ANALYSIS OF A CONTOUR-BASED REPRESENTATION FOR MELODY

Youngmoo E. Kim *Wei Chai* *Ricardo Garcia* *Barry Vercoe*

MIT Media Laboratory
Machine Listening Group
20 Ames St., E15-401, Cambridge, MA 02139 USA
{moo,chaiwei,rago,bv}@media.mit.edu

ABSTRACT

Identifying a musical work from a melodic fragment is a task that most people are able to accomplish with relative ease. For some time now researchers have worked to give computers this ability as well, as it would be the cornerstone of any *query-by-humming* system. To accomplish this, it is reasonable to study how humans are able to perform this task, and to assess what features we use to determine melodic similarity. Research has shown that melodic contour is an important feature in determining melodic similarity, but it is also clear that rhythmic information is important as well. The goal of this research is to explore what variation of contour and rhythmic information can result in the most efficient, robust, and scalable representation for melody. We intend for this to be the basis of a query-by-humming system that will be used to test the validity of our proposed representation.

1. INTRODUCTION

Identifying musical works is a skill that comes fairly easy to humans. Often, when we turn on the radio in the middle of a familiar song, we are able to identify it within a few seconds (or even less). Development of this skill is taught in music classes (the so-called *drop-the-needle* test). In this task, many different features of the music beyond the notes themselves are used such as lyrics, instrumentation and timbre, tempo, and dynamics.

Perhaps what is more remarkable is that we can still identify music without those additional features and even without all of the notes; just using melody alone. Even in cases where the melody is cut short, corrupted, or rendered inaccurately we can often identify what song is being referred to.

This task highlights several features of melody. Melody is an efficient construct. Just a few notes of a tune can fully identify a piece of music. It is also a fairly robust feature, resistant to corruption. Even when rendered inaccurately, a melody can often be identified. It also scales easily to large data sets. Becoming familiar with more songs generally does not diminish our ability to identify ones we already know. Thus, it is desirable for any mid- to high-level representation for melody to also have these features (efficiency, robustness, and scalability).

This research is based upon a contour representation for melody that incorporates features of rhythm. This type of representation quantizes the intervals between notes more coarsely than traditional musical notation, suggesting only a general type of movement from pitch to pitch. The most common has been a 3-level (+/-/0) contour description scheme that simply identifies the interval as going up, down, or having no change from the previous pitch. Our proposed representation is based on a 5-level contour, designed to be used as the basis of a query-by-humming system. This description scheme has been submitted as a proposal to the MPEG-7 International Standard, which has the intention of defining meta-data for multimedia content.

2. BACKGROUND

This section presents background material for the research that follows in subsequent sections, particularly with regards to important perceptual aspects of melody. A definition of melody is given. Prior research into judgements of melodic similarity is examined, and melody representations used in existing music search and retrieval systems are also discussed.

2.1. Melody as an auditory and musical construct

Defining what exactly is or is not a melody can be somewhat arbitrary. Melodies can be monophonic, homophonic, or contrapuntal. Sometimes what one person perceives to be the melody is not what another perceives. A melody can be pitched or purely rhythmic, such as a percussion riff. Our research does not attempt to address all of these cases and is limited in scope to pitched, monophonic melodies.

Even with these limitations, arriving at a definition for melody can be difficult. Levitin describes melody as "an auditory object that maintains its identity under certain transformations… along the six dimensions of pitch, tempo, timbre, loudness, spatial location, and reverberant environment; sometimes with changes in rhythm; but rarely with changes in contour" [1]. Although rather broad, this definition highlights several important features of melody. People are able to recognize melodies even when they are played on different instruments, at different volumes, and at different tempi (within a reasonable range).

More importantly for our purposes, a melody can still be uniquely identified after it has undergone transposition (we still recognize a familiar tune in a different key as being the same tune). For this reason, absolute pitch is not the best descriptor for melodic pitch information. More important than the absolute pitches are the relative intervals between successive notes in a melody, since intervallic relations are

also invariant to key transposition. Since contour information is a subset of interval information, it is clear that contour is also invariant to transposition.

