Two Novel Off-Screen Navigation Techniques

by

Mahtab Nezhadasl

A Thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
in partial fulfilment of the requirements of the degree of

Master of Science

Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada

Abstract

In large workspaces that do not fit on the screen space, users have to navigate to various regions outside the viewport to locate items of interest. Researchers have developed a variety of different navigation techniques to improve the performance of working with large workspaces. In this thesis I design, implement, and evaluate two novel navigation techniques to access off-screen content. I call these techniques Multiscale Window and Crystal Ball. The design of these two techniques was based on two hybrid interaction systems WinHop and Multiscale Zoom. Multiscale Window takes advantage of Multiscale Zoom to provide an overview of the context by incorporating full-detail object representations (proxies), and Crystal Ball is an improvement to WinHop. The implemented techniques were designed to alleviate the shortcomings of both hybrid techniques; Multiscale Zoom lacks the ability to provide detail information of overlapped proxies, and WinHop does not facilitate navigation to the off-screen region due to the animation.

I evaluated the Multiscale Window and Crystal Ball techniques in two experiments. In the first experiment (N = 14) a Tablet PC with a digital pen as an input device was used. Results showed that there was no significant difference between Multiscale Window and Multiscale Zoom. However, Crystal Ball showed improved effects over WinHop in most tasks. The second experiment (N = 14) compared the same techniques as in experiment one, on a PC with a mouse as input device. The results indicated that subjects were faster with Crystal Ball than WinHop. Like the first experiment, Multiscale Window did not show any significant improvement over Multiscale Zoom.

Acknowledgements

I would like to take this opportunity to gratefully acknowledge all the people who have contributed to the different aspects of this study. This work would not have commenced and been completed without their support, involvement, cooperation, and collaboration.

First and foremost, I would like to express my deep and sincere gratitude to Dr. Pourang P. Irani whom under his supervision I chose this topic and began the thesis. He was always available to discuss and share his ideas with me. I appreciate his support, advice, and guidance from the very early stages of this research as well as leading me to publications and teaching me methods of research. He spent his time reading the drafts and offering valuable comments and suggestions to improve both the substance and my English writing in published papers. I am a very lucky student to have such a nice advisor like Dr. Irani.

My thanks also go to the faculty members and staff of the Department of Computer Science at the University of Manitoba for their continuous assistance, help, and guidance throughout my Master's degree program.

Special gratitude is extended to my colleagues, the HCI (Human-Computer Interaction) Lab members, for being there whenever I needed their help and support, in particular Grant Partridge for providing me his code of the WinHop and Multiscale Zoom techniques, Chris Carmichael for proofreading my thesis, Erum Tanvir, Sean Gustafson, Dave McCallum, Kang Shi, and Ed Mak.

It is a pleasure for me to pay tribute also to all of the volunteers who participated in my experiments as well as all my friends for their help and for the days we spent together.

Dedication

This thesis would be incomplete without expressing my deepest appreciation to the most important people in my life, my dearest family to whom I dedicate this work: my father (Jalil), my mother (Mahin), and my loving sister (Mahsa). They have given me encouragement and financial support during my studies in Canada and consistently giving me all their love throughout my life.

Contents

List of Tables

List of Figures

List of Copyrighted Material

The following figures have been reproduced from other sources with permission:

Figure 1.1 on page 2 from [19] © 2006 ACM, Inc.

Figure 2.1 on page 9 from [16] © 2000 ACM, Inc.

Figure 2.2 on page 11 from [4] © 1994 ACM, Inc.

Figure 2.3 on page 12 from [32] © 2001 Pearson Education, Inc.

Figure 2.4 on page 14 from [32] © 2001 Pearson Education, Inc.

Figure 2.5 on page 17 from [26] © 2004 ACM, Inc.

Figure 2.6 on page 19 from [3] © 2003 ACM, Inc.

Figure 2.7 on page 20 from [34] © 2003 ACM, Inc.

Figure 2.8 on page 24 from [6] © 2005 ACM, Inc.

Figure 2.9 on page 25 from [22] © 2004 ACM, Inc.

Figure 2.10 on page 26 from [33] © 2004 ACM, Inc.

Figure 2.11 on page 27 from [19] © 2006 ACM, Inc.

Figure 2.13 on page 31 from [18, 25] © 2007 authors.

Chapter 1

Introduction

The development of new devices and the widespread use of interactive tools have created an environment in which users are interacting daily with large electronic documents. Navigation is one common interaction technique that allows users to 'move' around and inspect different areas of large documents. In tasks that require editing or word processing, users need to inspect and work on content that can be spread throughout various regions of a document. With maps and graphical workspaces, users need to locate various types of landmarks or regions of interest. As a result, there is a constant flow of movement between different regions in a document.

