QUERY BY HUMMING WITH THE VOCALSEARCH SYSTEM

Don't know the composer, performer, or title? Let the system match the theme you know to the song you want.

By William Birmingham, Roger Dannenberg, and Bryan Pardo

When one wishes to find a piece of music through Apple Computer's iTunes or at the local public library, the usual approach is to enter some textual information (metadata) about the piece (such as composer, performer, or title) into a search engine. However, when one knows the music, but not its metadata, standard search engines are not an option. One might instead hum or whistle a portion of the piece, providing a query for a search engine based on content (the melody) rather than on metadata. Systems able to find a song based on a sung, hummed, or whistled melody are called query by humming, or QBH, even though humming is not always the input.

Music fingerprinting is a related approach to finding a particular piece [2], where a portion of the sought-after recording is provided as the search example. Unlike QBH, music fingerprinting uses precise measurements of acoustic features of the original recording as the search key, or fingerprint. By design, music fingerprinting systems work only when the search query is exactly the same as the recording in the database. An alternate performance (such as a hummed melody) will stymie a fingerprint system, preventing it from making a match.

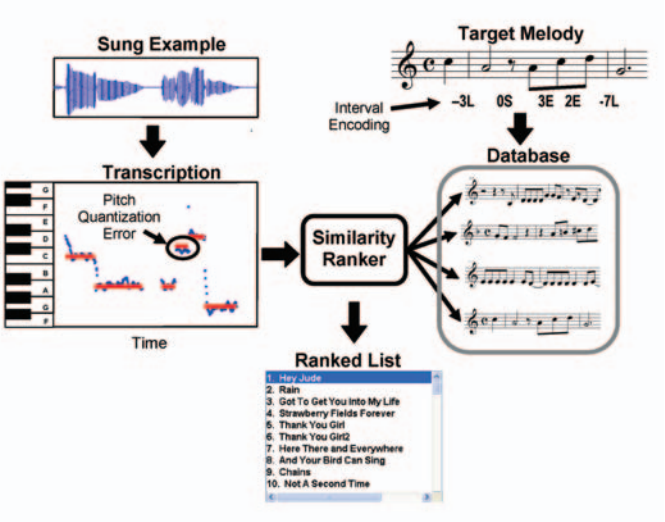
Music librarians attest to the usefulness of QBH, as they often take phone calls in which patrons hum a tune over the phone to try to identify its title. This musical query is natural for many people and requires no access to a copy of the desired recording at the time of the query. Moreover, QBH systems may provide more general results than fingerprinting systems, as a user may have a tune in mind but not a particular artist (such as "Retrieve both Phish's and ZZ Top's versions of 'Jesus Just Left Chicago'").

Here, we discuss QBH using the VocalSearch [9] system developed at the University of Michigan and Carnegie Mellon University (with support from a grant from the National Science Foundation) as a running example because it provides performance on par with the best current systems and involves a simple design and user interface (see the figure here). Queries are usually in the form of a user-supplied melody, theme, hook, instrumental riff, or some other memorable part of a piece [9]. Most QBH systems are therefore designed for music that is melodic. Music that is predominantly rhythmic or timbral is less amenable to QBH techniques, though some interesting recent work has addressed rhythmic recognition (www.songtapper.com) [4].

The typical QBH melody-comparison techniques—string alignment, n-grams, Markov models, dynamic time warping—compare monophonic (a single melody line) melodies in the database to a monophonic query. Since most Western music is harmonic or polyphonic (multiple concurrent voices), individual monophonic melodies must first be drawn from music to make it searchable. These melodies are not extracted directly from the audio, since audio with multiple concurrent instruments render standard pitch-tracking methods unreliable. This open problem may be avoided by using symbolically encoded music (such as musical instrument digital interface, or MIDI, protocol files).

A MIDI file contains instructions detailing which notes to play, the order of the notes, and the duration and volume of each of them. MIDI files are available for tens of thousands of pieces, thanks (in part) to the popularity of MIDI-based cell phone ringtones. Polyphonic MIDI files may be automatically converted to sets of monophonic files in various ways [8, 12]. The resulting files may then be used as search keys in a melodic database.

