LOUD: An Immersive Music Exploration System

Joseph Polastre, Christina Heyl, Mehrad Noori
Soda Hall, Department of Computer Science
University of California, Berkeley
Berkeley, CA 94710 USA
polastre@cs.berkeley.edu, cheyl@sims.berkeley.edu, msnoori@uclink.berkeley.edu

ABSTRACT
We propose a system, called LOUD, for immersive, multi-dimensional music exploration based on music similarity algorithms. LOUD permits non-linear browsing, searching, and navigation of songs, genres, and musical properties using a physical interface. The system follows a navigation metaphor relating the physical controls to a map of musical styles.

Keywords
Music exploration, music similarity, immersive design, navigation, exploration.

INTRODUCTION
Many researchers have worked on the problem: "What characteristics make two pieces of music similar?" The motivation for music similarity research is clear: business uses and advertisement, identifying broadcasted music for copyright enforcement, and simply finding new music by features of other music. Our goal is not to revisit the music similarity problem, but instead to enable ordinary users to benefit from this research and easily find new music by adding an intuitive physical interface.

We are specifically interested in how people are exposed to new music. A short list includes concerts, friends, the radio, movies, television, online file sharing, listening stations at record stores, recommender or collaborative filtering systems like amazon.com, and software like SoundFisher [1] or Marsyas [3]. The major flaw with all of these methods is they are either a very linear focused search, or depend entirely upon chance. Many of these systems also require an advanced knowledge of sound properties that most people do not possess. These systems use cumbersome and confusing interfaces, or contain extremely limited selections of music. Instead, we wanted to enable users to explore music in a multi-channel environment with a variety of options at each point and the ability to reliably browse the music dataset non-linearly without worrying about the underlying technology.

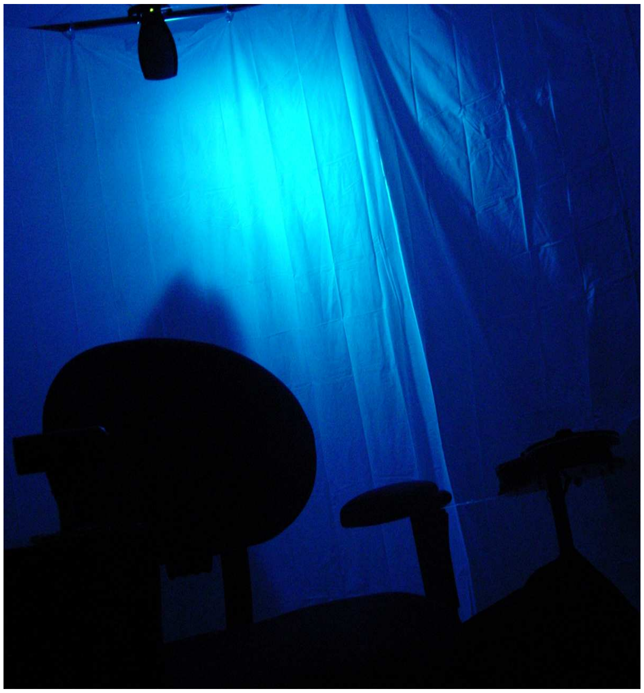

Figure 1: The LOUD music exploration system.

DESIGN
In order to entice users to find new music, we wanted to create a fun environment that would immerse the user. Immersion allows the user to concentrate on the task of enjoying music. Since the user is free of distractions, we added the idea of making the immersion three-dimensional.

The user sits in an enclosed booth where they are able to "navigate" through 360 degrees of music using controls mounted to the chair they are sitting in. As the user turns the chair around, four sets of speakers, each playing a different song, fade in and fade out accordingly so that the speakers the user is facing are the loudest. The user can then choose any of the four songs as a point of reference for choosing subsequent songs. For example, the user can specify the next song to be faster or slower and from an earlier or later time period than the current song they are listening to. To allow for these selections and to provide the user with a familiar metaphor that would allow for intuitive use, we designed a navigation metaphor that is based on a car-like instrument cluster (implemented in

Figure 4). There is also a "genre shifter" mounted to the right hand armrest which allows the user to specify the next set of songs to be more like or less like the current song based on genre. The user recognizes the gauges and the shifter as tools of indirect motion and is thus able to easily dial in their selections to maneuver through the virtual map and find their ideal track.

