# Feature Selection for Polyphonic Music Retrieval

Jeremy Pickens
Department of Computer Science
University of Massachusetts
Amherst, MA 01002 USA
jeremy@cs.umass.edu

## 1. INTRODUCTION

The content-based retrieval of Western music has received increasing attention recently. Most of this research deals with monophonic music. Polyphonic music is more common, but much more difficult to represent [3]. Music information retrieval systems must extract viable features before they can define similarity measures. We summarize and categorize features that have been used for polyphonic retrieval with the aim of laying standardized groundwork for future research on feature extraction. Comparisons with and extensions to monophonic approaches are given and a new feature is proposed.[1]

We do not consider music in audio form. The lowest-level representation with which we are concerned is the event: the pitch, onset, and duration of every note in a 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.

## 2. EXISTING FEATURES

**Monophonic Music** In the words of Blackburn [2], “feature extraction can be thought of as representation conversion, taking low-level representation and identifying higher level features.” For monophonic music, this procedure is fairly straightforward. First, independence between pitch and duration is assumed. Then, a number of decisions are made, between absolute and relative encoding, whether to treat notes as a set or as a sequence, and whether or not to apply structural, music theoretic analyses (a deeper, richer level of feature identification) to features extracted at previous stages.

Techniques based on relative measures naturally assume a sequence of notes, since a sequence of at least two notes is needed for relative encoding. The interval between two contiguous pitches or the ratio between two contigous notes is used to standardize sequences. A change in tempo or transposition across keys does not significantly alter the music information expressed. Intervals and ratios may be exact magnitude or they may be contour-based. Short sequences are built into larger n-gram features using sliding windows, repeating pattern detection, evolutionary pattern detection, or automatic segmentation of a entire source into musically salient phrases. [5, 6, 7, 10]. Those techniques which detect repeating and evolutionary patterns have found note sequences which are both contiguous and non-contiguous within the original source.

Other monophonic techniques extract richer structural or music theoretic features. An example of such a feature for *text* information retrieval is a part-of-speech tagger, which identifies words as nouns, verbs, adjectives, and so on. Similarly, there exist techniques which examine a set or sequence of note pitches and do a probabilistic best fit to diatonic pitch set (equivalent to key and mode), or which define rhythm complexity values over duration segments.

**Polyphonic Music** Polyphony poses serious challenges to many monophonic feature extraction techniques. It is difficult to speak of the “next” note in a sequence when there is no clear one-dimensional sequence. Features such as pitch interval, duration contour, even rhythm complexity are no longer immediately available, because there is not always one exclusive, salient pitch or duration at any given time step.

For monophonic music, most researchers assume independence between the pitch and duration of a note. For polyphonic music, researchers additionally assume independence between overlapping notes. In both cases, these features are not truly independent, but the simplifying assumption makes retrieval much easier.

There are numerous methods by which overlapping notes are segmented, multiple dimensions are reduced to a single dimension, features are extracted. One of the oldest approaches to polyphonic feature selection is what we call *monophonic slicing.* A monophonic slice is a sequence constructed from a polyphonic source by selecting at most one note at every time step. This monophonic sequence is then further broken down using aforementioned monophonic methods. Existing techniques extract sequences equal to the source length [11], but current research suggests that automatic extraction of shorter, musically significant n-gram sequences directly from a polyphonic source, using clues such as repetition and evolution, will soon become possible.

Another approach to polyphonic feature selection we call *homophonic slicing.* An entire set of notes is removed at every time slice, recasting the polyphonic source as a sequence of non-overlapping note sets. The manner in which homophonic slices are created differs. Various approaches consider only notes with simultaneous attack time [4], all notes which are currently sounding [8], or all notes within a larger, time or rhythm based window.

[1] This material is based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623, and NSF grant number IIS-9905842. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor(s).

*SIGIR'01,* September 9-12, New Orleans, Louisiana, USA..
Copyright 2001 ACM 1-58113-331-6/01/0009 ...$5.00.

Homophonic slices are further tempered by structural methods such as octave equivalence, harmonicity (e.g.: fitting the pitch set to major and minor triads and 7th chords), and best-fit key signature. Once independence between overlapping durations is established, one may even transform a sequence of pitch slices ($S = S_1\ S_2 \ldots S_n$) into a sequence of pitch slice intervals ($D = D_1\ D_2 \ldots D_{n-1}$) thereby recapturing transposition invariance [8]:

```
for $i := 1$ to $n - 1$ do
    for each $a \in S_i$ and $b \in S_{i+1}$ do
        $D_i := D_i \bigcup \{b - a\}$
```

Other statistical methods have been applied to homophonic slices, including but not limited to highest, lowest, average and total note counts, chord counts, pitch class counts and pitch class entropy.

## 3. EXTENSIONS

We propose an extension to the homophonic slice feature. The motivation for this extension comes from observations made by [9] and [1]. 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 musically salient passages. Rather than abandon intervals and return to absolute pitch atomic units, we create "secondary" intervals or contour. In other words, we extract pitch intervals between notes of non-contiguous homophonic slices. To our knowledge this is a new feature for polyphonic music retrieval, one which accounts for durational independence (slicing), transposition invariance (intervals), and secondary contour (non-contiguity).

We once again transform the sequence of homophonic 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:

```
for $i := 1$ to $n - 1$ do
    for $j := (i + 1)$ to $(i + 1 + k)$ do
        for each $a \in S_i$ and $b \in S_j$ do
            $D_i := D_i \bigcup \{b - a\}$
```

**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 {+0, +4, +7, +3, +7, +10, +7, +11, +14}. The +7 interval has the highest frequency, and therefore might be the strongest, most salient, and useful for retrieval purposes.

**Variation B** The previous variation may be further extended by weighting intervals found at increasingly distant slices. Though non-contiguous intervals are useful for dealing with ornamentation and other variations, they potentially add noise. The naive algorithm equally considers all non-contiguous intervals within the distance, $k$. Variation B downweights the interval as a function of 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 also be periodic, varying with the rhythm or beat of the polyphonic source.

## 4. CONCLUSION

Unfortunately, lack of test collections, query sets, and relevance judgements has made evaluation of these polyphonic features difficult. Currently we are creating collections of both MIDI and Humdrum files (www.musedata.org) to test these features.

Feature selection for text information retrieval has undergone decades of research. Word features, and the regular-expression technique for extracting words, are accepted standards. Higher-level features such as word stems, synonyms, and part-of-speech tags, or statistical measures such as *tf* and *idf* are common and fairly well understood.

Polyphonic music IR has not developed standardized methods for feature extraction, even for simple, low-level features such as words (or their equivalent analogue, if any [3]). Polyphonic features are a much more difficult challenge than text features. Better understanding of the motivations and principles underlying existing techniques, as well as an introduction of a number of new features, is needed. By categorizing existing techniques and proposing a musically motivated extension to one such technique, this work confronts both requirements.

## 5. REFERENCES

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

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

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

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

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