# The Calibration and Evaluation of Speed-Dependent Automatic Zooming Interfaces

A thesis
submitted in partial fulfilment
of the requirements for the Degree
of
Master of Science in Computer Science
in the
University of Canterbury
by
Josh Savage

University of Canterbury
2004

**Examining Committee**

| | |
|---|---|
| Supervisor | Associate Professor Andy Cockburn |
| External Examiner | Dr Steve Jones |

To my Parents, my little Brother, Annie, Aunty Suzette and Uncle Tommy for all their love and support.

# Abstract

Speed-Dependent Automatic Zooming (SDAZ) is an exciting new navigation technique that couples the user's rate of motion through an information space with the zoom level. The faster a user scrolls in the document, the 'higher' they fly above the work surface.

At present, there are few guidelines for the calibration of SDAZ. Previous work by Igarashi & Hinckley (2000) and Cockburn & Savage (2003) fails to give values for predefined constants governing their automatic zooming behaviour. The absence of formal guidelines means that SDAZ implementers are forced to adjust the properties of the automatic zooming by trial and error.

This thesis aids calibration by identifying the low-level components of SDAZ. Base calibration settings for these components are then established using a formal evaluation recording participants' comfortable scrolling rates at different magnification levels.

To ease our experiments with SDAZ calibration, we implemented a new system that provides a comprehensive graphical user interface for customising SDAZ behaviour. The system was designed to simplify future extensions — for example new components such as interaction techniques and methods to render information can easily be added with little modification to existing code. This system was used to configure three SDAZ interfaces: a text document browser, a flat map browser and a multi-scale globe browser.

The three calibrated SDAZ interfaces were evaluated against three equivalent interfaces with rate-based scrolling and manual zooming. The evaluation showed that SDAZ is 10% faster for acquiring targets in a map than rate-based scrolling with manual zooming, and SDAZ is 4% faster for acquiring targets in a text document. Participants also preferred using automatic zooming over manual zooming. No difference was found for the globe browser for acquisition time or preference. However, in all interfaces participants commented that automatic zooming was less physically and mentally draining than manual zooming.

# Table of Contents

# List of Figures

# List of Tables

# Chapter I

# Introduction

Today's computer users must frequently navigate in information spaces that are too large for the viewing window. Scrolling, panning and zooming are the typical techniques used to overcome the screen 'real-estate' problem in documents such as text files, maps, spreadsheets and pictures. Document scrolling is commonly provided using vertical and horizontal scrollbars that are manipulated by moving a scroll handle. This handle controls and indicates the placement of the viewport in the document. Another common mechanism is panning, which allows the user to drag the space within the viewport. This directly manipulates the location in any direction. Zooming increases and decreases the size of the information inside the viewport and can be used to gain context in a document or to focus in on important parts.

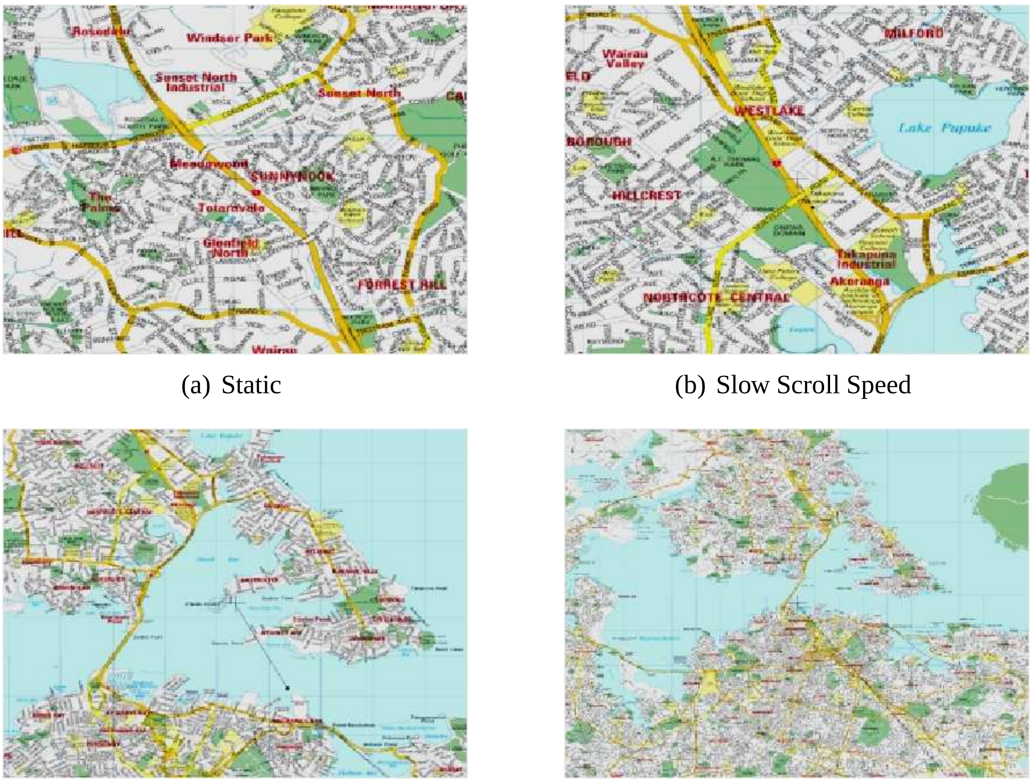
There are numerous limitations to scrolling, panning and zooming. One limitation identified by Igarashi & Hinckley (2000) is the additional overhead involved with the user's attention being drawn back and forth between the scroll-bars and the document. To scroll the user must first acquire the scroll handle, which is sometimes very small and difficult to target, by moving the mouse cursor to the handle and clicking the mouse button. The user must then concentrate on the effect the operation has on the document. Scroll handle manipulation becomes more difficult in larger documents because a small movement can cause a large jump in the information space, which can cause disorientation and confusion. Manipulating the zoom level adds another level of difficulty to navigation. Typically, users need to move slowly when zoomed in to avoid disorientation and more quickly when zoomed out to avoid frustration.

The fundamental problem with panning is that constant mouse movement is required to scroll. One solution is a rate-based scrolling interface (Zhai et al. 1993). Rate-based scrolling maps the displacement of the cursor to the scrolling velocity. This technique is used in most interfaces in Microsoft Windows by holding the middle mouse button (or scroll wheel) and moving in the desired scrolling direction. The displacement between the initial clicked point and the current mouse pointer determines the speed of movement in the document. Another type of input device commonly used in rate-based scrolling is an isometric joystick which map the force exerted to velocity in the document. The IBM ScrollPoint II™ is a mouse with a small joystick in place of the scroll wheel that is used for rate-based scrolling. Rate-based scrolling, however, has a few drawbacks from scroll bars: users can not quickly leap to a different position in the document and users are limited by a maximum scrolling speed. Configuring the maximum scroll rate is difficult in rate-based scrolling. If it is set too fast the user will become disorientated by the extreme visual flow, but if it is set too slow the user will become frustrated at the time it takes to scroll to a distant known location. Frustration will occur when a user knows the location (s)he wants to go to but is forced to wait as the document slowly scrolls there.

The lack of smoothness between magnification levels is a major problem with many zoomable interfaces. 'Jumpy' transitions can cause users to lose context when zooming in and out of a document. Smooth transitions are often not implemented to improve the performance of the browser. However, performance has now become less of an issue with the current powerful CPUs and video cards, and the focus has turned more to usability.

Speed-Dependent Automatic Zooming (SDAZ) was first proposed in an attempt to solve the common problems of scrolling, panning and zooming (Igarashi & Hinckley 2000). SDAZ couples automatic zooming with rate-based scrolling. The system attempts to keep the information flow across the screen constant by automatically zooming out from the document as scroll speed increases. When stationary or scrolling slowly the document is zoomed in (see Figure 1.1(a)), and as the user scrolls more quickly the document is smoothly zoomed out to give an overview (see Figure 1.1(d)).

An informal preliminary study (n=7) of the technique by Igarashi & Hinckley found that in web and map browsing tasks, the efficiency with SDAZ was, on average, the same or slightly worse than traditional scrolling methods. Subjective preferences were also divided. However, recent studies (n=12) of SDAZ in similar document and map browsers preformed by Cockburn & Savage (2003) showed that SDAZ compared to standard scrolling was on average 22% faster for




(a) Static (b) Slow Scroll Speed

(c) Fast Scroll Speed (d) Fastest Scroll Speed

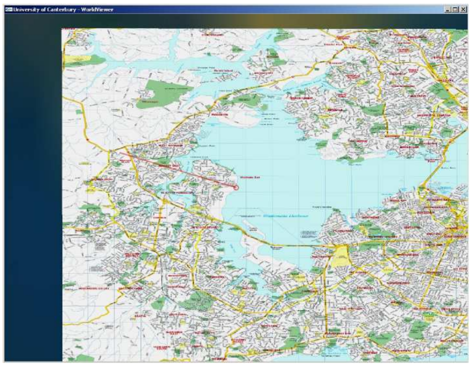
Figure 1.1: Screen-shots from Cockburn & Savage's Map Browser

document browsing tasks and 43% faster for map browsing tasks. All participants except one preferred the SDAZ interfaces. As smoothness is essential for SDAZ implementations Cockburn & Savage suggest that the contrast in results was primarily due to their smooth openGL implementations compared to Igarashi and Hinckley's 'jumpier' Java implementations. Other factors that may have lead to the difference in results are discussed in Section 2.4.2.

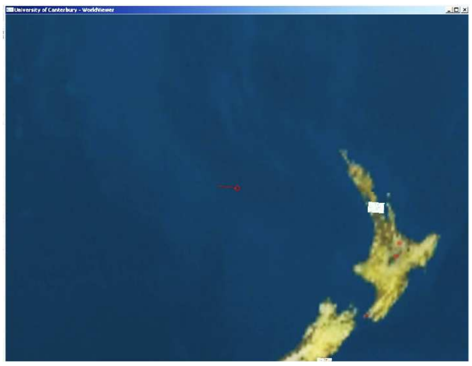
Cockburn et al. (2003) recently implemented a 3D 'globe browser' prototype using SDAZ navigation. The globe browser allows smooth and rapid navigation between detailed close-ups of city maps and a global overview of the Earth. In the globe browser, images of the Earth's surface are stitched together to form a globe, onto which map details of several cities/regions are overlaid (see Figure 1.2). Dragging the mouse in any direction causes the globe to rotate so that the region under the cursor moves towards the centre of the display. Displacing the cursor further from the mouse's initial clicked position increases the rate of rotation, and consequently increases the zoom level. When the user stops over a region with no underlying detailed map, the zoom level will not fall below the global overview's minimum altitude (see Figure 1.2(c)). However, if the user stops or moves slowly over a city map, then the zoom level will increase to a street-name level of detail (see Figure 1.2(a)). When moving over a detailed map or in the global overview, typical SDAZ behaviour is present — faster movement results in high zoom levels (see Figure 1.2(b)). When the user moves off the edge of the map, they rapidly 'fly' up to the world view (see Figure 1.2(c)). Full-speed motion over the globe zooms out to a level where a full global hemisphere is visible (see Figure 1.2(d)). The efficiency or usability of the globe browser has not been evaluated.

The need for efficient navigation is also vital for small screen displays, such as PDAs and cellphones, because of their inadequate amount of screen space. Jones et al. (2004) implemented two SDAZ interfaces for small screen displays: a document browser and a map browser. These interfaces were evaluated against interfaces using standard navigation techniques (scroll-bars, pan and manual zoom). They found that SDAZ was, on average, 29% slower for browsing text documents on small screen displays. No significant difference was found for mean task completion times when map browsing. Participants using SDAZ, however, were more accurate and performed significantly fewer actions when completing the tasks. Patel et al. (2004) also conducted research into the efficiency of SDAZ on small screen displays. The focus of their research was the effective browsing of photograph collections. They evaluated SDAZ against discrete zoom and gesture zoom interfaces. In the discrete zoom interface thumbnails of photographs are presented ordered by creation time. Users can click/tap the desired photo to view an enlarged version, and click/tap again to return to the thumbnail view. This interface is similar to commercial browsers Apple iPhoto (*Apple iPhoto* 2002) and ACDSee Picture Viewer (*ACDSee* 2001). In the gesture interface, scroll speed is proportional to vertical mouse displacement (as in rate-based scrolling interfaces) and zooming is proportional to horizontal mouse displacement (see Figure 1.3). A larger horizontal displacement results in a higher zoom level. They found that the SDAZ and gesture zoom interfaces support faster navigation, higher accuracy and

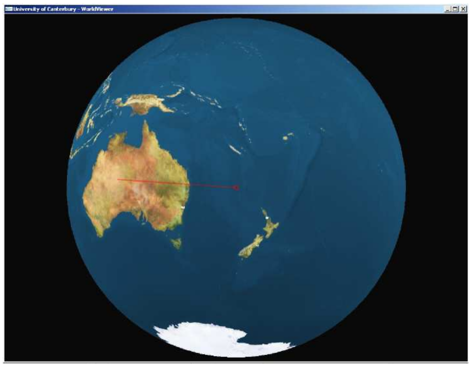

(a) Slow scrolling in city view.



(b) Fast scrolling in the city view.



(c) Slow scrolling in the globe view.



(d) Fast scrolling in the globe view.

Figure 1.2: The range of zoom-levels in Cockburn et al.'s globe browser. Screenshots taken as the user effortlessly navigates from a detailed street map of Auckland to the globe overview.

have lower subjective task load levels than the standard discrete zoom interface. Further work with SDAZ on small screens is discussed in Section 2.4.3.

Visual flow is the quantity of information moving across the screen at any point in time. The investigation into visual flow rates is a fundamental part of building SDAZ interfaces because its efficiency relies on automatically zooming out to reduce visual flow to allow faster scroll speeds. There are, however, many questions that are unanswered regarding visual flow in SDAZ, such as "at what scroll speed/visual flow rate should the system begin zooming?" and "what is the relationship between zoom level and scroll speed?"

At present, there are few guidelines for the implementation and calibration


(a) Moderate scroll speed and small image reduction.

(b) Maximum speed and minimum size..

Figure 1.3: Patel et al.'s gesture zoom interface.

of SDAZ interfaces. Previous work by Igarashi & Hinckley (2000) and Cockburn & Savage (2003) fails to give values for their predefined constants governing their automatic zooming behaviour. The absence of formal guidelines means that SDAZ implementers are forced to adjust the properties of the automatic zooming by trial and error.

This thesis focuses on two areas: SDAZ calibration and its performance against rate-based scrolling and manual zooming interfaces. The issues involved with the calibration of SDAZ interfaces are discussed in Chapter 3. This is followed by a psychological evaluation of maximum visual flow rates for the calibration of SDAZ automatic zooming (see Chapter 4). Using the results from this evaluation and our discussion of calibration, we designed and implemented new SDAZ interfaces allowing straightforward calibration and testing (see Chapter 5). These interfaces are then evaluated to compare efficiency and user preference with traditional rate-based scrolling and manual zooming interfaces (see Chapter 6). This evaluation also establishes physiological foundations for SDAZ in terms of Fitts'

Law (Fitts 1954) bandwidth measurements. This will assist in the comparison of future scrolling techniques by providing standardised comparable data. Future work and conclusions then follow.

# Chapter II

# Related Work

This chapter presents past work relating to SDAZ. Since the mouse is the primary interaction device used in our SDAZ systems, we first discuss the low-level details of its operation (see Section 2.1). We then discuss previous attempts to improve scrolling techniques, focusing on alternative scroll-bars and the development of rate-based scrolling (see Section 2.2). Zooming interfaces are then presented in Section 2.3. This leads to the development of SDAZ in Section 2.4. In this section, Igarashi & Hinckley's work is first presented followed by Cockburn & Savage's study and then work applying SDAZ to small-screen displays. Several 3D flying systems and an animation technique are presented in Section 2.5. This is followed by the discussion of several techniques proposed to aid in navigation analysis (see Section 2.6). Additional focus plus context techniques are discussed in Section 2.7. Finally, previous work on the processing of visual flow is presented in Section 2.8. This work aids in the theoretical calculations in our visual flow evaluation (see Chapter 4).

## *2.1 Control-Display Gain*

As the mouse is the primary interaction device used in our SDAZ systems, it is necessary to summarise the low-level details of the mouse operation and related work.

The control-display gain determines the mapping between the physical displacement of the pointing device (usually the mouse) and corresponding movement of the cursor on the screen (MacKenzie & Riddersma 1994). The mapping is typically controlled by one or two user-configurable parameters. These values are generally termed 'acceleration' and 'threshold'. When the mouse moves slowly, a base mapping between physical mouse motion and cursor movement is applied. The default value for base movement is normally around one to four, meaning that the cursor moves four centimetres for each centimetre of physical mouse motion. The cursor is accelerated, when the mouse is moved further than a threshold value within the mouse's sample rate, by multiplying the base movement by the acceleration value. Typically, the sample rate is 40Hz to 60Hz, however, some mice designed for games have higher sample rates (100Hz).

Control-display gain adaptation has previously been used to increase target acquisition performance. 'Sticky Icons' improves target acquisition by decreasing the mouse acceleration to one while the user is inside the icon. Worden et al. (1997) conducted an evaluation that found Sticky Icons to be efficient for selecting small targets. Cockburn & Firth (2003) also found Sticky Icons to be significantly faster for small target acquisition than normal selection.

Semantic pointing is another selection technique designed to increase target acquisition through control-display adaptation (Blanch et al. 2004). This technique adjusts the control-display gain according to cursor distance to nearby targets making the target feel larger to the user. An evaluation (n=12) conducted by Blanch et al. found no significant difference in task completion time and error rates compared to normal selection.

## *2.2 Scrolling Techniques*

### *2.2.1 Alternative Scroll-bars*

The AlphaSlider, first proposed by Osada et al. (1993), is an alternative scrolling technique for precise selection in large lists (see Figure 2.1(a)). The AlphaSlider is used like a horizontal scroll-bar with an index on the bottom and the text displayed in the top left. The main advantage of the AlphaSlider interface is that it is very efficient in its screen space use — it uses only one line of text output. One of its disadvantages is that it allows less use of context in searching. With a typical drop-down list-box interface, there are other words in the interface from which the subject can see where (s)he is in the list and how fast the alphabet is being moved through.

Ahlberg & Shneiderman (1994) conducted an evaluation investigating the efficiency of three different implementations of AlphaSliders against traditional scroll bars. They found no significant difference between the most efficient AlphaSlider implementation and traditional scroll bars for novice users.

Two designs that improve upon the AlphaSlider are the FineSlider (Masui et al. 1995) and the Stretch Button Scroll-bar (Smith & Henning 1996). The FineSlider is based upon the AlphaSlider but uses an elastic technique based on a rubber band metaphor (see Figure 2.1(b)). A control object such as a scroll bar is moved by pulling the object with a rubber band between the object and the mouse cursor. An evaluation where participants searched for a movie title from a title list with 10,000 entries showed that on average the FineSlider compared to the AlphaSlider is faster for novice users and only very slightly slower for expert users. The Stretch Button Scroll-bar uses a completely different approach (see Figure 2.1(c)). The scroll handle is modified with two extra buttons, one on either side. These buttons allow fine movement similar to that provided by the arrow movement buttons on traditional scroll bars. An evaluation designed to compare search performance of the Stretch Button Scroll-bar, the standard scroll-bars and the Alphaslider found that stretch button scroll bar was 23% and 16% faster than the Alphaslider and standard scroll bars, respectively.


(a) The AlphaSlider. (b) The FineSlider. (c) The Stretch Button Scroll-bar.

Figure 2.1: Alternative Scroll Bars.

One of the major drawbacks of scroll bars is that they do not offer adequate cognitive aids to the user to form a mental model of the material. Many overview plus detail visualisation techniques have been designed as a solution to this problem (see Section 2.7). Overview plus detail scroll bars have also been proposed. Two such solutions are the Bookmark Scroll Bar and the Calendar Scroll Bar (Laakso et al. 2000). The Bookmark Scroll Bar interface includes two scroll bars (see Figure 2.2). One looks and behaves as a traditional scroll-bars, the other displays bookmarked pages and when used snaps the user to the nearest displayed bookmark. New bookmarks can be added in the interface by clicking the new bookmark button. The goal of this interface is to “transfer the knowledge in the user’s memory to the knowledge in the world as effectively as possible” (Laakso et al. 2000). The Calendar Scroll Bar takes the idea of presenting an overview on the scroll bar further. The scroll area not only has an overview of the calendar but also includes visual cues of interesting days. Neither of the interfaces have been evaluated.

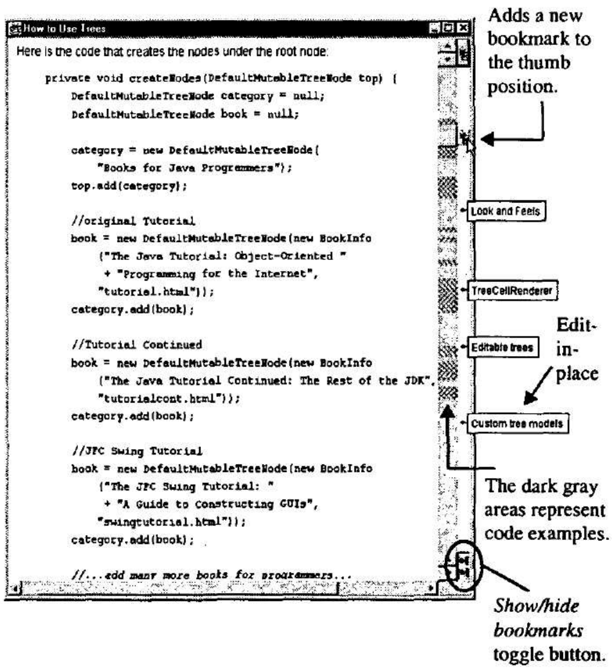



Figure 2.2: The Bookmark Scroll Bar.

### *2.2.2 Rate-based Scrolling*

Rate-based scrolling is ideal for isometric self-centring joysticks, such as the Joystick-Mouse (JMouse) and the In-keyboard Joystick, both shown in Figure 2.3. Several evaluations investigating the efficiency of these devices have been performed. Zhai et al. (1993) conducted an evaluation timing users searching a ten page web document for a specific hyperlink using four different scroll devices: a standard two button mouse, a scroll wheel mouse, a JMouse and the In-keyboard Joystick with the dominant hand on a standard mouse for hyperlink selection. Their evaluation found no significant difference between performance using the JMouse or using the In-keyboard Joystick with the dominant hand on the mouse. They did find, however, a significant difference between tasks completion times for the two isometric joystick devices and the two other mouses. Tasks were completed 22% and 31% faster with a JMouse compared to the wheel mouse and standard mouse, respectively. Using In-keyboard joystick and the mouse tasks were completed 26% and 35% faster compared to the wheel mouse and standard mouse, respectively.

(a) The JMouse. (b) In-keyboard Joystick.

Figure 2.3: Isometric Joysticks used for rate-based scrolling. Figures taken from Zhai et al.'s paper on improving browsing performance.

Hinckley et al. (2002) conducted an evaluation comparing the IBM Scroll-Point Mouse, a commercial device based on the JMouse against three different scroll wheel configurations. The scroll wheel configurations used were a standard configuration (3 lines per notch), and two configurations using different wheel acceleration algorithms. They found no significant difference between the standard scroll wheel configuration and the ScrollPoint. The scroll wheel configurations using acceleration algorithms were slightly faster. Their analysis indicated that short distance tasks (100 lines or less) were performed significantly slower with the ScrollPoint. However, the ScrollPoint's performance dramatically increased for long distance tasks, with ScrollPoint's time equalling the fastest wheel acceleration algorithm at the greatest distance. Users also indicated they felt it was faster scrolling long distances compared to the other techniques.

## 2.3 Zooming Interfaces

Studies have been carried out by Combs & Bederson (1999) to determine whether zooming improves image browsing. They found that zoomable user interfaces and 2D interfaces perform equally.

Pad (Perlin & Fox 1993) and Pad++ (Bederson & Hollan 1994) are zoomable user interface toolkits that provide continuous zooming as a central navigation tool. Objects are spatially organised in 2D space, and the user acquires an object by using scrolling and zooming operations. The drawback to this type of interface is that a user must control both scrolling and zooming interfaces. Speed-dependent Automatic Zooming (Igarashi & Hinckley 2000) attempts to integrate scrolling with zooming so that users need only be concerned with required information. Neither, Pad or Pad++ have been empirically evaluated.

