# Improving Interfaces for Navigating Continuous Audio Timelines

**Eric Lee, Thorsten Karrer, and Jan Borchers**
Media Computing Group
RWTH Aachen University
52056 Aachen, Germany
{eric, karrer, borchers}@cs.rwth-aachen.de

## ABSTRACT

Recent advances technology have dramatically increased production and consumption of digital media such as audio. The intrinsically time-based nature of audio, however, presents unique problems, in particular with navigation. Unfortunately, audio navigation interfaces do not differ widely from the decades-old tape recorder metaphors. In this paper, we discuss our current work in exploring audio navigation interfaces for non-professional audio producers and consumers based on user studies that we have conducted and systems that we have developed for audio timeline navigation.

### Author Keywords

Audio interfaces, audio scrubbing, audio timeline navigation, empirical study, time-based media, time-stretching.

## INTRODUCTION

Technology advancements in recent years have made continuous time-based media such as audio and video a popular medium for electronic communication. Increasing availability of computers for content creation, the internet for content distribution, and portable media players such as the iPod (www.apple.com), have dramatically increased both creation and consumption of these digital media types. Radio, for example, has in recent years been reborn in digital form as "podcasts"; with today's technology, it is possible for even the average home user with a computer to create such content, and distribute it to a global audience. According to Apple, over one million podcasts were already subscribed to, just two days after the iTunes podcast directory became available [1].

Despite this increased popularity, however, audio navigation interfaces do not differ significantly from the "tape recorder" metaphors of *play*, *stop*, *fast-forward* and *rewind* from the 1950's. In contrast, text document navigation interfaces have been examined much more thoroughly in literature, with studies comparing rate and position controls conducted since the 1970's: a detailed overview of these works is given in [12]. There do not appear to be, however, any existing attempts to generalize the conclusions drawn from this body of research to audio navigation.

In the following sections, we will briefly outline our work for better supporting navigation through a continuous audio timeline. We begin with a brief comparison of text and audio document navigation, and from this discussion we propose a design space of audio navigation interfaces. We also present a technique we developed for audio navigation using direct manipulation, and conclude with open questions that could result in potential topics for discussion at the workshop.

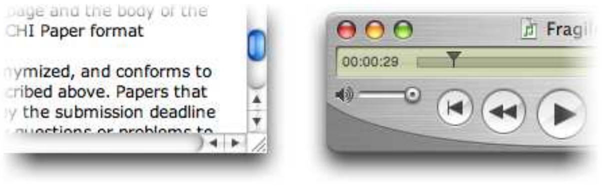


**Figure 1. The scrollbar for document navigation (left) is analogous to the timeline slider for audio navigation (right).**

## A COMPARISON OF TEXT AND AUDIO NAVIGATION

Navigating through the timeline of continuous time-based media, such as an audio recording, is similar to navigating through a text document in many respects. Despite the spatial nature of documents, as opposed the temporal one of audio, the *input* techniques used for scrolling through a document often apply to audio as well. Audio navigation differs from document navigation, however, in how *feedback* is provided to the user while scrolling.

### Input

A common software interface widget for scrolling through audio is the timeline slider, analogous to a scrollbar in a document window. The *wiper* inside the scrollbar, which controls the current viewing area in a document, is the *playhead* in an audio timeline slider. The arrow buttons at either ends of a scrollbar correspond to the *fast-forward* and *rewind* buttons (see Figure 1).

Zhai et al. [13] observed, however, that the scrollbar interface for navigating through a document suffers from at least three drawbacks: time is required to acquire the wiper; scrollbars are ill-suited for continuous scrolling with precision; and navigating to the scrollbar shifts the user's locus of attention away from the target. The timeline slider uses the same mappings, and thus suffers from similar drawbacks; moreover, the playhead in a timeline slider is usually very small, and thus acquiring it is even more difficult than with a scrollbar (see Figure 1).

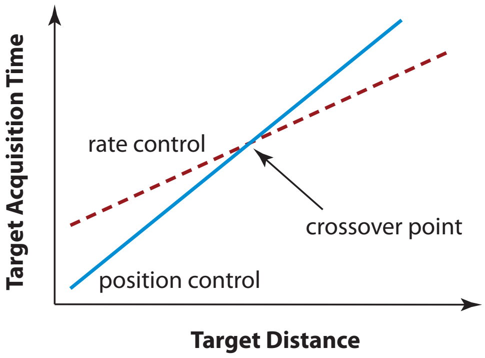


**Figure 2. Hypothesized crossover effect. Position control was shown to be faster when acquiring close targets, and we expect rate control to be superior for more distant targets.**

As a result, alternative interfaces to more efficiently navigate a document have been studied. These interfaces typically control *rate* (user input maps to scrolling velocity), or *position* (input maps to viewing area position). For audio navigation, a rate control would similarly control the play rate, and a position control the current playhead position. Unlike spatial navigation, where both position and rate controls have been studied extensively and their merits widely debated [12], rate controls appear to be the de facto standard for audio navigation; the sliders for playback speed in QuickTime Player and Windows Media Player, and even the ubiquitous *fast-forward* and *rewind* are rate controls.

