in *Proceedings of ACM Multimedia Workshop on Audio and Music Computing for Multimedia (AMCMM 2006)*, Santa Barbara, CA, USA, October 23-27, 2006

Karaoke Callout: Using Social and Collaborative Cell Phone Networking for New Entertainment Modalities and Data Collection

David A. Shamma[1]
Yahoo! Research Berkeley
1950 University Ave, Suite 200
Berkeley, CA 94704
+1 (510) 704-2419
shamma@yahoo-inc.com

Bryan Pardo
EECS, Northwestern University
Ford Building, Room 3-323, 2133 Sheridan Rd.
Evanston, IL, 60208, USA
+1 (847) 491-7184
pardo@northwestern.edu

ABSTRACT
We have developed a user-trainable query by humming (QBH) system that develops an error probability model of a user's singing. While the training is effective, it is also tedious and time consuming, requiring the user to sing dozens of melodies to the system before the system can be trained. To make training fun, we introduce a new interactive, distributed karaoke game, called *Karaoke Callout*, played over a cell phone. The user selects a song and sings it into the cell phone. The audio is sent to a server which rates the quality of the singing by measuring how closely it resembles a canonical example of the song stored in the server database, sending a score back to the user. The user may then challenge anyone in the phone's contact list. An SMS text challenge is sent to the challenged person's cell phone. The challenged person sings the song, attempting to better the performance of the challenger. This challenge may then be repeated, with either party selecting a new song with which to "call out" the other party. Over the course of an interaction, numerous examples of each party's singing are created and stored. These may then be used to train a QBH to the idiosyncrasies of each user's singing, as well as providing new query targets for the system.

Categories and Subject Descriptors
H.5.5 [Sound and Music Computing] *Methodologies and techniques. Signal analysis, synthesis, and processing.*

General Terms
Measurement, Human Factors

Keywords
Karaoke, Query by Humming, Entertainment, Music, Audio.

1. INTRODUCTION
Most currently deployed music search engines, such as Amazon.com and local libraries, make use of metadata about the song title and performer name in their indexing mechanism. Often, a person wishing to find a recording is able to sing a portion of the piece, but cannot specify the title, composer or performer. Query by humming (QBH) systems solve this mismatch between database keys and user knowledge by creating a content-addressable music database. QBH systems transcribe a sung or hummed query and search for related musical themes in a database, returning the most similar themes as a play list. Melody matching [1-4] and query by humming [5-8] been investigated by many research groups in recent years and has even found its way into a prototype commercial song-finding application (http://www.sloud.com).

In our work, we have developed a user-trainable query-by-humming system called VocalSearch [9]. VocalSearch develops an error probability distribution for a user's singing, to improve search results. To learn this distribution, a sequence of melodies is played to the user. The user sings each melody back to the system. The sung melody is compared to the original and the difference between the two is recorded. Repeating this process builds a probabilistic model of the combined error of the singer and transcription system. The learned error model is then used to find the most probable sequence transformation from the query to each database element. The sequence with the most likely transformation into the query is deemed the correct target melody and returned as the answer.

While the system is effective, training is tedious and time consuming, requiring the user to sing dozens of melodies to the system before use, limiting the effectiveness of the approach. In order to deal with this weakness, we have taken a cue from a tagging approach used for photo collections [10] and re-cast system training in the form of a new interactive, client-server karaoke game: *Karaoke Callout.*

Karaoke Callout is played over a cell phone. The user selects a song and sings it into the phone. The audio is sent to a server which rates the quality of the singing by measuring how closely it resembles a canonical example of the song stored in the server database, sending a score back to the user. The user may then

[1] Work was conducted while author was at Northwestern University.

challenge anyone in the phone's contact list. An SMS text challenge is sent to the challenged person's cell phone. The challenged person sings the song, attempting to better the performance of the challenger. This challenge may then be repeated, with either party selecting a new song with which to "call out" the other party. Over the course of an interaction, numerous examples of each party's singing are created and stored. These may then be used to train a QBH system to the idiosyncrasies of each user's singing, as well as providing new query targets for the system.

2. RELATED WORK

One primary inspiration for Karaoke Callout is The ESP Game, a web-based image labeling game developed at Carnegie Mellon University [10]. The game is played by two randomly assigned partners that log onto a website. The partners are presented with a series of images. The goal for each player is to guess a label their partner would type in for the current image. Each player is allowed multiple guesses, but is not allowed to see what the partner is guessing. Once their guesses agree on a label, they move on to the next image. Scores are based on the number of images the players can process in a fixed time. The more labels they agree on, the higher the score. This encourages players to quickly converge on canonical image tags. As the game goes on, the system stores these image tags and the ESP Game uses this online collaboration as verification of indexing and retrieval tags for an image.

