A Survey of Feature Selection Techniques for Music Information Retrieval

Jeremy Pickens
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01002   USA
jeremy@cs.umass.edu

ABSTRACT

The content-based retrieval of Western music has received increasing attention recently. Much of this research deals with monophonic music. Polyphonic music is more common, but also more difficult to represent. Music information retrieval systems must extract viable features before they can define similarity measures. We summarize and categorize representation features that have been used for polyphonic retrieval with the aim of laying standardized groundwork for future feature extraction research. Comparisons with and extensions to monophonic approaches are given, and a new feature, an extension of duration-independent pitch slices, is proposed.

Key words: Music retrieval, polyphonic features

1. INTRODUCTION

Many research projects in music information retrieval are concerned with building retrieval systems, defining similarity measures, and otherwise finding occurrences and variations of a musical fragment within a collection of music documents. These systems necessarily start with descriptions of the features used for matching, but often make the matching algorithms rather than features the primary object of their research.

In this paper we turn our attention to the many features that have been used for content-based, ad hoc music information retrieval. The features one extracts naturally influence which types of systems can or cannot be built, but this issue belongs to a later stage of research. At the moment, music IR is in its infancy. There exists few standard techniques for feature extraction, and existing techniques have not been categorized. This paper attempts to fill that gap in the research.

This paper distinguishes itself from the related field of *audio* music retrieval in that the lowest-level representation with which we are concerned is the event: the pitch, onset, and duration of every note in a music source is known. In *monophonic* music, no new note begins until the current note has finished sounding. Sources are restricted to one-dimensional note sequences. *Homophonic music* adds another dimension; notes with different pitches may be played simultaneously, but they must still start and finish at the same time. *Polyphonic* music adds yet another complication. A note may begin before a previous note finishes.

"Feature extraction can be thought of as representation conversion, taking low-level representation and identifying higher level features" [6]. Features at one level may build upon features at a lower level. The techniques employed range from string-matching algorithms familiar to a computer scientist to "deep structure" approaches more familiar to a music theorist. The goal, however, is not an understanding of music. It is not to develop a better theory of music, or even to analyze music. The goal is retrieval. Computational and music-theoretic analyses of music might aid that goal, but we consider them important only insofar as they aid feature extraction, and thus the retrieval effort.

Section 2 contains other surveys relevant to music representation. Sections 3 and 4 present features used for monophonic and polyphonic music, respectively. Section 5 discusses other music-theoretic techniques which might be useful for future feature extraction methodology. Section 6 presents an extension of the polyphonic duration-independent pitch slice feature.

2. RELATED WORK

A few summaries of interest to polyphonic feature extraction are available. 12] lists various representational issues associated with music. 7] provide a comprehensive summary of the problems of polyphonic music information retrieval. Many of the issues raised in their paper are directly related to the challenges encountered in polyphonic feature extraction, and thus provide good background and context for this paper. 17] discusses audio music information retrieval, rather than the symbol-based approach which is the focus of this paper.

3. MONOPHONIC FEATURE SELECTION

Most of the current work in music IR has been done with *monophonic* sources, often called "melodies". In monophonic music, no new note begins until the current note has finished sounding. While the focus of this paper is *polyphonic* (section 4) rather than monophonic feature extraction, the latter is a proper subset of the former. Not only do certain polyphonic feature extraction techniques require the exact same techniques prescribed for monophonic music (section 4.1), but other monophonic techniques may be modified and extended into the polyphonic realm. Therefore, it is necessary

to review those features which have been used for monophonic music.

3.1 Relative vs. Absolute Measures

The most basic approach to monophonic feature extraction reduces notes to a single dimension. Pitch is extracted and duration is ignored, or vice versa. Both pitch and duration information may be used in the final retrieval system, but as features they are treated separately. Arguments may be made for the importance of absolute pitch or duration. However, most music IR researchers favor relative measures because a change in tempo or transposition across keys does not significantly alter the music information expressed [15, 30, 18, 26, 4, 22, 37, 24].

