THE IMPORTANCE OF SEQUENCES IN MUSICAL SIMILARITY

Michael Casey

Goldsmiths College, University of London
New Cross, London SE14 6NW
m.casey@gold.ac.uk

Malcolm Slaney

Yahoo! Research
Sunnyvale, CA 94089
malcolm@ieee.org

ABSTRACT

This paper demonstrates the importance of temporal sequences for passage-level music information retrieval. A number of audio analysis problems are solved successfully by using models that throw away the temporal sequence data. This paper suggests that we do not have this luxury when we consider a more difficult problem; that is finding musically similar passages within a narrow range of musical styles or within a musical piece itself. Our results demonstrate a significant improvement in performance for audio similarity measures using temporal sequences of features, and we show that quantizing the features to string-based representations also performs well, thus admitting efficient implementations based on string matching.

1. INTRODUCTION

This paper describes methods for incorporating temporal information into methods for passage-level retrieval from musical audio. In our study, passages that are considered similar by a human listener, but that are acoustically distinct, are considered equivalent. An example application is retrieval of all the thematic repe[Submitted to ICASSP 2006. Confidential until published.] melodies) in classical works or popular music tracks[illegible] ness to changes in the lyrics, instrumentation, tempo, rhythm, chord voicings and so forth. The applications of such similarity methods are far-reaching, and have immediate relevance to music browsing, computational musicology, audio thumbnailing, music structure identification and audio synthesis by *musaicing*.

Previous studies in music retrieval have used various spectral features such as timbre [1] or chroma [2] to generate time series of data vectors that are then treated as unordered sets, or "bags of frames," and applied to document-level (i.e. whole work) classification and similarity tasks. Examples are genre classification, artist recognition, musical key classification and speaker identification. The "bags of frames" approach has also been used successfully for identification of specific acoustic content via a noisy channel, e.g. music fingerprinting applications, where the time alignment between query and target feature sequences is guaranteed to be one-to-one but with distortions due to noise masking or broadcast channel dynamic effects processing, [4]

One of our primary motivations for this study is to support applications of passage-level retrieval to song databases on the order of millions of documents, with the least possible computational complexity. To achieve this we need to find highly efficient methods, such as enabled by symbolic hashing. Our ultimate goal is to perform passage-level similarity retrieval with the same efficiency that text-based information retrieval systems perform document retrieval from large collections, such as the world-wide web.

This research was supported, in part, by EPSRC grant GR/S84750/01 (Hierarchical Segmentation and Semantic Markup of Musical Signals).

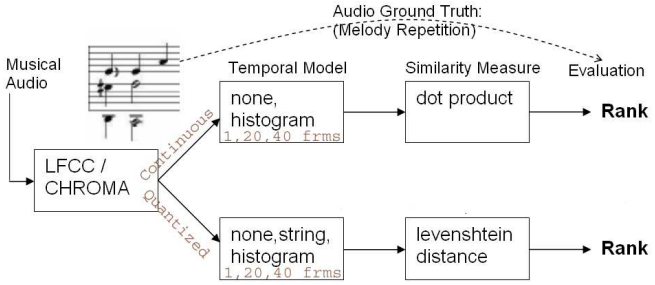

Fig. 1. Overview of the audio feature extraction, temporal modeling and evaluation of quantized and continuous features.

This paper continues with a description of our method for generating quantized features for audio, similarity matching and evaluation in section 2; we present empirical results in section 3, and draw [illegible] section 4.

2. METHOD

Figure 1 shows an overview of our processing system. First we extract continuous-valued feature vectors from two different musical data sets described in section 3. Then the processing chain follows two paths yielding both quantized features and continuous features. The processing chain is further split into features that did not utilize temporal order information, and those that did. We perform evaluation using each of the features and the rank of returned segments for each ground truth query is returned for each end node in the processing chain. We discuss the individual stages in more detail below.

2.1. Feature Extraction

The processing chain begins with a 44.1kHz sampled audio signal and breaks it into a sequence of short overlapping windows; using a window size of 375ms with a hop size of 100ms to generate a constant-Q power spectrum with $\frac{1}{12}$th-octave resolution, corresponding to pitch classes in Western tonal music, [7]. This power-spectrum is log-normalized and a linear transform applied consisting of DCT basis functions, for log frequency cepstral coefficients (LFCC), and octave-equivalence basis functions for chromagram features. Both of these representations are similar to features used in previous work in musical audio information retrieval, such as [10] and [2],

2.1.1. Vector Quantization

The LFCC and chroma vectors are used to train a vector quantization (VQ) model by unsupervised learning over a fixed third of the training data. VQ has been successfully employed in a wide-range of applications in automatic speech recognition and music information retrieval– aside from passage-level music retrieval which is the subject of our study. Three K-means models with 8, 16 and 64 clusters were used to generate sequences of cluster indicies using nearest-cluster assignment; Euclidean distance between K cluster centers and each of the feature vectors was used for the assignment. The sequence of cluster assignments for each song was then further processed to produce short-term windows of clusters of 2s and 4s duration with a hop size of 0.1s . For each work in the data set, the windowed VQ sequences were represented in a database as fixed-length strings using the ASCII base-64 encoding standard which maps binary indices to a character from the ASCII encoding standard. The VQ strings can be understood as a new feature with the same dimensionality as the window length used to produce them.

