Datalogisk Institut
KØBENHAVNS UNIVERSITET

DIKU

Ph.D. Dissertation

Interaction with Sound and Pre-Recorded Music: Novel Interfaces and Use Patterns

Tue Haste Andersen

Technical Report no. 05/05
ISSN: 0107-8283

HCØ Tryk
University of Copenhagen • Universitetsparken 1
DK-2100 Copenhagen • Denmark

Ph.D. Dissertation

Interaction with Sound and Pre-Recorded Music: Novel Interfaces and Use Patterns

by

Tue Haste Andersen

Department of Computer Science
Faculty of Science
University of Copenhagen
Denmark

June 2005

Abstract

Computers are changing the way sound and recorded music are listened to and used. The use of computers to playback music makes it possible to change and adapt music to different usage situations in ways that were not possible with analog sound equipment. In this thesis, interaction with pre-recorded music is investigated using prototypes and user studies.

First, different interfaces for browsing music on consumer or mobile devices were compared. It was found that the choice of input controller, mapping and auditory feedback influences how the music was searched and how the interfaces were perceived. Search performance was not affected by the tested interfaces. Based on this study, several ideas for the future design of music browsing interfaces were proposed. Indications that search time depends linearly on distance to target were observed and examined in a related study where a movement time model for searching in a text document using scrolling was developed.

Second, work practices of professional disc jockeys (DJs) were studied and a new design for digital DJing was proposed and tested. Strong indications were found that the use of beat information could reduce the DJ's cognitive workload while maintaining flexibility during the musical performance. A system for automatic beat extraction was designed based on an evaluation of a number of perceptually important parameters extracted from audio signals.

Finally, auditory feedback in pen-gesture interfaces was investigated through a series of informal and formal experiments. The experiments point to several general rules of auditory feedback in pen-gesture interfaces: a few simple functions are easy to achieve, gaining further performance and learning advantage is difficult, the gesture set and its computerized recognizer can be designed to minimize visual dependence, and positive emotional or aesthetic response can be achieved using musical auditory feedback.

Acknowledgments

This Ph.D. study was undertaken in three years, starting in May 2002. It was funded by the Faculty of Science, University of Copenhagen, Denmark and was primarily carried out at Department of Computer Science, University of Copenhagen (DIKU). I am thankful that the Faculty of Science sponsored my Ph.D. study. I have learned a great deal about interaction design and research in general.

During my Ph.D. I have had the privilege to experience daily discussions with my supervisor Kristoffer Jensen who provided invaluable. I have learned from his gifted insight in musical signal processing. Kristoffer was always a steady support throughout my study. Together, we extended the work initiated during my Master's thesis and we investigated the role of different perceptual parameters for use in a real-time beat track system. Personally, you have been a great friend; I hope we will keep in touch and continue to collaborate.

I efforts to further my work, I focused on Human Computer Interaction (HCI). The HCI group at DIKU, with Erik Frøkjær, has been very supportive and together with Kasper Hornbæk provided valuable input into my initial explorations in HCI. They guided me, and pointed me in the direction of professionalism and outstanding research in this area. I am truly thankful for what you have done for me. Georg Strøm has also been very helpful and has given me valuable feedback on my work, especially in the last year of my research.

I got valuable feedback from my support group; Staffan Björk, Steffen Brandorff and Jens Arnspang. During my daily activities at DIKU, many people besides those already mentioned, were always enthusiastic and provided a positive environment, especially Kenny Erleben, Declan Murphy and August Engkilde.

I participated in many conferences and workshops. Two have been especially influential upon me: the summer school on Ubiquitous Computing at Dagstuhl in 2002, and the doctoral consortium at NordiCHI 2002. While at Dagstuhl I received a great deal of inspiration as well as valuable feedback and obtained contacts at NordiCHI. In particular, at NordiCHI I got the chance to meet Shumin Zhai, whom I later visited as part of my Ph.D. research. In Spring, 2004, while visiting Shumin at IBM Almaden Research Center I had the privilege of conducting a research project under his supervision. What I learned from Shumin was a throughout and systematic approach to investigation of interaction techniques and general research methodology. Shumin was a tremendous resource; we had daily discussions that greatly inspired me. I am in debited to you Shumin, you have taught me so much about science, and proved yourself a great friend; I hope we will continue our collaboration in the future. At IBM everybody was helpful; I have never experienced such welcoming and open people as the people I met there. Shumin Zhai and Pernilla Qvarfordt, provided a stimulating work atmosphere, however I also have great memories of our trips in the countryside and skiing at Lake Tahoe. I had many conversations with Per-Ola Kristensson about work, HCI and life in Scandinavia. I gained valuable insight from Christopher Campbell, Malcolm Slaney and Barton Smith; you all made me feel welcome, especially Tamara Lo and Christopher Campbell who introduced me to the Californian lifestyle with great food, movies and surfing in Santa Cruz. Thank you all.

Finally, thanks to my wife Pernille Haste Andersen for her endless support, especially while we were thousands of kilometers away from each other; and to the rest of my family for their support and advice in life.

Contents

Chapter 1

Introduction

Many aspects of music have changed over the last century: the music itself has changed, new musical instruments are constantly appearing and mobile devices change the way music is listened to and used. In this thesis, various new ways to interact with music through computers are investigated.

Recordings of music made the music independent of the performing musicians. Music could be listened to in a radically different setting from where it was recorded. The use of recorded sound in composition and performance was first introduced by the French musician Pierre Schaeffer in the 1940'ies with the musique concrete, and later brought into popular music by DJs in the 60'ies and 70'ies by use of turntables as instruments in musical performance.

As computers became more powerful, they could be used in musical performance. Today, in academia new interfaces for musical expression are explored and presented where the focus is to create interfaces for musicians. This area is especially relevant to research in Human Computer Interaction (HCI) since musical performance places increased demands on the musician in terms of cognitive workload and cooperation with other musicians. Designing computer systems that will aid in musical performance is thus a challenging area from which we can learn general principles that can benefit other areas of HCI (Buxton 1995).

Not only are the musicians benefiting from the computers; musical file playback capabilities have improved tremendously with media players that allow for visual feedback and tools for organizing the music. Music distribution over the Internet allows ordinary users access to vast amounts of music.

As computing becomes ubiquitous (Weiser 1991) and move to mobile platforms, new ways of interacting and using music can be envisioned. An example is the coupling of music playback with common interaction tasks such as text input (Andersen and Zhai 2005b), where the playback is changed depending on the writing patterns of the user. Another example was recently demonstrated by Siemens that coupled music playback with running speed in a mobile device (Gizmodo 2005). This approach of coupling musical playback with information from the environment could potentially improve aesthetic aspects of our life without degrading the performance of our work.

1.1 Feedback and mappings

Ancient philosophers called the human senses “the windows of the soul,” (Encyclopædia Britannica 2005) as they are used to observe and communicate. The primary senses are probably visual and auditory; however, kinesthetic and olfactory are among our senses also used in communication. When interacting with computers we can benefit from using several senses and thus several modalities. This principle was first demonstrated by Bolt with the “Put That There” system where pointing and speech were used as a combined modality (Bolt 1980). Multimodal interaction with computers can be divided into input and feedback where Bolts system is an example of multimodal input. Research in multimodal interaction has focused mostly on input (Oviatt 2002) but recently multimodal feedback has received more attention. Multimodal feedback is about presenting information to the user through different channels. The goal of providing feedback through more than one feedback channel is to reduce workload of the user and allow for a broader set of usage situations, for instance operation of computers in a heads-up situation.

Today, interacting with computers is dominated by the desktop metaphor based on the WIMP (Windows, Icons, Menus and Pointer) paradigm. WIMP interfaces are primarily based on visual feedback. Transferring objects and tasks from the natural environment to the computer includes the notion of space and objects, metaphors that are easily conveyed through visual information (Deatherage 1972). Graphical interfaces have evolved greatly in the last two decades. Adding advanced graphical visualization techniques to an application is possible with little effort by the programmer through the use of graphical widgets and toolkits. Haptic and auditory toolkits have only marginally improved in the same period. As a result, auditory and haptic feedback have received less attention. However, auditory and haptic feedback promises to deliver great advances in interaction technology. Designing for ubiquitous computing (Weiser 1991) often implies a mobile usage situation where the visual channel is overloaded and cognitive workload is high. Presenting information through non-visual modalities could reduce workload and allow for heads-up operation.

As improvement in interaction is possible by adding input or feedback channels to an interface, another possibility is to reorganize the information presented through a particular sense or channel. For instance, a map may be easier to read if displayed with information about the users’ current location. In this way, adding or removing information from the interface could in some situations be more effective ways to improve interaction compared to adding new modalities.

When designing interaction based on adding or removing feedback, the mapping between task and feedback channel is important. In learning psychology feedback is divided into inherent and augmented feedback. Inherent feedback is feedback that is tightly coupled to the task, e.g. the sound resulting from whistling. On the other hand, augmented feedback is supplementary to the task, e.g. a waveform display showing the energy level in a sound file is feedback that augments the inherent auditory feedback.

Here feedback is divided in two based on its mapping to the feedback channel:

1. Natural feedback. Feedback that is inherent and mapped to a natural feedback channel. For instance, when interacting with a spatial map, the natural mapping is to a modality using the visual channel. When listening to music, a natural mapping exists between the music and the auditory channel.

2. Artificial feedback. Feedback that is either inherent or augmented and mapped to a feedback channel different from the natural feedback channel. An example is augmenting a map with an auditory display to allow navigation without looking at the map. While such a mapping may be possible and useful it will be difficult to design using an ecological approach (Gaver 1993). An artificial or abstract mapping between the (non-verbal) sound and the map would have to be established.

In working with multimodal feedback, some of the major problems include how artificial feedback can be added and how natural feedback can be changed to improve interaction.

In the following, research projects of this thesis are summarized with respect to feedback mapping.

Three aspects of improving feedback are explored, all related to pre-recorded music: First, interfaces for browsing and searching in music are examined. Type of audio feedback and input control are examined in an experiment. This is an example of using natural feedback, auditory feedback mapped to music playback. In the experiment, evaluation is done based on trying to improve the presentation of the existing modality. Second, DJ work practices and a digital DJ solution are studied. Visual displays of audio were designed to augment the music playback and aid the DJ in performing certain tasks. This is an example of augmenting the natural feedback, the music presented through audio, with artificial feedback, a graphical representation of the music. In this study, adding meta-data that describes the rhythmic structure and beat of the music is also used to improve interaction. Finally, the coupling of sound and music with pen-gestures are explored to evaluate the influence of sound on performance, learning and aesthetics of pen-gesture interfaces. A pen-gesture interface is inherently spatial since the gestures are symbols with a spatial extent. Adding auditory feedback to this interface thus requires an abstract mapping between the gestures and the audio.

The papers presented in this thesis can be placed into a framework of feedback as shown in Table 1.1. Music search refers to (Andersen 2005b), DJ tools to (Andersen 2005a), Writing with Music to (Andersen and Zhai 2005b), ELEW to (Andersen and Zhai 2005a) and Scrolling to (Andersen 2005c).

1.2 Benefits

The benefits of coupling interaction with sound and music are significant. Mobile playback devices offer a vast amount of digital music; however, ways to efficiently browse and navigate the music collection are needed. Knowledge of

	Auditory	Visual/Spatial
Natural	Music search DJ tools	Writing with Music ELEW Scrolling
Artificial	Writing with Music ELEW	Music search DJ tools

Table 1.1: HCI projects divided by type of feedback.

usage situations and behavior patterns may help to construct new and better interfaces that are more efficient and enjoyable to use.

In the pursuit of mobile and ubiquitous computing there is an increasing need for multimodal interfaces, interfaces that allows for communication using more than one modality. Here, sound and music could play an important role to input and present information through the auditory channel (Gaver 1989). Based on this perspective of the interaction, not only performance and learning may be relevant places to gain benefits, but also the increased focus on factors such as joy (Blythe, Monk, Overbeeke, and Wright 2003) and aesthetics (Norman 2004) in HCI is highly relevant when working with music.

For professional musicians new tools have the potential to reduce cognitive workload and provide new input to the creative process of composition and performance, allowing for a wider range of creativity than existing analogue and computerized tools provide. The development of such tools may also benefit amateurs and ordinary computer users that may find it possible to compose their own music with limited or no musical skills.

1.3 Research methodology

The research methodology followed has been based on empirical studies. The projects I worked on were based on literature studies and prototype development, followed by user studies in a laboratory setting. For one project I also studied work practices of the existing usage situation, and in another I worked on developing a theoretical framework.

1.4 Contributions

During the Ph.D. I have made a number of contributions to the scientific community in form of papers, talks, demonstrations, and freely accessible software. The publications, including papers that are currently under review, are:

- K. Jensen and T. H. Andersen. Real-Time Beat Estimation Using Feature Extraction. In Computer Music Modeling and Retrieval: International Symposium, Montpellier, Lecture Notes in Computer Science, pages 13-22. Springer Verlag, 2003.
- D. Murphy, T. H. Andersen, and K. Jensen. Conducting Audio Files via Computer Vision. In Gesture-Based Communication in Human-Computer Interaction: 5th International Gesture Workshop, Genova, Lecture Notes in Computer Science, pages 529-540, April 2003.
- T. H. Andersen. Mixxx: Towards Novel DJ Interfaces. In Proceedings of the New Interfaces for Musical Expression (NIME'03) conference, Montreal, May 2003. Design report.
- K. Jensen and T. H. Andersen. Beat Estimation on the Beat. In Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New York, October 2003.
- T. H. Andersen and K. Jensen. Importance and representation of phase in the Sinusoidal Model. Journal of the Audio Engineering Society, 52(11): 1157-1169, November 2004.
- T. H. Andersen. A Simple Movement Time Model for Scrolling. In Proceedings of CHI 2005, pages 1180-1183. ACM Press, April 2005. Late breaking result.
- 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.
- T. H. Andersen. Searching for music: How Feedback and Input-Control Change the Way We Search. In Proceedings of Interact 2005. Springer Verlag, September 2005. To appear.
- T. H. Andersen. DJing with Dance Music: The Design and Evaluation of a Digital DJ System. 12 pages. Under review.
- T. H. Andersen and S. Zhai. Robust Pen-Gesture Recognition with Low Visual Attention Demand — The design of the ELEW Character Input System. 4 pages. Under review.
- T. H. Andersen and S. Zhai. “Writing with Music:” Exploring the Use of Auditory Feedback in Pen Gesture Interfaces. 24 pages. Under review.

In many of these publications, the software Mixxx[1] was used. Mixxx is an open source project I developed during the past three years designed for DJing with digital sound files. Mixxx facilitates music information retrieval, audio navigation, logging of user interaction, easy configuration of user interface and

[1] See `http://mixxx.sourceforge.net/`

flexible mappings to GUI and external controllers. I am the primary developer of this project, but various other developers and DJs contributed with code and highlighting important design issues. The software is used by researchers and DJs and has been downloaded from SourceForge in more than 50.000 copies.

In working with pen-gesture interfaces, I developed an open source driver, JWacom[2] for the Wacom tables, to be used with Java under Linux.

Of the publications listed above, six papers constitute the main body of this dissertation. I have chosen to divide these papers into three areas that each describes an aspect of interaction with pre-recorded music or sound. The contributions of these three areas are summarized below:

1D navigation in music and text. One short paper (Andersen 2005c) presents a new movement time model that accounts for the time spent when scrolling and searching for a specific feature in a text document. The model supplements a previous model for scrolling (Hinckley, Cutrell, Bathiche, and Muss 2002), and is especially important because it shows how perception, in addition to motor control, can play an important role in modeling human performance. The second paper (Andersen 2005b) describes a study on searching in sound, again using a simple means of input control. A new way of synthesizing sound to give a better representation of music during a fast forward operation is presented and compared to standard synthesis types. Also two different types of input control are compared and evaluated. The results can serve as a foundation to improve search interfaces in mobile and home media players.

Musical performance with pre-recorded music. One of the first studies I did was on the implementation and refinement of a musical beat tracking system (Jensen and Andersen 2003). Several auditory features and signal analysis parameters were compared and evaluated using hand annotated musical data. We found that using a one dimensional feature weighing the high frequency content was especially useful for estimating the beat of popular music. Automatic beat estimation was implemented in the Mixxx DJ software to aid in certain DJing tasks. During exploratory studies of DJs using Mixxx it was found that an automatic beat matching system was unlikely to provide the required precision to ease their workload. Instead, based on studies of DJ work practices, hand annotated sound files were used to aid in the task of beat matching. We found unique data on DJ work practices, and conducted an initial lab study of the use of digital DJ software (Andersen 2005a). The paper on digital DJing (Andersen 2005a) contributes an important understanding to DJ work practices; based on these findings a new interface was implemented in Mixxx. Two interface mappings were tested by experienced DJs. In general, the DJs preferred the interface that provided automatic beat synchronization over the interface that was more similar to the traditional DJ setup.

[2] See `http://www.diku.dk/~haste/jwacom/`

Feedback in pen-gesture interfaces. This study was done in collaboration with Shumin Zhai and was initiated to explore how audio feedback could be used with pen-gesture input systems to aid in either learning or performance (Andersen and Zhai 2005b). We compared auditory feedback with visual feedback, and found that neither contributed significantly to shape and direction aspects of the quality of writing. We followed with a second study evaluating aesthetic and subjective factors in a study where different auditory feedback was given for different conditions. Music playback coupled to the average writing speed was the most engaging (Andersen and Zhai 2005b). Later, we explored the idea of refining the gesture recognizer rather than the feedback to achieve a robust pen-gesture input system for heads-up use (Andersen and Zhai 2005a).

1.5 Thesis outline

The rest of this thesis contains a chapter for each area investigated:

- 1D navigation in music and text
- Musical performance with pre-recorded music
- Feedback in pen-gesture interfaces

Each chapter contains two papers. This is followed by a general thesis conclusion in Chapter 5.

Chapter 2

1D navigation in text and music

2.1 A simple movement time model for scrolling

Appeared in:
T. H. Andersen. A simple movement time model for scrolling. In Proceedings of CHI 2005, pages 1180-1183. ACM Press, April 2005. Late breaking result.

A Simple Movement Time Model for Scrolling

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

ABSTRACT

A model for movement time for scrolling is developed and verified experimentally. It is hypothesized that the maximum scroll speed is a constant at which the target can be perceived when scrolling over the screen. In an experiment where distance to target and target width were varied, it was found that movement time did not follow Fitts' law. Rather, it was linearly dependent on the distance to the target, suggesting a constant maximum scrolling speed. We hypothesize that the same relationship between movement time and target distance apply to other computer interaction tasks where the position of the target is not known ahead of time, and the data in which the target is sought is not ordered.

Author Keywords

Scrolling, movement time model, Fitts' law.

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation: User interfaces.

INTRODUCTION

The design of the computer desktop is among other things guided by quantitative models of movement time. “Laws of action” have been discovered that accurately describes the average movement time in a number of interaction situations, such as pointing and steering [2]. Scrolling is another widely used interaction technique [13], but a satisfactory and widely accepted model has yet been developed that accounts for the time used in the general case of scrolling. Scrolling is important to both the classical desktop WIMP interfaces, but also as computing moves to new areas such as mobile computing, where display size is limited. An example is the PDA that today offer applications for people on the move, but with significant less screen real estate.

A model predicting movement time is especially relevant in the design and evaluation of input techniques for scrolling. With a model that adequately accounts for the time spend in scrolling, it is possible to compare techniques across varying experimental conditions, which otherwise would not have been possible.

CHI 2005, April 2–7, 2005, Portland, Oregon, USA.

ACM 1-59593-002-7/05/0004.

Scrolling experiments to date have been focused on target acquisition [13, 5], effectively turning scrolling into an aimed movement paradigm just like pointing. This paradigm focuses on only the last part of the scrolling process. In practice the process of controlling scrolling for the purpose of finding the target in motion could be just as important an aspect as capturing the target upon finding it.

Hinckley and colleagues [5] have shown that certain scrolling patterns can be modeled by Fitts' law. In this paper we build on their work by investigating movement time when the target position is not known ahead of time. We question the applicability of Fitts' law in the general task of scrolling and other tasks. Fitts' law is developed for “aimed” movement whose velocity profile is typical a bell curve (accelerate and decelerate) with no speed plateau [1]. We believe that scrolling in unsorted data where the target position is not known, is not only dependent on the time it takes to physically move a pointer or other means of control as Fitts' law predicts, but is also limited by human perception, specifically visual scanning. Visual scanning limits the maximum speed beyond which one can keep track of the visual content for scrolling purpose, and thus we expect to observe a speed plateau in the velocity profile of scrolling.

Our initial exploration includes studies of the following factors: distance to target, target width and control/display ratio. An experiment is presented where the movement time is observed for the common task of scrolling in text documents. Distance to target and target width was varied, and the results were evaluated using quantitative data analysis.

SCROLLING IN ONE DIMENSION

To develop a movement time model, we study the simplest case of scrolling, scrolling in one dimension, to establish a base relationship. Scrolling in one dimension is most commonly done in text, but also scrolling in the time dimension of audio and video is increasingly common for both desktop and mobile users. In the work presented here we focus on scrolling in text.

Scrolling techniques and devices

Techniques for scrolling in one dimension can be classified by the type of mapping used. Most common is position based mappings, such as the mouse wheel and the scroll bar. Alternative to position based mappings are mappings that employ a higher order transfer function. An example of such a mapping is used in the Scrollpoint [13], where a small isometric joystick controls the rate at which the document is moved. The advantage with direct control of the rate is that the user has the ability to scroll very fast, and at the same time the precision needed to reach an exact point regardless of document length. Recently, several novel techniques has been proposed using circular motion [9, 10].

In practice the aforementioned techniques are often combined into a single pointing device, making the device *multi-stream* [13], such as the wheel mouse or Scrollpoint mouse.

Human motor capacity

Interaction techniques such as pointing and scrolling have been evaluated and modeled using Fitts' law of aimed movement. Fitts' law predicts the movement time T to a target with distance D, also called the amplitude of movement, and target width W in one dimension. In HCI it is often formulated as [8]:

$$T = a + b \log_2(D/W + 1) \qquad \text{(Equation 1)}$$

where a and b are empirically determined constants. The logarithmic part of Equation 1 is often defined as Index of Difficulty (ID). In Fitts' original experiment on aimed movement, movement time was measured between tapping two visible targets with a pen. The task was performed in succession over a time duration, and the average movement time was used in evaluating the human motor capacity. A similar setup was used to evaluate movement time in scrolling by Hinckley et al. [5]. In that experiment participants repeatedly scrolled up and down between two target lines as fast as possible. The participants knew the target position, and thus they did not have to rely solely on visual feedback to find the target. What Hinckley effectively measured was the performance of the input device used in scrolling, minimizing possible effects caused by the feedback during scrolling. In studying the applicability of Fitts' law in HCI, MacKenzie [8] notes that Fitts' law is likely to be usable only in situations with negligible or known preparation time. However, in many tasks it is not only the mental effort used in preparing the task, but also during the task that can limit the applicability of Fitts' law. An interaction technique can be limited by the human motor capacity, but also by perception and cognition.

Another issue is the effect of control movement scale on input. Control movement scale was formally studied in the steering law [2] where it was found that performance for tasks with same ID but different movement scales did result in different completion time, but the difference was relatively small compared to the impact of ID change. In the preliminary exploration phase of this work a number of experiments was carried out with few participants where control-display ratio was varied, using either a relative or absolute mapping. We did not observe any significant effect on movement time. This result is in agreement with [2] since the effect is relatively small compared to varying the ID.

Feedback

When scrolling text, the primary form of feedback is visual. The feedback can appear as a smooth scroll when scrolling at lower speed, or as smeared or blurred when scrolling at high speed.

To avoid blurring when scrolling at high speed, Igarashi et al. [6] devised a technique that continuously adapted the zoom factor during scrolling, depending on the speed of scrolling. At high speed, the text was zoomed out, so that the scrolling speed remained constant. The technique makes it possible to maintain overall orientation in the document when scrolling at high speed, but reading the text or other means of perceiving the content at a higher level of detail is not possible. The technique was in the study evaluated to be equally effective when compared to traditional scrolling, but the participants liked the new technique better.

Other examples of helping to maintain perception of position in the document has successfully been applied using additional visual feedback [12, 7].

Visual scanning of static data (non moving) is either done attentively or pre-attentively. When scanning for words or sentences that do not have outstanding typographic features the scanning will be attentive and depend linearly on the number of distractors [11]. For moving text, the visual scanning time is likely to be the limiting factor in movement time. If the features that are sought can be scanned pre-attentively, such as a line with a different color than the rest of the text, the scanning time is independent of the number of distractors. However, in scrolling only a portion of the whole document is shown at a time and thus the speed at which the document can be moved and the target line seen is likely to be a limiting factor here too.