As documents become larger, it is common for most users to interact with

Figure 1.1: Example of a large document space that does not fit entirely in the display region. Large pockets of data are outside the user's viewport. Navigating to the different locations is important for a number of tasks. In this example the R's represent restaurants and M's represent metros on the map. Image from [19] © 2006 ACM, Inc. Reprinted by permission.

visual documents that do not entirely fit on their screens. Particular documents, with textual content or maps, can be much larger than the limited display space. The problem is even more severe for users of hand-held devices, like PDAs, because small-screen devices have a limited display space (Figure 1.1). Similarly on large displays, such as wall-sized displays, viewing and interacting with documents is not trivial. To locate items of interest users have to navigate to various regions outside the viewport. This can lead to disorientation, significant overhead in interaction, and reduction of navigation efficiency. Therefore, using an appropriate navigation technique, which enables the user to navigate efficiently between different areas of a visual document, is essential for proper interaction. A variety of techniques have been designed to resolve the problem of off-screen navigation. However these techniques still require a large number of 'trips' outside the user's viewport and can result in inefficient navigation.

1.1 Goal of the Thesis

The main goal of the thesis is to improve the efficiency of the two existing interaction techniques - WinHop and Multiscale Zoom - which are a hybrid from a number of previous developed techniques.

In this thesis, I develop three novel interaction techniques, which I call Multiscale Window, Spring, and Crystal Ball, based on Multiscale Zoom and WinHop. Multiscale Window and Spring were designed to overcome the problem of Multiscale Zoom to provide full-detail representation of overlapping proxies. Moreover, Crystal Ball improves the efficiency of WinHop by facilitating the task of navigating off-screen target.

1.2 Methodology

I used a three stage process - design, implementation, and evaluation - for accomplishing the work outlined in this thesis. In the first phase, I produced several different designs to improve the WinHop and Multiscale Zoom techniques.

These designs were paper prototypes and sketches outlining the various components. After selecting the best competing design, I implemented the selected techniques in the second phase. The implementation was done in C#.NET on the Windows platform. In the final stage I evaluated the implemented techniques by comparing them to WinHop and Multiscale Zoom.

1.3 Organization of the Thesis

Overall, my thesis is organized as follows.

Chapter 2 introduces a review of the literature related to the research. It summarizes the previous work into four categories: 2D navigation techniques, off-screen object visualization, proxies and portals, and finally hybrid navigation techniques.

Chapter 3 provides the detailed design of the three developed techniques: Multiscale Window, Spring, and Crystal Ball. It also discusses how these techniques overcome the drawbacks of the WinHop and Multiscale Zoom techniques.

Chapter 4 describes the user evaluation of the designed techniques. Two experiments that were designed to compare the new techniques to the hybrid approaches were described. It also discusses the employed methodology and obtained results from the above-mentioned experiments.

Chapter 5 summarizes the contribution of this thesis, and also outlines future directions for this work.

Chapter 2

Related Work

A large body of research has been developed and devoted to the study of interactive navigation methods. The work being proposed has been largely inspired by a number of existing techniques. I first provide a review of the literature concerning 2D navigation techniques, the use of off-screen object visualization to depict awareness of objects outside the viewable region, a glance at proxy and portal techniques to access distant objects, and finally a review of two hybrid techniques upon which a large part of this research is based on.

2.1 2D Navigation Techniques

Several types of 2D navigation techniques have been developed including scrolling, panning, zooming, overview+detail, focus+context, and chunking techniques.

2.1.1 Scrolling, Panning, Zooming

Scrolling, panning, and zooming are some of the most fundamental navigation techniques that are used to view off-screen content in common mainstream applications. Scrolling is a primary and very common strategy to access data inside and outside of the visible area. Scrollbars and scroll rings [24, 31] are software widgets used to enable users to move horizontally and/or vertically over a visual workspace. Scrollbars are vertical or horizontal bars that appear on the side or bottom of a window, and only support scrolling in one dimension. They contain arrows at either end, a body, and an elevator. While scrollbars can transfer the user to any point in the document, they lose precision in very long documents. Furthermore, they interrupt the user's workflow [24]. With scroll bars users have to locate the arrows, the body or elevator. This in itself requires a large movement away and back to the working content area.

Several other implementations of scrolling have been developed to facilitate the user's navigation tasks. Scroll ring is a software simulation of a hardware scroll ring, which is a device mapping circular motion of the user's finger into vertical scrolling motion [24]. Scroll ring is useful in portable devices, and it performs at least as well as a mouse wheel for medium and long distances [24]. Hardware dependent alternatives to scrollbars and scroll rings are keyboard scroll arrows and mouse with a wheel or isometric joystick [13, 35].

A substantial amount of research has been done to improve the efficiency of the scrolling experience including rate-based scrolling [35] which maps the displacement of an input device to the velocity of the scrolling experience, and speed-dependent automatic zooming (SDAZ) [16] that decreases the motion-blur effect at high scroll speeds - the limitation of rate-based scrolling. SDAZ automatically changes the zoom level dependant on the scroll rate (Figure 2.1). One ecologically oriented evaluation [9] reveals that users are significantly faster with SDAZ in both text document and map browsing tasks, compared to traditional scrolling techniques. In addition, workload assessments as well as subjects' preferences confirm the efficiency of the SDAZ [9]. However, it is not appropriate for abstract or symbolic information (such as a dictionary) of small or large size, relative pointing input devices, and for non-expert users; moreover, its dynamic interaction can confuse the users who use the system for the first time [16].

