Encoding Timing Information for Musical Query Matching

Bryan Pardo
University of Michigan
EECS Department
1101 Beal Avenue, Ann Arbor, MI, USA
+1-734-369-3207
bryanp@umich.edu

William Birmingham
University of Michigan
EECS Department
1101 Beal Avenue, Ann Arbor, MI, USA
+1-734-763-1561
wpb@eecs.umich.edu

ABSTRACT
We compare representing note timing as Inter Onset Intervals (IOIs) and as the ratio of adjacent IOI values. A variety of log2 and linear quantizations of IOI and IOI ratios are considered for each representation. The utility of encoding with a particular quantization is measured by the ability of a simple string-matcher to differentiate between themes in a melodic corpus. Results indicate that time is best represented by IOI ratios quantized to a logarithmic scale.

1. MUSIC IR AND STRING MATCHING
Finding the best match between a melodic query and the melody of a target piece in a database has been a subject of great recent interest in the music IR world [1-5]. Both query and target are typically represented as a sequence of symbols drawn from a finite alphabet. Such sequences are commonly called strings.

Standard string matching [3, 6-8] measures the distance between two strings, S_1 and S_2, as the number of operations required to turn S_1 into S_2, given a fixed set of allowable operations. A typical set of edit operations would include deletion of a string element, insertion of an element, and matching between an element of S_1 and an element of S_2. Given a query string, Q, and a set of target strings drawn from a database, the closest target is the one with the lowest cost transformation into Q.

2. TEMPO SCALING WITH IOI RATIOS
Melody matching has, in general, concentrated on matching pitch contour. Typically, rhythm is encoded implicitly. The number of contiguous, fixed-duration frames with the same pitch indicates note length. Figure 1 shows three sequences as piano roll. Vertical lines indicate the boundaries of fixed-duration frames. A letter above each frame indicates the pitch class present in that frame.

From the figure, it is clear that S_1 and S_2 represent the same melody performed at two different speeds and S_3, while containing all the notes in S_1, is another melody entirely. A string matcher, using the three edit operation from the previous section and the frame-based representation, finds the least-cost alignment between S_1 and S_2 is the same as the least-cost alignment between S_1 and S_3. The problem of dealing with tempo scaling issues may be resolved by explicitly representing durations and normalizing them to a reference duration, d_{ref}.

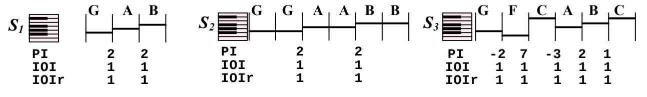

Figure 1. Three Sequences

Ideally, d_{ref} would be the duration of a beat, as in written notation. Unfortunately, it is not always clear what the beat is in a sequence generated from a sung query. A simple substitute for the beat is the duration of the previous event. If note durations are measured using the *Inter Onset Interval* (IOI), the ratio between the current and next IOI may be used. We call this the *IOI ratio* (IOIr). Figure 1 shows pitch interval (PI) as measured in half steps, IOI (measured in frames) and IOI ratio. Here, the final IOI ratios default to unity. Representing the strings as <PI, IOIr> duples make S_1 and S_2 identical, while S_3 is distinctly different.

3. ALPHABET SIZE
Advanced string matchers [6] allow for inexact matches between strings, based on the likelihood of co-occurrence between element i of S_1 and element j of S_2. Allowing for inexact matches in melody matching provides a way to model singer error. This is done by creating a probability distribution over the set of possible observations for each possible note. Given that the size of the alphabet of possible observations, n, is the same as the size of the alphabet of notes, the full observation-probability table is size n^2.

Observation-probability tables require many observation-state pairs to provide good estimates of the probabilities. As the size of the alphabet increases, so does the difficulty of collecting sufficient data to fill the table with good probability estimates; without good estimates, it is not possible to have a good matcher.

4. EXPERIMENTAL SECTION
If we represent notes as <PI, IOIr> duples, the size of the observation-probability table increases quadratically as the number of duration values increases linearly. Given this, we want to minimize the number of duration values used, without hurting a matcher's ability to differentiate melodies based on rhythm. This section compares different encodings of IOI and IOIr values, with these goals in mind.

4.1 Corpus Construction
We compiled 417 MIDI files created from themes in an index of themes of classical pieces [9]. Themes were entered into the music-typesetting program Finale and saved as MIDI by undergraduate music students at our university. For this corpus, the largest IOI is 5100 milliseconds. The smallest is 50 milliseconds, 102 times shorter than the largest. IOI values in the corpus fall into 35 unique values, resulting in 82 unique IOI ratios.

4.2 Time Encoding
We encoded only timing information (i.e., no pitch information) for each theme in the corpus as either IOI, or IOI ratios. Values were encoded as integers using a 2^n level quantization, with a saturation point of s. Bins were spaced evenly, either in the linear domain or in the log2 domain. Log2 bins were evenly distributed in the range $–s$ and s. Linear bins were evenly distributed between 0 and s. Values greater than s were put in bin number 2^n-1. Log2 values less than $–s$ were placed in bin 0. Linear values below 0

were placed in bin 0. Linear vs. log2 bins and values for *n* and *s* were varied as experimental parameters.

