A NOVEL INTERFACE FOR AUDIO SEARCH

Sarah Ali and Parham Aarabi

University of Toronto
Department of Electrical and Computer Engineering
10 Kings College Road, Toronto, Ontario, Canada

ABSTRACT

In this paper a novel cyclic interface for searching through a song database is proposed. The method, which merges multiple audio streams on a server and broadcasts only a single merged stream, allows the user to hear different parts of each audio stream by cycling through all available streams. Experimental results on 21 users illustrate that the proposed interface requires less listening time as compared to traditional list-based interfaces when the desired song/audio clip is among one of the audio streams. The average search time for the proposed interface was 7.3 seconds, compared to 12.1 seconds for the traditional list-based interface when searching for a song which is included among the audio streams.

1. INTRODUCTION

Searching for an audio clip or a song is inherently different than searching for a website or an image. One key difference is the fact that the result of the search, for images or websites, is readily displayed in brief/thumbnail form without requiring extensive bandwidth. In other words, Google and Yahoo can display the top 20 matches to your search criteria almost immediately, for the case of website search or image search.

However, what is the thumbnail method for displaying song or audio clip search results? Do you simply select a random part of the clip and play it back? If so, how can you ensure that this random part contains enough information for the user (i.e. the person doing the search) to realize whether this is the song that they wanted or not?

With audio, it is extremely difficult to select a small portion that is fully representative of the entire song. As the segment size is increased, it of course becomes easier to identify the song. For example, as an extreme case, if the entire song is played back in its entirety it should be fully possible to identify the song. However, playing a large portion of the song requires bandwidth as well as listening time (i.e. bandwidth to download the portion of the song, time for the user to actually listen to it). As our network connections continue to improve, the bandwidth required to download the song becomes insignificant. However, the listening time which is a function of the human perception system is biologically limited. Hence, no matter how fast your internet connection, the process of searching for a song which yields a possible X matches (each being of duration T seconds) will require a total bandwidth as well as a listening time that is proportional to XT. In other words, the search engine must perform extremely well, otherwise a user will have to spend a long time sequentially evaluating the songs in order to determine the correct result.

If an audio file could be successfully summarized into only the most recognizable part of the song, then search times could be reduced. [1] presents a method for locating the key phrase within a song, which is the portion of the song which repeats most frequently. The approach in [2] tries to locate the chorus of a song based on the cyclic properties of the music. However, the main problem with most summarization techniques is that the part of the song that occurs most frequently may not be most recognizable by the user. For example, the summary may be an instrumental piece or not contain the song title. Furthermore, summarization techniques are only well suited for summarizing certain genres of music. Another approach to create a more efficient search is to create an intuitive visual component for a search interface. The majority of work in multimedia visualization is limited to the effective presentation of video media, where any focus on audio is limited to speech audio only [3, 4]. In [4] effective search of speech audio is achieved by speeding up audio playback in order to decrease search times. [5] presents a system for classifying speech-based audio based on speaker identity and speech content, which relies completely on text for visualization of search results. Initiatives with the goal of making the music search process more efficient focus on visualizations which convey the acoustical properties of music such as pitch, tempo and musical structure [6, 7, 8, 9, 10, 11]. [11] presents a methodology for visualizing databases of music media which is based on the psychoacoustic properties of the music content. Visualization is in the form of grouping media with similar psychoacoustic properties close to each other in the visual space to form clusters. These types of visualizations are suited for speeding up search times for users who are looking for content with certain acoustical properties, or who are interested in discovering new content belonging to a certain genre, rather than for searching for a specific song or piece of music. For internet search engines, however, the

primary goal is to find a specific song which is desired by the user.

In this paper, we propose an acoustic interface methodology which when coupled with an acoustic search engine, would only require a total bandwidth as well as listening time that are proportional to T (i.e. are independent of the number of songs in the results, with certain limitations).

