Comparing Speed-Dependent Automatic Zooming with Traditional Scroll, Pan, and Zoom Methods

Andy Cockburn & Joshua Savage

Human-Computer Interaction Lab, Department of Computer Science, University of Canterbury, Christchurch, New Zealand

Tel: +64 3 364 2987
Fax: +64 3 364 2569
Email: {andy, jps42}@cosc.canterbury.ac.nz

Speed-dependent automatic zooming couples the user's rate of motion through an information space with the zoom level—the faster the user moves the 'higher' they fly above the work surface. Igarashi and Hinckley [2000] proposed using the technique to improve scrolling through large documents. Their informal preliminary evaluation showed mixed results with participants completing scrolling tasks in roughly the same time, or more slowly, than when using traditional methods. In this paper, we describe the implementation and formal evaluation of two rapidly interactive speed-dependent automatic zooming interfaces. The ecologically oriented evaluation shows that scrolling tasks are solved significantly faster with automatic zooming in both text document and map browsing tasks. Subjective preferences and workload measures also strongly favour the automatic zooming systems. Implications for the future of scrolling interfaces are substantial, and directions for further work are presented.

Keywords: Navigation, scrolling, zooming, speed-dependent automatic zooming, evaluation.

1 Introduction

Scrolling, panning and zooming are used to navigate through information spaces that are too large to be conveniently displayed within a single window. While scrolling and panning move the workspace within the window, zooming alters its scale. Because zooming changes the proportion of the workspace shown in each window, more scrolling is necessary when zoomed in, less when zoomed out. Most systems for browsing text and graphical documents support scrolling and zooming, and many also support panning.

Until recently, there had been surprisingly little research into understanding and improving the psychomotor performance of scrolling. Zhai et al [1997] showed that mouse-driven scrolling can be improved through the use of isometric controls that vary scroll rate with force. Hinckley et al [2002] showed that scrolling is accurately modelled by Fitts' Law [1954] even though it involves acquiring targets beyond the edge of the screen, and that mouse-wheel scrolling is improved by acceleration algorithms. These findings aid the theoretical understanding of scrolling, but they do not alter its basic behaviour. Consequently, they do not address the fundamental limitations of scrolling.

One of these limitations, identified by Igarashi and Hinkley [2000], is the disorientation caused by excessive visual flow when scrolling rapidly. In long documents a small movement of the scrollbar thumb causes a large movement in the document, and the rapid rate of change can be too great for the user to perceive, resulting in a visual blur. Although users can ease this problem by altering the zoom level before and after scrolling, doing so involves tedious interface manipulations.

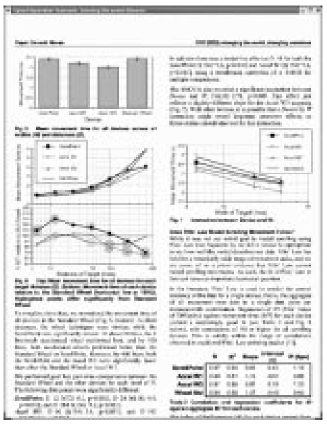
Igarashi and Hinckley proposed speed-dependent automatic zooming (SDAZ) as a solution. SDAZ automatically varies the zoom level dependent on the scroll rate. When scrolling quickly the display is zoomed out, and when stationary or scrolling slowly the display is zoomed in, as shown in Figure 1. An informal preliminary study (n=7) of the technique 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. Their paper and their prototype implementations[1] provide dramatic and compelling demonstrations of the technique. There is, however, a risk that their results were adversely affected by the informal nature of the evaluation and by implementation compromises that were necessary to aid rapid and fluid interaction in their Java prototypes.

This paper describes a formal evaluation of speed-dependent automatic zooming in support of everyday document navigation tasks. Section 2 describes the design and implementation of our document and map browsing applications. The experimental design and results are presented in Sections 3 and 4. Results are discussed, compared with related work, and used to direct further work in Section 6. Conclusions are presented in Section 6.

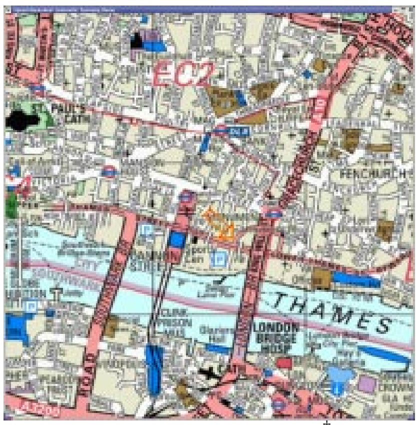
2 Document and Map Browsing Applications

Igarashi and Hinckley described five prototype SDAZ applications: a web browser, a map viewer, an image browser, a dictionary browser, and a sound editor. The image browser, dictionary browser and sound editor were not promising (as discussed in Section 6), so they evaluated only the web browser and the map viewer.