## 2.4 Speed-Dependent Automatic Zooming

### 2.4.1 Igarashi & Hinckley's Work

The concept of speed-dependent automatic zooming (SDAZ) was first proposed by Igarashi & Hinckley. They implemented several SDAZ prototype systems in the Java™ programming language: a web browser, a map viewer, a sound editor, an image browser, and a dictionary viewer. Of these, the web browser and map viewer were successful in providing a potentially more efficient browser.

Igarashi & Hinckley initially proposed using Equations 2.1 and 2.2 to calculate the scrolling speed and document scale, respectively. They then revised their equations to achieve a more natural interaction. In the revised equations, shown in Equations 2.3 and 2.4, the current zoom level (or scale) of the document is calculated from the distance the user moves the mouse, and scroll speed is then calculated from the resulting zoom level. These new mappings conflict with the notion that zooming is dependent upon scroll speed in order to keep visual flow constant. Igarashi & Hinckley's paper failed to give predefined constants for their equations.

$$speed = C \times dy \tag{2.1}$$

Equation 2.1: Igarashi & Hinckley's initial equation to calculate speed, where *speed* is the vertical scrolling speed in the document, $C$ is a predefined constant modifying the scroll speed and $dy$ is the current distance the cursor has travelled.

$$scale = \frac{v0}{speed} \tag{2.2}$$

Equation 2.2: Igarashi & Hinckley's initial equation to calculate document scale, where $v0$ is the predefined speed reached before scaling starts, and *scale* is the magnification scale of the document.

$$speed = \frac{v0}{scale} \tag{2.3}$$

Equation 2.3: Revised mapping for calculating the scrolling speed in the document, where $v0$ is the predefined speed reached before scaling starts, and *scale* is the magnification scale of the document.

$$scale = s0^{(dy-d0)/(d1-d0)} \tag{2.4}$$

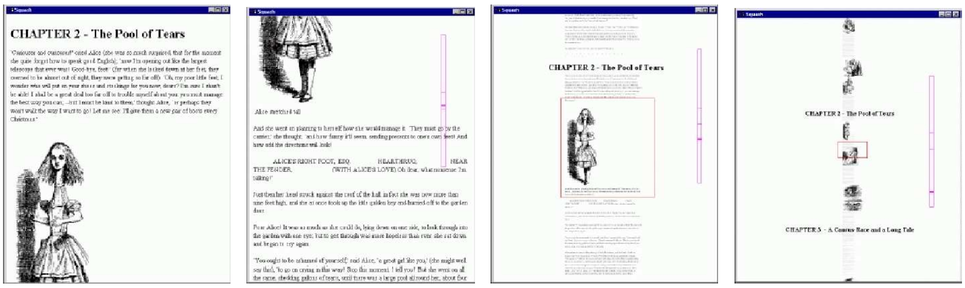
Equation 2.4: Revised mapping for calculating the scale of the document, where $dy$ is the current distance the cursor has travelled, $s0$ is the predefined minimum scale, $d0$ is the predefined mouse movement required before zooming starts, and $d1$ is the predefined maximum mouse movement.

Igarashi & Hinckley further modified their equations, during implementation, by introducing a delay in scaling. They found this was necessarily to prevent the document appearing to 'swell' suddenly when the user reversed the scrolling direction.

The following subsections describe Igarashi & Hinckley's SDAZ interfaces and the results from the evaluations conducted with the interfaces.

*Web Browser*

Igarashi & Hinckley's prototype web browser implemented in Java$^{TM}$ is shown in Figure 2.4. When the user presses the mouse button (presumably the left), a pink slider showing the document scroll speed appears on the right hand side of the screen. The scroll speed is increased by moving the mouse up or down while holding down the button. As the scroll speed increases, the document text becomes smaller, giving an overview of the document. When the user releases the mouse button, the text gradually returns to original size.




(a) Static view. (b) Scrolling slowly. (c) Scrolling fast. (d) Scrolling very fast.

Figure 2.4: Igarashi & Hinckley's Prototype Web Browser.

The web browser study timed seven subjects searching for images in a web document. They found that subjects using SDAZ and conventional scrolling performed approximately equally well, even though subjects had little experience with the new SDAZ interface. They also found that six out of seven subjects preferred automatic zooming, which suggests it might be a suitable alternative to traditional scrolling.

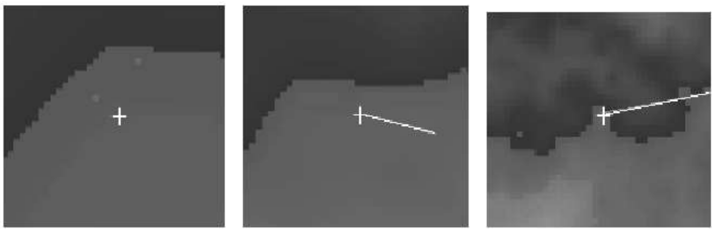
*Map Browser*

Igarashi & Hinckley's map browser (Figure 2.5) has two methods of interaction. One method is holding the mouse button down and dragging the mouse. The relative position between the point of the initial click of the mouse button and the current mouse position specifies the direction and speed of movement. The second method of interaction is using a joystick. The more the user tilts the stick, the faster (s)he scrolls, and the smaller the view gets. This method was used in user evaluations.

The map browser uses an artificially generated map based on Perlin's noise function (Perlin 1985). This increased performance of the browser with minimum implementation effort.

The evaluation consisted of users locating different targets in a 2D map using traditional scrolling/zooming and SDAZ. The task completion time results were mixed compared to the web browser. The authors suggest that this was because the map navigation task was more difficult than browsing a document, which caused users to employ a range of different strategies. The most efficient strategy is to zoom out until the target appears on the screen, move to the target, and then zoom in. However, some subjects slowly moved to the target without zooming out, which took a long time.

The results from the subjects' qualitative evaluations were also mixed, with four out of the seven subjects preferring the SDAZ interface.



(a) Static view. (b) Scrolling slowly. (c) Scrolling fast.

Figure 2.5: Igarashi & Hinckley's Prototype Map Browser.

*Unsuccessful Deployments of SDAZ*

Igarashi & Hinckley also experimented with three 'unsuccessful' domains for SDAZ: a sound editor, an image browser and a dictionary viewer. Their observations and the reasons for failure are informative for those considering SDAZ deployment.

Igarashi & Hinckley experimented with a sound editor with automatic zooming. They found the system was very difficult to use because of the lack of appropriate visual landmarks needed for automatic zooming targeting. The continuously transforming waveform was also confusing.

A prototype image browser was developed to browse a collection of images (Figure 2.6). The images were aligned in a horizontal list which user browsed by moving along. As the scrolling speed increased, the images became smaller so that more of them could be placed on the screen.

(a) Static view. (b) Scrolling slowly. (c) Scrolling fast. (d) Scrolling very fast.

Figure 2.6: Igarashi & Hinckley's Prototype Image Viewer.

Igarashi & Hinckley found that their automatic zooming image browser was less effective than a standard static array of thumbnails. This is because the image browser lacked the same abstraction given by the other browsers. For example, when browsing with the map browser, narrow streets fade away in the zoomed-out view, and highways and area labels become landmarks. In the web browser, the smaller body of text becomes blurry and only pictures and headings are clear for navigation. In the Image Browser, however, each image is unique and the order of the images is not important, so the same abstraction is missing.

In the dictionary viewer, the zooming-out effect is achieved by pruning out the less important items (2.7). Words are displayed in alphabetical order and are scrolled vertically by holding the middle button down and moving the mouse up and down. As the user scrolls faster, words are elided from the list.

Igarashi & Hinckley found the elision effect confusing to use. In addition, they found it required a significant cognitive overhead to locate a target word in the zoomed-out view because the user has to constantly think about the alphabetical order between the visible and target words. The example given by Igarashi & Hinckley shows that when searching for "bear", the user has to steer between "bavarian" and "befogging" in the zoomed-out view.


(a) Static view. (b) Scrolling slowly. (c) Scrolling fast.

Figure 2.7: Igarashi & Hinckley's Prototype Dictionary Viewer.

### *2.4.2 Cockburn & Savage's Work*

Cockburn & Savage (2003) implemented two SDAZ interfaces: one for viewing text documents the other for map documents (see Figures 2.8 and 1.1). The interfaces were used to conduct an ecologically oriented evaluation comparing SDAZ against traditional scroll, pan, and zoom techniques. Their evaluation results showed that tasks were completed, on average, 22% and 43% faster in the SDAZ Document and Map Browser, respectively, and that all participants but one preferred SDAZ. Cockburn & Savage give the following potential explanations for why their results differ from Igarashi & Hinckley:

- Implementation — Cockburn & Savage's systems were implemented in C and OpenGL exploiting high frame-rates and fluid animation available through hardware acceleration. Igarashi & Hinckley's systems were implemented in Java most likely having slower frame-rates. In their document interface, discrete font sizes were used to simulate dynamic font scaling and to improve performance in zoomed-out views, text was depicted as horizontal lines rather than miniaturised fonts. In their map browsing evaluation an artificially generated map was used rather than a real one.

- Experimental Objectives — Igarashi & Hinckley's primary objective was to describe their fascinating new interaction technique. Their experiment was intended only to provide initial impressions of the technique's effectiveness. Cockburn & Savage suggest that with different experimental objectives, and with greater statistical power from wider participation, they may have been better able to discriminate between performance with the interfaces.

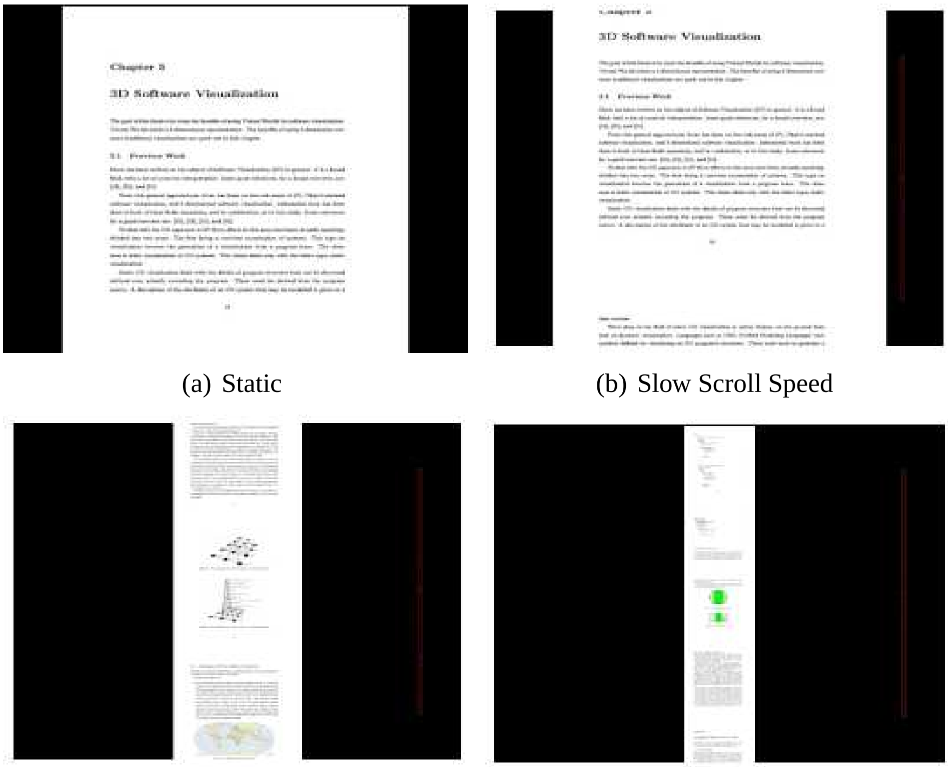
- Competing Interfaces — Cockburn & Savage's SDAZ interfaces were compared against commercial implementations of traditional systems (Adobe Acrobat Reader and Paint Shop Pro), and all interfaces were controlled using a mouse. Igarashi & Hinckley's SDAZ document browser was compared against a mouse-driven unspecified web browser. They did not specify whether their competing web browser supported zooming; their screen shots suggest it did not. Cockburn & Savage believe it is unlikely that Acrobat Reader's zooming facilities cause slower performance in their evaluation, accounting for the comparative efficiency of SDAZ.

  In Igarashi & Hinckley's map navigation tasks both interfaces were controlled using a joystick, allowing users to pan and zoom concurrently when using traditional interfaces. Cockburn & Savage's mouse control, however, required scrolling and zooming to be carried out either in series (using discrete zoom adjustments, then scrolling) or in a combined action (using Paint Shop Pro's centring zoom).

- Task Types — Igarashi & Hinckley's map browsing tasks involved locating a highlighted dot in an artificial terrain. The dot was continually visible in a small 'global radar' in the corner of the display. Their tasks test abstract target acquisition modeled by Fitts' Law. Cockburn & Savage's tasks were ecologically orientated involving extraction of meaningful information from different levels of magnification.

- Participants — In Igarashi & Hinckley's evaluation participants were all "good" or "average" computer users, with four of the seven reporting that they played computer games "sometimes", "frequently" or "almost every day". All twelve of Cockburn & Savage's participants were expert computer users (graduate level Computer Science students), and all but three regularly played computer games.

Cockburn & Savage used different equations from Igarashi & Hinckley to control the scroll speed and zoom level in their interfaces. Equations 2.5 and 2.6 show Igarashi & Hinckley formulae calculating the scroll speed in the document and map browsers. These equations, unlike Cockburn & Savage's, use cursor displacement directly to control the speed. The zoom level is then calculated from




(a) Static

(b) Slow Scroll Speed

(c) Fast Scroll Speed

(d) Fastest Scroll Speed

Figure 2.8: Screen Shots from Cockburn & Savage's Text Document Browser.

the resulting speed using Equation 2.7. Cockburn & Savage do not state their predefined constants in their paper.

$$scrollspeed = |Y_{ip} - Y_{cp}| \tag{2.5}$$

Equation 2.5: Calculates the scroll speed in a 1D text document, where $Y_{ip}$ represents the initial clicked y position of the cursor and $Y_{cp}$ represents the current y position of the cursor.

$$scrollspeed = \sqrt{(Y_{ip} - Y_{cp})^2 + (X_{ip} - X_{cp})^2} \tag{2.6}$$

Equation 2.6: Calculates the scroll speed in a 2D document, where $X_{ip}$ and $Y_{ip}$ represent the initial clicked x,y coordinates of the cursor, and $X_{cp}$ and $Y_{cp}$ represent the current x,y coordinates of the cursor.

$$zoomlevel = k \times scrollspeed - threshold \tag{2.7}$$

Equation 2.7: Calculates the zoom level, where $k$ is a predefined constant controlling the rate of zooming, $scrollspeed$ represents the scrolling speed in the document, and $threshold$ is a predefined constant representing the scroll speed required before zooming begins.

### *2.4.3 Application of SDAZ on Small Screen Devices*

Jones et al. (2004) conducted an evaluation comparing SDAZ document and map browsers against standard scroll, pan and zoom browsers on small screen displays. They found SDAZ was, on average, 29% slower for browsing text documents. No significant difference was found for mean task completion times when map browsing. SDAZ users, however, performed a significantly lower number of actions completing tasks in both interfaces. The mean number of actions for tasks completed in the document browser was 2.76 using SDAZ and 6.92 using standard techniques. In the map browser, an average of only 1.97 actions were performed using SDAZ compared to an astonishing 17.66 actions using standard techniques. Accuracy was also 40% higher using SDAZ in the map browser, compared to standard techniques. However, the reverse is true in the document browser where accuracy is 29% higher using standard techniques.

Eslambolchilar & Murray-Smith (2004) implemented tilt-based and stylus-based SDAZ text browsers on a Pocket PC. Tilt-based input used an accelerometer attached to the bottom of the Pocket PC to measure the pitch and roll angles. These angles controlled the scrolling speed in the document, with an increase in the tilt increasing the scrolling speed in that direction. As the scrolling speed increased the system automatically zoomed out. In the stylus-based interface scrolling was initialised by placing the stylus on the Pocket PC screen, the displacement of the stylus from the initial point controls the scroll speed. A greater displacement causes a faster scrolling speed and a higher level zoom in the document. No formal evaluations have been performed to assess the efficiency or the usability of these new techniques.

Control feedback for the SDAZ map interface on mobile devices was explored by Alley et al. (2004). Figure 2.9 demonstrates the proposed feedback for the map browser. The inner circle indicates when zooming begins, the outer circle indicates when zooming finishes, the rectangle is the point the system zooms-in to, and the straight line indicates the size and direction of the pointing device's displacement. Alley et al. conducted an evaluation comparing three levels of feedback detail: *all* (all feedback shown in Figure 2.9), *circles* (threshold circles and mouse line only), and *none* (mouse line only). They found that feedback configurations seem to have little impact on the success or performance when navigating to a particular destination. Alley et al. suggest that the intrinsic feedback of SDAZ (transformation of the information space) is sufficient.

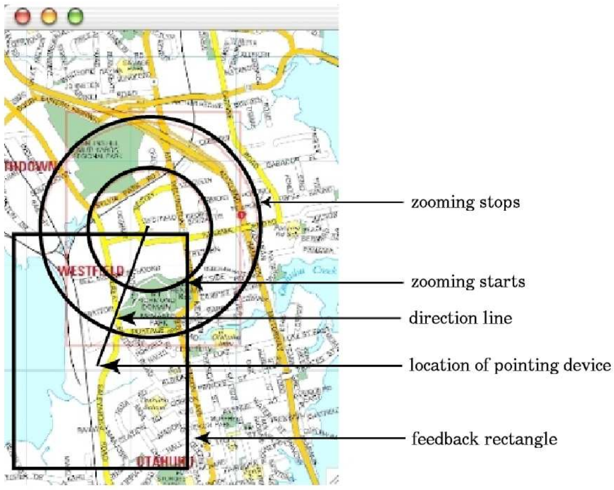
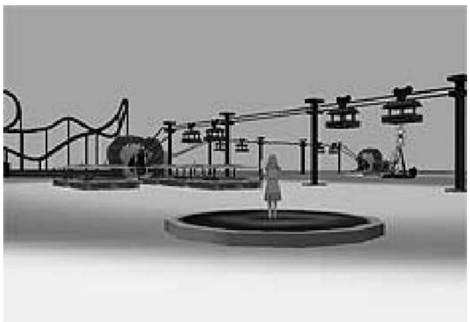


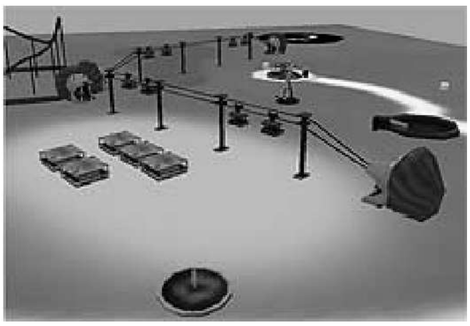

Figure 2.9: Alley et al.'s proposed control feedback design for the SDAZ map browser.

### *2.5 3D Systems*

Depth-modulated flying (Ware & Fleet 1997) and speed-coupled flying with orbiting (Tan et al. 2001) are both techniques used to navigate in a 3D world. Depth-modulated flying uses continuous height sampling to control the flying speed. The idea is to fly slower when closer to the ground and faster when higher. Ware & Fleet conducted an evaluation comparing five different implementations of depth-modulated flying: no velocity scaling (velocity is varied by user interface widgets only), near point scaling (nearest point in the scene determines speed), far point scaling (farthest point in the scene determines speed), average depth scaling (the average depth in the scene determines speed) and average depth scaling with hot spot (the average depth in the scene determines the speed with the weight of the depths in the current viewed area multiplied by five). They claim that near point scaling, average depth scaling and average depth scaling with hot spots were superior, however no statistical tests were used to prove significance. Speed-coupled flying uses the opposite approach. It modulates the height and tilt of the camera based on the flying speed (Figure 2.10). Speed-coupled flying is essentially based on SDAZ but in a 3D world. Tan et al. also added orbiting to speed-coupled flying to allow a user to fly around ('orbit') an object by clicking the object and dragging the mouse. Two evaluations where participants located different objects in a 3D scene were conducted. Speed-coupled flying with orbiting was the most efficient navigation technique compared to three competing techniques: basic driving navigation (with and without orbiting), speed-coupled flying without orbiting, and ephemeral world compression (allows users to shrink the 3D world to gain an overview).



(a) Local view of scene while standing still or moving slowly.



(b) Overview of scene while moving fast.

Figure 2.10: Speed-coupled flying with orbiting.

### *2.5.1 Animation*

One of the goals of SDAZ is to smoothly navigate between two close-up views. van Wijk & Nuij (2003) discuss the 'optimal' animation between two known points of interest based on perceived velocity. They define 'optimal' as smooth and efficient. A smooth animation path should be continuous in the sense that no sudden steps are made or abrupt changes in direction occur. The most efficient path is the shortest path in pan and zoom space so that the camera moves from point A to point B as fast as possible. This would be a straight line between A and B if the smoothness and the viewer's maximum perceived velocity were not considered. By zooming out, the perceived velocity is less than the actual document velocity. Therefore, the camera can be moved faster when it is moved further away from the document, with its perceived velocity remaining the same.

The computational model proposed by van Wijk & Nuij is limited to the perceptual level, and it does not take in account cognitive aspects such as memory, meaning of image or complexity of the image shown. These aspects are extremely hard to incorporate into a formal model. They suggest that a perceptually smooth motion will aid in cognition. The computation model assumes that each part of the image is equally interesting (same characteristics, same visually density etc) but this is not the case in real-world applications, such as cartography where urban areas are more interesting than uniformly coloured oceans.

van Wijk & Nuij suggest that their computational model could be used to implement a system similar to SDAZ. They propose a system that uses the scroll handle on scroll bars to determine the destination point. In the proposed system, the user can scroll to a distant location by acquiring the scroll handle and moving to the desired location; van Wijk & Nuij's model then is used to smoothly animate between the two points. This proposed navigation technique has not yet been evaluated.

## *2.6 Navigation Analysis*

### *2.6.1 Fitts' Law*

Fitts' Law (1954) is a quantitative human performance model used for predicting the time taken to perform one-dimensional movement tasks. Shannons' formulation, shown in Figure 2.8, is an adaption of Fitts' Law for 2D pointing tasks in graphical user interfaces (MacKenzie 1992). The time $MT$ is predicted by Equation 2.8 and relies on the logarithm of the distance that the pointer must move (the amplitude) $A$ over the width of the target $W$. MacKenzie & Buxton suggest setting $W$ to the smaller value of the width and height dimensions. The constants, *a* and *b*, represent the cognition and motor preparation time, and the hand-eye co-ordination, respectively. Since task difficulty is analogous to information, the rate of task execution can be interpreted as the human rate of information processing. Therefore, an index of performance (IoP) or 'bandwidth' in bits per second can be calculated using Equation 2.9.

$$MT = a + b \underbrace{\log_2(\frac{A}{W} + 1)}_{Index\ of\ difficulty} \tag{2.8}$$

$$IoP = \frac{1}{b} \tag{2.9}$$

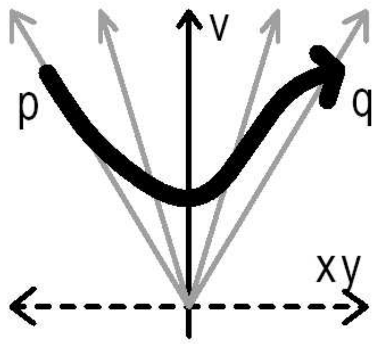
Guiard et al. (1999) suggest that Fitts' law can be extended further to cover multi-scale navigation. They conducted a preliminary evaluation showing that users can handle higher levels of task difficulty ($IoD > 12bits$) with two-scale rather than traditional one-scale pointing control.

### *2.6.2 Space-Time Diagrams and U,W Space Diagrams*

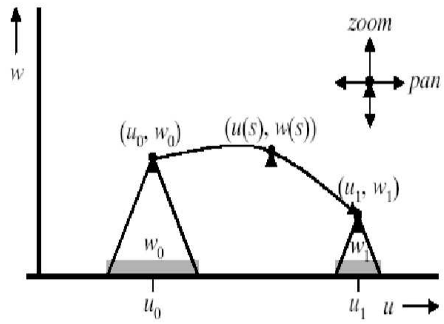
Space-time diagrams and u,w space diagrams are ways of representing movement in 3D space (see Figure 2.11). Furnas & Bederson (1995) created space diagrams to visually depict multiscale viewing (see Figure 2.11(a)). The horizontal axis denotes space, while the vertical axis denotes scale. Furnas & Bederson show how this diagram can be employed to attack a variety of problems associated with multiscale viewing, including optimal pan-zoom trajectories. Optimal is defined as the shortest path. van Wijk & Nuij (2003) adapted the space-time diagram to create a u, w space diagram, where u denotes panning and w denotes zooming (see Figure 2.11(b)). van Wijk & Nuij claim u, w space diagrams are simpler to understand and work with.

