OrthoZoom Scroller: 1D Multi-Scale Navigation

Caroline Appert
LRI/INRIA Futurs
Bât. 490, Univ. Paris-Sud, 91405 Orsay, France
Caroline.Appert@lri.fr

Jean-Daniel Fekete
INRIA Futurs/LRI
Bât. 490, Univ. Paris-Sud, 91405 Orsay, France
Jean-Daniel.Fekete@inria.fr

ABSTRACT
This article introduces the OrthoZoom Scroller, a novel interaction technique that improves target acquisition in very large one-dimensional spaces. The OrthoZoom Scroller requires only a mouse to perform panning and zooming into a 1D space. Panning is performed along the slider dimension while zooming is performed along the orthogonal one. We present a controlled experiment showing that the OrthoZoom Scroller is about twice as fast as Speed Dependant Automatic Zooming to perform pointing tasks whose index of difficulty is in the 10-30 bits range. We also present an application to browse large textual documents with the *OrthoZoom Scroller* that uses semantic zooming and snapping on structure.

Author Keywords
Interaction technique, multi-scale navigation, pointing task, scrolling task.

ACM Classification Keywords
H.5.2. [User Interfaces]: Interaction styles; I.3.6 [Methodology and Techniques]: Interaction techniques.

INTRODUCTION
One-dimensional (1D) navigation and selection tasks such as using a slider or a scrollbar involve selecting a value over a bounded range through pointing. Screen-size and resolution limitations pose problems when the range becomes too large to map one value per pixel. For example, in a 1000 pixel-wide slider representing a range from 1 to 10,000, each pixel represents ten values. It is therefore impossible to continuously scroll over a large document in which the number of pages far exceeds the number of pixels in the scrollbar.

Information visualization applications commonly use sliders to select values within large ranges but require a text field entry to specify a precise value. Entering text to select a value over a continuous range breaks Shneiderman's principles of direct manipulation [21] and consumes screen real-estate.

To deal with large spaces, multi-scale interfaces [10, 20] introduce the scale dimension, sometimes called Z (we use the words scale and zoom interchangeably in this article). We introduce *OrthoZoom Scroller*, a mouse-based multi-scale 1D scrolling and pointing technique that performs better than the only other multi-scale technique using standard input devices. *OrthoZoom Scroller* allows users to achieve very difficult 1D pointing tasks (~30 bits) by controlling panning with one mouse dimension and zooming with the other. Using our technique, a user could select one base pair out of the 3 billions (~32 bits) of the human genome in one continuous multi-scale pointing gesture.

We first review related work and then present the *OrthoZoom Scroller*. We evaluate it by comparing it to the *Speed Dependant Automatic Zooming* technique [13] which aims at similar goals and has been well studied before. Finally, we present an application to browse large textual documents with the *OrthoZoom Scroller*.

RELATED WORK
We classify interaction techniques to select a precise value within a range in two categories: discrete techniques and continuous techniques.

Discrete techniques
Discrete techniques use non-continuous mechanisms such as filtering to remove values from the range or multiple interactions to control the zoom.

BinScroll [15] is a technique that requires four buttons to perform a dichotomic search in a list of textual data. Two buttons allow the user to progressively reduce the list by selecting the top half or bottom half of the list relative to a current item. The two other buttons are used to select an item or cancel an operation.

LensBar [16] is a listbox augmented by a slider and a text entry field to perform selections in a large list of data. The list can be reduced by entering a pattern into the textual entry to select only matching data or by performing zooming around the current item. Clicking on the current

item and moving to the left zooms out, displaying a coarser sampling of items. LensBar controls the visibility of items using a degree of interest function (DOI) computed around the current item. Thus, LensBar requires both a keyboard and data pre-processing to assign a DOI to each item. The scale factor and pan are controlled using the mouse but cannot be specified in one continuous interaction.

The Alphaslider [1] is an augmented slider consisting of two or three sub-sliders, each one representing different granularity of movement within the depicted range of the whole slider. Although each sub-slider is manipulated in a continuous way, switching between two sub-sliders "breaks" the interaction. Furthermore, the granularity is limited to three levels. The FineSlider [17] extends the Alphaslider's idea by allowing users to adjust the granularity of the slider's control: clicking on the slider at a spot other than the knob moves the knob toward the cursor location at a speed proportional to distance between the knob and the clicking point. The PVSlider [2] also uses a rubber-band metaphor to adjust the granularity of parameter manipulation. The FineSlider and PVSlider have a wider range of precision than the AlphaSlider but, once again, switching between two different granularity levels is not continuous.

The Control Menu [18] uses a circular menu to trigger the control of continuous parameters. In their article, they show how to navigate in a zooming interface using a continuous zoom triggered by a horizontal item and a pan triggered by a vertical item, at the current zooming level.

Continuous techniques

Guiard et al. [10] demonstrate that high precision selection tasks can be thought of as multi-scale navigation tasks. Continuous techniques make use of two categories of dimensions: the scale/zoom dimension and the pan/scroll dimension, to select a precise value with a continuous interaction.

Using a non-standard input device

Some techniques use non-standard input devices to perform navigation in a multi-scale world.

Zhai et al. [24] show the benefits of controlling zoom and pan with bi-manual techniques. For example, in [10, 11], users control panning by moving a stylus on a tablet with their preferred hand while they control zooming with a joystick with their non-preferred hand. These techniques are challenging to transfer to handheld devices.