2.2. Temporal Matching

We looked at two different forms of acoustic matching: a continuous-domain matched filter and a string-to-string distance using discrete symbols for each acoustic frame. This same distinction between discrete and continuous distributions was also an issue when HMM speech recognizers were first built [3].

2.2.1. Matched Filter

We measure acoustic similarity in the continuous domain by computing vector differences between the feature data; ignoring the first coefficient, which encodes acoustic energy. We match longer sequences of music by integrating this frame-by-frame measure over a rectangular window in time, implementing a simple matched filter for each query. This window, in our work, is either 1, 20, or 40 (100ms) frames in duration.

2.2.2. Approximate VQ String Matching

We expect the VQ string sequence between a query segment and each of its ground truth target segments to be slightly different due to the distinctness of acoustic phenomena in performance. However, what we seek is a representation that is equivalent when two musical passages are considered equivalent *from the listenser's perspective*. This is precisely what makes musical similarity matching difficult to achieve.

Similarity for strings of states was computed using the Levenshtein distance metric, or *string edit distance*, which counts the minimum number of insertions and deletions (indels) and substitutions (swaps or replacements) required to make a query string match the target string. The Levenshtein distance that we used treats all substitutions as equally eroneous, [5]. This is, perhaps, not a bad approximation when quantizing in a high-dimensional space because symbols tend to be equidistance from each other.

2.2.3. Exact Matching on Fuzzy VQ Strings

Our primary motivation is toward scalable audio-similarity matching for music applications with a large number of documents. In pursuit of this goal, we recognize that a possible implementation is a hash-table lookup over the VQ strings. However, our target task consisted in repetitions of melodic segments subject to variations in timing, timbre, pitch voicing, instrumentation, rhythmic content and lyrics (voice content) due to natural variation in musical performance. Thus, we expect that melodic repetitions will not result in literal string repetitions, but that they might be subject to temporal re-orderings (swaps), insertions and deletions.

We require a method to represent strings where such likely confusions are computed within the string representation itself, rather than by a similarity computation at query time, which is computationally expensive. To this end, we made a new feature consisting of the set of unqiue state labels that occured in each VQ string ordered deterministically by alphabetical ordering of cluster labels. For example, the length 10 string (zzaaabbzbc) was represented by the indicator histogram (abcz), as is the length 5 sequence (zcbca).

The indicator histograms are variable in their dimensionality (length). The miniumum dimensionality is one; which occurs when a single cluster occupies the entire string window. The maximum dimensionality is the length of the VQ string window; occuring when each time point is occupied by a unique cluster instance. Such VQ indicator histograms are similar to a mixture model describing a weighted combination of states; in our case, the weights were drawn from $[0, 1]$. Exact matches in this representation are invariant to temporal ordering of the VQ strings, thus temporal information is eliminated in our fuzzy indicator histogram representation.

We also calculated the confusability between symbols such that symbols with a high probability of substitution, for a repeated melody, were combined to make a new symbol that was invariant to such confusions. However, there is not sufficient space in this article to include a comprehensive discussion and evaluation of this concept. So we focus instead on the comparison of continuous and quantized features that either used temporal information or not.

2.2.4. String Similarity

The Levenshtein distance is defined as the minimum number of insertions, deletions and swaps that are required to make a test sequence into a query sequence. For example, lev('abc','abc')$=$ 0, lev(abc,abbc)$=$ 1, and lev(abc,cba)$=$ 2. Efficient computation of the Levenshtein distance is the subject of much research across computational disciplines, and has been explored in great detail for applications such as biological sequence comparison and text-based information retrieval, [6, 5]. In most applications, a dynamic programming algorithm is used to find the minimum possible distance, with implementations being worst-case polynomial order 2 with respect to the string length and linear with respect to the number of strings in the database. Exact string matching by hash table lookup, in contrast, has computational complexity on the order of constant time with respect to both string length and the number of strings in the database,[5].

2.3. Experimental framework

We collected two data sets for evaluation over contrasting musics. The first was a corpus of classical music works performed on a real piano and recorded in a reasonably reverberation-free acoustic space. Human performances of three classical works by Bach and Beethoven were recorded into a MIDI file via a Clavinova and played back on a Yamaha Disklavier, an electronically-driven acoustic piano. Each performance was then marked up for repeated melodic content using an audio editor. The pieces were selected for their considerable use of repetition throughout.

