Searching for Music: How Feedback and Input-Control Change the Way We Search

Tue Haste Andersen

Department of Computer Science, University of Copenhagen,
Universitetsparken 1, DK-2100 Copenhagen Ø, Denmark
haste@diku.dk

Abstract. The growing amount of digital music available at desktop computers and portable media players increases the need for interfaces that facilitate efficient music navigation. Search patterns are quantified and evaluated across types of feedback and input controllers in an experiment with 12 participants. The way music is searched and the subjective factors varied significantly across input device and type of audio feedback. However, no difference in task completion time was found for the evaluated interfaces. Based on the experiments, we propose several ways in which future designs may improve searching and browsing in recorded music.

1 Introduction

Today it is not uncommon to have a large collection of digital music available, some of which has never been heard by the user before. CD players and PC or mobile based media players offer only limited control of playback for browsing and searching in music. A way to quickly browse a large number of songs to find one that was heard on the radio or one that fits the user's musical preference and mood is needed. Content based retrieval [6] is often identified as the solution to this type of problem. However, content based retrieval requires an extensive amount of computing power and infrastructure, something that is not yet available as services to customers of consumer audio equipment. With better interfaces for browsing and listening to the music at the same time could help improve this situation.

As opposed to the somewhat primitive interface for music navigation offered by CD players and common computer based media players, several interfaces previously presented used information extracted from the audio signal to aid in navigating the music [9,2]. Goto proposed an interface to improve trial listening of music in record stores [7] where visualization and structure information was used. The interface was evaluated in an informal study where it was found that both the visual display and the structure jump functions were more convenient than the traditional CD player interface. However, we do not know if the increase in convenience was due to the use of segmentation information (meta-data), or because the visual display and better means of input control were provided.

M.F. Costabile and F. Paternò (Eds.): INTERACT 2005, LNCS 3585, pp. 144–157, 2005.

This paper presents a baseline study that investigates the role of different types of aural feedback, along with different types of input control. The evaluation was done on both qualitative measures and observations, and on quantitative measures such as task completion time. The results can serve as guideline for future designs and research in music browsing interfaces.

In section 2 we review current interfaces and research in music browsing. Section 3 presents the experiment. In section 4 we discuss the results and its implications on future designs.

2 Interfaces for Browsing Audio and Music

In this section interfaces for browsing music is briefly reviewed, along with interfaces used in research of this area. A major difference between these interfaces is how the browsing interface is controlled. Different controller types are used, but little is known about how the controller influences the searching of music. Another area where these interfaces differ is in how audio feedback is presented during the search. Different audio synthesis methods for search feedback are described.

2.1 Today's Interfaces

In the following we classify today's interfaces for music browsing and searching based on their input device.

Buttons are probably the most common type of input controller used in interfaces for music browsing and searching. Standard CD players provide a set of buttons to control playback. Audio feedback is provided as the music is being played. In addition, a visual indication of the position in the current playing track is given together with the number of the current playing track. Buttons for play, pause, fast forward, fast backward, and previous and next track are provided. These functions are often combined into fewer buttons which activate different functions based on how long a button is pressed. CD players can play back audio in two modes: Normal playback mode, where the audio is played back at the speed at which it was recorded, and fast forward/backward mode where the audio is played at a faster speed than what is was recorded. Common scale factors are in between 4 and 5 times normal playback speed, although some players allow for even faster playback.

Sliders are most often used to control playback position in digital media players on PCs and portable devices. On PCs sliders are implemented as graphical widgets controlled with a mouse, but on media players the control is often done through a touchpad directly mapped to the graphical representation of the slider. Sliders provide random access to the play position within a track, as opposed to the buttons where the playback position can only be changed relative to the current position. The slider functions both as an input device and as visual and/or haptic display, giving feedback about the current play position relative to the length of the track. Media players also facilitate ways to browse music by title, album, or other meta-data if available.

Rotary controllers are used in DJ CD players, where they are referred to as jog wheels. Here the playback position can be changed relative to the current playback position. The rotary as implemented on DJ CD players allow for fine-grained control of the playback position, while at the same time the inertia of the rotary helps to move the play position to fast forward with little physical effort. The change from playback mode to fast forward or backward mode is not explicit as with the buttons, but continuous. The feedback is also changed continuously by linear interpolation or other interpolation methods that preserve pitch [11,10].