Zoom Sliding, or *Zliding* [20], does not necessary require both hands by using a pressure-sensitive tablet. It fluidly integrates zooming via pressure control with panning via x-y cursor movement. The limited range and precision of pressure levels requires additional techniques such as clutching or using the keyboard to achieve a precise control of scaling.

Using a standard input device

Two recent techniques use circular motion to control the zoom factor. Both use clockwise motion to scroll the view down and counterclockwise motion to scroll it up.

The Radial Scroll Tool [22] uses the circle radius to adjust the scrolling rate: smaller circles mean faster scrolling, while larger circles mean slower scrolling. The Virtual Scroll Ring [22] interprets circular movements differently: it uses the distance traveled along the circumference of the circle instead of the radius. Larger or faster movements produce faster scrolling while smaller or slower movements produce slower scrolling. On some input devices, such as the mouse, circular movements can be difficult to do. Furthermore, controlling a linear parameter using a circular dimension can be disturbing for novice users.

The other techniques use two linear dimensions to control zoom and pan.

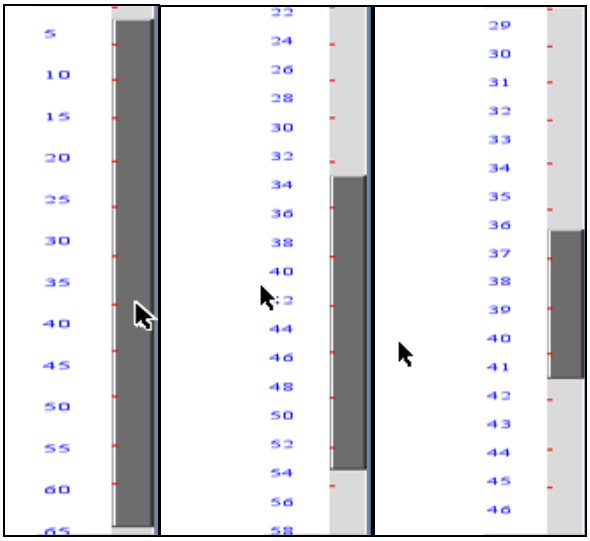
The Position+Velocity Slider is a stylus-based technique proposed in LEAN [19], a prototype to manage video streams. To browse videos, the user begins a drag anywhere in the video window, moves horizontally to browse and vertically to adjust the browsing velocity (the user always starts at the minimum velocity). The authors have qualitatively evaluated the whole interface making difficult to measure the benefits of the Position+Velocity Slider.

The InfoVis Toolkit [6] provides multi-resolution sliders: the precision is increased with the orthogonal distance to the slider track. However, no evaluation has been conducted on the effectiveness of the technique and no feedback is provided so users are usually not aware of the feature.

Speed-Dependent Automatic Zooming (SDAZ) [13] is designed to facilitate navigation tasks over large spaces. Navigation is controlled by a dragging interaction that can be activated anywhere. The scrolling speed is proportional to the distance between the clicking point and the current point. This technique also keeps the visual flow of the navigation constant by adjusting the zoom factor dynamically: the zoom factor is linked to the scrolling speed. This behavior allows users to continuously adjust their granularity. It requires fine tuning to adapt the visual flow to the user's abilities. Cockburn et al. [5] have shown that SDAZ was the most efficient technique to reach a target in a scrolling interface for large text documents.

All the techniques controlling a 2D space require either a non-standard input device or linking two dimensions such as the scrolling speed and the zoom factor, as in SDAZ.

THE ORTHOZOOM SCROLLER



(a) (b) (c)

Figure 1. (a) low precision (b) medium precision (c) high precision

The OrthoZoom Scroller (OZ) extends a traditional slider into a 1D multi-scale navigation technique. It behaves like a traditional slider when the mouse is moved within the bounds of the slider. When dragging the mouse outside of the bounds of the slider, it continuously changes the granularity/zoom of the slider (Figure 1). The granularity decreases when the mouse cursor goes farther away from the slider bounds. In other words, moving the mouse along the slider orientation performs a pan whereas moving it orthogonally performs a zoom.

Control area: orthogonal dimension

Let us consider the selection of a value in a range R containing r values with a slider S of s pixels. A value can be selected with a precision r/s, which is equivalent to looking at the range at a zoom factor s/r. Thus, some values are not reachable if $r>s$.

We adjust the input precision by using the orthogonal direction of the slider. The OrthoZoom Scroller has a *control area* greater than the area of its graphical representation. The larger the orthogonal distance between the slider and the cursor, the higher the zoom factor (precision) (Figure 2). Thus, the initial zoom factor can be chosen by starting the drag interaction at any orthogonal position, provided that the whole window is available to OrthoZoom.

We map a maximal zoom factor, Z_{max}, onto the maximal orthogonal distance, D_{max}, and interpolate $[s/r, Z_{max}]$ on the interval $[0, D_{max}]$. Typically, Z_{max} is fixed at 1, i.e. the zoom factor allowing selection of any value in the range. Following Guiard et al. [9], for a displacement of Δx, we apply a zoom of Δz. Applying this zoom consists in multiplying the current zooming factor by Δz, thus, for a position x between 0 and D_{max}, the zoom factor z is:

$$z = \Delta z^{x} \text{ with } \Delta z = \sqrt[D_{max}]{s/r}$$

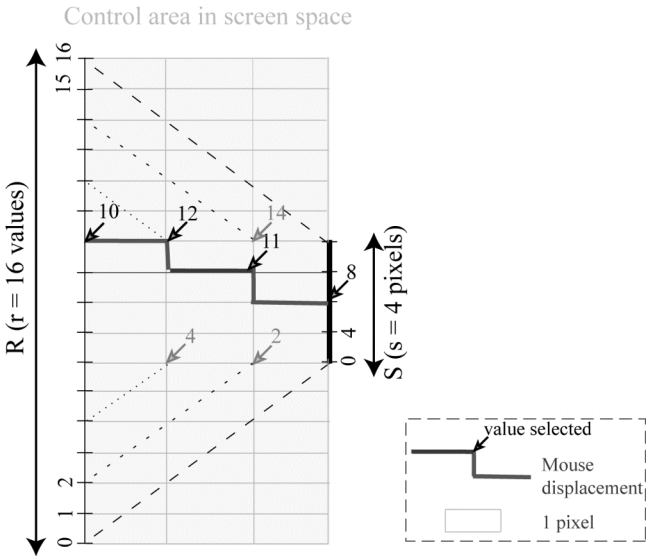


Figure 2: Using the orthogonal direction to adjust the control precision, i.e. the zoom level

Control area: collinear dimension

The control area of an OrthoZoom Scroller is not limited to the graphical bounds of the slider. The space on the orthogonal direction is used to control the zoom factor. However, allowing the user to adjust the zoom factor z raises some problems when $r \times z > s$: the user can not reach the two bounds of the range R by bringing the cursor to the graphical bounds of the slider S since the slider is mapped to a sub-part of R (Figure 3).

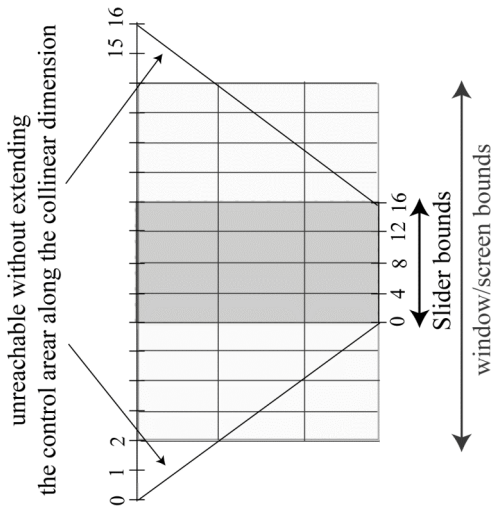


Figure 3. Reachable values within the bounds of the slider and within the bounds of the screen/window

To solve this problem, the control area is not only extended on the orthogonal direction but also on the collinear dimension out of the slider bounds. Since the coordinates of the input device are limited to the screen/window bounds, we trigger a fixed rate scrolling when the mouse is dragged out of the edges of the screen/window.

OZ is a continuous multi-scale technique using a standard input device similar to Speed-Dependant Automatic Zooming (SDAZ). Additionally, SDAZ integrates rate-based scrolling and zooming to overcome the optical flow problems when the user scrolls a document at high speed. The user only controls the scrolling speed, and the system automatically adjusts the zoom factor so that the visual flow remains constant. The user controls the scrolling by a dragging interaction that specifies a vector between the initial point and the current point. The scroll speed is proportional to the length of this vector while the scroll direction is determined by its direction.

EXPERIMENT

We conducted a controlled experiment to compare the efficiency of the OrthoZoom Scroller (OZ) with Speed-Dependant Automatic Zooming (SDAZ) for several indices of difficulty. SDAZ being the only mouse-based multi-scale scrolling technique with continuous control that outperforms standard scrolling interfaces [5]. We designed this experiment to measure the limit performance of the two techniques in pointing tasks, following Hinckley's approach [12].

Subjects

Twelve unpaid adult volunteers, 11 males and 1 female, aged 26 years on average, served in the experiment. The experiment was divided into two parts, one part per technique. We explained participants how to achieve the task using the technique for ten minutes and let them practice 20 randomly-chosen trials before each part.

Apparatus

We used a HP workstation with a 2 GHz Pentium 4, using a 1280×1024 LCD monitor and an optical mouse. The program was written in Java 1.4 using the Piccolo Toolkit[3]. We set the window size to 600×800 pixels and document length of $600 \times 400 \cdot 2^{31}$ pixels.

Task

The participants' tasks involved pointing as fast as possible on successive targets appearing one at a time in a document too large to be viewed at its natural size without scrolling. The participants had to scroll the window vertically to bring the target at the center of the view, indicated by a horizontal black line. An arrow showed the direction of the target from the current view, pointing up when the target was above the view, down when it was below, and left when it was within the view. Because the target was not visible at every zoom factor, a horizontal orange line, insensitive to the zoom factor, showed the target location. The target was also surrounded by concentric circles sensitive to the zoom factor. The pointing task was finished when the target had been kept under the cursor for one second at a zooming factor of 1. The target, initially red, became blue. We used those target indicators (orange line and concentric circles) to avoid the situation where the user was lost in the document (Figure 4). They did not bias the task because the trial was over only when the zoom factor was 1. Furthermore, the orange line was not "snappable", i.e. the system did not use its own knowledge of the location of the target to help pointing at it. Indeed, pointing at the orange line and zooming in would certainly miss the target by several pages in an unpredictable direction for large indices of difficulty (IDs).