Despite the simplicity of scrolling mechanisms, navigating with scrolling is lin-

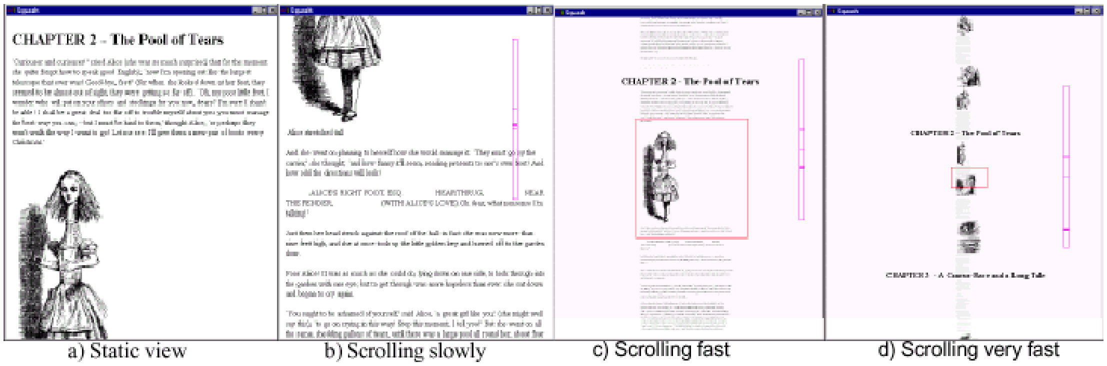


Figure 2.1: SDZA is a good alternative to scrolling. As it automatically shows various aspect ratios of the workspace based on the scrolling rate. Image from [16] © 2000 ACM, Inc. Reprinted by permission.

ear; i.e. to go to the last page of the document, all the pages in between have to be inspected and also assume the situation in which you have a document so large in both width and height, and you are in the left top corner of the document so that you can see only a portion of it on your screen. To reach to a point in the right bottom corner, you have to do at least two movements: right and then bottom or bottom and then right. If you could navigate in a nonlinear manner, you would have accessed your desired point with only one movement and save the time. Thus the scrolling technique is time consuming. Moreover, scrolling requires the user to create and maintain an internal spatial map of the various areas of the workspace. A slight improvement to scrolling is panning.

Panning is another technique to view content outside the visible area by providing smooth movement of a fixed viewport over the workspace or a movement of the workspace under the viewport. To accomplish this goal, panning provides a "hand" drag tool, a widget that supports two-dimensional smooth scrolling [31]. Some studies (e.g. [20, 21]) have compared different panning methods over a variety of tasks. One result [20] revealed that on touch-based displays, panning the workspace - pushing the background - was better than either dragging the viewport or touching the window border. The problem with this form of interaction is that the user has to perform multiple discreet movements to pan to distances of the workspace that are further away from the viewport, as well as its similarity to scrolling in presenting off-screen content linearly.

Zooming is another navigation technique that enables the user to access multiple perspectives of the data by changing its scale. Zooming provides a non-linear navigation approach which can make this method faster to access off-screen objects, compared to scrolling and panning. For efficient navigation, most systems use zooming accompanied by scrolling or panning. For instance, Pad++ [4, 5] is a multi-scale graphical environment where dynamic objects can be placed at any position and at any scale. It also supports panning and zooming. It lays out the most highly rated information larger and makes it more visible than related but lower rated information [4]. Pad++ uses semantic zooming [14, 32] to represent abstraction of objects at one level and details at another magnification level [4] (Figure 2.2). Semantic Zooming is a user interface which unlike geometric zooming

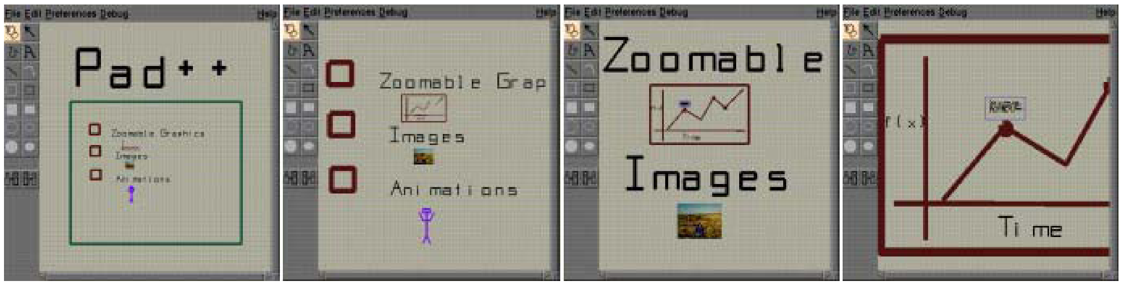


Figure 2.2: Pad++, an example of one of the earliest zooming prototypes. (From left to right the view has been zoomed in to access the desired picture). Image from [4] © 1994 ACM, Inc. Reprinted by permission.