Sliders and buttons or other hybrid controllers are used for example in sound editors. Sound editors are application tools used by musicians and audio engineers that provide several ways to navigate within a music track. The mouse can be used to randomly change play position by clicking on a slider or waveform display, similar to the way play position is changed in a media player. Playback rate can often be adjusted, and finally sound editors provide better means of visual feedback in form of waveform displays supplemented by color codes [12] and spectrograms.

Because audio is tightly related to the time dimension, input controllers or widgets used in controlling audio playback are most often one dimensional. However, the controllers differ on how the absolute position can be changed. Sliders allow for fast random seek while buttons used on CD players do not. Active haptic feedback in input controllers for audio navigation has been used [13]. However it is not entirely obvious if this type of feedback can improve aspects of audio navigation [3].

2.2 Audio Feedback at Different Levels

Little research addresses the searching and browsing of music. For speech, Arons [2] presented an extensive study on the interface design in speech skimming. The design of Arons' SpeechSkimmer interface was based on the notion of a *time scale continuum*, where speech could be auralized using different time compression ratios, based on what type of skimming the user wanted. On the lowest level of the time-scale continuum was uncompressed speech, continuing to pause removal where the same signal was compressed by removing pauses. The highest level was pitch-based skimming where the individual words were no longer audible, but the pitch of the speaker's voice could still be perceived.

Interfaces for music browsing may benefit from a similar time scale continuum, because most music is characterized by repetitions at many different levels. First, most popular music has a beat, where percussive sounds often are repeated with an interval between 0.25 to 2 seconds. Second, music has a rhythm, a certain structure that the beat follows. Third, popular music is divided into sections so that verses are separated by a repeating chorus and other structures such as breaks.

When doing fast forward playback using CD players and sound editors, the sound is transformed in some way to allow for faster playback. In CD players, feedback serves mainly to inform the user that the fast forward mode is activated.

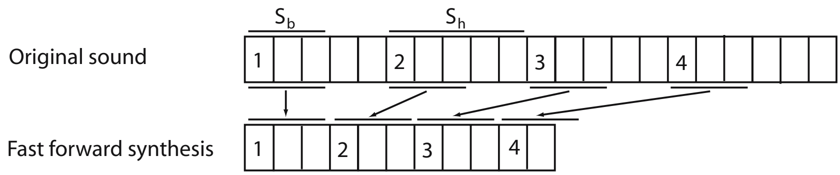

Fig. 1. Figure showing sound synthesis for fast forward as implemented in standard CD players. Blocks of samples (S_b) are copied from the original sound to the fast forward synthesis sound. The hop size (S_h) determines the scale factor.

The average volume level can still be perceived while playing in fast mode, but features such as rhythm, instrumentation and timbre are hard or impossible to perceive. In sound editors time scaling techniques [11,10] are used that preserve timbre and other features, but the interval between consecutive beats is changed. This can be a problem when the listener is searching for a song with a particular style or mood. Using a time scale continuum for music where local features such as rhythm and timbre are still perceivable while searching at high speed could therefore be a way to improve browsing interfaces.

2.3 Synthesizing Audio for Fast Forward Playback

The most widespread method to synthesize audio in fast forward mode from recorded audio is that of playing small chunks of audio with a block size, S_b between 5 and 20 msec, see Figure 1. The method is essentially an isochronous sampling of the audio. Blocks of audio are sampled from the original recording to form a new signal that is used for playback in fast forward mode [14]. Window functions can be used to smooth the transition from one frame to the next [5]. In backward searching, the individual frames are usually played in a forward direction. The method is used in most CD players today as it is suitable for implementing in a CD player with very limited further processing or buffering. The method is most often implemented with a fixed speed of four to five times the normal playback speed, but in principle it can be used at an arbitrary speed factor, by changing the hop size, S_h, shown on Figure 1. The fast forward audio synthesis as implemented on CD players serves at least two purposes: It allows for perception of some acoustic features while scanning, and it gives the user feedback about which mode is activated. When listening to fast forward synthesis on a CD player, the user is not in doubt that the CD is in fast forward mode. On CD players the block size S_b is very short, but it could be increased to allow for better perception of beat, rhythm and instrumentation at the expense of having a strong feedback about which mode the CD player is in. When increasing S_b to be in the range of seconds rather than milliseconds and increasing the speed by increasing S_h, the effect becomes close to a high level skimming of the music, similar to the high level skimming in the time scale continuum used in SpeechSkimmer.