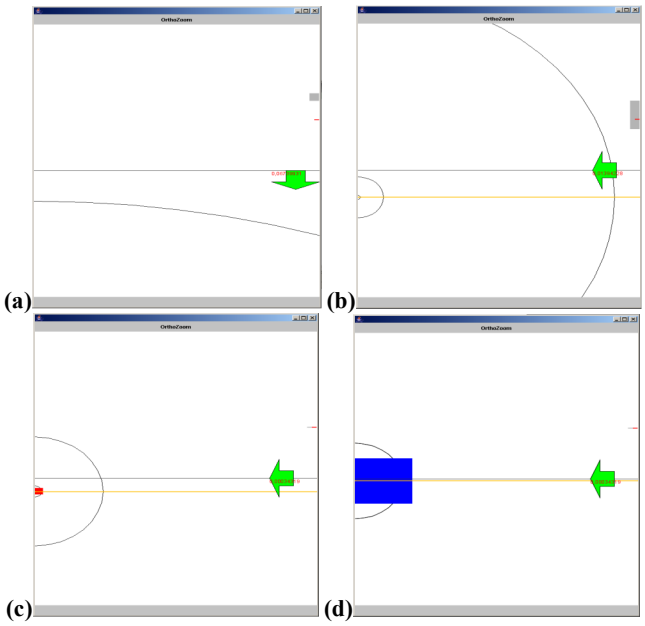


Figure 4. Task: (a) target under the viewport (b) target in the viewport (orange line) but not visible at this scale factor (c) target in the viewport and visible at this scale factor (d) target has been pointed for at least one second

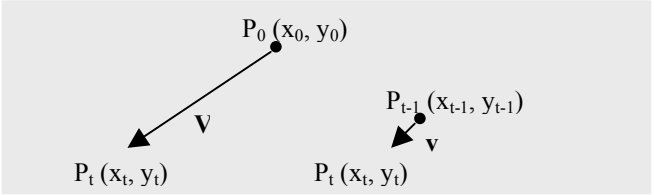
As soon as the target became blue, the ending time of the current trial was logged. Another trial began as soon as the subject pressed the mouse button on the target that he had just reached. This target disappeared and a new target appeared at another location and the beginning time of a new trial was logged.

Hypothesis

We predicted OrthoZoom would be faster than SDAZ in all cases and was a good technique to achieve tasks with large IDs. Furthermore, we predicted that users would prefer controlling the zoom factor themselves, independently of navigation speed.

Hypothesis 1: OrthoZoom is faster for multi-scale pointing than SDAZ

Speed Dependant Automatic Zooming (SDAZ) and OrthoZoom (OZ) both use a mouse displacement to control scrolling speed (precision). To control zoom factor, SDAZ uses the distance between the mouse coordinates (P_t) and the initiating point (P_0) while OZ uses the *x* coordinate of the mouse in the window (x_t). To control scrolling direction, SDAZ uses the sign of the dragging vector (V) while OZ uses the sign of the movement vector (v) (Table 1). Thus, it is easier to change the scrolling direction at any zoom factor with OZ than with SDAZ. With SDAZ, the user must initially reach y_0 to move away in the other direction, losing the current zoom factor. Every intermediate step before reaching y_0 leads to a document movement in the non-desired direction. OZ is likely to be more efficient than SDAZ because changing direction is common in pointing tasks[23].



	Scrolling rate (Δy)	Scrolling direction (+: up, -:down)
SDAZ	$\Delta y = f(y_t - y_0)$	$y_t - y_0$
OZ	$\Delta y = f(x_t)$	$y_{t-1} - y_t$

Table 1. Scrolling rate and direction with SDAZ and OZ

Hypothesis 2: OrthoZoom deals well with difficult pointing tasks

Multi-scale navigation allows dealing with very large documents provided that one can easily control the zoom factor. We hypothesize that using the orthogonal dimension is an effective way to adjust the zoom factor. To test the effect of task difficulty on OZ performance, we used different indices of difficulty in our experiment. The index of difficulty of a pointing task is given by Fitts' law [7]: ID = log(1+D/W) where D is the target distance and W its size. Experiments dealing with very difficult pointing tasks have used IDs up to 30 bits [9, 11]. In our experiment, we have used the following IDs: 10 – 15 – 20 – 25 – 30. Hinckley [12] warns that there may be a "*Device* by *W*" interaction when evaluating scrolling techniques. Thus, we used three different target sizes: 1/10 × H, 1/5 × H, 1/2 × H, where H is the window height (800 pixels). We computed the corresponding values for D using the relation: $D = 2^{ID} \times W$. Thus, in this experiment, we had three independent variables: *Technique*, *ID* and target width *W*.

2 technique conditions (OZ and SDAZ[1])
x 5 IDs (10 – 15 – 20 – 25 and 30)
x 3 W (1/10 – 1/5 and 1/2)
= 30 possible tasks.

We grouped the trials into two identical series, one series per condition, to avoid disturbing subjects with successive changes among techniques. Each participant was exposed to the 2 technique conditions by performing one series with OZ and the other with SDAZ. We computed 6 different series. 2 participants were randomly assigned to a series: one began with the OZ technique while the other began with the SDAZ technique. Thus, we had 2 groups of 6 participants: one group performing the order OZ – SDAZ and the other performing the order SDAZ – OZ. We chose this experimental design to minimize ordering effects.

[1] To calibrate SDAZ, we used Cockburn's formula with k=0.05 and threshold = 20 [6].

There were 2 blocks per series; each block consisted of 45 trials (5 IDs x 3 W repeated 3 times). Presentation order of the trials within a block was randomized. Thus, each subject performed:

2 blocks of 45 trials
x 2 technique conditions
= 180 trials per subject.

For each trial, we logged *Movement Time*, *Release* errors and *Overshoot* count. A *Release* error occurred when a participant released the mouse button without having reached the target. An *Overshoot* occurred when the target passed through the cursor (i.e. horizontal black line) without stopping. We recorded this data to gather information about the strategy used to reach the target.

