A Musical Query-By-Humming Implementation Project

JJ Robertson, David Ramirez, Riyan Setiadji, and Rashed Alneyadi
IAFSE, Arizona State University

***Abstract*—Query-by-humming (QBH) is the process of matching a user query, which may be whistling, humming, or singing, to a target song. Many different QBH systems have been implemented, each with a different music encoding, matching algorithm, or transcription technique, with varying degrees of success. Even the state-of-the-art QBH systems offer modest performance, and there is still some distance to go before commercial viability is reached. A brief survey of QBH research as well as some details of this project's implementation are presented.**

I. Query-By-Humming Overview

Audio signal processing research includes compression, enhancement and recognition of sounds. In his project, we address audio recognition and specifically the problem of Query-by-humming (QBH). QBH remains an unsolved problem within the music information retrieval (MIR) domain. A QBH system allows a user to hum, sing, or provide some other means of imitating a song, and will be able to return the song in question. This has proven to be a challenging problem over the years, and a commercially viable system has not yet been created. This is despite attempts from a number of organizations and an abundance of research literature.

Tunebot, developed by Bryan Pardo is a well-documented QBH system [5-6]. The system involves matching a sequence of user produced musical notes to a database of song notes. First human vocalizations are quantized using pitch estimation and a note-like data encoding scheme. Next, this sequence is compared to a database using a variety of string-matching or dynamic-programming-based comparison techniques. Recent publications related to Tunebot development describe a crowdsourced matching model [7-8]. To our knowledge somewhat similar methods were used in the SoundHound system. Both of these systems are highly dependent on learning from user submitted queries. In such a system any user can submit their own recorded version of a song. To improve performance, a song database can be seeded with professionally recorded vocals, as in [8] and [9].

Instead of including complete songs within a database, many implementations simplify into meaningful musical themes. A musical theme is defined as a sequence of notes as in a melody. One target song will yield many such musical themes with some of the most prominent being prioritized. A user is likely to query a recognizable and distinct theme of a part of a song. The theme extraction technique removes extraneous information in our targets, thereby improving speed and matching. Musical theme and pattern extraction is a rich research area and algorithms in [9-11] will be leveraged to create such musical themes.

II. Implementation

To create an operational QBH system the first step is transcribing the pitch of the user query. The harmonically complex waveform is simplified to a sequence of single notes based on the pitch. This information is aggregated by the system into <pitch, time_start, time_end> tuples representing each note event. This procedure is known as monophonic music transcription (single instrument).

When building a database of searchable songs or themes, the MIDI format is utilized. Transcribing a polyphonic musical recording into musical notes is a very challenging problem. Instead the database is built from absolute pitch and timing information contained in a MIDI file. Within this file, notes are already grouped into channels by instrument, thus the prominent instruments can be easily identified.

To simplify matching further, a more general encoding is leveraged. Encodings that represent events with relative (versus absolute) pitch changes and time duration between onsets, known as the inter-onset interval (IOI), have empirically worked well for QBH applications. Many systems further generalize and represent timing information as the ratio between adjacent IOIs, known as the inter-onset interval ratio (IOIr). By taking the time ratio between notes, this system in effect has a dynamic time warping effect when comparisons are made. Furthermore, [6] was able to show that the log of the IOIr is possibly the most effective means of storing timing information.

Acknowledgements

This implementation used concepts and data (Tunebot data) from the Interactive Audio Lab at Northwestern University

References

[1] A. Spanias, T. Painter, V. Atti, Audio Signal Processing and Coding, ISBN: 9780471791478 and 0-471-79147-4, Wiley, March 2007.

[2] Ted Painter++ and Andreas S. Spanias, "Perceptual Coding of Digital Audio," *Proceedings of the IEEE*, pp. 451-513, Vol. 88, April 2000.

[3] J. Thiagarajan, A. Spanias, Analysis of the MPEG-1 Layer III (MP3) Algorithm Using MATLAB, Morgan & Claypool Publishers, Synth. Lect. on Algor. & Software, ISBN: 978-1608458011, Nov. 2011.

[4] Q. Shen+ and A. Spanias, "Adaptive Active Sound Reduction, *Noise Control Engineering Journal*, J44 (6), pp. 281-293, Nov. 1996.

[5] R. B. Dannenberg, W. P. Birmingham, G. Tzanetakis, C. Meek, N. Hu and B. Pardo, "The MUSART Testbed for Query-by-Humming Evaluation," Computer Music Journal, vol. 28, no. 2, pp. 34-48, 2004.

[6] B. Pardo and W. Birmingham, "Encoding Timing Information for Musical Query Matching," ISMIR, Paris, 2002.

[7] A. Huq, M. Cartwright, B. Pardo, "Crowdsourcing a Real-world On-line Query by Humming System," Proc. SMC, Barcelona, July 2010.

[8] M. Cartwright and B. Pardo, "Building a Music Search Database Using Human Computation," Proc. 9th SMC, Copenhagen, July 2012.

[9] Soundound.com

[10] T. Collins et al., "SIARCT-CFP: Improving Precision and the Discovery of Inexact Musical Patterns in Point-Set Representations," Proc. of ISMIR, pp. 549-554, 2013.

[11] D. Meredith, K. Lemström, and G. A. Wiggins, "Algorithms for Discovering Repeated Patterns in Multidimensional Representations of Polyphonic Music," JNMR, vol. 31, no. 4, pp 321-345, 2002.