(where the scale determines the apparent size of objects linearly), enables objects of the content to have a more complex relation between their appearance and their scale. Figure 2.3 shows how this technique works.

Overviews, resulting from zooming out, provide users with off-screen content awareness, and perform better than regular scrolling systems [21]. Zooming provides the chance of seeing data from any distance; however, it allows only a single type of view at any time and requires the user to create internal connections between different scaled views.

Numerous other techniques have been designed to overcome the limitations of scrolling, panning, and zooming. However, these navigation techniques are only slowly making into mainstream applications.

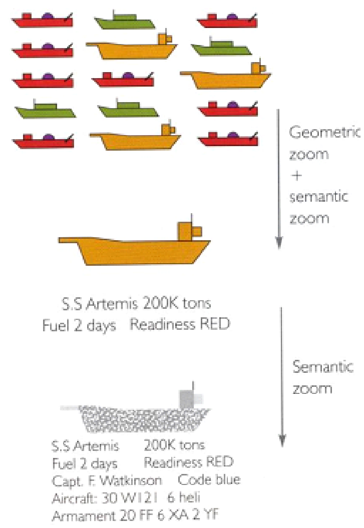


Figure 2.3: In Semantic Zooming, at a high magnification level more abstraction is apparent; As the user 'zooms' in more details or semantic information is revealed. Image from [32] © 2001 Pearson Education, Inc. Reprinted by permission.

2.1.2 Overview Plus Detail

Sometimes it is necessary to have different views of the same content simultaneously, and this problem is addressed by the Overview Plus Detail approach [14]. This method provides an overview and a detail view of the data; generally the overview presents the entire workspace in small size, and the detail view shows a close-up of a portion of the data at normal size. The overview provides a visual facility such as a box to identify the current position of the detail view within itself. This facility decreases re-orientating time spent for switching between detail and overview. For some tasks overview+detail technique surpasses both panning and fisheye representations; moreover, users claim that it is simple to understand [15].

However, integrating the different scales of visual information into a single consistent spatial mental model is difficult. Consequently, re-orientating time when switching between windows increases the interaction overhead.

2.1.3 Focus Plus Context

Focus Plus Context techniques result from altering the multiple views techniques so that both overview and detail view can be in the same window. In other words, focus+context uses only a single view to inspect the content so that there is no need for split windows or split attention. By magnifying the most relevant information and reducing less important content, the entire workspace can fit into the viewport. Focus+context methods include the Bifocal Display [17, 32], the Perspective Wall [17, 32], and Fisheye Views [27]. The bifocal display, an early computer-based distortion-oriented display technique, divides the display screen into three viewports; the space of interest is displayed in the larger central viewport, while those parts of the data lying outside this area are compressed horizontally and/or vertically so that they can be entirely displayed within the two outer viewports (Figure 2.4). Although bifocal displays are relatively easy to implement and provide spatial persistence between regions, there is no continuity of magnification at the boundary between the detailed view and the distorted view [23].

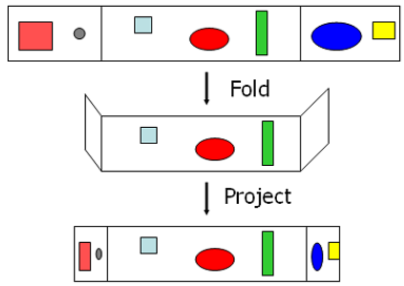


Figure 2.4: The principle of the bifocal display. Image from [32] © 2001 Pearson Education, Inc. Reprinted by permission.

Perspective wall is a three-dimensional implementation of the bifocal display. It smoothly integrates detailed and contextual views to assist in the visualization of linear information [23]. The main difference between bifocal display and perspective wall is in the out-of-focus regions, where the perspective wall de-magnifies at an increasing rate, while the bifocal display does it at constant rate [23]. The reason is due to the angle; the greater the angle, the flatter the slope. Like the bifocal display, perspective wall includes discontinuity in the magnification function at the points where the two side panels meet the middle panel; the bigger the angle, the greater the discontinuity [23].

Fisheye views magnify the vertices of greater interest and correspondingly reduce the vertices of lower interest in a way that the entire view still occupies the same amount of screen space [27]. Using non-linear magnification, fisheye views create balance between magnification and compression of the information [11]. Some studies [12, 2, 15] show that fisheye interfaces are more appropriate than overview+detail, panning, and zoomable interfaces for time-critical tasks when obtaining a more complete understanding of the document is less important. However, using fisheye views, focus-targeting (the task of moving the focus to a new target) is difficult; since objects seem in movement as the fisheye lens reaches them [11]. Another problem is the difficulty of remembering items and locations in the workspace [29].

Speed-coupled flattening (SCF) [11] and Visit Wear [29] are two proposed techniques to improve usability of fisheye views. The former technique based on the velocity and acceleration of the pointer dynamically decreases the motion effects of magnification [11]; thus addresses the focus-targeting issue. The latter method is for solving the memory problem by visually representing the previously visited places [29]. Both above-mentioned techniques have tradeoffs.