Hypothesis 3: Users integrate the scrolling and the zoom dimensions

As mentioned earlier, OZ allows users to control zoom factor and panning direction independently. We hypothesize that users can control these two dimensions in an integral fashion [14]. Most interaction styles separate the zooming and panning controls (e.g. the "hand" and "zoom" tools in Adobe's software) or phases [4], leading to a movement represented by the light gray curve of Figure 5 in a space-scale diagram [8]. Since OrthoZoom integrates those controls into one mouse interaction, we expected a smoother curve in the space-scale diagram, closer to the dark gray one. To study the movements in the space-scale diagram, we recorded the mouse positions in window coordinates and document coordinates during each trial.

To summarize, this experiment had four dependent variables: task completion time, number of *Release* errors, number of *Overshoot* and mouse positions. At the end of the experiment, a short questionnaire was administered to collect subjects' preferences. We asked which of the two techniques was preferred and why.

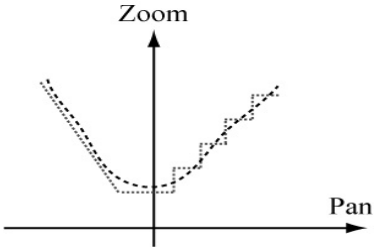


Figure 5. Space-scale diagram

RESULTS

Analysis of variance reveals a significant main effect on Movement Time for Technique ($F_{1,11}$ = 393,094, p < .0001). Figure 6 supports hypothesis 1: OZ is about twice as fast as SDAZ.

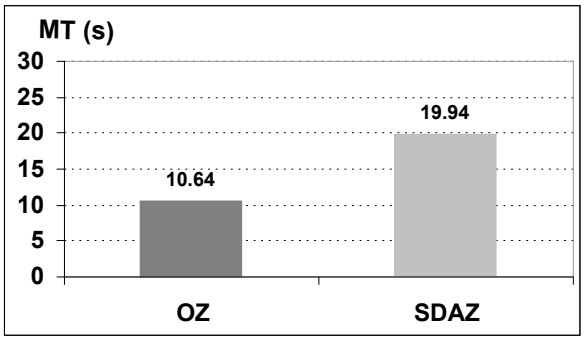


Figure 6. mean Movement Time by technique

As might be expected from Fitts' law, our results reveal a significant main effect on *Movement Time* for *ID* ($F_{4,44}$ = 62,657, p < .0001). However, the *ID* effect comes from *Distance* because there were no significant main effect on *Movement Time* for *Width* ($F_{2,22}$ = 1,004, p = 0,367). This is probably due to the negligible effect of W on the ID value because we were constrained to values of W fitting within the window. There were also a significant effect of *Technique*ID* ($F_{4,44}$ = 6,291, p < .0001) interaction. Figure 7 (plotting *Movement Time* vs. *ID*) supports hypothesis 2: the OZ curve is almost twice as flat as the SDAZ curve revealing that OZ is a promising technique to achieve very difficult pointing tasks, beyond 30 bits.

Figure 8 shows the evolution of *Movement Time* over the 180 trials subjects performed for each *Technique*. The slope of the OZ curve is less than the slope of the SDAZ curve, showing that users are faster at understanding the OZ technique.

Analysis of variance revealed a significant main effect on *Release Errors* for *Technique* ($F_{1,11}$ = 317,918, p < .0001), but not on *Overshoot count* for *Technique* (Figure 9). These numbers reveal more the strategy used by subjects in our experiment than actual errors for both techniques. Subjects used more *Releases* per pointing task with *SDAZ* than with *OZ*. Neither Technique*ID or ID have a significant effect for either measure.

Neither *Technique*ID* interaction or *ID* does not have a significant effect on both measures.