To increase precision and recall, QBH systems typically search over a database of melodic themes (such as the first four notes of Beethoven's "Fifth Symphony" or the verse of a song) rather than over a full piece. Restricting the database this way—single voice, representative themes—reduces the chance of a serendipitous match between the query and an obscure component of some unrelated piece of music (such as a descending scale in a query that matches a walking bass line). Having a reduced amount of data to search can speed operation by a factor of 100 or more. For example, each song in VocalSearch's Beatles database is represented by a theme for the verse and a different theme for the chorus. The figure outlines an example of these themes in the form of standard music notation.



How queries are processed in the VocalSearch QBH system.

Developing a system to identify themes is a special challenge. Themes can occur anywhere in a piece; they can be played by any instrument or voice; for example, Dvorak's "American Quartet" introduces the theme with a viola, an instrument rarely used to introduce thematic material. Themes can be repeated with variation, sometimes subtle, sometimes not. Some themes are simple and easily confused with the nonmelodic components of a piece. Historically, theme databases have been built by highly skilled musicians, a labor-intensive process that does not scale well. However, automatic thematic extraction methods have also been shown to be effective [7].

What Is That Noise?

Query quality is wide ranging, and even trained musicians do not necessarily present "better" queries to QBH systems [9]. Individuals may not recall the theme correctly, may mix themes, or mix voices (such as lead guitar and vocalist) into a composite query. Individuals also have production problems; they may be unable to sing the desired pitch and may laugh and cough, all while presenting the query. The computer "hears" all the unintended artifacts and considers them part of the query.

Transcription, or converting audio to musical notation, is another source of noise. Transcribers often introduce artifacts, drop notes, and botch rhythms. Thus, even a "perfect" query may not appear perfect after transcription. The transcription portion of the figure is an example of what the transcription of several well-sung notes looks like. The blue dots at the top of the figure represent a sequence of fundamental frequency estimates of the sung query; note the pitch glides and wavering pitch. The red lines represent the transcriber's note estimates (pitch and duration) quantized to the nearest piano key; note the quantization error on the fourth note, raising what should be a C to a C sharp.

Such errors, whether due to the poor pitch of an untrained singer or to a poor transcription must be included when matching the transcribed query to targets in the database. One approach to handling quantization error is to avoid it altogether by matching the pitch contour directly to database themes without quantization [5]. VocalSearch handles transcriber and singer error through prior estimation of error likelihood. It does this by training the system on a set of sung melodies of known pieces. The transcribed melodies are compared to the original, and the differences are recorded. Training builds a probabilistic model of the singer and transcriber error.

Best Guess

Many researchers have looked into melodic-comparison techniques [3, 6, 12], while others have looked into polyphonic comparison techniques [8, 11]. See [1] for a comparative analysis of the most popular melodic-similarity measures for QBH, including n-grams, dynamic time warping, Markov models, and string alignment. Many articles on the subject are also available in the International Conference on Music Information Retrieval research paper repository at www.ismir.net/all-papers.html.

> MANY RESEARCHERS HAVE reported results showing good performance in terms of processing time, precision, and recall.

VocalSearch [9] uses a probabilistic string-alignment algorithm to measure similarity between targets and queries. String-alignment systems determine how similar the query is to each of the targets based on edit distance; the "cost" (such as number of character changes) is what it takes to transform one string (the query transcription) into another string (the database theme). The theme with the lowest-cost transformation into the query is considered the best match.

In VocalSearch, these costs are based on the trained error model discussed earlier, which involves the type of errors singers make when singing. For example, it is very likely that when attempting to jump the interval of a minor 3rd, a singer will instead jump a major 3rd; it is very unlikely the singer will jump a major 7th. Thus, if we are trying to match a query to two themes, one with a minor 3rd and one with a major 7th, and the query presents a major 3rd, we would consider the first query to have the lower edit cost.

The melodic encoding used for queries and targets has a strong effect on the kinds of errors a QBH system can handle. For example, to get around the problem of key mismatch between query and target(s), QBH SYSTEMS CONSIDER tens of thousands of themes, a feat few humans without experience with music retrieval can match.

some researchers use intervallic distance (such as the number of half steps between successive notes) rather than absolute pitch. To account for differences in vocal range, representations of themes and queries generally discard much of the octave information and restrict melodies to a range of one or two octaves. VocalSearch uses intervallic distances and represents rhythm by the ratio of note durations. The figure outlines an example of a theme from the Beatles' "Hey Jude"; the number of half steps between adjacent pitches is encoded as an integer, while the duration of the second note, relative to the first, is encoded as *longer* (L), *shorter* (S), or *equal* (E).