Other alternatives to compressing music exist. Linear interpolation of the sample values effectively scales the audio in a similar way to changing the tempo on a vinyl record. Not only is the tempo changed but also the pitch. An alternative is time scaling techniques that preserve the pitch, such as the phase vocoder [11,10]. These methods preserve the timbral characteristics up to about twice the normal playback speed. However, for fast forward playback speeds, generally above two times normal playback speed is wanted.

3 Experiment: Comparing Interfaces for Mobile and Consumer Devices

In the experiment, participants had to search for a song containing an audio segment heard in advance. The task is similar to searching for a track heard on the radio. The interfaces used in this comparison are all applicable to audio consumer devices and a mobile usage situation. Two independent variables were used in a fully crossed experiment: Input controller and audio feedback. As reference, the CD player interface was compared with three other interfaces. We wanted to test if an input controller such as the rotary controller, which allowed for fast change of playback position when compared to the buttons on a CD player, resulted in fast task completion time and improved satisfaction. Second, we wanted to examine how much time was spent in fast forward mode when searching for a particular song, and how the audio feedback given during fast forward influenced the search performance and satisfaction.

3.1 Interface Design

We used two independent variables: Type of audio feedback (skip, play), and type of input controller (button, rotary). Each variable had two levels as described below.

Audio Feedback. The audio feedback during normal playback was the same in both levels of the feedback variable. When in fast forward or backward mode the audio feedback was synthesized using isochronous sampling as describe above. In the first condition, block size was the same as used on a CD player, $S_b = 10$ msec. The condition is referred to as "skip" because it sounds like the CD read head is skipping. The other level of the feedback variable we choose to refer to as "play" where the block size is one second. In both cases, the step size between each block played is adjusted according to the movement speed. The two types of feedback are fundamentally different, in that "skip" results in a strong sensation of moving forward, but makes it impossible to perceive most features of the music. Play feedback on the other hand does not give a strong sensation of the movement speed, but local features such as rhythm and instrumentation can still be perceived.

Input Control. One level was a set of five buttons, referred to as "buttons," identical to the input method found on most CD players. There was a pair of buttons for switching to previous and next track, a pair of buttons for moving at fast backward or forward, and finally a button controlling playback/pause mode. The fast backward and forward buttons moved at four times normal playback speed. The second condition was the use of a PowerMate[1] rotary controller ("rotary"); moving the rotary knob to either left or right initiated a fast forward or backward operation. The speed at which the rotary was moved determined the movement speed (rate). Playback/pause was controlled by pressing the non-latching button built into the knob. There are two fundamental differences between these two interfaces: The input controller and the mapping. The mapping used with the rotary controller allows for a continuous change of the fast forward playback speed, while the mapping used with the buttons provide one fixed fast forward speed and the ability to do a non-linear jump to the next track. We chose to use different transfer functions for the two input controllers to achieve natural mappings. However, this could have been avoided by mapping the rotary controller to a constant fast forward speed regardless of the speed at which the rotary was operated, and to jump to the next track when above a certain threshold.

Before the experiment, we hypothesized that the play feedback would be superior to skip feedback. The time spent in fast forward mode could be used to listen to local features of the music and could potentially provide more useful information than the skip feedback. Also, we expected the rotary controller to be superior in terms of performance compared to the buttons, primarily because the buttons only allowed a constant, and relatively slow, fast forward speed. With the rotary it would be possible to quickly move to a different part of the track, but on the other hand would require slightly more effort to move to the next track. We decided to do a fully crossed experiment because we did not know if the audio feedback would influence the task performance in any way, depending on the type of controller.

The interfaces were developed using the software Mixxx [1], but with a modified user interface. In the experiment we sought to minimize the influence of visual feedback, and provide a usage situation similar to mobile music devices. Thus we only provided visual information about which track was currently loaded, the absolute position inside the track, and the length of the current loaded track.