Focus+context approaches introduce distortion which avoids precise judgments in some interactions such as targeting.

2.1.4 Chunking Methods

Chunking methods break large workspaces into manageable and viewable portions [19]. For example, Flip Zooming [8, 17], a distortion-oriented technique, segments the entire workspace into rectangular units called tiles. At any given time only one tile is in focus, and it is presented in the center of the display area at its normal size. The other tiles which are out of focus are placed around the focused tile in small size and as a different display e.g. as a thumbnail sketch. This arrangement allows maximum available screen space to segments (pages) in focus. In the flip zooming method, navigation is accomplished by flipping through the tiles in sequence, and every segment comes into focus either by flipping to it or clicking on its non-focus representation.

The difference between most distortion techniques and flip zooming is that flip zooming allows seeing both focus and context objects from straight ahead and without large geometric distortions. This property increases recognition of any given object, since the only variables are the size and location of the object. Moreover, flip zooming's linear scaling does not hide remote objects; rather, all objects are visible simultaneously [8].

ZoneZoom [26, 19] is another chunking technique that divides up the given view of the information space into nested sub-segments. It combines panning and zooming interactions into one cognitive chunk [26] (Figure 2.5). In spite of providing the ability to traverse large information spaces, ZoneZoom, like other zoomable user interfaces, does not keep track of the user's location in the workspace.

Chunking methods like overview+detail techniques require extra effort to in-

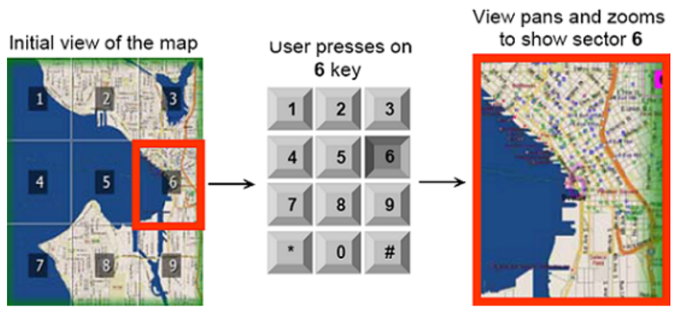


Figure 2.5: ZoneZoom chunks the space by mapping regions onto the input device's segments. Image from [26] © 2004 ACM, Inc. Reprinted by permission.

terpret the overview and recreate the relationships between the details and the whole [19]. Table 2.1 indicates the principles, advantages, and disadvantages of the above-mentioned 2D navigation techniques.

Several techniques have been specifically designed to improve interaction with content located outside the user's viewport. In particular, several visualization techniques have been implemented. These techniques lack the level of interactivity that facilitates navigation, but nevertheless provide some interesting insight and principles upon which the work in my thesis will be founded.

	Scrolling	Panning	Zooming	Overview+Detail	Focus+Context	Chunking Methods
Principle	Linear movement in X or Y directions separately around workspace.	Smooth linear movement in X and Y directions simultaenously around workspace.	Multiple discreet views.	Simultaneous multiple views at different scales.	One view combining focused details while maintaining context.	Manageable chunks of entire workspace.
Advantage	Easy Simple	Easy Simple	Fast because of its nonlinear navigation Multiple levels of detail	Simple to understand	No splitting attention	Highly visible areas of focus Excellent context for the segments of interest Good for navigating on small screen sizes
Drawback	Only a viewable subset at a time Need to create a cognitive map Tedious for large workspaces	Multiple discreet movements for large documents	Single type of view at a time Need to connect internally between different scaled views	Split attention Wasted space Coordinate multiple windows High re-orientating time	Distortion	Extra effort to interpret overviews Recreate relationships between details and the whole

Table 2.1: Principles, advantages, and disadvantages of the common 2D navigation techniques.

2.2 Off-screen Object Visualizations

Halo [3, 30] is a visualization technique, used for small mobile devices, that enhances spatial cognition by identifying the location and distance of off-screen objects. This approach uses rings to encircle off-screen objects so that every off-screen object is surrounded in the center of a ring. In addition, these rings should be big enough so that it could be seen from the border region of the current window in the shape of arcs (Figure 2.6). Such a design allows users to infer the distance and direction of off-screen objects. Halo combines most of the advantages offered by other

Figure 2.6: Halos to show items of interest that reside outside the viewport. Image from [3] © 2003 ACM, Inc. Reprinted by permission.

approaches. For example, it offers a single non-distorted view to show detailed information without losing context. Moreover, unlike arrow-based visualizations, Halo does not require additional distance annotation; since using arcs not only indicate the direction of off-screen objects, but also visually shows their distance.

Thus, the user's performance significantly increases with Halo [3]. The drawback of Halo is its higher error rate in contrast with arrow-based interfaces [3]. This discrepancy may be because of the shape of arcs; changing arcs to ovals may be useful, as suggested by Baudisch and Rosenholtz [3].

