The Calibration and Optimisation of Speed-Dependent Automatic Zooming

November 2003

Andrew Wallace
ajw125@student.canterbury.ac.nz
Department of Computer Science and Software Engineering
University of Canterbury, Christchurch, New Zealand

Supervisor: Andy Cockburn
andy@cosc.canterbury.ac.nz

Abstract

Speed Dependent Automatic Zooming (SDAZ) has already been shown to be an effective navigation technique for document browsing. We investigate here three further aspects of SDAZ. We begin with an analysis of the existing methods of document navigation. We present a taxonomy of document navigation goals and examine how these goals are supported by the current navigation techniques. Our focus here is on how SDAZ fits in with existing techniques and how it can assist in the support of these different goals. Our second area of investigation, which is the main focus of this work, is on the issues surrounding the calibration of SDAZ systems. The extent of the difficulty of obtaining an optimal system has not been made clear in the previous literature. We discuss in detail the implementation and fine-tuning issues surrounding SDAZ systems, with a view to obtaining an optimal implementation. Finally, using our SDAZ system we perform a small formal evaluation comparing SDAZ to rate-based scrolling. SDAZ is a modified rate-based system, and our evaluation shows it is superior to a standard rate-based system, and therefore proves that the modifications are worthwhile.

Acknowledgements

The suggestions and help of Andy Cockburn, and the OpenGL document viewer written by Josh Savage were of invaluable assistance in this work. Thanks also to Neville Churcher and Oliver Hunt for proofreading.

Contents

Chapter 1

Introduction

Scrolling is one of the most common tasks performed when writing, editing or browsing a document. As it is done so often, even a small improvement results in a significant overall saving in time. Hence, if it is not optimised, a large amount of time and money is being wasted on the mundane task of scrolling. This provides a strong motivation to ensure that the methods we are using are indeed the best. In a study of five hours of web use, Byrne, John, Wehrle & Crow (1999) observed that users spent 40 minutes scrolling. They commented that "An obvious case where widget design could make a difference is scrolling."

The traditional techniques used in document navigation are panning, a scrollbar, and keyboard based scrolling. Modern Windows operating systems have standardised the concept of a new type of scrolling system: rate-based scrolling. In Windows XP, pressing the middle mouse button inside a window that has a scrollbar invokes rate-based scrolling. It has therefore become important to understand how rate-based scrolling complements the existing scroll techniques, and how to optimise rate-based systems.

Rate-based scrolling is an alternative approach where the displacement of the mouse from the point where scrolling began is mapped to the speed of scrolling. Thus, the further the mouse is moved when in scrolling mode, the faster the document scrolls. This method successfully provides smooth scrolling within large documents. However, rate-based scrolling suffers from a limitation on its speed. Attempting to scroll very rapidly means that the information becomes blurred. This imposes a relatively low maximum limit on the speed of scrolling.

A solution to this problem which attempts to retain all the advantages of the rate-based system is Speed Dependent Automatic Zooming (SDAZ). The concept of SDAZ was first investigated by Igarashi & Hinckley (2000) who suggested that the system invoke an automatic zoom-out once the scrolling speed exceeded a predefined velocity. The zoom-out reduces the problem of blurring as the document may be scrolled more rapidly without the information moving across the screen as quickly. SDAZ thus retains the advantages of rate-based scrolling while removing the problems associated with rapid scrolling, hence allowing the rapid acquisition of targets that are a great distance away.

The major difficultly surrounding SDAZ is its calibration. Implementing an SDAZ system requires the developer to make a number of non-trivial decisions regarding the functioning of the system. We had initially thought to compare SDAZ to other forms of scrolling, but were immediately plagued with the problem of calibration — the difficultly of which had not been clearly stated in the previous literature.

This report elaborates on the issues in calibration of SDAZ systems. We look at the mathematical equations and constants driving the system, and also the human-system interface questions. We provide a comprehensive review of the difficulties and attempt to obtain an optimal implementation of SDAZ. Using our implementation of SDAZ, we compare SDAZ against the more standard scrolling techniques: scrollbars, and rate-based scrolling. To do this comparison we use the experimental paradigm developed by Hinckley, Cutrell, Bathiche & Muss (2002), which models scrolling using Fitts' Law.

Fitts' Law (Fitts 1954) has been a well recognised formula for modelling the acquisition of visible targets in computer interfaces since the work of MacKenzie (1991). However, Hinckley et al. proposed that this model could be extended to model scrolling — where the targets being acquired are not initially visible. They believe that the Fitts' Law parameters of distance and tolerance are suitable for being used to provide experimenters with a practical method to quantify scrolling techniques. We have adopted their paradigm for our formal evaluation of SDAZ.

Previous work on SDAZ and related topics is described in Chapter 2. In Chapter 3 we propose a classification of document navigation goals and discuss how rate-based scrolling and SDAZ complement the traditional document navigation techniques in aiding users to achieve these goals. We examine both the mathematical issues surrounding the calibration of SDAZ systems and the human-system interface aspects of the system in Chapter 4. In this investigation we attempt to obtain an optimal implementation through user feedback. In Chapter 5 we describe our formal evaluation of SDAZ against both rate-based and normal scrolling. A discussion of our results follows in Chapter 6, and our Conclusions with suggestions for future work in Chapter 7.

Chapter 2

Related Work

The related work done on this topic falls into four main categories:

- Previous work on SDAZ.
- Orders of Control
- Fitts' Law studies.
- Physiological analysis.

2.1 Previous Work on SDAZ

Igarashi & Hinckley (2000) identified the problem of disorientation caused by rapid scrolling, and proposed Speed-Dependent Automatic Zooming (SDAZ) as an attempt to solve this. An SDAZ system couples an automatic zoom control with scrolling in such a way that when the user scrolls rapidly, the system automatically zooms-out to reduce blurring. Igarashi & Hinckley created several applications that used SDAZ to scroll: a web browser, a map browser, a sound editor, an image viewer, and a dictionary viewer. They concluded that the sound editor, image viewer and dictionary viewer were unsuccessful and not worth pursuing further as applications for SDAZ because SDAZ did not appear appropriate for applications of that nature. However, they believed that the web browser and map browser were worth further investigation — though their informal preliminary evaluation of the systems showed no significant difference between SDAZ and scrollbars.

The idea that there are potential gains to be made by coupling zooming and scrolling has already been validated by Fisheye systems. Gutwin & Skopik compared performance of three types of Fisheye systems with two normal scrolling systems at path following, and found the performance of the Fisheye systems to be significantly better. This shows that there are potential gains to be made by coupling the concepts of scrolling and zooming.

Cockburn & Savage (2003) picked up on the ideas of Igarashi & Hinckley and performed a more formal evaluation of SDAZ and scrollbars on a document browser and a map browser. They found SDAZ to be a 22% faster on the document browsing tasks, which involved the locating of pictures and headings. On the map browsing task they found that participants were 43% faster using SDAZ to locate targets. Their analysis of user preferences also indicated that participants preferred the SDAZ system to the standard scrollbar system.

An earlier paper written by us this year for a COSC 411 (Advanced Human-Computer Interaction) assignment, investigated the maximum speed of information flow that users found tolerable. A modified version of this paper has been accepted for publication in January 2004 (Wallace, Savage & Cockburn 2004). Users had to estimate, at two different magnifications, what they felt a "comfortable" and "maximum-tolerable" scrolling speed would be. Our conclusions of this work focused on the applications of our results to SDAZ systems. We suggested that an appropriate maximum scroll rate at full magnification was probably somewhere between 1.5 and 2.7 pages per second, with the optimum likely being much closer to 1.5. We also investigated our theory that in SDAZ systems the magnification could be altered automatically in such a way as to always keep information on the screen for a fixed length of time regardless of the scroll speed. Our data, however, did not support this idea and suggested that the desirable on-screen-time of data is dependent on magnification. Our results did serve to validate the fundamental idea of SDAZ — our data showed that incorporating zooming "allows for much more rapid navigation through the document space while maintaining the same level of user comfort."

2.2 Orders of Control

Computer control systems generally fall into one of two categories: position control, or rate control (Zhai, Smith & Selker 1997). Position control, also known as zero order control, maps the device's displacement to displacement with the system being controlled. For example, a scrollbar is a zero order control system as there is a linear relationship between the position of the mouse when dragging the scroll thumb and the position within the document. Rate control, also known as first order control, maps the input device's displacement to velocity within the controlled system. For example, in rate-based scrolling the distance the mouse moves corresponds to the speed at which the document scrolls. The relationship can be modelled by a first order differential equation, hence the name "first order control" (Poulton 1974).

One known advantage of position control systems is that they support reversible actions better (Balakrishnan, Baudel, Kurtenbach & Fitzmaurice 1997). It is very easy, for example, for a user to grab the scroll thumb to go up a couple of pages to check something and then come straight back to where they were working before very rapidly. Balakrishnan et al. point out that this is not as easy to do with a rate control system. Our analysis in Section 3.1 and our experimental results (see Chapter 6) agree strongly with this.

It seems to be accepted that for such tasks as navigation in large spaces, rate control systems are better (Balakrishnan et al. 1997). Zhai et al. point out that when scrolling through a large document space it is desirable to control not only the position of the document but also the rate of the information flow so that the user can comfortably scan the document, and suggest that a rate control system would be superior to a position control system for doing this. They argue that with a position control device, the user may not be able to control the speed of movement continuously and that physical constraints (such as reaching the edge of the mouse pad) would mean the user has to stop scrolling and the re-engage when scrolling over a long distance.

The SDAZ system proposed by Igarashi & Hinckley and examined by Cockburn & Savage is a rate control system. SDAZ can thus be viewed as a modified version of rate-based scrolling. Therefore the natural comparison to perform with SDAZ is to compare it to a standard rate-based system, with the question in mind of "do the modifications make it superior to the standard system?" Though Cockburn & Savage demonstrated that SDAZ was superior to standard position control systems, it does not necessarily validate the idea behind SDAZ as it has not been compared to rate control systems. Therefore our evaluation of SDAZ seeks to compare it directly to a standard rate-based system — we also include scrollbars in our evaluation for completeness, but they are not the focus of our comparison.

2.3 Fitts' Law

Guiard, Beaudouin-Lafon & Mottet (1999) suggested that Fitts' Law (Fitts 1954) could be extended beyond modelling acquisition of a single visible target, to a model of navigation in general. Fitts' Law is a very well understood and heavily researched model that has long been used to model target acquisition (See, for example, such studies as Card, English & Burr (1978), Epps (1986), Jagacinski & Monk (1988), Gillan, Holden, Adam, Rudisill & Magee (1990), and MacKenzie (1992)).

Fitts' Law states that the time for target acquisition is linearly dependent on the Index of Difficulty (IoD):

$$T = a + b\,\mathrm{IoD} \tag{2.1}$$

The IoD, which is measured in bits, is calculated as a logarithm of the ratio of distance to target over target width:

$$\mathrm{IoD} = \log_2(\frac{D}{W} + 1)$$

Guiard et al. (1999) suggested that Fitts' Law could be extended from a model of visible target acquisition to a model of navigation in general. In particular, they had in mind the ability to model multi-scale pointing systems. Multi-scale target acquisition is where the user is provided with two controls (usually scrollbars) one of which performs fine-grained movements within the navigation space and the other course-grained movements. Therefore to locate the target, the user first moves the course-grained bar to approximately the right position and then makes minor adjustments with the finer-grained control. Such controls are usually used where the information space is very large compared to the level of accuracy required. In a large information space (for example a thousand page document), a normal scroll thumb becomes very small and moving it one pixel can jump multiple pages through the document, making navigation extremely difficult. The relevance of Guiard et al.'s research to our work is the idea that Fitts' Law can be used to accurately model scrolling.

Hinckley et al. (2002) took this idea a step further and proposed "a formal experimental paradigm designed to help evaluate scrolling techniques" using Fitts' Law as the basis for this paradigm. They believe that the Fitts' Law parameters of distance and tolerance are suitable for being used to provide experimenters with a practical method to quantify scrolling techniques. They propose the use of the standard "reciprocal tapping task" (Fitts 1954) where the user moves back and forth several times between the two targets as rapidly as possible. After discarding the data from the first couple of taps (as the user is still getting a feel for where the target is) the mean time it took for each distance is calculated for each scrolling system. A graph can then be plotted comparing time against IoD for each system, which should give a linear relationship according to Equation 2.1. This can yield interesting comparisons between scrolling systems — for example, one system might be better for short distances and another better for long distances.