When identifying a melody, the listener perceives not only the pitch/interval information in the melody, but how those notes correspond to particular moments in time (i.e. rhythm). Rhythm is one dimension in which melodies in general can not be transformed. The following simple example illustrates the importance of consistent rhythmic information in melodic description.

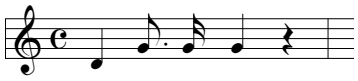

Figure 1: First four notes of the *Bridal Chorus* from *Lohengrin* (Wagner), i.e. *Here Comes the Bride.*

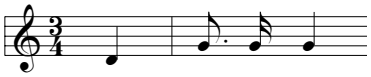

Figure 2: First four notes of *O Tannenbaum*.

It is apparent that these are two very distinct melodies, yet the sounding pitches, intervals, and note durations are identical. The difference lies not only in the respective meters (time signatures) of the songs, but in the position of the notes relative to the metric structure of each piece. The time signature of the first example is 4/4, and the strong beats occur on the first and third beats of a measure, which correspond to the first and fourth notes of the piece. The second example has a time signature of 3/4, and the strong beat is on the first beat of the measure, corresponding to the second note. From this example, we clearly see the advantages of incorporating rhythmic information in a melodic representation.

2.2. Systems for melodic similarity search and retrieval

Much work has been done in research and development of search and retrieval systems for melodic similarity. Existing systems use a variety of representations for melody, and usually aim for flexibility in order to accommodate variations in data and query format. Often, data is stored in a format analogous to traditional musical notation, such as MIDI [2][3][4][5]. However, some systems store contour information only [6][7].

The data used in different systems of course varies greatly, but consists primarily of classical and folksong repertoires, since the copyright on most of these works has expired and they are now in the public domain. The data is taken almost exclusively from Western music, or at least music using the Western tonal system (12 half-steps per octave). Again the reasons for this are primarily practical, since MIDI and other machine-based formats for storing notation were designed for Western tuning, and there are neither standard formats nor standardized methods of adapting existing formats for non-Western tuning.

Several of these database projects have implemented full query-by-humming systems [2][6][7]. This includes processing of an audio input signal in order to extract the necessary query information. However, parsing an acoustic signal to accurately and consistently produce note segmentation and pitch, interval, or contour information is a difficult task, which has been the focus of several papers [2][6]. As a result, these systems sometimes require sung queries to consist of discrete notes (separated by silence) [6], to use particular syllables (such as 'ta' or 'da' that are easy to separate into individual notes) [2], or to use whistling (instead of singing or humming) [7].

Most of these databases for melody search and retrieval have implemented sophisticated search engines to determine similarity and matches from a query. Systems that use only basic (3-level) contour information are the simplest to implement [5][6][7]. Other systems allow a variety of different queries, such as exact interval or finer (>3-level) contour information [2][3][4]. Rhythmic information is included in a few of the databases [2][3][4], but is ignored in others [5][6][7]. One of the query-by-humming systems only attempts to identify the beginnings of melodies [2].

In general, relatively little has been done to align machine-based melodic representations (for music databases) with our knowledge of human perception of melody. We believe that a mid/high-level perceptually-motivated representation of melody will also result in a representation that is more accurate and efficient for searching and similarity matching. The representation proposed in this paper suggests one possibility towards achieving that goal.

3. MELODY REPRESENTATION

In this section, we present our proposed representation for melody. We first discuss the significance of melodic contour and justify its use as the basis of our representation. This is followed by the specification of the notation and parameters of our representation.