3.2 Tasks

We used 20 tasks; each participant completed five tasks in each condition. Each task consisted of a list of nine tracks selected from the Popular, Jazz and Music genre collection of the RWC Music Database [8]. The database was used to ensure that the participants had not heard the music before, and to get a representative selection of popular music. For the 20 tasks, a total of 180 unique music tracks were used. For each task, the list of tracks was constructed in such a way that the tracks were reasonably similar in style, but there was no logical order to the tracks in each list. The music varied in style between tasks, some were instrumental, and some was sung in English and others in Japanese. The 20 tasks were divided into four groups so that each group contained five tasks. In

[1] See `http://www.griffintechnology.com/products/powermate/`

the experiment each group was used in one condition. The four groups were always presented in the same order, but the order of the four conditions was balanced using a Latin square pattern.

The target track of each task was selected randomly, but ensuring that the average target in each group was at position 5 in the track list. This was done to ensure that the groups were close to each other in terms of task complexity.

For each task, the participant was presented with a 10 second long audio excerpt of the target track. The excerpt was chosen to contain the refrain or other catchy part of the song and could be heard as many times as the participant wanted. The participants were required to hear it three times in succession before starting to search for the corresponding song. They were also allowed to hear the except during the search if they had forgot how it sounded. They could navigate the songs using the interface provided, and pressed a button when they had found the song from which the audio excerpt was taken.

3.3 Design and Setup

A fully-crossed within-subjects factorial design with repeated measures was used. Twelve people participated in the experiment. Their musical skills ranged from being a professional musician to having no particular interest in music, with most participants having no musical skill. Independent variables were feedback and input control type. We used two types of dependent variables:

Task Performance. Completion time, error in identification, and time spent in fast forward/backward mode.

Subjective Satisfaction. Rating on five scales based on Questionnaire for User Interface Satisfaction [4] related to ease of use, learning and aesthetics. The scales are shown in table 1.

The experiment was conducted in a closed office, using a computer with CRT screen and attached studio monitors. Studio monitors were chosen over headphones to allow for easy observation of the participants during the experiment. The participants completed one training task followed by five tasks in each condition. After each condition, they rated the interface on the five subjective satisfaction scales before proceeding to the next condition. At the end of the experiment, they could adjust the ratings of all interfaces. The experiment was concluded with an open ended interview. During the experiment, an observer was present and the interaction was logged.

3.4 Results: Task Performance

The data was analyzed using analysis of variance with feedback and input control as independent variables, and task completion time, number of errors and time spent in fast forward mode as dependent variables. To examine how the individual tasks influenced the experiment, and to look for learning effects, an analysis was also performed with task as an independent variable.

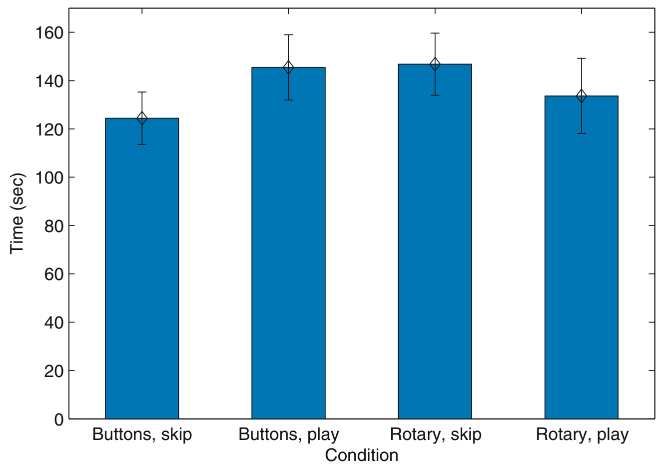

Fig. 2. Mean time used in each condition

Figure 2 shows the mean and standard deviation of the completion time in seconds for each condition. No significant difference was found for feedback, $F(1, 11) = 0.000, p = .998$ or input control, $F(1, 11) = 0.022, p = .885$. However, when looking at the completion time as a function of task there was a significant difference, $F(19, 209) = 4.493, p \leq .001$. This is also shown in Figure 3 (top). A post-hoc comparison using a Bonferroni test at a 0.05 significance level revealed that task 2 and 7 differed significantly at $p = .037$, task 2 and 17 at $p = .039$ and finally task 3 and 19 at $p = .046$. During the experiment, it was observed that for some participants there was a large learning effect in the first few tasks, whereas others did not seem to improve during the course of the experiment. The significant difference between some of the tasks is likely caused by a combination of long completion time (learning) for the first three tasks, as seen on Figure 3 (top), and from the difference in target track position. The target track in task 17 and 19 is track 2 and 3, respectively, which means that the completion time on average is relatively short, since only one or two tracks has to be searched before the target is reached. In comparison, the target track of task 2 and 3 is track number 9 and 6, respectively, where a long completion time is observed.

