A Model of Navigation for Very Large Data Views

Michael Glueck[†‡], Tovi Grossman[‡], and Daniel Wigdor[†]

[†]Dept. of Computer Science, University of Toronto [‡]Autodesk Research, Toronto, Canada

ABSTRACT

Existing user performance models of navigation for very large documents describe trends in movement time over the entire navigation task. However, these navigation tasks are in fact a combination of many sub-tasks, the details of which are lost when aggregated. Thus, existing models do not provide insight into the navigation choices implicit in a navigation task, nor into how strategy ultimately affects user performance. Focusing on the domain of data visualizations, the very large documents we investigate are very large *data views*. We present an algorithmic decision process and descriptive performance model of zooming and panning navigation strategy, parameterized to account for speed-accuracy trade-offs, using common mouse-based interaction techniques. Our model is fitted and validated against empirical data, and used to evaluate proposed optimal strategies. Further, we use our model to provide support for interaction design considerations for achieving performant interaction techniques for navigation of very large data views.

Keywords: Navigation model; panning; scrolling; very large data view; zooming.

Index Terms: H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

1 INTRODUCTION

As the size of digital documents and datasets continues to grow, so does the need for efficient ways to navigate them. While traditional scrolling user interfaces (UI) include some utility to support large documents, research suggests that changes in viewing position beyond just a few screen widths can be significantly more efficiently carried-out by zooming-out, and then back -in to the desired position in the document [16] (see Figure 1). Thus, for sufficiently large documents, some degree of zooming is essential for digital content consumption. In fact, many applications now offer some degree of zooming interaction, in addition to scrolling, to better support the growing size of documents. Productivity software, such as Microsoft Office, readers, such as Adobe Reader, and even web browsers all provide zooming and scrolling controls to aid navigation of large documents.

Our particular domain of interest lies within the visualization of very large single dimensional data, which generally display numerical data along a horizontal axis. We will refer to these applications as *horizontal data viewers*, and associated documents as *data views*. Horizontal data viewers can be found for a variety of data, such as financial (Google Finance), weather (WeatherSpark), and social metrics (SumAll). These applications display data at different levels-of-detail based on zoom level, and as such, zooming and panning navigation becomes a first-order interaction. As such, we were curious whether users are efficiently zooming and panning while performing common navigation tasks, such as seeking a known location within a dataset, or searching for an unknown target using knowledge of landmarks in the data.

Zooming and panning[1] are fundamental and ubiquitous interaction elements in desktop computing contexts, and, individually, have long been the focus of research in Human-Computer Interaction. A body of research has specifically investigated user interactions in zooming interfaces, where both zooming and panning are used to navigate. Guiard and Beaudouin-Lafon studied target acquisition in zooming interfaces and developed a holistic performance model [17]. However, their model does not elaborate on the contributions of individual zooming and panning operations to the whole, and thus does not shed light on the user choices or any strategies that may be involved.

Navigation in zooming interfaces is a complex interplay of sequential zooming and panning operations, each dependent on the previous. While zooming can be implemented around a fixed viewport position or cursor position, the latter only differs in that panning is replaced by pointing. In both cases, parameterizing this sequence will foster a systematic exploration and comparison of *navigation strategy* as more than just the sum of its parts – elucidating the trade-offs integral to optimal navigation. Until now, optimal strategies have been posited, such as by Furnas and Bederson [16], but have never been empirically validated. As shown in Figure 1, a near target requires no zooming, only scrolling (left). The "naïve optimal" strategy is zooming-out until the target is visible in the viewport (middle). The posited "optimal" strategy is to undershoot the target: zooming-out until the target is near the viewport, but not within view (right).

In this paper, we explore navigation strategies in the context of mouse-based desktop interaction with very large data views. We theorize a user's navigation strategy can be characterized as a sequence of interactions with trade-offs that can be parameterized; thus this work offers four key contributions. First, we present an algorithmic process highlighting the decisions contributing to these trade-offs, where sub-optimal choices are linked to negative impacts on overall performance. Second, we develop a parameterized strategy model that describes the zooming and scrolling components of navigation. Third, we derive costing functions for this model based on empirical results, and evaluate proposed optimal strategies. Finally, we present interaction design considerations to guide the development of scrolling and zooming interface navigation techniques for very large data views.

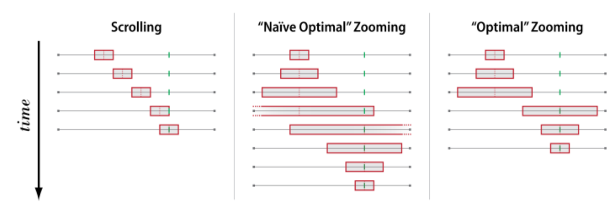

Figure 1. Navigation strategies posited in [16]: document represented by a horizontal line, target by a vertical mark, and viewport by a rectangle. Larger viewports correspond to a lesser zoom level.

[1] As in [17], we do not distinguish between *scrolling* and *panning*. For the purposes of this paper, both terms will be used interchangeably.

† {mglueck|dwigdor}@dgp.toronto.edu
‡ tovi.grossman@autodesk.com

2 RELATED WORK

Panning and zooming have been a part of UI design since the early 1960's [28]. "Infinitely" zooming interfaces were first introduced with the Pad [25] and Pad++ [4] systems, supporting interaction with multi-scale documents. We focus our review here on works which investigate how users zoom and pan content ('documents'), rather than of specific user interface controls.

Furnas and Bederson presented space-scale diagrams as a means of understanding zooming and scrolling interfaces [16]. Based on a visual information complexity metric, they presented intuition towards an optimal zooming and panning trajectory, but these were not empirically validated.

A method developed by van Wijk and Nuij calculates the optimal zooming and panning trajectory, while maintaining visual context, to smoothly animate transitions between views in a zooming interface [29]. In contrast, we analyse interactively-defined user trajectories to identify where sub-optimal decisions are made and to quantify these deviations from the optimal.

Text document navigation patterns were characterized in a log study of Microsoft Word and Adobe Reader usage [1]. In map interfaces, navigation patterns have also been analysed to interpret task completion time differences between conditions [21] and the percentage of time spent zooming and panning has been compared between different tasks [23]. Our work seeks to explicitly parameterize the components of these navigation patterns and to quantify the impacts of deviations from the optimal strategy.