Hinckley [3] showed that for document navigation, position control is faster than rate control for closer search targets, but as the search target distance increases, the performance difference becomes less significant until a "crossover point", when rate control becomes superior (see Figure 2). We believe this same crossover effect applies to audio navigation as well, and in a recent user evaluation, we showed that position control is significantly faster than rate control for audio targets that are 90 to 100 seconds away from the current position [8]. We are currently conducting more extensive studies to explore the nature of this crossover effect for audio in more detail.

### Feedback

While spatial and temporal navigation may share many similarities in input, they differ in how feedback is presented to the user. Hürst observes that when navigating through continuous time-based media, only the smallest unit (e.g., one video frame, or one audio sample) can be conveyed to the user at any moment in time [5]. In spatial media such as a text document, many lines of text can be displayed to the user at the same time. Moreover, temporal media must often be perceived over time; a single audio sample, for example, has no meaning by itself.

Fortunately, workarounds to this limitation have been developed. Audio editors, for example, represent the temporal dimension spatially by displaying a section of the audio waveform, thus allowing the user to visualize multiple instants of the audio timeline concurrently. While such visualizations are useful for locating "meta-features", such as pauses between words and sentences, even experienced audio editors are unable to derive the words of a speech recording, or the melody of a song by simply looking at its waveform. Thus, Hürst's statement still holds true, and audio must be interpreted over time to fully understand all nuances of its semantics.

Presenting audio recordings at arbitrary rates can be challenging, however, and a surprising number of audio systems today do not support variable-speed audio playback. In our survey of existing audio devices, we have identified four possible feedback types for audio timeline navigation:

***None***: Systems that do not provide audio feedback while scrolling still provide a means to play the audio at its nominal rate (e.g., *play* button). While it may seem obvious that no feedback would result in poor audio scrolling performance, we include it as the baseline case for comparison. It is also common in existing systems – no feedback is given when *scrubbing* through audio (moving the playhead back and forth over a waveform visualization, often used to mark cut and trim points) using an iPod or Audacity, for example.

***Skipping***: A short segment of audio (tens of milliseconds) is played at regular speed when the playhead position is changed. This allows the user to experience feedback at arbitrary scroll rates without any pitch-shifting artifacts. The resulting audio is choppy, however. Many CD players and answering machines provide *skipping* feedback when the *fast-forward* and *rewind* buttons are held down. It is also common in video editors such as Final Cut Pro.

***Resampling***: The audio is *resampled* to allow playback at arbitrary rates. Resampling also pitch-shifts the audio; the effect is the same as varying the play rate of a vinyl record player. While disc jockeys (DJ's) make use of this feature for artistic effect, pitch shifts to the audio are typically undesirable as it makes the audio more difficult to comprehend. Adobe Audition supports this type of feedback for scrubbing as a separate mode ("tape-style" scrubbing).

***Time-stretching***: The audio is processed to allow playback at arbitrary rates without changing the pitch. The processing, unfortunately, introduces artifacts into the resulting audio. Algorithms such as waveform similarity overlap-add (WSOLA) [11] are efficient, but produce unsatisfactory results for polyphonic audio (e.g., orchestral music) and/or large stretch factors. The phase vocoder and its variants [6, 7], have been developed to address these limitations, but still exhibit "transient smearing" and "reverberation" artifacts. Arons' *SpeechSkimmer* [2], and Hürst et al.'s *Elastic Audio Slider* [4] use time-stretching.

To our knowledge, there is no existing analysis or empirical evaluation on the effects of using these various feedback types on audio navigation performance.

We performed a series of interviews and evaluations of these feedback types for audio editing tasks, with both professional and non-professional users. The aim was to collected qualitative feedback. Audio editing tasks primarily involve extracting segments of audio from a raw recording to, for example, produce a two minute program from a thirty minute interview. Audio editing software, if they provide any feedback at all during scrubbing, typically use *resampling* (also referred to as “tape-style” scrubbing). However, we found that such a feature is only useful for professional editors with experience cutting with physical reels of tape, as such interfaces are designed to reproduce this interaction. However, for non-professional editors, or editors without tape-cutting experience, the continuously varying pitch shifts render the audio incomprehensible.