The concept is an exhibit-like personal music exploration shown in Figure 2. Users navigate through a virtual map of music to find new songs that they may be interested in. The main point of control is a chair that swivels 360 degrees, surrounded by four sets of speakers set at 90 degree intervals. The user manipulates a dashboard where they can "dial-in" settings. One of these settings is a similarity control that specifies whether the next song should be "more like this" or "less like this." Other settings include beats per minute (speed), vocals (instrumental, male, female, both), and time (when the song was first composed or published). These settings are analyzed relative to the song currently playing directly in front of the user. The system provides a new selection of songs and starts them playing around the user. The user can then select a new point of reference, and dial in new settings based on this current selection. This process continues indefinitely.

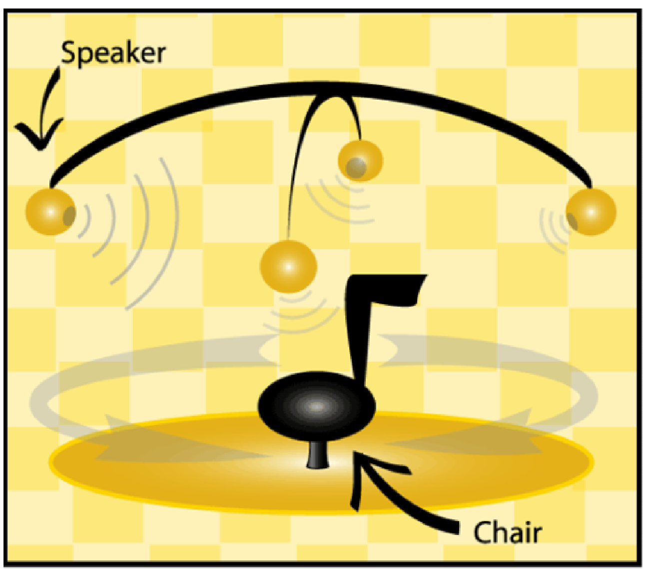

Figure 2: An idealized view of the LOUD design.

IMPLEMENTATION

The implementation consists of a modular expandable approach. All communication in LOUD occurs through TCP/IP sockets, permitting various components to excute on different networked machines. Adding a new set of speakers to the system, for example, simply consists of starting another instance of Winamp and sending it commands through sockets. The center of control is the the command logic which communicates with the Winamps, SQL database, input devices, and output devices. A diagram of our implemenation is depicted in Figure 3.

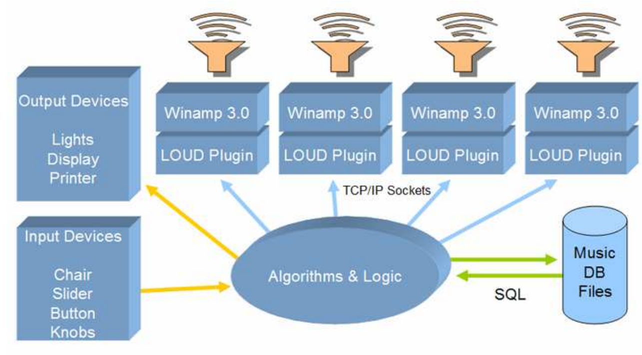

Figure 3: System Implementation of LOUD

Physical Space

The physical space is a booth comprised of eight speakers hanging with opaque shower curtains from a six-foot wide copper-pipe octagaon. The user walks into the booth where a rotating chair sits in the center with dashboard and genre shifter mounted on it (Figure 4).

The space is dark with blue flood lights creating ambiance. We chose a single color with flood (uniform) light distribution to pacify the user and evoke a feeling of tranquility (Figure 1).