2.4 Physiological Analyses

It is worth having some idea of exactly what causes fast moving objects to blur. The contents of this section are dealt with at greater length in our earlier work (Wallace et al. 2004).

Blurring occurs when information is moving too rapidly across the retina of our eyes for our visual systems to process properly. A comfortable image velocity is around 2 degrees per second (Kelly 1979), though a person with good sight can tolerate an image retinal velocity of up to 3 deg/s without blurring occurring (Morgan & Benton 1989). The human eye has a built-in functionality whereby it naturally tracks moving objects, thereby reducing their retinal velocity and stopping blurring. This is known as smooth-pursuit eye movement (Eckert & Buchsbaum 1993). The use of smooth-pursuit eye movement allows us to watch objects that are moving in excess of 9 deg/s (Eckert & Buchsbaum 1993), perhaps even up to 100 deg/s (Blohm & Schreiber 2002), as the tracking of the target reduces the retinal image velocity to a manageable level.

Blurring is caused, therefore, when the smooth-pursuit system cannot be used properly. In document scrolling, for example, this can occur because the user must fix on a feature of the document and start following it as it moves across the screen with their smooth-pursuit system in order to identify what it is. By the time they have identified it and found that it is not the piece of information they are looking for and jumped their eyes back to another prominent feature, other information may have scrolled off the screen. The user will likely then give up trying to track all the targets and hence just see the information on the screen as a blur.

Card, Moran & Newell (1987) provide a model of human cognitive capabilities which suggests it takes about 200ms to spot a prominent feature in a display. It then takes about 150ms for the smooth-pursuit mechanism to activate (Rashbass 1959), and the image must then be stabilised on the retina for 100–125ms before it can be identified (Burr (1980), Card et al. (1987)).

The situation in document scrolling is more complex however. There is the presence of multiple prominent moving features on the display at once, and the fact that information is continually disappearing of one edge of the screen and more is appearing on the other.

We are also unaware of any previous work (apart from our own) that investigates the effect magnification has on the human ability to recognise moving targets.

Chapter 3

Document Navigation: Goals and Support Mechanisms

Some forms of document navigation are performed effectively by current methods, while others are not as well catered for. It is worth examining different types of document navigation in order to gain an understanding of the strengths and weaknesses in current methods of navigation and to obtain insight into just how the addition of SDAZ functionality would complement the standard techniques.

3.1 Our Classification of Document Navigation

We are not aware of any existing taxonomies of different navigation actions. We therefore propose the following characterisation of the different types of document navigation:

- Spatial-Locate

  This is where the user knows where in their document their target is. For example "I want to look at a page which I remember is about one-third of the way through this document." This is analogous to picking up a book and knowing approximately which section of the book you want to open it at.

- End-Point

  This is where it is desirable to navigate to the very start or very end of the document. This might often form the first part of a two part navigation action. For example, "I want to go to the page that is two pages from the end of the document".

- Pagination

  This is where the user wants of make minor adjustments to the information displayed on their screen. For example the user might be focusing on an item near the bottom of the screen and want to centre the screen on that location, or perhaps to look at the previous page.

- Search-For

  This is where the user remembers a certain phrase that occurs somewhere in the document and wish to navigate to that occurrence, or all occurrences thereof.

- Page-Locate

  This is where the user knows the page number of their target location. They may have a hard-copy open in front of them at the desired page, or they may have a reference to a page number in the document.

- Reacquire

  This is where the user wishes to return to their last location. An example of this would be the "Back" button on a web-browser. In document navigation this action would occur if the user was editing a section of a document and wanted to quickly check another relevant section before jumping back to the section they were editing.

- Zooming

  This is where the user wishes to change the magnification of the document, either to zoom-in and view it in finer detail in order to read small print or a detailed diagram, or to zoom-out to obtain a more general overview, or to return to the normal level.

- Paging

  This is where the user is reading or skim-reading the document and wants to keep the part they are examining on the centre of the screen as they move slowly and linearly through the document.

- Section Jump

  This is where the user wishes to navigate to a section heading. For example "I want to go to the 'Related Work' heading" or "I want to go to Chapter 4".

- Visual Browsing

  This is where the user is unfamiliar with the document and wants to get a feel for its contents. The user wants to get an overview of the type of information contained in the document, important sections and graphics. This is akin to riffling the pages of a book in order to see approximately what it contains.

- Visual Search

  This is where the user has a known visual target in mind. For example "I want to find the red graph" or "I want to find a particular page and I'll know it when I see it".

3.2 The Place of Current Methods within this Classification

It is clear that the current tools achieve some of these tasks effectively. For example, a scrollbar seems by far the most effective tool for Spatial-Locate tasks. The user can grab the scroll-thumb and move it to the appropriate location. Even if the document is too large for use of the scroll-thumb to accurately acquire the target, it will still be best for the scrollbar to adjust the position as close to the target as possible and then to acquire the target by using Visual Search. End point navigation is also a solved-problem: It is standard practise for Ctrl-Home and Ctrl-End to navigate instantaneously to the beginning and end of a document. Pagination also is easily achieved by use of the Pg-Up and Pg-Down buttons, and most applications achieve Search-For navigation by providing a "Find" function or similar search utility.

There seems to be little specific support for Page-Locate and Reacquire actions in most existing systems. Though the Adobe Acrobat PDF viewer displays the number of the current page beside the scroll-thumb when the users users it to scroll, which allows for rapid Page-Locate actions. Past research suggests that position control devices such as scrollbars are actually quite effective at performing reacquire actions (Balakrishnan et al. 1997). This is presumably because a position control interface turns a Reacquire task into a relatively simple Spatial-Locate task.

In our experience, zooming is usually performed fairly rarely within document browsers. Users tend to set the application to a readable level and from thereafter remain in that mode. Nevertheless, many document browsing applications provide the ability for the user to alter the magnification as part of their standard functionality. This functionality is usually implemented from a drop-down list of different settings, or from a tool bar button changing the system mode so that the pointer is a magnifying glass and clicking results in the document magnification changing.

It is in Paging that rate-based scrolling finds its niche. Paging can be accommodated in small documents by dragging the scroll thumb slowly, however in large documents this is impractical as the smallest movement of the thumb causes the document to jump rather than move smoothly. Use of the Pg- keys is inadequate for the same reason — the sudden jump these cause disorientates the user who has to re-orientate themselves before continuing to read. Far more satisfactory for this task is rate-based scrolling where the document scrolls continuously at a speed controlled by the user. This allows the user to keep their focus at the centre of the screen while the document flows continuously and smoothly at a rate they control. Panning has often been used to accomplish paging, however rate-based scrolling seems preferable to panning as it is smoother and does not require continuous mouse movement.

Section Jump is performed most effectively by the use of an overview+detail system, where one window contains a listing of the main sections in the document allowing one click to jump straight to them, while the other contains the document itself. A major disadvantage of such systems is that the extra window adds to the screen clutter.

3.3 The Place of SDAZ within this Classification

Current methods do not so effectively achieve Visual Search, and Visual Browsing. It is in these areas (and to a lesser extent Paging and Section Jump), that we believe SDAZ has something worthwhile to offer.

The most obvious advantage of SDAZ is in Visual Search. This was the task type tested in the evaluation by Cockburn & Savage and we believe it was the reason they obtained such positive results. SDAZ provides a unparallelled way of smoothly navigating through large documents while significantly reducing the blurring limitation of standard rate-based scrolling. The rate-based foundation of SDAZ means that scrolling is smooth, while the automatic zoom-out means that the user can navigate more rapidly through the document.

Visual Browsing is facilitated by SDAZ as a result of the zoom-out. Whenever the user scrolls rapidly, the automatic zoom-out allows the user to view a wider area of the document giving them a better feel for its overall contents. In Paging, SDAZ is equivalent to rate-based scrolling since at low speeds there is no difference between the systems. In performing a Section Jump, SDAZ is perhaps not as good as an overview+detail system. However, since section headings are likely to be large and bold, they will stand out when zoomed-out (though perhaps they will not be readable) and could be easily navigated to if the user had a moderate knowledge of where in the document to find the section they were looking for. Also, an overview+detail system usually requires the user to change their point of focus to a different window in order to selection the section, whereas in SDAZ systems their focus is maintained on the document.

SDAZ cannot be properly said to support “Zooming” as we have classified it. Certainly the automatic zooming functionality provided by SDAZ is a form of zooming. However zooming in SDAZ occurs to aid visual search, while normal “Zooming” as we have classified it here is performed either for obtaining an overview of the document or for viewing an image close up.

In the evaluation of SDAZ performed by Igarashi & Hinckley, the zoom-out was semantic based — section headings and graphics were kept fairly large while body text was shrunken further. By using our classification it can be seen that this would have resulted in a trade-off. Visual Search and Visual Browsing capabilities were being sacrificed in favour of Section Jump. Their system did not preserve page layouts which we believe would have reduced its usefulness for performing Visual Search and Visual Browsing, and their system became increasingly an overview+detail system as it zoomed out — probably increasing its ability to perform a Section Jump.

3.4 Using SDAZ with Complementary Methods

Having examined the different types of document navigation and analysed the suitability of the current techniques and SDAZ at performing these, we are in a position to judge how SDAZ should be combined with the standard methods.

We have seen that SDAZ should be viewed as an addition to the current arsenal of scrolling techniques rather than a replacement for them. SDAZ represents a modification of rate-based scrolling which is equivalent to it at low speeds, meaning SDAZ performs well in certain types of document navigation which are not handled well by the more traditional scrolling techniques. SDAZ hence should be combined in a system with the standard navigation methods in order to complement them, rather than replacing them.

Chapter 4

Calibrating SDAZ

Obtaining the optimal SDAZ system is a non-trivial matter. It is on the analysis of this problem that the bulk of our work here is focused. The settings that must be calibrated fall roughly into two categories:

- Control equations and constants.

  These are the mathematical formulae behind the system. Setting these involves addressing such questions as:

  – How fast does the user need to be scrolling before the system should start zooming out?

  – When the zoom out does start, what formulae govern the relationships between mouse displacement, document speed, and magnification?

- Human-System interaction

  These are the significant questions that must be addressed regarding the user's interaction with the interface.

  – What does the user have to do to scroll?

  – How does the system represent the scrolling mode and the information associated with it to the user?

  – What happens to the cursor when scrolling?

  – How does the system convey the zoom-in location?

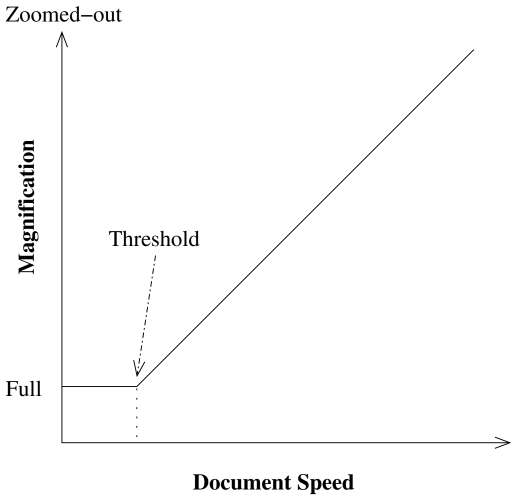
In order to address these questions a process of iterative implementation was used. Fourteen different users (mostly computer science students very familiar with scrolling and 3 with significant previous experience with SDAZ) used and commented on the system over the course of development. The basic SDAZ system upon which these modifications were analysed was developed originally by Joshua Savage in C/OpenGL (Savage 2002).

4.1 Control Equations and Constants

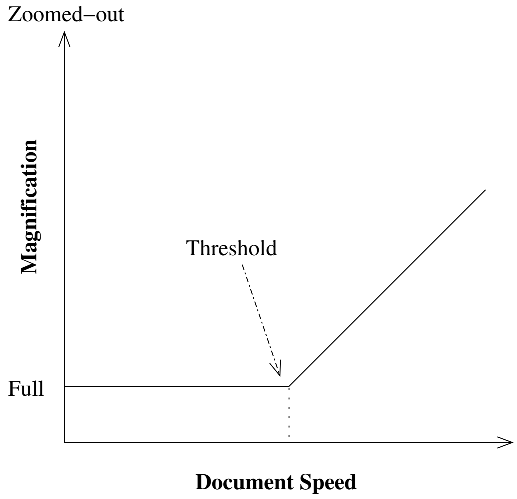
The following is a summary of the units and terms used in this section:

- *Document speed* is the rate at which navigation through the information space is occurring, and is measured in units of *pages per second*.