A simple model

We propose a linear model for movement time T in scrolling:

$$\mathrm{T} = a + b\, D \qquad \text{(Equation 2)}$$

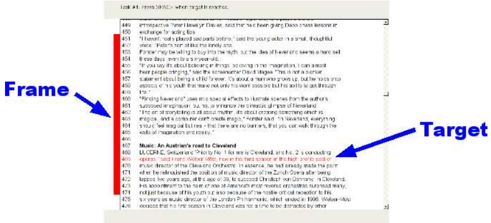
where a and b are empirically determined constants, and D is distance to target. T is limited by the maximum rate at which the target can be perceived when scrolling over the screen. The constants a and b accounts for the maximum scrolling rate and will depend on factors such as input device, control-display ratio, display size and type of target.

EXPERIMENT

In the experiment we use a setup similar to that used by Hinckley et al. [5], but with an important difference. The user has to scroll until a target line is reached, but the position of the target line is unknown at the beginning of the trial. When the target line becomes visible the user will adjust the target position to be within the required range and press a button.

Design

We ran the experiment on a computer sufficiently fast to avoid latency and non-smooth scrolling. We used a 21 inch CRT screen, with a text window 847 pixels wide, and 600



Task #1: Press SPACE when target is reached.

449 introspective Peter Llewelyn Davies, said that he'd been giving Depp chess lessons in
450 exchange for acting tips.
451 "I haven't really played sad parts before," said the young actor in a small, thoughtful
452 voice. "Peter's sort of like the lonely one."
453 Forster may be willing to buy into the myth, but the idea of Neverland seems a hard sell
454 these days, even to a six-year-old.
455 "If you say it's about believing in things, believing in the imagination, I can almost
456 hear people cringing," said the screenwriter David Magee. "This is not a blanket
457 statement about being a child forever. It's about a man who grows up, but he holds onto
458 aspects of his youth that make not only his work possible but his ability to get through
459 life."
460 "Finding Neverland" uses mild special effects to illustrate scenes from the author's
461 supposed imagination, but not to enhance the climactic glimpse of Neverland.
462 "The art of storytelling is all about myth, it's about creating something which is
463 magical, and a computer can't create magic," Forster said. "In Neverland, everything
464 should feel magical but real - that there are no barriers, that you can walk through the
465 walls of imagination and reality."
466
467 **Music: An Austrian's road to Cleveland**
468 LUCERNE, Switzerland "Priority No. 1 for me is Cleveland, and No. 2 is conducting
469 operas," said Franz Welser-Möst, now in his third season in the high-profile post of
470 music director of the Cleveland Orchestra. In essence, he had already made the point
471 when he relinquished the position of music director of the Zurich Opera after being
472 tapped five years ago, at the age of 39, to succeed Christoph von Dohnányi in Cleveland.
473 His appointment to the helm of one of America's most revered orchestras surprised many,
474 not just because of his youth but also because of the hostile critical reception to his
475 six years as music director of the London Philharmonic, which ended in 1996. Welser-Möst
476 decided that his first season in Cleveland was not a time to be distracted by other

Figure 1. Screenshot of running experiment.

pixels high. Each line was 21 pixels high, and the length of the document used was 1200 lines. The smallest possible movement of the scroll bar corresponds to exact 2 lines in the text. Figure 1 shows a screenshot.

An optical mouse was used as input technique directly mapped to the handle of the scrollbar. The mouse cursor was not displayed, and the user did not have to click the handle before moving the mouse. This mapping was chosen to avoid the delay needed for users to point and click the scroll bar handle, and to avoid the users interacting with the scrollbar in other ways. We chose to use an ordinary optical mouse with a scroll bar and ordinary text display as input technique, because this is a commonly used technique for scrolling documents and because the focus of this paper is not on device/technique comparison.

The text document used was a collection of newspaper articles from the culture section of International Herald Tribune. To the left of each line was displayed a line number. The articles were marked with an empty new line and the heading displayed in Bold. The target line was marked by red text. To ensure a certain uniformity of the target lines, all target lines were at least as long as the average line length used in the document and no target line was a heading, i.e. marked in bold.

The goal was to find the target line, and position it within the range specified by the frame, a red vertical bar on the left of the text display. The tolerance defined by the frame is called target width.

The choice of marking the target with red, was partly because it is identical to the method used in [5], partly because it is a method that can be expected to interfere less with reading speed and higher order cognitive processes compared to searching for a specific word or sentence. The red line was relatively easy to identify when scrolling through the document, but still could be missed. The marking of the target is artificial in the sense that it is not a common task to search for red lines in a document. The result we obtain from the experiment is highly dependent on the type of target we use. However, since our hypothesis is that the scrolling speed is dependent on the maximum perceivable rate of the target, the relationship is not changed by the type of target, only the maximum scrolling speed.

In choosing a target more close to a natural task, such as finding a specific sentence or description, we might however observe other interactions. For example the scrolling direction is likely to influence the maximum scrolling speed, because it can be expected that people read faster forward than backwards.

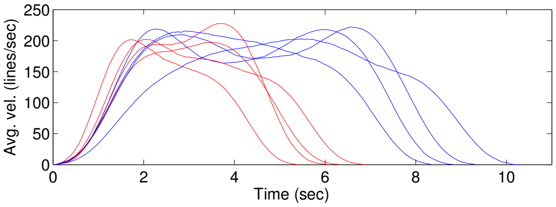


Figure 2. Average velocity as a function of time plotted for 1 subject in 8 trials using two target distances. *D*=673 lines (red), *D*=1109 (blue).

Procedure

A fully-crossed, within-subjects factorial design with repeated measures was used. Twelve subjects participated in the experiment. Independent variables were the target distance (amplitude of movement, D=16, 235, 454, 673, 892, 1109 lines), width (W=4, 12 and 24 lines), and starting point (top or bottom). In the condition with a target width of 4 lines, there were only two valid positions for the target line to reach the target. A target of 24 lines, is close the the screen length of 28.6 lines. Each condition was repeated twice. Starting point was varied by alternating the starting point to be at the top or the bottom of the document. Target distance and width was ensured to be repeated twice in each start position; otherwise the presentation order of these two variables was chosen randomly.

Analysis and Results

The dependent variables we used were movement time and error. In Figure 2 example velocity profiles are shown for two target distances. It can be seen that the speed does not follow a bell curve, but it reaches a plateau as predicted.

Figure 3 shows a correlation of movement time as a function of target distance. A linear fit is done for all data points, with a correlation of 0.97 (*a*=4573.6 msec, *b*=9.6 msec/line). This suggests that indeed there exist a tight linear relationship between target distance and average movement time. In comparison the correlation between the data and linear regression of movement time as a function of ID is 0.72 (Figure 4). A significant difference in movement time was observed for target distance ($F_{5,55}$=13.90, p<0.001), and for target width ($F_{2,22}$=8.13, p=0.002). Pairwise comparison showed significant difference between target width of 4 lines and the two other width conditions (p<=0.021), but no other significant differences. The mean movement time for each width condition is shown in figure 5. No significant differences in movement time across start position ($F_{1,11}$=3.641, p=0.083) was observed. There were no significant difference in number of errors across target distance ($F_{5,55}$=0.53, p=0.756), width ($F_{2,22}$=2.10, p=0.146) or start position ($F_{1,11}$=0.044, p=0.838).

Target width only slightly affected the results of the experiment but is likely to play a role in scrolling, as shown by Hinckley. The total scrolling time can be formulated as bound by at least two factors: the time used in visual

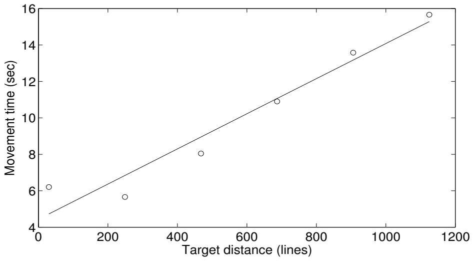


Figure 3. Linear regression of target distance against movement time.

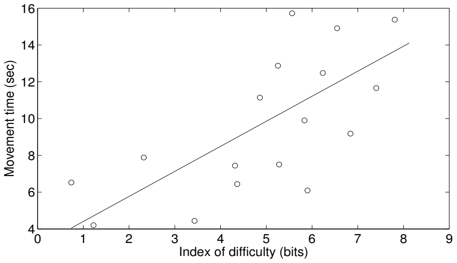


Figure 4. Linear regression of ID against movement time.

scanning and the time used to perform aimed movements. In the beginning and ending phases of a scrolling action, the movement time is likely to be bound by human motor performance, i.e. in the task of speeding up and slowing down to reach the right point. Between the starting and ending phases of the scrolling, the visual scanning seems to be the limiting factor rather than the motor performance as measured through an input device. Given a different search task where the target cannot be found by pre-attentive scanning, we might expect the movement time to be significantly higher than what is observed in this experiment, caused by the limit in scanning speed rather than motor performance. Exact how much time is used in each phase, and how movement time can be described in the starting and ending phases of the scrolling movement should be a subject of further investigation.

CONCLUSION

We have examined the relationship between movement time and scrolling in a text document. We found that for a task that more closely resembles scrolling than what Hinckley and other have presented, we find a tight linear relationship between movement time and distance to target, with a correlation coefficient of 0.97. In comparison the correlation coefficient between observed data and a regression based on Fitts' law is only 0.72.

Applying this to the design of scrolling techniques is especially important. Evaluating scrolling techniques using an experimental setup similar to the one used here, ensures that not only the input device is evaluated but also the type of feedback, which is an equal important part of scrolling. The proposed model is likely to apply to other interaction tasks where the target position is not known ahead of time, and the data in which the target is sought is not ordered.

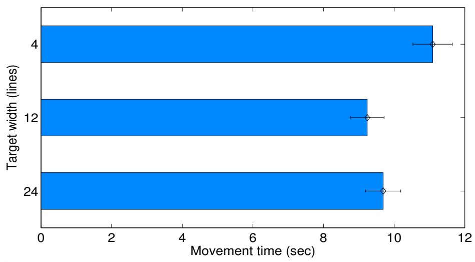


Figure 5. Mean movement time for different target width.

Looking at input devices in isolation, the relationship between movement time and target distance is in favor of rate based scrolling techniques, where an optimal scroll rate can be held constant without any limb movement as opposed to position based devices that requires continuous adjustment to maintain a constant rate.

ACKNOWLEDGEMENT

Thanks to Shumin Zhai and Kasper Hornbæk for invaluable discussions and feedback.

REFERENCES

1. Abend, W., Bizzi, E., & Morasso, P. Human arm trajectory formation. *Brain*, 105 (1982), 331-348.

2. Acott, J. and Zhai, S. Beyond Fitts' law: Models for Trajectory-Based HCI Tasks. In *Proc. CHI 1997*, ACM Press, 295-302.

3. Acott, J. and Zhai, S. Scale Effects in Steering Law Tasks. In *Proc. CHI 2001*, ACM Press, 295-302.

4. Fitts, P.M. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47 (1954), 381-391.

5. Hinckley, K., Cutrell, E., Bathiche, S. and Muss, T. Quantitative Analysis of Scrolling Techniques. In *Proc. UIST 2002*, ACM Press, 65-72.

6. Igarashi, T. and Hinckley, K. Speed-dependent Automatic Zooming for Browsing Large Documents. In *Proc. UIST 2000*, ACM Press, 139-148.

7. Kaptelinin, V., Mäntylä, T. and Åström, J. Transient Visual Cues for Scrolling: An emperical Study. In *Ajunct Proc. CHI 2002*, ACM Press, 620-621.

8. MacKenzie, I. S. Movement time prediction in human-computer interfaces. In *Proc. Graphics Interface* 1992, Canadian Inf. Proc. Society , 140-150.

9. Moscovich, T., Hughes, J. F. Navigating Documents with the Virtual Scroll Ring. In *Proc. UIST* 2004, 57-60.

10. Smith, G.M. The Radial Scroll Tool: Scrolling support for... In *Proc. Of UIST 2004*, ACM Press, 53-56.

11. Ware, C. Information Visualization. (2000). Morgan Kaufmann Publishers.

12. Wirth, M. and Zhai, S. Visual scrolling feedback and method of achieving the same. 2000. US patent 6,476,83.

13. Zhai, S., Smith, B.A. and Selker, T. Improving Browsing Performance: A Study of Four Input Devices. In *Proc. INTERACT* 1997, 286-292

2.2 Searching for music: How feedback and input-control change the way we search

To appear in:
T. H. Andersen. Searching for music: How feedback and input-control change the way we search. In Proceedings of Interact 2005. Springer Verlag, September 2005.

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.

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. 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.

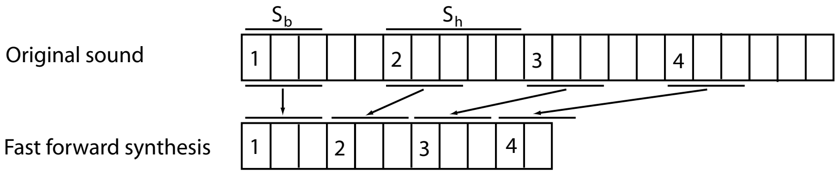


Fig. 1. Figure showing sound synthesis for fast forward as implemented in standard CD players. Blocks of samples (S_b) are copied from the original sound to the fast forward synthesis sound. The hop size (S_h) determines the scale factor.

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

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

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 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.

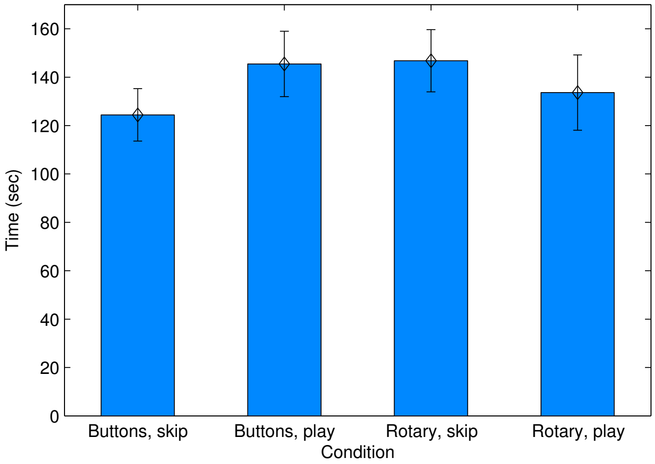


Fig. 2. Mean time used in each condition.

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.

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.

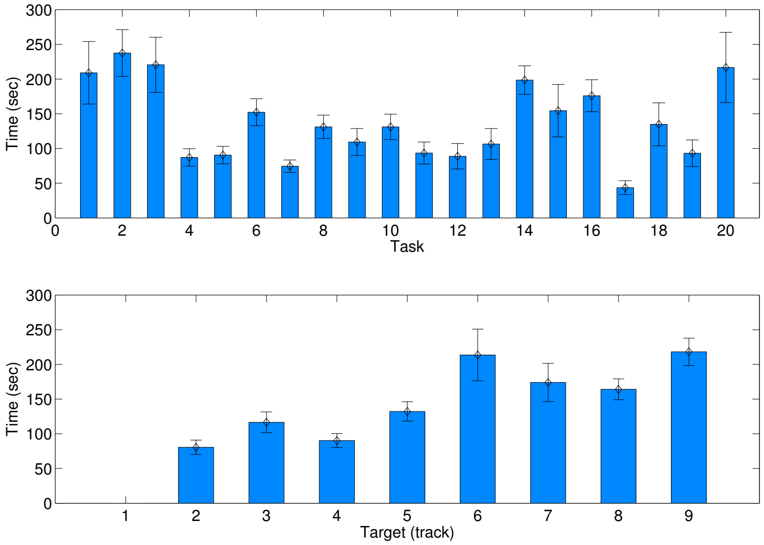


Fig. 3. Completion time as a function of task (top) and as function of target track (bottom).

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 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.

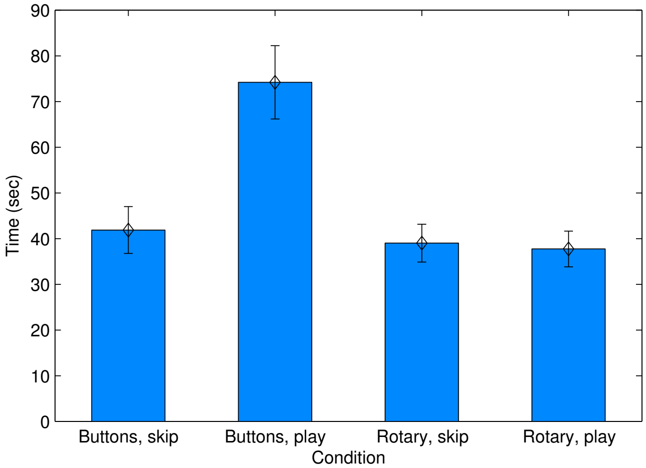


Fig. 4. Mean time used in search mode in each condition.

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 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 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

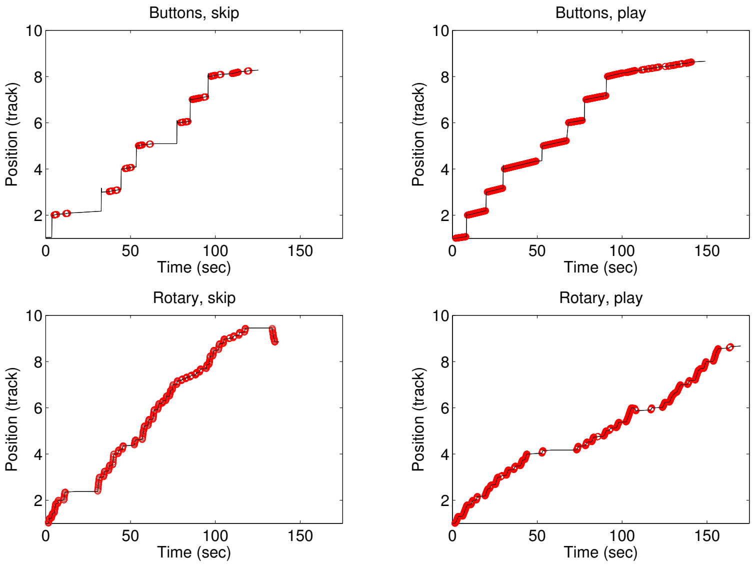


Fig. 5. Example. Position as a function of time plotted for one trial of each condition of task 16.

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)	$F(1,11) = 3.313, p = .096$	$F(1,11) = 6.600, p = .026$

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 editor 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

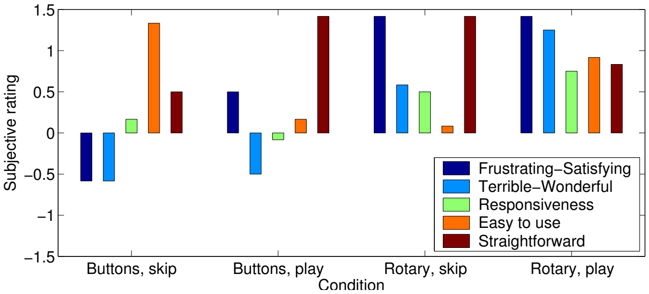


Fig. 6. Subjective ratings of the 4 interfaces on scales ranging between -2 and 2.

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.

5 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.

Chapter 3

Musical performance with pre-recorded music

3.1 Real-time beat estimation using feature extraction

Appeared in:
K. Jensen and T. H. Andersen. Real-time beat estimation using feature extraction. In Computer Music Modeling and Retrieval: International Symposium, Montpellier, Lecture Notes in Computer Science, pages 13-22. Springer Verlag, 2003.

Real-time beat estimation using feature extraction

Kristoffer Jensen and Tue Haste Andersen

Department of Computer Science, University of Copenhagen
Universitetsparken 1
DK-2100 Copenhagen, Denmark,
{krist,haste}@diku.dk,
WWW home page: http://www.diku.dk/~krist

Abstract. This paper presents a novel method for the estimation of beat interval from audio files. As a first step, a feature extracted from the waveform is used to identify note onsets. The estimated note onsets are used as input to a beat induction algorithm, where the most probable beat interval is found. Several enhancements over existing beat estimation systems are proposed in this work, including methods for identifying the optimum audio feature and a novel weighting system in the beat induction algorithm. The resulting system works in real-time, and is shown to work well for a wide variety of contemporary and popular rhythmic music. Several real-time music control systems have been made using the presented beat estimation method.

1 Introduction

Beat estimation is the process of predicting the musical beat from a representation of music, symbolic or acoustic. The beat is assumed to represent what humans perceive as a binary regular pulse underlying the music. In western music the rhythm is divided into measures, e.g. pop music often has four beats per measure. The problem of automatically finding the rhythm include finding the time between beats (tempo), finding the time between measures, and finding the phase of beats and measures. This work develops a system to find the time between beats from a sampled waveform in real-time. The approach adopted here consists of identifying promising audio features, and subsequently evaluating the quality of the features using error measures.

The beat in music is often marked by transient sounds, e.g. note onsets of drums or other instruments. Some onset positions may correspond to the position of a beat, while other onsets fall off beat. By detecting the onsets in the acoustic signal, and using this as input to a beat induction model, it is possible to estimate the beat.

Goto and Muraoka [1] presented a beat tracking system, where two features were extracted from the audio based on the frequency band of the snare and bass drum. The features were matched against pre-stored drum patterns and resulted in a very robust system, but only applicable to a specific musical style.

Later Goto and Muraoka [2] developed a system to perform beat tracking independent of drum sounds, based on detection of chord changes. This system was not dependent on the drum sounds, but again limited to simple rhythmic structures. Scheirer [3] took another approach, by using a non-linear operation of the estimated energy of six bandpass filters as feature extraction. The result was combined in a discrete frequency analysis to find the underlying beat. The system worked well for a number of rhythms but made errors that related to a lack of high-level understanding of the music. As opposed to the approaches described so far Dixon [4] built a non-causal system, where an amplitude-based feature was used as clustering of inter-onset intervals. By evaluating the inter-onset intervals, hypotheses are formed and one is selected as the beat interval. This system also gives successful results on simpler musical structures.

The first step of this work consists of selecting an optimal feature. There are a very large number of possible features to use in segmentation and beat estimation. Many audio features are found to be appropriate in rhythm detection systems, and one is found to perform significantly better. The second step involves the introduction of a high-level model for beat induction from the extracted audio feature. The beat induction is done using a running memory module, the beat probability vector, which has been inspired by the work of Desain [5].

The estimation of beat interval is a first step in the temporal music understanding. It can be used in extraction and processing of music or in control of music. The beat detection method presented here is in principle robust across music styles. One of the uses of the beat estimation is in beat matching, often performed by DJs using contemporary electronic and pop music. For this reason, these music styles has mainly been used in the evaluation. The system is implemented in the open source DJ software Mixxx [6] and has been demonstrated together with a baton tracking visual system for the use of live conducting of audio playback [7].

2 Audio Features

The basis of the beat estimation is an audio feature that responds to the transient note onsets. Many features have been introduced in research of audio segmentation and beat estimation. Most features used here have been recognized to be perceptually important in timbre research [8]. The features considered in this work are: amplitude, spectral centroid, high frequency energy, high frequency content, spectral irregularity, spectral flux and running entropy, all of which have been found in the literature, apart from the high frequency energy and the running entropy.

Other features, such as the vector-based bandpass filter envelopes [3], or mel-cepstrum coefficients have not been evaluated. Vector-based features need to be combined into one measure to perform optimally, which is a non-trivial task. This can be done using for instance artificial neural nets [9] that demands a large database for training, or by summation [3] when the vector set is homogeneous.

Most features indicate the onsets of notes. There is, however, still noise on many of the features, and the note onsets are not always present in all features. A method to evaluate and compare the features is presented in section 3, and used in the the selection of the optimal feature. In the following paragraphs, a number of features are reviewed and a peak detection algorithm is described.

2.1 Features

The features are all, except the running entropy, computed on a short time Fourier transform with a sliding Kaiser window. The magnitude $a_{n,k}$ of block n and FFT index k is used. All the features are calculated with a given block and step size (N_b and N_s respectively).

The audio features can be divided into absolute features that react to specific information weighted with the absolute level of the audio and relative features that only react to specific information. The relative features are more liable to give false detection in weak parts of the audio.

The amplitude has been found to be the only feature necessary in the tracking of piano music [10]. This feature is probably useful for percussive instruments, such as the piano or guitar. However, the amplitude feature is often very noisy for other instruments and for complex music.

Fundamental frequency is currently too difficult to use in complex music, since it is dependent on the estimation method. It has been used [9] in segmentation of monophonic audio with good results, though.

One of the most important timbre parameters is the spectral centroid (brightness) [11], defined as:

$$SC_n = \frac{\sum_{k=1}^{N_b/2} k a_{n,k}}{\sum_{k=1}^{N_b/2} a_{n,k}}. \tag{1}$$

The spectral centroid is a measure of the relative energy between the low and high frequencies. Therefore it seems appropriate in the detection of transients, which contain relatively much high frequency energy.

An absolute measure of the energy in the high frequencies (HFE) is defined as the sum of the spectral magnitude above 4kHz,

$$HFE_n = \Sigma_{k=f_{4k}}^{N_b/2} a_{n,k}. \tag{2}$$

where f_{4k} is the index corresponding to 4 kHz.