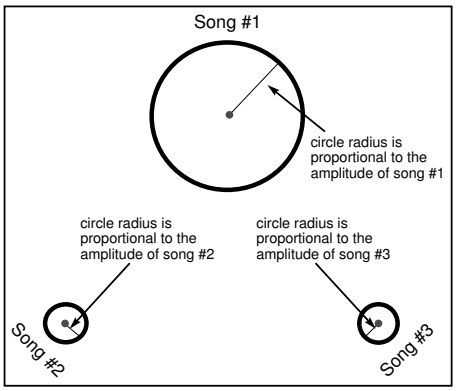
2. METHODOLOGY

The audiovisual interface proposed in this paper allows a user to simultaneously evaluate multiple songs. Consider a search engine which returns X search results. The goal is to combine all (or a subset) of the search results into a single audio file in a manner in which each individual song can be identified. This is accomplished by weighting the song corresponding to the i^{th} search result by $R_i^2(t)$, where

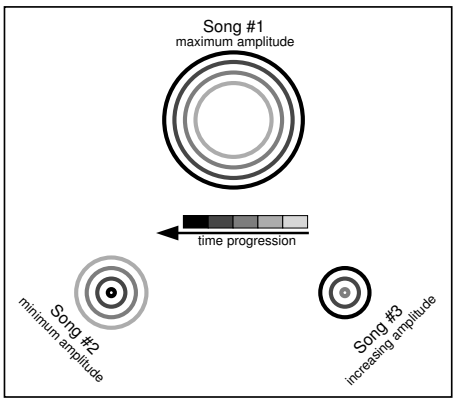
$$R_i(t) = \sin(\omega t + \theta_i) + 1, 0 \leq t \leq T, \quad i = 1, 2...X \quad (1)$$

The phase θ_i of each sinusoid is chosen so that the sinusoid for search result i is out of phase with the sinusoids corresponding to other search results. Specifically, for the i^{th} result, $\theta_i = \frac{2i\pi}{X}$. In other words, θ_i is chosen so that the peak volume of each song occurs at a different time. The frequency ω is chosen such that a listener may distinguish between multiple songs. If ω is too large, it becomes difficult to distinguish between multiple songs. On the other hand, setting ω too low comes at the cost of an efficient search. ω is constant for all search results. The weighted song waveforms are summed to form a single waveform. This waveform is compressed into a single mp3 file and is the audio component of the interface. The result is a single audio file containing the X search results combined in an interleaving fashion, dictated by the weights $R_i^2(t)$. Since the search will return songs with different intensity levels and differing amounts of power at different times in the songs, before the songs are weighted by $R_i^2(t)$, it is important to normalize the local power. Therefore, before applying weighting, each individual song file is segmented into segments of length a and normalized so that all songs have equal power within the same segment. It is therefore the normalized waveforms which are weighted according to $R_i^2(t)$, rather than the raw song waveforms.

The visual component of the interface allows the user to distinguish between the multiple songs in the single audio file. Each of the X search results is represented visually by a circle of time-varying radius proportional to $R_i^2(t)$ (Figure 1). The X circles are organized in the space such that none of them are overlapping in space. This is the visual component of the interface, and with visual feedback, any of the songs are easily distinguishable. When the visual and audio components are started together at time $t = 0$ (Figure 1b), the time-varying behaviour of the circles provides the user with visual feedback with respect to the acoustic behaviour of the audio file. Based on the visual feedback, the user can easily identify which song they are hearing at a given time. The visual aspect of the interface is readily accomplished using client-side scripting which can be synchronized with the audio media. Therefore, the inclusion of visual media does not increase the bandwidth requirement of this system from the bandwidth required to send a single audio file.



(a) Search Result 1 at maximum volume



(b) Time Progression

Fig. 1. Visual Component of Interface

3. ADDITIONAL BENEFITS

With current interfaces for audio search, in order to evaluate X search results a user must download each individual search result file and listen to the files one-by-one. The identifiable part of a song occurs in general at different times for different songs. While some songs are most easily identified by their opening verse, other songs begin with a long introduction before reaching the portion of the song the user can easily recognize. Furthermore, the most memorable part of a song may vary from user to user, so even using the chorus of a song is not always appropriate. Therefore, in general, a song must be evaluated from the beginning. For sequential playback this means the user must listen to each song from the beginning,

waiting for a certain period of time until they are able to decide whether they are hearing the desired song or not. For X songs of length T, the search time can reach XT seconds in the worst case. For the proposed interface, in the worst case, only T seconds (the length of the single file) are required. Furthermore, the bandwidth required for the proposed interface is proportional to T rather than XT.