## *2.7 Additional Focus Plus Context Techniques*

A well established technique for allowing navigation in large information spaces is overview plus detail (Baldonado et al. 2000, Plaisant et al. 1995, Hornbæk & Frøkjær 2001). This visualisation technique provides a small window, which is either part of the main window or a separate overlaying window, containing an




(a) Example of Furnas & Bederson's space-scale diagram displaying automatic zooming behaviour (v=scale, xy=space, p=initial position, q=target position).



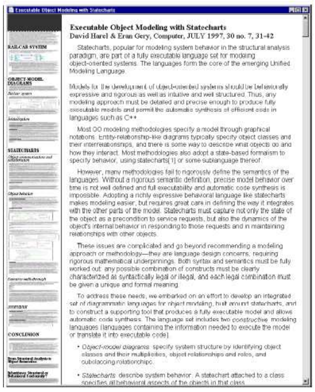

(b) Example of Wijk & Nuij's u,w diagram displaying animation between two points (w=zoom, u=pan, $(u_0,w_0)$=initial point, $(u_s,w_s)$=intermediate point, $(u_1,w_1)$=final point).

Figure 2.11: Examples of diagrams used to represent movement in multiscale environments.

overview of the information space (see Figure 2.12(a)). This window is not only used to show the current location and context in the document, but also to quickly shift from one location to another, a technique generally performed by clicking the desired location inside the overview window. The drawback to this technique is that the overview takes up precious screen space.

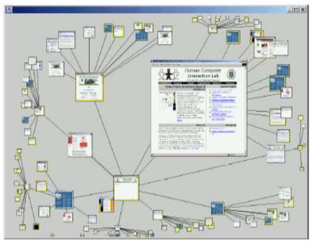
Focus plus context screens, introduced by Baudisch et al. (2001) extend the use of overview plus detail visualisations. Generally, they consist of a wall-size low-resolution display for context with an embedded high-resolution display for detail (see Figure 2.12(b)). The obvious drawback to these types of displays is their cost and sheer size. Baudisch et al. (2002) found that information can be extracted from static documents 21% and 36% faster with focus plus context screens compared to overview plus detail visualisations and zooming panning, respectively.

Another focus plus context technique designed to reduce information overload is the Fisheye view (Furnas 1986). Fisheye views reduce the size of the document by magnifying the focused area, while the non-focused areas are compressed but still visible (see Figure 2.12(c)). A major drawback of Fisheye views




(a) Hornbæk and Frøkjær's overview plus detail visualisation. Overview of the text document is displayed on the left hand side, while the detailed view is displayed on the right.


(b) Baudisch et al.'s focus plus detail screen.



(c) Gutwin's interactive Fisheye view for browsing a web site. The focus point is tied to the mouse cursor, and objects grow larger as the focus approaches them.

Figure 2.12: Common focus plus context techniques.

is that focus-targeting is difficult for most users because targets move in the opposite direction to the motion of the magnifying lens. A focus target will move towards an approaching pointer, and away from a retreating one, making it more difficult to precisely position the focus point relative to the underlying visualised data. To make matters worse targets move fastest when the cursor enters the target, making it difficult for a user to precisely position the pointer over the target. A solution to this problem is speed-coupled flattening (Gutwin 2002). This technique reduces the distortion level of the focus, based on pointer velocity and acceleration. Speed-coupled flattening was proven to significantly reduce both targeting time and targeting errors in Fisheye environments.

Hornbæk & Frøkjær (2001) conducted a study (n=20) comparing the usability of three interfaces: a standard linear interface, a fisheye interface, and an overview plus detail interface. They found essays were written slower using overview plus detail but received higher grades. Documents were read the fastest using the fisheye interface. The standard linear interface was found to be inferior to the fisheye and overview plus detail interfaces in most aspects of usability. All participants but one preferred the overview plus detail interface. Hornbæk & Frøkjær recommend overview plus detail interfaces for electronic documents, while Fisheye interfaces should be used for time-critical tasks.

### *2.8 Human Processing of Visual Flow*

Visual flow is defined as the amount of visual information passing across the screen at any one point in time. The information can be in the form of text and/or images. This section will focus primarily on the discussion of past research related to human processing of visual flow.

Previous research into human visual perception discovered that the human eye summates signals over a period of 120-125ms (Card et al. 1987, Burr 1980, Zagier 1997). Image blurring should then be predicted to occur by displacing the data over a 120ms time slot. Burr (1980) found, however, that the blurring of the image is significantly less than previously predicted. This is because the human visual system reduces the perceived blur by tracking moving objects with 'smooth-pursuit' eye movement (Eckert & Buchsbaum 1993). Tracking moving targets with the eye reduces the relative velocity of the image across the retina.

An image can travel 3 degrees per second across a human retina without an effect on the image acuity (Morgan & Benton 1989). A comfortable retinal-image velocity is only 2 degrees per second (Kelly 1979). Using visually induced conjugate eye movements, viewing velocities are dramatically increased (Missal & Keller 2002). The oculomotor repertoire, controlled by the nervous system, contains two different kinds of visually induced conjugate eye movements: smooth-pursuit and saccades. The smooth-pursuit system is used when an object is moving less than 100 degrees per second (Krauzlis & Stone 1999). As soon as an object starts to move, the smooth-pursuit system initiates an eye velocity equal to that of the target. This reduces the relative velocity on the retina so that the image is stabilised on the foveal region. The saccades system induces a rapid shift of the visual axis between different positions. As no visual feedback is used to guide the orientation of the eyes during the movement, the velocity can be extremely high, up to 1000 degrees per second. However, image acuity is dramatically reduced.

Preattentive processing allows humans to automatically process the organisation of the visual field. Examples of visual features that can be detected in this way include hue, intensity, orientation, size and motion. A task that is processed preattentively (i.e. without the need for focused attention) is completed in less than 200 milliseconds (Healey et al. 1996). This is because eye movements take at least 200 milliseconds to initiate.

### *2.8.1 Rapid Serial Visual Presentation*

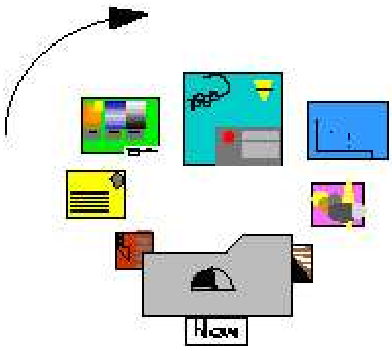
A technique that takes advantage of preattentive processing is Rapid Serial Visual Presentation (RSVP). RSVP is described as the "electronic equivalent of riffling a book in order to assess its content" (Spence & deBruijn 1999). Words or pictures are presented successively, often at fast rates, at the same spatial location on the screen. Generally, objects in RSVP systems are visible for 200–400 milliseconds (Spence 2002). Humans have the ability, however, to recognise the presence of a target image in as little as 100 milliseconds or even less (Healey et al. 1996).

The concept of RSVP was first introduced by Forster (1970) to study the comprehension and processing of written language. He performed four similar experiments in which words were displayed in rapid succession; participants were then asked to record what they had seen. He found that an increase in the syntax complexity or information content of the sentence, required an increase in expose time to the visual display, otherwise information is lost by the participant.

In RSVP, objects are presented in the same spatial location to prevent the breakdown of the subject's capacity to organise the input sequence, affecting the comprehension of the information (Forster 1970). If the individual words of a sentence are consecutively displayed so that each word falls on roughly the same region of the retina, then the occurrence of each new word eliminates the preceding word from the sensory storage system (Kahneman 1968). Thus, any possibility of a cumulative sensory storage of the input is eliminated, and the subject is forced to process each word as it occurs.

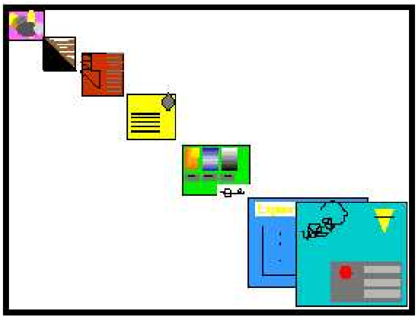
The adaption of RSVP for computer systems was first established by Aaronson & Scarborough (1977). Since then the technique has been generalised to the field of information navigation (Spence & deBruijn 1999). Spence (2002) suggests the main RSVP navigation interfaces are 'keyhole', 'carousel', 'collage', 'shelf' and 'floating'. The keyhole interface is similar to a slide show. A single image is presented and updated at regular intervals. The images shown using the keyhole interface are always at the same location on the screen. The carousel interface displays a series of images on the screen simultaneously. The images begin small and are rotated in a clockwise direction. As the images come closer to the top of the rotation, they increase in size (see Figure 2.13(a)). The collage interface overlays new images over the top of previous images (see Figure 2.13(b)), similar to a person dropping photos one by one on a table top infront of the user (Spence 2002). In shelf RSVP, images are initially displayed at 'full size' in the lower right-hand corner of the display. Images are moved at a constant speed, along a linear trajectory, towards the upper left-hand corner of the display. The images are decreased in size as they move towards the upper left-hand corner (see Figure 2.13(c)). In floating RSVP, new images appear in the distance and move towards the user until they disappear from the screen. This gives the user an effect similar to driving down a highway watching billboard advertisements pass by (see Figure 2.13(d)).

Spence & deBruijn conclude that RSVP is a valuable technique for searching and browsing information on small screen displays, such as PDAs and mobile phones. Research by Rahman & Muter (1999) has shown that RSVP can be used to present text on small displays, and is as efficient as the normal page format. Normal page format is analogous to text on the page of a book. Participants generally dislike using RSVP, compared to other techniques such as times square (Kang & Muter 1989), sentence-by-sentence and normal page format. Attempts to improve preference for and feasibility of RSVP have been explored (Rahman & Muter 1999). Rahman & Muter found that user preference was significantly improved over normal RSVP by including a completion meter, punctuation pauses, interruption pauses and pauses at clause boundaries.

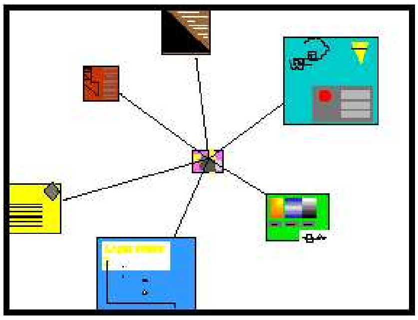



(a) Carousel RSVP.

(b) Collage RSVP.



(c) Shelf RSVP.



(d) Floating RSVP.

Figure 2.13: Spence RSVP modes.

# Chapter III

# Fundamentals of SDAZ Behaviour

The commercial viability of SDAZ largely depends on its 'out of the box' configuration. Its default behaviour needs to be as optimal as possible because default settings are rarely modified. Configuration tools should still be provided for advanced users to tailor personal settings.

The primary goal of this chapter is to identify the low level components of SDAZ behaviour so that parameters and optimal values for the parameters can be established. The identification of components will also help in the development of tools to accurately and completely tailor SDAZ.

The secondary goal of this chapter is to identify and discuss additional features enhancing the usability of SDAZ for future implementers (see Section 3.2). Features discussed in this section include: how the user scrolls in the interface, what is displayed to convey the scroll speed and direction, and what region the system rezooms to after the user has finished scrolling.

## *3.1 Low Level Behaviour of SDAZ*

This section describes the low level components of SDAZ behaviour: the core mappings and the constraints limiting the rate of change in these mappings. There are three core mappings performed in series determining the behaviour of SDAZ (enumerated below). The constraints placed on these mappings are discussed in Section 3.1.1.

1. **Mouse Motion to Cursor Displacement** — The computer's control-display gain maps the physical displacement of the mouse to the corresponding movement of the cursor on the screen (see Section 2.1). Generally, this mapping should be left to the Operating System settings.

2. **Cursor Displacement to Scroll Speed** — This mapping describes the relationship between cursor displacement and the corresponding scroll speed in the document. An example linear relationship is illustrated in Figure 3.1(a). By itself this mapping describes the behaviour of rate-based scrolling.

3. **Scroll Speed to Magnification Level** — This mapping describes the automatic zooming behaviour: the relationship between the scroll speed and the magnification of the document. An example relationship is illustrated in Figure 3.1(b).

When using SDAZ, mouse movement is translated using the control-display gain settings to cursor movement, this is mapped to a document scrolling speed and the zoom level is then automatically adjusted to keep visual flow across the screen constant.

This thesis focuses on the calibration of the second and third mappings: the relationship between cursor displacement and scroll speed, and the relationship between scroll speed and magnification. Values for these mappings are provided in Section 5.3. The control-display mapping is not calibrated in this thesis because we believe it should be left to the Operating System settings for consistency with other interfaces.

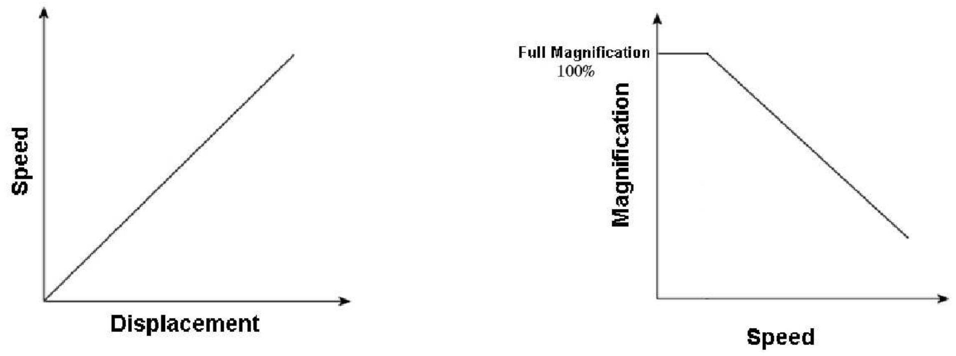
### *3.1.1 Mapping Constraints*

This section focuses on constraints limiting the rate of change of scrolling velocity and magnification. We have defined four possible constraints, however, we believe only three are necessary to achieve optimal calibration.

1. **Maximum Ascent Rate** — limits the maximum decrease in magnification per second. Prevents users from zooming out of the document too fast.

2. **Maximum Descent Rate** — limits the maximum increase in magnification per second, while the user is scrolling. Reduces the 'slamming' effect experienced when changing direction (see Figure 3.2).

3. **Maximum Fall Rate** — limits the maximum increase in magnification per second, after the user has finished scrolling.




(a) Linear mapping between cursor displacement and scroll speed in the document.

(b) An example mapping between scroll speed in the document and magnifi cation measured as a percent.

Figure 3.1: SDAZ's core mappings. These graphs are designed for demonstration only. Recommended relationships are provided in Section 5.3. The current magnification is found from the current scroll speed which is mapped from the cursor's displacement.

4. **Maximum Scrolling Acceleration** — describes the maximum increase in scrolling speed per second. Depending on the configuration, this constraint is optional.

*Smooth Navigation — Max Ascent Rate Vs Max Scrolling Acceleration*

The goal of maximum ascent rate and maximum scrolling acceleration is to facilitate smooth navigation. Smooth navigation is seamless movement, horizontally or vertically, occurring at a rate that is comfortable to the user without being frustratingly slow. The behaviour we are trying to eliminate with these constraints is dramatic increases in scrolling speed or magnification caused by rapid changes in displacement. For example, when the user quickly increases the cursor displacement to scroll to a distant position or when they suddenly reverse the scrolling direction.

We believe that only one constraint, either maximum ascent rate or maximum scrolling acceleration, is required to prevent dramatic changes in speed. This is because magnification is dependent on scroll speed, which means if scroll speed is constrained then the maximum increase in magnification is also constrained, and vice versa. If both are incorporated into the model only the lowest constraint is in effect. The decision between which constraint to use is up to the implementer. This thesis has used the maximum ascent rate rather than the maximum scrolling acceleration to configure SDAZ.

*Rezooming Constraints — Max Descent Rate Vs Max Fall Rate*

The function of maximum descent rate and maximum fall rate is to limit the speed at which rezooming occurs in the document. The only difference between these rates is that one takes effect while the user is scrolling (maximum descent rate) and the other takes effect when the user stops scrolling by releasing the mouse button (maximum fall rate).

The maximum descent rate limits the maximum increase in magnification per second, while the user is scrolling. The primary goal of this constraint is to prevent rapid rezooming when reversing direction. Without this constraint, when reversing direction in a zoomed out view, the user is rapidly 'slammed' from an overview, to fully zoomed in, to an overview again (see Figure 3.2). Users find this disconcerting and it should therefore be avoided. When configuring the maximum descent rate, a compromise needs to be made between a low descent rate providing almost no magnification change when reversing direction and a high descent rate which increases zoom responsiveness.

The maximum fall rate limits the speed at which the user falls back into the document when they release the mouse button. Without using a fall rate, the user is suddenly zoomed to full magnification without any animation after releasing the button. This can lead to disorientation. Generally, the maximum fall rate is set higher than the maximum descent rate because when a user has finished scrolling they typically want to be brought to full magnification as fast as possible, while maintaining orientation. Section 3.2.2 describes rezoom animation paths using the maximum fall rate.

## 3.2 Additional Features Enhancing Usability

Section 3.1 described the low level behaviour of SDAZ: the relationship between mouse displacement, cursor displacement, scroll speed and magnification, and the

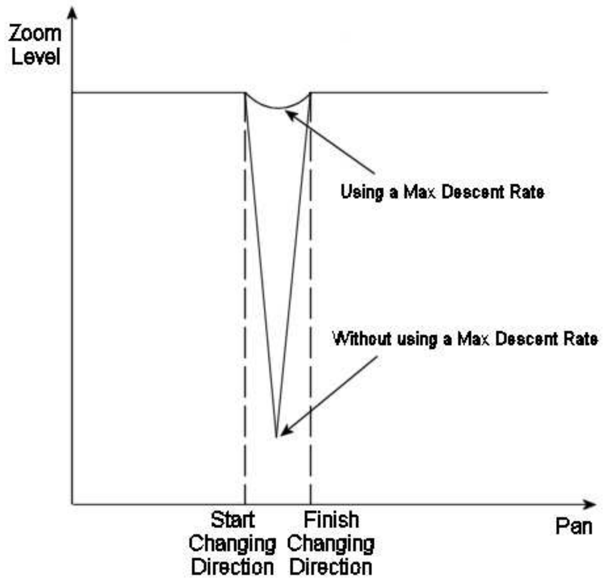



Figure 3.2: Demonstrates how the maximum descent rate reduces the ‘slamming’ effect when users reverse their scrolling direction.

constraints used to limit the rate of change in some of these relationships. This section focuses on additional features that enhance the usability of SDAZ.

### *3.2.1 Mouse Behaviour*

In Cockburn & Savage and Igarashi & Hinckley’s systems scrolling is performed by holding down the left button and moving the mouse in the desired direction. MacKenzie et al. (1991) state that dragging while holding down the mouse button is slower and more difficult than normal pointing. This suggests that SDAZ could be improved by using a modal system (a discrete mouse click) rather than a dragging one. For example, rate-based scrolling could be initialised and ended with a single click. Microsoft’s rate-based scrolling systems use modal interaction (the middle button is used to start and end scrolling) but they also permit dragging with the middle button. Generally, modal systems are avoided because users can forget which mode they are in, resulting in unwanted actions. However, modal systems could be extremely useful if the user’s dominant action is scrolling: for example, when a user is previewing a document to see if it contains the information they require. If the user is currently in the scrolling mode they only need a small movement of the mouse to scroll to the next area of interest.

Mouse behaviour in our systems is similar to Microsoft's — dragging with the middle button mouse increases scroll speed. The left button was not used because it is needed when interacting with the widgets such as buttons and menus. Modal interaction was removed from our interfaces because we did not feel it was necessary and could potentially confuse users.

### *3.2.2 Zoom Focus*

Another issue of concern is the location the system rezooms to, after the user has released their mouse button. Does the system move and zoom to the user's cursor or just zoom to the centre of the screen?

The SDAZ systems evaluated by Cockburn & Savage used the centre point as the zoom-in location when scrolling has finished. This can present problems to first-time users of the system because users generally focus on the cursor and expect the display to zoom to that point. The first-time user can become confused if the point they were just focusing on disappears from the screen when the document is remagnified. There are a few potential solutions to this problem:

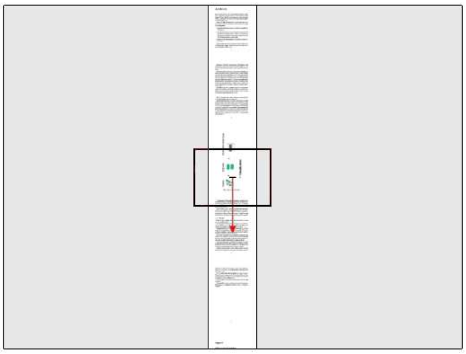
1. Glue the cursor to the centre of the screen while scrolling.

2. Indicate the zoom-in position is the centre (see Figure 3.3)

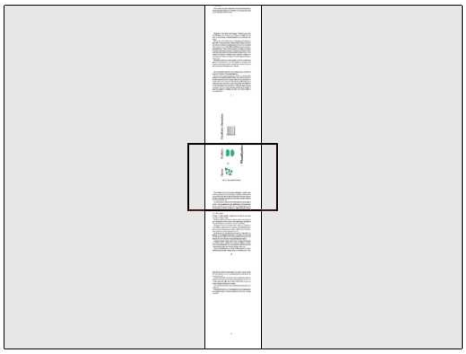
3. Zoom to the current cursor's position (see Figure 3.4)

Fixing the cursor in the centre of the screen while scrolling generally keeps the user's eye focused on the cursor at the centre of the screen. However, it does present feedback problems because the user does not have any control over the cursor.

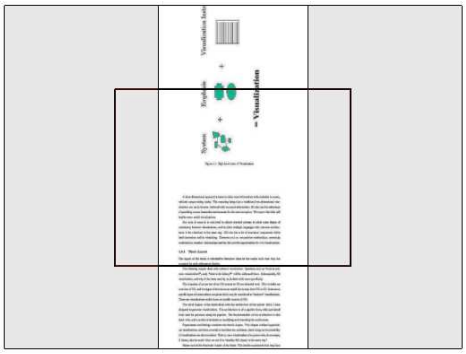
The second solution indicates the zoom-in position is the centre by placing a red box around the zoom-in region. This may also cause focus problems because a user will be focusing on his cursor while scrolling, then when they have found their target will have to make sure their target is inside the box to be zoomed.



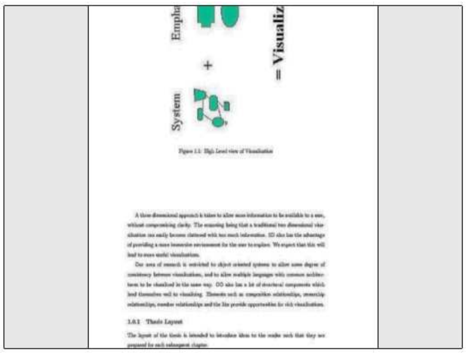
(a) Scrolling in the text document at maximum speed.



(b) Mouse button is released. Zero scroll speed in document. Vertically falling to 100% magnifi cation.

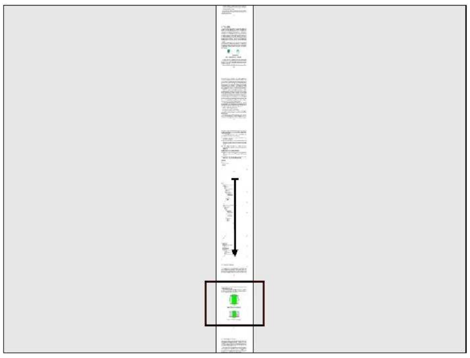


(c) Vertically falling to 100% magnifi cation.

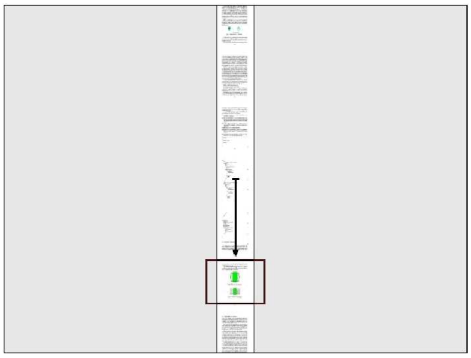


(d) Finished zooming. Document is at 100% magnifi cation.