Another absolute measure, the high frequency content (HFC) [12] is calculated as the sum of the amplitudes and weighted by the frequency squared,

$$HFC_n = \Sigma_{k=1}^{N_b/2} k^2 a_{n,k}. \tag{3}$$

These features are interesting because they indicate both high energy, but also relatively much high frequency energy.

The spectral irregularity (SPI), calculated as the sum of differences of spectral magnitude in one block,

$$SPI_n = \Sigma_{k=2}^{N_b/2} \left| a_{n,k} - a_{n,k-1} \right|, \tag{4}$$

and the spectral flux (SPF), calculated as the sum of spectral magnitude differences between two adjoining blocks,

$$SPF_n = \Sigma_{k=1}^{N_b/2} \left| a_{n,k} - a_{n-1,k} \right|, \tag{5}$$

are two features known from the timbre perception research. These features give indication of the noise level and the transient behavior that are often indicators of beats.

Note onsets can be considered as new information in the audio file. Therefore the running entropy, calculated on a running histogram of the 2^{16} quantization steps is considered. First the probability of each sample value is estimated for one block,

$$H_n(s(l)) = H_n(s(l)) + \frac{1}{N_b}, l = (n-1)N_s + 1 \cdots (n-1)N_s + N_b, \tag{6}$$

then the probability is updated with $1 - W_h$,

$$H_n = W_h H_n + (1 - W_h) H_{n-1}, \tag{7}$$

and finally the entropy in bits is calculated,

$$Ent_n = -\Sigma_{k=1}^{2^{16}} H_n(k) \log_2(H_n(k)). \tag{8}$$

These are the features evaluated in this work. The note-onsets are considered to occur at the start of the attacks, but the features generally peak at the end of the attacks. To compensate for this delay the time derivative is taken on the features. The second derivative is taken on the running entropy. The maximum of the derivative of the amplitude has been shown to be important in the perception of the attack [13]. In addition, the negative values of each feature are set to zero.

An example of the resulting time-varying extracted features can be seen in fig. 1 for a contemporary music piece[1]. On the figure manually marked note onsets are indicated by dashed lines. It is clear that most features peak at the note onsets. There is, however, still noise on many of the features, and some of the note onsets are not always present in the features.

2.2 Peak detection

The features considered in the previous section all exhibit local maximums at most of the perceptual note onsets. To identify a note onset from a given feature a peak detection algorithm is needed. The peak detection algorithm used here chooses all local maximums, potentially using a threshold,

$$p = (F_{n-1} < F_n > F_{n+1}) \wedge (F_n \geq th) \tag{9}$$

where F is an arbitrary audio feature. In addition to the peak detection, a corresponding weight, w_k is also calculated at each peak k (at the time t_k),

[1] Psychodelik. Appearing on LFO - Advance (Warp 039), January 1996.

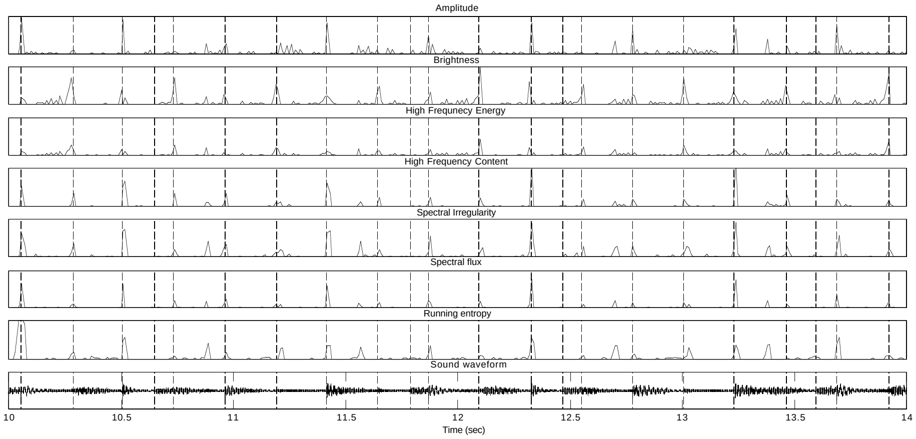


Fig. 1. Audio features from the LFO -Psychodelik piece (excerpt) as function of time. The features are shown at arbitrary scales. The vertical dashed lines indicate the manual marked transients.

corresponding to the time steps where p is true. This weight is later used in the beat probability vector, and in the detection of the phase of the beat. The threshold is used in the selection of the optimal feature, but not in the final beat estimation system.

3 Feature analysis

To compare features, different musical pieces has been analyzed manually by placing marks at every perceptual note onset. The marking consists in identifying the note onsets that are perceptually important for the rhythmic structure. These note onsets are generally generated by the hi-hat and bass drum and any instrument with a transient attack. In practice, some parts of the pieces lack hi-hat and bass drum, and the rhythmic structure is given by other instruments. The manual marking of the note onsets in time has an error estimated to be below 10 msec. In all eight musical pieces were used, with an average of 1500 note onsets per piece.

These manual marks are used as basis for comparing the performance of the various features. In order to select the optimum feature, three different error measures are used, based on matched peaks, that is peaks located within a time threshold (20 msec) to a manual mark. An unmatched peak is located outside the time threshold from a manual mark.

3.1 Error measures

To find the signal to noise the value of a matched (P), or unmatched ($\hat{P}$) peak is calculated as the sum of the feature at both sides of the peak where the slope is continually descending from the peak center. The signal to noise ratio is then calculated as,

$$s_n = \frac{\Sigma_{n=0}^{N_{matched}} P_n}{\Sigma_{n=0}^{N_{unmatched}} \hat{P}_n}. \tag{10}$$

The missed ratio is calculated as the number of manual marks minus the number of matched peaks, divided by the number of manual marks,

$$R_{missed} = \frac{N_{marked} - N_{matched}}{N_{marked}}, \tag{11}$$

and the spurious ratio is calculated as the number of unmatched peaks, divided by the number of manual marks,

$$R_{spurious} = \frac{N_{unmatched}}{N_{marked}}. \tag{12}$$

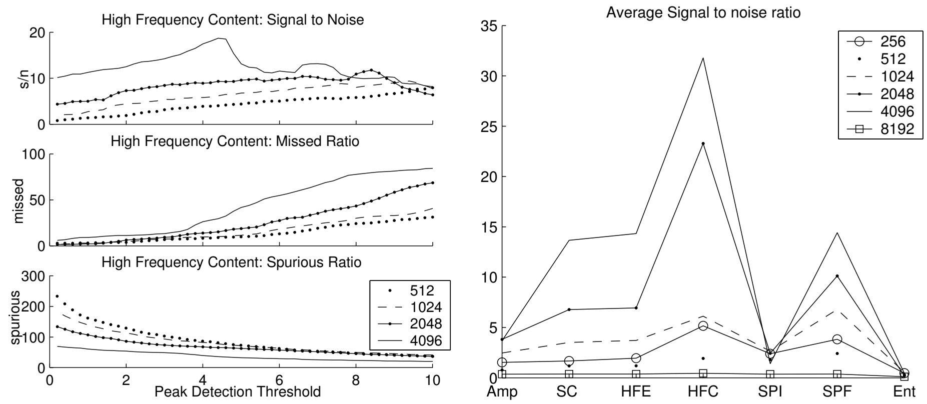


Fig. 2. Left: Example of error measures calculated using different block sizes of the HFC feature for the piece Train to Barcelona. Right: Average signal to noise for fixed threshold, and all music pieces for many block sizes and features.

3.2 Analysis and selection of feature

In order to evaluate the features the error measures are now calculated on the music material using a varying peak detection threshold. An example of the error measures for the piece Train to Barcelona[2] is shown in the left part of fig. 2. For low thresholds, there are few missed beats, and for high peak detection threshold, there are many missed beats. The spurious beats (false indications)

[2] By Akufen. Appearing on Various - Elektronische Musik - Interkontinental (Traum CD07), December 2001.

behave in the opposite way, for low thresholds there is up to several hundred percents, whereas the spurious ratio is low for high peak detection thresholds. Under these conditions it is difficult to select an optimum peak detection threshold, since both low missed and spurious ratio is the optimization goal and they are mutually exclusive. The signal to noise ratio generally rises with the peak detection threshold, which indicates that the few found peaks contain most of the energy for the high thresholds. There seem to be no optimum way of selecting the threshold.

An analysis of the error values for all features and music pieces gives no clear indication of the best feature. Therefore a different approach has been used.

Initial tests have shown that the beat estimation method presented in the next section need at least 75% of the note onsets to perform well. The threshold for 75% matched beats (25% missed) is therefore found for each features/block size pair and music piece. The signal to noise ratio is then found for this threshold. The average signal to noise ratio is calculated for all music pieces. The result is shown in the right part of fig. 2.

Several results can be obtained from the figure. First, it is clear that the extreme block sizes, 256, 512, and 8192 all perform inadequately. Secondly, several features also perform poorly, in particular the amplitude, the spectral irregularity, and the entropy. The best features are the spectral centroid, the high frequency energy, the high frequency content and the spectral flux. The HFC performs significantly better than the other features, in particular for the block sizes 2048 and 4096, which has the best overall signal to noise ratio.

4 Beat estimation

The analysis of the audio features has permitted the choice of feature and feature parameters. There is, however, still errors in the detected peaks of the chosen features. As described in other beat estimation systems found in the literature, a beat induction system, that is a method for cleaning up spurious beats and introducing missing beats, is needed. This could be, for instance, based on artificial neural nets, as in [9], but this method demands manual marking of a large database, potentially for each music style. Another alternative is the use of frequency analysis on the features, as in [3], but this system reacts poorly to tempo changes.

Some of the demands of a beat estimation system are stability and robustness. Stability to ensure that the estimation is yielding low errors for music exhibiting stationary beats and robustness to ensure that the estimation continues to give good results for music breaks without stationary beats. In addition, the system should be causal, and instantaneous. Causal to ensure real-time behavior, and instantaneous to ensure fast response.

These demands are fulfilled by the use of a memory-based beat probability vector that is based on the model of rhythm perception by Desain [5]. In addition a tempo range is needed to avoid the selection of beat intervals that do not occur in the music style. The tempo is chosen in this work to lie between 50 and 200 BPM, which is similar to the constraints used in [3].

4.1 Beat probability vector

The beat probability vector is a dynamic model of the beat intervals that permits the identification of the beat intervals from noisy features. The probability vector is a histogram of note onset intervals, as measured from the previous note onset. For each new note onset the probability vector $H(t)$ is updated (along with its neighboring positions) by a Gaussian shape at the intervals corresponding to the distance to the previous peak. To maintain a dynamic behavior, the probability vector is scaled down at each time step. At every found peak k the peak probability vector is updated,

$$H(t) = W^{t_k - t_{k-1}} H(t) + G(t_k - t_{k-1}, t), t = 0 \dots \infty \tag{13}$$

where W is the time weight that scale down the probability of the older intervals, and G is a Gaussian shape which is non-zero at a limited range centered around $t_k - t_{k-1}$. The current beat interval is identified as the index corresponding to the maximum in the beat probability vector, or, alternatively, to $t_k - t_{k-1}$ if the interval is located at the vicinity of the maximum in the beat probability vector.

The memory of the beat probability vector allows the detection of the beat interval in breaks with missing or alternative rhythmic structure. An instantaneous reaction to small tempo changes is obtained if the current beat interval is set to the distance between peaks at proximity to the maximum in the vector.

In [5] multiples of the intervals are also increased. Since the intervals are found from the audio file in this work, the erroneous intervals are generally not multiples of the beat. Another method must therefore be used to identify the important beat interval.

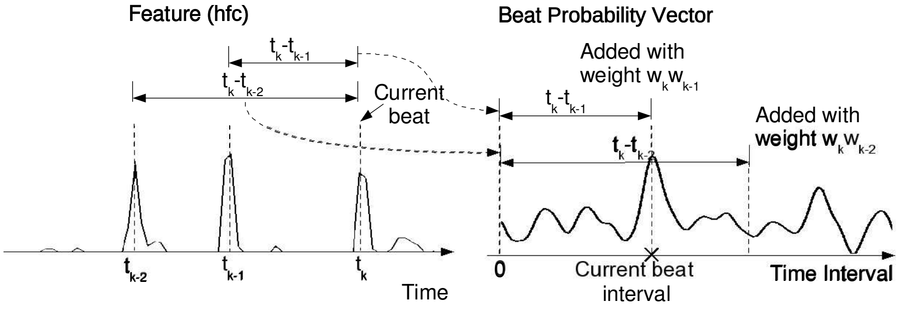


Fig. 3. *Selection of beats in the beat probability vector. For each new peak (left), a number of previous intervals are scaled and added to the vector (right). The maximum of the beat probability vector gives the current beat interval.*

4.2 Update with multiple intervals

To avoid a situation where spurious peaks create a maximum in the probability vector with an interval that does not match the current beat, the vector is updated in a novel way. By weighting each new note and taking multiple previous note onsets into account, the probability vector $H(t)$ is updated with N previous weighted intervals that lie within the allowed beat interval,

$$H(t) = H(t) + \Sigma_{i=1}^{N} w_k w_{k-i} G(t_k - t_{k-i}, t), t = 0 \ldots \infty \tag{14}$$

For simplicity, the time weight W is omitted in this formula.

This simple model gives a strong indication of note boundaries at common intervals of music, which permits the identification of the current beat interval.

An illustration of the calculation of the beat probability vector can be seen in figure 3. It consists of the estimated audio feature (left), the estimation of probable beat and the updating of the running beat probability vector (right). The current beat interval is now found as the interval closest to the maximum in the beat probability vector. If no such interval exists, the maximum of the beat probability vector is used.

5 Evaluation

The beat estimation has been evaluated by comparing the beat per minute (BPM) output of the algorithm to a human estimate. The human estimate was found by tapping along while the musical piece was playing, and finding the mean time difference between taps.

To evaluate stability of the algorithm 10 pieces of popular and electronic music was randomly selected from a large music database. In all cases the algorithm gave a stable output throughout the piece, after a startup period of 1 to 60 seconds. The long startup period is due to the nature of the start of these pieces, i.e. non rhythmic music. In six of the cases the estimated BPM value matched the human estimate, while in the remaining four cases, the algorithm estimate was half that of the human estimate. The problem of not estimating the right multiple of BPM is reported elsewhere [3], however, it is worth noting that in the case of controlling the tempo of the music, it is of primary importance to have a stable output.

In addition, informal use of the system in real-time audio conducting [7], DJ beat matching and tempo control [6] has shown that the beat estimation is stable for a large variety of music styles.

6 Conclusions

This paper presents a complete system for the estimation of beat in music. The system consists of the calculation of an audio feature that has been selected from a large number of potential features. A number of error measures have been calculated, and the best feature has been found, together with the optimum threshold and block size, from the analysis of the error measures. The selected feature (high frequency content), is further enhanced in a beat probability vector. This vector, which keeps in memory the previous most likely intervals, renders an estimate of the current interval by the maximum of the beat interval probabilities.

The paper has presented several new features, a novel approach to the feature selection, and a versatile beat estimation that is both precise and immediate. It has been implemented in the DJ software Mixxx [14] and used in two well proven real-time music control systems: Conducting audio files [7] and DJ tempo control [6].

References

1. Goto, M., Muraoka, Y.: A real-time beat tracking system for audio signals. In: Proceedings of the International Computer Music Conference. (1995) 171–174
2. Goto, M., Muraoka, Y.: A real-time beat tracking for drumless audio signals: Chord change detection for musical decisions. Speech Communication **27** (1998) 311–335
3. Scheirer, E.D.: Tempo and beat analysis of acoustic musical signals. J. Acoust. Soc. Am. **103** (1998) 588–601
4. Dixon, S.: Automatic extraction of tempo and beat from expressive performances. Journal of New Music Research **30** (2001) 39–58
5. Desain, P.: A (de)composable theory of rhythm. Music Perception **9** (1992) 439–454
6. Andersen, T.H.: Mixxx: Towards novel DJ interfaces. Conference on New Interfaces for Musical Expression (NIME'03), Montreal (2003)
7. Murphy, D., Andersen, T.H., Jensen, K.: Conducting audio files via computer vision. In: Proceedings of the Gesture Workshop, Genova. (2003)
8. McAdams, S., Winsberg, S., Donnadieu, S., Soete, G.D., Krimphoff, J.: Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes. Psychological Research **58** (1995) 177–192
9. Jensen, K., Murphy, D.: Segmenting melodies into notes. In: Proceedings of the DSAGM, Copenhagen, Denmark. (2001)
10. Dixon, S., Goebl, W., Widmer, G.: Real time tracking and visualisation of musical expression. In: II International Conference on Music and Artificial Intelligence. Volume 12., Edinburgh, Scotland (2002) 58–68
11. Beauchamp, J.: Synthesis by spectral amplitude and "brightness" matching of analyzed musical instrument tones. Journal of the Acoustical Society of America **30** (1982)
12. Masri, P., Bateman, A.: Improved modelling of attack transient in music analysis-resynthesis. In: Proceedings of the International Computer Music Conference, Hong-Kong (1996) 100–104
13. Gordon, J.W.: The perceptual attack time of musical tones. J. Acoust. Soc. Am. **82** (1987)
14. Andersen, T.H., Andersen, K.H.: Mixxx. http://mixxx.sourceforge.net/ (2003)

3.2 DJing with Dance Music: The Design and Evaluation of a Digital DJ System

Submitted for publication.
T. H. Andersen. DJing with Dance Music: The Design and Evaluation of a Digital DJ System. 10 pages. 2005.

DJing with Dance Music:
The Design and Evaluation of a Digital DJ System

Tue Haste Andersen
Department of Computer Science
University of Copenhagen
DK-2100 Copenhagen Ø
Denmark
haste@diku.dk

ABSTRACT

A formal study of DJ work practices with traditional DJ equipment, such as turntables and CD players, is presented. On average, 47% of the time used by 10 professional DJs playing live was used on a task related to beat matching. Beat matching with traditional equipment requires high cognitive workload despite tangible input and feedback. Based on this study and on informal explorations we present the design of a digital DJ system, Mixxx, that features automatic beat synchronization designed to reduce the cognitive workload and maintain a high level of control. An evaluation of Mixxx with nine DJs using two different interface mappings was performed. Although the DJs were unfamiliar with Mixxx, and the interface lacked the feel of the traditional equipment, all DJs were almost instantly able to use Mixxx. On average the DJs preferred the interface mapping that offered automatic beat matching capabilities, compared to a mapping similar to the traditional setup.

KEYWORDS: Disc Jockey, DJ, turntable, CD player, mixer, tangible interfaces, mappings, work practices, video.

INTRODUCTION

Over the last 40 years a new type of musician has evolved, the Disc Jockey (DJ). According to one of the classics in modern DJing [7], the role of the DJ is to "track down greatness in music and squeeze it together." The musical form and expression of DJing are unique in that already recorded and produced music tracks form the unit of music produced by the DJ. The DJ takes what others have created, blends and alters it, to form a unique DJ set.

We chose to focus on club DJing of popular dance and electronic music which is probably the most common type of DJ. We present the design and evaluation of a digital DJ system, Mixxx, designed to replace traditional equipment such as turntables and mixers.

Computers offer obvious advantages for DJing. Visualization techniques may aid song navigation and synchronization, tools for information retrieval and meta-data may help in performing tasks that are manual and tedious, an unlimited array of software sound effects may be used, and finally a laptop based setup is small and light to carry compared to turntables and vinyl records.

DJing with WIMP interfaces, however, has several shortcomings over the tangible [12] and direct manipulation interface [21] of traditional DJ equipment. WIMP interfaces lack proper feel and haptic feedback and performing live with for instance a laptop hinders visual communication [22] with other musicians. The difficulties at constructing a user interface that has the same feel and power as the traditional equipment is a major obstacle in the transition to laptop based equipment.

To gain a better understanding of how digital DJ systems can complement or replace the traditional setup, we present a formal study on DJ work practices, followed by the design and evaluation of the digital DJ system Mixxx. Ten DJs playing live were recorded on video and later analyzed. The video recordings were complemented by interviews. In the design phase of Mixxx, various elements of the interface were tested in informal sessions with DJs. Finally, an evaluation with nine DJs was performed. Two setups of Mixxx were compared to test how well DJs liked the interfaces. We wanted to see if factors such as task performance and acceptance were influenced by introducing tools to perform tasks automatically that traditionally were performed manually.

In the following sections we present a brief overview of DJing today and discuss related work. This is followed by a section on the study of DJs current work practices. The DJ system Mixxx is presented and evaluated, and finally we conclude on the findings of this work.

DJ'ING TODAY

Like the playing of most other musical instruments, DJing involves manual tasks that demand specialized gestures and leads to a high cognitive workload. It often takes years of practice to perform the tasks adequately for live performance. The most important tasks include song selection and mixing. Mixing two songs so that they blend together, without an audible start and end of one song, is referred to as beat mixing [7]. The process requires beat matching [7, 5], i.e. that two tracks are matching in tempo and sometimes also in

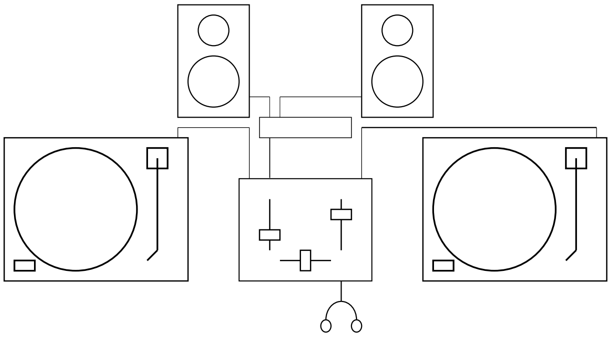


Figure 1: A typical DJ setup with two turntables, mixer, sound system and headphones.

pitch, in the region where the two songs is mixed. The process involves not only the haptic skill of using the mixer and turntable, but also split hearing [7], the ability to attend to different musical pieces in each ear.

A typical DJ setup is shown in Figure 1. It consists of at least two playback devices and a mixer. Traditionally DJs have been using analog equipment such as turntables and mixers. However, as more music is distributed digitally, we currently see a shift toward digital DJ tools such as CD players. In the following we describe the equipment common to DJ setups. The description does not include laptops as they are not commonly used in dance clubs.

Playback device

Turntable. The standard DJ turntable provides a rotating platter on which a record is placed. The vibrations caused by the moving record under the stylus are transformed into electrical energy that represents the sound pressure level when the signal is played through loudspeakers. Since the sound is reproduced by this simple mechanical system, and since both the arm holding the stylus and the platter can be touched and controlled directly by hand, the turntable offers several direct ways of manipulating the sound playback:

1. The stylus allows for random access of play position within the musical tracks on one side of a record
2. Musical tracks are selected through the tangible interface of vinyl records
3. Playback speed is adjusted by touching the record while playing. This can range from small temporary adjustments in speed to reverse playback through halting.

The turntable has two other controls: A push button to control start/pause mode and a slider to adjust playback rate. The slider is often referred to as a pitch slider, because changing the speed of playback also changes the pitch.

The DJ turntable offers auditory, visual and haptic feedback. The auditory feedback is given when playing a record, visual from light reflections in the grooves of the record and from the rotating platter, and finally haptic feedback when touching the platter and slider.

The understanding of the controls provided by the turntable is important since it is the primary instrument of DJing. Most DJs have spent years of practice to learn and use it and are interested in transferring and reusing their hard learned skills when using new instruments.

CD player. The DJ CD player has evolved over the last 10 years from the home stereo CD player to become similar to the DJ turntable with regards to the interface. Pioneer evolved the DJ CD Player with models such as CDJ-100 that provided a large jog wheel and pitch slider. One feature that sets the CD player apart from the turntable is the ability to adjust pitch independently of tempo. With Pioneer CDJ-1000 this interface was further evolved by using a larger jog wheel with a feel closer to that of touching a vinyl record. However, these CD players all lacked active haptic feedback from the platter since no motor was provided to rotate the platter at a speed corresponding to the rotating vinyl platter. There are several problems caused by the lack of the rotating platter. First, it has a different feel compared to a turntable. Second, it is hard to judge if the CD player is in playback mode by looking at it. Third, an extra mode is needed to use the jog wheel for both temporary tempo change and scratching. With the Technics CD player, SLDZ1200, these problems are compensated for by introducing a large rotating platter that emulates the feel of a vinyl turntable.

Mixer

Using the mixer it is possible to blend different sound signals and listen to one signal in the headphones independently of what is being played in the loudspeakers. Frequency based filters or equalizer in three bands is usually provided to boost or dampen volume in one or more of the three frequency bands. To prevent distortion when boosting a frequency band, a pre gain volume is used to adjust volume before it is affected by the equalizer. External sound effects such as flanger and echo can usually be connected to the mixer.

As input, the mixer takes the audio signal from each playback device and control values from the interface. As output two different sound signals are provided: One for the loudspeakers and one for the attached headphones.

A mixer typically has several ways to control volume of the mix. A volume fader is provided for each channel and a crossfader is used to fade between two channels. Knobs and buttons are used for adjusting filters, gain and external sound effects.