City Lights [34, 19] use similar principles as Halo. They use rectangular shapes, integrated along the borders of the current view, to represent the direction of off-screen objects. For showing the distance of off-screen objects, city lights take advantage of colour as a visual cue; grey for indicating far objects and color for showing near ones. Figure 2.7 presents the graphical user interface of the City Lights technique. City Lights support direct navigation by moving the view to the

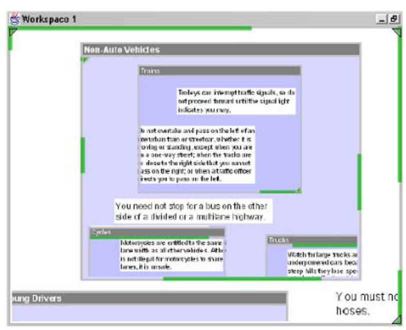


Figure 2.7: City lights provide awareness of the existing off-screen objects by showing them in form of rectangular bars along the borders of the current view. Image from [34] © 2003 ACM, Inc. Reprinted by permission.

nearest object indicated by the clicked rectangular block. Using City Lights, except navigation, users can also obtain the physical and informational properties of off-screen objects (such as their size, color, shape, type, state etc) as well as abstract

	Halo	**City Lights**
Principle	Showing location and distance of objects.	Providing information of objects outside viewport.
Advantage	Non distorted view Enhanced spatial awareness	No significant clutter Enhanced visualization by including a mapping of off-screen object attributes
Drawback	Inaccurate estimation of distances	Time-consuming for some navigation to off-screen locations

Table 2.2: Principles, advantages, and disadvantages of Halos and City Lights.

information (such as history, degree of interest and so on) [3].

Although Halo and City Lights are effective techniques to provide off-screen object awareness, on their own they do not assist users in navigating to the object for inspection or manipulation [19]. Table 2.2 summarizes the two above mentioned off-screen object visualization techniques, Halo and City Lights.

2.3 Proxies and Portals

The development of large displays including interactive walls and multiple desktop displays [28] has led to the development of a new series of interaction techniques. It is because previous interaction methods seem impractical for these kinds of displays like walking towards remote objects, mismatch between input types, as well as the existence of bezels (physical obstructions such as display rims). Contrary to the interaction on desktops and small screen devices that users can reach most displayed items, data on large displays often resides in an unreachable location. This type of visualization causes the user not to be able to view all parts of the screen [7]. Consequently, the user has to walk back and forth to the local and remote spaces. Mismatch between input types occurs when using touch/pen-operated screens, since many of the existing techniques are based on indirect input devices such as mice, track pads or joysticks, while touch/pen input was designed for the immediate correspondence between input space and display space. As a result, users have to adapt their input behavior to the display system physicality [1]. Regarding to the existence of bezels, one vivid example may be dragging an icon across a bezel; the user has to drop the icon half way across the bezel and then pick it up at the other side [1].

Since these new methods are designed for interacting with large screens, they bring distant objects closer to the user's interaction space by using proxies - temporary duplicates of the object that allow actions on the original - or portals that provide the user with a window into a remote area of the workspace [19].

Drag-and-Pop and Drag-and-Pick [1], proxy-based interaction techniques, allow the user to access remote objects by creating temporary local copies of them close to the user's cursor. Drag-and-Pop is based on traditional drag-and-drop; however, once starting to drag an item towards a target item, drag-and-pop moves potential target icons, located in the direction of the drag motion, to the current position of the cursor. Drag-and-pick is an extension of drag-and-pop such that it allows users to activate items, for example opening folders and the like. To accomplish this operation, drag-and-pick starts when the user drags on an empty screen space. In this case, instead of showing only compatible types of items, all items located in the direction of the drag motion will crop up. Afterwards, dragging and then releasing the mouse cursor over one of the targets activates the folder, file or application associated with that icon - like double click. Drag-and-pop/pick is ideal for single target operations, and addresses the problem of walking toward remote objects as well as crossing the display bezels [1]. However, there are some shortcomings with this technique including target identification (since the cursor's vector of initial movement determines the sector of influence), number of proxies is limited, (otherwise the cursor's surroundings would be quickly filled up with them), multiple operations (such as drag and pop is accomplished by a single pen-down operation, it prevents multiple operations that require various pen-up/down events for a single task) and layout of proxies (it does not completely resemble the original target layout) [6].

The Vacuum [6] was proposed to address some of the limitations of the drag-and-pop. It includes a circular widget with a user controllable arc of influence at its point of invocation. This arc spreads out to the borders of the display, and allows selecting and manipulating remote objects, inside of it, in the form of proxies brought closer to the widget's center (Figure 2.8). The main drawback of the Vacuum is the scale of the proxies, compared to drag-and-pop technique [6]. The reason is that to maintain the relative distances between the original objects, the Vacuum must shrink the proxies [19]. Bezerianos and Balakrishnan [6] suggest that high resolution of the display can alleviate some of the scale issues. Arc of influence also produces another problem; when the Vacuum reaches to the edge of the workspace, arc of influence vacuums in large numbers of off-screen objects that may not be of interest [19].