The ESP Game allowed researchers to recast the tedious task of labeling a large database of images with tags as entertainment. Placing it online let researchers label 293,760 images over a period of four months. This was achieved by over 13,000 people playing the game. A number of these people played for over 50 hours each. This work inspired us to develop a game that similarly motivates our users.

Both our approach and the ESP game also take inspiration from the success of the Open Mind Common Sense initiative [11]. Here, a web site prompts site visitors to enter a piece of common sense knowledge (like "Hot stoves can burn you.") into a text box. These "facts" are compiled automatically into a common sense reasoning knowledge base (KB). This simple model, encouraging contribution from the general public, has let the Open Mind Common Sense unverified knowledge collection grow to be the second largest available KB. This contrasts with the estimated 600 person-years of dedicated, paid researcher time devoted to creating the largest KB currently available, Cyc [12].

With respect to automated singing interaction systems, there has been much work into karaoke machines. The most related work to Karaoke Callout is "Karaoke Revolution," a video game developed by Harmonix and released by Konami in 2003 for the Sony PlayStation 2 platform (later released for the Microsoft Xbox and Nintendo Game Cube, as well). As a single player game, Karaoke Revolution lets players select a song to sing. During the game, players see how sharp or flat they may be singing as well as how long to hold each note using a player piano type rolling indicator: allowing players to modulate their voice to fit the exact recording. Points are gained for exactly matching the player piano roll. A multiplayer game allows a small group of players to sing three songs. Each player selects which song they wish to sing before the game begins. The player with the highest point total by the end of three songs wins the challenge.

3. THE GAME EXPERIENCE

Karaoke Callout is a distributed karaoke challenge game, played over the cell phone. The game begins when a player launches the app from a cell phone and start to sing or hum a 10 second excerpt of a song in the database. Currently, the database of known songs contains The Beatles' collection. Once they have sung their part, it is scored by our QBH backend. A ranked list of songs is returned and the player selects which song was sung. Once the player identifies the song, the system returns the singing score.

Score in hand, the player can then opt to 'challenge a friend' or 'sing again'. If the player chooses to challenge a friend, he or she is prompted to specify which friend they wish to challenge. The selection is done using the player's cell phone phonebook or contact list. Once the player to challenge has been selected, a SMS text message is sent to the new opponent. This message has a short message describing what Karaoke Callout is, where they can download and install the client, and a challenge ID.

The opponent runs Karaoke Callout and selects the 'Accept challenge' option. They are then prompted with the name of the song to sing. The player accepts the song challenge and sings the song. This new audio is used, once again, as a query which is ranked and scored. The opponent then gets to see his or her score and a dialog box displays congratulations if they won the callout. The outcome of this challenge is sent (win or loss) via SMS to the challenger. The nature of this game play is not time-sensitive (currently challenges do not expire) and players can have multiple challenges in operation at any single point in time.

4. THE SYSTEM

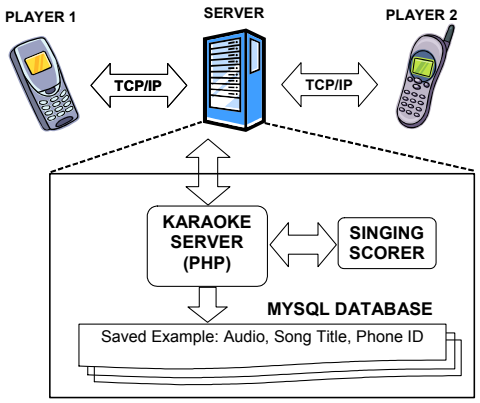

Figure 1. System Overview

A QBH system requires hardware well beyond the capacity of most hand held devices available today. Many popular cell phones currently lack even a simple floating point unit. Thus, we decided to implement Karaoke Callout as a TCP/IP client—server (or client—server—client) application. This allows for a greater reach from several platforms and modalities, letting us build Karaoke Callout clients for anything from a cell phone to a personal computer with a Java-enabled web browser.

This architecture also lets us isolate the client cell phone from the mechanics of persistence and search by using it simply as a means

to record the audio and as a user interface, with all computation taking place on the server, connecting to the client via a simple PHP web service.

The system (see Figure 1) is divided into three main components. The first of these is the Karaoke Server (written in PHP), which handles communication with the clients, queries the Singing Scorer and stores sung examples in the database. The Singing Scorer is a QBH system that returns a similarity score between a player's sung example and the canonical version of the song. The Singing Scorer is modular and separate from the Karaoke Server, allowing each component to be updated at its own rate. The final component is a MySQL database to keep track of audio queries, scores, and challenges.