Sampler

A sampler is used to sample a short duration of a sound source for later replay by pushing a button. The sample can be looped, i.e. played continuously over and over. A sample is recorded from a sound source by pressing and holding a button in the time corresponding to the length of the sample. Using this method it can be hard to record the sample with the exact timing needed to play the sample as a loop. More advanced samplers can to some extend estimate the beat, and provide the correct timing information to make a loop.

RELATED WORK

Over the last five years, research and development of digital DJ systems has started and evolved. Today some of the leading commercial systems [2, 3] offer a complete package for playback and mixing. In FinalScratch the traditional DJ turntables acts as input device to the DJ application and thus bring together the efficiency and portability of digital music with the effective control of the traditional analog setup. The traditional setup has proved useful over the entire history of DJing, but was shaped not only by its practitioners but also by the limitations of the analog equipment. Only few attempts to provide new ways of interacting with the music have been explored in a commercial setting while focusing on the music track as the fundamental unit of musical performance. One example of this is Ableton Live [1] that can be described as a combination between a sequencer and a looping application. With Ableton Live the DJ can edit a mix to perfection at home, while still maintaining a level of control during live usage.

In academia, projects such as AudioPad [18] and Block Jam [16] have demonstrated new and intriguing ways of controlling musical playback. However, such systems are based on sequencing software not designed for playback of already edited music tracks but rather composition from pre-arranged samples and synthesis engines.

WORK PRACTICES

Plenty of resources exist on DJs work practices today. In research, scratching [9] has been studied and turntable interaction [5] explored. From DJs there is extensive documentation on the Internet and from books [7]. These are qualitative descriptions of how DJs work and the tasks the work involves. However, no quantitative data has been reported on their work practices. To further deepen our understanding of the DJ work practices we have conducted a field study of DJs playing live. The DJ sets were recorded on video and analyzed. The data obtained from the video recordings were further complemented by interviews.

Method

Video recordings have previously been used to inform interaction design of novel fields in HCI research [15]. Since contextual inquiry [6] during the recording of the DJ sets was impossible we conducted separate interviews of DJs.

Field study. 10 DJ sets with different DJs were recorded in popular dance clubs. We used recordings of DJs playing in the Vega Nightclub and Krudttønden, both located in Copenhagen, Denmark. Vega is one of the major dance clubs in Copenhagen for popular electronic music covering musical genres such as pop, house and techno. Krudttønden was at the night we recorded used for a special dance event where the musical style spanned house and techno. Each of the 10 DJs was recorded in one hour, between 12 PM and 5 AM in the night by placing a video camera behind the setup. The video camera was placed behind and above the DJ giving a clear view to the devices in use. The night-vision feature of the camera was used to avoid additional lightening sources. Each DJ was performing the set following a certain structure. Most DJs started with relatively ambient music, gradually playing faster or more aggressive music culminating in a hit. To obtain the highest uniformity between the recorded DJ sets, we selected an hour of video in the middle of each DJ set. The recorded DJs were all professional from Denmark, England and Germany. Some of the DJs ranged among the top club DJs in those countries.

The video was later analyzed by importing the video into a digital video editing application, for efficient playback and seeking. We manually counted how much time was used in each task by watching the video at slow speed, and making extensive use of the playback controls such as stop, rewind, and seeking. To determine which task was performed we used cues such as the music that was playing, where the DJs hands were placed and where the DJ was looking and standing. For instance, if both hands were placed on the mixer we recorded this as the DJ was mixing. However, sometimes the DJ was performing a secondary task such as dancing or adjusting the light on the dance floor. In all cases we only recorded the primary task.

Interviews. Ten open ended interviews were conducted with two amateur and eight professional DJs, all male. Three of the professional DJs were also participating in the field study. The amateur DJs were skilled DJs and had played live many times. The DJs were between 25 and 40 years of age, all played dance music in genres spanning pop, latin, dub, house and techno. They were selected for the interview based on their willingness to participate in testing computer based DJ systems.

Tasks

We divided the time used by the DJs into six tasks: Mixing, cueing, searching for records, adjust light, social activity and being passive. The tasks are described in the following:

Mixing. Mixing is adjusting the audio output by crossfading between playback devices, adjust filters and using the sampler built into the mixer. Mixing is typically done when one record is playing, or when two records are cued to play in synchronization. Using the mixer is often a highly creative process where the sound is changed to sound different from the original recording. However, using the mixer also involves more tedious work such as adjusting gain and headphone sound pressure level, or sampling a beat so that it loops smoothly.

Cueing. Cueing is the process of finding the position and tempo in a record where mixing can be started from. On a turntable this involves placing the stylus on the right spot on the record, adjusting playback speed on the pitch slider, and adjusting the phase of playback by touching the platter to temporarily speed up or slow down the playback. On a DJ CD player this is done in a similar way, but an advantage is that cue points can be stored on a flash memory card and inserted in the CD player along with the corresponding CD.

Search for records. The records were placed behind the mixing desk typically in a box or binder. Searching through the records was done by looking at the cover and label of the records.

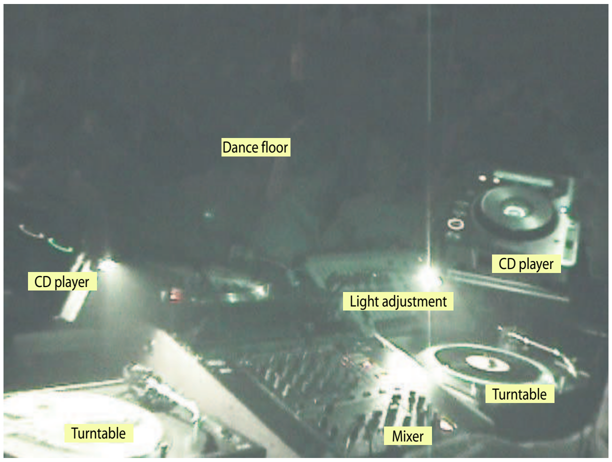


Figure 2: The DJ setup used in Vega Nightclub.

Adjust light. Light on the dance floor was adjusted by a special mixer with sliders, placed in front of one of the turntables. Often the light was controlled during otherwise passive periods, or periods waiting for the cueing of two records.

Social activity. Drinking and talking with friends or audience.

Passive. Dancing or looking at the audience while not touching the equipment. Even though labeled passive, the DJ could use the time for monitoring reactions from the audience.

Results

In the interviews, all DJs said they used time to prepare a DJ set before playing live. However, the professional DJs playing live several times a week used down to 15 minutes, only to select which records to bring with them. For the amateur DJs and DJs playing less often, several hours could be used for preparation. When asked what the most important thing was for a DJ, many said that it was to know his/her collection of records, to know where they could be mixed, and which ones played well together. The DJs said they select between 30 and 100 records to bring with them for a set. This is in agreement with the observations done in the field study, where each DJ would bring between 50 and 100 records. Seven DJs said they would change their set to a large degree depending on the audience, i.e. the number and type of people. Four said they would to a large extend keep playing what they have planned, with only minor changes to which tracks and the order of the tracks.

The setup used in Vega, shown in Figure 2, included two Technics 1210-MK2 turntables, two Pioneer CDJ-1000 CD players, one mixer with build in sampler, and a mixer for controlling the lighting on the dance floor. Records were placed behind the mixing table.

The average time used in each task is shown on the pie chart in Figure 3. The time spent in each task by each DJ is shown in Table 1 along with information on mean and standard deviation for each task. The task that the DJs on average is using most time on is cueing that accounts for 47% of the time. We expected the amount of time used in cueing to be relatively high since cueing mostly involves the process of beat matching. Only rarely was cueing used for other tasks such as search through a record to find a specific track or part of a track. When confronting the DJs participating in the interviews with statistics from the video analysis, all DJs agreed that they used a lot of time on beat matching, but most were surprised that it was that much time. All but one agreed that beat matching was a manual task that they would rather be without. Instead they would use the time for more creative aspects of DJing, such as playing with filters, sound effects and loops.

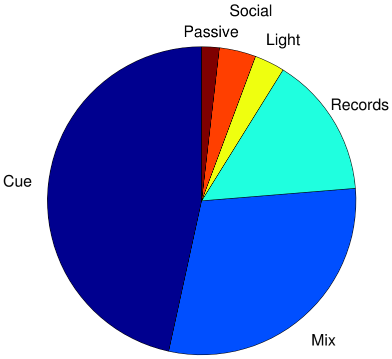


Figure 3: Pie chart showing the average time spent in each task and the standard deviation for each task.

The time used for mixing was also relatively high. Thirty percent of the time was used in mixing that accounts for most of the creative aspects of DJing. Searching for records was done in 15% of the time. DJ #1, 2 and 3 used 15% of their time on adjusting the dance floor lighting. These were DJs regularly appearing in Vega Nightclub, and thus were familiar with the setup. The foreign DJs at Vega (#4, 6 and 7) rarely used the light, but instead another DJ was controlling it at irregular intervals. In Kruddtønden the DJ could not control the light which instead was controlled by another person. The time used in social and passive activity was low, on average between 4 and 5 percent. Typically the DJ responded when people were talking to them or when tapped on the shoulder, but rarely for more than 10 seconds. The aforementioned factors regarding the relatively little time used in lighting adjustment, social and passive activity are all indications that a high workload is required to cue and mix. The DJs tried to minimize the time used in these activities to gain more time to cue and mix.

Figure 4 shows an example of tasks performed by DJ #2 and #6 during 20 minutes.

All DJs in the interviews used both vinyl and CDs in their mix, although five preferred vinyl to CD. No DJ preferred to use CDs rather than vinyl. Three of the DJs used CDs that they had previously edited and prepared on computers to make them start exactly on the beat, or by playing a loop over and over as one track on the CD. However, none of the DJs used the flash memory feature of modern CD players to store

DJ #	Cue	Mix	Rec.	Light	Social	Passive
1	24	20	9	5	0	2
2	24	19	9	6	1	1
3	20	21	9	5	3	2
4	31	19	9	0	1	0
5	26	17	10	1	5	1
6	29	20	8	0	1	2
7	28	14	16	0	2	0
8	37	14	4	0	2	3
9	31	16	6	2	5	0
10	29	18	9	0	3	0

Mean (minutes)

	27.9	17.8	8.9	1.9	2.3	1.1

Standard deviation (minutes)

	4.7	2.5	3.1	2.5	1.7	1.1

Table 1: Time in minutes used for each task by the 10 DJs with mean and standard deviation for each task.

cue and loop points. Only two had previously used computers in a live DJ situation, and these two were using computers when they DJed together with other musicians. However, six of those not using computers said they would like to explore and use computers in the future. One DJ, despite his interest in using computers live, noted that it would be problematic to use computers live. Having to install and connect the computer to the sound system late in the night when the party was well underway was the main reason why he had chosen to use CD players instead of computers.

As mentioned previously all the recorded DJs followed a certain pattern in their DJ set, starting slowly or with more ambient music gradually building up. We did not observe any direct evidence that the DJ monitored and responded to the audience through the music being played. Visual information such as reflections from the record grooves was only observed to be used by two of the 10 DJs. The CD players in Vega were placed in such a way that it was difficult to monitor the visual displays. In general, the DJs seemed to rely on haptic and auditory feedback rather than visual.

Discussion

First, the most surprising finding of this study was that 47% of the time used by the DJs was used on cueing. The time used in cueing mostly accounts for time spent on beat matching. The number differed from how much time the DJs thought they used on this task. We did not find any significant difference in how much time was spent in cueing between highly skilled and less skilled DJs. However, there may still be unquantified differences since we did not measure the quality of the beat matching and secondary task. These factors may have been influenced by the experience and training of the DJ.

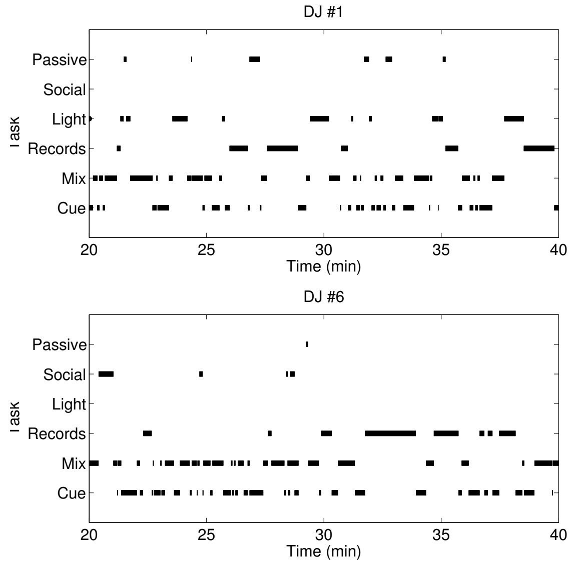


Figure 4: Task performed by DJ #1 (top) and DJ #6 (bottom) as a function of time during 20 minutes of DJing.

Second, we observed indications of a high workload. Relatively little time was used on social activity or passivity. When engaged in social activity such as conversation, the conversation was usually not initiated by the DJ, and was most often finished in less than 10 seconds.

Third, all but one DJ participating in the interviews regarded beat matching as a manual task that they would rather be without. Three of the professional DJs tried actively to lower the burden of beat mixing, by using special CD's that could be started exactly in phase with other tracks. Thus the only thing that needs adjustment in the live situation was the tempo of the song.

Fourth, we found a relatively positive attitude towards using computers in a live situation. However, this may have been influenced by the fact that the participants were selected on their interest in participating in an experiment with computer based DJ systems.

MIXXX: A DIGITAL DJ SYSTEM

In designing a digital DJ system, it is important to know the target group. There is a great variety in musical expression and how DJ tools are used depending on the type and style of music being played. Here we focus on dance and club DJs. These DJs primarily use complete musical tracks as the main unit of their DJ sets in contrast to other types of DJs such as turntable instrumentalists [9] and DJs that play with other musicians.

In the following we describe the main concepts of the interface employed in Mixxx, the underlying architecture, how it was developed, and the use and integration of information retrieval techniques.

Metaphors: The basic design

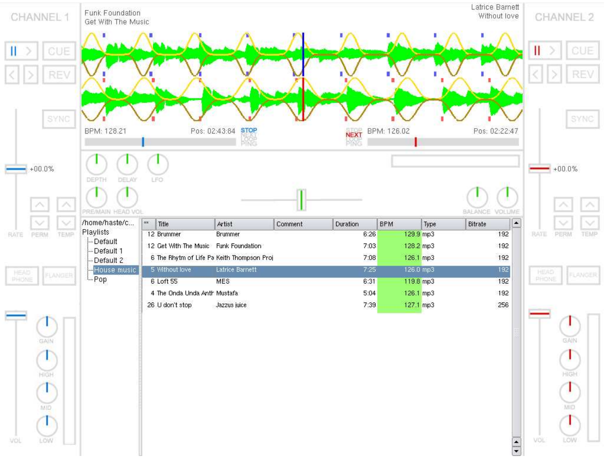
We based the design on the traditional setup metaphor, with two playback devices and a mixer. There are several reasons for this choice. First, since the focus is not on new tangible ways of interacting with the system, the problem of achieving low latency operation with, for instance, vision based object recognition methods are avoided. In professional audio equipment the latency between interface and sound is usually not above 10 ms. Second, the musical tracks remain central in the interface. This is the DJs main unit of composition. While sequencers such as Ableton Live and tangible interfaces such as AudioPad may provide new ways of interacting with recorded music, it places a burden on the DJ to prepare and arrange songs and samples ahead of time. We want to avoid this. Third, when constructed around the traditional setup metaphor, it is easy to understand the meaning of buttons and displays for DJs familiar with the traditional setup. Basic skills and techniques learned from analog equipment can be used right away, while new techniques, unique to the digital interfaces, can be explored and learned gradually. Fourth, the traditional setup does not conflict with automating manual tasks such as beat matching when physical arrangement and mappings of the playback devices are changed. We wanted to provide automatic beat matching in Mixxx. Such a feature could potentially reduce the cognitive workload and free up time for more creative aspects of DJing.

Basic features

A screenshot of Mixxx is shown in Figure 5. The screenshot shows many of the controls found on traditional mixers and playback devices. They include volume faders, pre-gain knobs, three band equalizer and other sound effects, headphone channel, volume and balance adjustments. The playback devices can be controlled by a set of buttons: Play/pause, reverse, fast forward, and fast backward. Pitch is controlled by a slider and a set of buttons. Two buttons can be used for fine adjustment of the pitch. The buttons are provided to compensate for the lack of precision in the GUI slider. Another set of buttons adjust the tempo up or down but only while the buttons are pressed. The buttons are used to temporarily speed up or slow down the playback to adjust the phase of the beats. The phase can also be adjusted by dragging the scrolling waveform display with the mouse and is designed to be similar to temporarily pushing or holding the platter on a turntable. Dragging the waveform backwards slows the playback down rather than reversing the playback direction. This mode of operation conflicts with emulating a scratching sound. Another input widget or mode is needed for scratching similar to CD players where no motor is provided to rotate the jog wheel However, scratching functionality was not implemented since the focus of Mixxx is dance DJs that rarely performs scratching and we did not want to confuse the user with unnecessary features.

Setting a cue point is another feature implemented in Mixxx. On vinyl records, cue points can be placed by using paper stickers to indicate good places to start playback. On CD players this is more conveniently accomplished by using a button to place the cue point, and using another to return to this cue point. Cue points can be used to start the playback at the exact position of a beat. By doing so, further adjustment of the phase is avoided and only the tempo needs adjustment to match the beats of two songs.

Finally, essential to a mixer is the ability to listen to one or more tracks in headphones while others are being played back for the audience. Mixxx implements this by use of multi-channel sound cards. The user can select which playback device is played in the headphone channel and which is played in the master channel.



**	Title	Artist	Comment	Duration	BPM	Type	Bitrate
12	Brummer	Brummer		6:26	129.9	mp3	192
12	Get With The Music	Funk Foundation		7:03	128.2	mp3	192
6	The Rhytm of Life Pa	Keith Thompson Proj		7:08	126.1	mp3	192
5	Without love	Latrice Barnett		7:25	126.0	mp3	192
6	Loft 55	MES		6:31	119.8	mp3	192
4	The Onda Unda Anth	Mustafa		5:04	126.1	mp3	192
26	U don't stop	Jazzus juice		7:39	127.1	mp3	256

Figure 5: Screenshot of Mixxx with the default skin.

Architecture and development process

Mixxx was developed in C++ using the cross platform QT library. Using C++ in combination with QT it was possible to develop for all major desktop operating systems, and at the same time take advantage of an object oriented code design and abstraction from hardware. Mixxx is available as an open source program distributed under the General Public License [11].

A number of developers and DJs contributed to the design and development of Mixxx. Support for sound APIs, file formats and features such as playlist import has been added by other developers. The design and development of Mixxx has benefited from this development model in a positive way.

Different algorithms are provided for change of tempo by use of the open source library, SoundTouch [17]. An architecture for Fourier based offline and real-time music information retrieval algorithms is also integrated in Mixxx [13].

Flexible mappings

The usability of Mixxx and other DJ software depends to a large extend on the type of input available. In Mixxx, basic controls for input and output are defined as either binary or range controls. Range controls are further divided into position and rate based controls. These controls can be mapped to GUI widgets, MIDI events, or other controllers such as keyboard, mice, joysticks and specific DJ devices. The mappings to GUI widgets are specified in a XML file together with the layout of the graphical user interface. Mixxx supports a number of different mappings. For instance, if most functionality is accessible from an external controller, control widgets can be removed from the screen leaving more room for the visual waveform displays.

We have designed Mixxx to use as few modes as possible. Only modes used in the traditional setup have been replicated, namely modes for headphone cueing and sound effects. Where modes could not be avoided, because of lack of physical buttons, we have instead used quasi modes [19] that only temporarily set the mode while the user is activating the corresponding mode setter.

The flexibility of the system was particularly useful during the design of Mixxx, where it allowed for experimentation of the setup of the application. A logging facility was also used to get detailed information from user trials.

We tested the use of various physical controllers for playback control in Mixxx. Specifically we tested commercial controllers designed for DJing such as the Mixman DM2 controller and Hercules DJ console. Both provide jog wheels, and a set of sliders and buttons. None of them come close to the feel and experience of using a turntable, but the Hercules console is somewhat similar to using a DJ CD player.

Visual displays

The waveform displays in Mixxx were carefully designed to support and aid novice DJs. In Figure 5 the two waveform displays show a detailed view of the energy in approximately 10 seconds around the playback position. In electronic music the beat is usually visible from such a display where the local maximums and minimums correspond to the location of the bass drum. As playback proceeds the display scrolls from right to left. In the first Mixxx prototype we experimented with the use of Fisheye displays [8, 4] to improve overview of distant musical events while still being able to perceive local characteristics such as beat information from the display. The size of the fisheye was controlled independently of other parameters. This proved to be very ineffective. The distorting lens effect of the fisheye seemed confusing and the independent control of the fisheye size complicated the use of the application unnecessary. The display could possibly be designed in a better way, for instance by controlling the magnification factor and size of fish eye automatically [10].

Another problem with visual displays typically found in DJ applications is that the displays are laid out horizontally, side by side, rather than vertically as in Mixxx. This makes it impossible to use the displays as an aid in beat matching. However, despite the vertical arrangement, the displays did not allow for visual synchronization of the tempo when first implemented in Mixxx. The playback speed affected the speed at which the waveform displays were scrolling. Only the point exactly at the playback position could be used for visual synchronization. Instead we applied a principle similar to the automatic zoom used in scrolling interfaces [10] to ensure a constant scrolling rate regardless of playback speed. Using automatic zooming it was possible to use the waveform displays as a visual aid in beat mixing.

To compensate for the overview originally provided by the fisheye, we introduced a separate overview display [8] to show the amplitude envelope of the musical track, see Figure 6. The overview display allows for detection of breaks and other structural changes in the music, and is often used in wave editors [20].

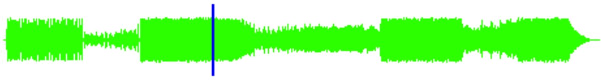


Figure 6: Overview display showing the amplitude en› velope of a musical track. The vertical line shows the current play position.

In the Beat

Based on observations from the field study, where it was found that almost half the time used by a DJ playing live was related to beat matching and that beat matching seems to require a high cognitive workload, we chose to design tools to perform beat matching and other manual synchronization tasks automatically.

By use of beat information, (i.e. rhythmic structure, exact placement of the first beat, and beat interval) automatic beat matching of dance music is possible. We experimented with techniques for automatic extraction of beat information [14, 13]. In Figure 5 the marks shown at the edges of the waveform display is based on this automatic extraction. Since it is very difficult, if not impossible, to design a beat extraction algorithm that will *always* be exact, the ability to use manually entered beat information was added to Mixxx.

A number of features were implemented, centering on the availability of correct beat information:

Automatic beat matching. This feature was implemented by setting the time scale factor and position of the track to match the tempo and beat position of the other playing track, when a button was pressed.

Beat synchronized loops. Instead of using the manual loop feature where the DJ has to exactly time the length of the loop, we implemented a feature that looped the four bars that were currently playing when a loop button was pressed. To allow a similar level of expressivity, when compared to the manual looping functionality, this function could be extended with a scale parameter that would allow control of the loop length.

Beat synchronized seeking. Seeking is done in Mixxx by clicking on the waveform overview display. Beat synchronized seeking means that the requested position is adjusted to be in phase with the music currently playing. Using this feature it is possible to seek in the file during playback without any noticeable disturbances in the perceived rhythm.

BeatFilters. We devised a new sound effect based on adding a time varying envelope to the sound. The form and phase of the envelope could be changed to form a time localized filter around each beat. The use of BeatFilters is shown in Figure 7 where the curved lines represent the BeatFilter envelope. The BeatFilter is essentially a time localized filter and is useful, for instance, to remove or amplify percussive instruments in the music.

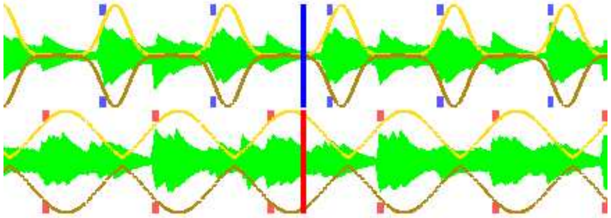


Figure 7: Beat marks and beat lters.

Music track selection and navigation

The music track selection interface is designed based on how DJs use their record collection today. Since the number of records they bring with them are fairly limited, a way to organize the tracks into lists is offered. A list could thus hold the tracks used for a DJ set. A track is selected by clicking on the list entry and selecting playback device from a popup menu. More advanced ways of pre-listening and selecting tracks could be provided. However, since the DJs know their music very well we felt that this was not necessary.

EVALUATION

An experiment with nine male DJs, seven professional and two amateurs was conducted. All participants also participated in the interviews on work practices. The amateur DJs were skilled DJs and had played live many times. The goal of the experiment was to assess the usability of Mixxx and to get an indication of how the automatic beat matching features compared to manual mixing. The DJs were asked to use Mixxx with two different interface mappings, and after a trial period with each mapping, make a small mix of 10 minutes with each. During and after the testing the DJs were interviewed and asked how they compared the setups with their own equipment. The tests and interviews were recorded on video and used in the following analysis.