4.3 Encoding Evaluation

Given a number of bits for the encoding, *n*, and a saturation point, *s*, timing information for every theme in the corpus was encoded to an alphabet of size 2^n. We encoded timing information in the 417-theme corpus a total of 336 different ways. The data were encoded as either IOI or IOI ratio. Values were scaled either linearly or as log2. The number of bits, *n*, for the encoding was varied between 2 and 8 (giving between 4 and 256 bins). Finally, the saturation point, *s*, was varied between 1 and 12. For consistency between the log2 and linear encodings, the saturation point became the saturation exponent in linear encoding. Thus, saturation point of 5 in the log2 world is saturation point 2^5in the linear world.

A query set of 30 themes was chosen at random from the corpus. This query set was used consistently through all trials. For each trial, an encoding of the data was selected and a simple string matcher was used to rank all themes in the corpus by similarity of timing information (no pitch information was used) to each of the 30 themes in the query set. Ideally, when theme *i* is the query, the single best-scoring theme is theme *i*. If the encoding makes some themes indistinguishable, the number of themes with the maximal score increases. Thus, the average number of themes receiving a maximal similarity score provides a relative measure of how well an encoding preserves timing information.

4.4 Results

The best value possible for an encoding is 1 (the query melody is always the single best match). The worst possible value depends on how many themes in the corpus have the same length as the query. This is because the matcher can always distinguish themes based on number of events, even if all events are given the same label. For the query set, the average number of themes the same length as a query is 26. Figure 2 show the mean number of maximal-scoring themes returned in each trial, broken down by encoding, scaling, number of bits, and saturation point.

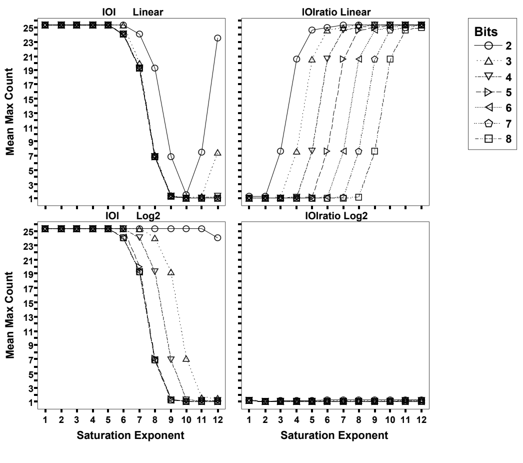

Figure 2. Count of themes indistinguishable from correct theme by encoding and scaling

The data show that, given a good choice of saturation point, it is possible to represent rhythmic values quite well using IOI with as few as 2 bits to encode the data. In the case of this corpus, the best saturation exponent point is at 10. This corresponds to 2^{10}, or 1024 milliseconds. The data show that IOI is extremely sensitive to the choice of saturation point. Log2 based encoding of the IOI data helps some when the saturation point is set too high, but does not help when the saturation point is too low.

Linear IOI ratios show the best saturation exponent is around 1. As the saturation point is placed further from it, more bits are needed in order to maintain sufficient resolution below 1. Log2 IOI ratios (LIRs) are a different story. The data show LIRs are extremely insensitive to variations in saturation point and number of bits. In fact, the worst result returned by any LIR encoding was an average of 1.3 themes per query.

5. CONCLUSIONS

The use of explicit representation of time, coupled with expressing timing variation as IOI ratios, normalizes rhythmic information between two performances at different tempos. Explicit encoding of time increases the alphabet size, which makes creating observation probability tables more difficult.

Encoding timing as the log2 of IOI ratios preserves sufficient information to discriminate between themes in our corpus using as few as 2 bits (4 bins), minimizing the size of the state alphabet. This allows the use of explicit representation of rhythmic relation that is invariant with respect to tempo and still manageably small in alphabet size, indicating this is a promising method for encoding timing information for musical query retrieval.

6. ACKNOWLEDGMENTS

We gratefully acknowledge National Science Foundation grant IIS-0085945, and The University of Michigan College of Engineering seed grant to the MusEn project. The opinions in this paper are solely those of the authors and do not necessarily reflect the opinions of the funding agencies.

7. REFERENCES

[1] Mazzoni, D. and R.D. Dannenberg. Melody Matching Directly From Audio. ISMIR. 2001. Bloomington, IN.

[2] Bainbridge, D., MELDEX: A Web-based Melodic Locator Service, in Computing in Musicology, W.B. Hewlett and E. Selfridge-Field, Editors. 1997, The MIT Press: Cambridge, MA. p. 223-229.

[3] Lloyd Smith, R. McNab, and I. Witten, Sequence-Based Melodic Comparison: A Dynamic-Programming Approach, in Computing in Musicology, W.B. Hewlett and E. Selfridge-Field, Editors. 1997, The MIT Press: Cambridge, MA. p. 101-117.

[4] Shifrin, J., et al. HMM-Based Musical Query Retrieval. in JCDL. 2002. Portland, Oregon, USA.

[5] Hoos, H., K. Renz, and M. Gorg. GUIDO/MIR - an Experimental Musical Information Retrieval System based on GUIDO Music Notation. In ISMIR. 2001. Bloomington, IN.

[6] Gotoh, O., An improved algorithm for matching biological sequences. Journal of Molecular Biology, 1982. 162: p. 705-708.

[7] Gusfield, D., Algorithms on Strings, Trees, and Sequences. 1997, New York, NY: The Press Syndicate of the University of Cambridge.

[8] Needleman, S.B. and C.D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology, 1970. 48: p. 443-453.

[9] Barlow, H. and S. Morgenstern, A Dictionary of Musical Themes. 1975, New York , New York: Crown Publishers.