Processing multiple songs at the same time makes more efficient use of human perception capabilities. For example, if one of the search results begins with a long introduction, while waiting for that search result to reach the chorus, the user can concurrently evaluate other search results. Since the information from all search results is available at the same time, the user may be able to make a decision without waiting for each search result song to reach an identifiable part.

Furthermore, using current interfaces, if a user has listened to several search results, and decides to go back to reevaluate a previous one, the user must spend additional time to relocate the result and reevaluate it from the beginning. With the proposed interface, information from *all* songs is available at all times. Therefore using the proposed interface, there is a significant saving in listening time and in bandwidth requirements.

4. AUDIBILITY EXPERIMENTS

Experiments were conducted using the proposed audiovisual interface in order to demonstrate the ability to distinguish between the interleaved songs. If this can be verified, it follows that the search using the proposed interface will always be as fast or faster than evaluating songs one by one, since more than one song can be evaluated during any given interval of time. Experiments were conducted on 21 users.

The songs used in the experiments were chosen at random from a selection of 21 popular songs. Test subjects were given a song to search for using the proposed interface. When searching for a desired song, typically a user would be interested in searching for a song that they have previously heard. Therefore, before each experiment, test subjects were played a 5 second clip from the chorus of the desired song. Subjects were then timed as they searched for the desired song using the proposed interface. Testing was conducted using an interface consisting of three search results. There are two search scenarios of interest; the first is when the desired song is one of the three search results (Scenario 1) and the second is when the desired song is none of the search results (Scenario 2). The two scenarios were tested separately, without the knowledge of test subjects. Subjects were instructed to choose the search result corresponding to the desired song or to indicate that the desired song was neither of the search results. Two measurements were taken per scenario, per person.

The main parameter in the interface is $\omega = \frac{2\pi}{\tau}$, the speed of the cycling of the search result songs. For the experiments conducted here, $\tau = 8$ seconds.

The same set of users searched for songs using a traditional search interface, in which search results are organized in a list format. In this interface each song is listened to serially. In the initial state of this interface, the first search result is played from the beginning of the song. When the user clicks on a subsequent search result, the previous result stops playing and the new result begins to play from the beginning. This interface is analogous to current interfaces for audio search (without accounting for the download time required for each audio file). Testing was conducted for a list of three search results and for the same two scenarios described above (when the desired song was one of the search results, and when it was not one of the search results). In all experiments, the search results were ordered at random, but the desired song was always located within the search results such that it was not the first search result. Experiments were always conducted using the proposed interface first, and the traditional interface last. Therefore any learning of the introductions of the songs by the users throughout experimentation could only bias results towards the traditional interface.

5. RESULTS

For three search results, the average search time to correctly identify the desired song as one of the search results was 7.3 seconds. This is less than the times required to perform the search using the traditional list search interface. Average times are presented in Table 1. Figure 2 and Figure 3 show the spread of results for both interfaces for the two scenarios. Users who were initially apprehensive about evaluating multiple songs simultaneously were able to effectively use this interface.

Using the traditional list interface with three search results, on average it took 12.1 seconds to correctly identify the desired song. The length of time associated with listening to each search result frustrated some users and due to lack of patience some users made the wrong selection using this interface. This is especially true as the list increases in length.

These results in Table 1 demonstrate the advantage in search time provided by the proposed interface over current interfaces, under Scenario 1. Although the average search time for the traditional interface is substantially larger than that of the proposed interface, Figure 2 illustrates that this is largely the result of several outliers in the experimental data. These outliers can be attributed to situations where the test subject was unfamiliar with the introduction portion of the search song. However, even if we disregard the outliers, there is still an advantage in search time using the proposed interface compared to the traditional one.

6. CONCLUSIONS

The proposed interface combines audio and visual components to allow users to effectively search for audio content.