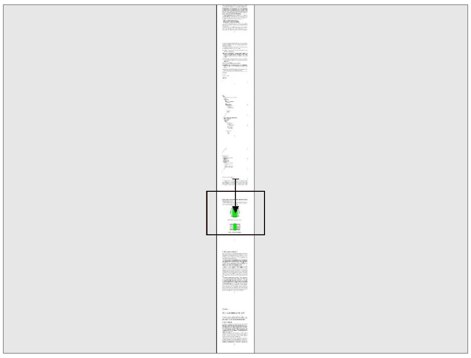
Figure 3.3: Demonstrates zoom-to-centre SDAZ.



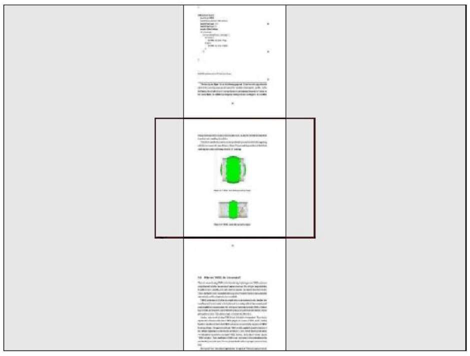
(a) Scrolling in the text document at maximum speed.



(b) Mouse button is released. Scrolling to mouse cursor at current magnification.



(c) Scrolling to mouse cursor at constant magnification.



(d) Mouse point reached. Vertically falling to 100% magnification.

Figure 3.4: Demonstrates zoom-to-cursor SDAZ.

The third solution — zoom to the current cursor's position — seems to be the best solution. At a demonstration evening over 30 participants used both zoom-to-centre and zoom-to-cursor techniques. All users stated that they found the zoom-to-cursor technique more intuitive. Some users felt frustrated with the zoom-to-centre technique because they saw the target area a distance away but had to wait until the area was positioned under the centre of the screen in order to acquire it. The potential drawback of zoom-to-cursor is that the user has less time to react to the oncoming target and may scroll past it. In practise, however, we found this not to be a large problem as users generally see the target a distance away, then move the red box to acquire to it. Therefore, we believe that zooming to the cursor should be used in SDAZ systems.

When implementing zoom-to-cursor several paths can be used, as shown in Figure 3.5. The simplest of the paths is to keep the magnification constant, scroll to the cursor's point then fall to full magnification (see Figure 3.5(a)). This path was used at the demonstration evening. Another possible path is van Wijk & Nuij's optimal animation path (see Section 2.5.1). This animation, however, may cause the user to feel out of control because the system will most likely zoom-out first then zoom-in again to achieve an optimal path (see Figure 3.5(c)). An alternative path is to move straight, using panning and zooming, to the cursor's position (see Figure 3.5(d)). Figure 3.5(b) shows zoom-to-centre for comparison.

In our systems we implemented zoom-to-cursor using the simplest method — keep magnification constant, scroll to the cursor's point then fall to full magnification. We used this method for three reasons. First, it provided a rapid movement to the position without being overwhelming. Second, it was the simplest path to implement. Finally, the method seemed natural to use because the system continues scrolling at the same speed and in the same direction the user was scrolling in before, until it reaches the cursor's point, then falls to 100% magnification. Future research needs to be conducted to evaluate users' preference and the efficiency of each scroll path when navigating to different targets.

### *3.2.3 Heads-Up-Display*

In our systems an arrow extends from the centre of the screen to the cursor's position to indicate the direction and speed of scrolling. In Cockburn & Savage's document viewer a box on the right-hand side of the screen is used to indicate scrolling velocity, zooming thresholds and location in the document (see Figure 3.6). We believe this bar is unnecessary because SDAZ will most likely be used in conjunction with a scroll-bar, which already indicates the document location.

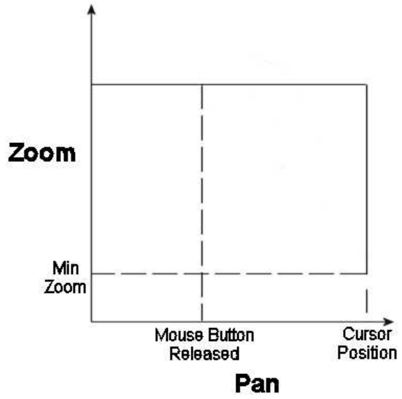
Alley et al. (2004) proposed threshold circles to indicate the cursor displacement required to begin and finish zooming (see Section 2.4.3 for further details of their study). Their evaluation, however, showed that the extra feedback seemed to have little impact on the success or performance when navigating to a particular destination. Alley et al. theorise that the intrinsic feedback of SDAZ (transformation of the information space) is sufficient. Our previous experience with SDAZ interfaces has led us to support their theory.

In Microsoft interfaces where rate-based scrolling is supported the initial clicked position is indicated by an icon showing the directions the user can scroll in (see Figure 3.7). While scrolling, the cursor is changed to an arrow indicating direction of movement.

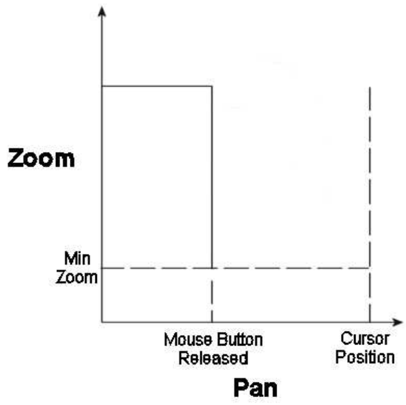
## *3.3 Summary*

This chapter identified the low level components of SDAZ behaviour for the development of calibration tools (see Chapter 5 for tool development). We have outlined the two core mappings requiring calibration: the relationship between cursor displacement and scroll speed (rate-based scrolling behaviour), and between scroll speed and magnification (automatic zooming behaviour). We have also defined four constraints limiting the rate of change of magnification and scroll speed. These were the maximum ascent rate, maximum descent rate, maximum fall rate and maximum scrolling acceleration. Recommended values for the mappings and constraints are provided Section 5.3.

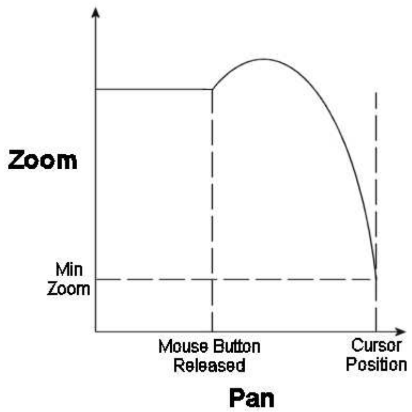
In Section 3.2 we identified additional features enhancing the usability of SDAZ. We first discussed the behaviour of the mouse, recommending holding the down the middle button to scroll instead of using the left button or modal interaction. We then discussed the position rezoomed when the user has finished scrolling, recommending zooming to the cursor using the simplest path — keep magnification constant until the cursor's location is reached (see Section 3.2.2). Finally, we discussed the Heads-Up-Display (HUD), advocating a standard line indicating the cursor's current displacement from the centre of the screen is sufficient feedback for the user. This is supported by Alley et al.'s study into potential SDAZ feedback (see Section 2.4.3).



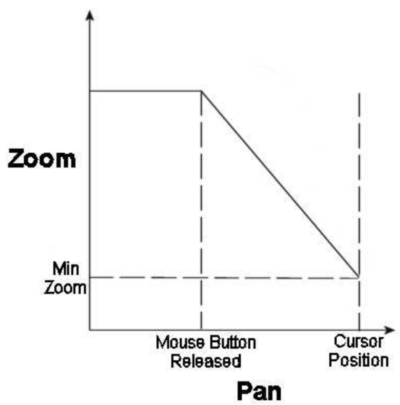

(a) Zoom to the cursor keeping zoom level constant until cursor position is reached.




(b) Zoom to the centre of the screen.




(c) Zoom to the cursor using Wijk & Nuij's optimal trajectory.




(d) Zoom to the cursor using straight zoom pan trajectory.

Figure 3.5: w,u space diagrams displaying zoom-in trajectories. The cursor position represents the position the cursor was over when the mouse button was released.

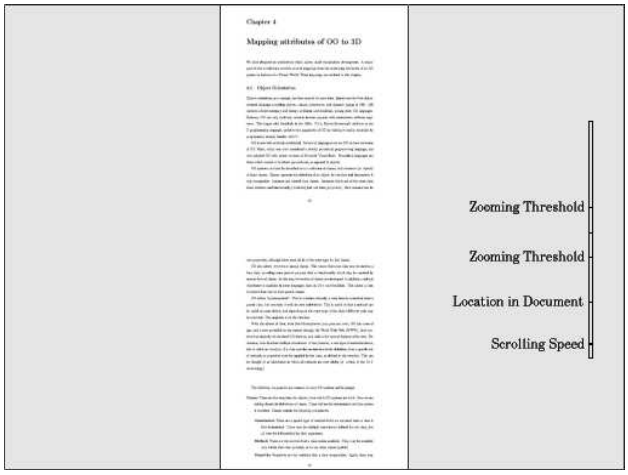



Figure 3.6: Cockburn and Savage's Document Viewer.

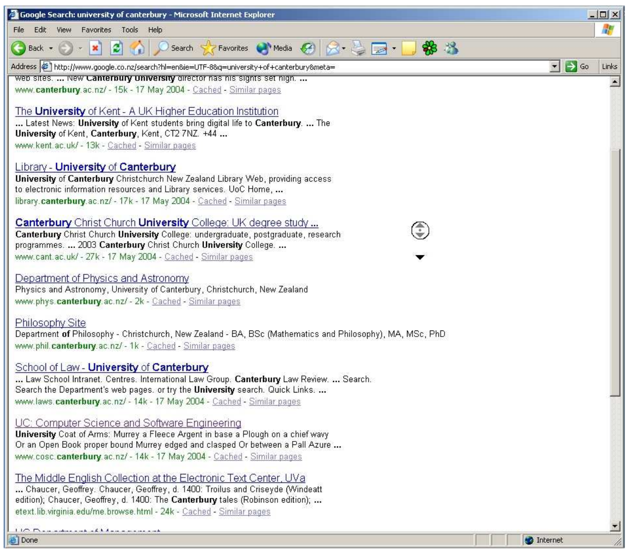



Figure 3.7: Rate-based scrolling in Microsoft Internet Explorer.

# Chapter IV

# Visual Flow Theory and Evaluation

## *4.1 Motivation*

The goal of this chapter is to provide optimal settings for the automatic zooming mapping describing the relationship between scroll speed and zoom level, identified in Chapter 3. An optimal relationship must facilitate smooth, natural navigation at the fastest comfortable speeds. Currently, optimal configuration is difficult to achieve; if the scroll speed is too fast before zooming occurs, users become overwhelmed by the rapid visual flow, but if the scroll speed is too slow users become impatient and frustrated because the system takes too long to scroll to the required information.

Previous work by Igarashi & Hinckley (2000) and Cockburn & Savage (2003) fails to present values for predefined constants governing automatic zooming behaviour. The absence of formal guidelines means that SDAZ implementers are forced to adjust the properties of the automatic zooming by trial and error.

For the commercial viability of SDAZ it is very important its default configuration is optimal for general users. Configuration options should be provided for advanced users even though generally users do not change default settings.

## *4.2 Approach*

To establish an optimal mapping between document scroll speed and magnification, we used a naturalistic approach measuring participants' scrolling speeds while locating targets at different document magnifications. Optimal, in this case, is defined as a mapping that allows users to scroll at the fastest speed, still perceived as comfortable, for each magnification level. The optimal mapping, alternatively, could have been defined as a mapping that allows users to scroll at the fastest tolerable speed for each magnification level. This would allow greater scrolling speeds, however, it would compromise the comfort and possibly users preference of the automatic zooming over traditional scrolling techniques. Therefore, this chapter's evaluation is designed to establish a relationship between participants' maximum comfortable scrolling speeds and magnification level.

A low level theoretical analysis of visual flow is discussed first in Section 4.3. This section calculates the theoretical maximum scroll speeds in our evaluation. The main visual flow evaluation method, procedure and design is outlined in Section 4.4. This evaluation is designed to obtain an optimal relationship between comfortable scroll speed and magnification. Results from the evaluation are presented in Section 4.5 and discussed in Section 4.6. Additional analyses on the effects of gaming experience, gender and direction of comfortable visual flow rates are presented in Section 4.7. Finally, new speed and magnification mappings are proposed for SDAZ in Section 4.8.

### *4.3 Theoretical Analysis of Maximum Visual Flow Rates*

This section attempts to calculate the theoretical upper scroll speed limits at different magnifications. First, we will derive a formula for calculating the maximum document scrolling speed trackable by the smooth-pursuit. We will then derive a formula calculating the approximate time in seconds an image must remain on the screen before it can be recognised at different magnifications. This is used to calculate the theoretical maximum document scroll speed that still allows participants to read the text or identify the image.

In the visual flow evaluation the participant's eye will be attempting to recognise fast moving objects (text headings and images) on the screen. Figure 4.1 displays the eye's movement angle when following an object in a straight line vertically or horizontally across the screen. This is approximately 31 degrees vertically and 42 degrees horizontally, assuming that the eye is 50 centimetres from the screen. The smooth-pursuit system is induced by the nervous system to track a target moving less than 100 degrees per second (see Section 2.8 for related work). This means the maximum movement across the screen using the smooth-pursuit system is approximately 87 cm/second vertically and 85 cm/second horizontally on the 19" Compaq monitor.

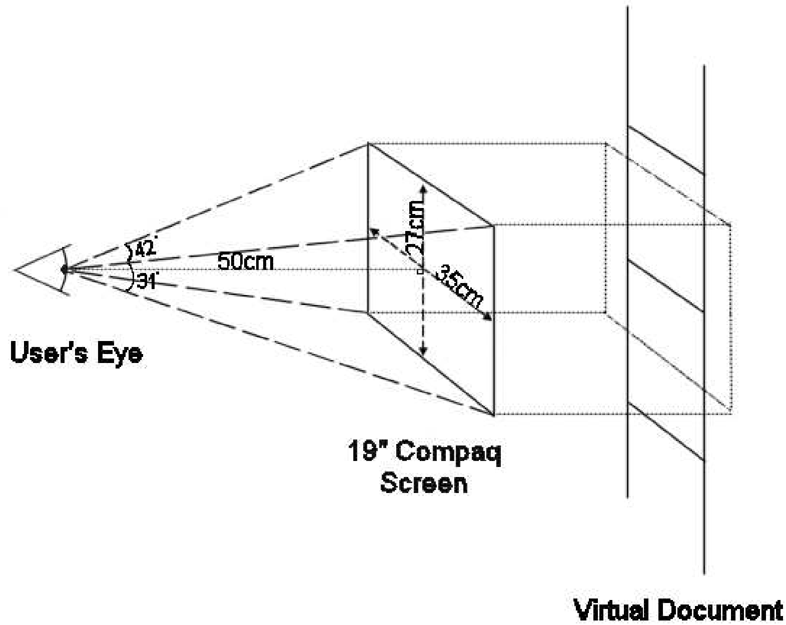



Figure 4.1: Diagram illustrating the theoretical calculation of the maximum smooth-pursuit velocities in our evaluation setup.

To simplify the discussion of visual flow we define two rates of movement: the screen movement rate and document movement rate. The screen movement rate is the speed pixels move across the screen. The document movement rate (or document scroll speed) is the speed the virtual document moves in 3D space. At full magnification (100%) the screen movement rate and document movement rate are equal. However, as magnification decreases the screen movement rate decreases proportionally, if the document movement rate remains constant. For example, if the magnification is reduced from 100% to 50% the screen movement rate is halved provided the document movement rate remains the same. The relationship between the screen movement rate and document movement rate is shown in Equation 4.1. This was used to derive Equation 4.2 calculating the maximum document movement rate trackable by the smooth-pursuit system in our evaluation setup.

$$DMR = \frac{SMR}{(mag/100)} \tag{4.1}$$

Equation 4.1: The relationship between the screen movement rate (*SMR*), measured in cm/sec and the document movement rate (*DMR*), also measured in cm/sec. *mag* is the magnification of the document.

$$DMR = \frac{87}{(mag/100)} \tag{4.2}$$

Equation 4.2: Calculates the theoretical maximum document movement rate trackable by the smooth-pursuit system in our evaluation setup. *mag* represents the document's current magnification. *DMR* is measured in cm/sec.

Now that we have established the upper limit to the smooth-pursuit tracking system, we will calculate the theoretical time to identify and recognise a target being tracked using the smooth-pursuit system. The enumerated list below outlines the steps required for this calculation.

1. To identify a scrolling image or heading the user must initiate the smooth-pursuit system. This takes between **100–150 ms** (Blohm & Schreiber 2002).

2. The user must then visually process the image or read the text. An image takes approximately **100 ms** to process (Spence 2002). Text can be read at approximately 280 words per minute (Kang & Muter 1989). Thus, reading an average three word heading will take the user, approximately **600 ms**. This is assuming that the words are read and not pattern matched as an image. If the headings are able to be treated like an image to the searching participant, processing should only require **100 ms**.

3. Once the image or heading is processed, it must be recognised as the target object. This requires one cycle of both the visual and cognitive systems, requiring approximately **200 ms** (Card et al. 1987).

The steps above can be summarised in a formula calculating the theoretical 'on-screen time' required for the recognition of a target (see Equation 4.3). On-screen time is the amount of time in seconds an image must remain on the screen before it can be recognised. To simplify the calculation we have assumed that the steps are completed in series, however, in reality some steps might be conducted in parallel. Using this formula to predict the on-screen time for the recognition of a target image and a target three word heading gives approximate on-screen times of 450 ms and 950 ms, respectively.

$$OnScreenTime = InitSPTime + IdentTime + RecogTime \tag{4.3}$$

Equation 4.3: Calculates the theoretical *on-screen time* required to recognise a target. *OnScreenTime* is the amount of time in seconds an image must remain on the screen before it can be recognised. $InitSPTime$ represents the time in seconds taken to initialise the smooth-pursuit system. $IdentTime$ represents the time in seconds taken to identify and process the image. $RecogTime$ is the time in seconds required to recognise the image as the target.

The screen movement rate can be calculated using the on-screen time for recognising moving text (950 ms) or a image (450 ms) with Equation 4.4. The application window size ($AppSize$) is the length of the application window in the direction the object is moving. For example, in the text document interface an image moves vertically across the screen so the application window size is 27 cm vertically, when the application is viewed full-screen. Using Equation 4.4 an image moving vertically across the screen can still be recognised at a screen movement rate of 60 cm/second and a heading moving vertically across the screen can still be recognised at a screen scrolling speed of 28 cm/second.

$$SMR = \frac{AppSize}{OnScreenTime} \tag{4.4}$$

Equation 4.4: Calculates the screen movement rate (*SMR*) in cm/second, given the application window size $AppSize$ in cm, and the on-screen time ($OnScreenTime$) in seconds.

The document movement rate can be calculated using Equation 4.5. Using this equation, Figure 4.2 graphically displays the theoretical maximum visual flow rates in the document viewer (where objects move vertically) for the smooth-pursuit system, recognising an image and recognising a three word heading. An almost identical graph can also be plotted for the theoretical visual flow rate horizontally.

$$DMR = \frac{100 \times AppSize}{mag \times OnScreenTime} \tag{4.5}$$

Equation 4.5: Calculates the theoretical document movement rate (*DMR*) in cm/second, given the magnification level (*mag*) as a percentage, the application window size (*AppSize*) in cm, and on-screen time (*OnScreenTime*) in seconds.

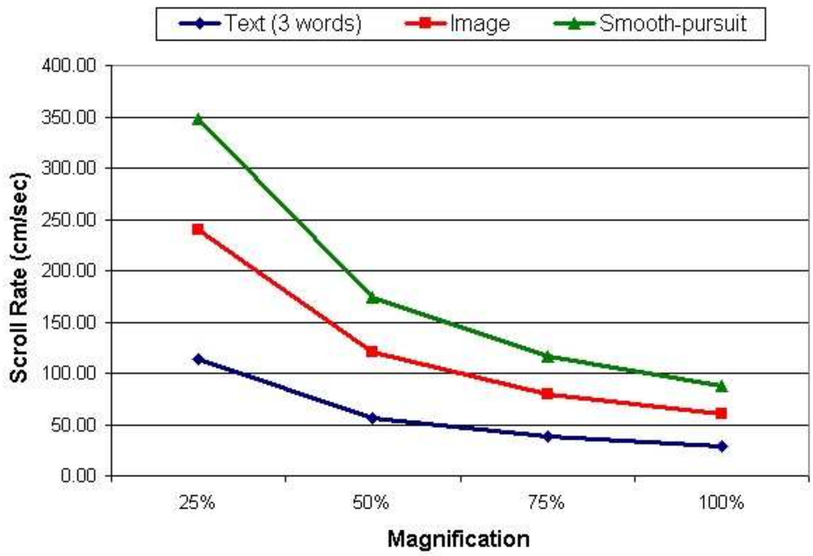



Figure 4.2: Theoretical document scroll rates for processing and recognising an image or three words of text at different magnifications with an application size of 27 cm. Also, displays the predicted maximum smooth pursuit rates at different magnifications.

## 4.4 Evaluation

### 4.4.1 Goal

The goal of this evaluation is to establish a relationship between participant's maximum comfortable scrolling speed and magnification. The evaluation's results will be used to calibrate our interfaces so users can scroll comfortably at optimal speeds, at each magnification level. This evaluation will also aid in the design of other rate-based scrolling interfaces, as it answers the question "how fast is too fast when scrolling"? It will help designers set upper limits to their rate-based scrolling interfaces, so that users do not feel out of control when they unintentionally move the mouse or joystick too far and begin to scroll at an extreme rate.

### *4.4.2 Apparatus*

The experiments were conducted on a Athlon 2200++ computer with 512Mb of RAM running Windows XP, with a Geforce 4 ti4600 video card outputting to a 19-inch Compaq monitor at a resolution of 1024 x 768 pixels. The Compaq monitor had a viewable screen width and height of 36cm and 27cm, respectively. The OpenGL interfaces were executed in the full-screen game mode, hence, the frames per second (fps) displayed was limited to the monitor refresh rate (70 fps/70 hz). Input was given through an optical 3 button Logitech mouse.

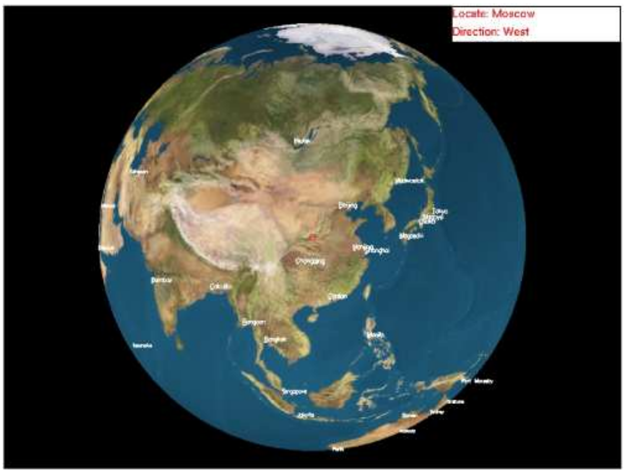
The interfaces looked very similar to Cockburn & Savage's document and map OpenGL viewers and Cockburn et al.'s OpenGL globe viewer. In all the interfaces, however, the automatic zooming feature was removed leaving only rate-based scrolling. Screen-shots of the three interfaces are shown in Figure 4.3.

At 100% magnification the document viewer displayed one full size A4 page, measuring 21x27 cm on the screen. The map viewer at 100% magnification displayed 2 grid squares vertically by 2.6 horizontally.

In our evaluation, the maximum and minimum magnification levels attempted to represent the levels users would view the document at when navigating. At the maximum magnification (100%) the smallest text (i.e. the document body) was clearly readable and at the minimum magnification the largest text (i.e. the headings) was 'barely' readable. We defined barely readable using Tullis et al.'s study into the legibility and reading times of different fonts and font sizes. This study found that a font size of 7.5 point arial is just readable but should be avoided. For experimental repeatability Table 4.1 gives font measurements for the smallest and target text in each of the interfaces at maximum and minimum magnifications. Only the target text size is provided in the globe viewer because the interface only has one text size. In our evaluation, a font's size was measured by its ascent, x-height, descent and maximum width. Figure 4.4 shows how these properties are measured (Williams 1998). In our evaluation, the target text is always the largest


(a) The document viewer.