Several techniques support simultaneous zooming and panning navigation. Speed-dependent automatic zooming (SDAZ) [22] is a technique that automatically changes the zoom level with respect to the speed of scrolling, controlled through velocity-based input. This technique has been shown to be more efficient than scrolling with manual zooming [27] and traditional scroll, pan, and zoom operations [10]. Zooming and scrolling control have been combined in novel navigation techniques for position-based input; OrthoZoom [3] is a recent example and empirical evaluations have shown that OrthoZoom outperforms SDAZ. While these advanced interaction techniques offer more efficient navigation of very large documents, they do not represent the interaction techniques presently found in a majority of applications "in the wild". Moreover, this initial investigation focuses on the speed-accuracy trade-off between zooming and panning when they are carried out in sequence, which would not be possible by using a technique where they occur simultaneously.

Performance models for scrolling have been presented and shown to be subject to Fitts' Law [14] for scrolling to targets of known location [6][20]. Scrolling to targets of unknown location has been shown to follow either a linear model using scrollbars [2] or Fitts' Law in a peephole pointing task [6]. The combination of simultaneous zooming and scrolling has been modelled and also shown to follow Fitts' Law [17]. Fitts' Law-based models provide us with valuable intuition of the total time we can expect a user to take to complete a task: an aggregate of all steps involved. It does not, however, provide insight into what those individual steps are, the sequence in which they occur, nor the parameterization of each, which is the focus of the current work.

While early studies [20][30] used unmodified and unconstrained input, subsequent studies have mapped mouse position directly to scroll position to remove acquisition times [2][11][17], or have used bimanual input techniques to support simultaneous zooming and panning [17][27]. Our work differs in two ways. First, since we are interested in studying sequential zooming and panning when fully controlled by a user, we analyse zooming and panning operations distinctly from each other, as well as the time it takes to switch between operations. Similar "traditional" sequential zooming and panning input scenarios have been previously studied [10], but only as an aggregated measure of total navigation time. Second, we are interested in typical mouse-based desktop situations where bimanual interaction techniques of the type described in prior work are uncommon.

In summary, while overall performance has been modelled, a deeper understanding of zooming and panning strategy is lacking. In particular, we explore how users conceptualize the navigation process, which strategies they employ, and what factors impact their choice of strategy and performance. Moreover, in developing generalized performance models, highly abstracted tasks were evaluated with very controlled input. In the present work we attempt to bridge this gap and validate a theoretical model by combining an abstracted navigation task with more ecologically valid input techniques.

3 TASK ENVIRONMENT AND CONSIDERATIONS

To investigate zooming interface navigation strategies, we consider an abstracted zooming and panning task environment that could generalize to most real-world horizontal data viewers. The environment consists of a horizontally-oriented document viewport with a scroll bar underneath, and zoom slider to the right (see Figure 2). Both provide only visual feedback of data view position and zoom level in the viewport and are non-interactive.

The document is an abstract 1-D data view, containing a red target line that remains the same width at all zoom levels. A light-blue goal zone appears in screen-space, fixed to the centre of the viewport. To mitigate *desert fog*, user disorientation when zooming and panning due to insufficient *critical zones* or cues [24], our data view is augmented with evenly spaced lines that subdivide and coalesce at different zoom levels. The zoom percentage is also displayed above the viewport, and the dark grey bars bounding the upper and lower edges of the data view change colour to signal when the content is zoomed back to 100%.

This interface supports a view-pointing task [18], previously applied to model zooming and panning performance [17], where the target must be found and positioned within the goal at a specified zoom level. In our variation of this task, the start and end zoom levels are identical, and zooming occurs with respect to the centre of the viewport, the *zoom pivot*. To successfully complete the task, one must zoom-out to find the target, and then perform a combination of interleaved zoom-in and corrective scroll operations to re-position it correctly.

Interleaved zooming-in and panning are necessitated by a phenomenon we have termed *zoom-in drift* – the appearance that the target shifts away from the zoom pivot when zooming-in. To zoom-out, document space must be compressed within screen-space, resulting in the binning of many document-space pixels into a single screen-space pixel. This process must be reversed when zooming-in. Even if the target appears to be positioned exactly at the zoom pivot prior to zooming-in, it will drift away due to this de-binning of pixels.

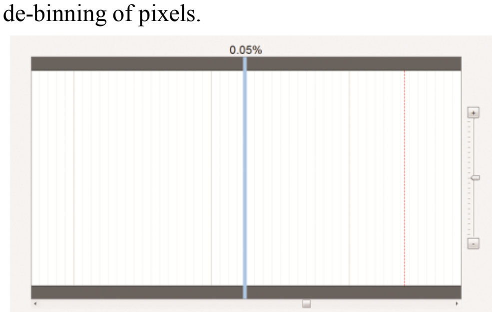

Figure 2. Experimental interface: goal shown in blue, target in red.

We now summarize factors important to zooming and panning tasks that have guided our model and experimental design:

Input Device: Our work focuses on single-handed, mouse-based input, including the mouse-wheel to increase external validity.

Viewport Sizes: Viewports larger than 40px have minimal impact on performance in zooming and panning tasks [17]. For this reason, in our study, we do not vary the viewport size.

Viewport Orientation: Based on our interest in large data visualization interfaces, we use a horizontal document orientation.

Interaction Techniques: We focus on 3 techniques for zooming and panning: click-and-drag pan, mouse-wheel scroll, and mouse-wheel zoom (with CTRL modifier). These are variants of techniques common to horizontal data viewer applications, as well as other software, such as readers and productivity suites.

Familiarity with Content: Knowledge of the target location could alter a user's strategy. Since we seek baseline performance, our work considers only navigation within unfamiliar content.

Target Saliency: In our domain, information is displayed at varying levels-of-detail based on the zoom level, ensuring that an aggregated form of the target will be visible at any zoom level. Thus, we assume the target is equally discernible across all zoom levels and display the target with fixed width in screen-space. To minimize the impact of visual search, no distractors are included.

Document Bounds: The scroll bar visually communicates the relationship between the total document size and portion displayed in the viewport. To ensure accurate scroll bar feedback, we limited panning to the document bounds.

Zoom Pivot: We use a viewport centre-based zoom pivot as a generalization of the fixed-position zoom pivots typically found in horizontal data viewers. Previously, centre-based zooming has been used in a simultaneous zooming and panning task [17]. We note that the efficiency of zooming interfaces is reduced when document-bounded scrolling is paired with a fixed zoom pivot: zooming at document edges is encumbered since the magnitude of corrective scrolls is limited to the document bounds. To avoid this pitfall, we choose trial conditions that bypass document edges.