3.1. The importance of melodic contour

It is clear that some type of interval information is important to representing melody, since melodic matching is invariant to transposition. However, instead of representing each interval exactly (e.g. ascending minor sixth, descending perfect fourth, etc.), the literature suggests that a coarser melodic contour description is more important to listeners in determining melodic similarity [8]. Experiments have shown that interval direction alone (i.e. the 3-level +/-/0 contour representation) is an important element of melody recognition [9]. As mentioned previously, several databases use this representation alone.

One possible reason for the importance of melodic contour is that this information is more easily processed and is at a higher (more general) level than interval information. But as one becomes more familiar with a melody and gains more musical experience, the specific intervals have greater perceptual significance [1].

There is, of course, anecdotal and experimental evidence that humans use more than just interval direction (a 3-level contour) in assessing melodic similarity. When recalling a melody from memory, most of us (not all!) are able to present information more precise than just interval direction. In an experiment by Lindsay [10], subjects were asked to repeat

(sing) a melody that was played for them. He found that while there was some correlation between sung interval accuracy and musical experience, even musically inexperienced subjects were able to negotiate different interval sizes fairly successfully. From a practical standpoint, a 3-level representation will generally require longer queries to arrive at a unique match. Given the perceptual and practical considerations, we chose to explore finer (5- and 7-level) contour divisions for our representation.

3.2. Proposed melody representation

We use a triple <*T P B*> to represent each melody, which we will refer to as *TPB* representation. *T* is the time signature of the song, which can change, but often does not. *P* is the pitch contour vector, and *B* is the beat number vector. The range of values of *P* will vary depending on the number of levels of contour used, but will follow the pattern of 0, +, -, ++, --, +++, etc. The first value of *B* is the location of the first note within its measure in beats (according to the time signature). Successive values of *B* are incremented according to the number of beats between successive notes. Values of *B* are quantized to the nearest whole beat. Clearly, the length of *B* will be one greater than the length of *P* because of the initial value.

Additionally, we use a vector *Q* to represent different contour resolutions and quantization boundaries. The length of *Q* indirectly reveals the number of levels of contour being used, and the individual values of *Q* indicate the absolute value of the quantization boundaries (in number of half-steps). For example, *Q* = [0 1] represents that we quantize interval changes into three levels, 0 for no change, + for an ascending interval (a boundary at one half-step or more), and - for a descending interval. This representation is equivalent to the popular +/-/0 or U/D/R (up/down/repeat) representation. *Q* = [0 1 3] represents a quantization of intervals into five levels, 0 for no change, + for an ascending half-step or whole-step (1 or 2 half-steps), ++ for ascending at least a minor third (3 or more half-steps), - for a descending half-step or whole-step, and -- for a descent of at least a minor third.

Thus, given a melody *M* and a resolution vector *Q*, we can get a unique contour representation:

$$\text{melody\_tpb}(M,Q) = \langle T\ P\ B \rangle. \quad (1)$$

4. EXPERIMENTAL RESULTS

In this section, we describe the results of three analyses aimed at evaluating three features: the relative importance of rhythmic information, an appropriate number of melodic contour levels, and the appropriate quantization boundaries for a given number of levels.

4.1. Data set

Thus far, we have assembled a data set of 50 multi-track MIDI files, containing a mixture of popular and classical music. The popular music selections span a variety of different countries. All selected songs had a separate monophonic melody sound track. A tool was written to find the melody track based on the track name, extract the melody information and represent it for each song in the following format:

nn	# numerator of time signature
dd	# denominator of time signature
ppqn	# PPQN (i.e. ticks per quarter note)
n_1	# midi note number of the first note in the melody
m_1	# measure number of the first note in the melody
b_1	# beat number of the first note in the melody
t_1	# tick number of the first note in the melody
n_2	# midi note number of the second note in the melody
m_2	# measure number of the second note in the melody
b_2	# beat number of the second note in the melody
t_2	# tick number of the second note in the melody
…	…

This data format was general enough to allow for a great deal of flexibility in final representation. The data was converted to the *TPB* representation in the following analyses.

