Music Database Retrieval Based on Spectral Similarity

Cheng Yang*
Department of Computer Science
Stanford University
yangc@cs.stanford.edu

Abstract

We present an efficient algorithm to retrieve similar music pieces from an audio database. The algorithm tries to capture the intuitive notion of similarity perceived by human: two pieces are similar if they are fully or partially based on the same score, even if they are performed by different people or at different speed.

Each audio file is preprocessed to identify local peaks in signal power. A spectral vector is extracted near each peak, and a list of such spectral vectors forms our intermediate representation of a music piece. A database of such intermediate representations is constructed, and two pieces are matched against each other based on a specially-defined distance function. Matching results are then filtered according to some linearity criteria to select the best result to a user query.

1 Introduction

With the explosive amount of music data available on the internet in recent years, there has been much interest in developing new ways to search and retrieve such data effectively. Most on-line music databases today, such as Napster and mp3.com, rely on file names or text labels to do searching and indexing, using traditional text searching techniques. Although this approach has proven to be useful and widely accepted, it would be nice to have more sophisticated search capabilities, namely, searching by content. Potential applications include "intelligent" music retrieval systems, music identification, plagiarism detection, etc.

Most content-based music retrieval systems operate on score-based databases such as MIDI, with input methods ranging from note sequences to melody contours to user-hummed tunes [2, 5, 6]. Relatively few systems are for raw audio databases. Our work focuses on raw audio databases; both the underlying database and the user query are given in .wav audio format. We develop algorithms to search for music pieces similar to the user query. Similarity is based on the intuitive notion of similarity perceived by humans: two pieces are similar if they are fully or partially based on the same score, even if they are performed by different people or at different tempo.

See our full paper [12] for a detailed review of other related work [1, 3, 4, 7, 8, 9, 10, 11, 14].

2 The Algorithm

The algorithm consists of three components, which are discussed below.

1. Intermediate Data Generation.

   For each music piece, we generate its spectrogram, and plot its instantaneous power as a function of time. Next, we identify peaks in this power plot, where peak is defined as a local maximum value within a neighborhood of a fixed size. Intuitively, these peaks roughly correspond to distinctive notes or rhythmic patterns, with some inaccuracy that will be compensated in later steps. We extract the frequency components near each peak, taking 180 samples of frequency components between 200Hz and 2000Hz. This gives us n spectral vectors of 180 dimensions each, where n is the number of peaks obtained. After normalization, these n vectors form our intermediate representation of the corresponding music piece.

2. Matching.

   In this step, two music pieces are compared against each other by matching spectral vectors in the intermediate data. We associate a "distance" score to each matching by computing the sum of root-mean-squared errors between matching vectors plus a penalty term

*Supported by a Leonard J. Shustek Fellowship, part of the Stanford Graduate Fellowship program, and NSF Grant IIS-9811904.

for non-matching items. A dynamic programming approach is used to find the best matching that minimizes this distance. Furthermore, a "linearity filtering" step is taken to ensure that matching vectors reflect a linear scaling based on a consistent tempo change.

3. Query Processing.

   All music files are preprocessed into the intermediate representation of spectral vectors discussed earlier. Given a query sound clip (also converted into the intermediate representation), the database is matched against the query using our minimum-distance matching and linearity filtering algorithms. The pieces that end up with the highest number of matching points are selected as answers to the user query.

See [12] for details and analysis of the algorithm.

3 Experiments and Future Work

We identify five different types of "similar" music pairs, with increasing levels of difficulty:

- Type I: Identical digital copy
- Type II: Same analog source, different digital copies, possibly with noise
- Type III: Same instrumental performance, different vocal components
- Type IV: Same score, different performances (possibly at different tempo)
- Type V: Same underlying melody, different otherwise, with possible transposition

Sound samples of each type can be found at `http://www-db.stanford.edu/~yangc/musicir/`.

Tests are conducted on a dataset of 120 music pieces, each of size 1MB. For each query, items from the database are ranked according to the number of final matching points with the query music, and the top 2 matches are returned. For each of the first 4 similarity types, retrieval accuracy is above 90%. Type-V is the most difficult, and better algorithms need to be developed to handle it.

We are experimenting with indexing schemes [13] in order to get faster retrieval response. We are also planning to augment the algorithm to handle transpositions, i.e., pitch shifts.

References

[1] J. P. Bello, G. Monti and M. Sandler, "Techniques for Automatic Music Transcription", in *International Symposium on Music Information Retrieval*, 2000.

[2] S. Blackburn and D. DeRoure, "A Tool for Content Based Navigation of Music", in *Proc. ACM Multimedia*, 1998.

[3] J. C. Brown and B. Zhang, "Musical Frequency Tracking using the Methods of Conventional and 'Narrowed' Autocorrelation", *J. Acoust. Soc. Am.* 89, pp. 2346-2354. 1991.

[4] J. Foote, "ARTHUR: Retrieving Orchestral Music by Long-Term Structure", in *International Symposium on Music Information Retrieval*, 2000.

[5] A. Ghias, J. Logan, D. Chamberlin and B. Smith, "Query By Humming – Musical Information Retrieval in an Audio Database", in *Proc. ACM Multimedia*, 1995.

[6] R. J. McNab, L. A. Smith, I. H. Witten, C. L. Henderson and S. J. Cunningham, "Towards the digital music library: Tune retrieval from acoustic input", in *Proc. ACM Digital Libraries*, 1996.

[7] E. D. Scheirer, "Pulse Tracking with a Pitch Tracker", in *Proc. Workshop on Applications of Signal Processing to Audio and Acoustics*, 1997.

[8] E. D. Scheirer, *Music-Listening Systems*, Ph. D. dissertation, Massachusetts Institute of Technology, 2000.

[9] A. S. Tanguiane, *Artificial Perception and Music Recognition*, Springer-Verlag, 1993.

[10] G. Tzanetakis and P. Cook, "Audio Information Retrieval (AIR) Tools", in *International Symposium on Music Information Retrieval*, 2000.

[11] E. Wold, T. Blum, D. Keislar and J. Wheaton, "Content-Based Classification, Search and retrieval of audio", in *IEEE Multimedia*, 3(3), 1996.

[12] C. Yang, "Music Database Retrieval Based on Spectral Similarity", *Stanford University Database Group Technical Report* 2001-14. `http://dbpubs.stanford.edu/`

[13] C. Yang, "MACS: Music Audio Characteristic Sequence Indexing for Similarity Retrieval", in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2001.

[14] C. Yang and T. Lozano-Pérez, "Image Database Retrieval with Multiple-Instance Learning Techniques", *Proc. International Conference on Data Engineering*, 2000, pp. 233-243.