to each other. The more constraints specified, the greater the similarity.

This paper will describe the model and its application to schema retrieval, similarity retrieval, theme retrieval and data mining. Performance measures will be presented, and a user interface described. Finally, the potential for applying the model to audio data and integrating it with alternative representations will be explained.

2. The model

In the data model underlying the system, music is conceived as an equally spaced time series of 12-dimensional vectors, representing the twelve pitch classes. A *time series* is a set of ordered observations on the value of one or more variables, taken at successive points in time. The observations may be equally spaced in time, as they are in this model. Each vector in the series describes the harmony content of the time interval corresponding to it, and is therefore called a *harmonic window*. For example, the harmonic window which starts at 2 s into the piece and ends 3 s into the piece will describe, for each pitch class, whether it is present or absent during the third second of the piece, and what role it plays within that time frame.

A pitch class is considered "present" in a harmonic window if there is any overlap between the time interval in which the corresponding note sounds and the time interval described by the window. Time intervals are associated with harmonic windows on the basis of two parameters: *onset interval* defines the time that elapses between window onsets and is somewhat analogous to the sampling rate used in audio files, as it defines how often a "harmonic sample" is taken and recorded as a harmonic window.

Window length describes the length of the time interval associated with each vector, and determines how close pitches need to be to each other in order to be considered simultaneous: a very small window implies that pitches must be played together, or almost together, in order to be in the same window, whereas a wide window allows for successively sounding pitches to be grouped together. The window length should not to be confused with the length of the musical event: events with longer duration will occupy more consecutive windows than events with short duration. The *role* is indicated by a value from the set {0, 1, 2, 3}, where 0 indicates absence, 2 indicates presence as bass, 3 indicates presence as top voice and 1 indicates presence which is neither top nor bottom. The top voice is defined as the class of the highest pitch in the window, and the bass is defined as the class of the lowest pitch in the window.

The time series transformation is preceded by a key-finding phase, in which the key present in the first few measures of the piece is determined, and a transposed copy is created from each musical piece: pieces beginning in a major key are transposed to begin in C major, whereas pieces beginning in a minor key are transposed to begin in A minor. These transposed copies are subsequently translated into time series. This ensures that a user specification such as "tonic in root position" will indeed find all tonics, regardless of key. Another way of viewing this process is as a conversion of all pitches in the piece to pitch classes, and these to distances from anchor, the anchor being the key in which the piece begins.

It should be emphasized that this entire calculation takes place only once per piece, when it is added to the database. When users pose queries to the system they may specify the query in their key of choice; the system runs the query against the distance-from-anchor data, and returns the target sequences in their original, un-transposed key.[1] The transposed time series serves as a normalized index to the data and facilitates the search.

3. Example

Table 1 shows records of notes in the database, obtained from the midi files by merging "note on" events with corresponding "note off" events.[2] Note onset times and durations are given in seconds. For a window length of 1 s and onset interval of 0.5 s, these records translate into the harmonic windows of the time series shown in Table 2.

Table 1. Note records in the database.

Onset time	Duration	Pitch class	Octave	Channel
1.250	0.461	G	2	1
1.719	0.148	E	2	1
1.875	0.930	C	3	1
2.500	0.617	G	1	2
2.500	0.617	E	1	2
2.812	0.062	D	3	1
2.891	0.062	C	3	1
2.969	0.062	B	2	1

Table 2. Harmonic windows in the time series.

Onset time	C	C♯	D	D♯	E	F	F♯	G	G♯	A	A♯	B
0.5	0	0	0	0	0	0	0	3	0	0	0	0
1.0	3	0	0	0	2	0	0	1	0	0	0	0
1.5	3	0	0	0	2	0	0	1	0	0	0	0
2.0	1	0	3	0	2	0	0	1	0	0	0	1

Table 2 shows that only G was present in the window [0.5,1.5], the windows [1,2] and [1.5,2.5] contained C, E and G with C being the top voice and E being the bass, and the window [2,3] contained C, D, E, G and B, with D being on top and E acting as bass. For windows containing a single pitch – such as [0.5,1.5] in this example – it is tagged as the top voice.

Sequence retrieval is implemented through SQL queries on the time series table and the notes table in the database. Each query is generated dynamically from constraints specified by the user, as will be demonstrated in subsequent sections.

4. Schema retrieval

The 1-7 . . . 4-3 is a style structure, or *schema* which was prevalent in 18th century western music. It has the following features.

(1) Four events, consisting of two event pairs that are several seconds apart.
(2) In the first pair, the melody descends from the 1st degree to the 7th. The harmony shifts from I to V while the bass moves from 1 or 3 to 2, or descends from 1 to 5.
(3) In the second pair, the melody descends from the 4th degree to the 3rd. The harmony shifts from V to I while the bass moves from 5 or 7 to 1.

A comprehensive discussion of this structure and its history is given by Gjerdingen (1988). This pattern was selected as a test case for the performance of the system, as it consists of complex events which have harmonic as well as melodic features, and these events may be

[1] To simplify reading of this paper, target sequences are shown transposed to C and not in their original keys.

[2] This was performed using the POCO program (Honing 1990).

interspersed or occur simultaneously with other patterns or occurrences.

Retrieval was performed on a database compiled from 505 midi sequences of music by W.A. Mozart obtained from Classical Music Archives. These included symphonies, piano sonatas, piano concertos, other concertos and piano trios.

To facilitate retrieval of the 1-7 . . . 4-3 structure, the following two queries were created.

4.1 The top voice query

The sequence contains four events:

(1) the first event includes pitches C, E, G, with C on top;
(2) the second event includes pitches G, B, D with B on top;
(3) the third event includes pitches G, B, D, F with F on top;
(4) the fourth event includes pitches C, E, G with E on top.

The maximum duration of the sequence is limited to five seconds. The maximum distance between the first two events is half a second. The maximum distance between the last two events is half a second. These parameters were chosen by a musician familiar with this schema who was using the system.