4.2. Query Set

The query set was a randomly generated subset of the data set, containing 10 different queries. Since the query length (number of notes contained in the query) obviously affects the performance of the search (how many songs are found to match the query), we truncated each query into different lengths for the various experiments.

4.3. Methods

We next describe the testing algorithms used in our analysis.

4.3.1. Algorithm 1: Efficiency testing

This algorithm is used to examine similarity performance with different contour levels and quanitzation boundaries. It makes use of the scoring algorithm, which is described in the following section.

Procedure:

1. Select a finite set of *K* quantization vectors, $Q = \{Q_1, Q_2, \ldots, Q_K\}$ to be tested.
2. Quantize the beats of the data set *D*.
3. Randomly generate the query set *H* from *D*.
4. Convert the data set *D* and query set *H* into our *TPB* representation.
   a. Compute values of *P* for each using quantization vector Q_k, where $1 \le k \le K$.
   b. Compute *B* using absolute beat number instead of <measure beat tick> representation.

   This results in converted data and query sets D_k and H_k, respectively.
5. For each song d_{ik} (i=1, 2, …, N, where N is the number of songs in data set D_k) and each query h_{jk} (j=1, 2, …, M, where M is the number of queries in query set H_k), compute the score s_{ijk} (algorithm described below) representing how well h_{jk} matches d_{ik}, resulting in the $N \times M$ score matrix S_k.
6. For each query h_{jk}, count how many songs d_{ik} result in a score s_{ijk} greater than or equal to that of h_{jk} and

the song it was originally generated from. This results in the count vector, $C_k=[c_{jk}]$, (j=1, 2, …, M).

7. For the entire query set H_k, compute the overall performance, i.e. how many songs will match a query on average:

$$P_k = \frac{1}{M}\sum_{j=1}^{M} c_{jk} \qquad (2)$$

8. Compare the results for different quantization vectors Q_k.

4.3.2. Algorithm 2: Scoring algorithm

This algorithm computes the score s_{ij} of a song $d_i = <T_i\ P_i\ B_i>$ and query $h_j = <T_j\ P_j\ B_j>$ to evaluate how well they match. A higher score indicates a better match.

Procedure:

1. If the numerator of $T_i \neq T_j$, then return 0.
2. Initialize the measure number, m = 1.
3. Align P_j with P_i from the m^{th} measure of d_i.
4. Calculate the beat similarity score as follows:
   a. For each beat, tally the number of matches between the subsets of P_j and P_i that fall within the current beat.
   b. Divide this number by the length (number of values) of the query subset that falls within the current beat.

   Thus, the maximum beat similarity score is 1. For example, if the current beat of the song contains two intervals and the query contains three, of which the first two match the song intervals, the beat similarity score would be 2/3.
5. Average the beat similarity scores over the total number of beats in the query, resulting in the overall similarity score starting at measure m: s_{ij}^{m}.
6. If m is not at the end of d_i, then $m = m + 1$ and repeat step 3.
7. Return $s_{ij} = \max\{s_{ij}^{m}\}$, the best overall similarity score starting at a particular measure.

4.4. Results I: Importance of rhythmic information

In spite of anecdotal evidence (such as the examples from Section 2.1), we wanted to explicitly verify the usefulness of rhythmic information in comparing melodic similarity. To test this, we used the simplest contour (3-levels, Q=[0 1]) for queries with and without the rhythmic information vector, B. Our results clearly indicate that rhythmic information allows for much shorter (and thus more efficient) queries (Figure 3). A fewer number of matches indicates better performance.

4.5. Results II: Comparison of different quantization boundaries

We examined 3-, 5-, and 7-level contour representations. For the 5- and 7-level contours, we also examined a variety of quantization boundaries (different vectors Q_k). The results, in terms of average number of matches vs. query length (number of notes) are presented below in Figures 4 through 6.