Setup

For the experiment we used Mixxx in combination with the Hercules DJ console as shown on Figure 8. For both mappings we used the Hercules DJ console that provides physical controls for both mixing and playback. The two different mappings are shown on Figure 9. The first mapping was constructed in such a way that it resembled the traditional setup, where pitch and phase were adjusted manually, and option to perform automatic beat matching was provided. The second mapping, centered on beat information, and provided automatic beat matching, beat filters, beat synchronized loops and seeking. However, no options for adjusting the pitch independently for each playback device were provided. The graphical user interface (skin) that we used had a very limited set of input widgets since most of the buttons needed to operate Mixxx was found on the physical interface. Due to a problem with the USB sound interface and the Linux kernel used in test the latency adjustment was set to 50 ms. In general, 50 ms is too high for musical applications.

Ten tracks of popular house music were used in the test. The tracks were selected based on their popular style and that they should be easy to beat mix. We did not expect the DJs to know the music in advance.

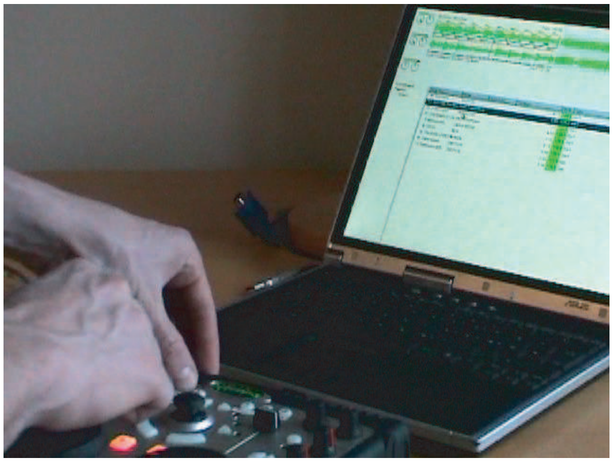


Figure 8: Experimental setup with Mixxx and the Her› cules DJ console.

Results

All participants were able to use Mixxx with both mappings after a short demonstration lasting a couple of minutes. Most errors were made with the manual mapping by all DJs, but still the manual mapping was preferred over the automatic by two DJs.

Visual displays. The two amateur DJs extensively used the visual displays, looking at the displays rather than at the console while mixing. Two of the professional DJs also looked at the visual displays while mixing, but the rest mostly looked at the console while mixing. All DJs commented that the waveform overview displays were useful, especially since they did not know the music. Some commented that part of this information could also be obtained from looking at a vinyl record but was missing from most CD players.

Cue points. Two of the professional DJs where annoyed by the latency, especially when using the cue point. The cue point was used by these DJs to start the track in phase with the other song, but this was not possible because of the high latency.

Mixing. All participants were satisfied by the mixer, even though most liked to use a slider for adjusting the volume, rather than a knob.

Jog wheels. All DJs made comments on the sensitivity of the jog wheels. The general comment was that it would take some time to get used to because of the high sensitivity compared to touching a turntable or a CD jog wheel. When asked, some commented that both the sensitivity used in the mapping and the lack of resistance in the jog wheels were a problem.

Beat synchronized loop. Three DJs used the looping functionality, and all commented that they found it very useful and liked it more than manual looping. Three commented that a way to adjust the loop length was needed.

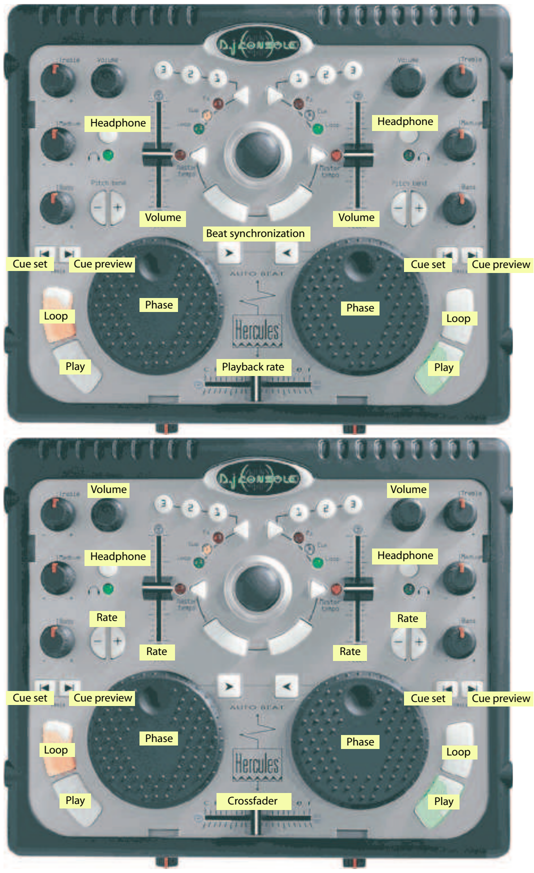


Figure 9: The two different mappings of the Hercules DJ console used in the experiment: automatic beat matching (top) and manual (bottom).

Beat synchronized seek. Beat synchronized seeking was only used by one DJ. Several commented on its usefulness. One DJ pointed out that he liked the idea but would never use it unless he could prepare seek points in advance. Clicking on the waveform display to seek live was too risky, as it could sound bad even when the beat was maintained.

Tempo adjustments. Using the manual mapping, three of the DJs complained that the pitch slider was not precise enough. Comparing the slider to a DJ turntable or CD player, the slider is half the length of these players and thus did not have the required precision. However, the DJs were able to reach a precise adjustment in combination with the pitch change buttons. Seven of the DJs liked the automatic beat synchronization feature. The beat synchronization resulted in accurate tempo scaling but lacked correct adjustments of the phase of the beat. Automatic phase adjustment was avoided in the design because we thought that the phase was an important creative parameter in the mix. It turned out that all DJs wanted an exact phase adjustment, and they very seldom mixed records that deviated from this adjustment.

Beat filters. The temporal beat filtering was introduced to all DJs when using the automatic mapping. Six of the DJs tried to use it. Of those, all found it useful and four found it particularly exciting. Two of the DJs that did not try the beat filters commented that they found it useful based on watching its use during the demonstration.

Unintended use. Two of the professional DJs used Mixxx in ways it was not originally designed for. The version of Mixxx used in this experiment used a pitch independent time stretch algorithm to change the tempo of the music. The sound quality of the algorithm was distorted when playing at very low speed. One DJ used this artifact as a sound effect. In another session, testing the public available version of Mixxx, one of the DJs used it with a looping sound and found a bug in the looping that would cause Mixxx to start at random points in the loop. The DJ liked the sound and wanted to use it live. This could be achieved by adding a stochastic parameter to the loop controls.

Discussion

The evaluation of Mixxx was successful in that the mapping offering automatic beat matching was preferred over the manual mapping by seven out of the nine DJs. The DJs that liked the automatic mapping were also positive towards using computers for live DJing. Providing tools for automatic synchronization, such as beat matching and looping, was a successful way of taking advantage of the computerized DJ setup. The setup was liked despite the lack of control (e.g. lack of precision in pitch sliders and high latency) when compared to the traditional setup. The visual overview displays were used and liked by all DJs, and the automatic zooming displays for aiding in beat matching were attended to by some DJs, especially those that were not as skilled in beat matching as were the professionals.

CONCLUSIONS

We presented a study on work practices of dance DJs. The study complements the descriptions of work practices found in the literature with quantitative data on time spent in task. The data was based on video recordings of professional DJs playing live. Observations obtained from the video recordings were verified in interviews with nine professional and two amateur DJs. We found that the time spent on each task did not vary significantly across the observed DJs. Surprisingly to the DJs, the time spent in beat matching was on average 47% of the time used to DJ. Nine out of the ten DJs interviewed regarded beat matching as a manual task that they would rather be without, and three of the DJs actively tried to lower the burden of beat mixing. The small amount of time used on social activity or passivity indicates a high cognitive workload. When engaged in social activity such as conversation, the conversation was usually not initiated by the DJ, and was usually finished in less than 10 seconds.

Based on this study we designed the DJ software Mixxx, building upon a metaphor of the traditional DJ setup with playback devices and a mixer. We presented a detailed overview of the features implemented in Mixxx and the design rationale behind each of them. To DJs, a major difference between Mixxx and the traditional setup is the ability to do automatic beat matching based on meta-data describing the rhythm and beat. This feature was provided to reduce the cognitive workload and free time for creative aspects of mixing.

Mixxx was evaluated by nine DJs. We found that seven of the nine DJs preferred a mapping that allowed for automatic tempo adjustment even though it was not possible to adjust the tempo of each player individually. All DJs made fewer errors with the automatic mapping, but were able to use both mappings.

The approach chosen here, with adding information to the interface rather than exploring tangible and direct manipulation input techniques, was successful in that the majority of the nine participants preferred the interface mapping where certain tasks could be accomplished automatically. Exploring the use of tangible interfaces in combination with this approach could further improve the design and interaction of future DJ systems.

ACKNOWLEDGMENTS

I would like to thank Vega Nightclub, in particular Thomas Bertelsen, for allowing the video recordings of the DJs. I also thank the DJs participating in the study: August Engkilde, DJ TOM, DJ Kruzh'em, Nima, Matt Vega, DJ Wunderbaum, DJ Soulitude, Andreas Hald, Martin Swanholm, Ralph Andersen, and others. The open source developers participating in the development of Mixxx: Ken Haste Andersen, Kenny Erleben, Kristoffer Jensen and many others. And finally Georg Strøm, Kasper Hornbæk, Kristoffer Jensen and other members of the HCI and Music Informatics groups at DIKU for helpful feedback and comments.

REFERENCES

1. Ableton live. http://www.ableton.com/. Accessed March, 2005.
2. Finalscratch. http://www.finalscratch.com/. Accessed March, 2005.
3. Traktor. http://www.native-instruments.com/. Accessed March, 2005.
4. T. H. Andersen and K. Erleben. Sound interaction by use of comparative visual displays. In *Proceedings of the Second Danish HCI Symposium*. HCØ Tryk, November 2002. Extended abstract.
5. T. Beamish, K. Maclean, and S. Fels. Manipulating music: multimodal interaction for DJs. In *Proceedings of CHI*, pages 327 – 334, 2004.
6. H. Beyer and K. Holtzblatt. *Contextual Design*. Morgan Kaufmann Publishers, 1998.
7. F. Broughton and B. Brewster. *How to DJ Right*. Grove Press, 2002.
8. S.K. Card, J.D. Mackinlay, and B. Shneiderman, editors. *Readings in Information Visualization*. Morgan Kaufmann Publishers, 1999.
9. K.F. Hansen. The basics of scratching. *Journal of New Music Research*, 31(4):357–367, 2002.
10. T. Igarashi and K. Hinckley. Speed dependent automatic zooming for browsing large document. In *Proceeding of the Symposium on User Interface Software and Technology*, pages 139–148. ACM Press, 2002.
11. Free Software Foundation Inc. GNU general public license. http://www.gnu.org/copyleft/gpl.html, June 1991. Version 2.
12. H. Ishii and B. Ullmer. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *Proceedings of CHI*, March 1997.
13. K. Jensen and T. H. Andersen. Beat estimation on the beat. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New York*, October 2003.
14. K. Jensen and T. H. Andersen. Real-time beat estimation using feature extraction. In *Computer Music Modeling and Retrieval: International Symposium, Montpellier*, Lecture Notes in Computer Science, pages 13–22. Springer Verlag, 2003.
15. W.E. Mackay, A. Ratzer, and P. Janecek. Video artifacts for design: Bridging the gap between abstraction and detail. In *Proceedings of DIS 2000, Conference on Designing Interactive Systems*. ACM Press, 2000.
16. H. Newton-Dunn, H. Nakano, and J. Gibson. Block Jam. In *Proceedings of the SIGGraph*, 2002. Abstract.
17. O. Parviainen. SoundTouch audio processing library. http://sky.prohosting.com/oparviai/soundtouch/. Accessed March, 2005.
18. J. Patten, B. Recht, and H. Ishii. Audiopad: A tag-based interface for musical performance. In *Proceedings of the Conference on New Interfaces for Musical Expression*, 2002.
19. J. Raskin. *The Humane Interface*. Addison-Wesley, 2000.
20. S.V. Rice and M.D. Patten. Waveform display utilizing frequency-based coloring and navigation. U.S. patent 6,184,898, 2001.
21. B. Shneiderman, editor. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley, Reading, MA, 2nd edition, 1992.
22. F. Vernier, N.B. Lesh, and C. Shen. Visualization techniques for circular tabletop interfaces. In *ACM Advanced Visual Interfaces (AVI)*, May 2002.

Chapter 4

Feedback in pen-gesture interfaces

4.1 Exploring the use of auditory feedback in pen gesture interfaces

Submitted for publication. T. H. Andersen and S. Zhai. Exploring the use of auditory feedback in pen gesture interfaces. 24 pages. 2005.

“Writing with Music:” Exploring the use of auditory feedback in pen gesture interfaces

TUE HASTE ANDERSEN
University of Copenhagen
and
SHUMIN ZHAI
IBM Almaden Research Center

We investigate the use of auditory feedback in pen gesture interfaces in a series of informal and formal experiments. Initial iterative exploration showed that gaining performance or learning advantage with auditory feedback was possible using absolute cues and state feedback after the gesture was produced and recognized. However, gaining learning or performance advantage from auditory feedback tightly coupled with the pen gesture articulation and recognition process was more difficult. To establish a systematic baseline, Experiment 1 formally evaluated gesture production accuracy as a function of auditory and visual feedback. Size of gestures and the aperture of the closed gesture were influenced by the visual or auditory feedback, while other features were not. Experiment 2 focused on the emotional and aesthetic aspects of auditory feedback in pen-gesture interfaces. Participants' rating on the dimensions of being wonderful and stimulating was significantly higher with musical auditory feedback. Our exploration points to several general rules of auditory feedback: a few simple functions are easy to achieve, gaining further performance and learning advantage is difficult, the gesture set and its recognizer can be designed to minimize visual dependence, and positive emotional or aesthetic response can be achieved using musical auditory feedback. These rules may serve as a foundation for future research and development in pen-gesture interfaces with auditory feedback.

Categories and Subject Descriptors: H.5.2 [**Information Interfaces and Presentation**]: User interfaces

General Terms: Experimentation, Human factors

Additional Key Words and Phrases: Audio, auditory interface, sound, music, gesture, pen, text input, feedback.

1. INTRODUCTION

This paper deals with issues at the intersection of auditory feedback and pen-gesture interfaces. Due to the adoption of mobile forms of computing, which lack mouse and keyboard-based input capabilities and large visual screens, research on these two topics is increasingly important.

Developing auditory interfaces has long been a captivating research topic in human-computer interaction [Buxton 1995]. As computing moves beyond the desktop to mobile and other pervasive computing forms, which often have a small screen that should not demand the user's constant visual attention, effective auditory feedback in user interface has become more necessary than ever.

Effective auditory interface is both an intriguing and difficult research topic. Successful examples of sonification, such as monitoring patients breath or heart beat by auditory interfaces [Loeb and Fitch 2002], does exist. Auditory feedback in common user interfaces, however, rarely goes beyond the simplest and most obvious forms, despite the decades of research. There can be many reasons for this state of affairs, including the dominance of the visual metaphors in interface design practice, the unique characteristics of auditory perception, and the challenge of mapping interaction tasks to the appropriate auditory dimensions.

The current study focuses on coupling sound patterns with pen-gestures, particularly those pen-gestures such as Graffiti, Unistroke [Goldberg and Richardson 1993], and SHARK Shorthand [Zhai and Kristensson 2003]. Several observations motivated us to focus on sound and gestures. First, pen-gesture interfaces are increasingly popular in mobile and other non-desktop computing devices. Second, it is highly desirable to minimize the visual demand of pen gestures in mobile computing environment [Goldberg and Richardson 1993]. Third, auditory modality is particularly sensitive to rhythm and patterns [Clarke 1999] that pen gestures often produce. For example, although it is difficult to tell which tone corresponds to which digit in modem dialing, one could more easily tell if the entire sequence is correct based on the overall rhythm. Another example is the game "Simon," in which the sound pattern may help the player memorize the previous sequence of keys. The powerful coupling between sound and kinesthetic patterns is most evident in musical activities such as dancing and instrument playing. It is difficult to imagine dancing without sound patterns (music). In the case of many musical instruments, the kinesthetic spatial movements of the musicians and the temporal sound patterns produced are inseparable. However, we should be aware that musical skills are acquired over intense and longitudinal practice, which is usually unacceptable to user interfaces, unless such an acquisition is implicit and gradual [Kurtenbach and Buxton 1994; Kristensson and Zhai 2004], or when the skill is necessary and unavoidable, such as touch typing skills on QWERTY keyboard.

Our research questions are whether and how sound feedback can be created to make pen gestures: 1. more memorable or learnable, 2. more accurately produced for reliable recognition, and 3. more fun, pleasant or engaging to use.

2. RELATED WORK AND INITIAL ITERATIVE EXPLORATION

Whether and how sound feedback can improve interaction with pen-gesture interface requires an exploration of a vast design space. We started this research with a combination of literature analysis and iterative cycles of design and informal testing. In the following sections we summarize literature on gesture set design for pen-gesture interfaces and its relation to feedback, followed by explorations of visual and auditory feedback in such interfaces.

2.1 Designing gesture sets for reproduction with limited feedback

Motivated by the ease of machine recognition, pen gesture-based text input has been dominated by novel alphabets, following the design of the Unistroke alphabet by Goldberg and Richardson [1993]. Unistroke characters are faster to write and easier to automatically recognize when compared to the traditional Roman alphabet. One of the critical design rationales of Unistroke was that single stroke alphabets support eyes-free operation, because the letters can be written on top of each other.

Many other novel alphabets have later been presented, including Graffiti and EdgeWrite [Wobbrock et al. 2003]. EdgeWrite further tries to simplify recognition by using a position dependent recognition system. To compensate for the need to reach absolute positions physical edges around the writing area is provided. The system is designed to support users with certain disorders or improving support for heads-up situation, and it is shown that EdgeWrite have slightly better performance when compared to Graffiti [Wobbrock et al. 2003].

2.2 Visual feedback

Digital ink, the pen trace shown on the screen during gesture articulation, is the most common form of feedback that may aid user's performance and learning. With less or no visual feedback people's writing traces are typically larger [van Doorn and Keuss 1992; Legge et al. 1964], and the absolute positioning of consecutive words on a horizontal line is less accurate [Smyth and Silvers 1987]. Clearly these effects may degrade people's writing quality on paper. In the case of character recognition on a pen-based computing device, however, the relative size or the absolute positioning of each letter may not be important as realized by Goldberg and Richardson [1993]. The lack of visual feedback may also have other effects on writing. Smyth and Silvers [Smyth and Silvers 1987] further examined writing errors in the number and order of strokes in producing letters and found that the number of errors increased when participants either had to perform a secondary task such as counting, or when no visual feedback was available. High level processes such as formulation of text, has been shown not to be influenced by visual feedback [Olive and Piolat 2002].

On the other hand, Teulings and Schomaker [1993] argue that feedback in general is too slow to be used for correction during fast handwriting. In handwriting, the time it takes for writing a sub stroke is 100 ms [Smyth and Silvers 1987], and reaction times for light stimuli, when measured as EMG potentials in arm muscles, is 125 ms [Luschei et al. 1967]. This means that there may not be sufficient time to make corrections based on visual feedback. The reaction time measured for audio signal in the same study by Luschei et al. was 80 ms.

2.3 Auditory feedback

Early guidelines on audio signal design can be found in Deatherage [1972]. In general, absolute position and spatial information are more difficult to represent in audio than in visual display [Welch 1999]. In contrast, time critical information and short messages can often be more effectively conveyed in audio than in visual displays. In the more recent HCI literature, a number of different paradigms for adding sound to interfaces have been proposed, most notably Auditory Icons [Gaver 1989] and Earcons [Blattner et al. 1990]. Both of these approaches have been primarily concerned with providing feedback about the state of a system or the product of an interaction transaction.

2.3.1 *Gestures and product feedback.* Providing audio feedback on the product (namely the end result) of gestural interaction is relatively easy. Pirhonen and Brewster [2002] used five basic hand gestures articulated on a touch pad to control a media player. Reduction in workload and task completion time was reported [Pirhonen et al. 2002] when compared to having no feedback. Using the SHARK shorthand-on-stylus keyboard text input system [Zhai and Kristensson 2003; Kristensson and Zhai 2004] as an example, we coupled the word SHARK Shorthand recognized to a speech synthesizer, making the system pronounce each recognized word. With such a function the user did not have to look at the screen to confirm whether the system has recognized the user's gesture as the intended word, hence saving the user's visual attention on the interface. Another type of simple and useful feedback we explored was auditory warnings when the pen deviates from the expected writing area. This type of feedback may complement or replace haptic feedback such as physical edges. Both of these auditory feedback mappings worked well and increased the usability of SHARK Shorthand in a heads up situation. We also tested the use of auditory feedback coupled to pen position, when the pen was just above the tablet, before making contact. Each letter on the stylus keyboard was associated with a unique instrumental sound. This could help the user in finding the right start position before the pen made contact with the tablet, hence enabling more accurate start position of the pen without looking at the stylus keyboard. However, we found that hearing sound feedback before the stylus contacts the surface could be both annoying and fun since it felt like playing a music instrument. To use such a scheme in practice, the sound level and number of sounding events have to be carefully adjusted according to the use situation.

2.3.2 *Feedback during articulation: Process feedback.* To couple sound feedback to the process of pen-gesture articulation is both more intriguing and more challenging. Continuous gesture production process (or state information) can be coupled directly to auditory feedback resulting in auditory patterns. These patterns might be recognizable by the user in the same way that modem dialing tones are recognized when dialing a familiar telephone number. One knows when a wrong button is pressed when dialing a familiar number, and thus the auditory feedback serves as a pre-attentive reference [Woods 1995], that is, a perceptual cue that is only consciously attended to when necessary. By providing auditory feedback that has pre-attentive references, and thus supports pre-attentive monitoring [Broadbent 1977; Woods 1995], it may be possible to respond to errors made during articulation when the audio signal deviates from its usual pattern in the given context, but without attending the auditory signal when it stays within its pattern for the given context. A response to a deviating audio pattern could be a corrective action or an early termination of a command.

There is some evidence in the literature suggesting that it is possible to gain performance from a tight coupling of sound with gesture production. One example in this regard came from Ghez and colleagues [Ghez et al. 2000] who, in an experiment, added continuous sound feedback to the limb movements of participants who lack proprioception. The sound used was continuous complex tones and melodic sequences with varying amplitude and pitch. It was found that with joint rotation and timing information encoded as auditory feedback, the participant was able to perform a certain arm movement synchronous with a musical beat, with the same timing and precision as when having visual feedback only. This research suggests that providing auditory feedback at the process level may help to reinforce movements used to produce certain gestural patterns.

Another example supporting the use of auditory feedback with gesture production is Brewster et al. [2003] in which a media player was controlled with a set of 12 multistroke hand gestures articulated on a touch pad. The recognizer was based on a coarse positional feature, and feedback was given during articulation by playing notes depending on the finger's position. The accuracy and workload were evaluated in a mobile usage situation. They found that accuracy was improved with auditory feedback but no difference in workload was observed [Brewster et al. 2003].

2.3.3 *Learning with auditory and geometric patterns.* One possible gain of coupling sound with gesture movement is to help the user remember the gesture better. The sound patterns therefore have to be highly discriminable. We first coupled the articulation of the SHARK shorthand (i.e. the pen movement in the writing area) with a continuous complex tone. The amplitude and frequency of the tone were mapped to the horizontal and vertical position of the pen tip respectively. This produced discriminable tones for different gesture patterns. However, as in many audio interface research prototypes, the sound produced by this mapping was not pleasant, and it was difficult to imagine prolonged use with this type of feedback. A few design iterations led us to a different approach that mapped the direction of the pen movement to a discrete space of sampled musical instrument sound. Moving the pen in each direction would result in a unique instrument sound. A gesture of a particular movement pattern, such as a triangle shape, sounds distinctly different from another, such as a squared shape. This scheme was both enjoyable and discriminable.

To test whether this audio feedback method could help in learning new pen gestures, we tested it with a blind participant using a digital tablet as the pen sensor. The participant was a computer scientist familiar with user interface technologies. A dozen pen gesture patterns were demonstrated to the participant by either holding his hand or letting him trace trajectories carved on a piece of cardboard overlaid on the digital tablet. The participant then practiced reproducing these gestures. It was evident that he had difficulty with memorizing these gestures in a short period of time. The blind person was chosen for this experiment because of the belief that a blind person has more developed hearing capabilities and thus would benefit more from the auditory pattern feedback than would a sighted participant. However, during the experiment he tended to memorize the gestures' spatial rather than sound patterns. Considering that we first showed the gestures to him by spatial information (e.g. tracing carved templates), we complemented that study with sighted users in which more emphasis was placed on the audio channel in the initial introduction of each gesture. We designed a number of basic pen gesture patterns consisting of straight lines of four directions only (up, down, left, and right); each movement direction was associated with one musical instrument. Participants were acquainted with these mappings first. Each gesture pattern was then presented to the participants by playing the corresponding sound pattern, rather than by visual appearance. At first it was hard to learn the gestures from the sound; it took many trials for the participants to figure out the gesture pattern from the sound pattern played. When the pattern was produced correctly, the participants tended to again remember the pattern by its spatial shape (e.g. a square) rather than its corresponding sound. These two informal experiments suggest that the improvement in memorability of gestures from sound feedback is unlikely to be immediate or dramatic. It is possible that people with musical experience may gain more advantage in this regard, but in light of the typical learning time of several years to master an acoustic instrument the finding may not be surprising.