The pitches in this example and the other examples in this paper are given relative to an anchor key of C Major. Section 8 will demonstrate how a user could describe this sequence in a different key or as key-neutral scale degrees. The system will retrieve all instances of the described structure regardless of the key in which they are written.

Figure 1 shows an example retrieved by this query, following its translation to SQL and submission to the database. The example is taken from the *Allegro con brio* in Mozart's Symphony No. 25 in G Minor, K183. This is a clear, homophonic example of the 1-7 . . . 4-3 structure. The annotation is a possible user interpretation to the information provided by the query. The query itself returns pointers to the locations, in seconds, of each instance of the pattern within the midi file.[3]

4.2 The channels and bass query

The sequence contains four events, described relative to C major:

(1) the first event includes pitches C, E, G, with E in the middle or bass and G in the middle or top;

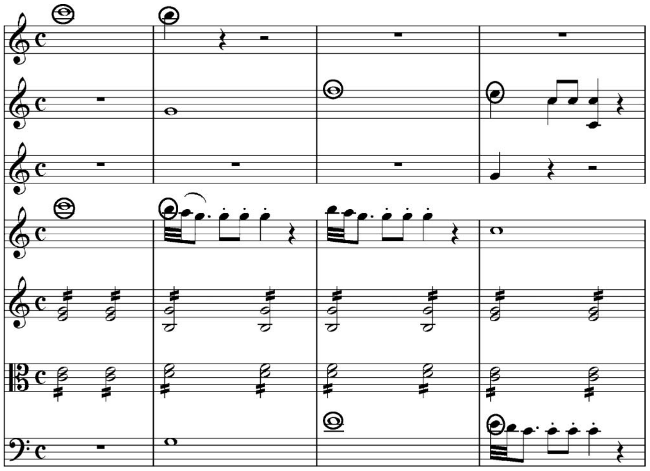


Fig. 1. Mozart Symphony No. 25 in G Minor, K.183, Allegro con brio, bars 30–33.

(2) the second event includes pitches G, B, D with D in the middle or bass and B in the middle or top;
(3) the third event includes pitches G, B, D, F with G and B in the middle or bass, and D and F in the middle or top;
(4) the fourth event includes pitches C and E with C as bass and E in the middle or top. G may not act as bass or top voice.

In the first two events, the melodic line C → B must play along a single channel. In the last two events, the melodic line F → E must play along a single channel. The maximum duration of the entire sequence is limited to five seconds. The maximum distance between the first two events is half a second. The maximum distance between the last two events is half a second. As before, this parameter setting was chosen by a musician familiar with this schema who was using the system.

Figure 2 shows an example retrieved by this query, from the *Rondo – Allegro vivo* of Mozart's Violin Concerto No. 2 in D, K211. The annotation highlights the melodic line C → B . . . F → E. This is a more complex, polyphonic case of the schema. It is initially heard in bar 14: C is played over I on the first beat, and B is played over V7 on the second beat. The F → E is delivered by the horn in bar 15, and repeats in bar 16. The Oboes play their own version of the schema in bars 16 and 17: the first Oboe provides the C and B, while the second plays the F and E. The system identifies this, too, as a "schema location", because the required melody line along with the required I → V7 → I background are present. The schema plays again in bars 18–20: the violas play the first half (C → B) in bars 18–19; the solo violin plays the second half (F → E) in bars 19–20.

[3] Bar numbers refer to the locations within the midi files as they appear in http://www.classicalarchives.com.

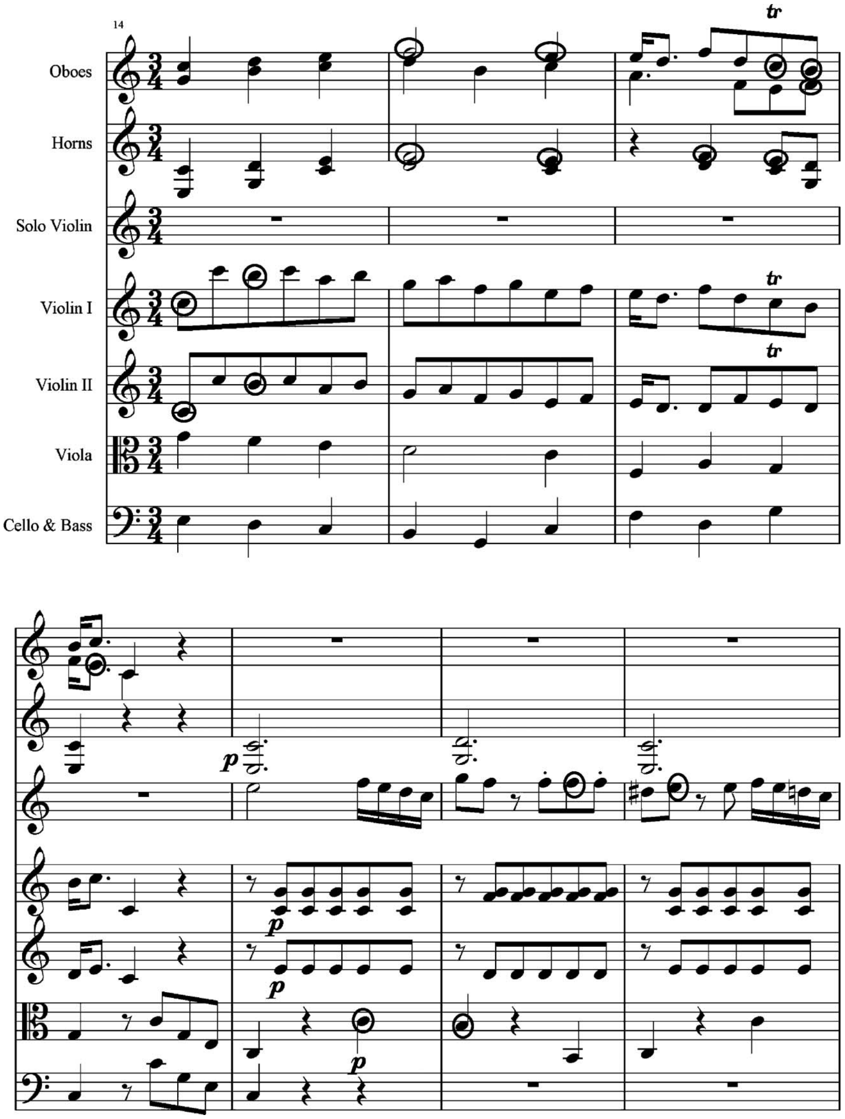


Fig. 2. Mozart Violin Concerto No. 2 in D, K.211, Rondo Allegro vivo, bars 14–20.