(b) The map viewer.




(c) The globe viewer.

Figure 4.3: Visual flow evaluation systems.

| Interface | Mag | Text Size | Ascent | X-Height | Descent | Max Width |
|---|---|---|---|---|---|---|
| Doc | 100% | Smallest<br>Target | 1 mm<br>2 mm | 2 mm<br>4 mm | 1 mm<br>2 mm | 3 mm<br>9 mm |
| | 25% | Smallest<br>Target | 0.25 mm<br>0.5 mm | 0.5 mm<br>1 mm | 0.25 mm<br>0.5 mm | 0.75 mm<br>2.25 mm |
| Map | 100% | Smallest<br>Target | 2 mm<br>4 mm | 4 mm<br>9 mm | 2 mm<br>- | 8 mm<br>17 mm |
| | 12.5% | Smallest<br>Target | 0.25 mm<br>0.5 mm | 0.5 mm<br>1.125 mm | 0.25 mm<br>- | 1 mm<br>2.125 mm |
| Globe | 100% | Target | 2 mm | 4 mm | 2 mm | 6 mm |
| | 50% | Target | 1 mm | 2 mm | 1 mm | 3 mm |

Table 4.1: Font measures for smallest and target fonts, in each of the interfaces, at maximum and minimum magnifications.

text in the document. In the map viewer, no descent value is given for the target text because it is always in upper case.

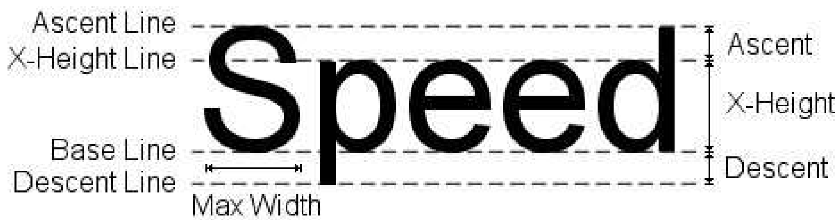



Figure 4.4: Diagram displaying how text was measured in our evaluations.

The three documents used in the evaluation were: a report on Virtual 3D Worlds for Enhanced Software Visualisation (Keown 2000) for the document viewer, an Auckland city map for the map viewer and a globe texture with major cities marked and labelled in white for the globe browser. Practise tasks in the document viewer were completed in Savage's report on Speed-Dependent Automatic Zooming (Savage 2002).

### 4.4.3 Method and Procedure

This evaluation attempts to maximise ecological validity by having users conduct 'natural' tasks while the software logged their scrolling velocities. Participants were asked to find a target at discrete magnification levels (determined by the evaluation system) using rate-based scrolling only, given the direction in the document. They were instructed to locate the target as fast as possible, while making sure the scrolling rate was still comfortable and not overwhelming.

Participants in the evaluation consisted of 14 males and 6 females. All were senior undergraduate students and had experience with computers.

At the beginning of each task, participants were given a description of the target picture or text. Targets were placed at least three screen lengths away from the initial position to allow participants to reach a comfortable scrolling speed before locating the target. In-case the subjects forgot what they were searching for, the current target's description and heading were displayed in the top right-hand corner of the screen (see Figure 4.3). The text describing the target and the direction were highlighted in red. The direction text was static, it did not change if the participant scrolled past the target. Example tasks are listed below:

- Document Interface Tasks
  - Locate the **"Mapping Attributes"** chapter heading located **down** from here.
  - Locate the **first bar graph** located **up** from here.
- Map Interface Tasks
  - Locate the suburb heading **TITIRANGI** located **north-west** from here.
- Globe Interface Tasks
  - Locate the city called **Teheran** located **south** from here.

Participants navigated in the document using rate-based scrolling at fixed zoom levels. The document scrolling speed was increased/decreased by holding down the left mouse button and increasing/decreasing the cursor's displacement from the initial clicked point. The rate-based scrolling used a linear mapping between displacement and scroll-speed. To assure that scrolling speed was not capped by system behaviour, the maximum scrolling speed in the system was made to substantially exceed the limits of the smooth-pursuit system.

During each task the scroll speed was recorded every 100 milliseconds. At the conclusion of each task the time taken, magnification level and task type were also recorded. To estimate the participant's maximum comfortable scrolling rate the upper quartile (75% percentile) of the scrolling speeds were calculated for each task. The upper quartile (UQ) was used because it is the median of the upper half of the data set. Therefore, it is the scrolling rate that occurs most frequently in the upper range of the data. To verify this decision, we graphed scrolling rates for tasks completed by five test subjects and experimented to find the closest visual approximation to the required comfortable scrolling rate. Ideally, the required comfortable scrolling rate is the speed at which the user scrolls at for the longest period of time. Figure 4.5 illustrates a typical participant's scrolling rate and the approximated comfortable rate used in our evaluation.

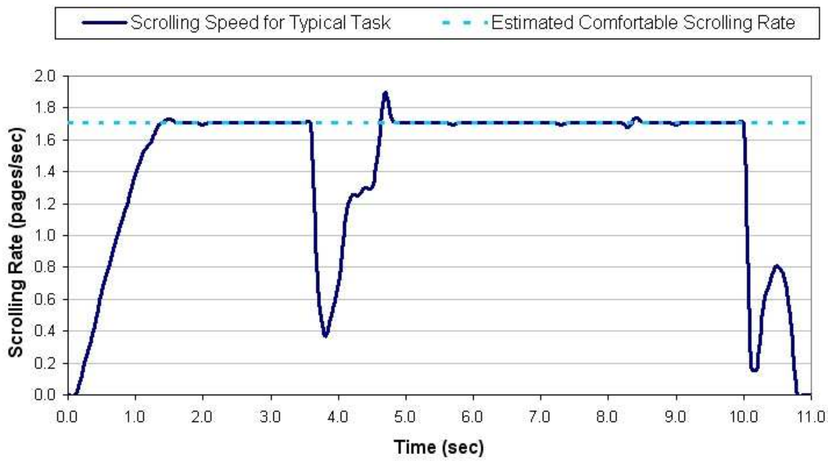



Figure 4.5: Typical scrolling rates observed during evaluation tasks. The dashed straight line overlaid on the observed scrolling speed represents the approximated comfortable scrolling rate for that task.

After each task, participants were encouraged to rest their eyes and look away from the screen. Initially, participants were forced to take at least 5 seconds rest after each task. This delay was removed after initial trials because subjects complained about being forced to wait and reported feeling no eye fatigue. Participants, however, were still required to wait one minute between changing interfaces to refresh their eyes.

In total each participant performed 32 tasks. One task was completed twice in the document interface for each target type, for each zoom level (detailed in Section 4.4.4). In the map and globe interfaces one task was completed three times for each zoom level. If the participant was not able to locate a target, after two minutes of searching the evaluation system recorded the ‘miss’ and automatically proceeded with the next task. To attempt to prevent ‘target missing’, four practise tasks were given for each interface and targets were placed almost directly in-line, from the initial location, with the given direction. We had planned to discard any result from a missed task. However, in the evaluation no results were discarded as no targets were missed.

At the beginning of the evaluation participants were asked to answer a short questionnaire to gather background demographics regarding age, gender, dominant hand and how often they played computer games. If the participant played more than occasionally, they were asked what type of games they generally played. We were particularly interested in experience with real-time games involving fast visual flow rates, such as ‘first person shooters’ like Doom and Quake, strategy games like Warcraft 3 and Starcraft, and car racing games like Need For Speed and Grand Theft Auto. We suspect participants that play games frequently will have faster comfortable scrolling rates than non-gamers. The results from the questionnaire are summarised in Table 4.2.

### *4.4.4 Experimental Design*

The dependent measure in the evaluation was document scrolling velocity, measured in pages/second (1 page = 27 cm) for the document browser, cm/second for the map browser and degrees/second for the globe browser. The map and globe interfaces varied one factor, magnification (as summarised below). The document interface varied two factors, magnification and target type. Target type was varied in the document interface because two tasks are performed in text documents, searching for a picture and searching for text.

| Category | Sub-category | Count |
|---|---|---|
| **Age** | 20-24 | 12 |
| | 25-29 | 7 |
| | 30-34 | 1 |
| **Gender** | Male | 14 |
| | Female | 6 |
| **Dominant Hand** | Left | 0 |
| | Right | 20 |
| **Game Play Frequency** | Never | 2 |
| | Occasionally | 9 |
| | Frequently | 7 |
| | Everyday | 2 |
| **Game Type Played Most Often** | None | 11 |
| | Turn-based | 2 |
| | Real-time | 7 |

Table 4.2: Summarised participant demographics taken before evaluation. Turn-based games were classified as games that require little hand-eye coordination and have low visual flow rates, such as, card games and computer board games. Real-time games were classified as games demanding a high level of coordination and involving fast visual flow rates, such as, first person shooters, strategy games, and car racing games.

- **Document Analysis —** 4x2 repeated-measures ANOVA for within subjects factors magnification and target type. The four levels for magnification were 100%, 75%, 50% and 25%. The two levels for target type were locate a picture and locate a text heading.

- **Map Analysis —** univariate ANOVA for the within subjects factor magnification. This factor had four levels: 100%, 50%, 25% and 12.5%.

- **Globe Analysis —** univariate ANOVA for the within subjects factor magnification. This factor had three levels: 100%, 75% and 50%.

Scrolling direction was not included as a factor in the evaluation because it was not suspected to influence results. However, we balanced the direction participants scrolled in to locate targets.

The order in which participants were exposed to the different interfaces was varied using a Latin square. To simplify the experimental design, the order of exposure to magnification and target type was alternated between participants.

## *4.5 Results*

### *4.5.1 Document Interface*

In total 320 tasks were completed by the 20 participants. The results were analysed using a 4x2 repeated analysis of variance (ANOVA). The mean comfortable scrolling rate over all magnifications was 2.64 pages per second with a standard deviation of 1.28 pages per second. Scrolling rates at different magnifications were significantly different ($F(3,57) = 77.24, p < 0.001$). As shown in Figure 4.6, as the magnification increases towards 100% the scrolling rate decreases. This means that SDAZ can allow faster navigation through a document by decreasing the magnification without affecting the user's comfort.

A regression analysis showed that the relationship between scrolling rate and magnification tends toward linear ($R^2 = 0.86$) relationship, with $scrollingRate = 0.7 \times Mag + 4.4$. Figure 4.6 displays the line of best fit.

There was no significant difference in participants' scrolling rates when locating a picture or a chapter heading. This finding is discussed later in Section

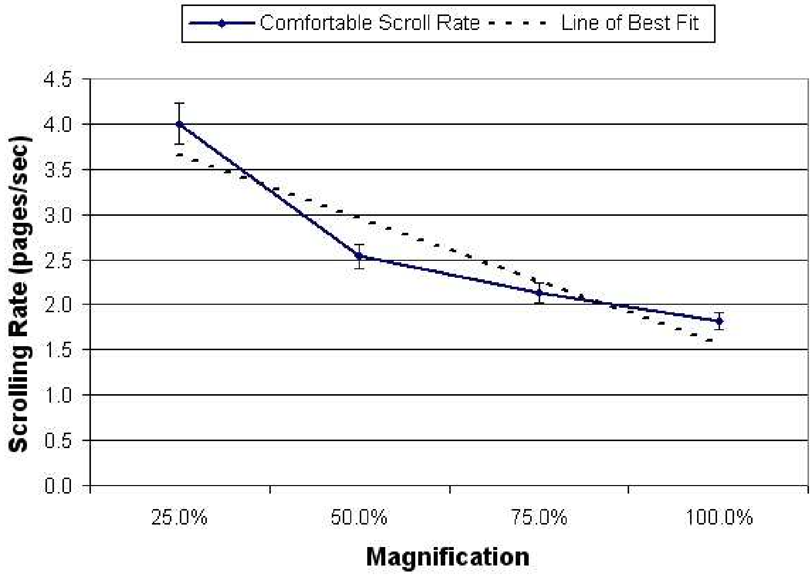


Figure 4.6: Mean comfortable scrolling rates (pages/cm) in the document viewer interface. The line of best fit is represented by a dotted line ($scrollingRate = 0.7 \times Mag + 4.4, R^2 = 0.864$). Note: 1 page = 27 cm.

4.6. There was also no interaction between target type and magnification level ($F = 0.831, p = 0.482$).

### *4.5.2 Map Interface*

The results from the 240 tasks completed in the map interface were analysed using a univariate ANOVA for four levels. The mean comfortable scrolling rate was 13.60 cm/sec with a standard deviation of 5.67 cm/sec. There was no significant difference in scrolling rate means at different magnifications ($F(3,57) = 1.55, p = 0.21$). This will be discussed later in Section 4.6.

The regression analysis gave a very low r-squared value ($R^2 = 0.019$) indicating that either the data is randomly distributed or participants' comfortable scrolling rates do not change as magnification decreases. As shown in Figure 4.7 the data points are relatively close to the horizontal line of best fit, suggesting scrolling rates do not change as magnification decreases.

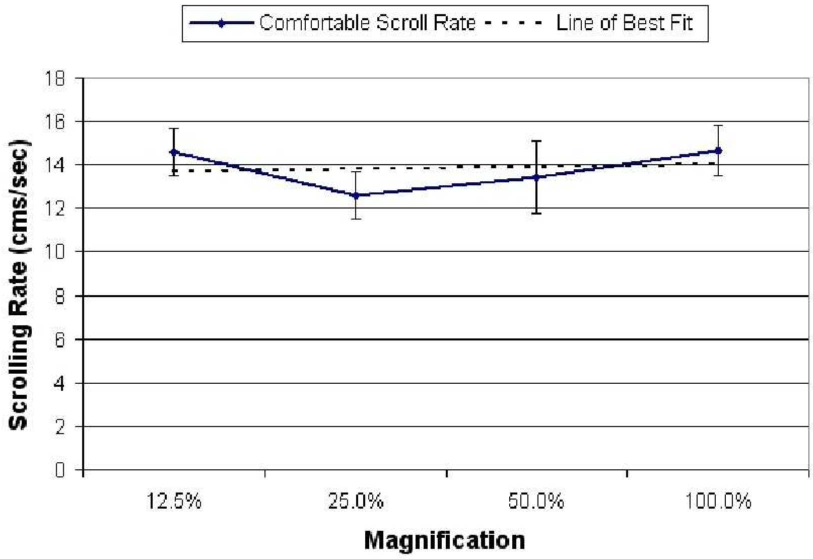



Figure 4.7: Mean comfortable scrolling rates (cm/sec) in the map viewer interface. The line of best fit is represented by a dotted line ($y = 0.1x + 13.6, R^2 = 0.019$).

### *4.5.3 Globe Interface*

The results from the 180 tasks completed in the globe interface were analysed using a univariate ANOVA for three levels. The mean comfortable scrolling rate was 24.59 degrees/sec with a standard deviation of 19.48 degrees/sec. The scrolling rate means at different magnifications differed significantly ($F(2,38) = 3.67, p < 0.05$).

Figure 4.5.3 shows a graph of participants' mean comfortable scrolling rates in the globe viewer interface.

## *4.6 Discussion*

The results from the document interface adhered to the theorised inversely proportional relationship between speed and zoom: as the magnification decreased, the participant's comfortable scrolling speed increased. This relationship is good news for SDAZ document browsers because it shows that the assumption made was correct; as magnification decreases the document's relative visual content on the screen decreases, empowering users to scroll comfortably faster. The docu-

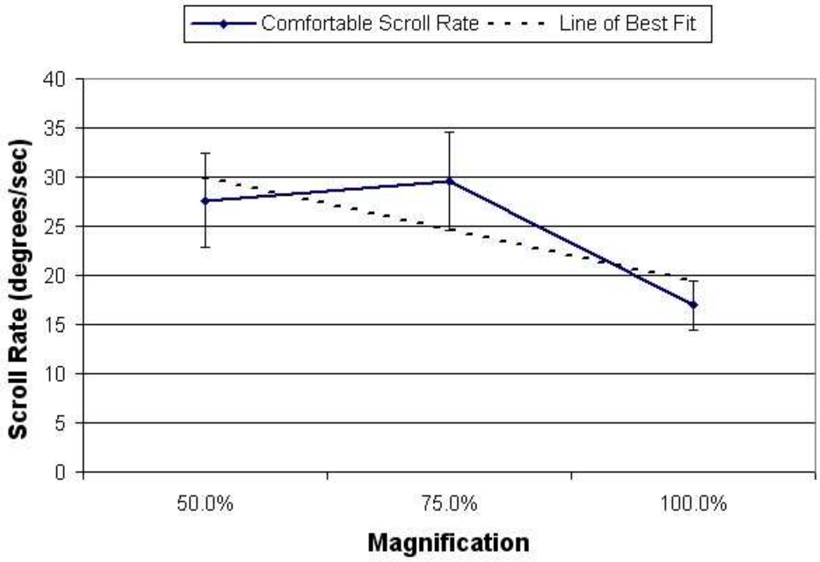



Figure 4.8: Mean comfortable scrolling rates (degrees) in the globe viewer interface. The line of best fit is represented by a dotted line ($scrollRate = -5.3 \times Mag + 35.3, R^2 = 0.613$).

ment viewer's inefficient use of screen space explains the difference between its results and the map and globe interface's — as the document shrinks black bars appear at the side.

There was no significant difference between the mean comfortable scrolling rates when locating a picture and locating a chapter heading. This allows for easy configuration of SDAZ systems because it removes the user's target as a configuration factor. The results, however, could be caused by the fact that the participants had not seen the picture before and were only given a description of the picture's composition, rather than given an icon of the picture. This meant that participants were unsure of the image's exact composition and size, possibly causing them to scroll slower to allow them time to compare the moving image with the described characteristics.

The results from the map interface were not as predicted. The relationship between magnification and comfortable scrolling rate, as shown in Figure 4.7, is approximately modelled by a horizontal line. We believe this unexpected relationship was caused by the relative visual information remaining constant between magnification levels; as the magnification decreases, fewer side streets can be seen, however, this is compensated for by the increased number of suburb names visible. This potentially causes a problem when calibrating the SDAZ map browser because the system automatically zooms out to reduce visual flow as scroll speed is increased. Section 4.8 discusses our recommendations for mappings for the map browser.

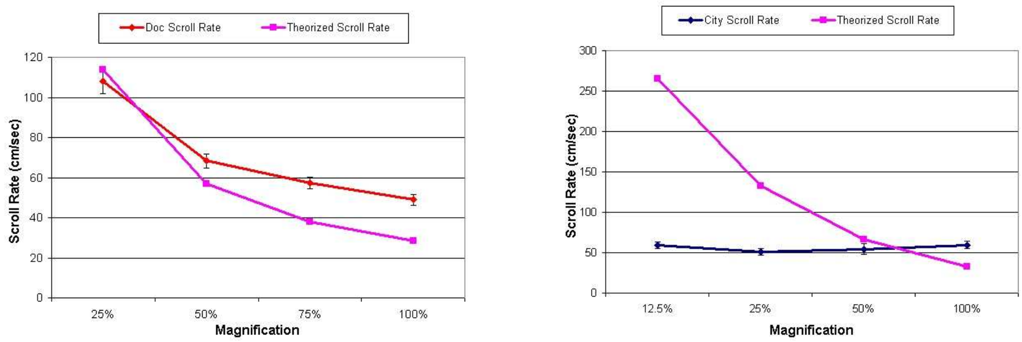
The results for the globe interface were disappointing because of the very high variance. We believe this is caused by our participants' varying knowledge of locations around the world — if they knew a particular place they would scroll there at a much faster rate. We tried to prevent this by requiring participants to search for unusual city names. Another factor that potentially increased variance was participants did not have to scroll as far to reach targets compared with the other interfaces. We also noticed that participants scrolled considerably faster East and West than North and South. The effects of direction are discussed further in Section 4.7.3.

In the globe viewer the scrolling rate increased between 100% to 75% magnification but decreased between 75% to 50%. This is most likely because at 50% magnification half of the Earth can be seen, so many city names need to be processed by the participant. Proposed relationships for the globe viewer's results are presented in Section 4.8.

### *4.6.1 Theorised vs Recorded Results*

The scrolling rates at each magnification level in the document interface were almost as theorised for three word text headings (see Figure 4.9(a)). The scrolling results observed for the map interface, however, were approximately half the theorised maximum scrolling rates for text headings. This is probably explained by the relative information content of a map compared to text document. A map is richer in information because it contains many place names located relatively close together, compared to a text document which contains fewer chapter headings placed in a sparse manner. Additionally, when zooming out in the document interface black gaps appear at the sides allowing users to scroll faster because fewer pixels are moving across the screen.

Scroll rates in the map interface compared to the document interface were approximately 10 cm/sec faster at a magnification of 100% (see Figure 4.9(b)). The scroll rates in the document interface, however, increase as the magnification decreases to a scroll rate of over 100 cm/sec (see Figure 4.9(a)).




(a) Recorded comfortable scroll rates in the document interface compared against theorised.

(b) Recorded comfortable scroll rates in the map interface compared against theorised.

Figure 4.9: Recorded comfortable scroll rates compared against theorised in the document and map interfaces.

## *4.7 Additional Analyses*

This section describes additional analyses studying the effects of gaming experience, gender, and direction. The reliability of the results is questionable because of the small sample sizes and unbalanced groups. However, the results are interesting and can serve as a base for future research.

### *4.7.1 Effects of Gaming Experience*

Data from the evaluations was analysed to see if gaming experience influenced participants' comfortable scrolling speeds. A gamer was classed as a participant who marked on the questionnaire that they frequently played real-time games, such as, first person shooters, strategy or racing games. Table 4.3 summarises the results showing that gamers have significantly faster comfortable scrolling rates in the document and map interfaces. The difference between means in the globe

| Interface | Mean (6 Gamers) | Mean (14 Non-gamers) | Unpaired T-Test |
|---|---|---|---|
| Document | 3.28 (1.25 SD) pages/sec | 2.34 (1.07 SD) pages/sec | $t(78) = 2.67$ $p < 0.01$** |
| Map | 66.18 (26.22 SD) cm/sec | 51.76 (20.20 SD) cm/sec | $t(78) = 2.67$ $p < 0.01$** |
| Globe | 31.44 (23.73 SD) degrees/sec | 21.83 (16.86 SD) degrees/sec | $t(58) = 1.78$ $p = 0.08$ |

Table 4.3: Comfortable scrolling rates by gaming experience. A significant difference between means is indicated with a '**'.

| Interface | Mean (8 Male) | Mean (6 Female) | T-Test Unpaired |
|---|---|---|---|
| Document | 2.60 (s.d. 1.13) pages/sec | 1.98 (s.d. 0.88) pages/sec | $t(54) = 2.22$ $p < 0.05$** |
| Map | 59.711 (s.d. 21.60) cm/sec | 41.16 (s.d. 11.90) cm/sec | $t(54) = 3.79$ $p < 0.001$** |
| Globe | 24.59 (s.d. 17.43) degrees/sec | 18.16 (s.d. 15.81) degrees/sec | $t(40) = 1.23$ $p = 0.23$ |

Table 4.4: Comfortable scrolling rates by gender for participant's with little to no gaming experience. A significant difference between means is indicated with a '**'.

interface was not significant because of the high variance. Unpaired two-tailed t-tests were used to test for statistical significance.

This result supports the hypothesis that gaming experience increases a user's comfortable scrolling speed. This is because games require the player to interact with images moving rapidly across the screen. If games are played frequently the player trains his/her eye to withstand the game's high visual flow.

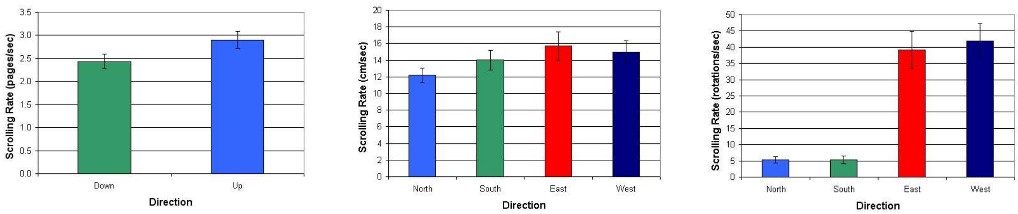
### *4.7.2 Effects of Gender*

Males have a significantly faster scrolling rate than females in the document and map interfaces (see Table 6.10). The difference between means in the globe interface was not significant because of the high variance. When analysing these results, as no females played games frequently, all gaming males were removed to avoid bias.