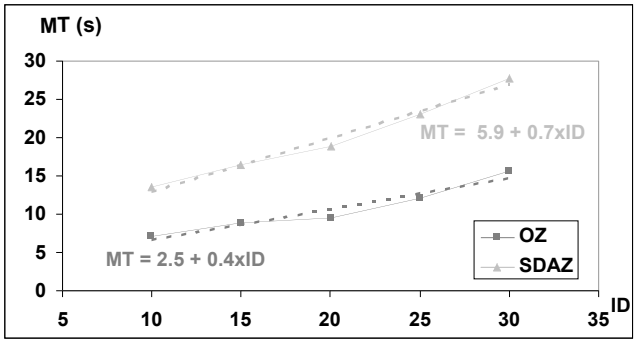


Figure 7. Mean Movement Time vs. ID

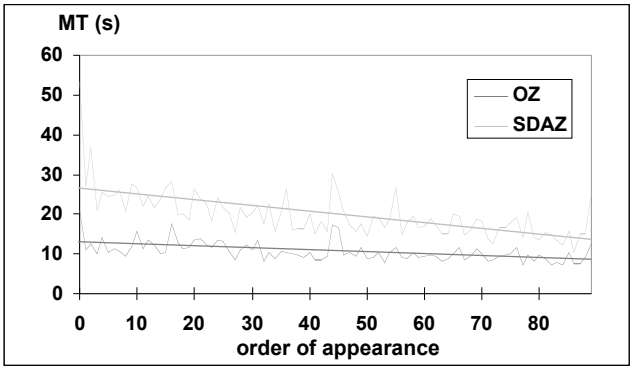


Figure 8. Mean Movement Time vs. order of trial in block

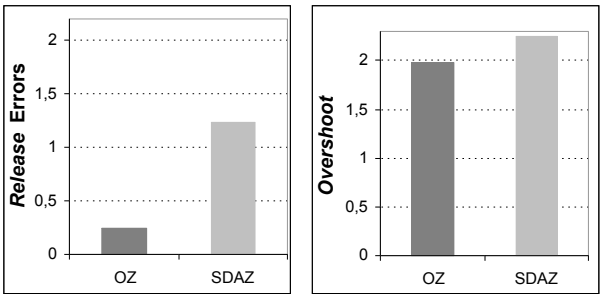


Figure 9. Mean number of errors per technique

We built a Java application to visualize the space scale diagrams. Because plotting a mean of these curves does not have much sense, our application plots them "on demand" by entering the subject number, block number and trial number. It gave us an indication on the integrality of the two dimensions. We collected curves showing that users commonly performed a zoom-out phase (vertical curve up) followed by a panning phase (horizontal curve) ended by a zoom-in phase (vertical curve down). Thus, hypothesis 3 regarding integrality is rejected. The only difference we noticed were that SDAZ curves often presented some vertical lines up and down "cutting" the horizontal line. Space scale diagrams do not include time. Because of the zoom factor effect, a panning phase taking a long time may not be visible in the space-scale diagram because the corresponding panning in document space may be small. To take a closer look at our data, we drew two curves along the time axis: panning curve and zooming curve (Figure 10). Some SDAZ zoom curves present peaks of two types: releasing the mouse button (leading to the maximal zoom factor) and changing direction. SDAZ zoom curves reveal every change of direction while OZ zoom curves only reveal some of them. To change direction, users perform one of the following actions:

- with SDAZ:
  - they release the mouse button, i.e. the zoom factor falls to its maximum;
  - they return to the initiating point to move away from it in the other direction, i.e. the zoom factor increases to 1 then decreases.
- with OZ:
  - they change cursor direction at the current zoom factor, i.e. the zoom factor remains the same;
  - they decrease the zoom factor to reach the level where the target is in the viewport, i.e. the zoom factor changes smoothly.

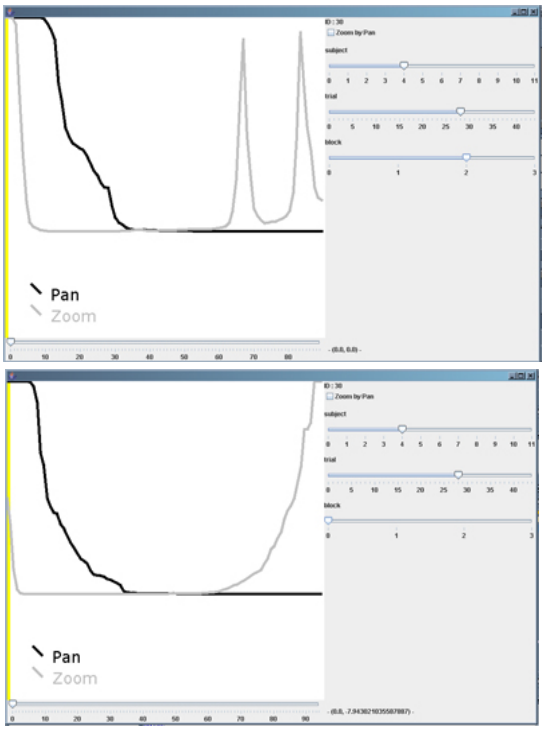


Figure 10. A typical example of SDAZ (top) and OZ (bottom) for the same trial. Pan (black) and Zoom (gray) vs. time (x-axis).

The short survey ending the experiment revealed that all the subjects preferred *OZ*. Subjects did not always succeed to explain why. Among the few remarks we collected, one said "it is enjoyable to control the zoom factor by myself" and another "it is amazing: I do not have the impression to scroll the document with the OZ technique".

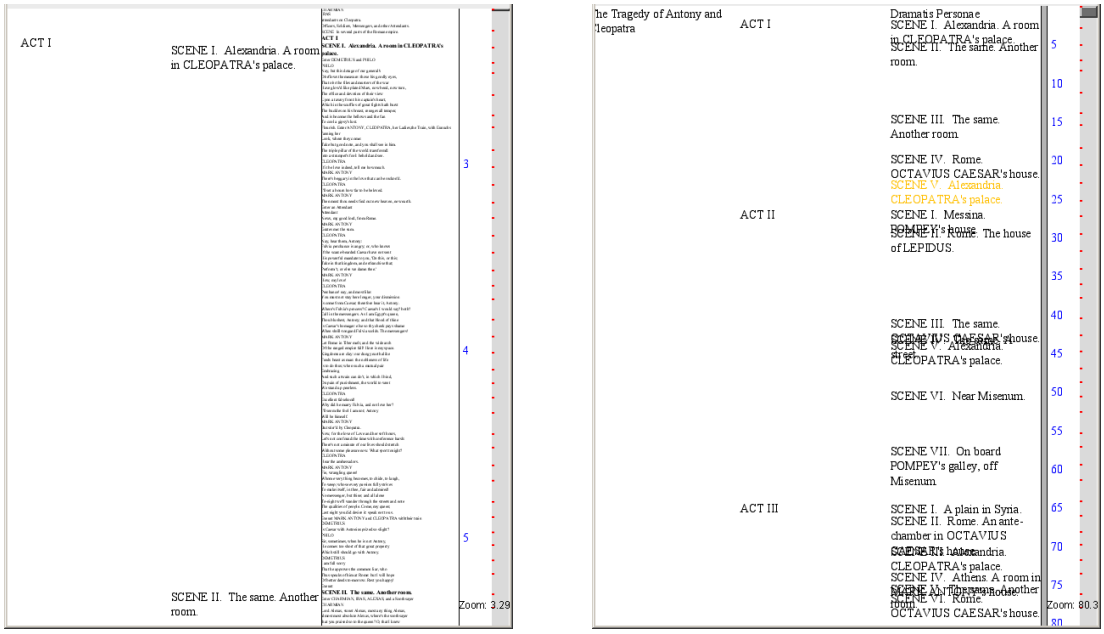


Figure 11. Multi-scale Navigation into the Shakespeare's plays: the Table of Contents appears to the left of the text when it scales down. On the right, only the structure remains visible.