Relative pitch has three standard expressions: exact interval, rough contour, and simple contour. *Exact interval* is the signed magnitude between two contiguous pitches. *Simple contour* keeps the sign and discards the magnitude. *Rough contour* keeps the sign and groups the magnitude into a number of equivalence classes. For example, the intervals 1-3, 4-7, and 8-12 become the classes "a little", "a fair amount", and "a lot". Relative duration has three similar standards: *exact ratio*, *rough contour*, and *simple contour*. The primary difference between pitch and duration is that duration invariance is obtained through proportion, rather than interval. Contours assume values of "faster" or "slower" rather than "higher" or "lower". In all above-mentioned relative features, intervals of "0" and ratios of "1" indicate no change from previous to current note.

In information retrieval terms, exact intervals and ratios yield high precision, while contour aids recall. Rough contours or equivalence classes attempt to balance the two, gaining some flexibility without sacrificing too much precision.

There are exceptions to pitch and duration as independent features [24, 10]. Relative measures of pitch and duration are combined into single objects. Some flexibility is retained through the relative measures, but some flexibility is lost through the combination of features. It becomes more difficult to search on pitch or duration only, to find matches with similar pitch sequences but different duration sequences.

3.2 Unigrams vs. N-grams

Collectively, the techniques in section 3.1 are known as *unigrams*. For certain types of retrieval systems, such as those that use string matching to compare melodic similarity, or those that build ordered sequences of intervals (*phrases*) at retrieval time [32], such features are rich enough. However, other retrieval approaches require "larger" basic features. Longer sequences, or *n-grams*, are constructed from an initial sequence of interval or ratio unigrams.

One of the simpler approaches to n-gram extraction is the use of sliding windows [16, 6]. The sequence of notes within a length n window are converted to a sequence of relative unigrams. Numerous authors suggest a tradeoff between unigram type and n-gram size. Where more precise (exact magnitude) unigrams are used, n-grams remain shorter, perhaps not to sacrifice recall. Where more flexible (contour) unigrams are used, n-grams remain longer, perhaps not to sacrifice precision.

A more sophisticated approach to n-gram extraction is the detection of repeating patterns [19, 40, 27, 1]. Implicit in these approaches is the assumption that frequency or repetition plays a large role in music similarity. Another alternative segments a melody into musically relevant passages, or "musical surfaces" [29]. Weights are assigned to every potential boundary location, expressed in terms of relationships among pitch intervals, duration ratios, and explicitly delimited rests (where given). The weights are then evaluated, and automatic decisions are made about where to place boundary markers using local maxima. The sequence of notes between markers becomes the n-gram window.

Finally, some approaches use string matching techniques to detect and extract n-grams [2, 21] We wish to distinguish between string matching retrieval and extraction algorithms. Although the methods are similar, the difference lies in the object to which each is applied. String matching retrieval algorithms treat queries and collections as unigram feature strings, and search for instances of the former within the latter. String matching n-gram extraction techniques use notions such as insertions, deletions, and substitutions to automatically pull n-grams from a source; no query string is required. These n-grams, unlike those from other techniques, are composed of unigrams which are *not always contiguous* within the original source.

3.3 Shallow Structure

Shallow structural features are what we call feature extraction techniques which range from simple statistical measures to lightweight computational or music-theoretic analyses. An example of such a feature for *text* information retrieval is a part-of-speech tagger [43], which identified words as nouns, verbs, adjectives, and so on. While music does not have parts of speech, there are roughly analogous shallow structural concepts such as key. A technique which examines a set or sequence of note pitches and does a probabilistic best fit into a known key is a shallow structural feature extractor [38, 23]. A sequence of pitches is thus restructured as a sequence of keys or tone centers.

Similar shallow structural techniques may be defined for duration as well as pitch. 39] describes techniques for defining the temporal pattern complexity of a sequence of durations. These methods may be applied to an entire source, or to subsequences within a source. A sequence of durations could be restructured as a sequence of rhythm complexity values.

Statistical features may also be used to aid the monophonic music retrieval process. We draw the distinction between a pitch interval as a feature, and the statistical measure of pitch intervals. Extraction of the latter depends on the identification of the former, while retrieval systems which use the former do not necessarily use the latter. 35] creates an "interval repertoire", which includes the *relative frequencies* of various pitch unigrams, length of the source, and "tendency" of the melody (i.e.: 3% descending or 6% ascending). Mentioned, but not described, is a "duration repertoire" similar to the interval repertoire, giving counts and relative frequencies of duration ratios and contours. There are other researchers which do statistical analyses of their sequential features as well [16]. It is clearly possible to subject most if not all of the features described in section 3.2 to statistical analysis.