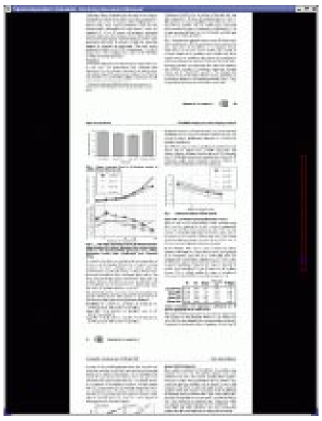
[1] A demonstration applet is available at www-ui.is.s.u-tokyo.ac.jp/~takeo/java/autozoom/autozoom.htm



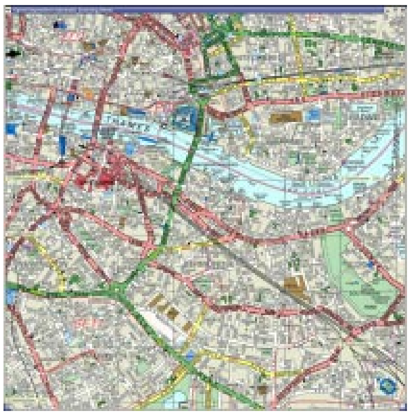
(a) Document, stationary/slow



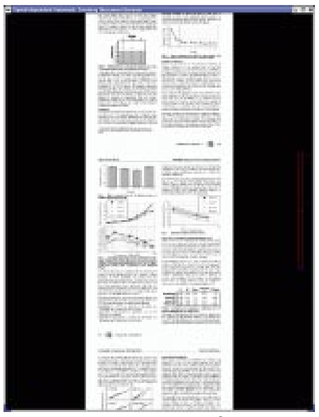
(d) Map, stationary/slow[†]



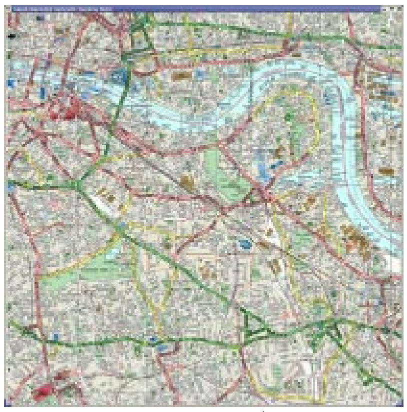
(b) Document, mid-speed



(e) Map, mid-speed[†]



(c) Document, fast



(f) Map, fast[†]

Figure 1. Slow, medium and fast scrolling with our document and map browsers. Short video clips of the systems in operation are available at www.cosc.canterbury.ac.nz/~andy/liter.html.

Their prototypes were written in Java, and required various implementation compromises to achieve rapid and fluid interactivity. In the web browser, plain text was rendered as simple horizontal lines when zoomed out, and when zooming in only discrete font-sizes were available. Although they experimented with well-known zoomable user interface toolkits Pad++ [Bederson et al 1996] and Jazz [Bederson and McAlister 1999], their performance was too slow. Their map browser was also limited because it used an artificially synthesized map to ease implementation and aid performance. They state “Although this prototype allows the user to experience zooming and panning in a multi-scale environment, an implementation using real map data would be necessary to obtain further insights.”

Our implementations are written in C using the OpenGL graphics libraries, allowing graphics hardware acceleration to provide smooth and fluid animation at more than 50 frames per second. These frame rates are possible using standard graphics cards available on consumer-level computers. The capability of consumer-level graphics hardware is emphasised by Rhyne [2002], who reports that the United States Department of Energy uses standard PCs with consumer-level graphics cards for scientific visualisation. Because OpenGL renders objects in 3D, the automatic zooming effect is easily implemented by moving the viewpoint away from the workspace surface.

The relationship between the zoom level and scroll speed is determined by the same formula in both the document and map browsing applications: shown in Equation 1. Zoom level indicates the perceived distance from the document; hence if the zoom level is high the document appears further away (smaller).

$$zoomlevel = k * scrollspeed - threshold$$ **Equation 1**.