Figure 4: LOUD's physical space design

Dials, Gauges, and Shifter

The core shape of the dashboard, which contains the dials and gauges, was created out of 3/8'' thick transparent Lexan. As shown in Figure 4, the dashboard guages are printed and glued to the bottom surface of the Lexan. Underneath the top sheet of Lexan are 50kΩ potentiometers that monitor the position of the dials. Each potentiometer also has an SPST switch. The switch turns on and off red LEDs that illuminate the underside of the dial if it is on. If the dial is off, the LED is off, and its functionality is not visible to the user in the dark space.

On the other side of the chair is a shifter from the gaming industry. It is a robust lever with a button to trigger user input. The shifter is implemented with a 5kΩ potentiometer.

Each potentiometer (4 in total) is connected to a wireless microcontroller (referred to as a UC Berkeley *mote* [2]). On a button press, the mote reads the values of each of the potentiometers at its ADC channels and transmits the values wirelessly to the LOUD control application. Wireless transmission was chosen since the dashboard is mounted on a surface that may be rotated around continuously.

Figure 5: Dashboard Controls used in LOUD

Shaft Encoder

In order to provide the control logic information about the direction of the user, we needed to use a shaft encoder on the swivel chair. The shaft encoder must have a minimum resolution of 2 bits to provide 4 directions. Additional resolution is useful for enhancing the user experience. With more directional information, music can fade as the user turns from one direction to the next.

The initial implementation of the shaft encoder involved a series of pins mounted parallel to the shaft of the chair (see Figure 6, left). The shaft, made of metal, acts as a switch when contact is made with the pins. By wrapping the shaft with electrical tape, a gray code (or other binary code) may be applied to the shaft and the pins used as the encoder. This design had some serious flaws; the most severe problem was the lack of constant contact between the pins and the shaft.

Our next iteration used phototransistors sensitive in the red-light wavelength (see Figure 6, right). Equipped with red LED emitters, the photodiode detects the reflectivity of paper—black (or printed) paper has less than 90% reflectivity and results in a logic one. White paper with greater than 90% reflectivity results in a logic zero. The encoder consisting of four phototransistors were mounted around the shaft.

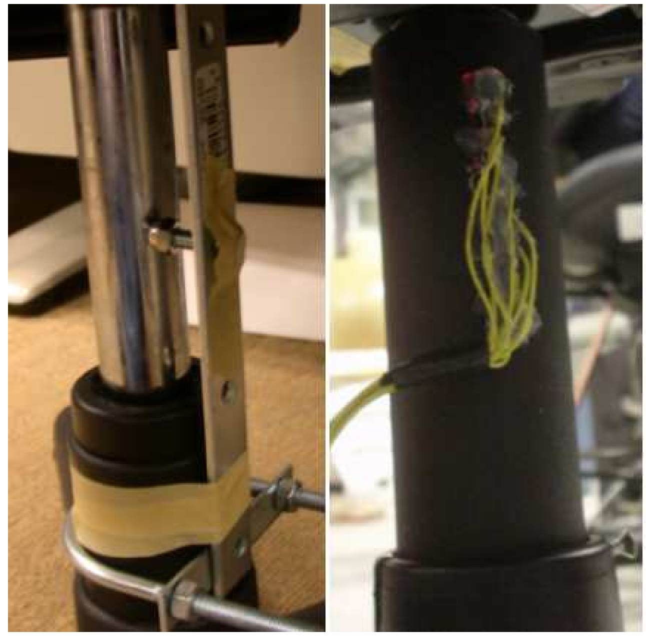

Figure 6: Mechanical shaft encoder (left) and optical shaft encoder (right)

Music Classification

In determining which parameters we would let the users manipulate, we chose to utilize beats per minute (BPM), vocal quality, year, and genre. To obtain basic data about each track, we extracted the information from the CDDB (Compact Disc Database), an online database. Unfortunately, while a field existed for BPM information in the CDDB, most records did not contain this data. However, we felt this was valuable search criteria, so we used a third party BPM extractor and incorporated this data as well. Year was extracted directly from CDDB, while vocal quality was classified based on the artist.