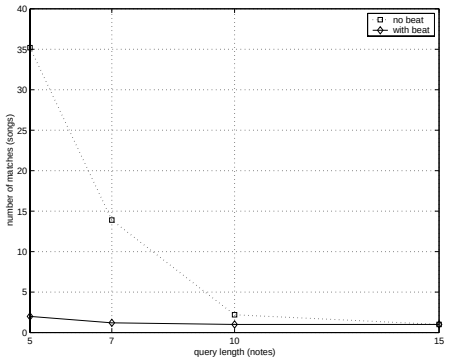

Figure 3: Comparison of representations with or without beat information. Both use 3 levels to represent pitch contours

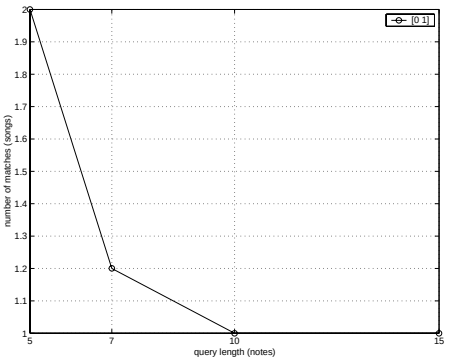

Figure 4: Performance of 3-level contour.

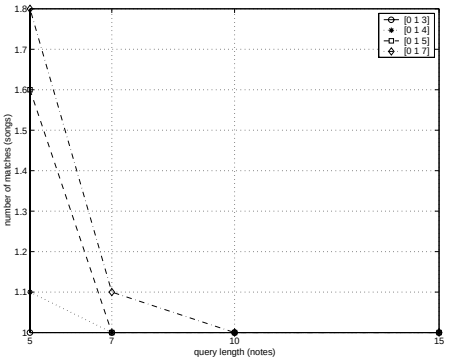

Figure 5: Performance of 5-level contour, with varying quantization boundaries, Q_k

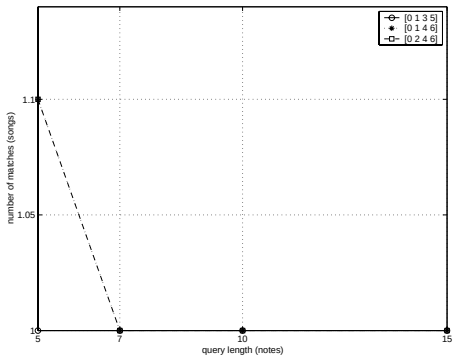

Figure 6: Performance of 7-level contour, with varying quantization boundaries, Q_k

It is clear that the performance of 5-level contours are generally better than the 3-level contour, and 7-levels is better than that. For quantization vectors, we limited our search to Q_k = [0 1 x …] cases only. Other values would have caused repeated notes (no interval change) to be grouped in the same quantization level as some amount of interval change, which does not make sense perceptually.

4.6. Discussion

It is an obvious result that greater numbers of levels in general result in more efficient searches. Clearly, more levels means more information, meaning less notes are needed to converge to a unique solution. What is more illuminating, is that the best 5-level contour was able to equal the performance of the 7-level contour. This suggests that a 5-level contour may be an optimal tradeoff between efficiency and robustness to query variation (more levels will cause more variations in queries).

Given our results, it is especially revealing to look at the histogram of interval occurrences in our data set.

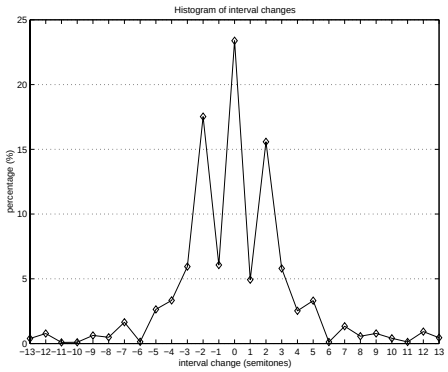

Figure 7: Interval distribution based on our data set.