4. POLYPHONIC FEATURE SELECTION

In section 3 we introduced *monophonic music*, and characterized it as a sequence of notes. *Homophonic music* adds another dimension; notes with a different pitch may be played simultaneously, they must start and finish at the same time. *Polyphonic* music adds yet another dimension. A note may begin before a previous note finishes.

Polyphony poses serious challenges to many monophonic features. It is difficult to speak of the "next" note in a sequence when there is no clear one-dimensional sequence. Explicit features such as pitch interval and duration contour are no longer viable, but the implicit assumption which led to those features are. This simplifying assumption is independence between dimensions.

For monophonic music, most researchers assume independence between the pitch and duration of a note. These features are not truly independent, but the simplification makes retrieval much easier. For polyphonic music, researchers have assumed independence between overlapping notes. The remainder of this paper is an exploration and categorization of the various methods in which overlapping notes are segmented and features are extracted.

4.1 Monophonic Reduction

One of the oldest approaches to polyphonic feature selection is what I call *monophonic reduction*. A polyphonic source is reduced to a monophonic source by selecting at most one note at every time step. This monophonic sequence of notes can then be further deconstructed using the monophonic feature selection techniques from section 3.

The monophonic sequence that most researchers attempt to extract is the melody, or theme. Whether this monophonic sequence is useful for retrieval is tied to how well a technique extracts the "correct" melody, in addition to how well any monophonic sequence actually represents a polyphonic source.

4.1.1 Short Sequences

3] does monophonic reduction, constructing short monophonic sequences of note pitches from polyphonic sources. However, the selection is done manually. Clearly, this becomes impractical as music collections grow large.

Automated methods become necessary. There exist retrieval algorithms which can search polyphonic sources for exact or evolutionary *known apriori* monophonic strings [13, 25, 20]. There exist feature extraction algorithms which automatically select salient monophonic patterns, not known apriori, from monophonic sources using clues such as repetition and evolution (section 3.2). Yet we know of no work which combines the two, automatically selecting short, salient strings from polyphonic sources. This would be a useful feature selection technique, and appears to be a solvable research problem.

4.1.2 Long Sequences

One might not trust the intuition that repetition and evolution yield salient, short monophonic sequences. An alternative is to pull out an entire monophonic note sequence equal to the length of the polyphonic source. A naive approach is described in which the note with the highest pitch at any given time step is extracted [41, 42, 33]. An equally naive approach suggests using the note with the lowest pitch [4]. Other 41 approaches use voice information (when available), average pitch, and entropy measures to wind their way through a source. Interestingly, the highest pitch approach yields fairly decent results.

Other techniques do not presume to extract a melodic line, but split a polyphonic source into a number of monophonic sequences [28, 9]. Each monophonic sequence can be searched independently, and the results combined to give a score for the piece as a whole. How

well this works depends not only on the technique, but on the music being split; some music might lend itself to easier decomposition.

4.2 Homophonic Reduction

A second popular technique for segmenting the overlapping notes common to polyphonic music is what I call *homophonic reduction*. Instead of selecting at most one note at a given time step, as was done in section 4.1, one selects every note at a given time step. Thus, a polyphonic source is reduced to a homophonic source by assuming independence between notes with overlapping duration.

Many names have been given to the homophonic objects created under this assumption: syncs, chords, windows, sets, slices, or chunks. I prefer the term *homophonic slice*. There are also slight variations amongst various methologies. Some homophonic reductions only consider notes with simultaneous attack time, i.e.: if note X is still playing at the time that note Y begins, only Y belongs to the homophonic slice [14]. Other approaches use all notes currently sounding, i.e.: if note X is still playing at the time that note Y begins, both X and Y belong to the slice [25]. Yet other approaches use larger, time or rhythm based windows in which all the notes within that rhythmic window belong to the slice [33, 11]. In all cases, the resulting homophonic source can be characterized as a sequence of pitch sets.