5. Similarity retrieval and theme retrieval

Musical similarity can be perceived along several dimensions, including melody, harmony, rhythm, timbre and lyrics. The relationship between a theme and its variations is similarly expressed in a variety of ways: some instances preserve melody, some preserve harmony, others preserve rhythm and some preserve all of these aspects.

Similarity retrieval is defined here as the task of retrieving phrases whose harmony is similar to that of a given query sequence. Theme retrieval is defined as the task of retrieving instances of a given theme that preserve the harmony of the theme. Theme retrieval can therefore be viewed as a special case of similarity retrieval, in which the theme acts as the query sequence. In this model, these retrievals are performed much like schema retrieval: the user specifies a list of constraints that describe the structure of the phrase or theme. The system generates an SQL query and retrieves passages that satisfy the constraints.

5.1 Example

Figure 3 shows the first theme in the first movement of Mozart's clarinet concerto in A, K622. A music analyst singled out eight structural events, and offered the following schematic description:

(1) the first event includes pitches C, E, G with G on top;
(2) the second event includes pitches C, E with E on top;
(3) the third event includes pitches F, A, C;

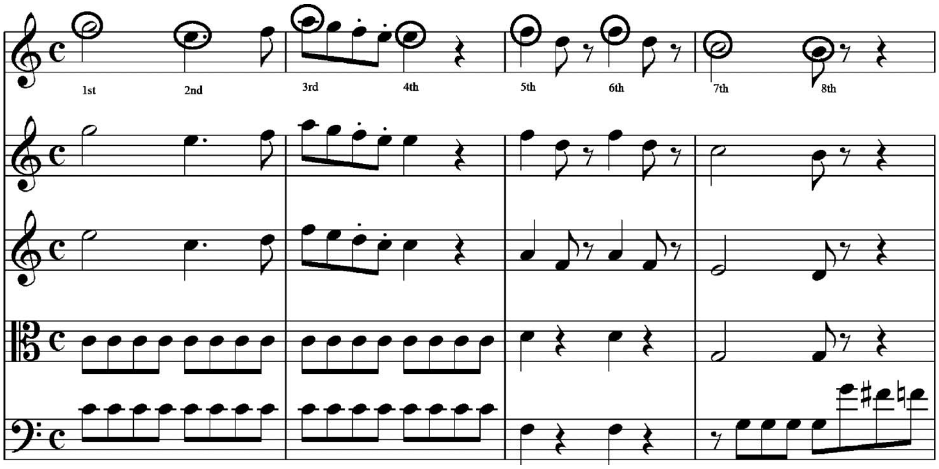


Fig. 3. Mozart Clarinet Concerto in A, K622, Allegro, bars 1–4.

(4) the fourth event includes pitches C, E;
(5) the fifth event includes pitches D, F, A;
(6) the sixth event includes pitches D, F, A with F on top;
(7) the seventh event includes pitches C, G;
(8) the eighth event includes pitches G, B, D, F;
(9) the maximum duration of the sequence is 15 s.

Figure 4 is an example retrieved by the system in response to this specification. It shows the first 12 bars of the third movement (Rondo) from Mozart's Piano Concerto No. 6 in B♭, K238. Two sequences which satisfy the constraints are presented at the beginning of the Rondo. Despite obvious differences in melody and even harmony, the reader may verify that this passage is very similar to the clarinet theme. Additional discussion and examples of theme retrieval are given by Berman et al. (2006).

6. Data mining

This representation model has also been applied to the task of mining the music database for recurring harmony sequences. A full discussion can be found in Berman (2006) and will be briefly mentioned here. The algorithm for sequence discovery, proposed by Agrawal and Srikant (1995) discovers frequently recurring sequences in a time series of vectors. This algorithm was applied to the musical time series data. Table 3 shows examples of sequences and their corresponding support levels in the database. The *support* of a sequence is the fraction of musical pieces in the database that contain it. As in the search system, instances of sequences are identified even if they are interspersed with other data. The task of extracting frequently recurring musical sequences has been performed in other studies: Bergeron and Conklin (2007) apply the aforementioned sequence mining algorithm to the study of multi-featured patterns in melodies by Brassens. Liu et al. (1999) and Meek and Birmingham (2001) describe alternative methods for the extraction of recurring musical patterns.

7. System performance

The system was tested on a database compiled from 505 midi sequences of music by W.A. Mozart obtained from Classical Music Archives. These included symphonies, piano sonatas, piano concertos, other concertos and piano trios. Each of the 505 sequences was transposed to begin in C major or A minor, following a human assessment of the initial key of each piece. This was done to ensure that results did not depend on the performance of a particular key finding algorithm. Consequently, the reader may assume that for very large databases where human key assessment is not possible, results may be somewhat inferior to those reported here, depending on the quality of the key-finder employed. For small data sets, or for databases where new pieces are added infrequently, it is perhaps best to perform individual human assessments of the initial key. As explained above, this is a one-time operation per piece, made when it is added to the database, and is not repeated for every query posed to the system. One may legitimately wonder why insist on scaling the data to a common ground, if the system could easily search for the sequence in all 12 keys? The reason for this is that for potential users of the system, finding the sequence in a key that has a significant role in the piece is a substantially different task from finding it in "any key". This will be elaborated upon in Section 8.