2.4 Discussion of the initial exploration

From these explorations it became clear that learning an audio mapping coupled to the process of pen-gesture articulation is difficult. As it is with most musical instruments, this mapping takes long time and extensive practice to learn. Gaining any improvement in performance (speed or accuracy) from this type of audio mapping would not be easy either. Careful design and considerations have to be done, and even so it is not clear if any improvement can be expected given the argument that feedback is too slow to be utilized in the process of handwriting [Teulings and Schomaker 1993]. Not only the audio feedback, but also feedback in general, seems problematic in terms of performance gain. Clearly the contributions of the two types of feedback in pen gesture production require a more formal and systematic study, so future researchers do not have to repeat the explorations we have done, or at least can start at a higher level of understanding.

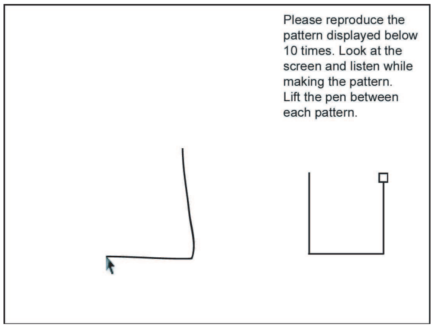
Our exploratory studies also suggest various positive benefits of auditory feedback. For example, it is easy to gain performance benefit from auditory feedback that gives simple information such as crossing the border of the writing area, or informs on the product (result) of the pen gesture, such as pronouncing the recognized word in SHARK Shorthand. Our exploration also showed that it is possible to employ audio to enhance users' subjective experience, making pen gesture input more fun and more enjoyable, which is an increasingly important goal of interaction experience design [Norman 2004]. Accordingly we pursued two formal studies, one focusing on a baseline performance comparison on pen-gesture production with the presence or absence of auditory or visual feedback as the experimental manipulation, the other focusing on users' subjective reaction of aesthetics and joy to different types of sound feedback of pen gestures.

3. EXPERIMENT 1: THE PERFORMANCE IMPACT OF VISUAL AND AUDITORY FEEDBACK ON PEN GESTURE PRODUCTION

This experiment aims at providing baseline measurements on the impact of visual and auditory feedback on pen-gesture production. Auditory and visual information were manipulated when the participants drew a set of arbitrary gesture patterns on a tablet digitizer. The patterns were practiced using both visual and auditory feedback, and later reproduced in four different feedback combinations. The difference between each articulated gesture and the template pattern were compared across the four conditions. The visual feedback used was the pen trace, and auditory feedback gave information about relative position and speed of movement. The audio was designed to be easily discriminable.

3.1 Task

The experimental task was to reproduce a set of gesture patterns as accurately as possible at a speed comparable to writing a character with Graffiti. For each gesture pattern, the participant was first shown its shape on screen, as illustrated in Figure 1. The participant then practiced drawing the pattern with both visual and auditory feedback. After practicing the gesture enough times to be confident in reproducing the gesture without looking at the template, the participant was asked to reproduce the gesture 10 times in various feedback conditions, resulting in



Please reproduce the pattern displayed below 10 times. Look at the screen and listen while making the pattern. Lift the pen between each pattern.

Fig. 1. Screenshot (illustration) of the software used in baseline experiment.

each participant producing a total of 800 gestures discounting the practicing trials. Instructions about the current condition were given verbally by speech synthesis in combination with written instructions visible on the screen. For the conditions without visual feedback the participants were instructed to close their eyes so that they could not receive visual feedback by looking at their hand. In the conditions where no auditory feedback was given, a noise signal was played to the participant's headphone to mask the sound produced when moving the pen over the surface of the digitizer tablet.

A set of 20 gestures of varying complexity, as shown in Figure 2, was used. Half of the gestures start and end in the same position – we call these closed gestures. The other half was open gestures. During the experiment the position of the pen tip was recorded when it was in contact with the tablet, along with timing information. A total of nine people, five men and four women, from 25 to 43 years of age, participated in the experiment.

3.2 Conditions

The reproduction of the gestures was performed in the presence or absence of visual and auditory feedback, resulting in four (2x2) conditions:

(1) Visual and auditory feedback (V+A)

(2) Visual feedback, no auditory feedback (V)

(3) Auditory feedback, no visual feedback (A)

(4) No visual or auditory feedback (None)

The order of the four conditions was randomized across participants.

3.2.1 *Visual feedback.* Providing visual feedback was straightforward: the pen trace was displayed on the screen, as illustrated in Figure 1. The user could see the pen movement as well as its entire history instantaneously. The participants were instructed to look at the writing area of the screen, rather than at the tablet. The pen trace disappeared as soon as the pen was lifted from the tablet. The pen trace of the ideal gesture was shown at one side of the screen but was only attended by the participants during practice.

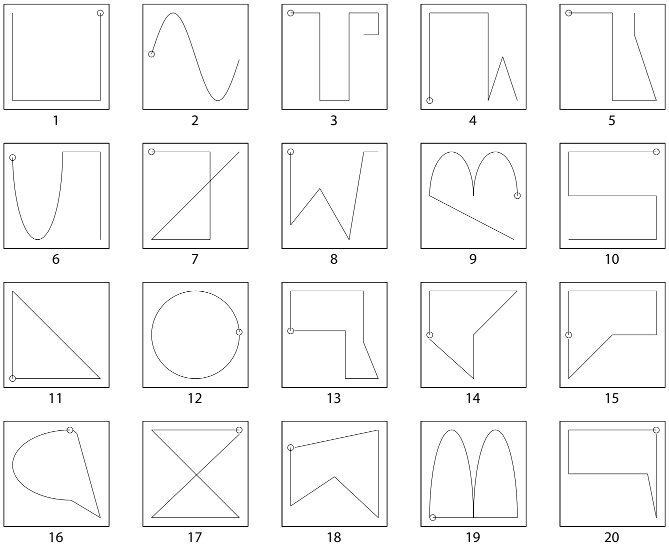


Fig. 2. Gestures used in experiment. The small circle signifies where the starting point of the gesture. Note that gesture 11-20 start and end at the same point.

3.2.2 *Auditory feedback.* Since the purpose of this experiment was to measure the strongest possible feedback impact on gesture reproduction accuracy, we made the auditory feedback as easily discriminable and as instantaneous as possible, without considering the aesthetic aspects. Based on our experience in the exploratory phase of this research, we chose a continuous sound feedback method which mapped the pen's position relative to its starting point to a number of perceptually sensitive parameters. Direction and speed of the pen movement are also used to take full advantage of the auditory modality.

As basis of the synthesis a complex tone was chosen because of its well identified perceptually important parameters [Kramer 1994]. Using a complex tone it is possible to create auditory feedback that is easily dicriminable on a number of dimensions. A complex tone is a tone composed of a number of sinusoids of varying frequencies added together over time. Each sinusoid in the complex tone is called a partial. If the distance in frequency between each partial is constant, and equal to the frequency of the first partial, the complex tone is said to be harmonic. The fundamental frequency of the complex harmonic tone is the frequency of the first partial. The fundamental frequency determines the perceived pitch of the tone, and is one of the most important perceptual parameters for tones [Plomp 1964]. If the difference between the partial frequencies is not constant the sound is inharmonic. If the change in partial frequency difference is systematic, proportional with the frequency of the partial, the tone is said to be quasi-harmonic, as is the case with tones produced by a piano [Fletcher et al. 1962]. Another perceptually important parameter is loudness or amplitude of the complex tone. For most sounds produced by acoustic instruments the amplitude of the first partial is largest, followed by a decrease in amplitude as a function of partial frequency. A complex tone with many partials sounds more rich compared to a tone with only one or a few partials. A tone with only one partial is a sinusoid. In user interfaces it is most practical to use sounds with many partials because they are less likely to be masked by other sounds in the environment [Patterson 1989], and thus is easier to perceive.

Amplitude was mapped to the speed of the pen, so that a fast pen speed created a loud tone. The mapping added rhythmic structure to the sound, and made the sound less intrusive when the pen was at a stop. Vertical position of the pen was mapped to the fundamental frequency of the tone: a higher tone was played if the pen was at the upper part of the tablet and a lower tone was played if the pen was at the lower part of the tablet. The horizontal position was mapped to the number of overtones, creating more pure tones on the left and richer sound on the right. Jitter, i.e. random frequency variations of each partial, and inharmonicity [Fletcher et al. 1962] were used to further increase perceptual difference in the horizontal dimension. Another perceptual parameter, brightness, was not used since it is known to be confused with pitch [Singh 1987]. The complex tone can be described as the sound pressure level s at the discrete time t by the following function:

$$s(t) = \frac{1}{n} \sum_{n=1}^{M} \sin(UF_0tni), \tag{1}$$

where M is the number of partials, U is an evenly distributed random number between 0.9 and 1.1, representing the jitter, F_0 the fundamental frequency, and i is inharmonisity.

Another perceptual parameter used to encode the horizontal position of the pen was the stereo sound source. If the pen was moved to the left of the starting point, the complex tone was perceived as coming from the left, whereas when it was moved to the right, the tone was perceived as coming from the right. The perceptual direction of the sound source was created by filtering the resulting complex tone with a Head Related Transfer Function obtained from [Algazi et al. 2001], using a Finite Impulse Response filter.

The resulting auditory feedback used in this experiment was thus designed to make different spatial patterns highly discriminable in the auditory space.

The total latency of the system, from pen movement to a change in the sound output, was about 31 milliseconds. For pen input a Wacom Graphire table was used with a latency around 28 milliseconds. The latency of the sound synthesis was about 3 milliseconds. The low sound synthesis latency was possible due to a low latency patched Linux kernel with ALSA sound drivers [MacMillan et al. 2001].

3.3 Results and Analysis

The dependent variables we extracted from the pen traces in the experiment included the following:

(1) Size of the gesture

(2) Speed at which the gestures were articulated

(3) Aperture: the Euclidean distance between the start and end position

(4) Shape distance between the articulated gesture and the template pattern

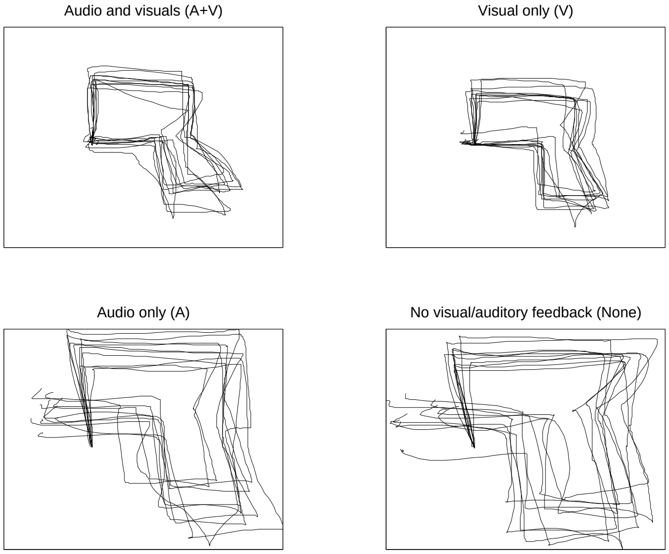


Fig. 3. Examples of gestures articulated by a participant in all four conditions. The gestures are plotted on the same scale with start positions alligned for each condition.

(5) Directional difference between the articulated gesture and the template pattern

Computationally, the template pattern was represented as a set of consecutive spatial points in the horizontal plane, $p(n), n = 1 \ldots M_p$, where M_p was the number of sampled points in the pattern. The length l of a pattern p was defined as the sum of the Euclidean distance between these points:

$$l = \sum_{n=1}^{M_p-1} \sqrt{(p(n+1)_x - p(n)_x)^2 + (p(n+1)_y - p(n)_y)^2} \tag{2}$$

where $p(n)_x$ and $p(n)_y$ denotes the x and y coordinate of the n'th point in p respectively. Articulated patterns, g, were defined in the same way as template patterns, but additionally timing information was recorded. When computing measure 3 to 5, the size of the articulated pattern and the template pattern were first normalized to the same scale and their start positions were aligned.

Within-subject repeated measurement ANOVA analysis was conducted on the five dependent variables, with the feedback conditions as the independent variable with four levels: V+A, V, A, None. We further performed Least Significant Difference (LSD) post hoc pairwise comparisons between these levels. Alternatively, we also treated audio feedback and visual feedback as two independent variables each with two levels (presence/absence) and evaluated their main effects and interaction. In general there was no difference in conclusions in the two analysis approaches, and we only report the former.

An example of the gestures made by a participant in the four feedback conditions is shown in Figure 3. The example illustrates some of the findings in the statistical

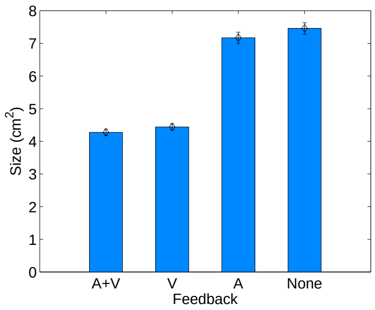


Fig. 4. Average gesture size for each condition.

analysis described in the following subsections.

3.3.1 *Size.* The size of a gesture is defined as the area in cm^2 of the bounding box of the gesture. This variable changed with the feedback conditions significantly (Figure 4): $F_{3,24} = 12.5, p < 0.001$. Post hoc tests showed significant differences between the visual conditions (V, V+A) and the non-visual conditions (A, None) at $p \leq 0.01$ level. In handwriting literature the effect of feedback on writing size has been observed in the visual channel by van Doorn and Keuss [1992], who demonstrated that handwriting without visual feedback was larger than with visual feedback. In our experiment, pairwise comparison of the audio condition (A) with the no-feedback condition (None) was near significance ($p = 0.076$), although the magnitude was small (Figure 4). The difference between A+V and V was not significant.

3.3.2 *Speed.* The average speed v at which a gesture was reproduced was calculated by the ratio of the length l of the articulated pattern and the time spent in gesture production, Δt:

$$v = \frac{l}{\Delta t}. \tag{3}$$

The average speed of movement changed significantly with feedback conditions: $F_{3,24} = 9.632, p < 0.001$ (see Figure 5). The visual conditions (V+A and V) and the non visual conditions (A and None) were not significantly different from each other ($p >= 0.417$). All pairwise comparisons between visual and non visual conditions were significant at $p \leq 0.02$ level. Comparing these results with previous studies, it is surprising that we see an increase in speed when removing feedback. According to van Doorn and Keuss [1993] the participants slowed down when no feedback was provided. However, their study differed from ours in that the resulting handwriting was not erased after each symbol or letter was written. In pen based input systems the pen trace is usually removed after each gesture, and thus the aesthetic appearance of the letter does not matter to the user. The user might therefore speed up as a result of not having to produce aesthetically pleasing gestures.

The average time used to produce a gesture pattern was 1.21, 1.23, 1.30 and 1.32 seconds in the A+V, V, A, and None conditions respectively (Figure 6). Between

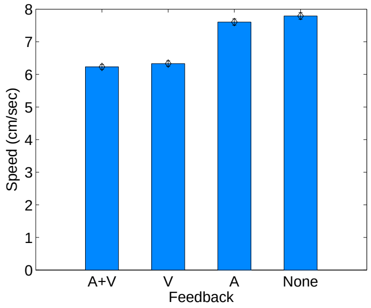


Fig. 5. Average speed of gesture production for each condition.

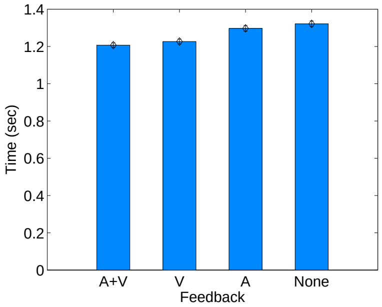


Fig. 6. Average gesture completion time for each condition.

visual and non visual conditions a significant difference was observed ($F_{3,24} = 5.638, p = 0.005$). However, the difference was smaller than the difference in speed and size. The larger gestures in the A and None conditions were compensated with faster pen movement. As a result we see a relatively constant production time for the gesture.

3.3.3 *Aperture.* For the 10 closed patterns, the Euclidean distance between the start and the end position of the articulated gestures indicated the ability to return to the same point under various feedback conditions. Figure 7 shows a template pattern and two articulated patterns. The pattern shown to the right has a large aperture between start and end position where the pattern shown in the middle of the figure has a relatively small aperture. Without closed-loop feedback, returning to a past point requires accurate reproduction of the order and direction of each sub-stroke (segment) as well as the relative proportions of the individual segments in a gesture. It reflects the same type of ability to cross the t's and dot the i's in handwriting research [Smyth and Silvers 1987].

Figure 8 shows the aperture results in the four conditions. A statistically significant difference was found: $F_{3,24} = 14.08, p < 0.001$. Post hoc tests revealed significant differences between all pairs of conditions at $p \leq 0.015$ level, except between V and V+A conditions. Both visual feedback and audio feedback helped

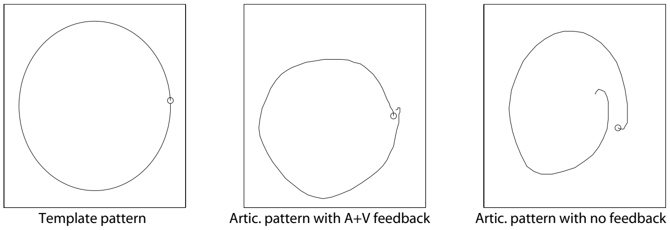

Template pattern
Artic. pattern with A+V feedback
Artic. pattern with no feedback

Fig. 7. Template pattern No. 12 (left). Example of articulation of the pattern with full feedback (middle) and no feedback (right).

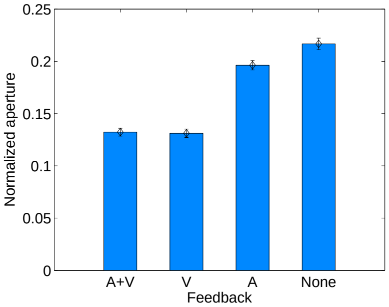


Fig. 8. Average aperture between start and end position of the closed gestures for each condition.

to reduce the gap between the start and the end positions, although the impact of audio was smaller and only significant when no visual feedback was present. One plausible cause that the audio condition (A) also helped reducing the aperture of the closed gesture was that we used stereo directional sound. One would hear the change of sound direction as soon as the pen tip passed its starting position in the lateral dimension.

3.3.4 *Proportional shape distance.* To reflect the overall match between the gestures and the templates, we measured "proportional shape distance" d, defined as the mean Euclidean distance between a point in the articulated gesture and a corresponding point in the template pattern. The distance measure is calculated by sampling the template pattern p to points $q(n), n = 1 \ldots K$ equally spaced along the length of the pattern, maintaining the order of p. The articulated gesture is sampled to the same number of points, $g(n), n = 1 \ldots K$, using a fixed interval between the points, again maintaining the order in which the gesture were articulated. The proportional shape distance d is defined as:

$$d = \frac{1}{K} \sum_{n=1}^{K} \sqrt{(q(n)_x - g(n)_x)^2 + (q(n)_y - g(n)_y)^2}, \quad (4)$$

where $q(n)_x$ and $q(n)_y$ denotes the x and y coordinate of the point $q(n)$ respectively, and $g(n)_x$ and $g(n)_y$ denotes the x and y coordinate of the point $g(n)$ respectively.

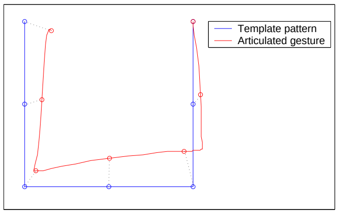


Fig. 9. Proportional shape matching. The corresponding sampling points on the two patterns are shown with connecting dotted lines.

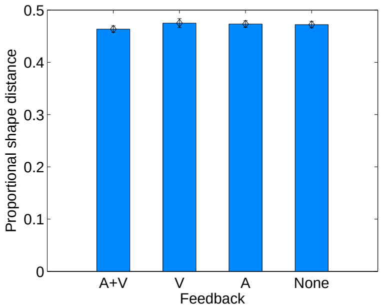


Fig. 10. Average proportional shape distance for each condition.

The method is illustrated in Figure 9. Sensitive to the overall shape proportion and direction of articulation, but not to referential features, this measure has previously been used as the most basic gesture recognition metric in complex multi-channel recognition systems such as SHARK Shorthand [Kristensson and Zhai 2004].

In the experiment, no significant difference was observed between the feedback conditions by the shape distance measure: $F_{3,24} = 0.626, p = 0.605$. (Figure 10).

3.3.5 *Directional difference.* Directional difference describes how well the directional movements in the articulated gesture match the directions of the lines in the template pattern, $\angle\vec{p}(n), n = 1 \ldots M_p - 1$. The measure preserves the order of the directional movements when comparing the directions of the two patterns, but the extent (length) of each movement is discarded. To achieve this, the directional difference is calculated based on segmentation of the articulated gesture into a list of directions, $\angle\vec{g}(n), n = 1 \ldots M_g - 1$. Because the two lists of directions are not necessarily of equal length ($M_p \neq M_g$), an element from one list can be compared with one or more elements from the other list. The mean directional difference between the two lists is the mean of the directional difference between the elements in the lists that results in the lowest mean distance. We used the lowest possible distance, since we are interested in the optimal match. In handwriting recognition systems, recognition is often done at the end of gesture articulation [Zhai and Kristensson 2003; Goldberg and Richardson 1993], and thus the best match approach

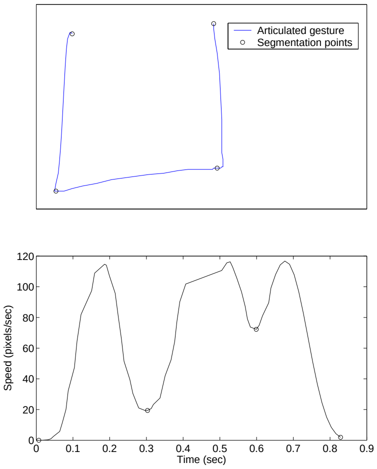


Fig. 11. The articulated gesture is shown (top), with the corresponding velocity profile (bottom). Segmentation points are shown on both graphs, marked with circles.

chosen here is valid not only as an analysis measure, but could also be used in a working implementation of a recognition system.

To segment the articulated gestures we apply the Two-Thirds Power Law in human motor control [Lacuaniti et al. 1983] which states that the pen velocity is related to the curvature of the stroke. The sharper a corner of the pen gesture is, the slower the pen speed is. This phenomenon is demonstrated in Figure 11 where the spatial extent of an articulated pattern is shown (top), and the corresponding velocity profile is plotted as a function of time (bottom) for a user writing a square on a tablet. The phenomenon as revealed by the Two-Thirds Power Law has previously been used to segment writing into substrokes [van Doorn and Keuss 1993]. Local minimums in the velocity profile are used as segmentation points. In our implementation we used a low pass FIR filter to filter out noise in the velocity profile, followed by a search for the local minimums.

Figure 12 shows the directional difference based on the 14 gestures in the experiment consisting of straight line segments only. There was no significant difference in this measure across the feedback conditions: $F_{3,24} = 2.28, p = 0.105$. This is another major aspect of pen-gesture production that is not influenced by feedback conditions.

3.4 Conclusions and Discussion of Experiment 1

The main conclusions and the implications we draw from this systematic study of feedback on pen gesture reproduction are as follows.

First, the tightly coupled auditory feedback on gesture reproduction has impact on some, but not all aspects of gesture production. Even though we relaxed the requirement on the aesthetic aspects of auditory feedback and focused only on making the auditory feedback discriminable and responsive (instantaneous), the difference caused by auditory feedback in the *shape* aspects of the gestures produced

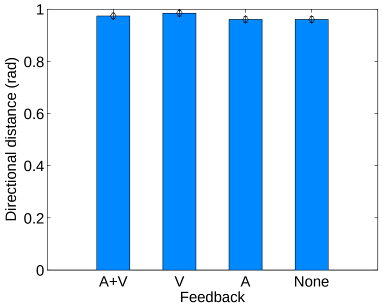


Fig. 12. Average directional distance for each condition.

was still statistically insignificant. For aperture we observed a statistical significant and positive influence of both auditory and visual feedback. Our results can explain Brewster and colleagues' positive finding [Brewster et al. 2003] of auditory feedback on gesture accuracy since they used a position based gesture recognition system. In the experiments by Ghez et al. [2000] a gesture was articulated repeatedly in synchrony with an external musical beat. Auditory feedback was given during the gesture production based on how well the timing of the movement matched the beat. Even though they observed a positive influence of auditory feedback, the difference to our experiment is evident in that we did not give feedback relative to the articulated gesture. Such feedback could not be provided in a real gesture input system since the resulting gesture is not known ahead of time.