BROWSING LARGE TEXT DOCUMENTS WITH ORTHOZOOM SCROLLER

Using the OrthoZoom Scroller for scrolling a large document requires adaptations of the feedback since a one pixel displacement of the thumb produces a large jump of the view. For example, a document containing the 37 plays of Shakespeare has roughly 150,000 lines of text. Assuming a comfortable text page displaying 40 lines with a window and scrollbar height of 1000 pixels, a one pixel displacement of the thumb will jump almost four pages. Traditional multi-scale scrolling techniques use a zooming interface where the document's scale factor is synchronized with the navigation scale factor so that the "optical flow" of the document becomes acceptable when scrolling. Zooming works well when the document is itself multi-scale, such as the image of the earth that remains meaningful at any scale factor. However, scaling-down a textual document turns it into a narrow illegible line. For example, scrolling and navigating at the level of a play requires a scale of approximately 1/150, leaving too few pixels to distinguish anything in the text.

To overcome this problem, we have implemented a Multi-Scale Table Of Contents (MSTOC) displayed on the left of the scaled-down text, as shown in Figure 11

Each section entry of the table of contents is displayed at a constant size but is aligned vertically with the position it has in the text, guiding the user during navigation. When the scale factor decreases, the MSTOC shrinks vertically and some entries at one level eventually collide, as seen in Figure 11-b. When this happens, we scale down the whole level until it becomes unreadable and is removed from the view. This scaling-down leaves room for higher-level entries in the MSTOC that start to appear to the left.

The MSTOC also provides snapping: when the OrthoZoom is active, the highest-level entry aligned with the cursor is highlighted. Releasing the OrthoZoom when an entry is highlighted scrolls the view at its exact position. Snapping is important because overshooting by one pixel in the scrollbar can lead to overshooting several pages in the document.

The MSTOC displays page numbers aligned with their vertical positions. These page numbers can be extracted from the document when available or computed from the view size; they provide regular tick-marks which help interpret the displayed scale and the document size. When the view-scale changes, we change the step of the displayed page numbers: every page, every two pages, five, ten etc. Page numbers are snappable, with a lower priority than the MSTOC entries.

Quickly pressing and releasing the pointer on the scrollbar triggers a quick animation showing the table of contents of the whole document and zooming back to the original position, providing a quick animated focus to context to focus journey.

All these interactive navigation features integrate the functions available in popular document readers. For example, Adobe Acrobat Reader provides at least 5 different ways to navigate through a document that could be replaced by OZ interaction: "hand" tool, next/previous button, page thumbnail selection, bookmark selection, page number entry as text field. Depending on the *x* position where OZ is used, the user can scan the document at the "hand tool" level, thumbnail level or bookmark/structure level. Navigating to a specified section can be done using zooming and snapping when the section is visible. Navigating at a specific page location can be done by snapping the page number. All these interactions integrate

smoothly in one place: the main window, which is the object of interest according to the rules of direct manipulation [21].

DISCUSSION

The evaluation did not take into account the problem of motion blur and focused only on the interaction. The reason for separating the interaction task from the visual perception task is that the latter is very dependent on the nature of the displayed information (map, graphics, text, etc.). The pointing task is relevant for the situation where the user knows exactly where to go. In realistic situations, the displayed information should provide indications on the relative location of the target. With the MSTOC, if the user wants to reach scene 5 of Act 3 of Macbeth, he is aware of his current position and can decide to scroll up or down at any zoom level.

CONCLUSION

We have presented the OrthoZoom Scroller, a technique for scrolling and pointing in large 1D documents using only a mouse. We have shown that – with IDs up to 30 bits – OZ performed two times faster than SDAZ which is known as the fastest multi-scale navigation technique using a standard input device. SDAZ and OZ follow Fitts' law but OZ curve is almost twice flatter than SDAZ curve: the difference of time between SDAZ and OZ increases with the ID.

With larger online resources available on the Web, techniques scaling to this level have a great potential. For example, navigating on the human genome from the base-pair scale to the whole chromosome has an ID of 32 bits. Navigating on the whole Google corpus requires about 33 bits.

We presented an application of OZ to navigate in large textual documents and showed how to integrate it with semantic zooming and snapping. We believe OZ could be integrated with current Web browsers and e-Books readers to improve navigation without requiring changes in layout or overall interaction.

Since OZ augments and replaces scrollbars and sliders, integrating it in future applications will ensure the scalability required by the continuous growth of information.

ACKNOWLEDGMENTS

The authors would like to thank the whole in|situ| lab. and especially Michel Beaudouin-Lafon, Renaud Blanch and Wendy Mackay for their useful comments and suggestions. The authors also want to thank Andy Cockburn and Ravin Balakrishnan on their help about respectively SDAZ and Zliding and Eric Lecolinet for the fruitful discussions about multi-scale interfaces.