Retrieval of the 1-7...4-3 schema was performed using several different window lengths and onset

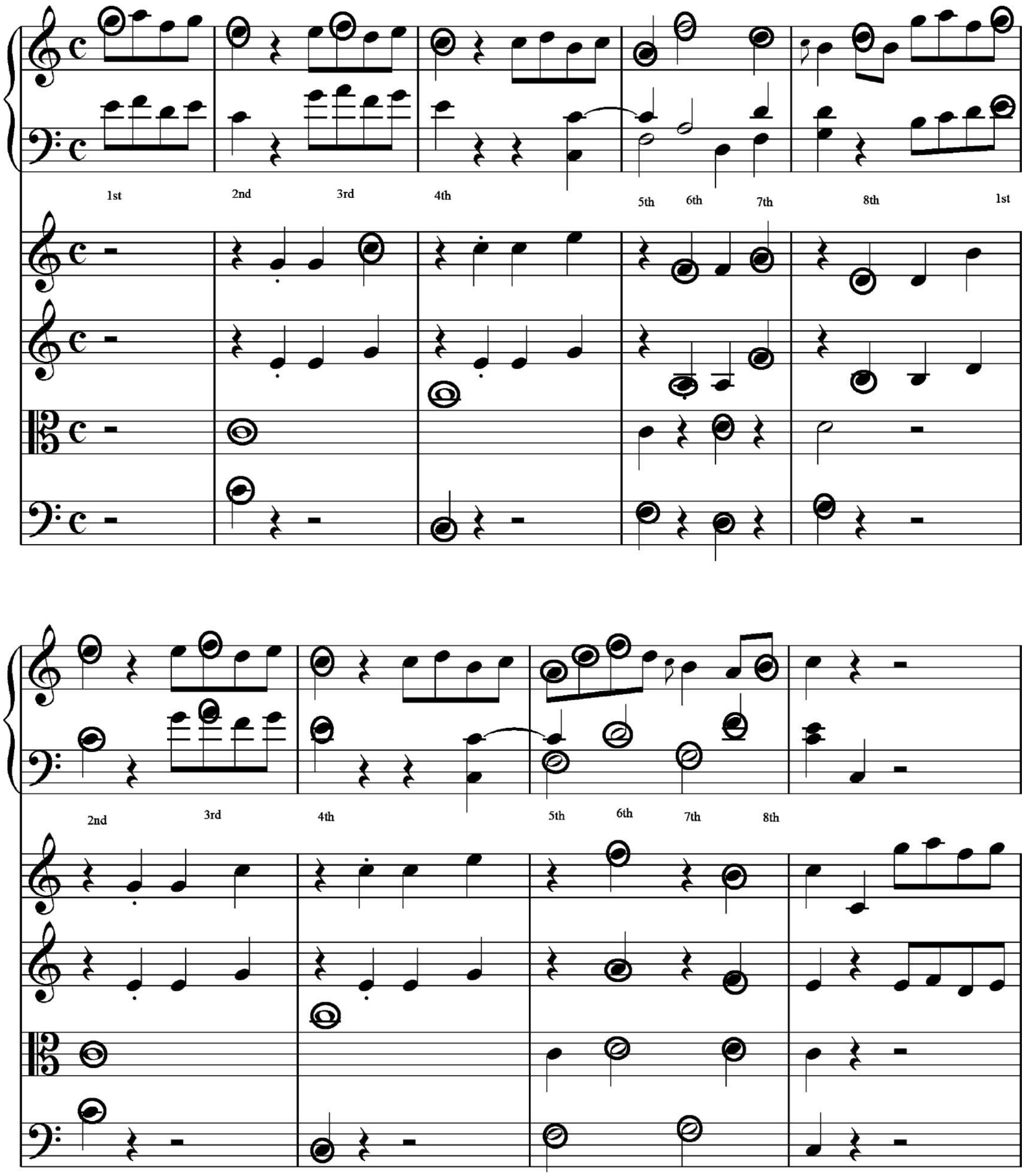


Fig. 4. Mozart Piano Concerto No. 6 in B♭, K238, Rondo, bars 1–8.

Table 3. Sequence support.

Database sequence	Harmony	Support
C&E&G → G&B&D	I → V	96.83%
G&B&D → C&E&G	V → I	97.03%
C&E&G → G&B&D → G&B&D → C&E&G	I → V → V → I	58.61%
C&E&G → F&C → G&B&D → C&E&G	I → IV → V → I	56.63%
C&E&G → D&F&A → D&F&A → C&E&G	I → ii → ii → I	41.58%

Window length = 0.5 s.
Onset interval = 0.25 s.
Maximal sequence duration = 5 s.

Table 4. Performance metrics.

Query type	Window length	Onset interval	Precision	Recall
TV	1.000	0.500	0.632	0.429
CB	1.000	0.500	0.282	0.857
TV	0.500	0.500	0.875	0.250
CB	0.500	0.500	0.500	0.464
TV	0.500	0.250	0.733	0.393
CB	0.500	0.250	0.317	0.714
TV	0.250	0.250	0.778	0.250
CB	0.250	0.250	0.538	0.500
TV	0.250	0.125	0.538	0.250
CB	0.250	0.125	0.333	0.571
TV	0.125	0.125	0.857	0.214
CB	0.125	0.125	0.600	0.536
1 vote	N/A	N/A	0.243	1.000
6 votes	N/A	N/A	0.833	0.536
7 votes	N/A	N/A	1.000	0.321

$\text{Precision} = \frac{\text{(number of correct retrieved instances)}}{\text{(number of retrieved instances)}}$.
$\text{Recall} = \frac{\text{(number of correct retrieved instances)}}{\text{(number of correct instances in dataset)}}$.

intervals, as shown in Table 4. Precision and recall were calculated on a subset of the data: 32 pieces containing 115 retrieved passages. This is due to the difficulty of obtaining rating by human listeners for all of the instances contained in the corpus. These 115 passages were reviewed by three musicians, to determine whether or not they constituted true examples of the schema. The review included auditory evaluation and score analysis. A retrieved passage was determined to be a valid instance of the schema if a majority of the musicians – at least two – rated it as such.