25] propose a number of modifications to homophonic slices. In one variation, duration information is discarded completely and the slice becomes a set of MIDI pitch values (0 to 127). Another variation uses octave equivalence to reduce the size of the pitch set to 12. A third variation creates transposition invariance by transforming the sequence of homophonic slices $(S = S_1 S_2 \ldots S_n)$ into a sequence of pitch interval sets $(D = D_1 D_2 \ldots D_{n-1})$:

```
1      for i := 2 to n do
2            for each a ∈ S_{i-1} and b ∈ S_i do
3                  D_{i-1} := D_{i-1} ⋃ {b − a}
```

Before the reduction of polyphonic to homophonic music, it was not possible to extract pitch intervals, because overlapping durations make it unclear which note is the "next" note in a sequence. Homophonic slices assume independence between duration overlaps and thereby recreate sequentiality and the ability to construct intervals.

4.3 Monorhythmic and Homorhythmic Reduction

There is a symmetry inherent in a monophonic sequence of notes. A note splits cleanly into its pitch and duration (time) component. It is as easy to extract a sequence of pitch intervals as it is a sequence of duration ratios. Polyphonic music is different. Almost by nature, the pitch of an individual note and the duration of that note do not carry equal weight. For example, when trying to extract the monophonic "melody" from a polyphonic source (section 4.1), it is reasonable to use the note with the highest pitch at a given time step. Using the note with the longest duration at a given time step makes much less sense. Similarly, when reducing a polyphonic source to a homophonic source, it is most often the pitch which comprises the homophonic slices. The set of durations co-existing at one time step does not seem as useful.

The problem might lie in the fact that monophonic and homophonic reductions are by definition pitch-centric, because notes or sets of

notes are extracted *at various time steps*. If duration is to be given fair treatment, notes or note sets should be extracted *at various pitch steps*. For example, one could examine a single pitch, such as Middle C (MIDI note 60) and reduce that pitch value to a set of onset and ending times. Thus, the set would reflect the attack time and duration of every occurrence of the middle C pitch in a polyphonic source.

Monophonic reduction is so named because it extracts one (variable) pitch at each (fixed) time step. This new duration-centric measure could be named *monorhythmic reduction*, for it extracts *each (variable) time step at one (fixed) pitch*. Doing the same rhythmic extraction for all 128 MIDI pitches (treating each pitch as an independent channel) would then be called *homorhythmic reduction*.

Though such a view of duration counters our intuition of musically salient features, it still might be useful for retrieval. At the very least, it gives different transformations of a polyphonic source, which might reveal otherwise unnoticed patterns than those shown by more traditional, pitch-based feature extraction techniques. No retrieval systems known to the authors have specifically addressed polyphonic duration features.

4.4 Shallow Structure

As with monophonic music, a shallow structural feature is the name we give feature extraction techniques which range from simple statistical measures to lightweight computational or music-theoretic analyses.

33] begins with octave equivalent homophonic slices and further tempers them by their harmonicity. The pitches are fit to a normalized array of harmonic classes. These harmonic classes are comprised of triads (major, minor, augmented, and diminished) and seventh chords (major, minor, dominant, and diminished minor) for every scale tone. A sequence of pitch sets thus becomes a sequence of harmonic chord sets. The pitches in a slice often fit more than one triad or seventh chord, so neighboring slices are use to disambiguate potential harmonic candidates.

Tempering a homophonic slice by its inherent harmonicity should sound familiar. In section 3.3, windows slid across monophonic sources to establish key signatures or tonal contexts. Homophonic slices are windows as well, albeit not necessarily as wide. Since a greater variety of pitches often coexist in a narrower time frame, polyphonic music might be an even better domain in which to apply this shallow structural technique; more context is provided by more notes.

5] proposes a number of other statistical and shallow structural features appropriate for polyphonic or reduced-homophonic music: the number of notes per second, the number of chords per second, the pitch of notes (lowest, highest, mean average), the number of pitch classes used, pitch class entropy, the duration of notes (lowest, highest, mean average), number of semitones between notes (lowest, highest, mean average), and how repetitive a source is. There are certainly many more shallow structural features possible for polyphonic music. Existing work is just beginning to enumerate the possibilities.

5. DEEP STRUCTURE

A *deep* structural feature is the name we give more complicated music-theoretic, artificial intelligent, or other form of symbolic cognitive techniques for feature extraction. Such research constructs its features with the goal of conceptual "understanding" or explanation of music phenomena. For information retrieval, we are not interested in explanation so much as we are in comparison or similarity measures. Any technique which produces features that aid the retrieval process is useful. Unfortunately, most deep structural techniques are not fully automated, especially for polyphonic music. These deeper theories therefore must inspire rather than solve our feature extraction problems. Some examples include Schenkerian analysis [36], AI techniques [8], Chomskian grammars [34], and other structural representations [31], to name very few.