We also found that each of these feedback types can benefit users performing specific types of audio editing tasks:

- Time-stretching should be employed for searching tasks where the play rate of the audio does not drop below roughly one-quarter nominal speed. Not only is audio time-stretched at extremely low rates disturbing to the user because of the artifacts it introduces – targeting performance can become *worse* than if no audio feedback was provided at all.
- For targeting tasks where the play rate frequently drops below roughly one-tenth nominal speed (e.g., when the user has zoomed far into the waveform for a precise cut), either *resampling* or *skipping* should be used for audio feedback. *Resampling* should be utilized for users with prior experience working with tape, as they have the ability to recognize certain cues more easily with audio shifted down in pitch. Most users, however, will prefer *skipping* feedback.

## DESIGN SPACE FOR AUDIO NAVIGATION TECHNIQUES

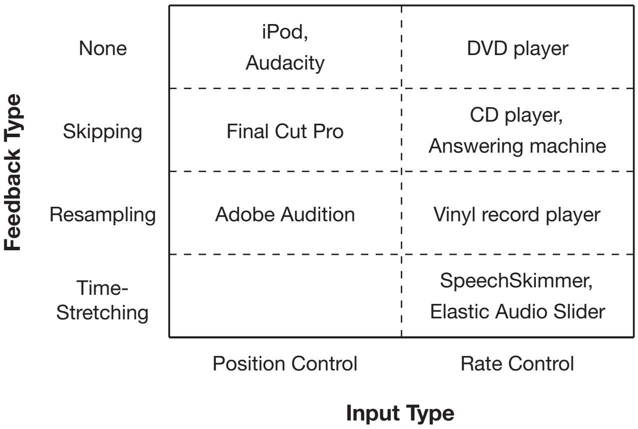
Based on the previous discussion, we now propose a design space for audio navigation techniques (see Figure 3). It consists of two orthogonal axes: input and feedback type. Input types are classified as position (also known as *zero order*, see [12]) or rate (*first order*). Higher order input methods, such as acceleration control (*second order*), have been previously demonstrated to be less efficient compared to zero and first order controls [10], and are thus less common, and we have not included these for the sake of brevity. The four feedback types are as mentioned previously: none, skipping, resampling and time-stretching.

Interestingly, there are no interfaces, to the best of our knowledge, that support position control with time-stretching feedback.

## DIMAß: DIRECT MANIPULATION AUDIO SCRUBBING

DiMaß fills the gap in our design space: it supports direct manipulation of an audio timeline using position control, with continuous, high-fidelity audio feedback. Details of the algorithm design and implementation are presented in [9], which we briefly summarize here.

DiMaß consists of three parts (see Figure 4). A motion estimator receives position events $p(t)$ from an input device such as a mouse and calculates the desired audio position $x(t)$, and velocity $v(t)$. These parameters are fed into an input tracker that computes an adjusted audio play rate $r(t)$. Finally, the audio is processed using PhaVoRIT, a time-stretching algorithm that preserves the original audio pitch with high-fidelity [6]. Unlike similar, existing time-stretching modules, PhaVoRIT supports arbitrary, including backwards, rates. After time-stretching, an updated audio position $a(t)$ is fed back to the input tracker to maintain precise audio to input synchronization.



| Feedback Type | Position Control | Rate Control |
|---|---|---|
| None | iPod,<br>Audacity | DVD player |
| Skipping | Final Cut Pro | CD player,<br>Answering machine |
| Resampling | Adobe Audition | Vinyl record player |
| Time-Stretching | | SpeechSkimmer,<br>Elastic Audio Slider |

**Figure 3. Design space for audio navigation techniques, populated with examples of existing devices.**

The improved synchronization algorithm is the key contribution of this work, and together with PhaVoRIT, enables the unique combination of position control together with time-stretching feedback. Our current implementation offers a “viscosity” parameter that allows for smoother playback at the expense of decreased responsiveness (for example, maximum responsiveness would result in choppy playback, similar to skipping).

We implemented DiMaß in an audio editor prototype (see Figure 5). In addition to the audio feedback while selecting an area on the waveform, a “toss” quasi-mode can be activated to allow the user to jump to distant parts of the waveform; such an interaction works especially well with a pen-and-tablet device and/or touchscreen.

## SUMMARY AND OUTLOOK FOR THE WORKSHOP

The results of our initial study on audio timeline navigation to date indicate that, despite the prevalence of rate control in current interfaces, position control is superior for closer targets. Further studies are currently being planned to characterize the hypothesized crossover effect for audio timeline navigation, analogous to the one Hinckley et al. observed for text navigation.