- *Screen time* is the length of time any given piece of information is displayed on the screen for, and is measured in units of *seconds*.



(a) Small Threshold



(b) High Threshold

Figure 4.1: Illustration of Thresholds

- *Magnification* of the document is measured in *pages*, and refers to the number of pages visible on the screen. A magnification of 1.0 is considered "full magnification" while a magnification of 3.0 would mean three pages are visible on the screen — and therefore the pages would be one-third their usual size.

Given these definitions, the following relationship holds:

$$Screentime = Magnification/Documentspeed$$

4.1.1 The Threshold Constant

The *threshold* is the document speed at which zooming out begins. Thus, at any speed less than the threshold, the system will be at full magnification. A threshold that is too high results in the document blurring when scrolled fast because the high threshold means the document is not zoomed-out far enough to compensate for the speed. A threshold that is too low is annoying to the user because even when scrolling slowly the system will counter-intuitively zoom the document out. Figure 4.1 gives an illustration of the effect the threshold value has on the function of the system by showing the relationship between magnification and document speed.

We presented our test users with a range of nine different settings for the threshold value, ranging from 1.0 to 5.0 pages per second, at intervals of 0.5. Users were given a working SDAZ system on which they could dynamically adjust the threshold value by pressing number keys corresponding to the nine available settings in increasing order.

All user preferred the lower end of the range available. One user, expressing the unanimous opinion of the group, commented that 4.0 pages/s was "massively too fast". All users felt that the optimal value lay somewhere in the range 1.0 to 2.5 pages/s. Several users preferred 1.0, but after further discussion opted for a higher value — we discuss this in the next section. One user, however, commented that "[1.0] is too quick [to zoom out], I do spastic actions."

We asked users to select a single most preferred value. Many users were uncertain about precisely which value they preferred. The majority of users, however, eventually decided that 1.5 pages per second was the optimal value.

Two Conflicting Types of Navigation

It became evident to us in testing our threshold values that there is a tension in the system between two types of scrolling task. Using the classification scheme presented in Section 3.1, this tension can be understood as Paging and Pagination versus Visual Search and Section Jump. Unfortunately it appears that they require quite different system behaviour. Visual Search or Section Jump involves scrolling rapidly to a distant portion of the document. For these tasks an immediate (but smooth) zoom-out is desirable. For Paging and Pagination it seems strongly preferable that the system remain at full magnification throughout the action. Yet with Pagination the user often wants to move the mouse quite fast in order to rapidly change their point of focus.

In our testing of threshold values, a few of the users (who evidently had only the task of Visual Search in mind) preferred the lowest threshold available: 1.0 pages/s. When the ideas of pagination and paging were discussed with them however, they all opted for 1.5 or 2.0 pages/s as optimal.

This observation of the two types of scrolling results in two options for the implementation of the system:

- To implement the two different tasks types as two different modes of SDAZ. This would mean that the SDAZ system would operate in two modes: with and without zooming. We think that this could be best implemented if pressing and holding the mouse button invoked normal rate-based scrolling (SDAZ with no zooming), but clicking and releasing the button invoked an SDAZ mode which would end when the mouse button was clicked a second time.
- To only support the one mode of scrolling.

Our observations suggest that the difference between the two types of task are not enough to warrant the disadvantages of a dual-mode system. Since most users seemed relatively happy with a threshold of 1.5 regardless of whether they had taken paging and pagination into account, we think that the gains to be made by using a rate-based-only mode in addition to a long-distance-SDAZ mode are probably negligible.

For the system used in our formal evaluation, we decided to use a threshold value of 1.5 pages/second and to support just the one mode of SDAZ.

4.1.2 Zoom Formula

An important question in the calibration of an SDAZ system is what the formula controlling the automatic zooming should be. The basic idea of SDAZ is that the single parameter of mouse displacement should be mapped onto the two variables of Document-Speed and Magnification.

There are three values dealt with in this discussion: *displacement* (referring to mouse pointer displacement — how far the mouse has been moved since scrolling began), *Document-Speed* (DS), and *Magnification*. There are thus three relationships that can be discussed — the relationships between any pair of these three variables. Two formulae will ultimately govern the system — mapping displacement onto the other two variable spaces.

A very important question in the calibration of the system is the choice of these functions. SDAZ appears to us easiest to understand by thinking of it as two equations, the first mapping displacement to DS and the second DS to magnification. It seemed to us initially intuitive that a linear relationship for both would be desirable. See Figure 4.2 for an illustration of this.

What these graphs say is that displacement is mapped linearly to DS. The distance the mouse is moved is proportional to the rate of navigation through the document. This seems to us intuitive — if you move the pointer twice as far, you go twice as fast. The second function maps DS linearly to Magnification (when DS is above the threshold). The idea of mapping from DS to magnification rather than displacement to magnification is that we consider this model easier to understand. Since the purpose of SDAZ is to reduce zooming when the information flow becomes too rapid, it seems intuitive to calculate the zoom-level based on the rate of information flow.

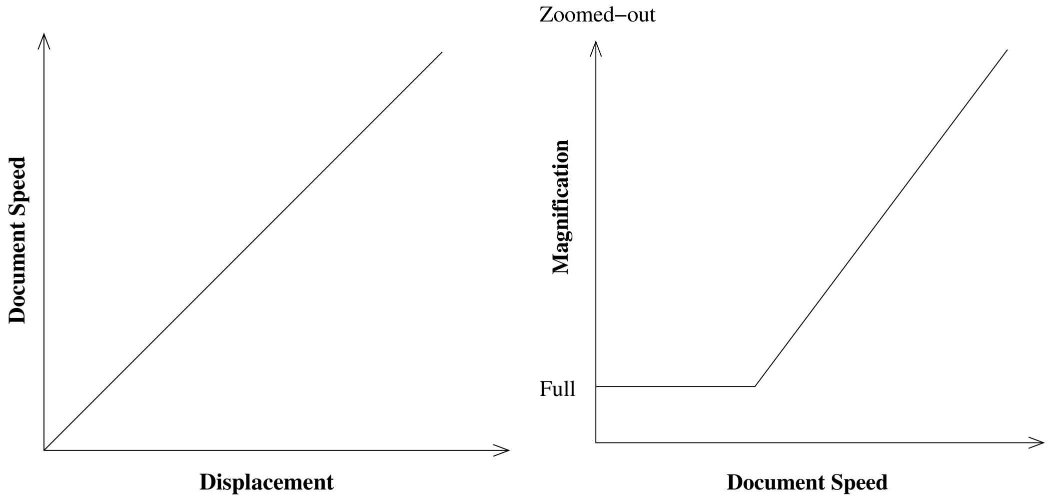


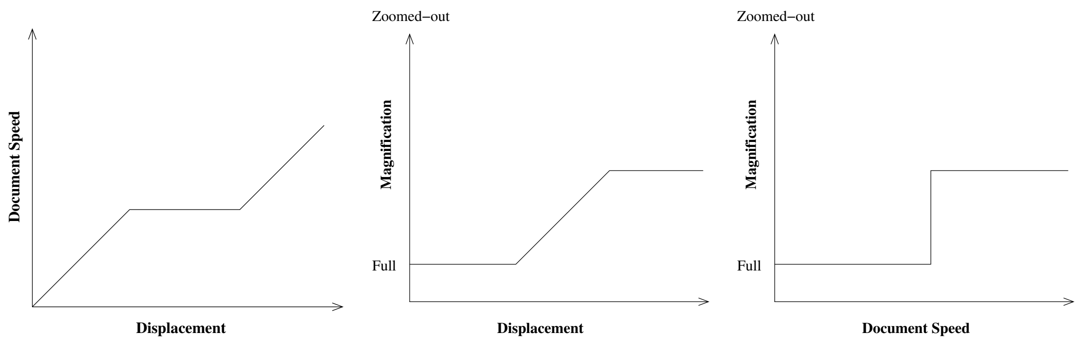
Figure 4.2: Two sample equations mapping displacement to DS to Magnification

An alternative model we considered, was to map displacement to magnification and then calculate DS in such a way that screen time was held constant. However our earlier investigation demonstrated that desirable screen time varied with magnification (Wallace et al. 2004).

One to One Mappings

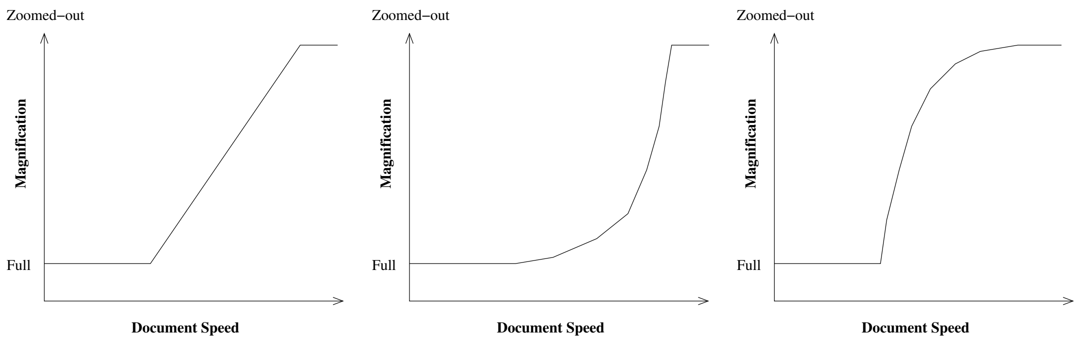
A consideration in the choice of these functions is whether it is desirable to choose the functions in such a way as to ensure a $1-1$ mapping across the three information spaces when the value of DS is exceeding the threshold. (That is, given a value of any one of the three variables, is it possible to calculate the other two?) If the functions chosen mean such a mapping is not preserved then the decision as to which variable spaces the functions should map between becomes relevant. An example is shown in Figure 4.3.

This shows that it will not always be possible to have a function mapping DS to Magnification. Therefore, though a mapping from Displacement to DS and then another from DS to Magnification is an easier system to comprehend (and will be used throughout this report), it must be acknowledged that this model *does* result in a loss of generality. A model which used a



(a) Mapping 1 (b) Mapping 2 (c) Result

Figure 4.3: The choice of the two mapping functions from Displacement to DS and Displacement to Magnification mean that the relationship between Magnification and DS is not a function in either direction.



(a) Linear (b) Exponential (c) Modified Exponential

Figure 4.4: The three main control equations investigated

Displacement → DS function and a Displacement → Magnification function would be sufficiently general.

For the sake of convenience and time we have limited our analysis to a direct linear mapping from Displacement to DS and concentrated on the relationship between DS and Magnification. It should be noted however, that Igarashi & Hinckley found a *non-linear* relationship between DS and Displacement to be more satisfactory — their settings are given in Table 4.2. Though we did not evaluate this directly, the results of our investigations have led us to think that Igarashi & Hinckley were probably right as we obtained non-linear results in our investigation of the relationship between displacement and DS.

Our Analysis

The situation is still quite complex when analysing only the DS → Magnification mapping. We considered three types of equations to govern this mapping: Linear, Exponential, and Modified Exponential. See Figure 4.4 for an illustration of these. These seemed to us to be the most important cases.

Obviously if we zoom out too far the information displayed becomes so small as to be unreadable. Therefore we imposed an arbitrary limit of Magnification $\leq$ 10 pages. Once the curve reaches a Magnification of 10 pages on-screen, it will not zoom out any further — though DS may continue to increase if the user continues to displace the mouse further.

We also considered the possibility of various slopes for these curves. The question of how steep the curve should be is very important. If too shallow, the system does not zoom out enough to compensate for the speed, which results in blurring — precisely the problem we are trying to avoid. Too steep, and the intermediate zoom levels are no longer easily accessible and the user feels disoriented when the system zooms out too fast.

The mathematic representations of the three equations we chose to investigate are displayed in Table 4.1. Note that these relationships only take effect when DS > Threshold. When DS $\leq$ Threshold, the magnification is set to 1.

$$Magnification = Slope * (DS - Threshold) + 1 \qquad \text{(line)}$$

$$Magnification = e^{(DS-Threshold)*Slope} \qquad \text{(exponential)}$$

$$Magnification = 10.0 - 9.0 * e^{(Threshold-DS)*Slope} \qquad \text{(modified exponential)}$$

Table 4.1: The three equations we investigated.