6. EXTENSIONS

Given the large number of features discussed for both monophonic and polyphonic music, we now propose an extension to one of these features. Attributes which appeal to us are the homophonic slice and the interval. The former creates durational independence and the latter, transposition invariance. A homophonic interval is the feature which encapsulates both notions, and was proposed by 25 in section 4.2. Our extension to this feature makes use of his notation and pseudocode.

The inspiration for this extension comes from observations made by 26] and 4]. Intervals formed from contiguous notes do not always reveal the true "contour" of a piece. Ornamentation, passing tones, and other extended variations tend to obscure rather than reveal musically salient passages. Rather than abandon intervals and return to absolute pitch atomic units, we create "secondary" intervals, a secondary contour [4]. In other words, we extract intervals between non-contiguous notes.

Our goal is to extend this notion of non-contiguous intervals to homophonic intervals. Recall from section 4.2 that homophonic intervals are constructed by taking the difference between all possible note combinations in two contiguous homophonic slices. We once again transform the sequence of slices ($S = S_1 S_2 \ldots S_n$) into a sequence of pitch slice intervals ($D = D_1 D_2 \ldots D_{n-1}$). However, each interval set D_i will no longer exclusively contain contiguous intervals. Non-contiguous intervals will be allowed between the notes at the current slice and the notes up to k slices ahead in the sequence:

```
1      for i := 1 to n − 1 do
2          for j := (i + 1) to (i + 1 + k) do
3              for each a ∈ S_i and b ∈ S_j do
4                  D_i := D_i ⋃ {b − a}
```

Variation A It is possible to allow duplicates within interval slices; the frequency of occurrence of each interval is given in the set. This variation could be useful for separating "strong" from "weak" intervals. For example, if one slice holds a C-Major triad, and a neighboring slice rises to a G-Major triad, the set of intervals will be $\{+14, +11, +10, +7, +7, +7, +4, +3, +0\}$. The +7 interval has the highest frequency, and therefore might be the strongest, most salient, and useful for retrieval purposes.

Variation B This variation works best as an extension of variation B. It deals with the relative importance of the intervals found at increasingly distant time slices. Though non-contiguous intervals are useful for dealing with ornamentation and other variations, they potentially add noise. The current algorithm adds all non-contiguous intervals to the current slice, regardless of the distance, k. So a natural variation is to downweight the occurrence of an interval based

on its distance from the current slice, using a simple distance formula or a one-tailed probability distribution.

The downweighting does not have to be monotonically decreasing; it could be periodic. If the rhythm or beat of the source is known, then the weighting function could experience a small resurgence every time a distant slice is located on the same rhythmically significant beat at the current slice. Slices which are not located "on-beat" are downweighted more. Furthermore, the entire periodicity could decay as a function of distance from the current slice; on-beat intervals closer to the current slice are weighted higher than on-beat intervals further away.

Variation C The converse of non-contiguity is concurrency. Just as it is possible to extract intervals from the current slice to another slice k time steps in the future, we may extract intervals from *within* the current slice. This could be useful for establishing harmonic context. For example, if intervals of +3, +4 and +7 are found, one could conclude that a major triad exists within the current slice. The changes to the algorithm are slight, and mostly insure that an interval between a note and itself is not taken:

```
1     for i := 1 to n − 1 do
2         for j := i to (i + k) do
3             for each a ∈ S_i and b ∈ S_j do
4                 if (i = j) and (b ≠ a) do
5                     D_i := D_i ⋃ {b − a}
```

7. SUMMARY

Few of the feature extraction approaches in this paper should be new for those familiar with work in music IR. Indeed, this paper attempts to summarize and categorize the various techniques that have been used for monophonic and polyphonic music retrieval. The categories proposed are rough and do not claim to be the final standard for future thought on music features. However, they provide a useful foundation for discussion of current work. With this foundation, we hope that readers will see additional holes or gaps, areas where new features may be proposed or current techniques may be extended. Section 6 was an example of one such extension.