4.1 Cell Phone Client and Karaoke Server

With the game itself isolated from the backend, we only needed an interface to facilitate the prototype and deployment of the cell phone game application. The core functionally the game requires is the ability to perform simple web operations as well as the ability to record PCM audio and send text messages. Given our cell phones are connecting TCP/IP over cellular GPRS, we wanted to reduce the amount of transactions from the phone to the DB and Singing Scorer.

We implemented Karaoke Callout in PyS60 (Python for Nokia Series 60 cell phones). PyS60 provides an excellent sandbox API to Nokia s60 cell phones. Queries can be easily recorded, compressed using gzip and sent via HTTP POST to our PHP service. The PHP service takes care of manipulating the audio query (uncompressing it and converting it to the desired bit rate if necessary). The PHP service then returns a ranked list in XML format. The XML results are displayed and challenges are sent from cell phone to cell phone via Short Message Service (SMS or text messaging).

4.2 Singing Scorer

The Singing Scorer generates a score for each sung example by determining the cost (in terms of edit operations) of transforming the sung example into the canonical version from our database. The fewer edits required for this transformation, the more points the singer earns.

Since the initial motivation for Karaoke Callout was the generation of training data for our query by humming system, we use the core technology of our QBH system [9] to generate scores for those playing Karaoke Callout. Figure 2 shows the functional breakdown of the Singing Scorer.

In the first step, the recording of the sung query is converted into a series of fundamental frequency estimates by an enhanced autocorrelation algorithm [13]. This is shown in the "transcription" portion of Figure 2. Blue dots indicate raw pitch estimates taken 100 times per second. Horizontal bars indicate estimated notes, quantized to the nearest pitch on an equal tempered piano tuned to A4 = 440 Hz.

The singing example is then encoded as a sequence of note transitions (pitch intervals between adjacent notes) whose durations are encoded as inter onset intervals (IOI). The durational ratios are evenly spaced in the log of the IOI ratios between notes, as research in music perception [14] indicates IOI ratios fall naturally into evenly spaced bins in the log domain.

Once transcribed, the sung example is compared to a canonical melody from the database. This comparison is done with a probabilistic local string alignment algorithm that finds the lowest cost transformation of A into B in terms of operations (insertion or deletion of characters). We use a method based on the dynamic-programming implementation of local alignment introduced by Gotoh, as described in Durbin, Eddy et al. [15]. This finds an optimal global alignment between two sequences, A and B, taking into account how likely it is that any given element in sequence A is related to sequence B.

In our system, the likelihood of association between an element of the canonical melody and an element of the sung example uses a model of singer error resulting from system training. The idea here is that singers are prone to predictable systematic error, such as an inability to reproduce rhythm accurately, or to sing flat. Such errors can be handled gracefully if an error probability distribution is maintained. To learn this distribution, a melody is played to the user. The user sings the melody back to the system and the sung melody is transcribed by the system. The transcribed melody is compared to the original and the difference between the two is recorded. Repeating this process builds a probabilistic model of the combined error of the singer and transcription system.

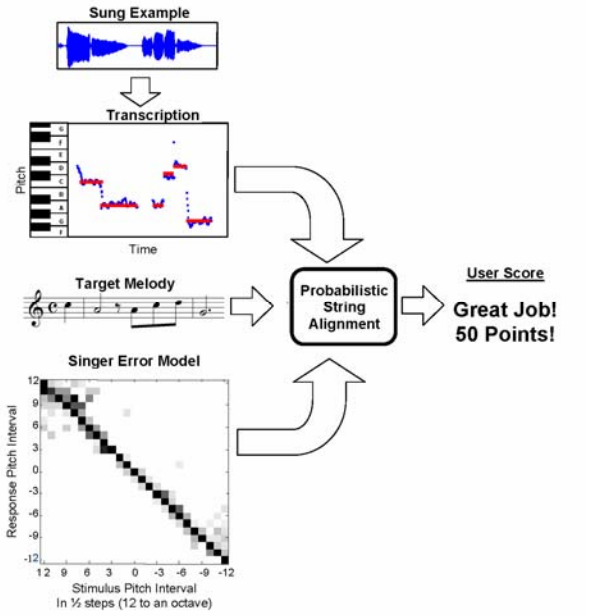

Figure 2. The Singing Scorer

Figure 2 shows the error model for a single singer's transcribed pitch reproduction as a confusion matrix. The horizontal axis represents the stimulus pitch-interval presented to the singer. The vertical axis represents the response generated by the combination of singer and pitch tracker. Each square represents the frequency with which a particular stimulus-response pair was observed. The darker the square, the more frequent the occurrence.

Of course, Karaoke Callout is played before the singing error model for an individual has been generated. In fact, one of the points for the game (from our perspective) is to generate numbers paired examples of canonical melodies and sung versions so that this error model may be generated. For the purpose of the game, a

general error model is used, based on the average error models of an initial group of experimental subjects.