Second, although visual feedback had a significant impact on more aspects of the gestures reproduced, the magnitude of these impacts was also small except in terms of size and the aperture of the closed gestures.

Third, the relatively small contribution of feedback, either visual or auditory, on shape aspects of gesture production supports open-loop theories of gesture control. Feedback appeared to be too slow for most aspects of gesture production. This is also consistent with the theory that handwriting relies on strong internal representations rather than feedback [Wright 1990], notwithstanding the differences between natural handwriting and the digital gesture production in the experiment, including the amount of experience and the style of practice (distributed in natural handwriting vs. massed in our experiment).

Fourth, an important design implication from this experiment is that it is possible, in light of findings here, to minimize reliance on visual attention when designing the gesture set and recognition method. As we can see, the overall shape, either in terms of length proportion (measure 4) or direction (measure 5) was not affected by the presence or absence of visual feedback. Only size and aperture were affected. The design of gesture interface should hence avoid using the latter two aspects as critical recognition information. In particular, local and referential features as exemplified by crossing the t's and dotting the i's in natural hand writing should be avoided in the design of gesture set and gesture recognizer.

Fifth, there were a few unquantified benefits or impacts of auditory feedback in gesture production. During the experiment, some of the participants were observed to follow the rhythm of the sound with their head or body. In the conditions where no audio feedback was given, three of the nine participants indicated that they tried to recall the sound feedback in their memory while articulating the gestures. Three participants indicated that they felt more certain or more confident in producing the gestures with the auditory feedback than without. These comments were made in response to a general open-ended request of comments at the end experiment, or made unsolicited during the experiment.

4. EXPERIMENT 2: THE JOY OF WRITING WITH MUSIC

In contrast to Experiment 1, which found some but relatively small impact of auditory feedback tightly coupled with gesture production, this experiment focuses on the aesthetic impact of auditory feedback.

Music is becoming an increasingly integrated part of our daily lives. Today we see the first generation of a new appliance for the home, the network media player, designed to stream music from the PC to the home stereo over the home network. Portable music is also becoming popular with products such as Apple's iPod. A relatively new industry that the record companies are starting to take an interest in is the distribution of copyrighted music as ring tones for mobile phones. We envision that the music will become an even more ubiquitous part of our everyday life, as computers become small and more powerful. Is it possible to integrate music in text entry applications, and modified the music according to the writing patterns? The goal in that case would not be to decrease learning time or improve performance, but simply to make the product more enjoyable, pleasant or stimulating.

Here we present an initial exploration of the idea of integrating and adapting music with a gesture based text input system. We develop two simple musical feedback methods and compare them to a silent situation, and the auditory feedback method used in Experiment 1. For this experiment we used a text input system that is simple, easy to learn, and resembles the widely used commercial text input system, Graffiti. The participants' performance in the various feedback conditions was evaluated, along with the aesthetic aspects of the system. This was done using a mixture of quantitative measures, a questionnaire designed for evaluation of user satisfaction [Chin et al. 1988], and open ended interviews.

4.1 Experimental task and gesture recognizer development

To test the aesthetic of auditory feedback, we first needed to develop a pen gesture input task more practical than the task in Experiment 1. We chose to design and implement a revised version of EdgeWrite [Wobbrock et al. 2003], using similar alphabet without the physical edge, as our experimental system. EdgeWrite is an English character input method using an alphabet resembling the movement of the natural English alphabet. Since the design of the recognizer is beside the theme of this paper, we will report it elsewhere [Andersen and Zhai 2005]. To investigate the issues concerned in this study, any character input system could have served as the experimental material.

4.2 Musical feedback

The experiment included four different auditory feedback conditions:

(1) Silence

(2) Continuous tone identical to the auditory feedback used in the previous experiment

(3) Rhythmic feedback: Musical feedback based on guitar sounds and a drum loop

(4) Song feedback: A song played at varying tempo determined by the users' average writing speed

The continuous tone is included here for comparison. Its perceptual features are similar to auditory displays commonly found in the literature [Kramer 1994].

The last two conditions can be called musical feedback. In general two directions can be taken in designing musical feedback. One is to compose the music algorithmically on-line based on component samples and synthesis models. This approach often requires large resources in terms of time and skill, to make the sound pleasant and stimulating over prolonged use. Even with such careful engineering, the music is likely to be restricted to the musical style that the developer chooses. The other approach is to use pre-recorded music in combination with meta-data describing the music, such as rhythm, key, and the structure of the musical piece. Using meta-data, it is possible to transform the music during interaction while maintaining the original identity of the music. The advantage of this approach in comparison to pure algorithmic composition is that the user can choose the type and style of music used as feedback. The user selects the songs that he/she wants, in the same way a musical piece is selected on the stereo or on an iPod. The music is transformed in real time based on the user's gesture input and will sound different each time it is played. However, it may be hard to get the required meta-data needed to transform the music and at the same time maintain a high sound quality. Some information can be extracted automatically, but in general it is hard to achieve the precision and success rate needed for transformation of music. As digital music distribution is becoming more popular and widespread, we may see that the music industry distribute meta-data with the sound files to compensate for the lack of physical media when the music is sold. Certain independent record companies has already started to distribute meta-data such as tempo information with the digital sound files, and meta-data is increasingly becoming available through Internet based services such as MusicBrainz[1].

In this experiment, we designed Conditions 3 and 4 based on these two approaches, resulting in "rhythmic feedback" and "song playback" respectively. The rhythmic feedback was based on custom samples that were synchronized to a specific beat. The beat was defined by a drum loop and was played when the user was writing. If no writing occurred for several seconds, the beat stopped and the system became silent. Whenever the user moved the pen in one of eight directions, a guitar sample was played. Each direction corresponded to a cord. The cords played in synchronization with the drum loop. This made the guitar sound blend nicely with the beat.

In the song playback condition the same song, "Baya Baya" by the group Safri Duo, was played during writing but the speed of playback was controlled by the average pen speed during the previous two seconds. In the beginning it sounded very

[1] See `http://www.musicbrainz.org/`

slow, but speeded up as the user wrote faster. If the user made a backspace stroke, the song immediately played backwards until a new character was written. The scaling of the sound in time was done using linear interpolation of the waveform. One artifact of this method was that it also altered pitch with speed. However, the method was simple to use and the change of pitch made it easier to perceive difference when writing at different speeds. To further keep the system simple, no meta-data such as tempo information was used.

4.3 Experiment, results and discussion

Sixteen participants, eight men and eight women, who had no training in music were selected, all between 20 and 50 years of age. Their prior experience with digital pen-based text input (mostly Graffiti) ranged from daily to none. The experiment task was to write the sentence "The quick brown fox jumps over the lazy dog" two times in each condition with the revised version of EdgeWrite. During writing the participant was asked to look at the screen where the only feedback was the recognized letters. A help screen showing the symbols for each letter in the alphabet could be displayed when the user pressed the space bar. The help screen was removed when the user started using the pen again. Before the experiment, a training session was given in which the participant wrote the alphabet from A to Z two times, and the testing sentence two times. There was no auditory feedback in the training session. Each participant participated in all feedback conditions. The order of conditions was balanced across participants using a latin square.

The data analysis was done using ANOVA repeated measure analysis, and a LSD post hoc test at 0.05 significance level. Performance measures used included the number of times the help screen was used, task completion time, and pen gestures per character (PGPC). As one measure of efficiency, PGPC was defined as the ratio between the number of gestures made and the number of characters in the resulting string. It was calculated exactly the same way as KSPC (keystrokes per character) used by Wobbrock et al., who adapted the term from stylus keyboarding studies [MacKenzie 2002] although in gesture interface there were no keystrokes per se. Note that to be comparable and consistent with the prior literature, back-space was not counted in the calculation [Wobbrock et al. 2003].

The experimental results showed no significant difference in PGPC between any of the conditions: $F_{3,45} = 0.897, p = 0.45$. The number of times the help screen was used was not significantly changed across conditions: $F_{3,45} = 0.938, p = 0.43$. Neither was the task completion time: $F_{3,45} = 0.283, p = 0.838$.

The average number of times that the user had to call the alphabet symbols set over the course of the experiment decreased to 1.24 (out of a minimum 86 gestures required to write in each condition), showing that the alphabet symbols was memorized easily (Figure 13, left). Completion time decreased to 139.3 seconds per "The quick brown ..." sentence, corresponding to 7.4 effective words per minute (WPM). PGPC decreased slightly in the course of the experiment to 1.24 in the last condition tested (Figure 13, right).

The joy and aesthetic aspects of the feedback conditions were evaluated using a modified version of the Questionnaire for User Interface Satisfaction [Chin et al. 1988]. The dimensions that were particularly relevant in the current context were: Terrible - Wonderful, Frustrating - Satisfying, Dull - Stimulating. Each scale was

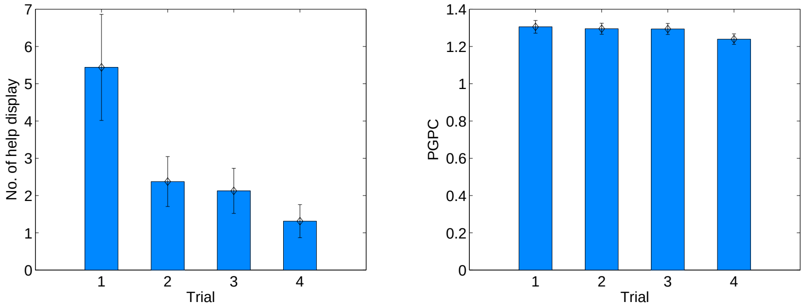


Fig. 13. Average number of times help was used for each condition (left). Average PGCP for each time the task was repeated (right).

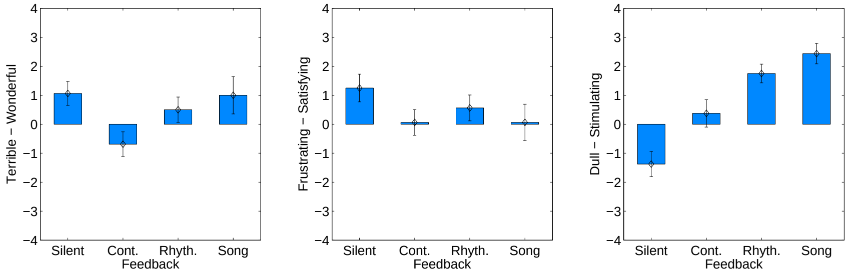


Fig. 14. Subjective ratings of the four conditions on three scales: Terrible-Wonderful (left), Frustrating-Satisfying (middle) and Dull-Stimulating (right)

used by placing a mark on a scale with 9 levels. After the participants had completed the questionnaire, an open-ended interview was conducted with each participant.

On the dimension of “Terrible-Wonderful” (Figure 14, left), participants’ ratings of the four sound conditions varied significantly ($F_{3,45} = 2.853, p = 0.048$). Pairwise post hoc analysis showed that “silent” ($p = 0.009$) was significantly better rated than “continuous” condition. “song” was marginally better rated than “continuous” ($p = 0.078$), and “rhythmic” was not significantly different from “continuous” ($p = 0.133$).

On the dimension of “Frustrating-Satisfying” (Figure 14, middle), participants’ ratings of the four sound conditions were not statistically significant ($F_{3,45} = 1.285, p = 0.291$).

On the dimension of “Dull-Stimulating” (Figure 14, right), participants’ ratings of the four sound conditions changed significantly ($F_{3,45} = 22.571, p < 0.001$). Post hoc analysis showed that all pairwise comparisons were statistically significant ($p \leq 0.016$), except between “rhythmic” and “song” ($p = 0.094$). The “silent” condition was rated the most dull, “continuous” was slightly more stimulating, “rhythmic” was even more stimulating, and finally the “song” condition was rated the most stimulating.

Several participants commented on the “song” feedback. Some found it frus-

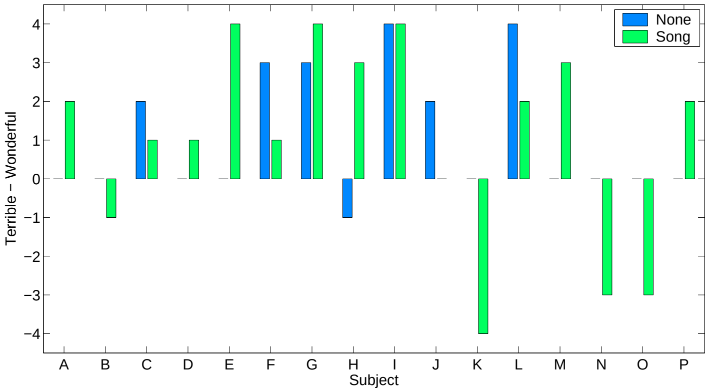


Fig. 15. Individual ratings of the None and Song conditions on the terrible-wonderful scale.

trating that the song speed increased as writing speed increased. Others liked it and thought it helped to stay in the "flow" of writing, and one participant wanted to keep writing to hear the whole song. Some commented on the song mapping, stating that they were not aware of what exactly was changed in the song during writing. Many participants further commented that they liked the continuous sound the least, but two participants thought that it would help them over longer use, even though they didn't like the sound. Some liked the rhythmic feedback better than the song feedback.

In summary, if and how well users like sound feedback for gesture production depends on the type of sound provided to them. In this experiment the participants did not like the continuous synthetic sound feedback, which was rated negatively on the dimension of "terrible - wonderful." The richer and more natural "rhythmic" and "song" conditions were much better liked in comparison.

Note that the best rated feedback, "song," was not judged better (or worse) than no sound at all ("silent") on the dimension of goodness (terrible - wonderful). This could be a result of the limited number of designs we have explored to date. There could potentially be still better designs to be developed. On the other hand, it is expected that there are situations and individuals that demand silence. Figure 15 shows the individual ratings of the silent vs. song condition, illustrating the personal preferences on the enjoyability of sound vs. no sound.

The experiment also unequivocally demonstrated that auditory feedback could make pen gesture production more exciting. All of the sound conditions, including the "continuous" condition, which was poorly rated on the dimension of "terrible - wonderful," were judged significantly more stimulating than the silent mode.

5. CONCLUSIONS

Auditory information is a compelling resource to exploit in human computer interaction, particularly in view of the growing use of small mobile devices whose visual display space is limited. The fact that rhythm and patterns are easily perceived and memorized in sound makes auditory feedback an even more attractive modality for dynamic pen gestures which in essence are temporal-spatial patterns. Against this background, we conducted a series of studies on auditory feedback in pen-gesture input systems. Our initial iterative explorations showed that while it was easy to gain benefit from simple auditory feedback methods such as pronunciation of the gestured word, it was difficult to gain further advantage in gesture performance or learning time by employing more complex auditory feedback methods. To establish a baseline estimation, Experiment 1 formally evaluated gesture production accuracy as a function of auditory feedback as well as visual feedback. We found that while features such as the size of a gesture and the aperture of the closed gestures were influenced by visual or auditory feedback, other major features such as proportional shape distance were not affected by either. Although the experiment does not eliminate the possibility that more dramatic benefit could be found through broader explorations of the vast design space (and this study may serve as an important reference for such explorations), the findings here support the theoretical hypothesis that any form of feedback is too slow for on-line gesture production which might be driven by a strong internal central representation [Wright 1990]. There are important design implications of these findings. For example, instead of relying on visual or auditory feedback to close the aperture of closed gestures or more generally any features that are referential to previous segments of the gesture or other absolute positions, the design of gesture set and its recognizer should minimize this type of features to support heads-up use. Given our results, future researchers of auditory pen-gesture interface should not need repeat experimentations like ours. Rather, they could use the current study as their design and exploration guidance, or take dramatically different directions from those we have taken.

In addition to performance, emotional appeal and fun have recently being brought to the forefront of HCI research [Norman 2004; Blythe et al. 2003]. It has been argued that a more aesthetic interface is also more usable [Tractinsky et al. 2000]. Experiment 2 was focused on these aspects of auditory feedback in dynamic pen-gesture interfaces. It measured participants' view of various subjective dimensions in four auditory feedback conditions when writing letters with a pen-gesture interface: silent, continuous feedback, and two types of musical feedback. While these auditory conditions did not alter writing performance in terms of speed, learning or error rate, they were perceived very differently on the emotional dimensions. Participants rated the silent and song conditions more "wonderful," the silent condition more "dull," and the song condition most stimulating.

Overall, our exploration has laid a foundation for future design and research on auditory feedback in pen-gesture interfaces. It points out several general rules of auditory feedback: a few simple functions are easy to achieve (such as warning and prononcing the results of recognition), gaining further performance and learning advantage is difficult, gesture set and its recognizer can be designed to minimize visual dependence, and positive emotional or aesthetic response can be achieved using musical auditory feedback.

6. ACKNOWLEDGMENTS

We thank Malcolm Slaney, Chris Campbell, Barton Smith and other colleagues at the IBM Almaden Research Center for their input and insight.

REFERENCES

ALGAZI, V. R., DUDA, R. O., THOMPSON, D. M., AND AVENDANO, C. 2001. The cipic hrtf database. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Electroacoustics*. Mohonk Mountain House, New Paltz, NY, 99–102.

ANDERSEN, T. H. AND ZHAI, S. 2005. Robust pen-gesture recognition with low visual attention demand - the design of the ELEW character input system. In preparation.

BLATTNER, M., SUMIKAWA, D., AND GREENBERG, R. 1990. Earcons and icons: Their structure and common design principles. In *Human Computer Interaction.* Vol. 16. 523–531.

BLYTHE, M., MONK, A., OVERBEEKE, C., AND WRIGHT, P., Eds. 2003. *Funology: From Usability to User Enjoyment.* Dordrecht: Kluwer.

BREWSTER, S., LUMSDEN, J., AND ET.AL., M. B. 2003. Multimodal 'eyes-free' interaction techniques for wearable devices. In *Proceedings of CHI 2003*. 473–480.

BROADBENT, D. 1977. The hidden preattentive processes. *American Psychologist 32,* 2, 109–118.

BUXTON, W. 1995. *Readings in Human Computer Interaction: Towards the Year 2000.* Morgan Kaufmann Publishers. Speech, Language & Audition, Chapter 8.

CHIN, J., DIEHL, V., AND NORMAN, K. 1988. Development of an instrument measuring user satisfaction of the human-computer interface. In *Proceedings of CHI*. 213–218.

CLARKE, E. 1999. *The Psychology of Music*, Sec. ed. ed. Academic Press, Chapter 13: Rhythm and Timing in Music, 473–500.

DEATHERAGE, B. H. 1972. *Human Engineering Guide to Equipment Design*, Revised Edition ed. U.S. Government Printing Office. Auditory and Other Sensory Forms of Information Presentation, 123–160.

FLETCHER, H., BLACKHAM, E., AND STRATTON, R. 1962. Quality of piano tones. *Journal of the Acoustical Society of America 34,* 6, 749–761.

GAVER, W. 1989. The SonicFinder: An interface that uses auditory icons. *Human Computer Interaction 4,* 1, 67–94.

GHEZ, C., RIKAKIS, T., DUBOIS, R. L., AND COOK, P. 2000. An auditory display system for aiding interjoint coordination. In *Proceedings of the International Conference on Auditory Display.*

GOLDBERG, D. AND RICHARDSON, C. 1993. Touch-typing with a stylus. In *Proceedings of InterCHI*. 80–87.

KRAMER, G. 1994. *Auditory Display: Sonification, audification, and auditory interface.* Santa Fe Institute studies in the sciences of complexity, Proceedings XVIII. Addison-Wesley. An Introduction to auditory display, 1–77.

KRISTENSSON, P.-O. AND ZHAI, S. 2004. Shark2: A large vocabulary shorthand writing system for pen-based computers. In *Proceedings of UIST*. 43 – 52.

KURTENBACH, G. AND BUXTON, W. 1994. User learning and performance with marking menus. In *Proceedings of CHI*. 258–264.

LACUANITI, F., TERZUOLO, C., AND VIVIANI, P. 1983. The law relating the kinematic and figural aspects of drawing movements. *Acta Psychologica 54*, 115–130.

LEGGE, D., STEINBERG, H., AND SUMMERFIELD, A. 1964. Simple measures of handwriting as indices of drug effects. *Perceptual and Motor Skills 18*, 549–558.

LOEB, R. AND FITCH, W. 2002. A laboratory evaluation of an auditory display designed to enhance intraoperative monitoring. *Anesthesia Analgesia 94*, 362–368.

LUSCHEI, E., SASLOW, C., AND GLICKSTEIN, M. 1967. Muscle potentials in reaction time. *Experimental Neurology 18*, 429–442.

MACKENZIE, I. 2002. Kspc (keystrokes per character) as a characteristic of text entry techniques. In *Proceedings of Mobile HCI*. Springer-Verlag, 195–210.

MacMillan, K., Droettboom, M., and Fujinaga, I. 2001. Audio latency measurements of desktop operating systems. In *Proceedings of the International Computer Music Conference*. 259–262.

Norman, D. 2004. *Emotional Design: why we love (or hate) everyday things*. Basic Books, New York.

Olive, T. and Piolat, A. 2002. Suppressing visual feedback in written composition: Effects on processing demands and coordination of the writing processes. *International Journal of Psychology 37,* 4, 209–218.

Patterson, R. 1989. Guidelines for the design of auditory warning sounds. In *Proceedings of the Institute of Acoustics, Spring Conference*. Vol. 2. Edinburgh: Institute of Acoustics, 17–24.

Pirhonen, A., Brewster, S., and Holguin, C. 2002. Gestural and audio metaphores as a means of control for mobile devices. In *Proceedings of CHI 2002*. 291–298.

Plomp, R. 1964. The ear as a frequency analyzer. *Journal of the Acoustical Society of America 36*, 1628–1636.

Singh, P. 1987. Perceptual organization of complex-tone sequences: A tradeoff between pitch and timbre? *Journal of the Acoustical Society of America 82*, 886–899.

Smyth, M. and Silvers, G. 1987. Functions of vision in the control of handwriting. *Acta Psychologica 66*, 47–64.

Teulings, H. L. and Schomaker, L. 1993. Invariant properties between stroke features in handwriting. *Acta Psychologica 82*, 69–88.

Tractinsky, N., Katz, A., and Ikar, D. 2000. What is beautiful is usable. *Interacting with Computers 13*, 127–145.

van Doorn, R. and Keuss, P. 1992. The role of vision in the temporal and spatial control of handwriting. *Acta Psychologica 81*, 269–286.

van Doorn, R. and Keuss, P. 1993. Does production of letter strokes in handwriting benefit from vision? *Acta Psychologica 82*, 275–290.

Welch, R. 1999. *Cognitive Contributions to the Perception of Spatial and Temporal Events*. Elsevier. Meaning, attention, and the “unity assumption” in the intersensory bias of spatial and temporal perceptions.

Wobbrock, J. O., Myers, B. A., and Kembel, J. A. 2003. Edgewrite: A stylus-based text entry method designed for high accuracy and stability of motion. In *Proceedings of the ACM Symposium on User Interface Software and Technology*. 61–70.

Woods, D. 1995. The alarm problem and directed attention in dynamic fault management. *Ergonomics 38,* 11, 2371–2394.

Wright, C. E. 1990. *Attention and performance XIII*. Erlbaum. Generalized motor programs: Reexamining claims of effector independence in writing, Hillsdale, NJ, 294–320.

Zhai, S. and Kristensson, P.-O. 2003. Shorthand writing on stylus keyboard. In *Proceedings of CHI*. 97–104.

4.2 Robust pen-gesture recognition with low visual attention demand — The design of the ELEW character input system

Submitted for publication. T. H. Andersen and S. Zhai. Robust pen-gesture recognition with low visual attention demand — The design of the ELEW character input system. 4 pages. 2005.

Robust pen-gesture recognition with low visual attention demand — The design of the ELEW character input system

Tue Haste Andersen
Department of Computer Science
University of Copenhagen
DK-2100 Copenhagen Ø, Denmark
haste@diku.dk

Shumin Zhai
IBM Almaden Research Center
650 Harry Road, NWE-B2
San Jose, CA 92150, USA
zhai@us.ibm.com

ABSTRACT

High recognition accuracy and low visual attention are two critical demands of pen-gesture interfaces. Using relative and non-referential features in the design of pen gesture set and its recognizer could be an effective approach to meet these demands. Using EdgeWrite as an example and inspiration, we designed a character input system, ELEW, that uses proportional and angular pattern matching in lieu of positional features in its recognition engine. Our study shows that ELEW provides a similar level of input speed and robustness to users without the assistance of the physical edges in EdgeWrite. The approach embodied in ELEW can complement other measures such as the use of physical edges in gesture system design.