A number of authors have suggested doing music retrieval in two phases. The goal of the first phase is to retrieve a large set of general matches, low in precision but high in recall. In the second phase, this set is tamed and false matches eliminated. The retrieval task is split because features used to obtain high recall are necessarily the same features appropriate for high precision. High precision retrieval tasks also have the potential to be computationally expensive, which furthers the need for intermediate phases which tame the entire collection. It is therefore of great use for the music information retrieval community to remain familiar with as many different feature extraction techniques as possible.

8. REFERENCES

[1] D. Bainbridge. The role of music ir in the new zealand digital library project. In *Proceedings of the first International Symposium on Music Information Retrieval (ISMIR 2000)*, http://ciir.cs.umass.edu/music2000/papers.html, October 2000.

[2] V. Bakhmutova, V. D. Gusev, and T. N. Titkova. The search for adaptations in song melodies. *Computer Music Journal*, 21(1):58–67, 1997.

[3] H. Barlow and S. Morgenstern. *A Dictionary of Musical Themes*. Crown Publishers, 1948.

[4] S. Blackburn and D. DeRoure. A tool for content-based navigation of music. In *Proceedings of ACM International Multimedia Conference (ACMMM)*, 1998.

[5] S. Blackburn and D. D. Roure. Music part classification in content based systems. In *6th Open Hypermedia Systems Workshop*, San Antonio, TX, 2000.

[6] S. G. Blackburn. Content based retrieval and navigation of music, 1999. Mini-thesis, University of Southampton.

[7] D. Byrd and T. Crawford. Problems of music information retrieval in the real world. To appear in *Information Processing and Management*, 2001.

[8] L. Camilleri. Computational theories of music. In A. Marsden and A. Pople, editors, *Computer Representations and Models in Music*, pages 171–185. Academic Press Ltd., 1992.

[9] H. Charnasse and B. Stepien. Automatic transcription of german lute tablatures: an artificial intelligence application. In A. Marsden and A. Pople, editors, *Computer Representations and Models in Music*, pages 143–170. Academic Press Ltd., 1992.

[10] A. L. P. Chen, M. Chang, J. Chen, J. L. Hsu, C. H. Hsu, and S. Y. S. Hua. Query by music segments: An efficient approach for song retrieval. In *Proceedings of IEEE International Conference on Multimedia and Expo*, 2000.

[11] M. Clausen, R. Engelbrecht, D. Meyer, and J. Schmitz. Proms: A web-based tool for searching in polyphonic music. In *Proceedings of the 1st International Symposium on Music Information Retrieval (ISMIR 2000)*, http://ciir.cs.umass.edu/music2000/papers.html, October 2000.

[12] R. B. Dannenberg. A brief survey of music representation issues, techniques, and systems. *Computer Music Journal*, 17(3):20–30, 1993.

[13] M. Dovey. An algorithm for locating polyphonic phrases within a polyphonic piece. In *Proceedings of AISB Symposium on Musical Creativity*, pages 48–53, Edinburgh, April 1999.

[14] M. Dovey and T. Crawford. Heuristic models of relevance ranking in searching polyphonic music. In *Proceedings of Diderot Forum on Mathematics and Music*, pages 111–123, Vienna, Austria, 1999.

[15] W. Dowling. Scale and contour: Two components of a theory of memory for melodies. *Computers and the Humanities*, 16:107–117, 1978.

[16] J. S. Downie. *Evaluating a Simple Approach to Music Information Retrieval: Conceiving Melodic N-grams as Text*. PhD thesis, University of Western Ontario, Faculty of Information and Media Studies, July 1999.

[17] J. Foote. An overview of audio information retrieval. *ACM Multimedia Systems*, 7(1):2–11, 1999. ACM Press/Springer Verlag.

[18] A. Ghias, J. Logan, D. Chamberlin, and B. Smith. Query by humming - musical information retrieval in an audio database. In *Proceedings of ACM International Multimedia Conference (ACMMM)*, pages 231–236, San Francisco, CA, 1995.

[19] J. L. Hsu, C. C. Liu, and A. L. P. Chen. Efficient repeating pattern finding in music databases. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM)*, 1998.

[20] C. Iliopoulos, M. Kumar, L. Mouchard, and S. Venkatesh. Motif evolution in polyphonic musical sequences. In L. Brankovic and J. Ryan, editors, *Proceedings of the 11th Australasian Workshop on Combinatorial Algorithms (AWOCA)*, pages 53–66, University of Newcastle, NSW, Australia, August 2000.