4 A Model of Optimal Performance

Furnas and Bederson posited that a "naïve optimal" strategy is to zoom-out until the target is visible before zooming back -in [16]. This was further refined by considering nearby targets separately from distant ones. They calculated that the optimal strategy for near targets is scrolling without zooming. By extension, the optimal strategy for distant targets was zooming-out until the target was *near* the viewport (rather than visible on-screen), then scrolling to the target, before finally zooming back -in. "Near" was estimated to be within 1-3 viewport widths (see Figure 1).

Naturally, such a model presumes that identifying the target is consistently difficult across zoom levels. This is true to a greater or lesser extent depending on the domain of the document. In the present work, we begin with the same assumption, but ultimately account for target salience in our strategy model.

Empirical results lend support to the findings of Furnas and Bederson: users prefer scrolling without zooming when targets are near [17]. The insight here is the existence of a crossover point where zooming-out is faster than scrolling. We suggest users make this distinction based on their familiarity with the content, spatial reasoning ability, and aptitude using navigation operations.

4.1 An Algorithmic Decision Process

We developed a decision process to conceptualize the iterative manner by which a user searches for a target in a zooming and panning interface (see Figure 3). Our decision process accounts for both choices and interactions – regardless whether or not optimal. At each choice (diamond), a user may misjudge the best decision and at every interaction (rectangle), a user may navigate sub-optimally. This process clarifies the distinct components of the navigation strategy, and provides a framework to support the discussion of user choices and interactions.

We highlight three phases, *Z-O*, *S*, and *Z-I*, which represent the zoom-out, scroll, and zoom-in steps of navigation, respectively. Note that a user is not required to engage in any zooming. If the user chooses not to zoom-out in Z-O, they will not reach Z-I, and the process elegantly reduces to simply describe scrolling in S.

Z-O encapsulates the zoom-out phase of interaction. The critical decision, *Is target close enough?*, is aided by knowledge of the target location. A misjudgement leading to an error, for example failing to zoom-out far enough, would require restarting the decision process. Without any knowledge of target location, only the "naïve" optimal strategy can be employed. However, whether or not a target is salient at all zoom levels affects a user's ability to follow the "naïve" optimal strategy. For example, content in multi-scale documents may only be visible at a particular zoom level. This leads us to consider a *target saliency threshold*, an upper bound on how far a user can zoom-out given content, which can undermine the "naïve" optimal strategy. Thus, we posit that both content familiarity and target saliency threshold impact the accuracy of choices made in Z-O.

S and Z-I, on the other hand, encapsulate the scroll and zoom-in phases of interaction. Judgment of the critical decision, *Has target shifted enough?*, is a complex trade-off in screen-space pixels between speed, accuracy, and magnitude of scrolling, additionally complicated by zoom-in drift, since there is a limit to how far one can zoom-in before having to reposition the target close to the zoom pivot. For example, if a user chooses a low threshold for "shifted enough", more scroll and zoom-in iterations would be required. Conversely, choosing a higher threshold would result in fewer scroll and zoom-in iterations, but at the increased risk of losing the target through desert fog. A performance balance must exist between "many small inaccurate scrolls and short zoom-ins" and "fewer larger accurate scrolls and long zoom-ins". Thus, we posit that a user's spatial reasoning ability and motor skills impact the accuracy of choices made in S and Z-I.

Sub-optimal navigation is accounted for within the decision process. However, major errors, such as getting lost in desert fog or completely losing track of the target, require restarting the de-

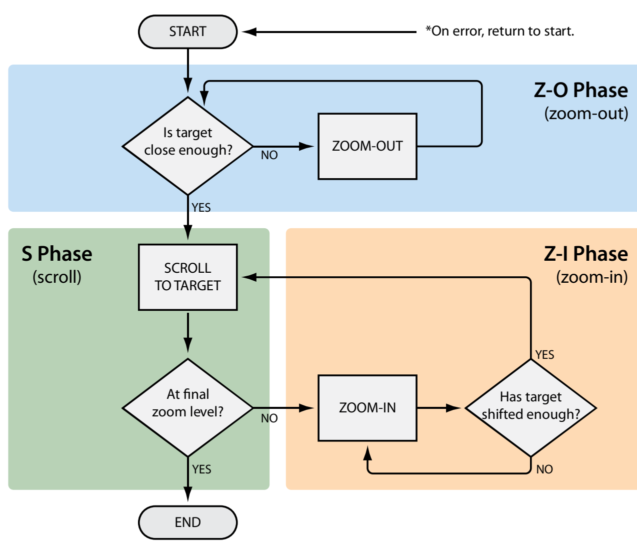

Figure 3. User decision process for zooming and scrolling navigation, divided into phases. An error results in restarting the process.

cision process altogether – a manner of recursion. We feel this accurately reflects how users approach navigation problems and account for errors in the real world. If, for example, a user zooms-in too fast, such that the target is out of range, they will need to decide whether the target is close enough to scroll to, or if they will zoom-out once again to relocate it.

Not only does this decision process underscore the complex iterative trade-offs involved in zooming and scrolling navigation, it also provides intuition for improving the choices a user makes at each step. For example, providing feedback about target location and designing interaction techniques that supplement both a user's motor and spatial abilities may help prevent interaction errors and improve overall performance.

4.2 A Zooming-Scrolling Strategy Model

Based on the intuition of our decision process, we develop a parameterized strategy model to calculate the expected navigation task completion time. The user choices described in the decision process become inputs to costing functions. The costing functions, in turn, account for both visual search and interaction time of each phase of interaction. This strategy model will allow us to evaluate both the trade-offs in strategy, such as between zoom and pan, and global contextual limitations, such as the target saliency threshold. The proposed model builds on rationale akin to coarse, high-level, predictive engineering models, such as KLM [7], but takes into account weighted factors for each method of navigation. This weighting enables the model to not only predict overall performance time, but also explain how the user performs during each portion of the navigation.

Without loss of generality, our model assumes a view-pointing task in a one-dimensional horizontal data view that can be zoomed and panned. For simplicity of navigation operation segmentation, zooming operations function with a viewport-centred zoom pivot, and the goal position is at the centre of the viewport. The scope of the model as presented is expected to include any interface with an arbitrary fixed zoom pivot position and arbitrary goal position.

The model calculates the task completion time, T, in milliseconds (ms) for a particular navigation strategy. A navigation strategy is defined as a set of 5 inputs to the model given 2 environmental constraints. We also introduce 2 descriptive variables to aid in the derivation. Table 1 provides an overview of all the terms used in the model, while Figure 4 illustrates the relationship between the inputs to the model.