The constant *k* affects the rate of change of the zoom level. The minimum value for the zoom level is zero (fully zoomed in), which applies when the cursor is stationary or moving slower than the threshold value. To begin scrolling the user presses the left mouse-button over the document and drags in the direction they wish to scroll (up or down in the document browser, any direction in the map browser). Displacing the cursor further from the initial selection point increases the scrolling speed (the value of *scrollspeed* in Equation 1). Releasing the mouse-button stops scrolling, and the zoom level is fluidly returned to zero through a rapidly animated ‘falling’ effect. The falling effect is also used to limit the rate that the user can zoom into the document, reducing disorientation. Without it the display can immediately change between fully zoomed-out and fully zoomed-in when the user changes direction by 180 degrees.

2.1 Document Browser

The document browser views portable document format (PDF) and postscript (PS) files: see Figure 1(a,b,c). The zoom level is determined through Equation 1, where the scroll speed is determined as follows:

$$scrollspeed = \left| Y_{ip} - Y_{cp} \right|$$ **Equation 2**.

where Y_{ip} and Y_{cp} represent the initial and current y-coordinates of the mouse.

A velocity scrollbar shows the scrolling speed and direction. When stationary, the velocity indicator is centred in the scrollbar, and when scrolling upward/downward at full speed the indicator is at the top/bottom of the scrollbar. Two additional marks in the velocity scrollbar indicate the threshold scroll speed beyond which zooming occurs.

In the implementation, each PDF or PS page in the document is converted to a separate Truevision Targa[2] (TGA) file for easy texture mapping in OpenGL. Splitting the document into separate pages increases performance because unseen pages do not need to be rendered.

2.2 Map Browser

The map browser, shown in Figure 1(d,e,f), behaves similarly to the document browser, except scrolling occurs on two dimensions rather than one. The zoom level is determined through Equation 1, with the scroll speed determined as follows:

$$scrollspeed = \sqrt{\left(Y_{ip} - Y_{cp}\right)^2 + \left(X_{ip} - X_{cp}\right)^2} \qquad \textbf{Equation 3}.$$

where Y_{ip}, Y_{cp}, X_{ip}, and X_{cp} represent the initial and current x and y-coordinates of the mouse.

When the user begins scrolling a red cross appears in the centre of the display and a vector shows the speed and direction of movement. When scrolling rapidly the view is zoomed out, revealing large areas of the map (see Figure 1f).

The implementation is similar to the document browser, with large maps constructed from separate TGA image files (normally OpenGL has a maximum image size limit of 2048×2048 pixels).

3 Experimental Design

The experimental objective was to compare the efficiency, preferences, and general usability issues of speed-dependent automatic zooming with traditional scrolling techniques. To increase ecological validity, the traditional interfaces were standard commercial applications—Adobe's Acrobat Reader[3] version 5 for document browsing tasks, and Paint Shop Pro[4] version 5 for map browsing tasks. Both of these interfaces support traditional scroll, pan and zoom facilities.

The experiment was a 2×2×2 repeated measures factorial design, and was repeated for document and map browsing tasks. The dependent measure was task completion time. The three factors were as follows:

- *Interface type*. The levels of this factor were the speed-dependent automatic zooming interfaces and their traditional equivalents (Acrobat Reader or Paint Shop Pro).
- *Task type*. When document browsing the two levels of this factor were 'locate a picture' and 'locate a text heading', and they involved finding specific items in the document. When map browsing the levels were 'locate from direction'

[2] www.truevision.com
[3] www.abobe.com
[4] www.jasc.com

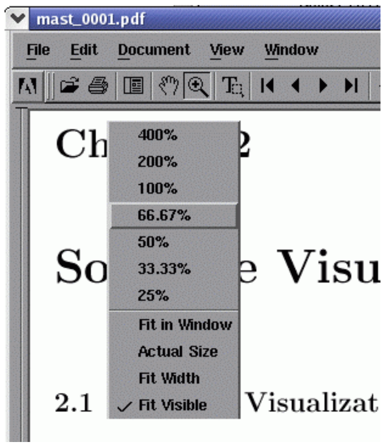


Figure 2: Zooming and panning controls in Acrobat Reader.

and ‘locate from path’. The map browsing tasks involved finding named schools when the search was cued with a compass direction or with a route to follow (such as a named street or river).

- *Scroll distance*. The levels of this factor were ‘short’ and ‘long’ distance. In short tasks the target was approximately five pages or five map squares from the initial location. Long distance targets were approximately 20 pages or 20 map squares from the initial location. This factor was intended to show whether one interface was particularly suitable to short or long scrolling activities.

The repeated-measures design reduces the number of participants required for statistical power. It also reduces the impact of variation between participants—a participant with particularly good hand-eye co-ordination, for instance, is likely to perform well with both interfaces.