### 4.7.3 *Effects of Direction*

The effects of the scrolling direction in the document interface was analysed using a paired t-test. The results showed that the means for scrolling speeds by direction are significantly different ($F(1,19) = 8.226, p < 0.05$) with scrolling up being the faster (2.90 pages/sec, s.d. 0.84 pages/sec) than scrolling down (2.43 pages/sec, s.d. 0.71 pages/sec). Figure 4.10(a) displays a graph of the mean comfortable scrolling rates by direction in the document interface.

Typically, when scrolling up users know the structure of the document because they have already scrolled down it. We believe this familiarity encourages users to establish a habit of scrolling faster up than down in text documents.




(a) Mean comfortable scrolling rates by direction in the document viewer interface.

(b) Mean comfortable scrolling rates by direction in the map viewer interface.

(c) Mean comfortable scrolling rates by direction in the globe viewer interface.

Figure 4.10: Mean comfortable scrolling rates by direction for the three interfaces.

In the map interface the effects of direction were analysed using a 4 level repeated measures ANOVA. The results in the map interface showed that means for scrolling speeds by direction are significantly different ($F(3,57) = 3.664, p < 0.05$) with scrolling West being the fastest (15.70 cm/sec, s.d. 0.73 cm/sec) followed by East (14.98 cm/sec, s.d. 6.00 cm/sec), South (14.02 cm/sec, s.d. 5.34 cm/sec) and North (12.18 cm/sec, s.d. 4.01 cm/sec).

Post-hoc analysis using the Tukey Test showed no significant differences between any pair of directions in the map interface, however, this is not unusual due to the conservative nature of post-hoc analyses ($HSD = 3.02$). Figure 4.10(b) displays a graph of the mean comfortable scrolling rates by direction in the map interface.

The effects of direction in the globe interface were also analysed using a univariate ANOVA for four levels. The results in the globe interface showed that means for scrolling speeds by direction are significantly different ($F(3,57) = 41.449, p < 0.001$) with scrolling West being the fastest (41.90 degrees/sec, s.d. 23.51 degrees/sec) followed by East (39.15 degrees/sec, s.d. 25.32 degrees/sec), South (5.31 degrees/sec, s.d. 5.16 degrees/sec) and North (5.26 degrees/sec, s.d. 4.57 degrees/sec). Post-hoc analysis using the Tukey test showed significant differences between North and East, North and West, South and East, and South and West ($HSD = 11.87$). Figure 4.10(c) displays a graph of the mean comfortable scrolling rates by direction in the globe interface.

In the globe viewer significant differences in scroll speed were found between scrolling North and South, and scrolling East and West. This is most likely because vertical targets were usually a shorter distance away because the globe can not be flipped vertically. In addition, participants scrolling North or South mostly moved over land with marked cities, forcing the participant to scroll slower to process the city names. When moving East and West, however, around half of the scrolling was over water, allowing the participant to scroll much faster.

### *4.8 Automatic Zooming Recommendations*

Table 4.5 summarises comfortable rates from the visual flow evaluation and our recommended rates for use in SDAZ interfaces. Our final recommendations are based on the comfortable rates adjusted to make the automatic zooming feel more natural and usable. Linear interpolation was used to map magnification between defined scroll speeds. For example, when scrolling in the document interface at 1.95 pages/sec the document's magnification should be approximately 87% using the recommended settings (see first two rows in Table 4.5).

A linear mapping between cursor displacement and scroll speed was used with the recommended mappings. The maximum scrolling speeds in the interfaces were approximately set to the largest scroll speed in the automatic zooming mapping — 4.44 pages/sec in the document interface, 60 cm/sec in the map interface and 60 degrees/sec in the globe interface.

In the document interface the comfortable viewing rates obtained from the visual flow evaluation did not require much modification to provide smooth, natural automatic zooming with fast navigation. In the map interface, however, the

| Interface | Magnification | Comfortable | Recommended |
| --- | --- | --- | --- |
| Text | 100% | 1.67 pages/sec | 1.78 pages/sec |
| | 75% | 2.96 pages/sec | 2.11 pages/sec |
| | 50% | 2.33 pages/sec | 2.48 pages/sec |
| | 25% | 4.00 pages/sec | 4.00 pages/sec |
| Map | 100% | 59 cm/sec | 10 cm/sec |
| | 50% | 54 cm/sec | 30 cm/sec |
| | 25% | 51 cm/sec | 40 cm/sec |
| | 12.5% | 59 cm/sec | 50 cm/sec |
| Globe | 100% | 16 degrees/sec | 15 degrees/sec |
| | 75% | 29 degrees/sec | 35 degrees/sec |
| | 50% | 28 degrees/sec | 50 degrees/sec |

Table 4.5: Summarises comfortable viewing rates and our recommended rates for automatic zooming. Linear interpolation was used to map magnification between defined scroll speeds. Recommendations based on a linear mapping between cursor displacement and document scroll speed.

comfortable viewing rates were almost constant across magnification levels making the system feel unnatural and difficult to use. To increase the usability of the system the scroll speeds at higher magnifications were decreased to form a linear relationship similar to Cockburn & Savage's map system. However, the maximum rate at 12.5% was used to set the upper scroll speed limit. In the globe interface we recommend similar values to the comfortable rates. The scroll rates at 75% and 50% were increased because test users felt they were too slow and these increases gave a more natural linear interaction.

# Chapter V

# System Implementation

To ease our experiments with SDAZ calibration, we designed and implemented a new system that provides a comprehensive graphical user interface for customising SDAZ behaviour. The system was designed to simplify future extensions — for example new components such as interaction techniques and methods to render information can easily be added with little modification to existing code.

To allow easy calibration of the system, a SDAZ behavioural model was created encapsulating the core mappings and their constraints identified in Section 3.1. The implementation of this model and recommended settings are discussed in further detail in Section 5.1.

The new system was implemented in C++ because it allows new functionality to be easily added with the use of objects and proven design patterns, while still providing the high performance available through C. Further details of how the system was designed and implemented are given in Section 5.2.

## *5.1 SDAZ Behavioural Model*

The SDAZ behavioural model encapsulates the core mappings and their constraints identified in Section 3.1:

- Mapping between cursor displacement and scroll speed — describes the rate-based scrolling behaviour.

- Mapping between scroll speed and magnification — describes the automatic zooming behaviour.

- Maximum ascent rate — limits the maximum decrease in magnification per second. Prevents users from zooming out of the document too fast.

- Maximum descent rate — limits the maximum increase in magnification per second, while the user is scrolling. Reduces the 'slamming' effect experienced when changing direction.

- Maximum fall rate — limits the maximum increase in magnification per second, after the user has finished scrolling.

- Maximum scrolling acceleration — describes the maximum increase in scrolling speed per second. Depending on the configuration this constraint is optional.

In the model view the two mappings, displacement vs scroll speed and scroll speed vs magnification, are displayed using two graphs (see Figure 5.1). In the graphs users can move points up and down, add new points and delete existing ones. The four constraints, maximum scroll speed acceleration, maximum descent rate, fall rate and maximum ascent rate can be adjusted by clicking the up or down arrows next to the desired constraint's label or they can be turned off so the constraint has no effect.

The model GUI appears as a transparent layer above the document. This means users, while still in the model view, can scroll the document and test new settings on the 'fly'. Once the user has finished configuring the model, it can be saved to a file which can be loaded again at a later stage.

### *5.2 System Design and Implementation*

The new SDAZ system was designed in C++. This allowed new components to be easily added to the interface with the concept of objects. For example, new interfaces such as picture browser can be easily added without changing the code handling the scrolling technique. It also allowed users to toggle between interface types (text, map and globe) and scrolling techniques (SDAZ, rate-based scrolling with manual) using the same window.

The OpenGL library package was implementing the new system to render the interfaces because it can display high quality 3D scenes at high frame rates. The OpenIL library package was also used because it allows fast loading of images into OpenGL without the need to resize the images to a power of two which is

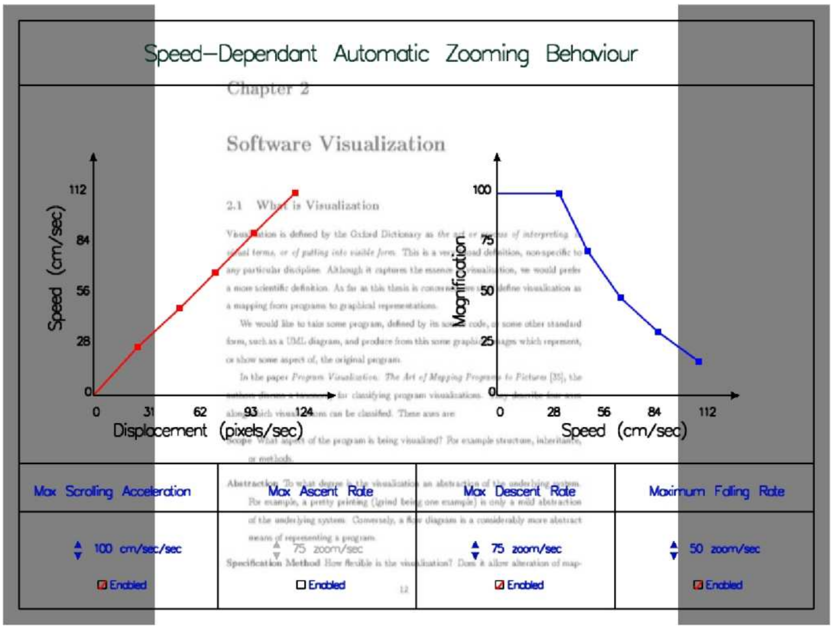



Figure 5.1: A screen shot of the GUI allowing easy calibration of SDAZ behaviour.

normally required by OpenGL (Woods 2000). OpenIL can also load many types of images into OpenGL including jpeg, paintshop, bmp and gif images.

Figure 5.2 shows a Class diagram of our implemented system. The function of the classes in the diagram are explained below:

- **ViewerApp** — This class first initialises the interface by making Document, Renderer and InteractionHandler objects. It then directs OpenGL's call-back methods to the appropriate object. All OpenGL's user interaction (mouse and keyboard) call-backs are directed to the Interaction Handler's appropriate function. The idle call-back function is directed to the Interaction Handler's idle function and the display call-back function is pointed to the Renderer's display function.

- **InteractionHandler** — This class provides the basic behaviour for interaction between the user and the document displayed.
  - **SDAZ** — This subclass provides SDAZ scrolling behaviour. It uses the SDAZ Model class to calculate the movement amount given the

mouse displacement. It then calls the Renderer's move function to reposition the document. The rezooming focus can be changed from centre to cursor using the toggle focus method.

    * **Globe SDAZ** — This subclass provides extra behaviour for SDAZ in the Globe Document so that users automatically rise and fall into cities marked on the globe.

  - **Rate and Manual Zoom** — This subclass provides rate-based scrolling behaviour with manual zooming.

    * **Globe Rate and Manual Zoom** — This subclass provides extra behaviour for rate-based scrolling and manual zooming in the Globe Document so that users can manually fall/rise into cities marked on the globe.

- **Document** — The Document class contains all the textures and relevant information to display a 1D document (such as a pdf) to the renderer. It uses OpenIL to load the textures into OpenGL from the image files.

  - **Map Document** — The Map Document is a subclass of Document that contains all the textures and relevant information to display a 2D map. For example, how many grid squares vertically and horizontally the map contains and possibly information about street locations.

  - **Globe Document** — The Globe Document is a subclass of Document that contains the information involved with storing a 3D globe with city maps.

- **Renderer** — This class provides a skeleton for all Renderer subclasses. The Renderer class contains a Document and a list of HUDs to be rendered to the screen by the display method. It also provides a move method that re-positions the document in the 3D scene.

  - **Doc Renderer** — This subclass is used to render basic 1D documents to the screen.

  - **Map Renderer** — This subclass is used to render 2D maps to the screen.

- **Globe Renderer** — This subclass is used to render a 3D globe with city maps to the screen.

- **SDAZ Model** — This class contains the current behavioural model for SDAZ (see Section 5.1). It is used by the SDAZ interaction handlers to calculate the size of the next movement based on the current behavioural model, current displacement and previous movements.

- **HUD** — The HUD class defines the displayHUD method used by the current Renderer to display the heads up display (HUD). We have implemented four types of heads up displays:

  - **Scroll HUD** — This HUD renders basic scrolling information to the screen. In our implementation it is used in the Map and Globe Renderers to show a line indicating the mouse displacement and direction.

    * **Doc Scroll HUD** — This Scroll HUD is only used in the Document Renderer because it provides a slightly different representation of the scrolling information.

  - **Info HUD** — This HUD renders strings to the top right of the screen. This was typically used to render statistics such as the current number of frames per second, the current scrolling technique, the scrolling speed and the current document's magnification.

  - **RezoomHUD** — This HUD is used to render the position zoomed to after the user releases the mouse button while scrolling.

  - **ModelHUD** — This HUD is used to render the current model to the screen as shown in Figure 5.1.

In the design of our interface we used several design patterns to increase reusability and the understandability of our design structure. Design patterns are proven solutions to frequently occurring problems in software design (Gamma et al. 1994).

The major design pattern used in our system was the model, view, controller pattern. This pattern attempts to separate the system into three main areas: the data to be presented (the model), the presentation of the data (the view) and the management over both (the controller). In our implementation the Document class represents the model because it holds information about the document being rendered. The Renderer class represents the view because it displays the document to the screen. Finally, the ViewerApp is the controller because it manages both these components.

The second design pattern used in this system was the strategy pattern (Gamma et al. 1994). The intent of this design pattern is to allow the interchanging of a family of classes by identifying the family's key components. A parent class is created composed of the family's key components. A variable is then created of the parent's type, which can be instantiated to any of the family's children. The child's implementation is then transparent to its use, meaning new children can be added and used with virtually no modification required to any other classes.

The strategy pattern was used to allow different types of renderers, documents, interaction handlers and HUDs to be easily interchangeable. For example, a new interaction handler can be changed without the renderer or document subclass's knowledge. New types of renderers, documents, interaction handlers and HUDs can be added to the system with little effort. For example, if a developer wants to add pan and manual zoom navigation to the application they only need to overwrite the methods defined by the InteractionHandler class and point the interaction handler variable, in the ViewerApp, to a new instance of this subclass.

### *5.3 Implemented Interfaces*

We implemented three SDAZ interfaces: a text browser, a map browser and a multi-scale globe browser. The text and map browsers were similar to Cockburn & Savage's implementations and the globe browser was based on Cockburn et al.'s 3D globe browser. In our text interface, however, the placement in the document was not displayed and no zoom threshold lines were displayed to indicate the scroll speed required to begin zooming (see Section 3.2.3). In all our interfaces only a line was displayed indicating the size of the cursor's displacement and the direction of the movement. This line's maximum size was limited by a maximum displacement. The maximum displacements were 150, 200 and 250 pixels, respectively, for the text, map, and globe interfaces.

The implemented interfaces used the automatic zooming settings recommended

| Zoom Constraint | Text | Map | Globe | |
|---|---|---|---|---|
| | | | **Overview** | **City View** |
| *Max Ascent Rate (mag/sec)* | 150 | 150 | 100 | 175 |
| *Max Descent Rate (mag/sec)* | 40 | 30 | 50 | 30 |
| *Max Falling Rate (mag/sec)* | 175 | 175 | 100 | 150 |

Table 5.1: SDAZ model constraints used in the implemented interfaces. Values are measured in magnification per second.

in Section 4.8. To scroll, a linear mapping between cursor displacement and scroll speed was used. The maximum scrolling speeds in the interfaces were set to the largest scroll speed in the automatic zooming mapping — 120 cm/sec (27 cm = 1 page) in the document interface, 60 cm/sec in the map interface and 60 degrees/sec in the globe interface.

In Section 3.1.1 four model constraints for SDAZ were defined — maximum scrolling acceleration, ascent rate, descent rate and falling rate. The maximum scrolling acceleration describes the maximum increase in scrolling speed possible over time. The maximum ascent rate limits the maximum decrease in magnification preventing zooming occurring at a speed that is too high for the user. The maximum descent rate dictates the maximum magnification increase speed. The maximum falling rate describes the maximum falling speed when the user has finished scrolling and has released the mouse button. Table 6.3 summarises the values used in the implemented systems for the model constraints. These values were found using trial and error.

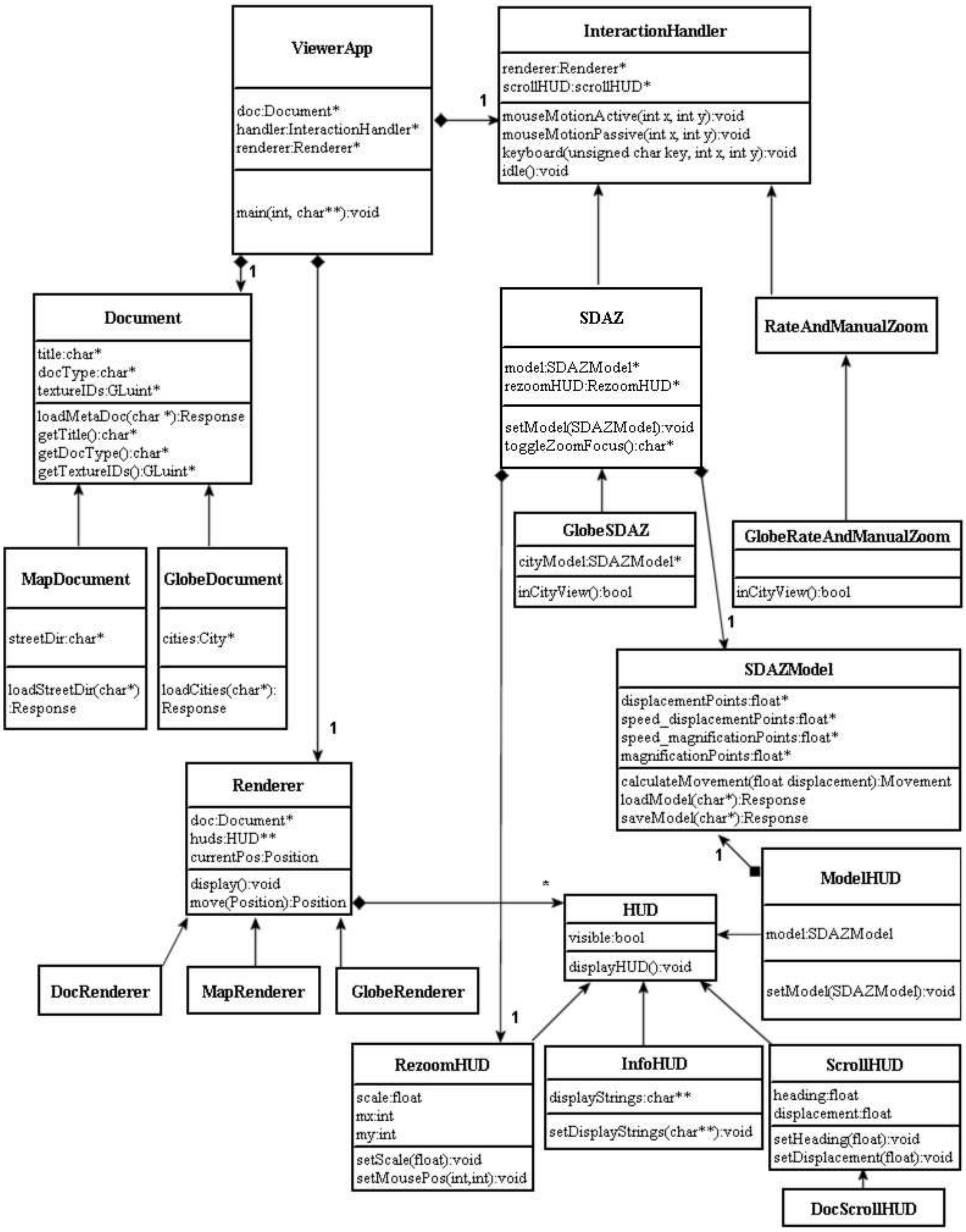



Figure 5.2: UML class diagram of system design.

# Chapter VI

# Performance Evaluation

Previous evaluations compared SDAZ with scroll-bars, panning and manual zooming (Igarashi & Hinckley 2000, Cockburn & Savage 2003). This evaluation compares SDAZ with rate-based scrolling and manual zooming because we suggest that SDAZ should serve to complement scroll-bars rather than replace them. SDAZ is essentially rate-based scrolling with the addition of automatic zooming, therefore if it is more efficient it should replace rate-based scrolling. SDAZ or rate-based scrolling is required when the exact location is not known and the user must search for the target. Scroll-bars are necessary when navigating a target at a known location because the user can simply drag the scroll handle to the target's location in the document. Hence, both types of techniques have an important role for users scrolling in documents too large for the display space.

The two primary objectives of this evaluation are to answer the following questions: Is SDAZ (rate-based scrolling and automatic zooming) significantly faster than rate-based scrolling and manual zooming, and do users prefer it? If results show that SDAZ is faster than and preferred to rate-based scrolling then major software vendors such as Microsoft and Adobe should seriously consider incorporating the functionality into their systems.

Another objective of this study is to determine whether Fitts' Law models the performance for off-screen targets, zooming target acquisition and multi-scale zooming target acquisition (in the globe browser). At present, only one study shows that Fitts' Law can model scrolling to off-screen targets in a text document (Hinckley et al. 2002). Target acquisition requiring zooming or multi-scale zooming has only been suggested to model Fitts', however, not proved (Guiard et al. 1999).

In this evaluation we compare SDAZ and rate-based scrolling with manual zooming in three different interfaces: a text document interface, a flat map interface, and a multi-scale globe interface. Section 6.1 presents the interfaces and the configuration of the scrolling techniques used in the evaluation. The evaluation details are presented in Sections 6.2–6.4. The results addressing the primary focus — whether SDAZ is faster and preferred by participants — is presented first in Section 6.5.1. The Fitts' analysis is then detailed in Section 6.5.2. NASA-TLX (task load index) results reporting subjective workload measurements are presented in Section 6.5.3. Additional analyses investigating the effect of gender and gaming experience on mean target acquisition times are presented in Section 6.5.4. This is followed by a final discussion of the evaluation results in Section 6.6.

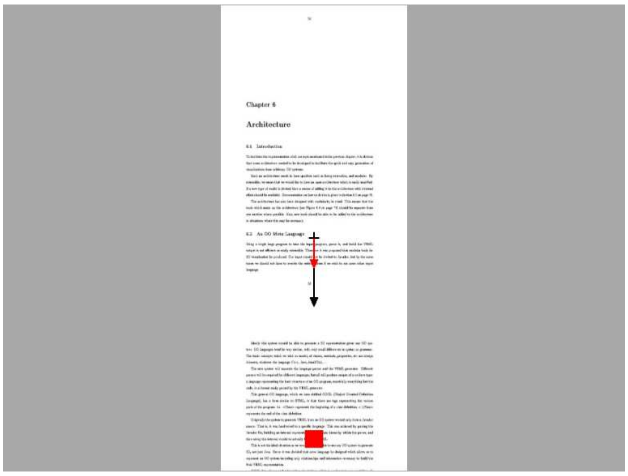
## *6.1 Interfaces*

The interfaces used in this evaluation were essentially the same as the interfaces described in Section 5.3 with one addition: a direction arrow was added to the interface so participants continually knew which direction the current target was located. Screen shots of the interfaces are presented in Figure 6.1.

The text document interface used a 157 page Master's thesis titled "Virtual 3D Worlds for Enhanced Software Visualisation" by Keown. One texture at a resolution of 512x512 pixels was created for each page. Each A4 page measured 21x27 cm on the screen. To increase performance manual culling was implemented so that only visible pages were rendered to the screen.

The flat map interface displayed a 5120x3072 pixel (360x216 cm on the screen) street map of Christchurch. The interface pieced the map together from 15 textures each with a resolution of 1024x1024 pixels. Initially, a resolution of 2048x2048 pixels was used for each section, however we found the frame rate dropped below minimum requirements for smooth performance (35 fps) because of the low performance video cards (Geforce 2 MX).

The globe interface used 200 textures, each at a resolution of 128x128 pixels, to form a globe of the Earth. Fifteen city street maps were overlaid on the globe. The cities with maps included: Auckland (New Zealand), Christchurch (New Zealand), Sydney (Australia), Singapore, Bangkok (Thailand), Beijing (China), Shanghai (China), New York (America), Washington (America), San Francisco (America), Seattle (America), Moscow (Russia), Douglas (Isle of Man), Kinshasa




(a) The document viewer.

(b) The map viewer.

(c) The globe viewer.