We also attempted to extract genre information from CDDB, but it proved to be inaccurate, inconsistent, or missing. Genre proved to be the most difficult quality to classify, as it is almost entirely subjective. Current research attempts to automate genre classification by statistical modeling [4], but at this time, no tools currently exist to facilitate this. In the absence of an objective classification tool, we manually classified the selection of music into 21 distinct genres.

In determining musical similarity, we constructed a musical graph, loosely based on the "Music Maps" and "Musical Styles" definitions at www.allmusic.com, using a numerical scale to indicate distance between genres. We were then able to represent the similarity finding problem as simply finding the nearest neighbors to the current song.

Database System

Data about each track is stored in a database implemented in MySQL, an open-source database server running on a local machine. Once the query is formed by the control application, a connection is opened to the database. The query is then sent, processed and returned using the MySQL++ API. Queries which return no results are repeated using gradually expanding parameters until a result set of the appropriate size is reached, at which time the connection is closed.

Software, Logic, and Winamp

Each component communicates with the LOUD control application and logic through TCP/IP sockets. The control application initially connects to the serial and parallel ports through TCP/IP forwarders that read input from the dashboard and shaft encoder. When the button on the shifter is pressed, the state of each potentiometer is recorded and sent wirelessly, then read via the serial port. The control application receives this data through the socket, sets its state, creates an SQL query statement, and queries the database. Four of the results from the query are randomly chosen and sent to the four instances of Winamp, each one controlling a different set of speakers.

The Winamp control consists of a Winamp plug-in using the Wasabi SDK. The plug-in monitors the state of Winamp. Upon receiving a command through a socket, the LOUD plug-in controls the state of Winamp. Through sockets, the LOUD control application can alter the state of all four Winamp instances where the state may consist of: change volume, play, pause, stop, previous, next, and add to playlist.

USER REVIEWS

Initial response to the LOUD system was extremely positive. Users commented on the usefulness of the system. Many expressed that they found the LOUD system entertaining and fun to use.

User testing also alerted us to a variety of areas that LOUD could use improvement, the first of which is response time. When the button is pressed on the shifter, it may take up to 3 or 4 seconds for the request to be processed and new music to start playing. Either that process needs to be sped up or the user needs to be alerted that their request has been received.

We found that with repeated use, the reliability of the shaft encoder decreased. Future iterations would benefit from an off-the-shelf industrial quality shaft encoder.

CONCLUSION

While the LOUD project was well received, it is clear that it would benefit from further iteration. Greater durability of the components, particularly the potentiometers and shaft encoder, would highly improve the performance of the interface. In addition, when research in the field becomes more complete, automatic genre classification would greatly assist the accuracy and completeness of the searching algorithms. User studies to evaluate the ease of use of the physical interface would be a valuable additional step. Obvious areas of expansion are to include a greater range of search parameters such as loudness or brightness, and to populate the database with a larger selection of possible tracks. We consider the proof-of-concept implementation of LOUD to be a success with its users and a promising method of exploring music.

REFERENCES

1. Thomas L. Blum, Douglas F. Keislar, James A. Wheaton, and Erling H. Wold. “Method and article of manufacture for content-based analysis, storage, retrieval, and segmentation of audio information.” United States Patent 5,918,223. 1999.
2. Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, Kristofer Pister. “System architecture directions for network sensors.” In the Proceedings of *ASPLOS*, 2000.
3. George Tzanetakis and Perry Cook. "MARSYAS: A framework for audio analysis." Organised Sound 4(3), Cambridge University Press, 2000.
4. George Tzanetakis, Georg Essl, and Perry Cook. Automatic musical genre classification of audio signals. In Proc. Int. Symposium on Music Inform. Retriev. (ISMIR), pages 205--210, Bloomington, IN, USA, October 2001.