3.1 Participants

Twelve volunteer participants (eleven males, one female) took part in the experiment. All were graduate level computer science students in their early twenties. None had previously used speed-dependent automatic zooming interfaces. All frequently used Acrobat Reader and half had previously used Paint Shop Pro 5.

Because speed-dependent automatic zooming creates fluid and dynamic visual effects, it was possible that experience with computer games would influence its use. Three participants reported that they never played computer games, three stated that they played between one and three hours a week, four played between five and ten hours a week and two played more than twenty hours each week. The results, reported in Section 4, suggest that game-playing experience was not a major factor in performance or subjective preferences.

Each participant’s involvement in the experiment lasted approximately forty minutes, including training time.

3.2 Materials

The experiment was run on a 1.2 GHz AMD Athlon computer with 640Mb of RAM and a Geforce 4 MX Video card. The 19-inch IBM display was set to a resolution of 1024×768 pixels. Input was provided through a standard Logitech 3-button mouse, which was cleaned after each evaluation. All experimental software ran under the Windows XP operating system. Timing data was recorded using a stopwatch.

Acrobat Reader, shown in Figure 2, provides a variety of interface features for scrolling, panning and zooming. When the magnifying glass tool is selected (shown selected in the toolbar of Figure 2) each left mouse-button click magnifies or diminishes the document by approximately 25%. The ‘View’ menu and the pop-up context menu (shown in Figure 2) provide a variety of shortcuts for setting the zoom level. Scrollbars allow horizontal and vertical scrolling as normal (no horizontal scrolling was necessary in the tasks). The ‘Continuous’ view option was used in the evaluation, which means that dragging the scrollbar thumb dynamically displayed page motion. Panning is achieved by selecting the ‘hand’ tool and dragging with the left mouse button.

Paint Shop Pro also provides a variety of zooming and scrolling features. Normal horizontal and vertical scrollbars are available on the bottom and right sides of the window. A shortcut scroll/zoom technique is possible by clicking on the image with the left or right buttons to zoom in or out. This simultaneously zooms and centres the display on the clicked location. A panning tool, similar to that in Acrobat Reader, allows the workspace to be panned without zooming.

The document used for the document-browsing tasks was a 157 page Masters Thesis (PDF format), consisting of nine chapters and two appendices. The map browsing tasks used a large Auckland City road map. Experimental tasks with Acrobat Reader were initially displayed at a ‘Fit to Width’ level (approximately 125% zoom) which is a comfortable reading size. In the map browsing tasks the initial zoom level was set so that individual street names were clearly legible. These zoom levels were similar to the initial stationary zoom levels with the SDAZ interfaces.

The NASA Task Load Index (TLX) worksheets [Hart and Staveland 1988] were used to measure subjective assessments of workload in the tasks. Responses to six measures were taken using a five-point Likert scale, from 1 (low) to 5 (high). The measures were ‘mental demand’, ‘physical demand’, ‘temporal demand’, ‘effort’, ‘performance’, and ‘frustration level’. The participants read explanations of these measures before using the worksheets.

3.3 Procedure

The document browsing tasks were completed first to better prepare participants for the 2D map tasks. The order of exposure to other factors was counter-balanced to minimise learning effects. Half of the participants used the SDAZ interfaces first, half the traditional interfaces first.

Each participant completed ten document-browsing tasks using both the SDAZ and Adobe Acrobat Reader interfaces. Two sets of ten similar document navigation tasks were created. The task sets consisted of two training tasks and two tasks in each combination of distance (long and short) and task type (locate picture and locate text heading). The task sets were also counter-balanced across interfaces, so that half of the participants used each set with each interface.

Each interface was described to the participants before they used it, and they were then allowed five minutes of practice. The training tasks were used to ensure that the participants understood how to use the interface, and to familiarise them with the mechanism for presenting tasks. One training task was a long distance ‘locate picture’ task, and the other was a long distance ‘locate text heading’ task. Timing data from the training tasks was discarded.

All tasks were presented to the user in a display in the top right-hand corner of the screen. The evaluator first read the task description to the participant, and they were then asked to read the task aloud. An example ‘locate picture’ task was ‘Locate the picture of the green world globe up from your current location’, and an example ‘locate text heading’ task was ‘Locate the Software Visualization heading below your current location’. The participants were informed that the clock would stop when they read aloud the first word of the picture caption or the first word below the section heading. They were also told that if they became lost they could ask to have the starting location identified, but that the task time would continue to run.