The evaluated queries included simple and aggregate queries. The first 12 queries listed in Table 4 are based on the previously defined "Top Voice" and "Channels and Bass" queries, and differ from each other in the settings for window length and onset interval. Recall that Top Voice (TV) queries required that the melody be in the top voice, whereas Channels and Bass (CB) queries placed requirements on the middle and bottom voices and required channel consistency.

The last three queries in the table are based on the previous 12 as follows: the "1 vote" query returns an instance if it was selected by any of the 12 simple queries, the "6 votes" query returns an instance if it was selected by at least 6 of the 12 simple queries (50% rule) and the "7 votes" query returns an instance if it was selected by at least 7 of the 12 simple queries (majority vote).

This data yields several observations; first, it appears that TV queries have better precision, whereas CB queries have better recall. Second, window overlap appears to improve recall, whereas no overlap appears to improve precision. Finally, wider windows improve recall, and optimal precision was obtained at 0.5 s windows.

There is a clear tradeoff between precision and recall. This is understandable if we remember that the judgment made is that of class membership: the more requirements posed to candidates, the more homogenous the group and the fewer members in it. This tradeoff, taken with the high precision achieved by the majority vote query, suggests that perhaps a ranked rather than boolean solution should be employed, i.e. that results be ranked based on the degree to which the candidates satisfy the specified constraints. This will be considered in future development of the system. Further improvement could be made by enabling the specification of additional types of conditions such as constraints on metric placement.

As is the case with interactive search systems, the quality of the results depends to a great extent on the quality of the definitions provided by the user. The performance data indicates that for a well-defined, well-studied test case such as the 1-7 . . . 4-3 schema, the system is capable of producing high-precision results.

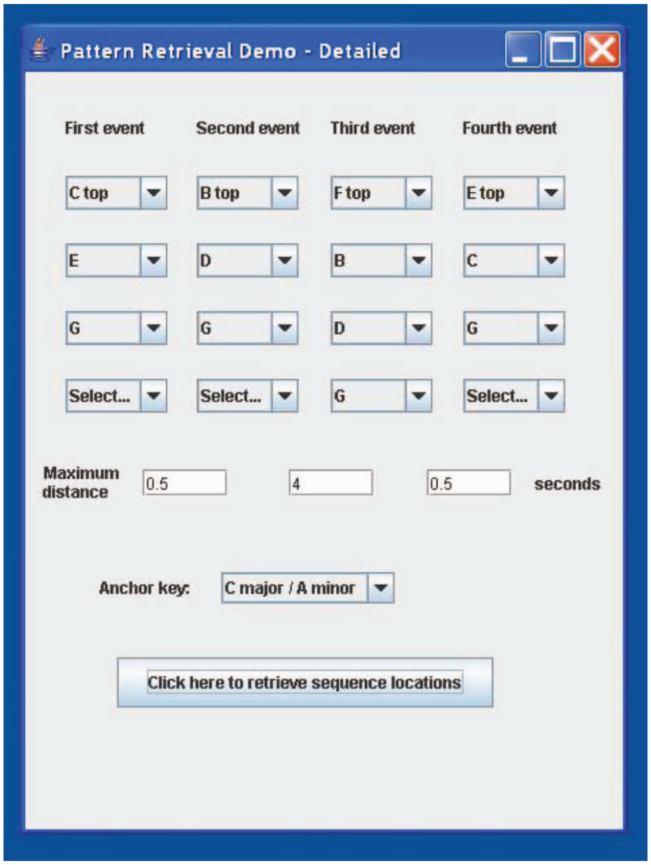
A point of special interest is the optimal precision achieved with half-second-long windows. Previous research in human perception has indicated that for auditory stimuli to be perceived as a united cluster, they should succeed each other at intervals no shorter than 0.3 s and no longer than 0.5 s (Wundt 1874). This finding is compatible with the optimal parameter setting for the system.

For small window lengths and onset intervals, events associated with "good" examples showed up in many adjacent harmonic windows in the result sets. Events which were reported by a single window in a given query were usually coincidental, did not surface in other queries and were rated as poor by the musicians. This characteristic could be used to assign higher ratings to schema events which show in many harmonic windows as compared with events that are recorded by few windows. Another use of this would be to allow the user to specify a constraint on the minimum length of the musical event itself, and not only on the distance between events.

8. User interface

Figure 5 illustrates how a user may describe the 1-7 . . . 4-3 structure using pitch names in the key of C major, as described in the Top Voice query. This user interface, which was created for demonstration purposes, allows up to 4 events per sequence; however, the underlying SQL-generating mechanism does not have this limitation and allows the specification of an unlimited number of events per sequence. The SQL that was generated from the specification in Figure 5 is shown in Figure 6.

Had this user wanted to describe the sequence in a different key, such as G major, he could have selected G major as the anchor and described the sequence as [G B D], [DF♯A], [D F♯A C], [G B D]. The flexibility of describing chords in this method is evident. Figure 7 shows results for this query. Figure 8 shows an almost identical specification using chord and inversion names. It is not completely identical, as the inversion definition harbors additional constraints which were left out of the detailed description. For example, the second event constraint in the detailed description does not require that D act as bass, which is part of the definition of "second inversion". This example illustrates how the detailed description feature supports the definition of new chord types on-the-fly, as it relates directly to the internal representation of the database index. One could envision how this form of description could be entered, for example, via a midi keyboard.