On figure 3, (bottom) completion time is plotted as a function of target track rather than task. There indeed seems to be a correlation between completion time and distance to target.

Number of errors did not change significantly across feedback, $F(1, 11) = 0.316, p = .585$, and input control, $F(1, 11) = 0.536, p = .496$. Error across task was significant at $F(19, 209) = 2.257, p = .003$, however, no learning effect or other systematic effect was observed.

Figure 4 shows time spent in search mode (fast forward or backward) for the four conditions. The time spent in search mode varied significantly depending on feedback, $F(1, 11) = 9.377, p = .011$, and input control, $F(1, 11) = 7.352$, $p = .020$. This reveals that even though task completion time did not vary with

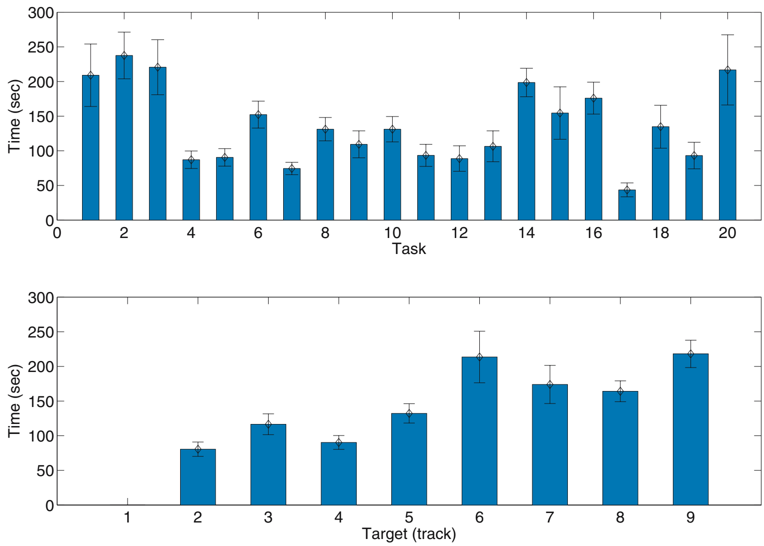

Fig. 3. Completion time as a function of task (top) and as function of target track (bottom)

interface and feedback, the participants' search patterns did. Figure 4 shows that the search time spent in the Buttons, Play condition was almost double that of the other conditions.

Figure 5 shows four examples of the search patterns in task 16, one for each condition. Position is shown as a function of time, and red circles mark where the user is engaged in a search action. Comparing the two button conditions, it is evident that in the examples shown, more time is spent in search mode in the play condition. Comparing the button conditions (top) to the rotary conditions, (bottom) a clear difference in search strategy is visible. In the rotary conditions, more parts of each track are heard by jumping forward in the track. In the button condition this is not possible without spending a lot of time, because the fast forward speed is only four times normal playback speed compared to the adjustable search speed with the rotary controller. In the rotary conditions, no immediate difference that can be explained by other observations is visible. The feedback does not seem to influence the search behavior here.