The derivation of T is:

$$T = Cost_{ZoomOut} + Cost_{Scroll} + Cost_{ZoomIn} + Cost_{CorrScroll} \qquad (1)$$

While our decision process defines 3 phases of interaction, our strategy model separates interaction into 4 components: *ZoomOut*, *Scroll*, *ZoomIn*, and *CorrScroll*. The S Phase is represented by two components, the first scrolling prior to zooming-in (*Scroll*), and the second scrolling during zooming-in (*CorrScroll*). This distinction allows the strategy model to account for situations where a user only scrolls and does not zoom-out, foregoing zoom-ins and corrective scrolls – calculating time to scroll alone.

By iteratively searching the space of task completion times using different parameter values, optimal task completion times and parameter values can be established. In this section, we describe the logic behind these parameterizations and costing functions. Later, we empirically derive values for each, and validate the model. We start by explaining the derivation of these parameters, and follow with a discussion of the costing functions.

4.2.1 Zooming-Out

Our model is based on the relative position of the viewport and the target at the start of a given task. We define D as the distance to the target from the goal, in document pixels. In our case, the goal is coincident with the centre of the viewport (see Figure 5).

We define V as the viewport size (horizontal) in screen pixels and z as the zoom level. The function $R(z)$ defines the number of document pixels displayed in the viewport in screen-space pixels for a given zoom level z (see Figure 6).

$$R(z) = V \cdot 2^z \qquad (2)$$

The document is viewed at "actual size" (zoomed to 100%) when $z = 0$. At this zoom level, screen-space pixels and document pixels are equivalent, so $R(0) = V$ (see Figure 6a). The document is decreased to 50% when $z = 1$, and so on. In general, zooming-out causes z to increase and zooming-in to decrease.

We define α as the actual zoom level achieved by the user during the zoom-out operation, given that users may either undershoot the zoom (perhaps "optimally", as posited [16], or limited by the target saliency threshold) or overshoot the zoom.

Constr.	Name	Measures	Eq #
$\boldsymbol{D}$	Target distance	Distance to target in document pixels	--
$\boldsymbol{V}$	Viewport size	Viewport size (width) in screen-space pixels	--
Inputs	**Name**	**Measures**	**Eq #**
$\boldsymbol{\alpha}$	Actual zoom-out	Number of levels zoomed-out	--
$\boldsymbol{\sigma}$	Scroll accuracy	Average screen-pixel distance to zoom pivot at end of a zoom-in step	--
$\boldsymbol{\rho}$	Scroll magnitude	Average screen-pixels scrolled for each zoom-in step	--
$\boldsymbol{\Delta z}$	Zoom-in delta	Number of levels zoomed-in at zoom-in step	(5)
$\boldsymbol{n}$	Zoom-in steps	Average number of zoom-in steps	(6)
Descr.	**Name**	**Measures**	**Eq #**
$\boldsymbol{z'}$	"Naïve optimal"	The "naïve optimal" zoom-out level	(3)
$\boldsymbol{m}$	First scroll ratio	The ratio of the viewport scrolled after zooming-out	(7)

Table 1. Summarizing meaning of strategy model terms: environmental constraints, inputs, and descriptive variables.

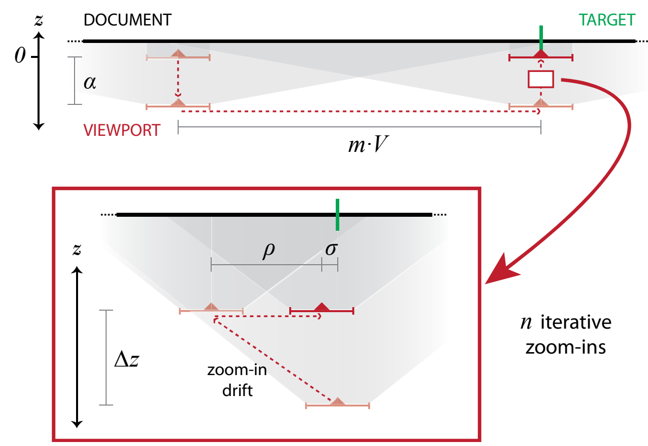

Figure 4. Diagram illustrating strategy model inputs, descriptions of which can be found in Table 1. First, a user zooms-out $\boldsymbol{\alpha}$ zoom levels, then scrolls $\boldsymbol{m \cdot V}$ px. Next, they zoom-in and corrective scroll $\boldsymbol{n}$ times, averaging $\Delta\boldsymbol{z}$ zoom levels and $\boldsymbol{\rho}$ px at each iteration. The precision of corrective scrolling is described by $\boldsymbol{\sigma}$ px, the average offset of the target from the goal at the end of each corrective scroll.

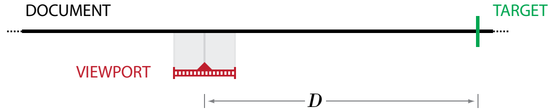

Figure 5. Diagram illustrating the distance to the target, $\boldsymbol{D}$, based on the initial the viewport position, in document pixels.

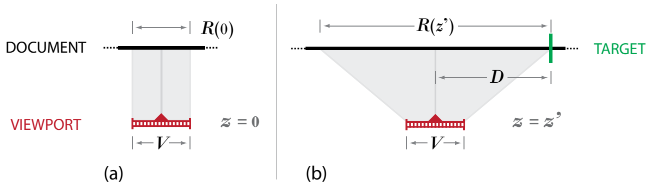

Figure 6. Diagrams (a) and (b) illustrate the viewport size, $\boldsymbol{V}$, and the range function, $\boldsymbol{R}(\boldsymbol{z})$, in screen-space pixels.

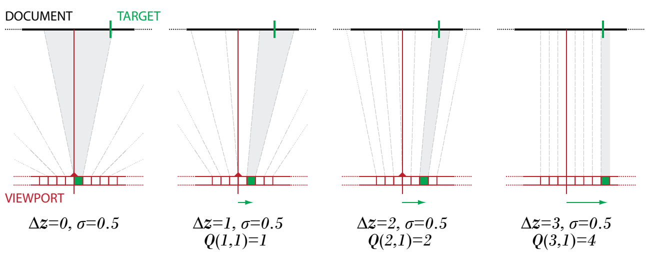

Figure 7. Diagram illustrating the progression of the zoom-in drift function, $\boldsymbol{Q}(\boldsymbol{\Delta z}, \boldsymbol{\sigma})$, in screen-space pixels.