Pattern Retrieval Demo - Detailed

First event	Second event	Third event	Fourth event
C top	B top	F top	E top
E	D	B	C
G	G	D	G
Select...	Select...	G	Select...

Maximum distance 0.5 4 0.5 seconds

Anchor key: C major / A minor

Click here to retrieve sequence locations

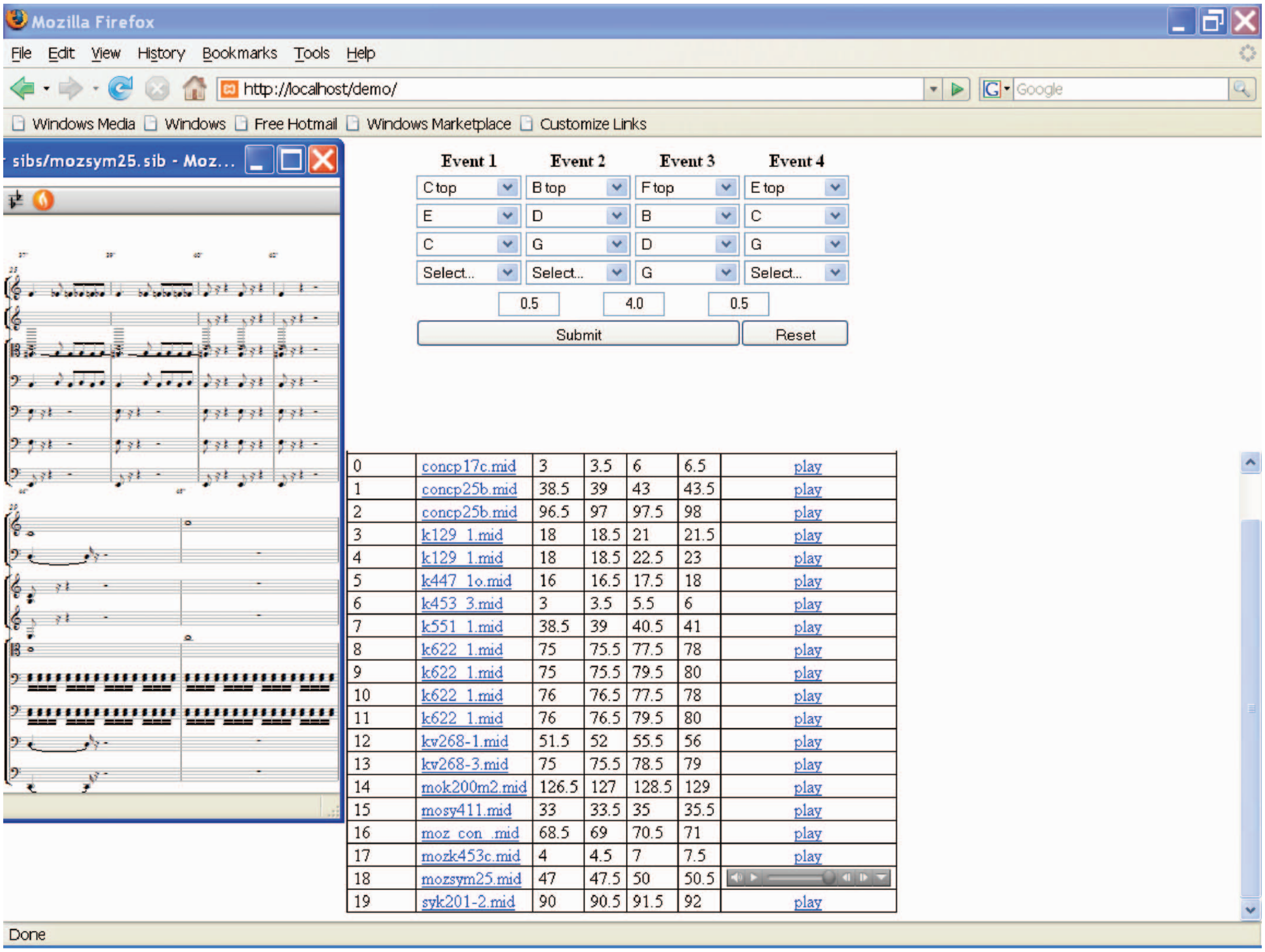
Fig. 5. Describing the 1-7 . . . 4-3: detailed method.

Due to the normalization of the database, sequences matching the query are retrieved regardless of the key in which they are written, but the modulation role is preserved. To illustrate this, let us return once again to the 1-7 . . . 4-3 example. The specification shown in Figures 5 and 8 would retrieve all instances of the sequence that are presented in the notated key (or, to use our terminology, "anchor key") of the piece in which they are written. However, to retrieve instances that are presented in the dominant key of their respective pieces, the sequence definition would need to be modified: for example, by changing the anchor to F major. If the user is not interested in finding the sequence in a specific modulation, he or she may choose "all keys" as the anchor, and the system would return all instances of the sequence, without preserving modulation role. To find a sequence that contains a modulation within it, the user would need to incorporate this modulation into the sequence definition. For example, the sequence [C E G], [F A C], [G B D], [C E G], [D F♯A] with C major as anchor would describe a I, IV, V sequence that presents in the notated key and then in the dominant key.