The ten map browsing tasks were similarly administered. The first training task was a long distance ‘direction’ task and the second was a long distance ‘path’ task. An example ‘direction’ task is ‘Locate Wairau Intermediate School in Sunnynook, North West of here’. An example ‘path’ task is ‘Follow State Highway 16 north. When you meet State Highway 1, follow it west to the Auckland Institute of Technology’. All long distance ‘path’ tasks included a junction such as that in the example. The remaining eight tasks consisted of two tasks in each combination of distance and task type.

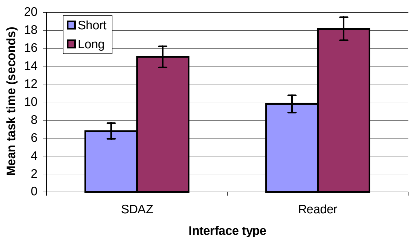
After completing the tasks with each interface the participants used the NASA TLX worksheets to assess their workload. Finally, they were asked to state which interface they preferred for browsing documents and maps.

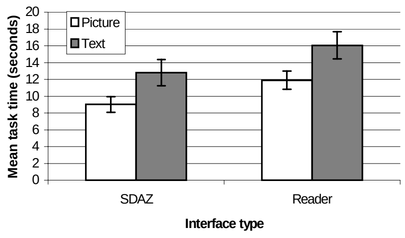
4 Results

The participants had no obvious problems with the experimental method or with adapting to the SDAZ interfaces.

Across both the SDAZ and traditional interfaces, and across short and long distances, the mean time to complete document-browsing tasks was 12.4 seconds (standard deviation 5.8), slightly faster than the mean for the two dimensional map-browsing tasks at 17.5 seconds (s.d. 9.0).

Timing data in the document and map browsing tasks was analysed using a 2×2×2 analysis of variance with repeated measures.





(a) Interface type by distance (b) Interface type by task type.

Figure 3: Mean task times for document browsing. Error bars show ± 1 standard error.

4.1 Document Browsing

Participants completed the tasks significantly faster with the SDAZ interface (mean 10.9 seconds, s.d. 5.3) than with Acrobat Reader (mean 14.0 seconds, s.d. 5.8): $F_{1,11}$=16.9, p<0.01. On average, the SDAZ interface reduced the task time by 22%.

As expected, short distance tasks were faster than long distance ones ($F_{1,11}$=35.3, p<0.01), with short and long means of 8.3 seconds (s.d. 4.7) and 16.6 seconds (s.d. 6.2). There was no interaction between interface type and distance ($F_{1,11}$<0.01, p=0.94), indicating that performance with the SDAZ and Reader interfaces deteriorated similarly as task distance increased (see Figure 3a).

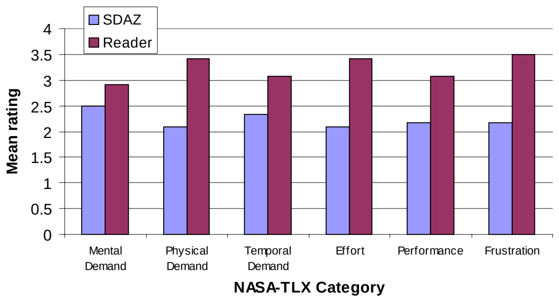
The two task types were also reliably different from one another ($F_{1,11}$=12.3, p<0.01), with pictures being located more rapidly than titles: means of 10.5 seconds (s.d. 5.1) and 14.4 seconds (s.d. 7.8) respectively. The larger size and greater visual distinctiveness of pictures compared to titles explains this effect. There was no interaction between interface type and task type ($F_{1,11}$=0.09, p=0.77) indicating that performance with the SDAZ and Reader interfaces deteriorated similarly between the task types (see Figure 3b).

Responses to the NASA-TLX worksheets showed that the participants rated task loads more lightly with the SDAZ interface. Figure 4a shows the mean responses to questions on the mental, physical and temporal demand of the tasks, and on the effort, performance and frustration levels. In all cases, the SDAZ interface has a lower workload rating than Acrobat Reader.

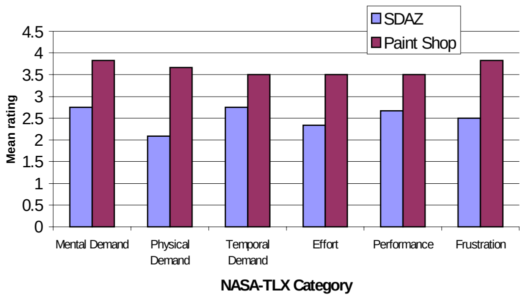
Finally, all but one of the participants stated that they preferred the SDAZ interface, and several made comments such as 'all scrolling should work like this!' The one participant who preferred the Reader interface objected to being 'forced' to zoom out when scrolling. He stated that he would have preferred the technique if there had been an option to freeze the zoom level while scrolling.