Addressing the problem of appropriate feedback for audio timeline navigation has unique challenges because of the temporal nature of audio. Our current results indicate that each of the feedback types (skipping, resampling and time-stretching) have their uses in audio editing, depending on the

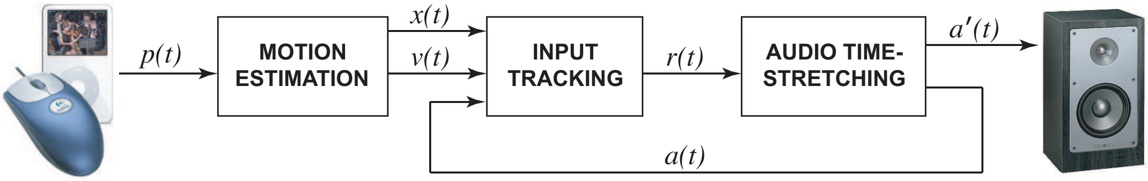


**Figure 4. DiMaß block diagram.**

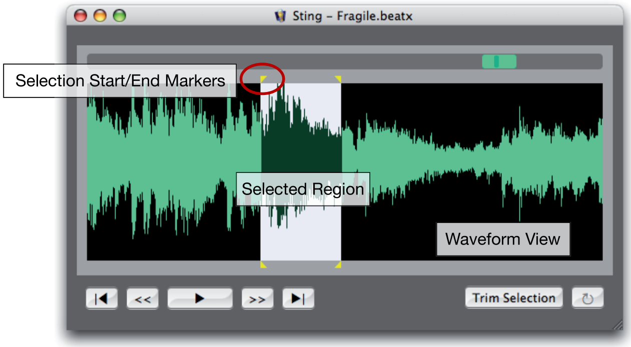


**Figure 5. DiMaß implemented as part of an audio editor. The user can modify the selection using the selection markers; as the cursor is "scrubbed" over the waveform, the audio at that position is played to the user.**

nature of the task (e.g., performing a rough cut of the audio, or zooming in and performing precise edits) and the experience of the user (professional vs. hobbyist). We are interested in exploring further possibilities for providing feedback (audio or otherwise) to users, as well as exploring other usage scenarios where specific feedback types are more appropriate than others.

As we continue this work, we are eager to share our experiences and results with the community, and solicit input on how our work could be incorporated in better supporting non-professionals working with digital audio and other time-based media. We envision, for example, to use the results of our work in the design of a "podcast editing" system. Such a system would employ a rate control with time-stretched feedback for scanning through material, and position control would be used for marking regions to be cut.

## REFERENCES

1. Apple. iTunes podcast subscriptions top one million in first two days. Press Release (June 2005).

2. Arons, B. SpeechSkimmer: a system for interactively skimming recorded speech. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *4*, 1 (1997), 3–38.

3. Hinckley, K., Cutrell, E., Bathiche, S., and Muss, T. Quantitative analysis of scrolling techniques. In *Proceedings of the CHI 2002 Conference on Human Factors in Computing Systems*. Minneapolis, USA, 2002, 65–72.

4. Hürst, W., Lauer, T., Bürfent, C., and Götz, G. Forward and backward speech skimming with the elastic audio slider. In *Proceedings of the 19th British HCI Group Annual Conference*. Edinburgh, Scotland, 2005.

5. Hürst, W. and Stiegeler, P. User interfaces for browsing and navigation of continuous multimedia data. In *Proceedings of NordiCHI 2002*. Århus, Denmark, 2002, 267–270.

6. Karrer, T., Lee, E., and Borchers, J. PhaVoRIT: A phase vocoder for real-time interactive time-stretching. In *Proceedings of the ICMC 2006 International Computer Music Conference*. ICMA, New Orleans, USA, 2006, 708–715.

7. Laroche, J. and Dolson, M. Improved phase vocoder time-scale modification of audio. *IEEE Transactions on Speech and Audio Processing*, *7*, 3 (1999), 323–332.

8. Lee, E. Towards a quantitative analysis of audio scrolling interfaces. In *Extended Abstracts of the CHI 2007 Conference on Human Factors in Computing Systems (Student Research Competition)*. San Jose, USA, 2007.

9. Lee, E. and Borchers, J. DiMaß: A technique for audio scrubbing and skimming using direct manipulation. In *Proceedings of AMCMM 2006 Audio and Music Computing for Multimedia Workshop*. Santa Barbara, USA, 2006.

10. Poulton, E. C. *Tracking skill and manual control*. Academic Press, New York, 1974.

11. Verhelst, W. and Roelands, M. An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech. In *Proceedings of the ICASSP 1993 International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1993, volume II, 554–557.

12. Zhai, S. *Human Performance in Six Degree of Freedom Input Control*. Ph.D. thesis, University of Toronto, Toronto, Canada (1995).

13. Zhai, S., Smith, B. A., and Selker, T. Improving browsing performance: A study of four input devices for scrolling and pointing tasks. In *Proceedings of INTERACT 1997 Conference on Human-Computer Interaction*. Sydney, Australia, 1997, 286–292.