The choice of absolute time as the basic building block of the representation was motivated by several considerations, including: (a) time information is readily and reliably available in midi data. Beats and bars cannot be inferred with the same degree of accuracy. (b) A mechanism for musical sequence retrieval that uses only pitch and time information could potentially be applied to audio data. (c) For events that are close together in time, such as schema events, beat and bar units are often too long for distance specification. A sense of beat may not even be established in very short time spans. (d) For events that are related to each other by the proximity of their presentation, such as

```
SELECT T1.PIECE_KEY, MUSICAL_PIECES.PIECE_DESCRIPTION,
T1.START_TIME AS t1, T2.START_TIME AS t2, T3.START_TIME AS t3,
T4.START_TIME AS t4 FROM HH_WINDOWS AS T1 INNER JOIN
MUSICAL_PIECES ON T1.PIECE_KEY=MUSICAL_PIECES.PIECE_KEY,
HH_WINDOWS AS T2, HH_WINDOWS AS T3, HH_WINDOWS AS T4 WHERE
T1.C=3 AND T1.E>0 AND T1.G>0 AND T2.B=3 AND T2.D>0 AND T2.G>0 AND
T3.F=3 AND T3.B>0 AND T3.D>0 AND T3.G>0 AND T4.E=3 AND T4.C>0 AND
T4.G>0 AND T2.PIECE_KEY=T1.PIECE_KEY AND
T3.PIECE_KEY=T2.PIECE_KEY AND T4.PIECE_KEY=T3.PIECE_KEY AND
T2.START_TIME>T1.START_TIME AND T3.START_TIME>T2.START_TIME AND
T4.START_TIME>T3.START_TIME AND ((T2.START_TIME-
T1.START_TIME)<=0.5) AND ((T3.START_TIME-T2.START_TIME)<=4.0) AND
((T4.START_TIME-T3.START_TIME)<=0.5) ORDER BY T1.PIECE_KEY,
T1.START_TIME, T2.START_TIME, T3.START_TIME, T4.START_TIME;
```

Fig. 6. Generated SQL.


Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/demo/
Google
Windows Media Windows Free Hotmail Windows Marketplace Customize Links
sibs/mozsym25.sib - Moz...
Submit
Reset
Done

Event 1	Event 2	Event 3	Event 4
C top	B top	F top	E top
E	D	B	C
C	G	D	G
Select...	Select...	G	Select...

0.5	4.0	0.5

0	concp17c.mid	3	3.5	6	6.5	play
1	concp25b.mid	38.5	39	43	43.5	play
2	concp25b.mid	96.5	97	97.5	98	play
3	k129_1.mid	18	18.5	21	21.5	play
4	k129_1.mid	18	18.5	22.5	23	play
5	k447_1o.mid	16	16.5	17.5	18	play
6	k453_3.mid	3	3.5	5.5	6	play
7	k551_1.mid	38.5	39	40.5	41	play
8	k622_1.mid	75	75.5	77.5	78	play
9	k622_1.mid	75	75.5	79.5	80	play
10	k622_1.mid	76	76.5	77.5	78	play
11	k622_1.mid	76	76.5	79.5	80	play
12	kv268-1.mid	51.5	52	55.5	56	play
13	kv268-3.mid	75	75.5	78.5	79	play
14	mok200m2.mid	126.5	127	128.5	129	play
15	mosy411.mid	33	33.5	35	35.5	play
16	moz_con_.mid	68.5	69	70.5	71	play
17	mozk453c.mid	4	4.5	7	7.5	play
18	mozsym25.mid	47	47.5	50	50.5	
19	syk201-2.mid	90	90.5	91.5	92	play

Fig. 7. Displaying results for the 1-7 . . . 4-3.

schema events, distance in seconds is often more important than distance in beats: for example, in rapid tempo pieces, schema events could be further apart – in bars – than in slow tempo pieces. The unvarying fact in both cases is that the events would be just a few seconds apart.

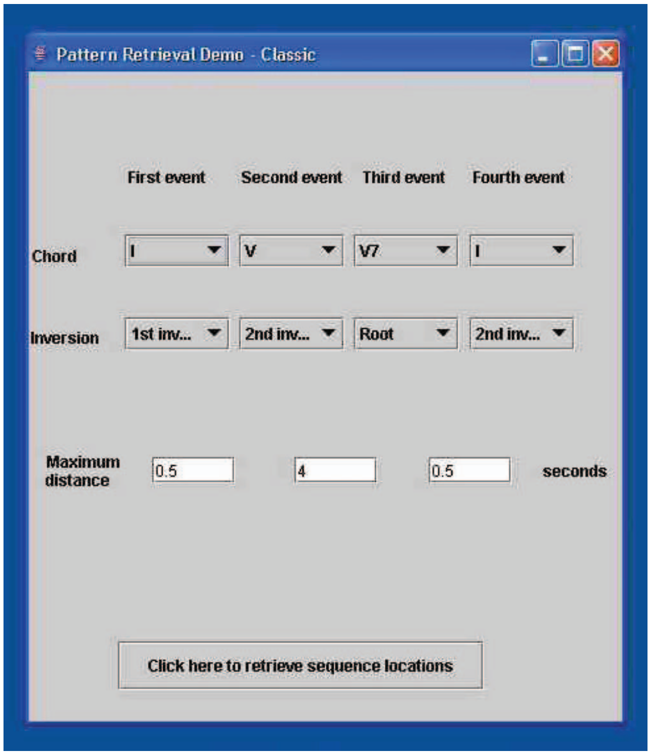
A concern that may arise is that analysing time information of performance data could make the system very sensitive to tempo differences between performances of a piece. This is a valid concern. Yet, if one assumes that the user specifies reasonable time constraints, then most reasonable renditions of the piece should surface. For example, if the specification is that two events should be no more than two seconds apart, then all renditions in which these events are two seconds *or less* apart will be retrieved. If the user suspects that some performers may play the piece even slower, the constraint should be set to, say, five seconds. Such a setting would likely worsen the precision of the query (i.e. some irrelevant instances could be retrieved) but would improve the recall to include even the slowest-tempo instances of the pattern.

An intended expansion of the system is to provide the user with a choice of alternative distance units, such as beats or bars.

9. Conclusions and future work

The system prototype described in this paper has demonstrated its ability to cope with the complex challenges inherent in the task of harmony retrieval. By creating a sophisticated index of the musical data and using dynamic SQL queries to access it, user-friendly and robust harmony search becomes possible.