4.3 Learning from the Data

Each time a player sings to Karaoke Callout, a database entry is made, containing information about the player's telephone ID, the correct title of the song, and the actual sung example, itself. This data gives us maximal flexibility to improve the associated QBH search engine, both on an individual basis and for users in general.

Recall that each sung example is labeled by the Karaoke Callout player. A database of labeled singing examples let us improve QBH performance in three ways: First, our QBH system depends on note segmentation of singing in order to create a sequence of notes used as a query. Since users select a song to play Karaoke Callout, these matched pairs of query and song title can be used to optimize our note segmentation system to increase the ranking of the correct label. Second, for each participant, we can create a singer-specific error model, allowing improved performance on matching to canonical examples. Finally, we are able to use the transcribed queries as alternate targets for each song. Thus, rather than having a single canonical example of a popular song ("Happy Birthday," for example), every attempt at singing the song in the course of a Karaoke Challenge can be transcribed and used as a potential target for the QBH system to use as a match for future queries.

5. FUTURE WORK

Having completed initial trials with the system, we plan to make the Karaoke Callout client software available as a free download at http://www.karaokecallout.net. While the current client works on any phone using the Symbian S60 operating system, we plan to make a new client for Java2 Mobile Edition (J2ME) enabled devices to reach a greater audience. The thin client architecture will remain the same. Complications introduced by new devices can be handled by the server side PHP layer.

We have begun storing the cellular location identifiers for every sung example. We do not use global positioning technology (GPS), a feature yet to appear in mass scale in cellular devices. Instead, we use the logical location specified by the cell phone's active tower connection, a technique that has be used for social photo spaces [16]. We hope to motivate players by creating a karaoke challenge based on location. A player could thus challenge anyone in their local town who has the Karaoke Callout client installed and ranking could be kept on a regional basis. A player in Berkeley, California, for example, will be able to see the high scores and most popular songs in Berkeley, as well as have the ability to challenge those songs and take ownership of the karaoke space within that neighborhood.

6. REFERENCES

[1] Hewlett, W.B. and E. Selfridge-Field, eds. Melodic Similarity: Concepts, Procedures, and Applications. Computing in Musicology. Vol. 11. 1998, MIT Press: Cambridge, MA.

[2] Hu, N., R. Dannenberg, and A. Lewis. A Probabilistic Model of Melodic Similarity. in International Computer Music Conference (ICMC). 2002. Goteborg, Sweden: The International Computer Music Association.

[3] McNab, R.J., et al., The New Zealand Digital Library MELody inDEX. D-Lib Magazine, 1997. May.

[4] Uitdenbogerd, A. and J. Zobel. Melodic Matching Techniques for Large Music Databases. Seventh ACM International Conference on Multimedia. 1999. Orlando, FL.

[5] Dannenberg, R., et al., The MUSART Testbed for Query-By-Humming Evaluation. Computer Music Journal, 2004. 28(2): p. 34-48.

[6] Meek, C. and W. Birmingham, A Comprehensive Trainable Error model for sung music queries. Journal of Artificial Intelligence Research, 2004. 22: p. 57-91.

[7] Pauws, S. CubyHum: A Fully Operational Query by Humming System. in ISMIR 2002. 2002. Paris, France.

[8] Unal, E., et al. Creating Data Resources for Designing User-centric Front-ends for Query by Humming Systems. in Multimedia Information Retrieval. 2003.

[9] Pardo B., et al., Name that Tune: A Pilot Study in Finding a Melody from a Sung Query. Journal of the American Society for Information Science and Technology, 2004. 55(4): p. 283-300

[10] von Ahn, L. and L. Dabbish. Labeling Images with a Computer Game. in CHI2004. 2004. Vienna, Austria.

[11] Singh, P., The public acquisition of commonsense knowledge, in.Proceedings of AAAI Spring Symposium on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access. 2002, American Association of Artificial Intelligence: .Palo Alto, CA.

[12] Anthes, G., Computerizing Common Sense, in Computerworld. April 2002. p. 49.

[13] Boersma, P. Accurate Short-Term Analysis of the Fundamental Frequency and the harmonics-to-noise ratio of a sampled sound. in Institute of Phonetic Science, Proceedings. 1993. University of Amsterdam: IFA Proceedings 17.

[14] Hutchinson, W. and L. Knopoff, The Clustering of Temporal Elements in Melody. Music Perception, 1987. 4(3): p. 281-303.

[15] Durbin, R., et al., Biological Sequence Analysis, Probabilistic models of proteins and nucleic acids. 1998, Cambridge, U.K.: Cambridge University Press.

[16] http://zonetag.research.yahoo.com/