The second dataset consisted of 7 popular music tracks with markup for melodic repetition provided by the MPEG-7 Audio group.

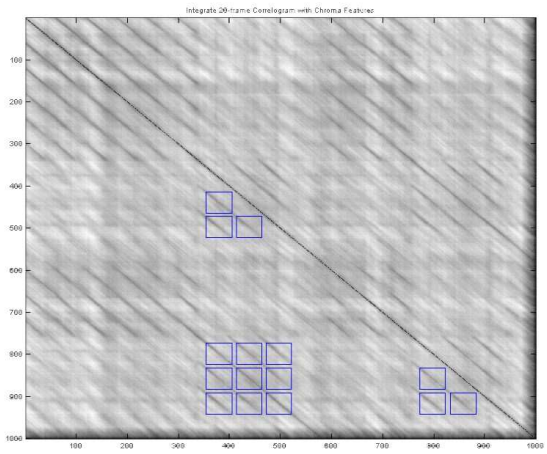

Fig. 2. Matched-filter temporal distance matrix (30-frame window) with the correlation mask for ground-truth repeats overlayed on the lower-triangular portion.

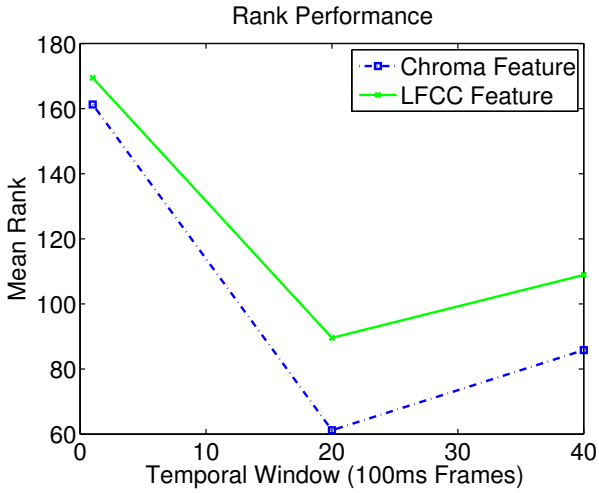

Fig. 3. Mean retrieval rank performance of LFCC and Chromagram featues.

Each repeat of a melody was indicated by its start time in seconds and milliseconds. Three of the classical works were marked up into a total of 24 repeated melodic segments, and the popular music tracks were also marked up into a total of 24 repeated melodic segments. We generated all possible pairs of ground-truth query/result segments for each song; i.e. C_2^n combinations for n repeated instances of a melody.

It was difficult to establish a ground truth for melodic repetition in both datasets, and especially in the popular works. A value judgement had to be made on wether repetitions were perceptually distinct or not. We restricted use of the popular music ground truth to those repeated melodic segments that occured within the chorus sections of the works. This is because the instrumentation in popular music is often not established in first verse, thus repeats of the verse melodic phrases are subject to more acoustic variation than repeats of melodic materials in the choruses. The decision to use choruses only may be justified by stating that repetitions in both chorus and verses are acoustically distinct, but that the verse repetitions are much more so. Also, one can argue that choruses are often the 'important' part or most memorable part of a song.

2.4. Evaluation by Result Rank

We devised an evaluation method that allows us to meaningfully compare the performance of different features. A target correlation matrix was constructed that consisted of a set of ground-truth target regions in the feature correlation matrix for each possible query. This temporal sequence correlation matrix is an extension of the S-matrix described in [1]. The S-matrix approach uses a window length of 1. Our measure of sequence similarity was calculated for each time point using 1-frame, 20-frame and 40-frame windows. Thus each window consisted of either a series of vectors, in the case of LFCC and Chroma features, or a sequence of states, in the case of VQ features, as described in Section 2.

For each song a relevance mask, of the same dimension as the correlation matrix, was computed based on the ground truth of repeated melodic fragments for the song, see Figure 2. Retrieved segments were rank-ordered by the distance measure, Euclidean distances in the case of LFCC and chroma features, and Levenshtein distance in the case of VQ features.

It is interesting to note that when we observed the distance correlation matrices for VQ string-based features, especially for window sizes more than 1-frame long, they were surprisingly similar to that shown in Figure 2 for continuous features.

3. RESULTS

We want each query to return all matching segments of the song with the lowest possible rank. In our implementation, the rank will never be one because the main diagonal has zero intra-frame distance and many queries will match over a small range of starting points. Because the songs in our test are fixed, the ranking measure allows us to rank alternative algorithms, even when the average ranks are not close to 1.