While not used in model calculations, we define z' as the "naïve optimal" zoom level [16], the minimum zoom level whereby the target is first visible in the viewport. This occurs when $R(z')/2 = D$ (see Figure 6b). We use this value as a baseline for comparison. Thus, substituting (2) and solving for z' yields:

$$\begin{aligned} \frac{V \cdot 2^{z'}}{2} &= D \\ \log_2 V + z' - 1 &= \log_2 D \\ z' &= \log_2 D - \log_2 V + 1 \end{aligned} \tag{3}$$

4.2.2 Zooming-In

Zooming-in is an interleaved process of zooming-in and corrective scrolling (due to zoom-in drift). When the target appears "closest" to the zoom pivot, it is still unlikely that they are coincident. In the average case, the distance between the target and the zoom pivot is 0.5 px in screen-space.

A user may not position the target as close as possible to the zoom pivot at every zoom-in step. We define σ as the average range in screen-space pixels from the zoom pivot to the target over all zoom-in steps, where $\sigma \geq 0.5$. Then, $Q(\Delta z, \sigma)$ is the zoom-in drift of the target in screen-space pixels, away from the zoom pivot, given a change of Δz zoom levels (see Figure 7).

$$Q(\Delta z, \sigma) = 2^{\Delta z} \cdot \sigma \tag{4}$$

We define ρ as the average distance in screen-space pixels that the target is scrolled over all zoom-in steps. Intuitively, some users may prefer that the target always remain on screen, thus ensuring that $\rho \leq V/2$. Others may be comfortable with a larger ρ, confident in their ability to re-locate the target despite shifting off-screen. If a user consistently stops zooming-in when the target is at the edge of the viewport and scrolls the target exactly to the centre of viewport, then $\sigma = 0.5$ and $\rho = V/2$.

Then, the average change in zoom level, Δz, that maintains the balance between σ and ρ occurs when $Q(\Delta z, \sigma) = \rho$. Thus, substituting (4) and solving for Δz yields:

$$\begin{aligned} 2^{\Delta z} \cdot \sigma &= \rho \\ \Delta z + \log_2(\sigma) &= \log_2(\rho) \\ \Delta z &= \log_2(\rho) - \log_2(\sigma) \end{aligned} \tag{5}$$

Since the task we are modelling requires the user to return to the same zoom level at which they started, we derive the total number of zoom-in iterations, n, from α (actual zoom level) and Δz (delta of zoom-in steps):

$$n = \frac{\alpha}{\Delta z} \tag{6}$$

The parameters σ, ρ, Δz, and n quantify the impact of the speed-accuracy trade-off of the *Has target shifted enough?* (see Figure 3, Z-I Phase). For example, a user may choose to be less precise when scrolling, with a larger σ and smaller ρ, but at the cost of a decreased Δz, and as a result an increase in n.

One corrective scroll is performed after zooming-out, to re-centre the target prior to zooming-in. We handle this operation uniquely to account the scroll distance relative to the actual zoom-out level (α). Since our model is defined relative to the initial distance to the target, we define m as the ratio of target distance, D, to *half* the zoomed viewport size, $R(\alpha)$:

$$m = \frac{D}{R(\alpha)/2} \tag{7}$$

The distance of this first corrective scroll in screen-space pixels is then $m \cdot V/2$. For example, if a user zooms-out precisely to the "naïve optimal" zoom level, z', then $m = 1$, and the distance to scroll is $V/2$.

4.2.3 Costing Functions

We elaborate on the costing functions introduced at the beginning of this section. To account for the time it takes the user to perform each navigation operation, the costing functions are of the form $Cost_{op}(r, d)$, which calculates the average time (ms) to complete r repetitions of distance d for the operation op.

We considered using Fitts' Law for the scrolling costing function, however to do so we would need both the distance and size of target. Unfortunately, it is not possible to definitively calculate the size of the target at each step of the zoom-in process, as perceived by the user.

Thus, we instead use linear costing functions for zooming operations (8), and logarithmic costing functions for panning operations (9). We chose a linear costing function for zooming because mouse-wheel input inherently has physical limitations on the magnitude of ballistic movements before clutching is required.

$$Cost_{op}(r, d) = r \cdot (A_{op} + d \cdot B_{op}) + \text{ceil}(r) \cdot C_{op} \tag{8}$$

$$Cost_{op}(r, d) = r \cdot (A_{op} + \log(d) \cdot B_{op}) + \text{ceil}(r) \cdot C_{op} \tag{9}$$

Values for A, B, and C will be empirically derived. A and B are parameters of a fit for physical operation performance and C represents the time required to acquire/initiate the operation and visual search. There are no accepted models for visual search performance [15]. While we assume a minimized cognitive load through pre-attentive search – that the absence or presence of the target is immediately distinguishable – our costing functions can later be extended to include models of visual search performance.

5 EXPERIMENT – MODEL VALIDATION

We ran an experiment to quantify the costing function parameters (A, B, and C) for all four operations, fit, and validate our model.

5.1 Participants and Apparatus

We recruited 12 paid ($10) volunteer participants (7 female; none left-handed) with mean age 24 (min: 19, max: 30). Mean self-reported computer usage was 24 hours per week. Self-reported frequency of using zooming and scrolling interaction techniques varied: 83% of participants reported using the mouse-wheel to scroll "often" or "always" when available. For input device, 50% of participants reported primarily utilizing wheel mice, 42% touchpads, and 8% touch mice.

Participants performed the study in a private study room using a desktop computer configuration running Windows XP, with a 21-inch LCD monitor displaying a resolution of 1280x1024 pixels, and a Logitech M510 wireless mouse. Participants were seated 20 inches from the monitor.

5.2 Methods and Design

Participants performed a view-pointing task. Trials began with the document zoomed to 100%. The goal was to find the target line, position it within the goal zone, and reset the zoom scale to 100%. The interface was setup with a fixed-sized viewport (800x450px), displaying a document 2 Gpx wide.

Following Guiard and Beaudouin-Lafon's zooming and scrolling study [17], we base our selection of distance to target (D) and size of goal (W) upon the *index of difficulty* (ID) [14], measured in bits. We used 5 evenly-spaced task difficulties, where zooming would be required, starting at 10 up to 26 [17]. The goal (8px) and target (1px) appeared with fixed-sizes in screen-space. Based on a pilot study, we found no significant effects of target direction, and simplify our design by limiting to right-directional navigation.

Three interaction techniques were made available: click-and-drag panning , mouse-wheel scrolling, and mouse-wheel zooming (with CTRL modifier). We included two methods of panning for external validity, and to evaluate whether participants had a preference. Control-display ratios for each technique were: 1:1 for click-and-drag pan, 60px per mouse-wheel detent for scrolling, and 0.5 zoom levels per mouse-wheel detent for zooming.