3.5 Results: Subjective Satisfaction

The ratings on the subjective satisfaction scales were analyzed using analysis of variance on each scale. The scales used in the subjective evaluation are shown in Table 1, along with F and p values for the independent variables interface and feedback. The mean ratings for each condition is shown in Figure 6. The input method influenced most of the scales. Buttons were more frustrating and more terrible than was the rotary. Both input method and feedback type were

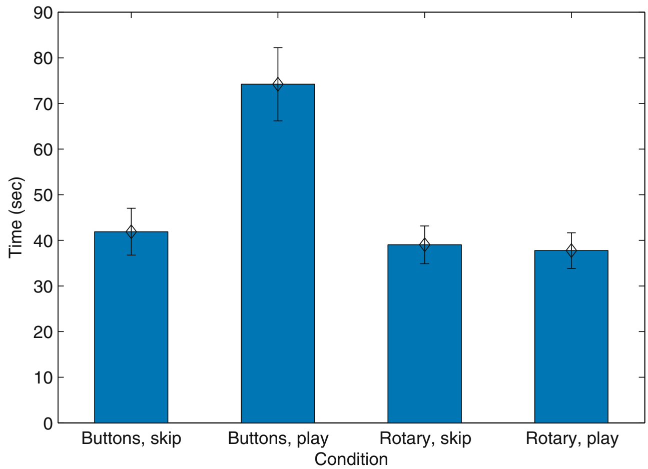

Fig. 4. Mean time used in search mode in each condition

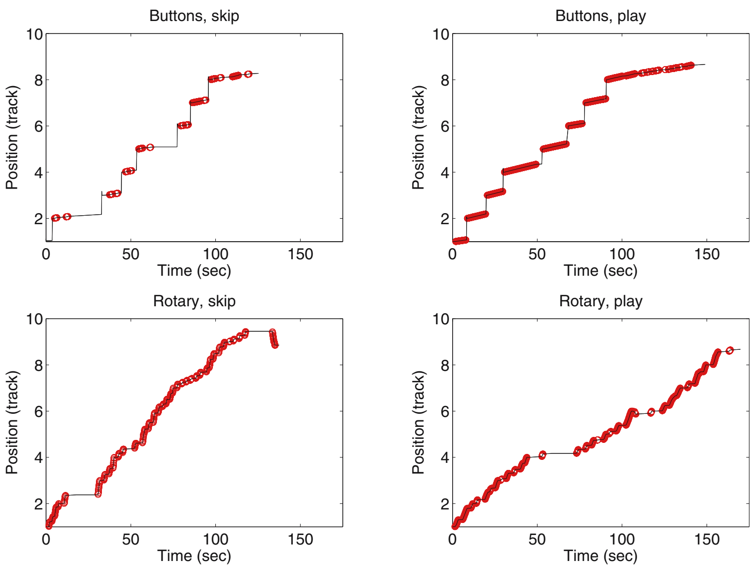

Fig. 5. Example. Position as a function of time plotted for one trial of each condition of task 16.

significantly different on the scale of Terrible-Wonderful, and buttons with play feedback were more wonderful than buttons with skip feedback. The case was the same for the rotary controller, and the rotary controller was overall more wonderful than the buttons. Many participants commented that they did not know how to interpret the responsiveness scale, but there is a significant difference across input type. On average, the participant perceived the rotary to be

Table 1. Scales used in subjective evaluation

Scale	Input method	Feedback type
Frustrating - Satisfying	$F(1,11) = 21.154, p \leq .001$	$F(1,11) = 1.222, p = .293$
Terrible - Wonderful	$F(1,11) = 5.260, p = .043$	$F(1,11) = 7.857, p = .017$
Not responsive - Responsive	$F(1,11) = 11.875, p = .005$	$F(1,11) = 0.40, p = .845$
Difficult - Easy	$F(1,11) = 0.208, p = .658$	$F(1,11) = 0.268, p = .615$
Straightfwd. (Never - Always)	$F(1,11) = 3.313, p = .096$	$F(1,11) = 6.600, p = .026$

more responsive than the buttons. This makes sense, since with the rotary it is possible to move faster forward in a song than it is with the buttons. The buttons with skip feedback was rated to be easiest to use. This can be explained by the fact that all participants were well acquainted with this interface through the use of standard CD players. It was not clear to the participants if the straightforward scale related to the interface or to the tasks; thus we chose not to analyze it further. Finally two participants commented that it was difficult to use the play feedback with the rotary controller because they were forced to look at the visual position display to maintain an idea of which part of the song the system was playing. These participants preferred to operate the system with eyes closed, which was possible in the other conditions.

Three participants commented that they especially liked the play feedback during search when using the buttons, because they did not have to leave the search mode. Few participants used the rotary to search at a relatively low speed. Instead many participants did search at high speed during short intervals to advance the play position. A few participants commented that they liked the skip feedback better here, because it made them aware that they were searching.