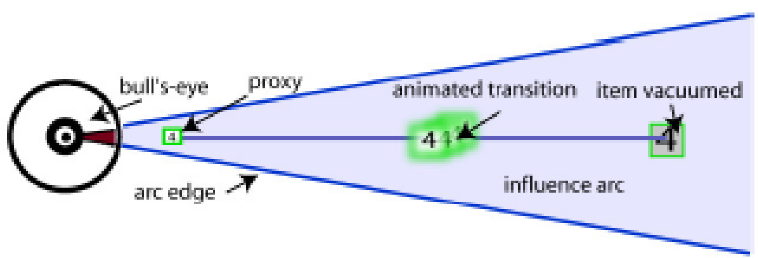
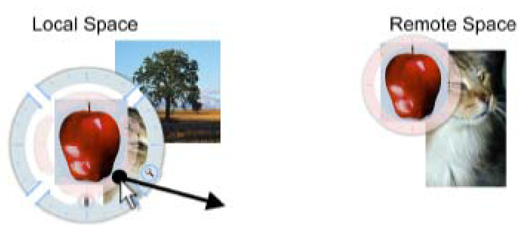


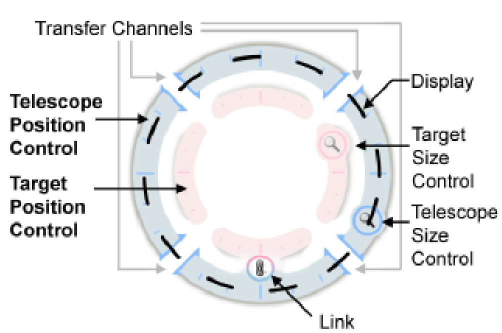
Figure 2.8: Vacuum is a circular widget including an inner bull's-eye center plus an arc of influence with a user controllable angle extending from the center along a user defined line to the display borders. Image from [6] © 2005 ACM, Inc. Reprinted by permission.

Both Drag-and-Pop/Pick and Vacuum do not provide off-screen object awareness. Moreover, they do not show the surroundings of the original object [19].

An alternative to interact with remote areas of a large display instead of walking back and forth between objects of local and remote spaces are portals. Portals behave like windows that facilitate viewing remote content of the workspace. The Frisbee technique [22] consists of a telescope located in the local space and a target in the remote area. Through the telescope, users can view and manipulate the remote content within the target (Figure 2.9 a). Khan et al. [22] showed that experienced users preferred the Frisbee at a distance of 4.5 feet rather than walking between local and remote spaces. One of the drawbacks of the Frisbee is that it has several controls on the telescope (Figure 2.9 b), which increases the amount of overhead required to interact with distant objects.



(a)



(b)

Figure 2.9: The function of Frisbee to access remote content (a), and the controls used on the telescope (b). Image from [22] © 2004 ACM, Inc. Reprinted by permission.

Another portal based technique is WinCuts [33]. WinCuts enables users to remotely interact with regions of interest contained within various windows. Users can specify regions of interest in the form of windows (WinCuts) from existing windows, and interact with content in them, just as they would in the original window (Figure 2.10). WinCuts are useful for content layout, peripheral awareness tasks, and rapid interface prototyping [33]. However, because of the simple creation of WinCuts, users usually end up with many more WinCuts than existed windows. So the management of groups of windows is problematic. Another shortcoming of WinCuts is that scrolling the source window may affect multiple WinCuts because they are defined only by the geometric region of the window, not the semantic content [33].

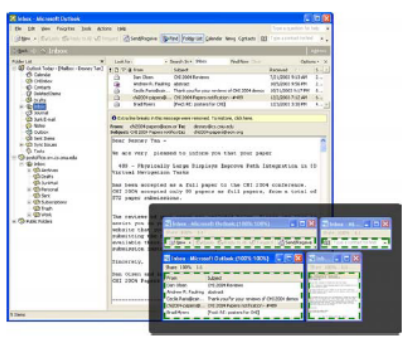


Figure 2.10: An example of WinCuts. The behind window (which is in black) is one of the interested windows; by wincutting sections of interest from any opened windows users can use less screen space and effort. Image from [33] © 2004 ACM, Inc. Reprinted by permission.

Hop (halo+proxy) [19] has been proposed to address the shortcomings of the previous interaction techniques. It combines Halo with a teleportation mechanism that shows proxies of distant objects. Hop takes the advantage of Halo to provide awareness of off-screen targets, and it uses a proxy-based technique to be transported fast to the context of the target. To accomplish this purpose, Hop uses a moveable 'laser beam' (like that used in the Vacuum) which interacts only with halos on the screen's border (Figure 2.11) [19]. With Hop, users are significantly