We presented users with an interface allowing them to switch between the three equations and select nine different settings of slope at equally spaced intervals by pressing number keys. These possible settings ranged from 0.1 to 0.9 on the line equation and 0.03 to 0.26 on the exponential equations.

User Comments

Users were encouraged to experiment with various settings available to them, they could toggle freely between the nine slopes by pressing the appropriate numeric key and between the three equations by a similar key-press.

It was our belief prior to performing this evaluation that users would prefer the exponential equation, as we assumed that a smooth curve would feel smoother and more natural. However, users universally disliked this curve and felt that both the line and the modified exponential were better. One user commented that the exponential curve just "seems odd".

We attempted to clarify the exact objection, by talking the problem over with the users. It appears that the problem is in the area of the curve nearest the threshold. Users said this shape was worse than the line because it did not zoom out as quickly. One user (showing remarkable insight into the shape of the equations) explained that this one did not zoom out as quickly as the line and that the opposite curve shape would be better — effectively describing the modified exponential curve (which he liked best when he tried it later).

Users unanimously felt that the line and the modified exponential curve were distinctly superior to the exponential curve. However the distinction between the former two was not so pronounced and it would often take users about ten minutes of using the system before they decided which one they liked better. All users eventually decided they preferred the modified exponential equation over the line equation, though most were very hesitant about making that judgement.

Regarding the optimal values for the slope, users mostly preferred values in the range 0.6 to 0.9 for the slope of the linear curve (0.7 seemed the single most preferred value) and 0.14 to 0.17 for the slope of the exponential. The preferred slope values for the modified exponential appeared to be widely distributed, with some people preferring values around 0.11, while others preferred 0.23 or 0.26. The consensus however appeared to be for a slope of 0.23 with the modified exponential.

Figure 4.5 shows a graph of the three control equations at the preferred threshold setting and slope values — a threshold of 1.5 and slopes of 0.7 (linear), 0.17 (exponential) and 0.23 (modified exponential). An interesting feature which immediately stands out is the way the three curves almost intersect around the area of $DS \approx 14.3$, $Magnification \approx 9.7$.

We believe that this feature is an indication of the optimal speed at the maximum magnification. As noted earlier, our magnification was limited to 10 pages, meaning that once the magnification had reached that level, no further zooming out could occur. We noticed in our testing process that for a large proportion of the time, the system was either fully zoomed in (when the user had stopped on their target) or fully zoomed out (when the user was scrolling to their target). Thus, we think that when we asked users what "slope" of each equation they liked, we were effectively simply asking them to select an appropriate DS value for a magnification of 10. Since we had already set the threshold value and defined the general formula of each curve, the user's choice of such a value would have uniquely defined the appropriate slope.

For our system used in our formal evaluation we use the modified exponential curve with a slope of 0.23, since this setting appeared to be preferred by the majority of users.

Two Additional Curve Types

We considered two other possible equation types, but did not test them. These were the Combined-Curve and Two-Level curve. (See Figure 4.6)

The Combined-Curve could be modelled either as an exponential and a modified exponential joined piecewise, or as a segment of a sine function. We thought of this curve as it is smooth throughout. Though we did not test the Combined-Curve directly, we believe that the same negative results would be obtained for it as were obtained for the exponential curve. Both curves share an initial shallow magnification increase which was strongly disliked by users.

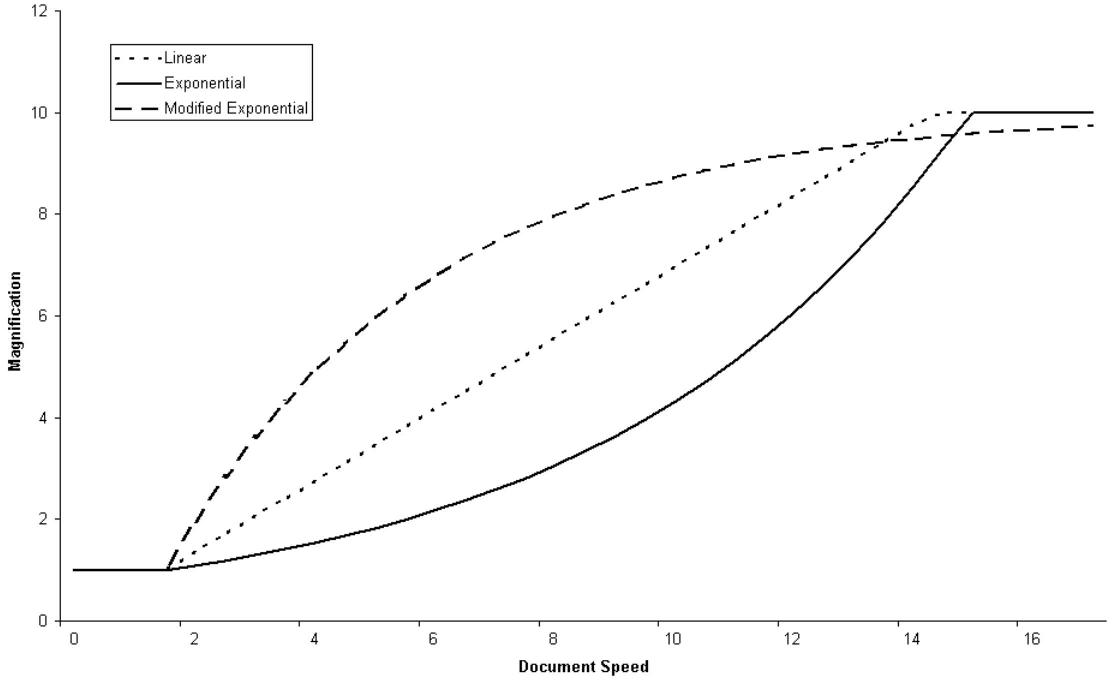
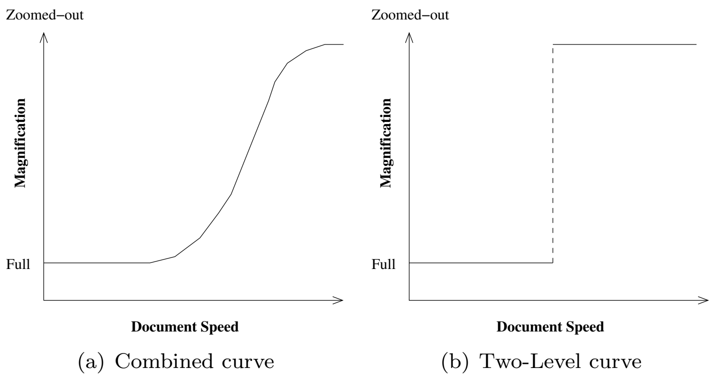


Figure 4.5: The most preferred settings of the three control equations investigated

The Two-Level curve we considered since many users seem to operate the system in such a way that for most of the time the system is either fully zoomed-in or fully zoomed-out. The Two-Level curve would have two levels of magnification and would animate smoothly between them when the threshold was crossed. We did not bother investigating this curve as many of our testers told us they preferred the functionality of having access to the intermediate zoom levels. Also, our users did not always choose the steepest slope out of those we offered, suggesting that an infinitely steep slope would not be appreciated.



(a) Combined curve (b) Two-Level curve

Figure 4.6: Two additional control equations considered

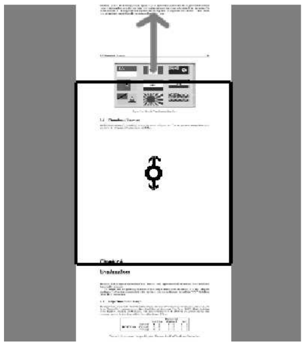


Figure 4.7: The “Hunting Effect”. As the system zooms-in on the current location, the target graphic moves away from the pointer until it is partially off the screen.

4.1.3 Falling Rate

If the user ends scrolling using a mouse button when the system is not at full magnification, then the system needs to be returned to full magnification. Both Igarashi & Hinckley and Cockburn & Savage used a smoothly animated transition since an instantaneous transition caused disorientation (this was our experience also). Both groups solved this by imposing a limit on how rapidly the document can ever zoom-in (the “falling rate”) and then zooming-in at the maximum rate when scrolling stops. Neither group, however, reported the value used for this rate. We tried a few settings informally and decided to use the following falling-rate (t indicates time, in seconds):

$$\frac{\delta(Mag)}{\delta(t)} = 10 * Mag$$

Hence if the current magnification is eight, then the falling rate will be 80 pages per second. However the falling rate is continuously recalculated during the zoom-in based on this formula. This will mean that as the document approaches full magnification the value of the magnification is not being changed as fast. This is necessary because there is a greater visible difference to the user between magnifications of 1 and 2 as compared to magnifications of 9 and 10.

We selected this formula ourselves based on our own preferences and the system given to the test users used this formula, as does our system in our formal evaluation. No testers made any comments about the falling rate.

4.1.4 The Hunting Effect

SDAZ has been noted by Igarashi & Hinckley to suffer from a problem where users overshoot the target due to the system zooming-in as the user slowed. If the pointer is not on the target, then zooming-in causes the target to move away from the pointer. The user responds by quickly moving the pointer toward the retreating target, but this rapid pointer movement causes the system to start zooming-out again. This problem is known as the “hunting effect” (see Figure 4.7). In short it is where the user has difficultly acquiring the target precisely as the target avoids the “hunting” pointer.

This issue has been also recognised problem in Fisheye systems. The magnification of the Fisheye means that the target is moving fastest across the screen when the pointer is nearest it. Gutwin (2002) presented a technique called Speed Coupled Flattening (SCF) as a proposed solution to this problem with regard to Fisheye systems. SCF works by holding the Fisheye’s magnification level to a minimum when the pointer is moving and then rapidly reapplying the magnification when movement ceases.

Our interpretation of Gutwin’s method with regard to SDAZ was to perform no zooming-in while scrolling and then to rapidly apply full magnification when scrolling ceases. In effect this means that it is not possible to zoom-in when scrolling.

This mechanism also appears to give users a greater feel of control as it allows, as a side effect, the ability to pause the zoom level. When the user reaches a zoom level they feel is adequate, they can reduce their speed (even to zero), without the system zooming back in. In a previous evaluation of SDAZ, one participant commented that such functionality would be desirable (Savage 2002).

An additional advantage of this modification is that it solves the problem of zoom-ins on direction change. Both Igarashi & Hinckley and Savage reported the problem that changing direction caused a sudden zoom-in. As the user changes direction their velocity is momentarily zero, meaning that the system sets the magnification back to full. As the user begins to move in the new direction, the system zooms-out again. From the user's point of view the document appears to jerk as magnification is momentarily set to full before being zoomed-out again. Igarashi & Hinckley solved the problem by introducing an arbitrary delay in the zooming-in process, while Savage decided not to include this delay in the system used for evaluation. A few of our test users also commented on this: "The bouncing thing is really annoying," complained one of them.

However, our users were not very positive in their assessment of our modification. They felt this was overly restrictive, and preferred the original version when they compared the two directly. All seemed to agree that the ability to zoom back in was a good thing, but that it should not happen as fast as it was occurring in the standard system.

We therefore decided to modify the falling rate control equation that is used in scrolling mode. The user "falls" back to the document at normal speed when scrolling ceases however. Our new falling equation is as follows (t indicates time in seconds):

$$\frac{\delta(Mag)}{\delta(t)} \leq 3 * Mag$$

4.1.5 Three Unevaluated Ideas

- Orders of Control

  We examined different orders of control in Section 2.2. We saw that SDAZ is a first-order control system as it maps mouse displacement to scrolling *velocity*, whereas traditional scrollbar and panning systems are a zero-order systems as they map mouse displacement to document *displacement*.

  Poulton (1974) says that "A control system of high order is improved by adding components of lower order." He notes that "Control systems may combine 2 or more orders of control" and gives the example of "rate aiding" which is where a first-order system is supplemented by adding a zero-order component. This would suggest that SDAZ could be improved by adding a zero-order component to the formula. Hence any SDAZ system could be improved by adding an additional movement formula mapping mouse displacement into document displacement.

  Our own observations suggest that such a addition would be very useful. When the user begins a scrolling movement, the system typically feels a little unresponsive at first because the user has increased the scrolling speed but the system has not scrolled any significant distance through the document. Whereas in a zero-order system such as a scrollbar, there is no lag between mouse movement and the document location being altered appropriately. We think that the addition of zero-order control to SDAZ would help the user to feel that the system was responding instantly to their wishes and that they really were moving toward their target rapidly.