3.6 Discussion

In conclusion, the most surprising finding of the experiment was that no significant difference in completion time was observed for the tested interfaces, even though a significant difference in search strategy was observed. We wanted to test if the average completion time was similar to that of navigating using a sound ed-

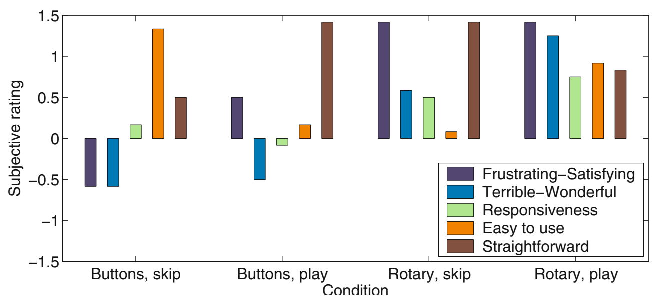

Fig. 6. Subjective ratings of the 4 interfaces on scales ranging between -2 and 2

itor with a waveform display and play position control through a graphical slider widget operated with a mouse. Therefore, we conducted a new experiment with six participants from the previous experiment. An informal comparison showed that no significant difference was observed. This indicated that the task completion times reported in this experiment is not likely to be affected by further improvement of visual feedback or controller input type.

We found that on average participants spent 35% of their time in search mode. There was a significant difference in how people used the interfaces, with participants spending significantly more time in search mode for the condition with button interface and play feedback. In the two conditions with skip feedback, it was hard to perceive features of the music. This explains why less time was used in search mode for these conditions, compared to the Play, Button condition. However, it is somewhat surprising that with play feedback, only the buttons resulted in more time spent in search mode. With the buttons it was only possible to move fast forward in a track by four times the normal playback speed, compared to an almost arbitrary fast forward speed using the rotary. Using the rotary, most users would move forward in a short time interval at fast speed, then stop the rotary motion to hear part of the song. Only a few participants moved slowly forward using the rotary to take advantage of the play feedback. Two problems were evident from the way the participants used the rotary with play feedback: first, no immediate feedback was given that a search was initiated or in progress. Only after one second of spinning the rotary was feedback audible. Second, to keep a constant fast forward speed, the participant would have to spin the rotary at a constant speed, and thus keep the hand in motion, as opposed to using the buttons, where constantly holding down a button would result in a constant fast forward speed. This suggests that a rate based mapping rather than a position based mapping might be a better alternative when using the play feedback scheme.

A large significant difference was found in how participants perceived the interfaces. In general, the play feedback was liked over skip feedback, and the rotary was preferred over the buttons. It is interesting that the responsiveness scale was not influenced significantly by feedback type, but a significant difference was observed for input control. Some participants commented that the perceived responsiveness was influenced by the type of feedback. In particular, one participant commented that he could not use the rotary with play feedback with eyes closed, because he lacked feedback about the playback position in the song. Overall, it seemed that the type of input control was more important than the type of feedback given to how participants liked the interface. The rotary was rated more satisfying and wonderful than the buttons. This may be due to aesthetic factors of the input controller, where the buttons was implemented using a standard keyboard, and the rotary was utilizing the PowerMate controller in aesthetically pleasing brushed metal. Another explanation could be that the mapping used with the rotary allowed for rapidly seeking to an arbitrary position in the track.

4 Conclusions

This study focused on feedback and input when searching in recorded music. We presented a novel feedback method, the “play” feedback, where segments of one second were played during fast forward or backward mode. The “play” feedback allowed for aural scanning of local features of music such as rhythm and timbre, while searching at fast speed. The method allows for perception of local features and seamlessly integration of structural information when available. The feedback method was compared to the “skip” feedback, identical to the feedback given by ordinary CD players. The two types of feedback were compared with two input controllers, one based on buttons and other based on a rotary controller, in a fully crossed experiment.

The most surprising finding of the experiment was that we did not observe a significant difference in task completion time or number of errors between any of the tested conditions. To get an indication of the performance of the tested interfaces we compared them to the performance of a state of the art interface, similar to interfaces implemented in sound editors. In the informal experiment, no significant difference was found in task completion time and number of errors. This indicates that no immediate gain in search performance can be expected by providing better means of input control or visual feedback.