A repeated measures within-participant design was used, with the independent variable *difficulty* (10, 14, 18, 22, 26 bits). The study was divided into 3 blocks, with 4 randomized repetitions of each condition per block. Prior to the trials, one practice repetition was administered and discarded. Each participant completed the experiment in a single session lasting approximately 30 minutes.

The participants were randomly assigned to two groups, the data from the first is used to fit the costing functions, the *fitting group*, while data from the second is used to validate the model, the *validation group*.

5.3 Data Analysis

We recorded total navigation task completion time and details of each navigation operation: type (scroll; zoom), method (wheel scroll; wheel zoom; click-and-drag), direction (left or right; in or out), and duration, as well as viewport position and zoom level. We extract uses of each interaction technique from these logs, using direction reversal and operation type change (zoom or scroll) to delimit groups of sequential operations. We did not use time as a delimiter for navigation operations. We also calculated the transition time between operations.

Since we are modelling error-free performance, we discard trials where users commit major errors, as defined by our decision process. Scrolling prior to a zoom-out operation changes the ID for the task, making analysis unwieldy. Corrective zoom-outs when zooming-in effectively "restart" the model, and as such would require a recursive model. For simplicity of analysis and modelling, we did not account for these and instead discarded the trials. In total, 19.4% of trials were discarded (140/720). We did not remove any outliers from the data.

While we were interested in both panning using the mouse-wheel and click-and-drag, only 9.7% of interaction samples came from mouse-wheel scrolling interaction. As a result we did not have enough data to calculate cost functions for mouse-wheel scrolling, and these data were not analysed.

5.4 Quantifying Costing Functions

We extracted magnitude and duration for each of the 4 operation types from the fitting group to estimate parameters for our costing functions. Table 2 summarizes the costing function parameters. We used a method similar to that used in Chapuis et al. to calculate fits on noisy, real-world data [8]. We took the performance samples from the first group of participants and divided the samples into 10-quantiles, with an equal number of samples in each. We fitted costing functions to the means of each quantile. Parameters A and B describe the fitted line (R^2 value reported), while C is the average transition time between the start time of an operation and the end time of the preceding operation. As suspected, zooming had a linear fit (Out: $R^2 = .779$, In: $R^2 = .975$). We suspect the lower R^2 value for ZoomOut is due to the smaller number of samples. Participants engaged in fewer zooming-out operations relative to the others. Meanwhile, both scrolls had a strong logarithmic fit (Scroll: $R^2 = .941$, CorrScroll: $R^2 = .939$).

5.5 Empirical Model Input Parameters

Through analysis of the interaction logs, we can quantify the parameters (α, σ, ρ, Δz, and n) of our zooming and panning strategy model for each trial. The averages for the fitting group are outlined in Table 3 and for the validation group in Table 4.

In both groups, the actual zoom-out level, α, and the number of zoom-in iterations, n, increase with ID, which follows intuition, since conditions with larger IDs have farther targets, and require more navigation overall. Interestingly, the average magnitude of zoom-ins, Δz, fluctuates minimally as ID increases and also has low variance, suggesting that participants were consistent in how far they zoom-in at each iteration. This suggests that differences

Operation	Type	R^2	A	B	C
ZoomOut	Linear	0.779	-447.095	157.672	808.285
Scroll	Log	0.941	161.453	124.217	1035.625
ZoomIn	Linear	0.975	36.511	100.532	329.175
CorrScroll	Log	0.939	103.116	92.558	704.918

Table 2. Summarizing costing function parameters. $\boldsymbol{A}$ and $\boldsymbol{B}$ define a fit, while $\boldsymbol{C}$ is the average transition time between operations.

Diff.	Time (s)	α	σ	ρ	Δz	n
ID10	7.2(1.2)	8.2(1.3)	9.7(5.2)	171.9(97.5)	4.3(1.0)	2.5(1.3)
ID14	9.5(1.7)	10.8(0.8)	19.8(11.5)	227.9(64.1)	4.0(0.9)	3.6(1.9)
ID18	12.3(2.7)	14.8(0.7)	13.6(8.7)	258.4(96.6)	4.3(0.9)	4.4(2.3)
ID22	14.3(2.6)	18.1(0.6)	19.3(12.9)	277.9(78.6)	4.4(1.1)	5.7(3.5)
ID26	19.8(4.2)	21.3(0.3)	32.6(10.9)	407.1(145.2)	3.6(0.6)	6.9(2.9)

Table 3. Mean (95% CI) model parameters for the fitting group.

Diff.	Time (s)	α	σ	ρ	Δz	n
ID10	6.4(1.0)	6.6(1.3)	14.1(7.3)	152.8(48.8)	3.8(0.8)	1.9(0.4)
ID14	8.5(0.9)	10.2(0.8)	13.0(7.8)	160.9(62.2)	3.9(0.5)	2.8(0.5)
ID18	11.1(1.8)	13.9(0.7)	12.2(6.1)	209.2(46.2)	4.5(0.8)	3.3(0.6)
ID22	13.8(2.5)	17.3(0.6)	12.0(6.9)	224.6(80.1)	4.5(0.7)	4.1(0.8)
ID26	18.2(2.9)	21.1(0.3)	21.2(7.7)	268.1(75.8)	3.8(0.6)	6.0(1.2)

Table 4. Mean (95% CI) model parameters for the validation group.

between participants may be primarily attributed to σ and ρ. In the fitting group, the variance of σ and ρ, in addition to the larger mean ρ values suggests these participants were less consistent.

Task completion time was on the order of 1 second faster within the validation group. This can be attributed to more consistent performance during zoom-in iterations, as there is less variance amongst the means of parameters σ and ρ, compared to the fitting group. In both groups, the increase of σ and ρ in the ID26 task suggests a shift in navigation strategy: participants sacrificed accuracy (σ) for larger scroll distances (ρ).

5.6 Model Fitting and Validation

We calculate the expected task completion time for the fitting group to test the fit against the observed data. Figure 8 (left) shows the observed navigation task completion time compared to the expected completion time. All expected times based on our model fall within the 95% confidence interval of the mean observed data. On average, the expected completion times are 8.6% faster than the observed completion times.

Next, to validate our model, we used the costing functions derived from the fitting group to calculate the expected task completion time given the strategy model inputs of the validation group (see Figure 8, right). Again, all expected times based on our model fall within the 95% confidence interval of the mean observed data. In this case, the fit between observed and expected completion times is much tighter, with expected times 2.1% faster on average than the observed times.