Future plans include transferring these methods to other types of music data, such as audio files and Music XML. The methods used in the cover song studies to extract pitch class information from audio data (e.g. chroma features and pitch class profiles) are extremely useful and could likely be applied here. Other planned improvements include support for specification of constraints on melody, rhythm, contour and metric


Pattern Retrieval Demo - Classic
Click here to retrieve sequence locations

	First event	Second event	Third event	Fourth event	
Chord	I	V	V7	I	
Inversion	1st inv...	2nd inv...	Root	2nd inv...	
Maximum distance	0.5	4	0.5		seconds

Fig. 8. Describing the 1-7 . . . 4-3: classic method.

placement, the enabling of multiple roles per pitch class, ranking the retrieved results, fuller automation of the similarity retrieval process and the option of specifying distance in alternative units, such as beats and bars. Integration with additional methods of representation is planned, in an effort to capture utmost information on the many facets of music.

Acknowledgements

This work was funded in part by National Science Foundation (NSF) grant IIS-0328471. The author gratefully acknowledges Richard Ashley, Bart Berman, J. Stephen Downie, Robert Gjerdingen and M. Cameron Jones for their help and insightful contributions.

References

Agrawal, R. & Srikant, R. (1995). Mining sequential patterns. In *Proceedings of the 11th International Conference on Data Engineering*, Taipei, Taiwan.

Bello, J.P. (2007). Audio-based cover song retrieval using approximate chord sequences: testing shifts, gaps, swaps and beats. In *Proceedings of ISMIR 2007*, Vienna, Austria.

Bergeron, M. & Conklin, D. (2007). Representation and discovery of feature set patterns in music. In *Proceedings of the International Workshop on Artificial Intelligence and Music, 20th International Joint Conference on Artificial Intelligence (IJCAI)*, Hyderabad, India, pp. 1– 12.

Berman, T. (2006). A method for discovering musical patterns through time series analysis of symbolic data. Doctoral dissertation, Northwestern University School of Music, USA (available at: http://wwwlib.umi.com/dissertations/fullcit/3212773).

Berman, T., Downie, J.S. & Berman, B. (2006). Beyond error tolerance: finding thematic similarities in music digital libraries. In *Research and Advanced Technology for Digital Libraries: Proceedings of the 10th European Conference, ECDL 2006*, Alicante, Spain.

Birmingham, W.P., Dannenberg, R.B., Wakefield, G.H., Bartsch, M., Bykowski, D., Mazzoni, D., Meek, C., Mellody, M. & Rand, W. (2001). Musart: music retrieval via aural queries. In *Proceedings of ISMIR 2001*, Bloomington, Indiana.

Classical Music Archives. http://www.classicalarchives.com

Doraisamy, S. & Ruger, S. (2003). Robust polyphonic music retrieval with N-grams. *Journal of Intelligent Information Systems, 21*(1), 53–70.

Downie, J.S. & Nelson, M. (2000). Evaluation of a simple and effective music information retrieval method. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, Athens, Greece.

Ellis, D.P.W. & Poliner, G.E. (2007). Identifying "cover songs" with chroma features and dynamic programming beat tracking. In *Proceedings of IEEE International Conference on Acoustics, Specch and Signal Processing (ICASSP 07)*, Hawaii, USA.

Foote, J.T. (1997). Content-based retrieval of music and audio. In *Proceedings of the SPIE Multimedia Storage and Archiving Systems II (Bellingham, WA)*, *3229*, 138–147.

Gjerdingen, R.O. (1988). *A classic turn of phrase: music and the psychology of convention*. Philadelphia: University of Pennsylvania Press.

Honing, H. (1990). POCO: an environment for analyzing, modifying, and generating expression in music. In *Proceedings of the 1990 International Computer Music Conference* (pp. 364–368). San Francisco: Computer Music Association.

Huron, D. (1999). *Music research using Humdrum*. Available online at: http://dactyl.som.ohio-state.edu/Humdrum/

Liu, C., Hsu, J. & Chen, A. (1999). Efficient theme and nontrivial repeating pattern discovering in music databases. In *Proceedings of the 15th International Conference on Data Engineering (ICDE'99)*, Sydney, Australia.

Logan, B. & Salomon, A. (2001). A music similarity function based on signal analysis. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, Tokyo, Japan.

Mardirossian, A. & Chew, E. (2006). Music summarization via key distributions: analysis of similarity assessment across variations. In *Proceedings of ISMIR 2006*, Victoria, Canada.

Marolt, M. (2006). A mid-level melody-based representation of calculating audio similarity. In *Proceedings of ISMIR 2006*, Victoria, Canada.

Meek, C. & Birmingham, W.P. (2001). Thematic extractor. In *Proceedings of ISMIR 2001*, Bloomington, Indiana, USA.

Meyer, L.B. (1973). *Explaining music: essays and explorations*. Chicago: University of Chicago Press.

McNab, R.J., Smith, L.A., Witten, I.H., Henderson, C.L. & Cunningham, S.J. (1996). Towards the digital music library: tune retrieval from acoustic input. In *Proceedings of the First ACM International Conference on Digital Libraries*, Bethesda, Maryland, USA.

Narmour, E. (1977). *Beyond Schenkerism; the need for alternatives in music analysis*. Chicago: University of Chicago Press.

Pampalk, E. (2006). Computational models of music similarity and their application in music information retrieval. Doctoral dissertation, Vienna University of Technology, Austria.

Pickens, J. (2004). Harmonic modeling for polyphonic music retrieval. Doctoral dissertation, University of Massachusetts Amherst, USA.

Serrà, J. & Gómez, E. A cover song identification system based on sequences of tonal descriptors. In *Mirex 2007*.

Tzanetakis, G., Ermolinski, A. & Cook, P. (2003). Pitch histograms in audio and symbolic music information retrieval. *Journal of New Music Research, 32*(2), 143–152.

Wundt, W. (1874). *Principles of Physiological Psychology*. 1st English edition published in 1904. New York: Sonnenschein, London and Macmillan. Translation of the 5th German edition (1902) by Edward Bradford Titchener.