4.2 Map Browsing

The participants were significantly faster with the SDAZ interface than with Paint Shop Pro: $F_{1,11}$=38.9, p<0.01. On average, the SDAZ interface was 43% faster,



(a) Document browsing.



(b) Map browsing.

Figure 4: Mean responses to NASA-TLX workload assessments using five-point Likert scales.

with SDAZ and Paint Shop means of 12.7 seconds (s.d. 7.2) and 22.3 seconds (s.d. 14.4) respectively.

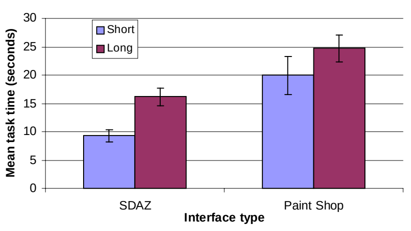
As in the document browsing tasks, short distance tasks were faster than long ones ($F_{1,11}$=5.5, $p<0.05$), with means of 14.6 seconds (s.d. 13.3) and 20.4 (s.d. 10.7). Also reflecting the document browsing tasks, performance in short and long distance tasks deteriorated similarly for both interface types (see Figure 5a): $F_{1,11}$=0.37, p=0.6.

The ‘locate from path’ task type was completed more rapidly than ‘locate from direction’, with means of 14.4 seconds (s.d. 10.1) and 20.3 seconds (13.8) respectively: $F_{1,11}$=5.6, $p<0.05$. This effect is explained by the relative simplicity of following obvious landmarks (such as a road) to a location, rather than following a relatively crude compass direction. Performance with both interfaces deteriorated similarly between ‘locate from path’ and ‘locate from direction’ task types, as shown in Figure 5b: $F_{1,11}<0.01$, p=0.9.

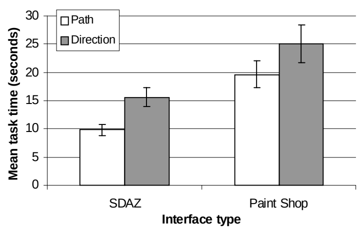
The NASA-TLX worksheet responses showed uniformly lower workload ratings with the SDAZ interface (see Figure 4b). All of the participants stated that they preferred the SDAZ interface.

4.3 Observations and Comments

Several participants commented that the traditional scroll, pan and zoom mechanisms were ‘too separate’, requiring several interface manipulations with



(a) Interface type by distance



(b) Interface type by task type.

Figure 5: Mean task times for map browsing. Error bars show ± 1 standard error.

different controls for essentially the same task. Consequently, the participants felt 'busy' with interface adjustments (increasing their Physical Demand ratings on the NASA-TLX worksheets) or they accepted the long scrolling distances associated with the default zoom level (increasing their Frustration level). The participants' annoyance at the separation of controls was particularly notable when map browsing because two separate scrollbars were needed for horizontal and vertical scrolling.

The abrupt transitions between discrete zooming levels with the traditional interfaces also caused problems. The absence of animation between the pre- and post-magnification views meant that the participants had to reorient themselves with each zoom action. This problem was particularly notable with Paint Shop Pro because of its zoom centring property—many participants expected the clicked location to remain in the same place, rather than move to the centre of the display.

In the first moments of exposure to the SDAZ interfaces, some of the participants made light-hearted comments that they expected it to induce motion sickness or to make them dizzy. None of the participants mentioned dizziness or motion sickness later in the experiment.

One criticism of the SDAZ interfaces was that the participants missed the spatial orientation normally provided by scrollbars. The location of the scrollbar thumb allows users to quickly determine their approximate spatial location within a document. Similarly, the thumb can be used to rapidly move to the start/end of the document and other approximate locations. This useful information can be easily added to SDAZ interfaces.

5 Discussion and Related Work

5.1 Comparison with Igarashi and Hinckley's results

Why did our experiment produce strong results in favour of speed-dependent automatic zooming when Igarashi and Hinckley's (2000) preliminary evaluation indicated that it was little different or worse than traditional schemes? There are several potential explanations.

Implementations. Our C and OpenGL systems exploit the high frame-rates and fluid animation available through graphics hardware. This avoids some of the implementation compromises necessary to achieve rapid interaction in Igarashi and

Hinckley's Java-based systems. In their document browser zoomed-out views of text were depicted as horizontal lines rather than miniaturised fonts, and discrete font sizes were used to simulate dynamic font scaling. In their map browsing evaluation an artificially generated map was used rather than a real one. It is possible that subtle differences between our implementations and theirs explain the discrepancy between results. In particular, differences in frame-rate are important in dynamic and fluid interactive presentations, as demonstrated by the poor quality of low frame-rate video.