[21] C. Iliopoulos, T. Lecroq, L. Mouchard, and Y. J. Pinzon. Computing approximate repetitions in musical sequences. In *Proceedings of Prague Stringology Club Workshop PSCW'00*, 2000.

[22] A. Kornstädt. Themefinder: A web-based melodic search tool. *Computing in Musicology*, 11:231–236, 1998.

[23] M. Leman. Tone context by pattern integration over time. In D. Baggi, editor, *Readings in Computer-Generated Music*. Los Alamitos: IEEE Computer Society Press, 1992.

[24] K. Lemström, P. Laine, and S. Perttu. Using relative interval slope in music information retrieval. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 317–320, Beijing, China, October 1999.

[25] K. Lemström and J. Tarhio. Searching monophonic patterns within polyphonic sources. In *Proceedings of the RIAO Conference*, volume 2, pages 1261–1278, College of France, Paris, April 2000.

[26] A. T. Lindsay. Using contour as a mid-level representation of melody. Master's thesis, MIT Media Lab, 1996.

[27] C. C. Liu, J. L. Hsu, and A. L. P. Chen. Efficient theme and non-trivial repeating pattern discovering in music databases. In *Proceedings of IEEE International Conference on Research Issues in Data Engineering (RIDE)*, 1999.

[28] A. Marsden. Modelling the perception of musical voices. In A. Marsden and A. Pople, editors, *Computer Representations and Models in Music*, pages 239–263. Academic Press Ltd., 1992.

[29] M. Melucci and N. Orio. Musical information retrieval using melodic surface. In *Proceedings of ACM Digital Libraries*, Berkeley, CA, 1999.

[30] M. Mongeau and D. Sankoff. Comparison of musical sequences. *Computers and the Humanities*, 24:161–175, 1990.

[31] E. Narmour. *The Analysis and Cognition of Basic Melodic Structures*. The University of Chicago Press, Chicago, 1990.

[32] J. Pickens. A comparison of language modeling and probabilistic text information retrieval approaches to monophonic music retrieval. In *Proceedings of the 1st International Symposium for Music Information Retrieval (ISMIR)*, October 2000. See http://ciir.cs.umass.edu/music2000.

[33] R. E. Prather. Harmonic analysis from the computer representation of a musical score. *Communications of the ACM*, 39(12):119, 1996. See: Virtual Extension Edition of CACM.

[34] C. Roads. Grammars as representations for music. In C. Roads and J. Strawn, editors, *Foundations of Computer Music*, pages 403–442. Cambridge, MA: MIT Press, 1985.

[35] H. Schaffrath. Representation of music scores for analysis. In A. Marsden and A. Pople, editors, *Computer Representations and Models in Music*, pages 95–109. Academic Press Ltd., 1992.

[36] H. Schenker. *Free Composition*. Longman, New York, 1979.

[37] E. Selfridge-Field. Conceptual and representational issues in melodic comparison. *Computing in Musicology*, 11:3–64, 1998.

[38] I. Shmulevich and E. J. Coyle. The use of recursive median filters for establishing the tonal context in music. In *Proceedings of the IEEE Workshop on Nonlinear Signal and Image Processing*, Mackinac Island, MI, 1997.

[39] I. Shmulevich, O. Yli-Harja, E. Coyle, D.-J. Povel, and K. Lemström. Perceptual issues in music pattern recognition - complexity of rhythm and key finding. In *Proceedings of AISB Symposium on Musical Creativity*, pages 64–69, Edinburgh, United Kingdom, April 1999.

[40] Y.-H. Tseng. Content-based retrieval for music collections. In *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR)*, Berkeley, CA, 1999.

[41] A. Uitdenbogerd and J. Zobel. Manipulation of music for melody matching. In *Proceedings of ACM International Multimedia Conference (ACMMM)*. ACM, ACM Press, 1998.

[42] A. Uitdenbogerd and J. Zobel. Melodic matching techniques for large music databases. In *Proceedings of ACM International Multimedia Conference (ACMMM)*, Orlando Florida, USA, Oct. 1999. ACM, ACM Press.

[43] J. Xu, J. Broglio, and W. Croft. "the design and implementation of a part of speech tagger for english". In *University of Massachusetts Technical Report IR-52*, 1994. http://ciir.cs.umass.edu/publications/index.html.