However, we did find a significant difference between feedback type and input control for the time spent in search mode. The interfaces using buttons as input control and the “play” feedback did result in a significantly higher portion of the time spent in fast forward mode compared to the other interfaces. Thus, in this condition, the participants had more time where it was possible to perceive features such as timbre, instrumentation and rhythm. A similar increase in time spent in search mode was not observed for the rotary controller with “play” feedback. This can be explained by the fact that the rotary controller allowed for faster change of playback position. Thus ordinary play feedback was needed earlier than one second after a search action was initiated.

We observed large significant differences in how the interface was perceived by the participants. On average participants found the rotary controller more satisfying, wonderful and responsive than the buttons. The “play” feedback was also significantly more wonderful than the “skip” feedback. During the open ended interviews, several participants commented that the play feedback was better than skip feedback, but did not result in a feeling of moving forward. Future interfaces may thus improve on both satisfaction and responsiveness by mixing the “play” and “skip” audio signal into one. Other ways to further improve the feedback could be to use segmentation information to jump only to places where the music changes, and to use beat information to perform the isochronous sampling relative to the beat phase, to ensure a smooth transition from one block to the next.

We know from research in speech navigation that meta data can improve search performance [2] and that meta data in musical interfaces influences subjective evaluation of the interface in a positive way [7]. However, we do not have any evidence that it will actually improve performance in music navigation in search tasks, even though it intuitively seems likely.

Acknowledgments

I would like to thank members of the HCI group at DIKU, in particular Kasper Hornbæk for his insight during the planning of the experiment and for feedback on the manuscript, as well as Georg Strøm who provided valuable feedback during the writing of the paper.

References

1. T. H. Andersen. In the Mixxx: Novel digital DJ interfaces. In *Proceedings of CHI 2005*, pages 1136–1137. ACM Press, April 2005. Demonstration and extended abstract.
2. B. Arons. SpeechSkimmer: A system for interactively skimming recorded speech. *ACM Transactions on Computer-Human Interaction*, 4(1):3–38, March 1997.
3. T. Beamish, K. Maclean, and S. Fels. Manipulating music: multimodal interaction for DJs. In *Proceedings of CHI*, pages 327 – 334, 2004.
4. J. Chin, V. Diehl, and K. Norman. Development of an instrument measuring user satisfaction of the human-computer interface. In *Proceedings of CHI*, pages 213–218, 1997.
5. G. Fairbanks, W. Everitt, and R. Jaeger. Method for time or frequency compression-expansion of speech. *Trans. Inst. Radio Eng. Prof. Group Audio AU-2*, pages 7–12, 1954. Reprinted in G. Fairbanks, Experimental Phonetics: Selected Articles. University of Illinois Press, 1966.
6. J. Foote. An overview of audio information retrieval. *Multimedia Systems*, 7(1):2–10, 1999.
7. M. Goto. SmartMusicKIOSK: Music listening station with chorus-search function. In *Proceedings of UIST*, pages 31–40, 2003.
8. M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC music database: Popular, classical, and jazz music databases. In *Proceedings of ISMIR*, pages 287–288, 2002.
9. D. Kimber and L. Wilcox. Acoustic segmentation for audio browsers. In *Proceedings of Interface Conference*, 1996.
10. J. Laroche and M. Dolson. New phase-vocoder techniques for real-time pitch shifting, chorusing, harmonizing, and other exotic audio modifications. *J. Audio Eng. Soc.*, 47(11):928–936, 1999.
11. M. Portnoff. Time-scale modifications of speech based on short-time fourier analysis. *IEEE Transactions on ASSP*, 29(3):374–390, 1981.
12. S. Rice and M. Patten. Waveform display utilizing frequency-based coloring and navigation. U.S. patent 6,184,898, 2001.
13. S. Snibbe, K. MacLean, R. Shaw, J. Roderick, W. Verplank, and M. Scheeff. Haptic techniques for media control. In *Proceedings of UIST*, pages 199–208, 2001.
14. T. Yokota, N. Kihara, and J. Aramaki. Disc playback method. US patent 5,553,055, September 1996. Sony Corporation.