Experimental Objectives. Igarashi and 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. With different experimental objectives, and with greater statistical power from wider participation, they may have been able to discriminate between performance with the interfaces.

Competing Interfaces. We compared our SDAZ interfaces with commercial implementations of traditional systems (Adobe Acrobat Reader and Paint Shop Pro), and all interfaces were controlled using a mouse. In their document browsing tasks, Igarashi and Hinckley compared SDAZ with a mouse-driven unspecified web browser. They do not mention whether their web browser supported zooming, but images in their paper suggest that it did not. It is possible (though we believe unlikely) that Acrobat Reader's zooming facilities caused slower performance in our evaluation, and that this accounted for the comparative efficiency of SDAZ.

In Igarashi and Hinckley's map navigation tasks both interfaces were controlled using a joystick, allowing users to pan and zoom concurrently when using traditional interfaces. Our 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's centring zoom). Although the contrasting input devices may have contributed to the divergent results, mouse input remains the standard on desktop computers.

Task Types. Igarashi and Hinckley's map browsing tasks involved locating a white dot in an artificial terrain. The dot was continually visible in a small 'global radar' in the corner of the display. These tasks test abstract target acquisition, and in later work Hinckley et al [2002] showed that scrolling performance conforms to Fitts' Law [1954]. Our tasks, in contrast, include the extraction of meaningful information from different levels of magnification. Evaluations based on abstract theoretical behaviour and on ecologically oriented tasks are both equally important in understanding the strengths and weaknesses of new interaction techniques. We intend to investigate both in our further work.

Participants. Differences between the participant pools may also have affected the results. All of our twelve participants were expert computer users (graduate level Computer Science students), and all but three regularly played computer games. Igarashi and Hinckley's participants were all "good" or "average" computer users, and four of the seven reported that they played computer games "sometimes", "almost every day" or "frequently". Although four of Igarashi and Hinckley's participants were female (compared to only one of ours), there was no notable difference between male and female performance in the data they present. We intend to conduct further evaluations with a broad participant pool to clarify whether SDAZ techniques remain effective for less experienced users.

5.2 Related work

As desktop computers increase in power, automatic zooming techniques such as fisheye views and SDAZ are becoming increasingly viable. The main icon panel in the MacOs X desktop[5], for instance, provides a dynamic fisheye that automatically magnifies icons as the cursor approaches.

This mainstream deployment of automatic zooming has prompted research into their use. McGuffin and Balakrishnan [2002] show that the acquisition of expanding targets conforms to Fitts' Law, and that selection time depends primarily on the final target size, even when the expansion begins very late in the movement towards the target. Gutwin [2002] describes a fisheye view 'hunting' problem in which dynamic distortion causes users to miss their targets because they move most rapidly when the cursor is near. He shows that acquisition times can be improved through 'speed-coupled flattening' which removes the distortion effect when the cursor moves rapidly. The distortion is reapplied when the user decelerates and stops the cursor on final target approach.

There have also been several ecologically oriented evaluations of automatic zooming techniques. Bederson [2000] evaluated "fisheye menus" as an alternative for selecting targets from long menus. With fisheye menus, items are magnified (displayed in a large font) when near the cursor, and diminished when far from it. The technique was faster than two types of scrolling menu, and about the same as cascading menus. Schaffer et al [1996] showed that a 2D graphical fisheye allowed faster navigation through hierarchically organised networks than traditional full-zoom techniques. Finally, Tan et al [2001] showed users were able to acquire and move targets in a virtual 3D scene more rapidly when using a "speed-coupled flying and orbiting" system that varied height above a 3D world with speed. The concept of coupling speed with height was first described by Ware and Fleet [1997] who evaluated a variety of schemes for automatically adjusting velocity from the user-controlled height in "fly-by" visualisations.

5.3 Scope, limitations and further work

Igarashi and Hinckley used five categories to discuss appropriate and inappropriate domains for speed-dependent automatic zooming—size of the data space, type of the data, frequency of access to the data, input device, and user level.

Size of the data space. We agree that SDAZ is best used for intermediate sized data spaces. In small data spaces (between one and four times the amount displayed in a single screen) traditional scrolling is sufficient, and in large data spaces (thousands of screenfuls) searching methods are needed. We are currently investigating ways of adapting SDAZ to extremely large data spaces.

[5] www.apple.com/macosx/theater/dock.html.