Conclusion

Despite these technical hurdles, QBH systems work remarkably well. Many researchers have reported results showing good performance in terms of processing time, precision, and recall. For example, [10] reported results near human performance on a Beatles database. QBH systems have been used on a trial basis over the past few years in music libraries [6]. The third author regularly uses VocalSearch to access his own personal digital music collection. Meanwhile, technologically interesting applications for personal music libraries, music education, and karaoke systems are becoming possible. While humans are still better at recognizing sung queries, QBH systems continue to improve and can consider tens of thousands of themes, a feat few humans without experience with music retrieval can match.

References

1. Dannenberg, R., Birmingham, W., Pardo, B., Hu, N., Meek, C., and Tzanetakis, G. A comparative evaluation of search techniques for query by humming using the MUSART testbed. *Journal of the American Society for Information Science and Technology 58,* 3 (Feb. 2007).
2. Haitsma, J. and Kalker, T. A highly robust audio fingerprinting system. In *Proceedings of the Third International Conference on Music Information Retrieval* (Paris, Oct. 13–7, 2002); www.ismir.net.
3. Hewlett, W. and Selfridge-Field, E., Eds. Melodic similarity: Concepts, procedures, and applications. In *Computing in Musicology.* MIT Press, Cambridge, MA, 1998.
4. Kapur, A., Bening, M., and Tzanetakis, G. Query-by-beat-boxing: Music retrieval for the DJ. In *Proceedings of the Fifth International Conference on Music Information Retrieval* (Barcelona, Spain, Oct. 10–15, 2004); www.ismir.net.
5. Mazzoni, D. and Dannenberg, R. Melody matching directly from audio. In *Proceedings of the Second International Conference on Music Information Retrieval* (Bloomington, IN, Oct. 15–17, 2001); www.ismir.net.
6. McNab, R., Smith, L., Witten, I., Henderson, C., and Cunningham, S. Towards the digital music library: Tune retrieval from acoustic input. In *Proceedings of the First ACM International Conference on Digital Libraries* (Bethesda, MD, Mar. 20–23). ACM Press, New York, 1996, 11–18.
7. Meek, C. and Birmingham, W. Automatic thematic extractor. *Journal of Intelligent Information Systems 21,* 1 (2003), 9–33.
8. Pardo, B. and Sanghi, M. Polyphonic musical sequence alignment for database search. In *Proceedings of the Sixth International Conference on Music Information Retrieval* (London, Sept. 11–15, 2005); www.ismir.net.
9. Pardo, B., Shifrin, J., and Birmingham, W. Name that tune: A pilot study in finding a melody from a sung query. *Journal of the American Society for Information Science and Technology 55,* 4 (Feb. 2004), 283–300.
10. Pardo, B. and Birmingham, W. Query by humming: How good can it get? In *Proceedings of the Workshop on Music Information Retrieval* (Toronto, 2003).
11. Pickens, J., Bello, J., Monti, G., Sandler, M., Crawford, T., Covey, M., and Byrd, D. Polyphonic score retrieval using polyphonic audio queries: A harmonic modeling approach. *Journal of New Music Research 31,* 1 (June 2003), 223–236.
12. Uitdenbogerd, A. and Zobel, J. Melodic matching techniques for large music databases. In *Proceedings of the Seventh ACM International Conference on Multimedia* (Orlando, FL, Oct. 30–Nov. 5). ACM Press, New York, 1999, 57–66.

William Birmingham (WPBirmingham@gcc.edu) is the chairman of the Computer Science Department and a professor of computer science and electrical engineering at Grove City College, Grove City, PA.
Roger B. Dannenberg (roger.dannenberg@cs.cmu.edu) is an associate research professor of computer science and art at Carnegie Mellon University, Pittsburgh, PA.
Bryan Pardo (pardo@cs.northwestern.edu) is an assistant professor in the Department of Computer Science with a courtesy appointment in the School of Music at Northwestern University, Evanston, IL.