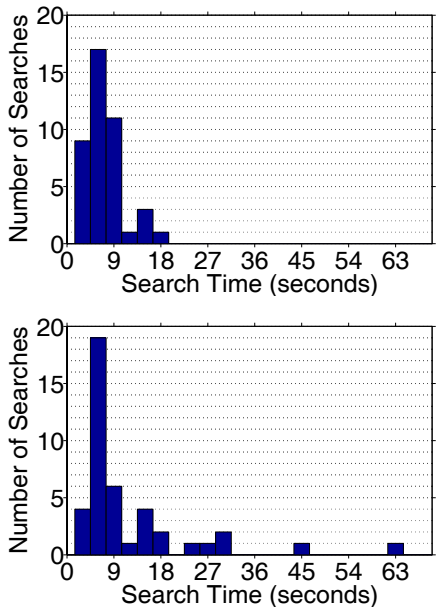


Fig. 2. Experimental Results for Scenario 1: Desired song included in search results, with the proposed interface (top) and with the traditional interface (bottom)

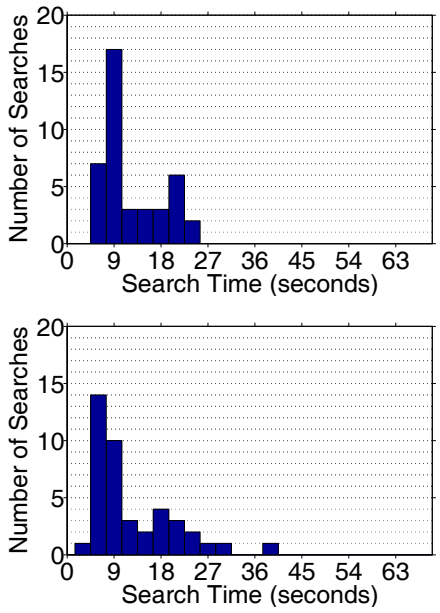


Fig. 3. Experimental Results for Scenario 2: Desired song not included in search results, with the proposed interface (top) and with the traditional interface (bottom)

Table 1. Average Search Times (in seconds)

	Scenario 1	Scenario 2
Proposed Interface	7.3	12.5
Traditional Interface	12.1	12.5

Experimental results show the ability to distinguish between interleaved songs and an advantage in search time over using a traditional list-based interface. Since the proposed interface requires only the bandwidth required to transfer one file in order to evaluate multiple search results, there is also a significant savings in bandwidth.

7. REFERENCES

[1] B. Logan and S. Chu, "Music summarization using key phrases," in *Proc. IEEE ICASSP*, 2000, vol. 2, pp. II749 – II752.

[2] M.A. Bartsch and G.H. Wakefield, "Audio thumbnailing of popular music using chroma-based representations," *Multimedia, IEEE Transactions on*, vol. 7, 2005.

[3] S. Srinivasan, D. Ponceleon, A. Amir, and D. Petkovic, "What is that video anyway?: In search of better browsing," in *Proc. IEEE ICMCS*, 1999, vol. 1, pp. 388–393.

[4] D. Ponceleon, A. Amir, S. Srinivasan, T. Syeda-Mahmood, and D. Petkovic, "Cuevideo: automated multimedia indexing and retrieval," in *Proc. of the seventh ACM international conference on Multimedia*, 1999, p. 199.

[5] S. Colbath, F. Kubala, D. Liu, and A Srivastava, "Spoken documents: Creating searchable archives from continuous audio," in *Proc. of 33rd Hawaii International Conference On System Sciences*, 2000, pp. 1–9.

[6] R. Hiraga, "Case study: a look of performance expression," in *Proc. of the conference on Visualization*, 2002, pp. 501–504.

[7] J. Zhu and L. Lu, "Perceptual visualization of a music collection," in *Proc. IEEE ICME*, 2005, pp. 1058–1061.

[8] S.M. Smith and G.N. Williams, "A visualization of music," in *Proc. of the 8th conference on Visualization*, 1997, pp. 499–503.

[9] R. Hiraga and N. Matsuda, "Visualization of music performance as an aid to listener's comprehension," in *Proc. of the working conference on Advanced Visual Interfaces (AVI)*, 2004, pp. 103–106.

[10] J. Foote, "Visualizing music and audio using self-similarity," in *Proc. of the seventh ACM international conference on Multimedia*, 1999, pp. 77–80.

[11] E. Pampalk, A. Rauber, and D. Merkl, "Content-based organization and visualization of music archives," in *Proc. of the tenth ACM international conference on Multimedia*, 2002, pp. 570–579.