Type of the data space. Although we agree that SDAZ is appropriate for navigating through spatial information, we are not convinced that it is 'difficult to apply to symbolic information such as a dictionary'. Their unsuccessful dictionary browser provided a scrolling list of words. Words were removed from the list when scrolling fast, but this caused confusion. We believe that SDAZ can provide a natural method for browsing symbolic data that is analogous to the way we riffle book pages (for example, when searching a physical dictionary).

In our further work we will develop and evaluate SDAZ in spatial, symbolic and abstract data types.

Frequency of access to the data. Igarashi and Hinckley stated that SDAZ would work best when the same data items are accessed repeatedly. Our evaluation tasks involved single accesses to data items, yet SDAZ outperformed traditional interfaces. We wish to further examine the suitable activities for SDAZ.

Input device. Igarashi and Hinckley recommended that SDAZ is particularly suited to self-centring and absolute input devices such as joysticks rather than relative pointing devices such as a mouse. Although self-centring devices may be particularly suitable for SDAZ, our evaluation shows that efficiency gains are produced with mouse-driven input.

User level. Igarashi and Hinckely recommended that SDAZ is less appropriate for novice users. Given that all our participants were expert users, we have little data to assess this recommendation, and we will investigate novice use in further work.

We will also intend to develop guidelines for the calibration of speed-dependent automatic zooming. Factors affecting the calibration of SDAZ include the perceptual issues of maximum useful visual flow at various zoom levels, and the maximum useful zoom-out level for documents of different types.

6 Conclusions

Scrolling is one of the most common activities in everyday computer use, yet there has been surprisingly little research into improving its performance.

Recently Igarashi and Hinckley proposed 'speed-dependent automatic zooming', which couples scroll rate with the zoom level. The objective was to overcome problems arising from excessive visual flow when scrolling fast. Their preliminary evaluation produced disappointing results, with similar or slightly worse performance than traditional methods.

The evaluation reported in this paper shows speed-dependent automatic zooming (SDAZ) in a new light. In ecologically oriented document browsing tasks, our participants were 22% faster when using SDAZ than when using a common commercial document viewer, and in map browsing tasks the performance benefits increased to 43%. Workload assessments, preferences, and the participants' comments all amplified the efficiency and effectiveness of the automatic zooming approach.

The results suggest that Igarashi and Hinckley's invention could have dramatic implications for the future of scrolling. Our further work will continue to assess and evaluate critical factors in the success and adoption of SDAZ interfaces. Although the technique needs relatively high levels of processing power, continual improvements in processing speeds and graphics hardware will soon negate this.

References

Bederson B, 2000 "Fisheye Menus", in *Proceedings of the 2000 ACM Conference on User Interface Software and Technology* (San Diego, California.) pp 217--225

Bederson B, Hollan J, Perlin K, Meyer J, Bacon D, Furnas G, 1996 "Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics" *Journal of Visual Languages and Computing* **7** 3--31

Bederson B, McAlister B, 1999 *Jazz: An Extensible 2D+Zooming Graphics Toolkit in Java*, Technical report: HCIL-99-07. Department of Computer Science, University of Maryland.

Fitts P, 1954 "The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement." *Journal of Experimental Psychology* **47** 381-391

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* Ed P a M Hancock, N pp 139--183

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

Igarashi T, Hinckley K, 2000 "Speed-dependent Automatic Zooming for Browsing Large Documents", in *Proceedings of the 2000 ACM Conference on User Interface Software and Technology* (San Diego, California.) pp 139--148

McGuffin M, Balakrishnan R, 2002 "Acquisition of Expanding Targets", in *Proceedings of CHI'2002 Conference on Human Factors in Computing Systems* (Minneapolis, Minnesota, 20--25 April) pp 57--64

Rhyne T, 2002 "Computer Games and Scientific Visualization" *Commmunications of the ACM* **45** 40--44

Schaffer D, Zuo Z, Greenberg S, Bartram L, Dill J, Dubs S, Roseman M, 1996 "Navigating Hierarchically Clustered Networks through Fisheye and Full-Zoom Methods" *ACM Transactions on Computer Human Interaction* **3** 162--188

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

Ware C, Fleet D, 1997 "Context Sensitive Flying Interface", in *Symposium on Interactive 3D Graphics* (Providence, RI) pp 127--130

Zhai S, Smith B, Selker T, 1997 "Improving Browsing Performance: A Study of Four Input Devices for Scrolling and Pointing Tasks", in *INTERACT'97: the Sixth IFIP Conference on Human Computer Interaction* pp 286--292