  We experimented with adding zero-order control to our system, but had difficultly finding an appropriate constant of proportionality for the mapping between mouse displacement and document displacement. We decided to not include this functionality in the system given to our testers as it would interfere with their choice of appropriate control equations, and we have not used it in our formal evaluation.

- A Lag in the Zoom-Out

  In the process of examining different threshold values we saw that there was a conflict between two different types of navigation action (see Section 4.1.1). One idea we considered using to solve this problem was an idea of a lag in the zoom-out. The idea was that the system should not start zooming out until the threshold value has been exceeded for a predefined length of time. Once the threshold had been exceeded for that length of time then the system could be set (either instantly or by smooth animation) to the appropriate zoom level for the current document speed, and normal SDAZ functionality would then occur.

  We implemented a lag which was followed by instantaneous acquisition of the appropriate magnification. We found the instantaneous acquisition to be disorienting and the lag itself to be annoying as the system would delay before responding to our actions. We did not use this functionality for either the informal testing process or the formal evaluation.

- Stopping Instantly

  Several users of our system commented that they would prefer if it was easier to stop. Scrolling could be ended at any time by using a mouse button (we discuss this further in the next section on Human-System Interaction), however users were concerned about the lack of ability to stop or reverse while remaining in scrolling mode. If a user overshoots their target or sees an item of interest, then they need to be able to stop or reverse quickly.

  To stop in standard SDAZ while remaining in scroll mode, the user is required to to move the mouse by exactly the distance that they have already moved it, but in the reverse direction. If the user is scrolling fast then it can take over a second to do this, during which time the document has been scrolling rapidly in an undesired direction.

  We suggest that it would be useful if users could just jerk the mouse in the opposite direction they are currently travelling, the system would respond by bringing them instantly to a complete halt. This action would need to be distinguished carefully from “legitimate” slowing that users do when navigating, which is prone to be far smoother in nature — involving less rapid mouse movement.

  One user also commented that it was difficult to stop completely while in scrolling mode. In our system, displacing the mouse by even one pixel results in scrolling occurring (though very slowly) and hence the users needed to be pixel perfect in order to be completely stationary in scrolling mode. We suggest a “null-speed region” of several pixels in order to give the users some leeway in this regard.

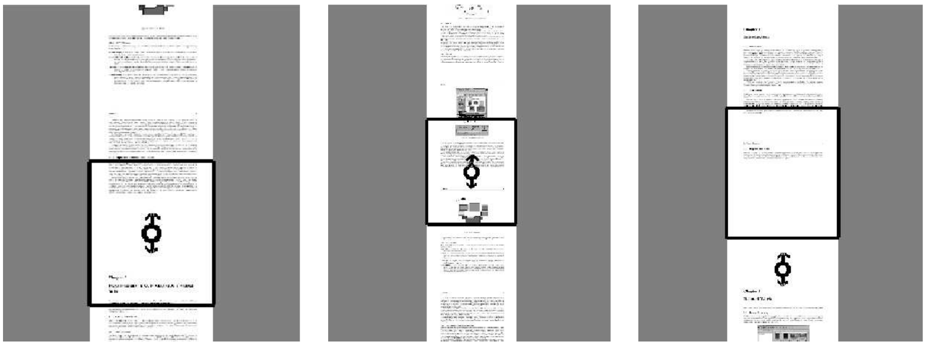
  Our system used in our formal evaluation does not have the instant-stopping functionality, and has a null-speed region of one pixel as described above.

4.2 Human-System Interaction

There are several important issues that need to be addressed regarding the user’s interaction with the interface. Since there are only a few discrete possibilities for most of these issues we are more certain of the accuracy of our results presented here than in our last section where we were trying to obtain the optimal value in a continuous range.

4.2.1 Zoom Location

When we first used Savage’s SDAZ system we noted a serious deficiency in the user interface. The mouse movement during scrolling caused the pointer to move around the screen which naturally invited the user to follow it with their eyes. Thus the user would be looking at the mouse pointer and stop scrolling once the mouse pointer moved over the target. The user would then be surprised when the system zoomed-in at the centre of the screen. This zoom-in at the centre of the screen, rather than the pointer, often meant that the target was no longer on the screen.



(a) Zoom-Pointer (b) Pointer-Centre (c) Zoom-Centre

Figure 4.8: Screen-shots of the three systems.

A number of alternative interfaces were investigated as solutions to this problem. Three possibilities were initially investigated. Figure 4.8 shows screen-shots of the three systems.

- Zoom-in at the moving pointer: *zoom-pointer*.
- Keep the pointer in centre of screen while scrolling: *pointer-centre*.
- Remind the user that zoom-in will be at screen centre while allowing the pointer to move: *zoom-centre*.

Following Cockburn & Savage's implementation we initially warped the pointer to the centre of the screen when scrolling began. We decided to interpret the idea of having a "moving pointer" to mean allowing the pointer vertical freedom, but in all our implementations we disallowed horizontal movement. This meant the pointer was always in the horizontal centre of the screen. We also, when in scroll-mode, replaced the pointer with a semi-transparent double-headed arrow to indicate to the user that mouse movement will cause scrolling rather than pointer movement. In all systems we clearly marked for the user the zoom-in location by using a visually clear red rectangle to indicate this — an idea taken from an as-yet-unpublished paper by Jones, Jones, Marsden, Patel & Cockburn regarding SDAZ on palm displays.

Even with the added visual draw of the rectangle in the centre of the screen, users eyes still naturally followed the moving cursor. Thus the zoom-centre system was a complete failure. "I don't like it" stated one typical user expressing the unanimous opinion on the subject.

Both the zoom-pointer and zoom-centre systems have a potential difficulty with the pointer hitting the top of the screen. Should this place a natural upper-limit on speed, or should the system allow the user to continue increasing speed? In our system this was not a serious problem as the user never moved the mouse as far as the edge of the screen because such a high speed would be inappropriate for the 30 page documents we were using to test it. By the time the user reached such a high speed in our system, they would be at the end of the document already.

Inverted Controls

We observed a serious problem with the zoom-pointer system. As the user scrolls more rapidly, the pointer moves toward the boundaries of the screen. Since the information is scrolling from the same boundary as the user is moving toward, this means the user has a progressively shorter distance in which to spot their target before it passes the pointer. See Figure 4.9 for an illustration of this. This resulted in users overshooting their target when moving at high velocities.

The solution to this that we investigated was to invert the controls. Instead of upward mouse movement causing scrolling toward the beginning of the document (the document moves downward), upward mouse movement causes scrolling toward the end of the document (the document

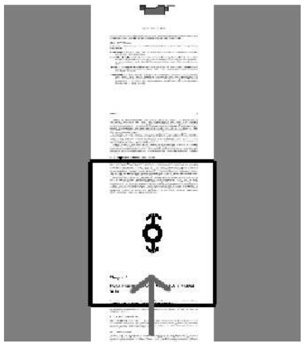


Figure 4.9: The direction of the information flow is indicated. The information has only a short distance to move before it passes the focus region.

moves upward). This means that the pointer is moving *away* from the oncoming information — giving the user an increasing distance over which to spot the approaching target. See Figure 4.10 for an illustration of this.

However, the user response to the inverted-pointer-zoom system was mostly negative. "I like it better the other way [non-inverted]... It's better normal." said one user, stating the almost unanimous opinion. We believe there are two main difficulties with an inverted system.

The first difficulty is that it is unnatural to move the mouse in the inverted direction. However, it could be argued that it is "unnatural" merely because people haven't had much practise at it yet. Indeed, one of our users commented, "This [inverted] is fine. It's no harder getting used to it with the reverse thing than the other way... maybe I've played too many computer games with inverted controls... It probably is better this way [inverted]." It might be argued that this is merely a matter of convention and it can be changed.

We have two objections to this argument. Firstly, having a counter-intuitive system decreases the system's learnability. We think that a novice user will naturally attempt to scroll in the usual way and be put off when the system does not respond as expected. Our second objection is that not only is the inverted direction unconventional, but it means that the user is being asked to move their pointer in the opposite direction to their target. It is asking the user to move *away* from the target in order to acquire it. It is unsurprising most users found this counterintuitive

The second difficulty we see in the inverted system stems from this idea that the pointer is moving away from the target. The pointer movement is actually going to *increase* the effective distance between the pointer and the target. The focus area is being moved away from the target

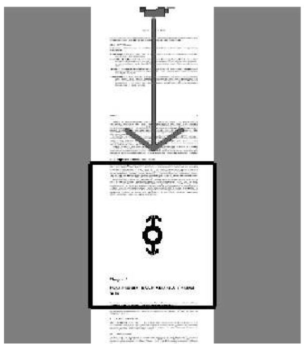


Figure 4.10: The inverted system. Here the information has a longer distance before it passes the focus region.

by the movement of the mouse. This could be avoided, if a zero-order supplement was used (see Section 2.2), so that the document was displaced toward the target by the same distance that the pointer was being moved away from it. However, the argument is moot as all users liked the pointer-centre system over any of the alternatives whether inverted or not.

Pointer-Centre is Best

Users were unanimous in their opinion that the pointer-centre system was superior to the other systems they used. The following comment was fairly typical. "That's probably a good idea actually... Yeah, that's good... I reckon with this way you've got more control... You can keep track better." Users consistently pointed out the way the focus didn't move in this system and that they liked it better because of this. Several commented that this stationary focus meant it was easier to control the system.

We think the stationary reference point also decreases the mental load of users as they do not have to keep track of two moving objects at once (the document and the reference point). We believe that keeping a stationary reference point allows the users to better determine their exact speed through the document. Calculating the speed difference between the moving document and a stationary reference point is doubtless easier for the user than calculating the speed difference between two moving objects.

We also noticed that users tended to focus on anything on the screen that was moving. Only when nothing else on the screen was moving did they focus their full attention on the background document that was scrolling. Obviously the best performance occurs when the users are fully focused on the document rather than on the moving reference point.

4.2.2 Speed Indication

The system created by Cockburn & Savage used an information area to the right of the screen to indicate both scrolling velocity (including the threshold value) and location within the document. Figure 4.11 shows an image of this information area.

As we discussed in Section 3.4 we believe that in a real system this would be a normal scrollbar as we do not view SDAZ as being a replacement for a scrollbar. The scrollbar will convey the location within the document to the user by position of the scroll thumb. This means that the information regarding scrolling velocity needs to be conveyed by some other means. Also, having it displayed at the side of the screen is not really satisfactory as the user's attention should be focused at the centre of the screen on their document.

In the rate-based systems used in modern versions of Microsoft Windows, clicking the middle mouse button of a three-button mouse places a double headed arrow at the current position on the screen and rate-based scrolling occurs with that as the reference point. As the user moves the pointer, the moving pointer takes the form of an arrow and its distance from the reference point indicates scrolling speed.

The zoom-centre system explored one possible way in which the displacement of the pointer from the centre of the screen indicated scrolling velocity. We implemented a second similar system which kept the double-arrow symbol in the centre of the screen and allowed the normal mouse pointer vertical freedom — its distance from the double-arrow indicating scroll-speed. Unfortunately this system suffered from the same failings as the zoom-centre system. Users would focus on the moving pointer and then be surprised when the system did not zoom in there.

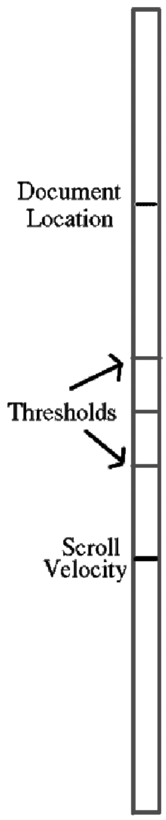


Figure 4.11: The information area used by Cockburn & Savage.

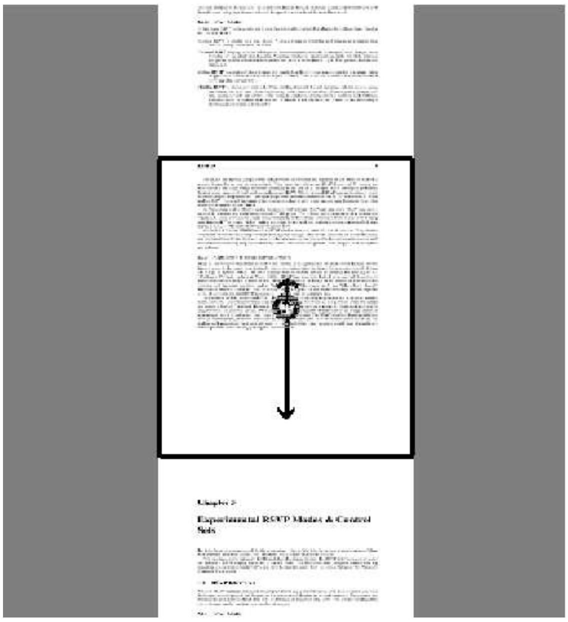


Figure 4.12: The size of the pointer indicates scroll-speed

A second option investigated was to have the pointer grow in size to indicate speed. See Figure 4.12 for an image of this. This possibility suffers from the same problem as the zoom-centre system. The users eyes are be drawn to the "moving" cursor — since the end of it is growing. They expect the system to zoom in on the end of the overgrown pointer.