Figure 3 shows two results for the continuous-matching domain. First the ranking performance was significantly better for time windows of 20 and 40 frames (2 and 4 seconds) than it was for frame-by-frame acoustic comparisons. This is not surprising since the extra frames are near-exact matches and significantly increase the precision of each query (by reducing the number of false positive matches). Performance was slightly worse for 40-frame windows than for 20-frame windows, perhaps because minor tempo variations build up over the longer windows and lead to lower recall rates.

Secondly, Figure 3 shows that chroma features perform much better than LFCC features. Contrary to previous genre-recognition studies, and in spite of the minor differences between our LFCC and their MFCC representations, the note-based chroma representation is better at matching intra-song repeats than the LFCC-based timbre measure.

For quantized features, the case of one symbol determining a match with a small number of VQ symbols leads to large distance estimates and poor ranking results. So the single-frame VQ feature case is not included in the evaluation.

Therefore, Figure 4 shows our results using quantized features for window lengths of 20 and 40 frames. The most striking result is

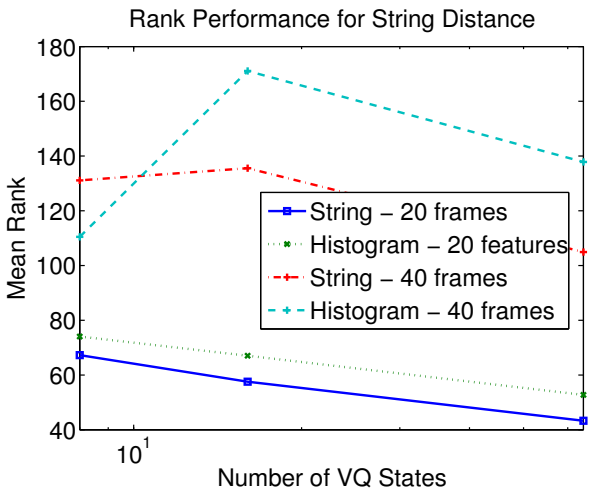

Fig. 4. Mean retrieval rank performance VQ features for three levels of quantization and two window lengths

that the Levenshtein distance in a quantized acoustical space works well for music similarity judgments when two to four seconds of audio are used as a musical query. For both short and long windows, the string-based features worked better than the histogram features, which throw away the temporal ordering and create a bag-of-frames representation of the sound. We are surprised to see that shorter time windows worked better than the longer windows. Perhaps our methods could not account for the extra variability due to the longer strings.

In general, both representations and window sizes got better as we increased the number of VQ symbols. This is to be expected because the quantization error is smaller and thus matching errors due to the underlying quality of the representation are diminished.

4. CONCLUSIONS

We have demonstrated the importance of temporal features in a music-similarity task. We looked at several different forms of musical representation and distance measures. We showed that temporal queries were more effective at retrieving musically similar segments of our music library. Because the distance measures were so different, we are not able to directly compare the performance of the continuous and the string-based retrieval methods. We are encouraged that string-based methods work so well, and we now wish to study efficient methods to perform fuzzy-string matching so we can apply our ideas to today's million-song libraries.

5. REFERENCES

[1] Jonathan Foote, "Visualizing music and audio using self-similarity," in *ACM Multimedia (1)*, 1999, pp. 77–80.

[2] Mark A. Bartsch and Gregory H. Wakefield, "To Catch a Chorus: Using Chroma-Based Representations for Audio Thumbnailing," in *Proc. WASPAA*, 2001.

[3] J. R. Bellegarda and D. Nahamoo, "Tied mixture continuous parameter models for large vocabulary isolated speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, pp. 13–16., 1999

[4] J. Herre, E. Allamanche, O. Hellmuth, T. Kastner, "Robust identification/fingerprinting of audio signals using spectral flatness features", *Journal of the Acoustical Society of America*, Volume 111, Issue 5, pp. 2417-2417 (2002).

[5] David Sankoff and Josepth Kruskal, *Time warps, String Edits and Macromolecules. The Theory and Practice of Sequence COmparison,* Mass.: Addison-Wlesey, 1983.

[6] Levenshtein, V.I.,"A binary code capable of correcting spurious insertions and deletions of ones," *Cybernetics and Control Theory*, 10(8):707-710, (1966)

[7] J.C. Brown and M.S. Puckette. "An efficient algorithm for the calculation of a constant Q transform." *J. Acoust. Soc. Am.,* 92(5):2698–701, November 1992.

[8] Jan Puzicha, Thomas Hofmann, and Joachim M. Buhmann, "Histogram clustering for unsupervised image segmentation," *Proceedings of CVPR '99*, 1999.

[9] C. J. van Rijsbergen, *Information Retrieval*, Butterworth, 1979.

[10] Eric Scheirer and Malcolm Slaney, "Construction and evaluation of a robust multifeatures speech/music discriminator," *IEEE Transactions on Acoustics, Speech, and Signal Processing* (ICASSP'97), 1997, pp. 1331 – 1334.