REFERENCES

1. Ahlberg, C. and Shneiderman, B. The alphaslider: a compact and rapid selector. in *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, ACM Press, Boston, Massachusetts, United States, 1994, 365-371.
2. Ayatsuka, Y., Rekimoto, J. and Matsuoka, S. Popup vernier: a tool for sub-pixel-pitch dragging with smooth mode transition in *Proceedings of the 11th annual ACM symposium on User interface software and technology* ACM Press, San Francisco, California, United States 1998 39-48
3. Bederson, B.B., Grosjean, J. and Meyer, J. Toolkit Design for Interactive Structured Graphics *IEEE Trans. Softw. Eng.* , *30* (8). 535-546
4. Bourgeois, F., Guiard, Y. and Lafon, M.B. Pan-zoom coordination in multi-scale pointing in *CHI '01 extended abstracts on Human factors in computing systems* ACM Press, Seattle, Washington 2001 157-158
5. Cockburn, A. and Savage, J., Comparing Speed-Dependent Automatic Zooming with Traditional Scroll, Pan and Zoom Methods. in *in People and Computers XVII (Proceedings of the 2003 British Computer Society Conference on Human-Computer Interaction.)*, (Bath, UK, 2003), Springer-Verlag, 87-102.
6. Fekete, J.-D. The InfoVis Toolkit. in *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'04) - Volume 00*, IEEE Computer Society, 2004, 167-174.
7. Fitts, P.M. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, *47* (6). 381-391.
8. Furnas, G.W. and Bederson, B.B. Space-scale diagrams: understanding multiscale interfaces in *Proceedings of the SIGCHI conference on Human factors in computing systems* ACM Press/Addison-Wesley Publishing Co., Denver, Colorado, United States 1995 234-241
9. Guiard, Y., Beaudouin-Lafon, M., Bastin, J., Pasveer, D. and Zhai, S. View size and pointing difficulty in multi-scale navigation in *Proceedings of the working conference on Advanced visual interfaces* ACM Press, Gallipoli, Italy 2004 117-124
10. Guiard, Y., Beaudouin-Lafon, M. and Mottet, D., Navigation as Multiscale Pointing: Extending Fitts' Model to Very High Precision Tasks. in *CHI*, (1999), ACM Press, 450-457.
11. Guiard, Y., Bourgeois, F., Mottet, D. and Beaudouin-Lafon, M., Beyond the 10-bit barrier: Fitts' law in multiscale electronic worlds. in *People and Computers XV Interactions without frontiers. Proceedings of IHM-HCI 2001*, (Lille, France, 2001), Springer-Verlag, 573-587.
12. Hinckley, K., Cutrell, E., Bathiche, S. and Muss, T. Quantitative analysis of scrolling techniques in *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves* ACM Press, Minneapolis, Minnesota, USA 2002 65-72
13. Igarashi, T. and Hinckley, K. Speed-dependent automatic zooming for browsing large documents in

Proceedings of the 13th annual ACM symposium on User interface software and technology ACM Press, San Diego, California, United States 2000 139-148

14. Jacob, R.J.K., Sibert, L.E., McFarlane, D.C. and M. Preston Mullen, J. Integrality and separability of input devices. *ACM Trans. Comput.-Hum. Interact.*, *1* (1). 3-26.
15. Lehikoinen, J. and Röykkee, M., BinScroll: a rapid selection technique for alphanumeric lists. in *CHI'00 Extended Abstracts on Human Factors in Computing Systems*, (The Hague, The Netherlands, 2000), ACM Press, 261-262.
16. Masui, T. LensBar - Visualization for Browsing and Filtering Large Lists of Data in *Proceedings of the 1998 IEEE Symposium on Information Visualization* IEEE Computer Society, North Carolina 1998 113-120
17. Masui, T., Kashiwagi, K. and George R. Borden, I. Elastic graphical interfaces to precise data manipulation in *Conference companion on Human factors in computing systems* ACM Press, Denver, Colorado, United States 1995 143-144
18. Pook, S., Lecolinet, E., Vaysseix, G. and Barillot, E. Control menus: excecution and control in a single interactor in *CHI '00 extended abstracts on Human factors in computing systems* ACM Press, The Hague, The Netherlands 2000 263-264
19. Ramos, G. and Balakrishnan, R., Fluid interaction techniques for the control and annotation of digital video. in *Proceedings of the 16th annual ACM symposium on User interface software and technology*, (Vancouver, Canada, 2003), ACM Press, 105-114.
20. Ramos, G. and Balakrishnan, R., Zliding: fluid zooming and sliding for high precision parameter manipulation. in *Proceedings of the 18th annual ACM symposium on User interface software and technology*, (Seattle, WA, USA, 2005), ACM Press, 143-152.
21. Shneiderman, B., Direct Manipulation: a Step Beyond Programming Languages. in *IEEE Computer*, (1983), 57-69.
22. Smith, G.M. and schraefel, m.c. The radial scroll tool: scrolling support for stylus- or touch-based document navigation in *Proceedings of the 17th annual ACM symposium on User interface software and technology* ACM Press, Santa Fe, NM, USA 2004 53-56
23. Soukoreff, R.W. and MacKenzie, I.S. Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI *Int. J. Hum.-Comput. Stud.* , *61* (6). 751-789
24. Zhai, S. and Smith, B. Multi-Stream Input: An Experimental Study of Document Scrolling Methods. *IBM Systems Journal*, *38* (4). 642-651.