Our conclusion is that it is preferable to give no indicator of scrolling speed. We found that it is best if nothing on the screen is moving or changing which might in any way distract the user from the information flow and their targeting. We believe that the information flow of the document itself provides an adequate gauge of the scrolling velocity and conveys it more naturally than any arbitrary scale we could devise. Thus, the system used in our formal evaluation gives no indication of scrolling speed.

4.2.3 Mouse Buttons

The implementation of SDAZ by Cockburn & Savage involved pressing and holding the left mouse button while moving the mouse. According to MacKenzie, Sellen & Buxton (1991), such dragging is slower and more difficult than normal pointing. This suggests that the system could be improved by having the user move the mouse freely rather than dragging with it.

We developed a modal interface where clicking invokes the zoom mode and clicking again ends it. Another example of such an interface is the "second mode" of the Microsoft IntelliMouse^{TM} described by Zhai et al. (1997).

During our user evaluations we noticed that a lot of our users were doing excessive clicking as a result of this. When acquiring a target, users generally miss the first time by half a page or so and then require a second scrolling movement to reach their target. In this second movement they are travelling only a short distance, and forcing them to click twice in order to do this (once to start scrolling and once to end it) seems excessive. We therefore decided to have our system support both the modal and the dragging behaviours.

Cockburn & Savage told their participants to use the left mouse button in their evaluation in order to invoke scrolling (though their system actually supported scrolling with any button). We believe that most applications will use the left mouse button already for a purpose other than scrolling. The left mouse button is typically used for issuing commands such as the clicking of a button, the relocation of the cursor, the selection of text, or the following of a hyperlink. The right mouse button typically invokes a context sensitive pop-up menu. Hence we decided to use the middle mouse button for scrolling in our system.

The decision to support both behaviours using the middle mouse button means our system is equivalent in these respects to the standard functionality of the middle mouse button in modern Microsoft Windows^{TM} systems (such as XP) — which invokes rate-based scrolling on a middle-button press with support for both the modal and dragging behaviours.

4.3 Final Settings

We feel it is helpful to give a complete listing of the settings used in the calibration. Table 4.2 lists the different settings of the SDAZ system. For comparison purposes we have included what we know about the settings used by Cockburn & Savage and Igarashi & Hinckley in their evaluations. As already noted, neither group comments much in their published work about what settings they used. However we have been able to obtain privately the information about the settings used by Cockburn & Savage and we publish it here. What we know of Igarashi & Hinckley's settings we have included in this table for the sake of completeness.

Setting	Wallace	Cockburn & Savage	Igarashi & Hinckley
Threshold (Th)	1.5	1.5	—
Max Zoom Level	10	6.6	—
Disp → DS Mapping	Linear	Linear	Exponential
	$DS = 0.05\text{Disp}$	$DS = 0.05\text{Disp}$	$DS = A * B^{\frac{Th-Disp}{Th-C}}$ [1]
DS → Mag Mapping	Modified Exponential	Linear	Linear
	$M = 10 - 9e^{0.23(Th-DS)}$	$M = 0.94DS - 0.4$	—
Falling Rate	$\frac{\delta(Mag)}{\delta(t)} = 10 * Mag$	$\frac{\delta(Mag)}{\delta(t)} = 6.7$	—
Zoom-In Restriction	$\frac{\delta(Mag)}{\delta(t)} \leq 3 * Mag$	None	Same as falling rate
Zero Order Aiding	None	None	None
Initial Pointer Location	Centre	Centre	—
Pointer Freedom	Fixed	Free	—
Zoom Location	Centre	Centre	—
Speed Indication	None	Info Area	Info Area
Mouse Button	Middle	Any	—
Mouse Interaction	Modal and Drag	Drag	—

Table 4.2: The settings of the different SDAZ systems

[1] A, B and C are unspecified constants. B is the inverse of the unspecified maximum zoom level, and C represents the maximum displacement of the mouse.

Chapter 5

Formal Evaluation

5.1 Goals

There are three main goals of this evaluation:

- To compare SDAZ to rate-based scrolling.
- To model SDAZ using Fitts' law.
- To provide further insight into the accuracy of the previous calibration of the system.

The evaluation also has a three secondary aims:

- To explore whether people more naturally use the dragging or the modal version of the SDAZ and rate-based systems, and how this choice affects their performance.
- We are interested in seeing how previous experience with SDAZ and previous experience at using computers affects ability with the system.
- To investigate the idea that a large document might adversely affect performance with scroll-bars while making no difference to the rate-based systems.

The evaluation is based on the experimental paradigm of Hinckley et al. (2002). They proposed that Fitts' law be used to model different scrolling systems. Participants perform what is known as a Fitts' reciprocal tapping task, where they scroll back and forth several times between two targets. This is repeated several times with targets different distances apart and with the different scrolling systems. Each scrolling system can then be modelled using Fitts' law (see Section 2.3).

Our experimental method differs from that employed by Hinckley et al. in two minor ways. Hinckley et al. tested their scrolling systems over different levels of accuracy — for example, for some tasks the users had to position the target in the centre half of the screen, while getting the target on the visible screen was sufficient in other tasks. Thus they varied the level of accuracy they were requiring from the users. We saw no point in doing this and therefore simply asked the user to get the target on the screen in all our tasks. Our reasoning for this is threefold. One, if Fitts' Law does accurately model scrolling, as has been shown (Hinckley et al. (2002) and Guiard et al. (1999)), then there is little to be gained by testing different accuracies as the model accounts for the accuracy and can therefore be used to predict the results for different accuracies. Two, the purpose of scrolling is to get information into the visible screen space, once it is visible it is a fairly trivial task for the user to get it to the precise location on the screen that they happen to desire and such minor pagination is not the concern of long distance scrolling systems. Three, it would increase the time required from our participants. Our experiment took participants between 20 and 40 minutes and we did not want to greatly increase this by adding further conditions.

The second way in which our method differed from that of Hinckley et al. was our lack of line numbers. Hinckley et al. numbered every line of the document up the margin hoping that this would simulate user familiarity with the document. For example, a user familiar with a document would know in which direction and approximately how far their desired target was compared to their current position in the document. Users were therefore given the line numbers of their targets so they could use that as a check against the line numbers in the margin of their current location. There were two reasons we did not use such line numbering. One, an integral part of SDAZ is that it allows the user to visually identify the target as it passes. If the users simply focused at the line numbers on the side of the screen there would be absolutely no difference between SDAZ and rate-based scrolling. Since it is precisely a comparison of those two systems that we are attempting, such an experimental condition would be entirely unacceptable. Two, there is the problem of deciding what to do with the line numbers in the SDAZ system — do the line numbers zoom out too, or do they bunch together so none can be clearly read, or are only some of them shown? We decided that the best solution was to use no line numbers. We instead made sure that the documents both had normal page numbers so that the users could use those if desired (none did) and the chapter and section headings of the larger document were numbered (which several participants did use for reference).

5.2 Participants and Apparatus

Fifteen volunteer university students (eleven male, four female) took part in the experiment. Eight regularly played computer games involving rapid information flow. Nine were full time computer science students. Five had used SDAZ before, and nine had used rate-based scrolling before.

The evaluation was done using the SDAZ system discussed at length in Chapter 4, which was modified to handle rate-based scrolling and a standard scrollbar. This system provided frame rates of approximately 150 frames per second during the experiment and was running on a 2.4 GHz Intel Pentium 4 computer with 512MB RAM and a Geforce 4 MX video card, running Redhat Linux 8. The 19-inch display was set to 1280x1024 pixels and the interface ran in full-screen mode.

5.3 User Tasks

The experiment was a 3x6 repeated measures design. The dependent measure was time taken to tap between the targets. The two factors were as follows:

- *Scrolling system*

  The levels of this factor were: rate-based, SDAZ, and scrollbar.

- *Target set*

  There were six pairs of targets that participants scrolled back and forth between eight times. That is, each target in each pair was acquired four times by the user.

Participants were given a print out of the pages of the document that their current target pair were on, and the targets themselves were highlighted on the hard copy. Participants had as much time as they desired between each target set to familiarise themselves with the document and with the location of their targets. Participants were then asked to scroll back and forth between the two targets as quickly as possible, pressing the space bar when they had reached each target (when the highlighted portion of the page was on screen). The data from their first acquisition of each target was discarded as the user was still learning the location of the target.

The target order remained the same for every participant and every interface. The targets were spread across two documents, the first a 47 page honours thesis and the second an 8 page conference paper. Information about the target sets and documents is displayed in Table 5.1. Participants performed all the tasks in one interface, then filled out a NASA Task Load Index subjective evaluation sheet (Hart & Staveland 1988), then repeated the process with the two remaining interfaces. Participants were order-balanced in their exposure to the three scrolling systems, using a Latin square, to avoid order effects. After performing all the experimental tasks, users were asked to rank the three systems from best to worst.

Target Set #	Distance	Doc #
1	35	1
2	20	1
3	5	1
4	5	2
5	3	2
6	1.5	2

Doc #	Length	Reference
1	47	Moyle (2001)
2	8	Hinckley et al. (2002)

Table 5.1: The target sets and documents used in the experiments. The target "Distance" and and document "Length" are reported in units of pages.

Chapter 6

Results and Discussion

6.1 Fitts' Law Models of the Systems

Figure 6.1 shows the mean times for each scrolling system plotted against the Indexes of Difficulty (IoD) tested, and the data itself is shown in Table 6.2.

According to the information theoretic basis of Fitts' Law, it is optimal to have a 4.13% error in the results (MacKenzie 1992). If the error level differs significantly from this, the IoDs can be adjusted statistically to account for it. Users in our evaluation made errors 3.95% of the time so we have not used any adjustments in our results.

Table 6.1 shows the equations of the lines of best fit for the three systems according to the Fitts' Law model (see Section 2.3 for a description of Fitts' Law). The high R^2 values are typical of Fitts' Law which is well known for R^2 values of 0.9 or above (MacKenzie 1992).

Notable is the way in which the fifth point drops off for all three curves. This suggests that the target set was easier than the rest for users to spot. That particular target set was the introduction

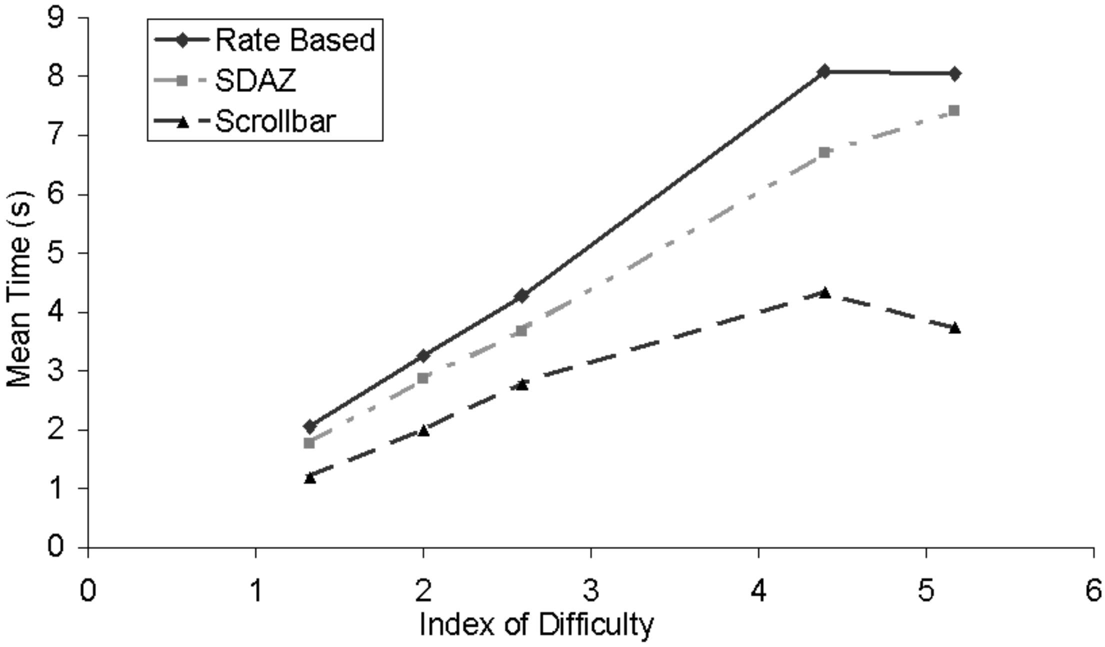


Figure 6.1: The performance of the three scrolling systems at different indexes of difficulty

System	Equation	Pearson's R^2
Scrollbar	$y = 0.71x + 0.67$	$R^2 = 0.843$
SDAZ	$y = 1.51x - 0.19$	$R^2 = 0.996$
RateBased	$y = 1.69x - 0.08$	$R^2 = 0.971$

Table 6.1: The equations of the lines of best fit

and conclusion headings of the document — both of which were followed by a small amount of text with a page or so of whitespace on either side, hence making these targets visually obvious.