The discrepancy in fit between the model fitting and validation groups can be attributed to two participants in the former who were inconsistent in navigation patterns.

5.7 Optimal Strategies

Optimal task completion times are calculated for each distance to target (D), by exploring a range of input parameters (see Figure 8). For the fitting group, the estimated optimal completion time is on average 38.1% faster than the observed times (26.5%, 31.7%, 38.4%, 40.3%, 53.5%). For the validation group, the estimated optimal completion time is on average 32.2% faster than the observed times (18.1%, 23.6%, 32.0%, 38.1%, 49.3%). These findings suggest that not only do users fail to strategize optimally, they also perform increasingly worse as the navigation task becomes more difficult. We suspect this is due to compounding suboptimal choices in the zoom-in phase with increasing iterations. With the potential for a 50% improvement in the most difficult condition, clearly, there is significant room for improvement in how users navigation very large data views.

Based on our model, the average optimal values for each ID is the parameterization $\alpha = z' + 1$, $\sigma = 1px$, and $\rho = 490px$, leading to two interesting implications. First, the speed-accuracy trade-off in zoom-in iterations favours precise and large scrolls. In fact, the model suggests that an optimal strategy is to zoom-in past the point where the target is still visible in the viewport. The caveat is that users may have difficulty keeping track of the target when off-screen due to desert fog, which can result in a major error as defined by our decision process. Second, the optimal zoom-out level, α, exceeds the "naïve optimal", z', posited by Furnas and Bederson, and contradicts their "optimal" strategy suggesting undershooting z' is more performant [16]. We suggest two possible explanations for this difference. Since we simulated an unfamiliar document, such an "optimal" strategy is impossible, since target location is not known. It has been suggested that human performance has a very low upper bound based on perceptual bandwidth [17]. This may have resulted in reduced ballistic movements by users, who were constantly engaged in the pre-

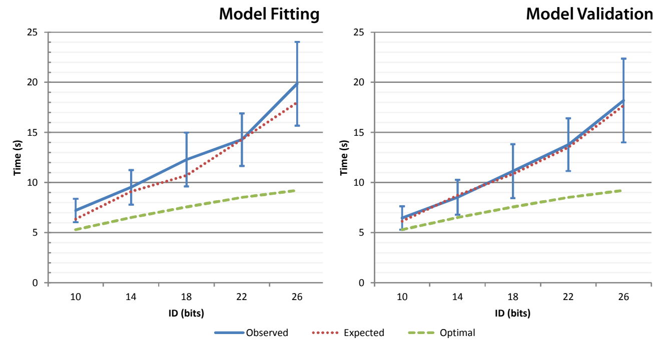

Figure 8. (Left) Model fitting results (95% CI error bars). (Right) Model validation results (95% CI error bars).

attentive search task. An alternative explanation is that the precise metric used to determine the "optimal" strategy is unclear. Since Furnas and Bederson's publication, advances in input devices, such as the mouse-wheel, has made multi-modal interaction easier, in particular in-place zooming activation. In contrast, mouse pointing interaction has changed little. By drastically reducing the cost of engaging in a zooming operation, the favour may have shifted from more scrolling to more zooming, as predicted by our model. Our results do also support some of Furnas and Bederson's intuition: that it is optimal to scroll without zooming for "small" distances. Our model suggests this crossover point, where zooming-out becomes more efficient than scrolling, occurs within a distance of one viewport width.

Thus, slower and more accurate wins. Minimizing σ and maximizing ρ pays dividends, as it increases the zoom-in range (Δz), thus reducing the number of iterations (n). Our model also suggests scrolling is slower than zooming, and an unconstrained zoom-out (α) is critical to optimal performance. A low target saliency threshold, which places a ceiling on α, will devastate navigation performance, forcing users to scroll more than needed.

5.8 User Errors

Error trials, which were discarded from the model fitting and validation, were also analysed. The vast majority of these trials resulted in errors committed during the zoom-in phase of navigation. In particular, zooming-in too far and succumbing to desert fog. This is evidenced by larger ρ values for these trials; on average 295.1px (95% CI: 163.9px) versus 235.9px (95% CI: 79.5px) for non-error trials. This suggests that participants may have attempted to engage in a more efficient strategy, by increasing the scroll distance, but committed more errors in so doing. This evidence supports the observation that the optimal strategy predicted by our model may be difficult for users to fully engage due to limitations in spatial reasoning ability.

6 Limitations

The generalizability of the costing functions presented depends on how well they describe a given user interface. Since we designed our study to mimic a typical desktop user context, these costs may need to be reassessed when applying the model to non-desktop contexts, or non-mouse interaction paradigms, such as touchpads, track-points, or direct touch manipulation. Also, we used Windows XP system drivers with default settings. Indeed, different drivers implement varying transfer functions [26]. The presence of customized drivers or settings in the real-world may impact the generalizability of the costing functions of the model.

In our study, the task required the user to engage in a "pre-attentive" visual search, where the judgement of presence or absence of a target could be made very quickly. In real world documents, distractors may be present, or the search target may not be uniformly salient across zoom levels, both factors which could increase the time required for the visual search task.

Despite these limitations, we have demonstrated the strategy model is robust to the studied context and thus should generalize to any horizontal zooming and panning interface with an arbitrary fixed zoom pivot position and arbitrary goal position.

7 Interaction Design Considerations

Based on the optimal strategy suggested by our strategy model, we elaborate two considerations for the design of interaction techniques for very large data views.

Maximize target saliency threshold. Our model shows the benefit of maximizing the actual zoom-out level (α), beyond the point where the target is first visible in the viewport. If the target is not visible at all zoom levels, then a ceiling is placed on α, which implies guaranteed sub-optimal performance. Techniques like semantic zooming [25] and visualization aggregation [13] can help to promote salience of targets at all levels of zoom. However, the task domain must be carefully considered when aggregating data, such that condensed representations do not lack the detail required to complete a task. Lack of detail may be continuous, such as details simply blending together as the user zooms, or discrete, such as when zooming-out of a map application results in street labels being removed.

Enhance motor and spatial skills. Optimal navigation benefits from being able to correctively scroll quickly and accurately (minimize σ, maximize ρ). Techniques such as snapping, semantic pointing [5], and control-display gain schemes [9] can augment a user's natural abilities in minimizing σ. Moreover, techniques such as overview+detail and focus+context (see [12] for a review) or contextual cues about off-screen target location [19] provide a means for users to maximize ρ, while mitigating potential major errors in navigation due to loss of target. Based on insights from error trials, users seem to want to leverage a more optimal strategy, but are limited by their spatial abilities. Thus, it is important to ensure that both precise and ballistic navigation operations are easily controlled.