Figure 6.1: Interfaces used in the evaluation. Target direction is indicated by the short arrow stretching from the centre of the screen. Current cursor displacement is indicated by the long arrow also stretching from the centre of the screen.

(Congo), and Tokyo (Japan). Texture resolutions for the globe interface ranged from 512x512 to 2048x2048 pixels depending on the detail required for the map. On the globe at 100% resolution each city measured 2x1 cm.

### *6.1.1 Scrolling Techniques*

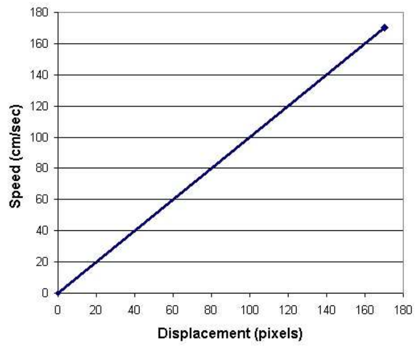
This section details the configuration of the evaluated scrolling techniques: SDAZ and rate-based scrolling with manual zooming. The configuration of the rate-based scrolling (cursor displacement to scroll speed mapping) was kept consistent for both techniques. The rate-based scrolling configuration is detailed first, followed by the automatic and manual zooming configurations.

#### *Rate-based Scrolling Configuration*

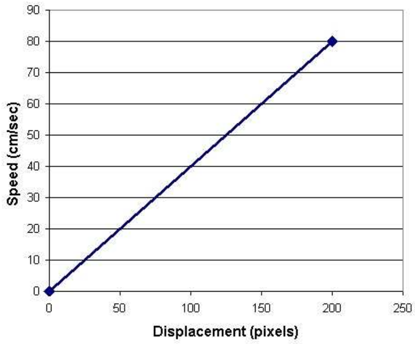
The text and map interfaces used a direct linear mapping between cursor displacement and speed (see Figures 6.2(a) and 6.2(b)). The maximum displacements were 150 pixels in the text document interface and 200 pixels in the map interface.

In the globe interface, the mapping between displacement and speed was adjusted from linear in the world overview to allow greater control when fully zoomed in. We found that, without the adjustment, users kept missing the city because they were scrolling too fast to fall in. To increase precision for fine movements we decreased the gradient of the relationship for the first 150 pixels of cursor displacement (see Figure 6.2(c)). The mapping between cursor displacement and speed in the city view was kept linear (see Figure 6.2(d)). The maximum displacements were 250 pixels in the global overview and 200 pixels in the city view.

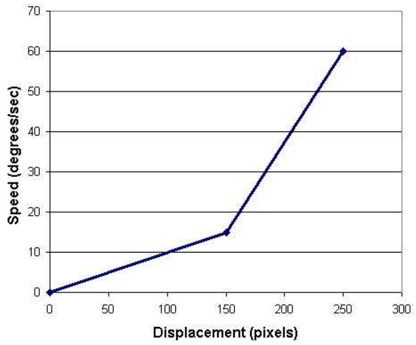
To aid target acquisition when manually zooming, the maximum scroll speed was limited by the magnification level. Without restricting the speed at each magnification level we found that participants would often struggle to place the centre cross over the target for selection. Table 6.1 contains the maximum scrolling speeds at maximum magnification and minimum magnification, and the maximum cursor displacement for each interface. Interpolation was used between discrete values.



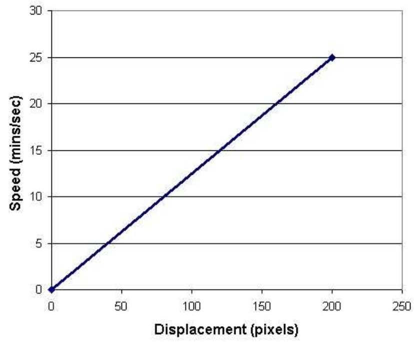

(a) Relationship between displacement and speed for the document interface.




(b) Relationship between displacement and speed for the map interface.




(c) Relationship between displacement and speed for overviews in the globe interface.




(d) Relationship between displacement and speed for city views in the globe interface.

Figure 6.2: Relationship between displacement and speed for each interface.

| **Interface** | **Max Mag** | **Max Speed** | **Min Mag** | **Max Speed** | **Displacement** |
|---|---|---|---|---|---|
| *Text* | 100% | 48 cm/sec | 12.5% | 170 cm/sec | 170 pixels |
| *Map* | 100% | 10 cm/sec | 12.5% | 80 cm/sec | 200 pixels |
| *Globe* | 100% | 15 degrees/sec | 45% | 60 degrees/sec | 250 pixels |
| *Globe City* | 100% | 5 mins/sec | 60% | 25 mins/sec | 200 pixels |

Table 6.1: Rate-based scrolling configuration. In the document interface 27 cm = 1 page. Scroll speeds are expressed in terms of the document on the screen at 100% magnification.

| Interface | Magnification | Recommended Scroll Speed | Evaluation Scroll Speed |
|---|---|---|---|
| *Text* | 100% | 48 cm/sec | 48 cm/sec |
| | 75% | 57 cm/sec | 57 cm/sec |
| | 50% | 67 cm/sec | 67 cm/sec |
| | 25% | 118 cm/sec | 100 cm/sec |
| | 12.5% | - | 118 cm/sec |
| *Map* | 100% | 10 cm/sec | 10 cm/sec |
| | 50% | 30 cm/sec | 30 cm/sec |
| | 25% | 40 cm/sec | 40 cm/sec |
| | 12.5% | 50 cm/sec | 50 cm/sec |
| *Globe* | 100% | 15 degrees/sec | 15 degrees/sec |
| | 75% | 35 degrees/sec | 35 degrees/sec |
| | 50% | 50 degrees/sec | 55 degrees/sec |
| | 45% | - | 60 degrees/sec |

Table 6.2: Recommended and evaluated automatic zooming configuration. In the document interface 27 cm = 1 page. Scroll speeds are expressed in terms of the document on the screen at 100% magnification.

*Automatic Zooming Configuration*

The configuration of SDAZ for the evaluation used the recommendations given in Section 5.3 with a few modifications. Table 6.2 summarises the recommended settings and the settings used in the evaluation.

In the text document interface the minimum magnification was decreased from 25% to 12.5% to give users a better overview of the document. The minimum magnification was still achieved when scrolling at 118 cm/sec.

The minimum magnification in the globe interface was decreased from 50% to 45%. The scroll speed at the minimum magnification was increased from 50 degrees/sec to 60 degrees/sec to make the automatic zooming feel more natural.

In all the interfaces SDAZ scrolled to the cursor after the mouse button was released. The simplest path to the cursor's position was used — retain scroll speed and current magnification until the cursor's position is reached then fall to full magnification (see Section 3.2.2 for further details).

The evaluation used the automatic zooming mapping constraints recommended in Section 5.3 (repeated in Table 6.3).

| Zoom Constraint | Text | Map | Globe | |
|---|---|---|---|---|
| | | | **Overview** | **City View** |
| *Max Ascent Rate (mag/sec)* | 150 | 150 | 100 | 175 |
| *Max Descent Rate (mag/sec)* | 40 | 30 | 50 | 30 |
| *Max Falling Rate (mag/sec)* | 175 | 175 | 100 | 150 |

Table 6.3: Evaluation automatic zooming constraints. Values are measured in magnifications per second.

*Manual Zooming Configuration*

In all interfaces manual zooming was performed using ‘a’ and ‘z’ on the keyboard. From maximum magnification (100%) to the minimum magnification or vice versa for the interface it required five taps of either ‘a’ or ‘z’. For example, in the map interface pressing ‘a’ or ‘z’ increased or decreased the magnification by 17.5%.

## *6.2 Apparatus*

All participants used identical Athlon 1600++ computers with 256Mb of RAM running Linux 9.0, and Geforce 2 MX video cards outputting to 19-inch Compaq monitors at a resolution of 1280 x 1024 pixels. The Compaq monitors had a viewable screen width and height of 36cm and 27cm, respectively. The OpenGL interfaces were executed in the full-screen game mode, hence the frames per second (fps) displayed was limited to the monitor refresh rate of 85 hz (85 fps).

Input was given through three-button Logitech mice with sample rates of 60Hz. The default RedHat 9.0 control-display gain settings were used: acceleration 2/1 pixels, threshold 4 pixels. This meant the cursor needed to move faster than 3.2 cm per second before the double speed mapping applied (on the 19” Compaq monitor).

Target sizes were kept constant for each interface. In the text document, map and globe interfaces the target sizes, at 100% magnification, measured 2x2 cms on the screen.

## 6.3 Method and Procedure

The tasks completed in the evaluation required participants to locate and select a fixed-size target, given the direction. The direction was indicated by a green arrow extending from the centre of the screen as shown in Figure 6.1. In the Globe interface the city containing the target was highlighted in red until the participant began falling into the city view, then it was shown as a red square within the city map.

The tasks in the evaluation represented real-world tasks requiring users to locate an area of interest for a detailed inspection. To complete a task, participants moved the red target to the centre of the screen (marked by a cross), zoomed to 100% magnification (at which point the target turned green indicating that it was selectable), and clicked the left button.

At the conclusion of each task, the target's distance, index of difficulty (IoD), and completion time were recorded for analysis.

The evaluation was held in a computer lab with 35 computer science students (30 male, 5 female) studying second year human-computer interaction. At the beginning of the evaluation a demonstration of the evaluation tasks using both scrolling techniques in all the interfaces was presented to the students. The participants then completed a questionnaire gathering background demographics regarding age, gender, dominant hand, and gaming experience. The results from the completed questionnaires are summarised in Table 6.4.

After the demonstration participants were introduced to the first interface using the first scrolling technique. Six practise tasks preceded the evaluation tasks with each scrolling technique for each interface. Participants completed a set of practise tasks and evaluated tasks for each scrolling technique for each interface.

After each set of evaluated tasks participants were asked by the system to complete a standard NASA-TLX worksheet (Hart & Staveland 1988) for the scrolling technique they had just used in the interface. The NASA-TLX worksheet asked the participants to rank the scrolling technique in six different categories of taskload (listed below) using a likert scale (1-low to 5-high). The participants were then asked to write which scrolling technique they preferred and why.

- **Mental Demand** — How much thinking was required when locating the targets with this scrolling technique?

| | | |
|---|---|---|
| **Age** | 19-24<br>25-29<br>30+ | 29<br>4<br>2 |
| **Gender** | Male<br>Female | 30<br>5 |
| **Dominant Hand** | Left<br>Right | 2<br>33 |
| **Game Time (per week)** | 0 hrs<br>1-2 hrs<br>2-5 hrs<br>5+ hrs | 9<br>10<br>7<br>9 |
| **Game Type Played Most Often** | None<br>Turn-based<br>Real-time | 9<br>1<br>25 |

Table 6.4: Summarised participant demographics taken before evaluation.

- **Physical Demand** — How much physical activity was required when using this scrolling technique (for example, button pushing, mouse moving, etc.)?
- **Temporal Demand** — How much time pressure did you feel under while performing the tasks with this scrolling technique?
- **Performance** — How well do you think you performed the tasks? For example, do you think you were able to scroll to the targets quickly?
- **Effort** — To accomplish your level of performance how hard did you have to work (both mentally and physically)?
- **Frustration level** — How discouraged, irritated, stressed or annoyed did you feel while completing the tasks?

### *6.4 Experimental Design*

The experiment analysis used a 2x6 repeated-measures ANOVA for within subjects factors scrolling technique and distance for each of the three interfaces: text document, map, and multi-scale globe. The two levels for scroll technique were

| Participant No. | Scrolling Technique Used First | Interface | | |
|---|---|---|---|---|
| | | First | Second | Third |
| 1 | SDAZ | Text | Map | Globe |
| 2 | Manual zoom | Map | Globe | Text |
| 3 | SDAZ | Globe | Text | Map |
| 4 | Manual zoom | Text | Map | Globe |
| 5 | SDAZ | Map | Globe | Text |
| 6 | Manual zoom | Globe | Text | Map |
| ... | ... | ... | ... | ... |

Table 6.5: Order of exposure to scrolling techniques and interfaces.

SDAZ and rate-based scrolling with manual zooming. The factor 'distance' determined how far the target was from the task's starting position. The six levels for distance are detailed below for each interface:

- Text interface: 5 pages, 10 pages, 15 pages, 20 pages, 25 pages, 30 pages (1 page = 27 cm on the screen at 100%)

- Map interface: 50 cm, 100 cm, 150 cm, 200 cm, 250 cm, 300 cm.

- Globe interface: 4 degrees, 8 degrees, 16 degrees, 32 degrees, 64 degrees, 128 degrees.

In the text and map interfaces, three evaluation tasks were completed for each distance with each scrolling technique. In the globe interface two evaluation tasks were completed for each distance with each scrolling technique. Before beginning the evaluation tasks six practise tasks were completed. This meant in total 48 tasks were completed in each of the text and map interfaces and 36 tasks were completed in the globe interface.

Two task sets containing target locations were created for each interface. The order in which participants were exposed to the interfaces and task sets was varied using a Latin square. Half of the participants used SDAZ first and half used rate-based scrolling first. The order of exposure to scrolling techniques and interfaces is illustrated in Table 6.5.

Results were studied using a standard Fitts' Law analysis. A standard Fitts' Law analysis plots the mean scrolling times against index of difficulty (IoD) for each scrolling technique. IoD is calculated using Equation 6.1. If the technique is modelled by Fitts' Law its results should produce a straight line. An index of performance (IoP) is then calculated from the inverse gradient of each technique's straight line. The IoP represents a standard measurement of the technique's effectiveness.

$$IoD = \log_2(\frac{A}{W} + 1) \tag{6.1}$$

Equation 6.1: Calculates the index of difficulty (IoD) for target acquisition tasks given 'A' the amplitude (distance) and 'W' the width of the target. MacKenzie & Buxton suggest setting 'W' to the smallest value of the width and height dimensions.

To prevent extreme outliers all acquisition times greater than the mean plus three standard deviations were removed from the results analysis.

## 6.5 Results

This section first presents the data collected from our evaluation addressing our study's primary research questions: is SDAZ significantly faster than rate-based scrolling and manual zooming, and do users prefer using it? The results from our Fitts' law analysis and subjective task load analysis (NASA-TLX) are detailed in Sections 6.5.2 and 6.5.3. Additional analyses into the effects of gaming experience and gender on target acquisition times are presented in Section 6.5.4.

### 6.5.1 SDAZ vs Rate-based and Manual Zooming Performance

The results from the evaluation are summarised in Table 6.6. To prevent extreme outliers all acquisition times greater than the mean plus three standard deviations were removed from the results analysis. This meant 4.5% of entries were removed in the text interface, 0.3% of entries in the map interface and 0.8% of entries in the globe interface.

The results show that, using SDAZ, targets were acquired 4% faster in the text interface ($F(1,34) = 5.72, p < 0.05$) and 10% in the map interface ($F(1,34) = 15.81, p < 0.001$). No significant difference between scrolling techniques was found in the globe interface. Hypothetical reasons for this are discussed in Section

| **Interface** | **Technique** | **Mean Time** | **S.D** | **F(1,34)** | **p** |
|---|---|---|---|---|---|
| *Text* | SDAZ<br>Manual Zooming | 6.78 sec<br>7.09 sec | 1.97 sec<br>1.95 sec | 5.72 | $< 0.05*$ |
| *Map* | SDAZ<br>Manual Zooming | 5.56 sec<br>6.21 sec | 1.73 sec<br>2.04 sec | 15.81 | $< 0.001*$ |
| *Globe* | SDAZ<br>Manual Zooming | 10.92 sec<br>11.05 sec | 2.21 sec<br>2.32 | 0.18 | 0.67 |

Table 6.6: Summarised evaluation results for SDAZ and manual zooming in the text, map and globe interfaces.

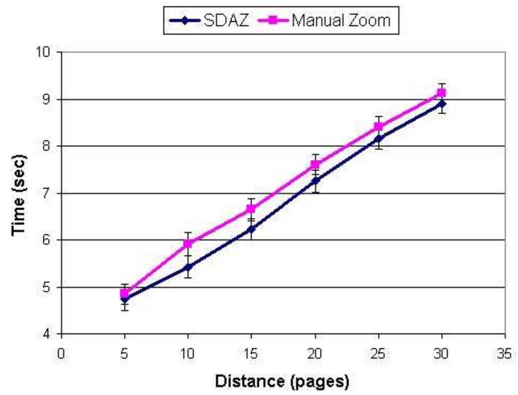
6.6.

Figure 6.3 shows mean times against distance for each scrolling technique in each interface. Significant interactions between scrolling technique and distance were found for all interfaces (text interface $F(5,170) = 6.06, p < 0.001$; map interface $F(5,170) = 99.79, p < 0.001$; globe interface $F(5,170) = 3.03, p < 0.05$).

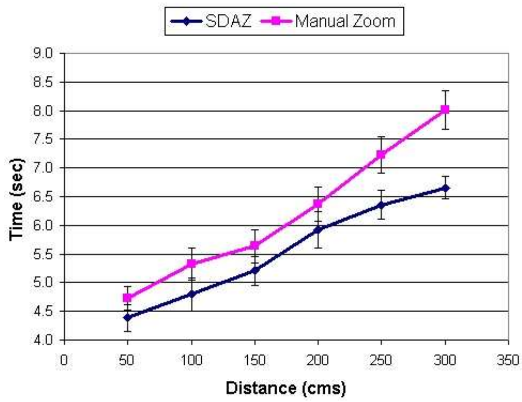
A significant interaction was found between scrolling technique and distance for the map interface ($F(5,170) = 5.43, p < 0.05$). This interaction is shown in Figure 6.3(b), as the distance increases mean acquisition time for manual zooming increases at a faster rate than SDAZ.

In the globe interface, participants frequently overshot the target city when the distance was only four degrees (see Figure 6.3(c)). This was because participants would zoom out of the last targeted city and then scroll as fast as they can in global overview in the direction of the next target. As the city was closer than the other target cities participants scrolled past it, accounting for the increased acquisition time.

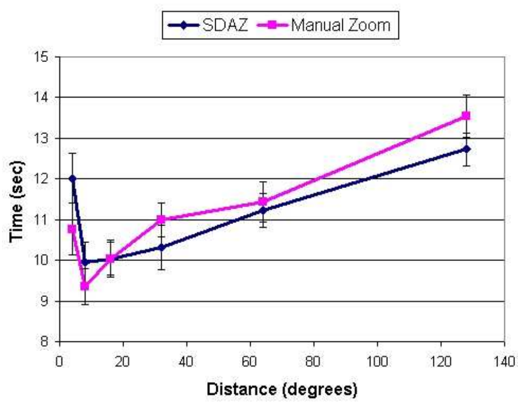
In the text and map interfaces, more participants preferred using SDAZ over rate-based scrolling with manual zooming (Chi-square test $df = 1$, $\chi^2 = 9.26$, $p < 0.01$ text interface, $\chi^2 = 7.31$, $p < 0.01$ map interface). In the globe interface more participants preferred using SDAZ, however the chi-square test failed to show significance ($df = 1$, $\chi^2 = 2.86$, $p < 0.01$). Figure 6.4 displays a graph of participant preferences of scrolling techniques.




(a) Mean target acquisition times against distance for SDAZ and rate-based scrolling with manual zooming in the text interface.




(b) Mean target acquisition times against distance for SDAZ and rate-based scrolling with manual zooming in the map interface.




(c) Mean target acquisition times against distance for SDAZ and rate-based scrolling with manual zooming in the multi-scale globe interface.

Figure 6.3: Mean target acquisition times against distance for SDAZ and rate-based scrolling with manual zooming in the evaluated interfaces.

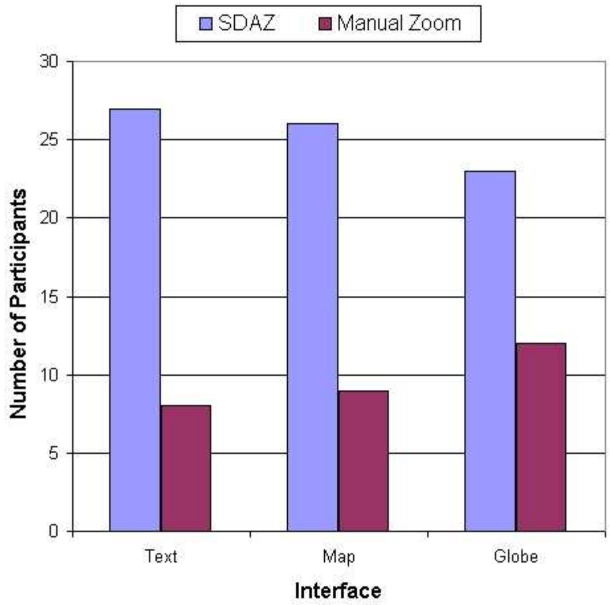



Figure 6.4: Participant preferences for scrolling technique for each interface.

### 6.5.2 *Fitts' Law Analysis*

Table 6.7 shows the equations of the lines of best fit and the indexes of performance (IoP) for each scrolling technique in each interface. The high $R^2$ values ($> 0.8$) indicating that the data is well correlated to the equation are typical with Fitts' Law analyses. As shown in Table 6.7, SDAZ has higher IoP values, supporting the efficiency findings of the ANOVA results.

The regression analyses show that Fitts' Law models the performance for off-screen target acquisition and zooming target acquisition for both scrolling techniques in the text and map interfaces (see Table 6.7).

The multi-scale globe interface was modelled by Fitts' Law when targets at a distance of four degrees (IoD of 7 bits) were removed. The unusual high acquisition times for the short distance tasks were discussed in Section 6.5.1.

### 6.5.3 *Subjective Workload Measures — NASA-TLX*

The NASA-TLX subjective results are shown in Table 6.8. All areas excluding temporal demand in the text interface and frustration in the globe interface, were

| Interface | Technique | Equation | $R^2$ | IoP |
|---|---|---|---|---|
| *Text* | SDAZ | $MT = 1.62 * IoD - 5.82$ | 0.92 | 0.62 |
| | Manual Zooming | $MT = 1.64 * IoD - 5.89$ | 0.95 | 0.61 |
| *Map* | SDAZ | $MT = 0.92 * IoD - 0.21$ | 0.93 | 1.09 |
| | Manual Zooming | $MT = 1.22 * IoD - 1.44$ | 0.87 | 0.82 |
| *Globe* | SDAZ | $MT = 0.22 * IoD + 8.96$ | 0.13 | — |
| | Manual Zooming | $MT = 0.60 * IoD + 5.28$ | 0.62 | — |
| *Globe* | SDAZ | $MT = 0.67 * IoD + 4.11$ | 0.84 | 1.49 |
| *(7 bit tasks removed)* | Manual Zooming | $MT = 0.98 * IoD + 1.27$ | 0.93 | 1.02 |

Table 6.7: Equations for lines of best fit and index of performance ratings. Index of Performance (IoP) is measured in bits per second.

significantly lower using SDAZ than rate-based scrolling and manual zooming (Wilcoxon's match-pairs tests).

### *6.5.4 Additional Analyses*

This section describes additional analyses studying the effects of gaming experience and gender. The reliability of the results is questionable because of the small sample sizes and unbalanced groups. However, the results are interesting and can serve as a base for future research.

#### *Gaming Experience*

To analyse the effect of gaming experience we used a 2x2 mixed design ANOVA for factors scrolling technique (SDAZ, manual zoom) and gaming experience (gamer, non-gamer). Participants were separated into two groups, those who played games for over two hours per week and those who did not, giving 16 gamers and 17 non-gamers. The results from the analysis are shown in Table 6.9.

Mean target acquisition times were significantly faster for the text and map interfaces for gamers ($F(1,33) = 19.55, p < 0.05$ text interface, $F(1,33) = 10.01, p < 0.05$ map interface). No significant difference for gaming experience was found in the globe interface ($F(1,33) = 0.18, p = 0.68$).