The vast superiority of scrollbars over the other two systems raises questions. Were Cockburn & Savage wrong when they found SDAZ to be faster than scrollbars? We believe the answer lies in the nature of the task. By our classification given in Chapter 3, the task in a Fitts' Law reciprocal tapping task is a Reacquire action. As we mention in that discussion, position control devices such as scrollbars are known to be especially efficient at Reacquire actions. Our findings clearly support that thesis. Some of our participants commented that they were simply remembering the location on the scrollbar that the thumb needed to be located rather than actually looking for the target within the document. Thus, they were performing a Spatial-Locate task rather than a Visual Search.

Therefore, to some extent, the experimental paradigm of modelling a scrolling system using Fitts' law tapping tasks is not universally valid. More specifically, comparison between a rate-control system and a position control system is unfairly biased in favour of the position control system. We can still, of course, draw results between rate-based systems though and conclude that our results show SDAZ to be faster than a standard rate-based system for performing Reacquire tasks. Hence we see no conflict between our results and those of Cockburn & Savage. Cockburn & Savage used Visual Search tasks in their evaluation, which we believe to be the strength of SDAZ.

Scroll-Type	IoD	Mean Time (ms)	SD	SE
Rate-Based	1.32	2051	1019	263
	2.00	3250	1150	297
	2.58	3572	1131	292
	2.58	4978	1801	465
	4.39	8088	2473	638
	5.17	8052	2687	694
SDAZ	1.32	1767	624	161
	2.00	2861	1180	305
	2.58	3263	1254	324
	2.58	4094	2119	547
	4.39	6691	2754	711
	5.16	7414	2626	678
Scrollbar	1.32	1191	429	111
	2.00	1997	647	167
	2.58	2446	609	157
	2.58	3118	1255	324
	4.39	4339	1933	499
	5.17	3727	1273	329

Table 6.2: The data for scrolling system vs IoD

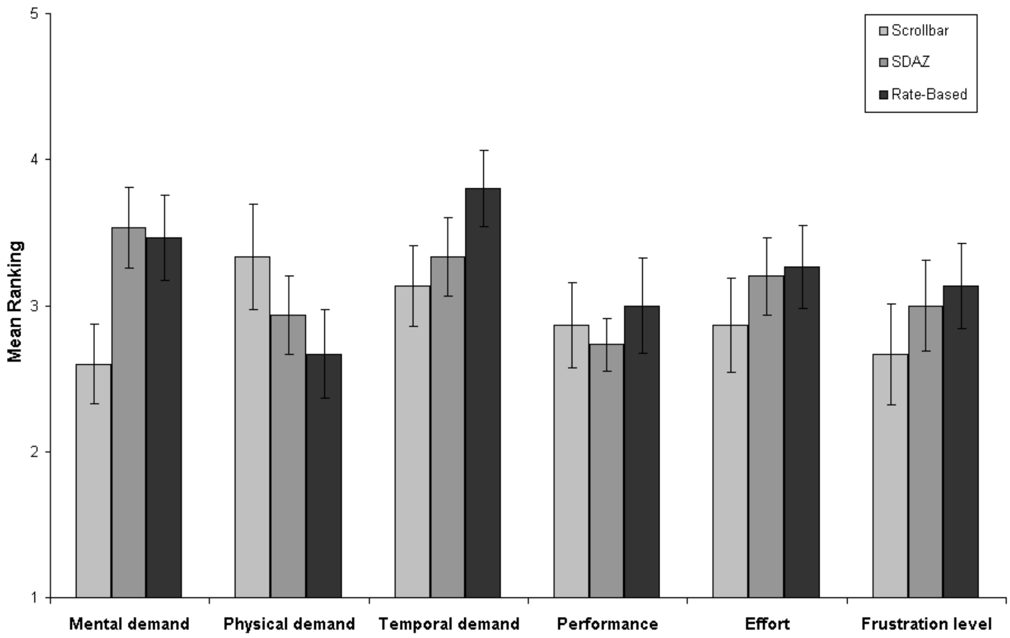


Figure 6.2: The NASA TLX subjective evaluation results.

6.2 SDAZ vs Rate-Based

As Figure 6.1 shows, SDAZ was faster than rate-based, on average, for every distance tested. An ANOVA run across the data of the three systems revealed a significant difference between the three scrolling systems ($F_{2,28} = 30.2, p < 0.001$). A paired t-test showed a statistical difference ($t_{538} = 17.8, p < 0.0001$) between SDAZ and rate-based. The overall means for the three systems are shown in Table 6.3. It can be seen that SDAZ was 13% faster than rate-based on average for these tasks.

From the participants comments, they clearly preferred SDAZ to rate-based scrolling. Three participants said something similar to "It [rate-based] was horrible". Some of the participants did not feel comfortable with any form of rate-based scrolling, as it was new to them.

Four participants made comments to the effect that SDAZ in general was better than rate-based scrolling because it gave them more time to see what was coming and made it easier to locate images because of this.

In the subjective ranking of the three systems, 67% of participants preferred SDAZ to rate-based scrolling. 60% of the participants ranked rate-based as the worst of all three systems, while only 13% ranked SDAZ as worst. (60% ranked scrollbars as best of all, but as has already been discussed, scrollbars were particularly useful in this evaluation).

The NASA Task Load Index subjective evaluation (see Figure 6.2) revealed no clear differences between rate-based and SDAZ systems. In the NASA TLX test participants rank each system from one (best) to five (worst) in six different categories.

System	Mean Time (ms)	SD	SE
Rate-Based	4998	1455	376
SDAZ	4348	1570	405
Scrollbar	2803	875	226

Table 6.3: The overall results for the systems

6.3 The Accuracy of the Earlier Calibration

One of our interests in this experiment was feedback about how our calibration of SDAZ seemed to perform in a real target acquisition task — rather than simply playing with the system as our earlier participants had done (see Chapter 4).

One participant wrote that the SDAZ system was "Too hard to stop without middle clicking" by which we understand him to mean that he found the null-speed region too small. This was something we had already identified as a potential problem — see Section 4.1.5. However the problem may just have been that this participant clicked the mouse rather than dragged with it.

One participant felt that SDAZ zoomed out the document too far and would have preferred a maximum magnification of about five pages. This surprised us as when we use the system ourselves we encounter the problem that our maximum magnification of ten pages seems too low. It may be that greater familiarity with the document means that a higher maximum magnification is appropriate. Alternatively, this could be an indication that our Document Speed → Magnification mapping is incorrect. The participant wanted to go faster on a lower magnification, while we want the system to reach higher magnifications. A linear mapping with a higher maximum magnification would seem to fit both of these requirements.

Two participants reported difficulties in changing the direction when they had overshot the target. They commented on the difficulty of changing direction and on the difficulty caused by the system zooming in as they reversed direction. We believed we had corrected the problem of the zoom-in on direction change (see Section 4.1.4) but our solution was apparently not sufficient for these users. The issue of a quick direction change we have already identified (see Section 4.1.5) but is not one we could find an adequate solution to.

One participant commented that SDAZ was annoying when scrolling really short distances (presumably because the zoom-out seemed counterintuitive when travelling such short distances). However two users made comments to the opposite effect — that they found SDAZ easier for short distance scrolling than they did over longer distances.

One participant complained that the speed control was oversensitive. This suggests that the slope of line mapping Displacement → DS was too steep (see Section 4.1.2 and Table 4.2). We had not so carefully investigated the calibration of this formula, as we believed the settings used by Cockburn & Savage were adequate.

6.4 The Use of Modal vs Dragging mouse interaction

To use the Rate-Based and SDAZ systems, participants were told to "use the middle mouse button". Both systems could be operated by either holding down the middle mouse button and dragging the mouse and releasing it to end scrolling, or by clicking the middle mouse button and then moving the mouse and clicking again to end scrolling. We decided to investigate this because the Windows implementation of rate-based scrolling provides for both functionalities we were unsure during our calibration of SDAZ which functionality was better.

Ten participants dragged with the mouse, and five participants used the modal operation. Only one of the fifteen users discovered both methods of using the mouse — they then chose to drag. There was no significant difference ($F_{1,13} = 0.01, p = 0.91$) between the performance of those who dragged ($mean = 4.78s$) and those who use the modal interface ($mean = 4.88s$). Neither was there a significant interaction ($F_{4,52} = 0.25, p = 0.91$) between the Index of Difficulty of the targets and the method of mouse interaction being used. There was also no significant difference ($F_{1,13} = 1.14, p = 0.31$) between the scrolling system used (SDAZ or rate-based) and the method of mouse interaction used (dragging or modal). This last however, is moderately close to being significant and its lack of statistical significance may be due to our small sample size. Our data shows SDAZ performs better when dragging is used (a mean of 4.34 rather than 4.77s) and rate-based performs better when the mouse is used modally (a mean of 4.98 rather than 5.22s).

Four of the fifteen users suffered "modal confusion" at some point where they attempted to use the mouse in one way and accidentally used it the other way. For example, a user that operated

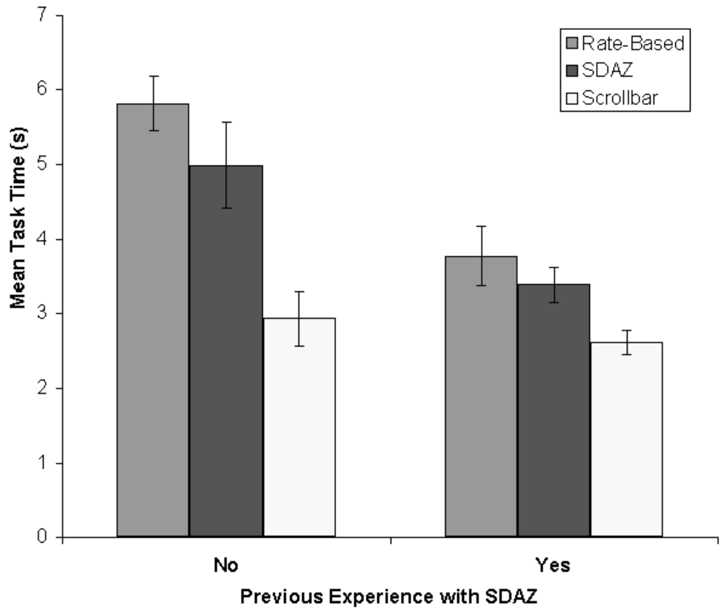


Figure 6.3: The effect of previous experience with SDAZ

the system by dragging, would click the button accidentally (or the system would interpret a very short dragging motion as a click) and then get stuck in scrolling mode until they clicked again. Users appeared to attribute this functionality to a bug rather than realising the system supported both types of interaction.

The scrollbar, like typical scrollbars, provided three methods of interaction. The user could click and drag the scroll thumb, click the arrows to scroll one line, or click outside the scroll thumb to move one page. Four participants actively discovered all three uses of the scrollbar (not including the one participant who discovered both methods of using rate-based and SDAZ scrolling) — though we assume that many of the others were aware of the alternate ways to use a scrollbar from their previous experience of them. All but one participants operated the scrollbar in the experiment purely by dragging the scroll thumb.