INTRODUCTION

Interaction techniques based on pen-gesture interfaces are increasingly important as computing moves to new and portable forms. There are many desirable goals in developing a pen-gesture interface for text or command input. High robustness and low attention demand are two of them (others include speed, ease of learning, etc). Various attempts have been made in the HCI field to reach the goals of high robustness and lower attention demand. Among those, two notable innovations are the design of the novel gesture set (i.e. alphabet) [3] and the use of haptic feedback [11].

The most influential work towards these goals is probably Unistroke by Goldberg and Richardson [3]. Unlike previous work on handwriting recognition based on natural alphabets, Unistroke employs a novel alphabet (gesture set) to remove multiple strokes in the Roman alphabet and increase the distance between gestures in the “sloppiness space.” In ordinary handwriting, aesthetics of the writing are also important. The writing has to be presentable and easy to read hence requiring a great deal of visual attention. In contrast, with unistroke the user need not care about the aesthetics of the written gestures but only whether the gesture can be correctly recognized by the system. A unistroke letter is insensitive to its location relative to the previously written letters. Letters can therefore be written on top of each other [3], hence supporting heads-up input.

More recently, researchers have explored a second approach to improve pen-gesture input systems: The use of feedback such as passive haptics [11] and audio [1]. EdgeWrite, developed by Wobbrock and Myers [11], is a text entry method for hand-held devices designed to provide stability of motion for people with motor impairments. Characters are articulated by moving the stylus on the surface of a touch screen overlaid by a plastic fixture with a square cutout. A user may feel the physical edges in articulating letters in text entry. All characters can be entered by traversing the edges and diagonals of the square cutout. A gesture is completed by lifting the stylus. Since all characters in EdgeWrite can be formed by traversing the edges or diagonals of the writing area, the visual layout of patterns is constrained. Wobbrock and Myers ingeniously designed a gesture set that mimics the kinematic movements of the corresponding Roman letters with the edge constraints. The feel of these gestures is reported to be close to the feel of writing real Roman letters. Each letter in EdgeWrite is defined by the sequence in which the corners of the square hole are hit. This makes the recognition system very simple. The order in which the corners are hit is matched against the order of the corners in the letter templates.

The current work focuses on a third direction for improving the robustness and decreasing visual attention in character/pen gesture input. This work is derived from the exploration of auditory and visual feedback in pen-gesture interfaces [1]. Our exploration found that while visual feedback is important for the articulation of the aspects of pen-gestures that are absolute in location or referential to a previous point of the gesture (e.g. returning to the exact point where the pen stroke started), it has little impact on the angular (directional) components of each sub-stroke or the overall proportional pattern of the gesture. These findings are consistent with research on handwriting where visual feedback is found to be critical to “dotting the i’s and crossing t’s”, but only to a limited extent affects other features of the articulated handwriting [8, 10]. We therefore hypothesize that a greater degree of robustness and lower degree of visual attention demand (supporting heads-up) can be achieved if absolute and referential features are avoided in the design of pen-gesture recognition systems.

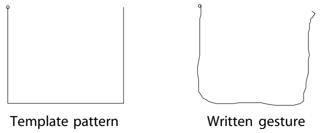


Figure 1: Template pattern (left), and example of written gesture (right). The circle signifies starting point of the pattern.

This idea, although implicitly used in the field (e.g. the SHARK Shorthand system [4]), has not been explicitly articulated or demonstrated either in the HCI or the handwriting recognition literature. In fact, absolute location and referential features are often used in gesture interface design (e.g. [2, 11]). We explore this hypothesis by developing and testing an example character input system, ELEW, based on a revised form of EdgeWrite. We demonstrate that by using relative and non-referential features instead of absolute locations (such as corners) in recognition, many of the benefits of the EdgeWrite system could be achieved without the assistance of the physical edges (hence ELEW: EdgeLess EdgeWrite). Note that the method we propose here complements rather than contradicts previous innovations. For example if both non-referential recognition and physical edges are used, character input performance would only be more robust.

DESIGNING ELEW RECOGNITION METHODS

We explore two recognition methods, both without absolute or referential features. One is based on direction matching and one on proportional shape matching. Both methods match the written gestures against a set of template patterns (also called prototypes). The template pattern that best matches the written gesture is selected as the recognized pattern (Figure 1).

Direction matching

Since the gesture set in EdgeWrite (and ELEW) consists only of straight lines, a gesture can be divided into straight line substrokes. In direction matching a gesture is first segmented into substrokes, each having a direction and a length. By only using directional (angular) information of the substrokes two aspects of the written gestures are discarded: Absolute position and length. This method preserves the order of the directional movements when comparing the directions of the two patterns. By discarding the length of the substrokes this method should increase robustness of recognition. For example the letter k in the EdgeWrite alphabet (see Figure 4) requires the written gesture to reach the upper left corner of the writing box, which is not required in natural handwriting of the letter k. If the recognizer only examines the directions of the four sub-strokes, a correct result can be attained even if the user moved only half way toward the upper right corner in articulating k.

To segment the written gestures into substrokes, we apply the Two-thirds power law in human motor control [5] which relates the pen velocity to the curvature of the stroke. The sharper a corner of the pen gesture is, the slower the pen speed is. This phenomenon is demonstrated in Figure 2 where speed and absolute direction are plotted as a function of time for a user writing the letter ’u’. The phenomenon as revealed by the Two-thirds power law has previously been used to segment writing into substrokes [9]. Local minimums in the velocity profile are used as segmentation points. In our implementation we used a low-pass FIR filter to reduce noise in the velocity profile, followed by a search for local minimums.

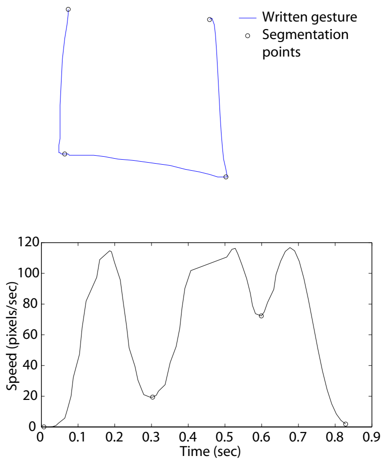


Figure 2: The written gesture is shown (top), with the corresponding velocity profile (bottom). Segmentation points are shown on both graphs, marked with circles.

After segmentation, the direction of each line segment is calculated from the end points of the line segment. This is followed by comparing the directions of these segments with the directions of substrokes in the template patterns. The comparison is implemented using exhaustive search for the best match.

Proportional shape matching

An alternative method is a simple algorithm based on calculating the mean Euclidean distance between two gestures. To avoid absolute features, such as position and scale, we normalize the written gesture to the size of the template pattern by scaling the gesture according to its bounding box. Also the written gesture is translated (shifted) to have the same start position as the template pattern. A template pattern is represented by ordered sample points $q(n), n = 1 \dots K$ equally spaced along the length of the pattern. The written gesture is sampled to the same number of points, $g(n), n = 1 \dots K$, using a fixed interval between the points and maintaining the order in which the gesture was articulated. The proportional shape distance d is defined as:

$$d = \frac{1}{K} \sum_{n=1}^{K} \sqrt{(q(n)_x - g(n)_x)^2 + (q(n)_y - g(n)_y)^2}, \quad (1)$$

where $q(n)_x$ and $q(n)_y$ denote the x and y coordinates of the point $q(n)$ respectively, and $g(n)_x$ and $g(n)_y$ denote the x and y coordinates of the point $g(n)$ respectively. The method is illustrated in Figure 3. Sensitive to the overall shape proportion but not to referential features, this measure has previ-

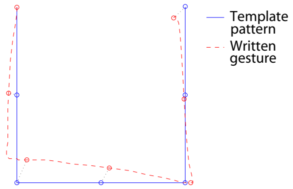


Figure 3: Proportional shape matching. The corresponding sampling points on the two patterns are shown with connecting dotted lines.

ously been used as the most basic gesture recognition metric in complex multi-channel recognition systems such as SHARK Shorthand [4], although its implication of low attention demand has not been previously studied.

Choosing simplicity

The two methods were tested informally by letting users write 20 gestures in two conditions with and without visual feedback. The recorded gestures were then matched against the corresponding template pattern using the two methods respectively. For both measures we found a good match between written gestures and the corresponding template patterns. Furthermore we did not observe any significant difference in the match score between visual and non-visual conditions, suggesting that the features chosen are indeed robust to the lack of visual feedback. The computation of the direction matching algorithm is more complex and will inevitably lead to longer processing times than proportional shape matching. Hence we chose the proportional shape matching as our preferred method.

THE ELEW GESTURE SET

We designed the ELEW gesture set based on the EdgeWrite gestures (Figure 4). The EdgeWrite gesture set was chosen as our basis because it has been demonstrated to have a quick learning time, partially because of its similarity to the kinesthetic patterns of Roman letters. However, not all of the EdgeWrite template patterns resemble Roman letters, and thus we chose to re-design a few templates, as shown in Figure 5. Some of the other letters, such as k, can also be improved for ease of learning, but we choose not to change too many for this study. The modified template patterns of ELEW were possible because of the proportional shape recognition method employed that did not rely on corner detection.

The gesture sets of both EdgeWrite and ELEW are easy to learn and use, due to their limited size and similarity to Roman characters. However, with larger gesture sets, as are developed in pen-gesture systems for improving throughput (e.g. SHARK Shorthand [4]), the recognition method used has an effect on template pattern complexity, and thus the ease of remembering them. With large gesture sets, corner-sequence recognition requires template patterns to have more substrokes and differ more in appearance from their Roman character equivalents than proportional-based recognition. In addition, error tolerance with EdgeWrite's corner-sequence recognition is less forgiving than with proportional-based recognition. One missed corner will cause a recognition error in EdgeWrite, whereas ELEW can correctly recognize a gesture that varies from the template (within a tolerance).

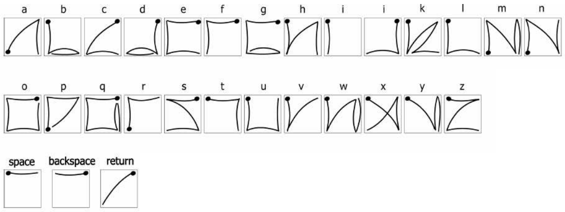


Figure 4: A subset of the original EdgeWrite character set [11].

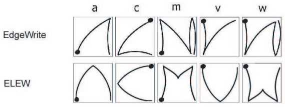


Figure 5: Template pattern difference between EdgeWrite and ELEW.

A SMALL USER STUDY

An ELEW prototype implemented with the proportional matching algorithm was tested in a small study. The goal of the study was to obtain an indication of the level of performance compared to EdgeWrite as reported in [11] and other pen-gesture character input systems. An exact and more rigorous user study would require longitudinal testing with a large or open set of testing phrases. However, we made sure that no bias was introduced in our limited test.

Sixteen people participated in the study; eight men and eight women; all between 20 and 50 years of age. The participants' prior experience with digital pen-based text input (Graffiti) ranged from daily to none. The experimental task was to write the pangram (covering all letters in the Roman alphabet) "The quick brown fox jumps over the lazy dog" two times. The task was repeated four times with breaks in between. The input device used was a Wacom Tablet digitizer. During writing the participant was asked to look at the screen where the only feedback was the recognized letters. A help screen showing the gesture templates for all letters in the alphabet could be displayed when the user pressed the space bar. The help screen was removed once the user started using the pen. Before the experiment, a training session was given in which the participant wrote the alphabet from A to Z two times, and the testing sentence two times.

Performance measures used included words per minute (W-PM), the number of times the help screen was viewed, task completion time, and pen gestures per character (PGPC). As one measure of efficiency reflecting the average number of attempts made to write each character successfully, PGPC was defined as the ratio between the number of gestures made and the number of characters in the resulting string. It was calculated exactly the same way as KSPC (keystrokes per character) used by Wobbrock et al., who adapted the term from stylus keyboarding studies [6], although in gesture interfaces there are no keystrokes per se. Note that to be com-

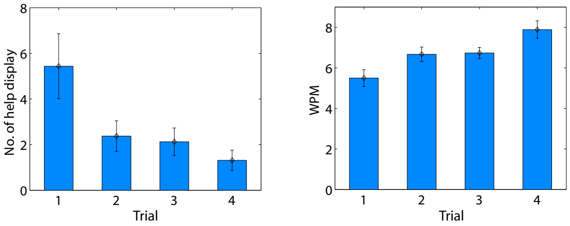


Figure 6: Average number of times help was used for each trial (left), and average words per minute for each trial (right)

parable and consistent with the prior literature, back-space was not counted in the calculation [11]. Wobbrock et al. [11] reported two studies of EdgeWrite; a qualitative study with four disabled people and a quantitative study with 10 able-bodied people. The results reported here are compared against the quantitative study of EdgeWrite with able-bodied people. Their study differed in that 20 different trials were used as opposed to our 8 identical trials (but note both ELEW and EdgeWrite are character level input and our test does cover all characters), and that their participants had no prior experience with handheld text entry (also note that our participants' experience was not with ELEW). Wobbrock et al. observed a rapid improvement during the first 12 trials of their experiment. To the disadvantage of the ELEW result, we compare all of the 8 trials in our study against the last and improved 8 trials of the EdgeWrite study.

The frequency with which the user had to display the alphabet symbols set over the course of the experiment decreased from 6.3% to 1.5% (1.32 out of 86 gestures required to write in each trial). Comparing this to the EdgeWrite study where the use of a paper help sheet by the end of 20 trials decreased to 0.46%, the result is in favor of EdgeWrite. However, the total time used in the EdgeWrite experiment was also longer and thus the subjects had a longer time to learn each of the gestures. The number of times the help screen was used in our study is shown on Figure 6 (left). Completion time decreased to 139.3 seconds per trial, corresponding to 7.4 effective words per minute (WPM), see Figure 6 (right). This is higher than the reported 6.6 WPM for EdgeWrite, where the tasks were more diverse with different sentences in each task. PGPC decreased slightly in the course of the experiment to 1.24 in the last task repetition compared to 1.26 for EdgeWrite.

Without the aid of the physical edges as in EdgeWrite and in less total time taken than in Wobbrock et al.'s study [11], ELEW reached a similar level of performance as EdgeWrite in terms of speed (WPM score) and accuracy (PGPC ratio), as reported in [11]. This level of performance, achieved in a heads-up condition in our experiment, was also comparable to Sears and Arora's study of Graffiti and Jot [7] in which novice users wrote words while looking at the writing area (heads-down). The results indicate that the approach of designing the recognizer to use non-absolute and non-referential features was indeed beneficial for developing character input systems with high robustness and low visual attention demand.

CONCLUSIONS

In summary, we have explored the design of a character input system, ELEW, whose recognition algorithm avoids absolute and referential features. As a result users could input characters in a heads-up fashion with a level of performance comparable to writing in Graffiti or Jot with visual feedback or EdgeWrite with the assistance of the physical constraints of edges. We should emphasize that the ideas presented here are not in competition with previous efforts such as the use of physical constraint. Instead, using non-absolute and non-referential features in recognition can complement these previous efforts. A future character input system can use haptic feedback, for instance through physical edges, or other feedback modalities in concert with proportional matching to further improve the overall performance and user experience of character input. Using patterns and relative features to improve robustness and relax attention demand is not necessarily limited to character input, but may be applicable to pen-gesture interfaces in general. Never explicitly articulated in the literature, we recommend this approach to gesture interface designers and developers, since it is relatively easy to implement with little cost to other design dimensions.

References

1. T. H. Andersen and S. Zhai. Explorations and experiments on the use of auditory and visual feedback in pen-gesture interfaces. Technical Report 04/16, DIKU, Copenhagen, December 2004.
2. S. Brewster, J. Lumsden, and M. Bell et.al. Multimodal 'eyes-free' interaction techniques for wearable devices. In *Proceedings of CHI 2003*, pages 473–480, 2003.
3. D. Goldberg and C. Richardson. Touch-typing with a stylus. In *Proceedings of InterCHI*, pages 80–87, 1993.
4. P-O. Kristensson and S. Zhai. SHARK^2: A large vocabulary shorthand writing system for pen-based computers. In *Proceedings of UIST*, pages 43 – 52, 2004.
5. F. Lacuaniti, C. Terzuolo, and P. Viviani. The law relating the kinematic and figural aspects of drawing movements. *Acta Psychologica*, 54:115–130, 1983.
6. I.S. MacKenzie. KSPC (keystrokes per character) as a characteristic of text entry techniques. In *Proceedings of Mobile HCI*, pages 195–210. Springer-Verlag, 2002.
7. A. Sears and R. Arora. Data entry for mobile devices: An empirical comparison of novice performance with jot and graffiti. *Interacting with Comp.*, 14:413–433, 2002.
8. M. Smyth and G. Silvers. Functions of vision in the control of handwriting. *Acta Psych.*, 66:47–64, 1987.
9. H. L. Teulings and L. Schomaker. Invariant properties between stroke features in handwriting. *Acta Psych.*, 82:69–88, 1993.
10. R. van Doorn and P. Keuss. The role of vision in the temporal and spatial control of handwriting. *Acta Psychologica*, 81:269–286, 1992.
11. J. O. Wobbrock, B. A. Myers, and J. A. Kembel. Edgewrite: A stylus-based text entry method designed for high accuracy and stability of motion. In *Proceedings of UIST*, pages 61–70, 2003.

Chapter 5

Conclusions and future work

In summary, three different areas were investigated in this thesis.

First, in a music search task similar to searching for a specific song on a CD, performance was not affected by different types of auditory feedback, input control or mappings. A preliminary experiment indicated that improving the interface to provide better input control and visual feedback did not improve search performance. However, a significant difference in how songs were searched and how the interfaces were perceived was observed. A rotary controller with variable search speed was preferred over buttons with fixed search speed. In general we found that the controller and mapping was more important than auditory feedback to how the interface was perceived. Auditory feedback should be designed to reflect change of position, allow for perception of local features such as rhythm and instrumentation, and finally should be pleasant to listen to. Indications that search time was linearly dependent on distance to target were observed. This was further investigated through the study of another task. In a common scrolling task, when searching for a specific occurrence in a text document, it was found that the time used was linearly dependent on distance to target. A model was developed and verified based on the hypothesis that the maximum scanning speed, at which the target can be perceived, is the largest limiting factor that determines the time spent in scrolling. Although the model only describes one usage situation of scrolling it clearly demonstrates that factors other than human motor control need to be considered when building human performance models in HCI.

Second, a musical performance situation was studied. Through an analysis of disc jockey (DJ) work practices it was found that a significant amount of time was spent in the manual task of beat tracking. Beat tracking was found to require a high level of cognitive workload. While CD players allow for reduction of workload compared to analog turntables they also limit the performance to what was previously arranged and prepared by the DJ. Using DJ software such as Mixxx that integrates a number of features centering on the availability of beat information, the workload could be reduced while maintaining flexibility. To extract beat information of recorded music a new algorithm was developed and a comparative study of different perceptual important parameters was presented.

Third, the use of auditory feedback in pen-gesture interfaces was investigated. Initial iterative exploration showed that gaining performance or learning advantage with auditory feedback was possible by coupling absolute cues and state feedback with auditory cues. However, gaining learning or performance advantage from auditory feedback tightly coupled with the pen gesture articulation and recognition process was more difficult. An experiment showed that auditory and visual feedback improved absolute positioning and referential features, but had little or no influence on shape and direction. A second experiment focused on the emotional and aesthetic aspects of auditory feedback in pen-gesture interfaces. Participants' rating on the dimensions of being wonderful and stimulating was significantly higher with musical auditory feedback. Learning from these experiments, a new character level text input system, ELEW, was designed. In ELEW, shape and directional features were used to recognize pen-gestures. An experiment where the system was used in a heads-up situation demonstrated that a similar performance level could be achieved with ELEW compared to EdgeWrite (Wobbrock, Myers, and Kembel 2003) and Graffiti in a heads-down situation.

Concluding from the perceptive of feedback mapping, changing the auditory feedback in a music search task did not yield any performance benefit. However, in the situation where fast forward was limited to four times normal playback speed a significant change in search strategy was observed. Other types of presentation could be more effective, especially when combined with meta-data. Input control with variable search speed resulted in increased satisfaction. However, judging by performance measure it is possible that listening to short pre-synthesized audio clips without interaction is a more effective way to identify a musical track than allowing the user to interact with playback speed. In studying a digital DJ interface, indications that adding visual display of the audio helped to reduce cognitive load. However, what role visualizations play in relation to professional musicians remains unclear. Professional musicians often do not look at their instrument while playing; some even close their eyes. Four out of the seven professional DJs participating in the evaluation did not use the displays and preferred to rely on their ears. The approach of adding meta-data reduced the importance of the displays and was an effective way to reduce workload for both amateurs and professionals. Finally, using the auditory channel in an artificial mapped modality proved difficult in a spatial task of pen-gesture input. It was not possible to gain improvements in performance or increase learning ability with the proposed designs of direct coupling of pen gesture articulation with sound. It is possible that other designs would result in a larger benefit from added audio, especially when evaluated over a longer period of time. However, judging from the explorations and studies, it seems unlikely that the benefit would be large. Rather, the approach of adding musical sound feedback to improve aesthetics seems more promising.

Looking forward, I find two areas highlighted by this work are especially promising for future research. First, the development of simple human performance models that include information about both sensory-motor control and high level perception was demonstrated with the scrolling experiment. Previous models such as Fitts' Law (Fitts 1954) have proven successful in the design of input devices and interaction techniques for desktop computing (Douglas, Kirkpatrick, and MacKenzie 1999). Such models could play an important role in the future design of interaction techniques for ubiquitous and mobile computing where visual attention and cognitive workload may be a larger limiting factor than sensory-motor control. Second, the use of music to provide feedback and improve aesthetic aspects of interaction was investigated in the pen-gesture studies. It seems to be a promising way to use sound in interaction and a way to bridge entertainment technology with ubiquitous computing, especially in light of the rapidly growing market for music sold through the Internet.

Bibliography

Andersen, T. H. (2005a). DJing with dance music: The design and evaluation of a digital DJ system. Full paper. Submitted.

Andersen, T. H. (2005b, September). Searching for music: How feedback and input-control change the way we search. In *Proceedings of Interact 2005*. Springer Verlag. To appear.

Andersen, T. H. (2005c, April). A simple movement time model for scrolling. In *Proceedings of CHI 2005*, pp. 1180–1183. ACM Press. Late breaking result.

Andersen, T. H. and S. Zhai (2005a). Robust pen-gesture recognition with low visual attention demand - the design of the ELEW character input system. 4 pages. Under review.

Andersen, T. H. and S. Zhai (2005b). "Writing with Music:" Exploring the use of auditory feedback in pen gesture interfaces. 24 pages. Under review.

Blythe, M., A. Monk, C. Overbeeke, and P. Wright (Eds.) (2003). *Funology: From Usability to User Enjoyment*. Dordrecht: Kluwer.

Bolt, R. A. (1980). "put-that-there": Voice and gesture at the graphics interface. *Computer Graphics 14*(3), 262–270.

Buxton, W. (1995). *Readings in Human Computer Interaction: Towards the Year 2000*, Chapter 8. Morgan Kaufmann Publishers. Speech, Language & Audition.

Deatherage, B. H. (1972). *Human Engineering Guide to Equipment Design* (Revised Edition ed.)., pp. 123–160. U.S. Government Printing Office. Auditory and Other Sensory Forms of Information Presentation.

Douglas, S., A. Kirkpatrick, and I. MacKenzie (1999). Testing pointing device performance and user assessment with the ISO 9241, part 9 standard. In *Proceedings of CHI*, pp. 215–222.

Encyclopædia Britannica (2005). "sensory reception, human". Encyclopædia Britannica Online, June 1. 2005.

Fitts, P. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology 47*, 381–391.

Gaver, W. (1989). The SonicFinder: An interface that uses auditory icons. *Human Computer Interaction 4*(1), 67–94.

Gaver, W. (1993). What in the world do we hear? an ecological approach to auditory source perception. *Ecological Psychology 5*, 1–29.

Gizmodo (2005, March). Cebit: Siemens runster prototype. http://www.gizmodo.com/gadgets/gadgets/household/cebit-siemens-runster-prototype-035386.php.

Hinckley, K., E. Cutrell, S. Bathiche, and T. Muss (2002, April). Quantitative analysis of scrolling techniques. In *Proceeedings of CHI 2002*.

Jensen, K. and T. H. Andersen (2003). Real-time beat estimation using feature extraction. In *Computer Music Modeling and Retrieval: International Symposium, Montpellier*, Lecture Notes in Computer Science, pp. 13–22. Springer Verlag.

Norman, D. (2004). *Emotional Design: why we love (or hate) everyday things*. New York: Basic Books.

Oviatt, S. (2002). *Multimodal Interfaces*, pp. 286–304. Lawrence Erlbaum. Chapter in The Human-Computer Interaction Handbook.

Weiser, M. (1991, September). The computer for the twenty-first century. *Scientific American*, 94–100.

Wobbrock, J. O., B. A. Myers, and J. A. Kembel (2003). Edgewrite: A stylus-based text entry method designed for high accuracy and stability of motion. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp. 61–70.