8 Future Work and Conclusion

We have presented a decision process and strategy model that provides empirical evidence in favour of long and precise panning for optimal zooming and panning navigation. However, results also indicate that users may be aware, but unable to engage such an optimal strategy due to limitations of their own spatial reasoning ability. We present two design considerations to aid users in achieving optimal strategy and review research which aims towards these goals. However, many do not yet scale to the complexities and magnitude of larger datasets. This is a clear area for future research.

Together, the decision process and strategy model provide a generalized approach to understanding of zooming and scrolling across different contexts. The generality of the particular costing functions we have supplied is an open question; it is up to the user of our model to determine the costing functions for their context. Evaluating our strategy model with cursor-based zoom pivots and more advanced simultaneous zooming and panning techniques is an open area of research we intend to pursue.

In our work, we have shown the benefit of a systematic analysis of the components of a complex navigation strategy. It is our hope that the model described in this paper will promote further discussion and comparison of the impact of user decisions on navigation strategy. A navigation strategy, after all, is more than just the sum of its parts.

Acknowledgements

We would like to thank John Hancock, Azam Khan, Jason Alexander, Kim Wright, and the anonymous reviewers for their thoughtful feedback.

References

[1] Alexander, J. and Cockburn, A. (2008). An empirical characterisation of electronic document navigation. *GI'08,* 123-130.

[2] Andersen, T.H. (2005). A simple movement time model for scrolling. *CHI EA'05*, 1180-1183.

[3] Appert, C. and Fekete, J.-D. (2006). OrthoZoom scroller: 1D multiscale navigation. *CHI'06*, 21-30.

[4] Bederson, B.B. and Hollan, J.D. (1994). Pad++: a zooming graphical interface for exploring alternate interface physics. *UIST'94*, 17-26.

[5] Blanch, R., Guiard, Y. (2004). Semantic pointing: improving target acquisition with control-display ratio adaptation. *CHI'04*, 519-526.

[6] Cao, X., Li, J.J., and Balakrishnan, R. (2008). Peephole pointing: modeling acquisition of dynamically revealed targets. *CHI'08*, 1699-1708.

[7] Card, S. K., Moran, T. P., and Newell, A. (1983). *The psychology of human-computer interaction.* Hillsdale, NJ: Lawrence Erlbaum Ass.

[8] Chapuis, O., Blanch, R., and Beaudouin-Lafon, M. (2007). Fitts' Law in the Wild: A Field Study of Aimed Movements. *Tech. Rep., LRI, Univ. Paris-Sud.*

[9] Cockburn, A., Quinn, P., Gutwin, C., and Fitchett, S. (2012). Improving scrolling devices with document length dependent gain. *CHI'12*, 267-276.

[10] Cockburn, A., Savage, J. (2003). Comparing speed-dependent automatic zooming with traditional scroll, pan and zoom methods. *BCSHCI'03*, 87-102.

[11] Cockburn, A., Savage, J., and Wallace, A. (2005). Tuning and testing scrolling interfaces that automatically zoom. *CHI'05*, 71-80.

[12] Cockburn, A., Karlson, A., Bederson, B.B. (2009). A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv.*, 41(1), Article 2, 31 pages.

[13] Elmqvist, N., Fekete, J.-D. (2010). Hierarchical Aggregation for Information Visualization: Overview, Techniques, and Design Guidelines. *IEEE TVCG*, 16(3), 439-454.

[14] Fitts, P.M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *J. Exp. Psy.* 47(6), 381-391.

[15] Forlines, C., Shen, C., Wigdor, D., and Balakrishnan, R. (2006). Exploring the effects of group size and display configuration on visual search. *CSCW'06*, 11-20.

[16] Furnas, G.W. and Bederson, B.B. (1995). Space-scale diagrams: understanding multiscale interfaces. *CHI'95*, 234-241.

[17] Guiard, Y. and Beaudouin-Lafon, M. (2004) Target acquisition in multiscale electronic worlds. *IJHCS*, 61(6), 875-905.

[18] Guiard, Y., Beaudouin-Lafon, M., Bastin, J., Pasveer, D., and Zhai, S. (2004). View size and pointing difficulty in multi-scale navigation. *AVI'04*, 117-124.

[19] Gustafson, S., Baudisch, P., Gutwin, C., and Irani, P. (2008). Wedge: clutter-free visualization of off-screen locations. *CHI'08*, 787-796.

[20] Hinckley, K., Cutrell, E., Bathiche, S.,and Muss, T.(2002). Quantitative analysis of scrolling techniques. *CHI'02*, 65-72.

[21] Hornbæk, K., Bederson, B.B., and Plaisant, C. (2002). Navigation patterns and usability of zoomable user interfaces with and without an overview. *TOCHI*, 9(4), 362-389.

[22] Igarashi, T. and Hinckley, K. (2000). Speed-dependent automatic zooming for browsing large documents. *UIST'00*, 139-148.

[23] Jakobsen, M.R. and Hornbæk, K. (2011). Sizing up visualizations: effects of display size in focus+context, overview+detail, and zooming interfaces. *CHI'11*, 1451-1460.

[24] Jul, S. and Furnas, G.W. (1998). Critical zones in desert fog: aids to multiscale navigation. *UIST'98*, 97-106.

[25] Perlin, K. and Fox, D. (1993). Pad: an alternative approach to the computer interface. *SIGGRAPH'93,* 57-64.

[26] Quinn, P., Cockburn, A., Casiez, G., Roussel, N., and Gutwin, C. (2012). Exposing and understanding scrolling transfer functions *UIST'12*, 341-350.

[27] Savage, J. and Cockburn, A. (2005). Comparing Automatic and Manual Zooming Methods for Acquiring Off-screen Targets. *BCSHCI'05*, 3, 439-454.

[28] Sutherland, I. E. (1963). Sketchpad, A Man-Machine Graphical Communication System. Ph.D. Thesis, MIT.

[29] van Wijk, J.J. and Nuij, W.A.A. (2004). A model for smooth viewing and navigation of large 2D information spaces. *IEEE TVCG*, 10(4), 447-458.

[30] Zhai, S., Smith, B.A., and Selker, T. (1997). Improving Browsing Performance: A study of four input devices for scrolling and pointing tasks. *INTERACT'97*, 286-293