No significant interactions were found between scrolling technique and gam-

| Interface | Category | Technique Mean (SD) | | Wilcoxon Values | |
|---|---|---|---|---|---|
| | | SDAZ | Manual Zoom | z | p |
| Text | Mental Demand | 2.3 (1.1) | 2.9 (1.0) | 3.32 | $< 0.05*$ |
| | Physical Demand | 2.2 (0.8) | 3.7 (0.8) | 4.55 | $< 0.05*$ |
| | Temporal Demand | 2.6 (0.9) | 2.8 (1.1) | 1.19 | $= 0.12$ |
| | Performance | 2.0 (1.1) | 2.3 (0.9) | 1.7 | $< 0.05*$ |
| | Effort | 2.6 (0.9) | 3.3 (0.9) | 3.21 | $< 0.05*$ |
| | Frustration | 2.3 (1.0) | 2.7 (1.2) | 1.99 | $< 0.05*$ |
| Map | Mental Demand | 2.3 (1.0) | 2.9 (1.0) | 2.87 | $< 0.05*$ |
| | Physical Demand | 2.1 (0.9) | 3.6 (1.0) | 4.89 | $< 0.05*$ |
| | Temporal Demand | 2.5 (1.0) | 2.9 (0.9) | 2.49 | $< 0.05*$ |
| | Performance | 2.0 (0.9) | 2.5 (0.9) | 2.27 | $< 0.05*$ |
| | Effort | 2.5 (1.0) | 3.1 (1.1) | 2.64 | $< 0.05*$ |
| | Frustration | 2.5 (1.1) | 2.8 (1.0) | 1.84 | $< 0.05*$ |
| Globe | Mental Demand | 2.7 (1.0) | 3.2 (0.9) | 2.30 | $< 0.05*$ |
| | Physical Demand | 2.6 (0.9) | 3.9 (0.9) | 4.27 | $< 0.05*$ |
| | Temporal Demand | 2.6 (1.0) | 3.0 (1.0) | 1.81 | $< 0.05*$ |
| | Performance | 2.1 (0.9) | 2.6 (0.8) | 2.17 | $< 0.05*$ |
| | Effort | 2.9 (0.9) | 3.4 (0.9) | 2.01 | $< 0.05*$ |
| | Frustration | 2.6 (1.0) | 3.4 (0.8) | 1.20 | $= 0.12$ |

Table 6.8: NASA-TLX subjective evaluation results. The Wilcoxon's statistical match-pairs test was used to test for significance.

| Interface | Gaming Experience | Mean (SD) | | |
|---|---|---|---|---|
| | | SDAZ | Manual Zoom | Average |
| Text | Gamer | 6.08 (1.80) | 6.98 (1.78) | 6.23 (1.79) |
| | Non-Gamer | 7.38 (1.95) | 7.68 (1.90) | 7.53 (1.93) |
| Map | Gamer | 5.00 (1.31) | 5.34 (1.39) | 5.17 (1.35) |
| | Non-Gamer | 6.02 (1.91) | 6.95 (2.20) | 6.48 (2.11) |
| Globe | Gamer | 9.77 (1.20) | 9.91 (2.11) | 9.84 (1.69) |
| | Non-Gamer | 11.89 (2.43) | 12.02 (2.07) | 11.95 (2.23) |

Table 6.9: Results from the analysis of gaming experience. Mean and standard deviation is measured in seconds.

| Interface | Gender | Mean (SD) | | |
|---|---|---|---|---|
| | | **SDAZ** | **Manual Zoom** | **Average** |
| Text | Male | 7.28 (1.90) | 7.43 (1.85) | 7.35 (1.87) |
| | Female | 7.66 (2.09) | 8.40 (1.89) | 8.02 (2.01) |
| Map | Male | 5.47 (1.31) | 6.35 (1.46) | 5.91 (1.45) |
| | Female | 7.56 (2.44) | 8.64 (2.98) | 8.10 (2.75) |
| Globe | Male | 10.94 (1.92) | 11.74 (2.10) | 11.34 (2.02) |
| | Female | 14.56 (1.58) | 12.81 (1.97) | 13.68 (1.92) |

Table 6.10: Results from the analysis of gender. Mean and standard deviation is measured in seconds.

ing experience for all interfaces (text interface $F(1,33) = 0, p = 0.97$; map interface $F(1,33) = 3.43, p = 0.07$; globe interface $F(1,33) = 0, p = 1.0$).

*Gender*

To compare gender we used a 2x2 mixed design ANOVA for factors scrolling technique (SDAZ, manual zoom) and gender (male, females). All male participants classed as gamers in the previous section were removed to avoid bias as no females played games more than two hours a week. This resulted in data for 9 males and 5 females. The results from the analysis are shown in Table 6.10.

Mean target acquisition times were significantly faster for males in the map interface ($F(1,17) = 11.57, p < 0.05$). No significant difference for gender was found in the text and globe interfaces ($F(1,17) = 1.69, p = 0.21$ text interface, $F(1,17) = 0.11, p = 0.75$ globe interface).

A significant interaction was found between scrolling technique and gender for the globe interface ($F(1,17) = 7.76, p < 0.05$). Female times decreased using rate-based scrolling with manual zooming, while males times increased.

No significant interactions were found between scrolling technique and gender for the text and map interfaces (text interface $F(1,17) = 2.39, p = 0.14$; map interface $F(1,17) = 0.14, p = 0.72$).

## 6.6 Discussion

### 6.6.1 Performance Summary

The results from our evaluation show that SDAZ is 10% faster acquiring targets in the map interface and 4% faster in the text interface. It is also preferred by most participants in both interfaces, commenting that they found SDAZ very easy to use and felt less "busy" with the technique as they had fewer controls to manage. They also commented that they found the red box indicating the zoom in position very helpful. NASA-TLX ratings showed that participants felt less drained both mentally and physically using SDAZ because they did not have to concentrate or physically control the zoom level. The NASA-TLX ratings also showed that users perceived they had performed better and felt less frustrated using the technique in the text and map interfaces.

In the globe interface no difference was shown between scrolling techniques for acquisition time or preference. Some participants commented that they felt they had less control using SDAZ in the globe interface and that is why they preferred to manually zoom. A couple of participants remarked that they felt trapped in the city map as it took them a while to scroll out, indicating a more efficient method needs to be used between the detailed map and the global overview. NASA-TLX results show that participants felt that they performed better using SDAZ. They also felt that less physical and mental effort was required. However, no difference was found between frustration ratings using each technique.

The Fitts' Law analysis showed that target acquisition using both scrolling techniques in all interfaces is modelled by Fitts'. This shows that the performance for off-screen target acquisition, zooming target acquisition and multi-scale zooming target acquisition can be also be explained using Fitts' robust model.

The additional analyses on the effects of gaming experience show, not surprisingly, that gamers have faster target acquisition times in the text and map interfaces. This is due to games increasing co-ordination and tolerable visual flow rates.

A major problem highlighted by many participants using SDAZ in the evaluation was target overshooting. Participants commented that they could get to the target's location quickly but they found it very difficult to move the centre of the screen over the target. One participant summarised their hunting experience remarking that they "played ping-pong over the final target". We believe the behaviour of SDAZ causing participants to overshoot was automatically scrolling to the cursor's position once the mouse button was released. Participants centre the target on the screen, then release the mouse button causing them to move away from the target's position. This strongly suggests that scrolling to the cursor's location should only be performed when the magnification is reduced. This means that SDAZ behaves exactly like rate-based scrolling when scrolling slower then zooming threshold (which determines the speed a user must scroll before automatic zooming is initiated).

Another potential factor causing overshooting is a low zooming threshold. If the threshold is too low the automatic zooming feels unnatural and difficult to control when fine precision movements are required. Reducing the magnification causes the target to suddenly appear closer to the centre of the screen; users then in a hurry release the mouse button which scrolls them past the target.

### *6.6.2 Experimental Concerns*

The evaluation contained non-ecological tasks requiring participants to locate red squares continually given the direction. The validity of using non-ecological tasks is questionable, however when designing these tasks we attempted to represent real-world tasks requiring users to locate an area of interest for a detailed inspection.

In the performance evaluation manual zooming was performed using the keyboard. To prove SDAZ is more efficient than manual zooming it needs to be compared with alternative types of manual zooming. Alternative techniques include zooming with the scroll wheel and gesture zooming (further discussion in Section 7.1).

Another potential confound is that the evaluation compared rate-based scrolling with manual zooming at its most efficient — many users browsing documents do not use the manual zoom facility. Hence, our evaluation may have shown artificially positive results for rate-based scrolling with manual zooming.

# Chapter VII

# Future Work and Conclusions

## *7.1 Future Work*

The following subsections discuss possible further work with SDAZ.

### *7.1.1 Ecological Validity*

The performance evaluation compared SDAZ against rate-based scrolling with manual zooming for semi-artificial tasks. Further research needs to conduct an evaluation comparing the same techniques for ecological based tasks similar to those used in Cockburn & Savage's evaluation.

### *7.1.2 Final Acquisition*

Our performance evaluation highlighted the need to improve final acquisition of the target once the user moves to the target's location. At present, users can move large distances quickly and easily, however the high precision movement required when moving the target into the centre of the screen is very difficult. We believe that most of the difficulty of high precision movements can be alleviated using SDAZ by zooming to the cursor only when the magnification is decreased.

### *7.1.3 Manual Zooming*

In the performance evaluation, manual zooming was performed using the keyboard. This could have accounted for the comparative efficiency of SDAZ. To prove that SDAZ is more efficient, it needs to be compared with other types of manual zooming. In many interfaces the scroll wheel is used instead of the keyboard to adjust the zoom level. When using this technique, however, it is difficult to scroll and zoom in parallel (if not using modal interaction), because the user needs to hold down the mouse button and roll the scroll wheel at the same time.

Gesture zooming is another alternative manual zooming technique for 1D documents. Patel et al. (2004) first proposed gesture zooming as a method to vertically browse a collection of photos. In the gesture zooming interface scroll speed is proportional to vertical cursor displacement (1D rate-based scrolling) and zooming is proportional to horizontal cursor displacement. A larger horizontal displacement results in a higher zoom level. Patel et al. compared gesture zooming against SDAZ and discrete zooming interfaces. They found that the SDAZ and gesture zooming interfaces support faster navigation, higher accuracy and have lower subjective task load levels than the standard discrete zooming interface. However, no significant difference was found between SDAZ and gesture zoom interfaces.

We suggest gesture zooming is well suited to browsing text documents because usually only vertical movement is required. An evaluation needs be conducted comparing gesture zooming with SDAZ for browsing text documents.

### *7.1.4 Rate-based Scrolling Mapping for SDAZ*

This thesis used a linear relationship between cursor displacement and scroll speed, however, this is not necessarily the most efficient. Further research needs to evaluate different displacement-speed mappings for efficiency and user preference. Fitts' IoP values could be used in the evaluations to give standard comparable measures independent of experiment details.

### *7.1.5 SDAZ Rezoom Position*

Formal evaluations need to be conducted studying the efficiency of different rezooming positions (see Section 3.2.2 for details). Currently, our systems can zoom to centre or zoom to the cursor's position after the user has finished scrolling, however we evaluated only zooming to the cursor's position, because users indicated that this seemed more intuitive. Also, different paths for scrolling to the cursor's position, such as van Wijk & Nuij defined optimal, need to be evaluated.

### 7.1.6 3D Globe Browser

At present, in the 3D globe browser users need to scroll outside the city boundaries to automatically zoom to the globe overview. This is inefficient and many participants commented during the performance evaluation that they found the feature frustrating. Future work in the globe interface should focus on more efficient methods to move between the different scales.

## 7.2 Conclusions

Speed-Dependent Automatic Zooming (SDAZ) is a new technique designed to efficiently navigate through documents. It couples rate-based scrolling with automatic zooming. The idea is to automatically zoom the document as the scroll speed increases, keeping visual flow across the screen constant. This allows users to efficiently navigate without having to manually switch between zooming and scrolling, and without becoming disoriented by fast visual flow.

The efficiency and user preference of SDAZ relies upon its calibration. To aid calibration, Chapter 3 identified the low-level components and parameters of SDAZ. To form base calibration settings, a visual flow evaluation was conducted recording participants' maximum comfortable scrolling rates at different magnifications (see Chapter 4).

Chapter 5 described a new SDAZ system providing a comprehensive graphical user interface for customising SDAZ behaviour. The system was designed to simplify future extensions — for example new components such as interaction techniques and methods to render information can easily be added with little modification to existing code. This system was used to configure (based on the settings established in Chapter 4) three SDAZ interfaces: a text document browser, a flat map browser and a multi-scale globe browser.

The three calibrated SDAZ interfaces were evaluated against three equivalent interfaces with rate-based scrolling and manual zooming interfaces (see Chapter 6). The evaluation showed that SDAZ is 10% faster for acquiring targets in a map than rate-based scrolling with manual zooming, and SDAZ is 4% faster for acquiring targets in a text document. Participants also preferred using automatic zooming over manual zooming. No difference was found for the globe browser for acquisition time or preference. However, in all the interfaces participants commented that automatic zooming was less physically and mentally draining than manual zooming.

The evaluation in Chapter 6 also showed that Fitts' Law models the performance for off-screen targets, zooming target acquisition and multi-scale zooming target acquisition. Until now, target acquisition requiring zooming or multi-scale zooming has only been suggested to model Fitts' Law, however, was not proven (Guiard et al. 1999).

In conclusion, our results show that SDAZ is significantly faster and preferred over rate-based scrolling with manual zooming. This indicates that major software vendors such as Microsoft and Adobe should seriously consider incorporating the functionality of SDAZ into their systems.

Acknowledgments

The author would like to thank:

- **Andy Cockburn** for his support, guidance and good sense of humour without which this Masters would not have been possible.
- **Julian Looser** for great suggestions, proof reading and support.
- **Jason Smith** for making this thesis seem like "a walk in the park" compared with his Masters. Also, for never crying after losing so many times to the author at C&C Generals.
- **Michael Jason Smith** for always having time to help and for inspiring the author to finish this thesis quickly.
- **Mum** for delivering yummy food during the final thesis grind.
- **Steve Violich, Jay Holland and Nadine Fea** for proof reading.
- **Behshid Ghorbani** for your support and suggestions.
- **Nicolas Volpato** (aka Frenchie) for all the interesting conversations particularly about 'softball'.
- **Warwick Irwin** for providing helpful insight into OO design patterns.
- **The Human Interaction Technology Laboratory NZ** for resource support.
- **NASA** for providing the world photograph used in the globe browser.
- **Wises** for providing the Christchurch and Auckland street maps used in the map browser.

# References

Aaronson, D. & Scarborough, H. S. (1977), 'Performance theories for sentence coding: Some quantitative models', *Journal of Verbal Learning and Verbal Behaviour* **16**, 277–303.

*ACDSee* (2001).
*http://www.acdsystems.com/english/products/acdsee/acdsee-node.htm

Ahlberg, C. & Shneiderman, B. (1994), The Alphaslider: A Compact and Rapid Selector, *in* 'Proceedings of CHI'94 Conference on Human Factors in Computing Systems Boston, April 24–28', ACM Press, pp. 365–371.

Alley, B., Jones, M. & Jones, S. (2004), 'Control feedback for speed-dependant automated zooming interfaces on small screens'. Unpublished.

*Apple iPhoto* (2002).
*http://www.apple.com/iPhoto

Baldonado, W., Woodruff, A. & Kuchinsky, A. (2000), Guidelines for using multiple views in information visualization, *in* 'Proceedings of the working conference on Advanced visual interfaces', ACM Press, pp. 110–119.

Baudisch, P., Good, N., Bellotti, V. & Schraedley, P. (2002), Keeping things in context: a comparative evaluation of focus plus context screens, overviews, and zooming, *in* 'Proceedings of the SIGCHI conference on Human factors in computing systems', ACM Press, pp. 259–266.

Baudisch, P., Good, N. & Stewart, P. (2001), Focus plus context screens: combining display technology with visualization techniques, *in* 'Proceedings of the 14th annual ACM symposium on User interface software and technology', ACM Press, pp. 31–40.

Bederson, B. & Hollan, J. (1994), Pad++: A zooming graphical interface for exploring alternate interface physics, *in* 'Proceedings of ACM Conference on User Interface Software and Technology, 1994'.

Blanch, R., Guiard, Y. & Beaudouin-Lafon, M. (2004), Semantic pointing: improving target acquisition with control-display ratio adaptation, *in* 'Proceedings of the 2004 conference on Human factors in computing systems', ACM Press, pp. 519–526.

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.

Card, S. K., Mran, T. P. & Newell, A. (1987), *The Psychology of Human-Computer Interaction*, Morgan Kaufmann Publishers Inc, chapter 2, pp. 22–97.

Cockburn, A. & Firth, A. (2003), Improving the acquisition of small targets, *in* 'Proceedings HCI 2003', pp. 181–196.

Cockburn, A., Looser, J. & Savage, J. (2003), Around the world in seconds with speed-dependent automatic zooming, *in* 'Demostration in the Proceedings of the ACM User Interface Software and Technology, Vancouver, Canada', pp. 35–36.

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

Combs, T. & Bederson, B. (1999), Does zooming improve image browsing?, *in* 'Proceedings of the 4th ACM Conference on Digital libraries, Berkeley, California, United States', ACM Press, pp. 130–137.

Eckert, M. & Buchsbaum, G. (1993), *The significance of eye movements and image acceleration for coding television image sequences*, MIT Press, chapter 8, pp. 90–98.

Eslambolchilar, P. & Murray-Smith, R. (2004), Tilt-based automatic zooming and scaling in mobile devices, *in* ‘Mobile HCI 2004’.

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

Forster, K. L. (1970), ‘Visual perception of rapidly presented word sequences of varying complexity’, *Perception and Psychophysics* **8**(4), 215–221.

Furnas, G. (1986), Generalized Fisheye Views, *in* ‘Proceedings of the CHI’86 Conference on Human Factors in Computing Systems III’, Amsterdam; North Holland/ACM, pp. 16–23.

Furnas, G. & Bederson, B. (1995), ‘Space-scale diagrams: Understanding multiscale interfaces’, *Proceedings of CHI 95* pp. 234–241.

Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1994), *Design Patterns*, Addison-Wesley Publishing Company.

Guiard, Y., Beaudouin-Lafon, M. & Mottet, D. (1999), Navigation as multiscale pointing: extending fitts’ model to very high precision tasks, *in* ‘Proceedings of the SIGCHI conference on Human factors in computing systems’, ACM Press, 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.

Hart, S. & Staveland, L. (1988), Development of nasa-tlx (task load index): Results of empirical and theoretical research, *in* ‘Human Mental Workload’, Elsevier Science, pp. 139–183.

Healey, C. G., Booth, K. S. & Enns, J. T. (1996), 'High-speed visual estimation using preattentive processing', *ACM Transactions on Computer-Human Interaction* **3**(2), 107–135.

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', ACM Press, pp. 65–72.

Hornbæk, K. & Frøkjær, E. (2001), Reading of electronic documents: the usability of linear, fisheye, and overview+detail interfaces, *in* 'Proceedings of the SIGCHI conference on Human factors in computing systems', ACM Press, pp. 293–300.

Igarashi, T. & Hinckley, K. (2000), Speed-dependent automatic zooming for browsing large documents, *in* 'Proceedings of the 13th Annual ACM symposium on User Interface Software and Technology, San Diego, California, United States', ACM Press, pp. 139–148.

Jones, S., Jones, M., Marsden, G., Patel, D. & Cockburn, A. (2004), 'An evaluation of integrated zooming and scrolling on small-screens'. Unpublished.

Kahneman, D. (1968), 'Method, findings, and theory in studies of visual masking', *Psychological Bulletin* **70**, 404–425.

Kang, T. J. & Muter, P. (1989), 'Reading dynamically displayed text', *Behaviour and Information Technology* **8**, 33–42.

Kelly, D. (1979), 'Motion and vision. ii. stablized spatio-temporal threshold surface', *Journal of the Optical Society of America* **69**(10), 385–386.

Keown, L. (2000), Virtual 3d worlds for enhanced software visualization. Masters Thesis from the University of Canterbury.

Krauzlis, R. J. & Stone, L. S. (1999), 'Tracking with the mind's eye', *Trends in Neurosciences* **22**, 544–550.

Laakso, S., Laakso, K. & A.J., S. (2000), Improved scroll bars, *in* 'CHI '00 extended abstracts on Human factors in computer systems', ACM Press, pp. 97–98.

MacKenzie, I. (1992), Movement time prediction in human-computer interfaces, *in* 'Proceedings of Graphics Interface '92.', Morgan Kauffman.

MacKenzie, I. & Buxton, W. (1992), Extending Fitts' Law to Two-Dimensional Tasks, *in* 'Proceedings of CHI'92 Conference on Human Factors in Computing Systems Monterey, May 3–7', Addison-Wesley, pp. 219–226.

MacKenzie, I. & Riddersma, S. (1994), 'Effects of output display and control-display gain on human performance in interactive systems', *Behaviour and Information Technology* **13**(5), 328–337.

MacKenzie, I., Sellen, A. & Buxton, W. (1991), A comparison of input devices in element pointing and dragging tasks, *in* 'Proceedings of the SIGCHI conference on Human factors in computing systems', ACM Press, pp. 161–166.

Masui, T., Kashiwagi, K. & Borden IV, G. (1995), Elastic graphical interfaces for precise data manipulation, *in* 'Conference Companion on Human factors in Computing Systems, May 07-11. Denver, Colorado, United States', ACM Press, pp. 143–144.

Missal, M. & Keller, E. L. (2002), 'Common inhibitory mechanism for saccades and smooth-pursuit eye movements', *The Journal of Neurophysiology* **88**(4), 1880–1892. information on smooth-pursuit and saccades systems.

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

Osada, M., Liao, H. & Shneiderman, B. (1993), Alphaslider: searching textual list with sliders, *in* 'Proceedings of the Ninth Annual Japanese Conference on Human Interface'.

Patel, D., Marsden, G., Jones, S. & Jones, M. (2004), 'An evaluation of techniques for browsing photograph collections on small displays'. Unpublished.

Perlin, K. (1985), 'An image synthesizer', *Computer Graphics* **19**(3).

Perlin, K. & Fox, D. (1993), Pad: An alternative approach to the computer interface, *in* 'Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, September.', ACM Press, pp. 57–64.

Plaisant, C., Carr, D. & Shneiderman, B. (1995), 'Image-browser taxonomy and guidelines for designers', *IEEE Software* **12**(2), 21–32.

Rahman, T. & Muter, P. (1999), 'Designing an interface to optimize reading with small display windows', *Human Factors* **41**(1), 106–117.

Savage, J. (2002), 'Speed-dependent automatic zooming'. Honours report from the University of Canterbury.

Smith, D. & Henning, R. (1996), Stretch button scrollbar, *in* 'Proceedings of CHI'96 Conference on Human Factors in Computing Systems Vancouver, April 13–18', ACM Press, pp. 301–302.

Spence, R. (2002), Rapid, serial and visual: a presentation technique with potential.
*http://www.iis.ee.ic.ac.uk/bob/RSVP.pdf

Spence, R. & deBruijn, O. (1999), Rapid Serial Visual Presentation: A Space-Time Trade-Off in Information Presentation, *in* 'Proceedings of the working conference on advanced visual interfaces, May 2000. Palermo, Italy.', Addison-Wesley, pp. 189–192.

Tan, D., Robertson, G. & Czerwinski, M. (2001), Exploring 3D Navigation: Combining Speed-coupled Flying with Orbiting, *in* 'Proceedings of CHI'2001 Conference on Human Factors in Computing Systems Seattle, Washington, March 31–April 6', pp. 418–425.

Tullis, T. S., Boynton, J. L. & Hersh, H. (1995), Readability of fonts in the windows environment, *in* 'Conference companion on Human factors in computing systems', ACM Press, pp. 127–128.

van Wijk, J. & Nuij, W. (2003), Smooth and efficient zooming and panning, *in* 'Proceedings of IEEE Symposium on Information Visualization 2003 (INFOVIS2003)', pp. 15–22.

Ware, C. & Fleet, D. (1997), Context sensitive flying interface, *in* 'Proceedings of the 1997 Symposium on Interactive 3D Graphics, Providence, Rhode Island, United States', ACM Press, pp. 127–130.

Williams, R. (1998), *The Non-Designer's Type Book*, Peachpit Press.

Woods, D. (2000), 'OpenIL'. http://openil.sourceforge.net/.

Worden, A., Walker, N., Bharat, K. & Hudson, S. (1997), Making computers easier for older adults to use: area cursors and sticky icons, *in* 'Proceedings of CHI'97 Conference on Human Factors in Computing Systems. (Atlanta, Georgia, March 22-27)', pp. 266–271.

Zagier, E. J. S. (1997), 'A human's eye view: Motion blur and frameless rendering', *ACM Crossroads* .

Zhai, S., Smith, B. & Selker, T. (1993), Improving browsing performance: A study of four input devices for scrolling and pointing tasks, *in* 'INTERACT '93 and CHI '93 Conference Companion on Human Factors in Computing Systems. April 24-29. Amsterdam, The Netherlands', ACM Press, pp. 123–125.