6.5 The Effects of Previous Experience

Six of the fifteen participants had previous experience with SDAZ. Unsurprisingly these people were significantly faster ($F_{1,13} = 6.4, p < 0.03$) at all types of scrolling ($mean = 3.26s$) than the inexperienced people ($mean = 4.58s$). There was also a significant interaction ($F_{2,26} = 6.2, p < 0.01$) between previous experience with SDAZ and the scrolling system used. Figure 6.3 shows a plot of this and Table 6.4 gives the data.

Eight of the fifteen participants said they regularly played computer games which involved rapid information flow. There was almost a significant main effect for game playing ($F_{1,13} = 4.3, p = 0.06$) and no significant interaction with the scrolling system ($F_{2,26} = 2.6, p = 0.10$).

Nine of the participants were computer-science majors. There was a significant difference ($F_{1,13} = 5.5, p < 0.04$) between those majoring in computer science ($mean = 3.54s$) and those not ($mean = 4.80s$). There was almost a significant interaction between computer science major and scrolling system ($F_{2,26} = 3.3, p = 0.052$). The difference at using scrollbars between those majoring in computer science and those not was quite small (means of 2.6 and 3.1), whereas the difference at using SDAZ between those majoring in computer science and those not was comparatively large (means of 3.6 and 5.5). This is very similar to the data obtained for condition of the previous

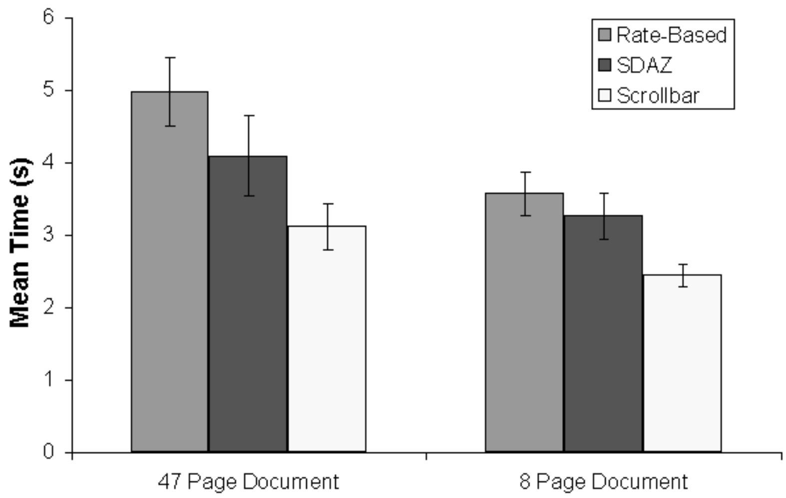


Figure 6.4: The performance of the three scrolling systems on the two different documents when scrolling five pages

experience with SDAZ. Together they suggest that experience benefits users of SDAZ more than it does users of scrollbars. This means that scrollbars are easier to learn to use effectively than SDAZ. This is unsurprising since scrollbars are relatively simple compared to SDAZ.

6.6 The IoD tested on both documents

Users scrolled a distance of five pages on both the larger and smaller documents. It was hoped that this would give some indication as to how a larger size document affected the different scrolling systems.

We thought that the scrollbar might perform worse on the large document than the smaller one as the larger document would require the scroll thumb to be positioned more accurately. Figure 6.4 shows a plot of the data for this IoD. This shows there is no such interaction occurring. The larger document was slightly slower in all systems, but suggests it is likely the case that the target pair used in the larger document was more difficult to acquire than the shorter document target pair was. We have already seen (with the example of the target pair with an IoD of 5.1) that target context can have a large effect on the acquisition time.

Perhaps the reason that the performance of the scrollbar didn't degenerate on our large document was our high screen resolution. Because of this the use would have had a margin of error of about 20 pixels in which to position the scroll thumb.

SDAZ Experience	Interface	Mean Time (ms)	SD	SE
	rate-based	3771	971	396
Yes	SDAZ	3392	572	233
	scrollbar	2609	396	162
	rate-based	5817	1109	370
No	SDAZ	4986	1723	574
	scrollbar	2933	1093	364

Table 6.4: Previous experience with SDAZ

Chapter 7

Conclusion

Speed-Dependent Automatic Zooming (SDAZ) is a new and promising method of scrolling that has already been shown to be superior to scrollbars in performing some tasks. We have analysed the different forms of document navigation and identified the place of SDAZ within the existing framework of navigation techniques, showing it to be complementary to the more traditional methods, but a replacement for rate-based scrolling.

The implementation of SDAZ requires making a number of design decisions in the interface and the selection of control formulae. The extent of the difficulty of this is not made clear in the previous literature, nor have the decisions made in the previous implementations been explained fully. The main part of our work here has been to explore this problem of calibration. We offer a complete survey of the issues we discovered in the calibration of an SDAZ system, and through an iterative development of the system through informal user feedback we have attempted to obtain an optimal SDAZ implementation and have reported the details here.

Finally we conducted a small experiment with the primary intention of comparing SDAZ to rate-based scrolling. We found that SDAZ was 13% faster that rate-based scrolling and preferred by most participants. This validates the hypothesis that the addition of zooming to rate-based scrolling is beneficial.

Future Work

Many areas for future work were discussed extensively in Chapter 4. Those areas in particular that we would like to see future work done will be mentioned briefly here.

After performing our evaluation we do not feel that our control equations are optimal. We now believe (in agreement with Igarashi & Hinckley) that an exponential based mapping between mouse displacement a scroll speed, with a linear mapping between scroll speed and magnification would be optimal. For the sake of simplicity we arbitrarily excluded non-linear mappings between displacement and scroll speed from our consideration early in our investigation and we now believe it would be worthwhile to examine them.

The problem of stopping and reversing the scrolling direction occurred repeatedly. We have suggested that it would be useful if users could just jerk the mouse in the opposite direction they are currently travelling, the system would respond by bringing them instantly to a complete halt. (See Section 4.1.5 for a detailed discussion of this problem.)

Our evaluation suggested that we needed to reconsider the maximum zoom level. We had arbitrarily decided on a maximum magnification of 10 pages, which we personally found to be too low, but other users found to be too high. Perhaps improved control equations with a higher maximum magnification would solve this problem.

The decision also needs to be made as to whether users can be presented with a “one size fits all” system, or whether it is best to offer users the ability to customise the system. In the latter case, it would be necessary to decide upon what the customisable options should be.

Bibliography

Balakrishnan, R., Baudel, T., Kurtenbach, G. & Fitzmaurice, G. (1997), 'The rockin'mouse: Integral 3d manipulation on a plane', *CHI97* .

Blohm, G. & Schreiber, C. (2002), 'The smooth pursuit system', `http://www.auto.ucl.ac.be/EYELAB/neurophysio/perception_action/SP.html`.

Burr, D. (1980), 'Motion smear', *Nature* **284**, 164–165.

Byrne, M., John, B., Wehrle, N. & Crow, D. (1999), 'The tangled web we wove: A taskonomy of www use', *Proceedings of CHI'99 Conference on Human Factors in Computing Systems* Pittsburgh, May 15–20 pp. 544–551.

Card, S., English, W. & Burr, B. (1978), 'Evaluation of mouse, rate-controlled isometric joystick,step keys, and text keys for text selection on a crt', *Ergonomics* **21**, 601–613.

Card, S. K., Moran, T. P. & Newell, A. (1987), *The Psychology of Human-Computer Interaction*, Morgan Kaufmann Publishers Inc, chapter 2, pp. 23–97.

Cockburn, A. & Savage, J. (2003), 'Comparing speed-dependent automatic zooming with traditional scroll, pan and zoom methods', *People and Computers XVII: British Computer Society Conference on Human Computer Interaction. Bath, England.* pp. 87–102.

Eckert, M. & Buchsbaum, G. (1993), The significance of eye movements and image acceleration for coding television image sequences, *in* A. Watson, ed., 'Digital Images and Human Vision', M.I.T Press, chapter 8, pp. 90–98.

Epps, B. W. (1986), Comparison of six cursor control devices based on fitts' law models, *in* 'Proceedings of the Human Factors Society 30th Annual Meeting', pp. 317–331.

Fitts, P. (1954), The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement, *in* 'Journal of Experimental Psychology', Vol. 47, pp. 381–391.

Gillan, D. J., Holden, K., Adam, S., Rudisill, M. & Magee, L. (1990), How does fitts' law fit pointing and dragging?, *in* 'Proceedings of the CHI '90 Conference on Human Factors in Computing Systems', New York: ACM, pp. 227–234.

Guiard, Y., Beaudouin-Lafon, M. & Mottet, D. (1999), Navigation as Multiscale Pointing: Extending Fitts' Model to Very High Precision Tasks, *in* 'Proceedings of CHI'99 Conference on Human Factors in Computing Systems Pittsburgh, May 15–20', pp. 450–457.

Gutwin, C. (2002), Improving Focus Targeting in Interactive Fisheye Views, *in* 'Proceedings of CHI'2002 Conference on Human Factors in Computing Systems Minneapolis, Minnesota, 20–25 April', pp. 267–274.

Gutwin, C. & Skopik, A. (2003), Fisheye Views are Good for Large Steering Tasks, *in* 'Proceedings of CHI'2003 Conference on Human Factors in Computing Systems Fort Lauderdale, Florida'.

Hart, S. & Staveland, L. (1988), Development of nasa-tlx (task load index): Results of empirical and theoretical research, *in* P. Hancock & N. Meshkati, eds, 'Human Mental Workload', Elsevier, pp. 139–183.

Hinckley, K., Cutrell, E., Bathiche, S. & Muss, T. (2002), Quantitative Analysis of Scrolling Techniques, *in* 'Proceedings of CHI'2002 Conference on Human Factors in Computing Systems Minneapolis, Minnesota, 20–25 April', pp. 65–72.

Igarashi, T. & Hinckley, K. (2000), Speed-dependent Automatic Zooming for Browsing Large Documents, *in* 'Proceedings of the 2000 ACM Conference on User Interface Software and Technology, San Diego, California.', ACM Press, pp. 139–148.

Jagacinski, R. J. & Monk, D. L. (1988), 'Fitts' law in two dimensions with hand and head movements', *Journal of Motor Behaviour* **17**, 77–95.

Jones, S., Jones, M., Marsden, G., Patel, D. & Cockburn, A. (n.d.), An evaluation of integrated zooming and scrolling on small-screen devices. This paper was unpublished at the time of writing.

Kelly, D. (1979), 'Motion and vision. ii. stabilized spatio-temporal threshold surface', *Journal of the Optical Society of America* **69**(10), 1340–1349.

MacKenzie, I. S. (1991), Fitts' law as a performance model in human-computer interaction, PhD thesis, University of Toronto.

MacKenzie, I. S. (1992), 'Fitts' law as a research and design tool in human-computter interaction', *Human Computer Interaction* **7**, 91–139.

MacKenzie, I., Sellen, A. & Buxton, W. (1991), A Comparison of Input Devices in Elemental Pointing and Dragging Tasks, *in* 'Proceedings of CHI'91 Conference on Human Factors in Computing Systems New Orleans, May', pp. 161–166.

Morgan, M. J. & Benton, S. (1989), 'Motion-deblurring in human vision', *Nature* **340**, 385–386.

Moyle, M. (2001), 'A flick in the right direction', Honours Thesis, University of Canterbury, `http://www.cosc.canterbury.ac.nz/research/reports/HonsReps/2001/hons_0107.pdf`.

Poulton, E. C. (1974), *Tracking Skill and Manual Control*, Academic Press, chapter 16, pp. 323–360.

Rashbass, C. (1959), 'Barbiturate nystagmus and the mechanisms of visual fixation', *Nature* **183**, 897–898.

Savage, J. (2002), 'Speed-dependent automatic zooming', Honours Thesis, University of Canterbury, `http://www.cosc.canterbury.ac.nz/research/reports/HonsReps/2002/hons_0208.pdf`.

Wallace, A., Savage, J. & Cockburn, A. (2004), Rapid visual flow: How fast is too fast?, *in* A. Cockburn, ed., 'Conferences in Research and Practice in Information Technology', Vol. 18, The 5th Australasian User Interface Conference (AUIC2004), Dunedin.

Zhai, S., Smith, B. & Selker, T. (1997), Improving Browsing Performance: A Study of Four Input Devices for Scrolling and Pointing Tasks, *in* 'Proceedings of INTERACT'97: the sixth IFIP conference on Human Computer Interaction', pp. 286–292. `http://www.almaden.ibm.com/u/zhai/papers/multistream/inter97.pdf`.