From this histogram, it is clear why certain quantization levels perform better than others. An optimal quantizer would divide the histogram into sections of equal area. Thus, for a 5-level contour we would like each level to contain 20% of the data. This is approximately true for the Q = [0 1 3] case, which has the best performance. No interval change (0) occurs about 23% of the time. Ascending half-steps and whole-steps (+1 and +2) are about 21% of the intervals, whereas descending half- and whole-steps (-1 and -2) represent approximately 23%. Other choices for quantization boundaries clearly have less-optimal probability distributions, which is why they do not perform as well.

While this result is dependant on the statistics of the data set, it is worth noting that it also correlates well with our knowledge of melody perception. Others have noted the apparent correlation of statistical independence and perceptual importance in acoustic features, which supports a theory of perception evolving from statistical efficiency [11]. Perhaps it is not surprising that these relationships may exist in higher-level features, such as melody, as well. Some surely will argue the reverse causality: that human perception has driven the statistics of melody, resulting in a distribution of intervals that is pleasing to human perception. Either way, it is a useful relationship that perhaps has not yet been fully exploited. The statistical features of this description for melody result in an efficient representation. And since the representation correlates well with our perception of melody, the representation becomes more robust since our queries are likely to be more accurate.

5. FUTURE DIRECTIONS

Our conclusions are based on a rather small set of data, and we have not yet satisfied the third requirement stated at the beginning of the paper: scalability. Clearly, enlarging the data set to include a wider variety of musical works is a step towards evaluating the scalability of our melody description. We are continually adding new works to our database. It would certainly be informative to run the same analysis on other large melody databases. Validating these results for an independent data set would certainly lend more weight to our conclusions.

An interesting and informative experiment would be to apply this type of analysis to non-Western music, to see if the relationship between the statistical distribution of intervals and the perception of melody is maintained. The results might reveal a cultural bias in the distribution of intervals, or may indicate some cross-cultural consistencies in melodic perception.

Another direction would be to investigate the interval relationships between more than two notes, i.e. comparing not only a note with the previous note, but also to ones before that. A system which uses 2nd-order (3-note groups) matching is implemented in [5].

This research is one piece of a query-by-humming implementation. We hope to have the full system up and running by Fall 2000.

6. REFERENCES

[1] Levitin, D. J. “Memory for Musical Attributes” from *Music, Cognition, and Computerized Sound*, ed. Perry R. Cook. Cambridge, MA: MIT Press, 1999. pp, 214-215.

[2] R.J. McNab, L. A. Smith, I. H. Witten, C. L. Henderson, and S. J. Sunningham. “Toward the digital music library: tune

retrieval from acoustic input." *Proc. ACM Digital Libraries,* Bethesda, 1996. http://www.nzdl.org.
[3] Themefinder™, Stanford University, http://www.ccarh.org/themefinder.
[4] MiDiLiB, University of Bonn, http://leon.cs.uni-bonn.de/forschungprojekte/midilib/english.
[5] Search by Humming, University of Southampton, http://audio.ecs.soton.ac.uk/sbh.
[6] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith. "Query by Humming: musical information retrieval in an audio database." *Proc. ACM Multimedia*, San Francisco, 1995.
[7] TuneServer, University of Karlsruhe, http://wwwipd.ira.uka.de/tuneserver.
[8] Handel, S. *Listening*. Cambridge, MA: MIT Press, 1989.
[9] Dowling, W. J. "Scale and contour: Two components of a theory of memory for melodies." *Psychological Review,* vol. 85, no. 4, pp. 341-354, 1978.
[10] A. T. Lindsay. *Using contour as a mid-level representation of melody*. Unpub. MS thesis. MIT Media Lab, 1996.
[11] P. Smaragdis. "Information theoretic auditory grouping." *Proc. 3rd IJCAI Workshop on Computational Auditory Scene Analysis,* Stockholm, Sweden, 1999.