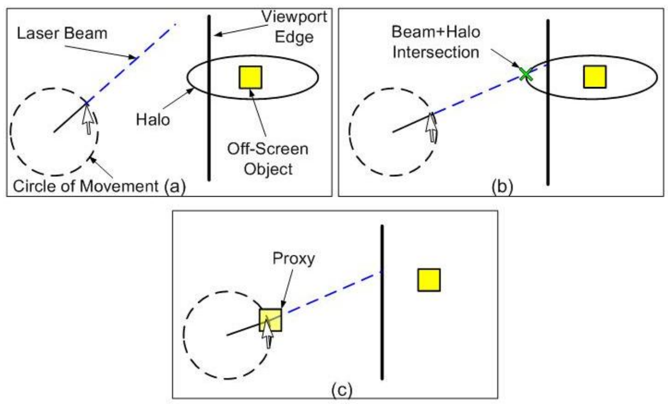


Figure 2.11: Major components in a Hop; (a) creation of a laser beam, (b) intersection of a halo and the beam, and (c) creation of a proxy. Image from [19] © 2006 ACM, Inc. Reprinted by permission.

faster at selecting off-screen targets than with two-level zooming or grab-and-drag panning [19]. Moreover, users claim that they are faster with Hop than both the Halo and proxy-based techniques. However, there are some limitations to this technique [19]. First, deciding on targets depends on contextual information. Users may need several teleportations to find the target. Second, the teleportation used in the Hop technique may cause users to lose their current location in the workspace. Finally, like other proxy-based techniques, the number of proxies that can be dis-

	Drag and Pop/Pick	**Vacuum**	**Frisbee**	**Hop**
Principle	Allowing quick access to remote areas of pen- and touch-operated displays.	Facilitating manipulation of distant objects of a large display.	Interacting with areas of a large display.	Using principals of halo and drag-and-pop techniques.
Advantage	No crossing bezels No walking towards remote objects	Predictable Consistent Minimized physical travel	Support of multiple concurrent users Minimized physical movement Minimized visual disruption	Rapid navigation to off-screen content Awareness of off-screen content Minimal amount of navigation to access off-screen content
Drawback	Layout of proxies Disrupted invocation of proxies	Scale of proxies	Significant interaction overhead for accessing information	A large number of teleportations for a certain number of tasks

Table 2.3: Principles, advantages, and disadvantages of the proxy and portal-based techniques

played simultaneously is limited. Table 2.3 summarizes the features as well as benefits and drawbacks of proxy and portal-based techniques.

2.4 Hybrid Techniques

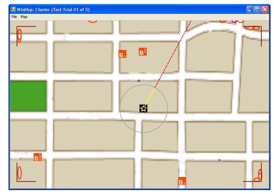
Hybrid techniques integrate complementary properties of multiple prior navigation systems. This integration can significantly improve the performance upon a wider range of off-screen navigation tasks [18, 25].

2.4.1 WinHop

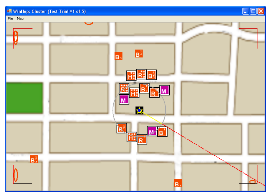
WinHop [18, 25] is a hybrid technique to access off-screen content by hopping through windows. As the name implies, it is based on Hop and portals (or windows). Like Hop, WinHop uses halos and proxies as the mechanism for finding off-screen objects, and uses a portal-based method to access them quickly through an inset window. However, WinHop does not include the shortcomings of the both techniques; Hop lacks the ability to provide contextual information, and portal-based techniques, such as Frisbee, do not facilitate navigation to the region. The reason is that, unlike Hop that teleports to the location of the off-screen target by tapping on a proxy, in WinHop a window starts opening when a user taps on a proxy. This window enlarges to a specific size, and then shows the off-screen target and its surroundings without actually leaving the current location. Zooming and panning are enabled in this window. Figure 2.12 shows the process of WinHop to locate off screen targets.

The problem of WinHop is that it uses animation to access off screen content that does not facilitate browsing off-screen content.

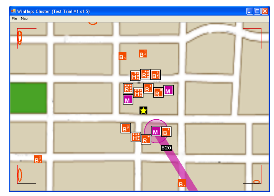
(a) The shape and direction of halos.



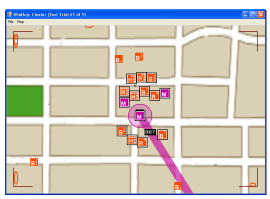
(b) Creating laser beam related proxy.



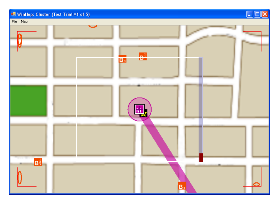
(c) Intersection of laser beam and halos bring proxies of objects to the view.



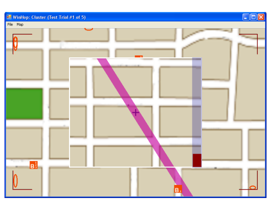
(d) Hovering on proxies shows the direction of the corresponding off-screen object and its distance from its related proxy.



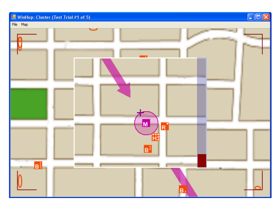
(e) Clicking on a proxy shifts the proxy to the center of the map besides the star.



(f) Then a window/portal starts growing with the center of the proxy.



(g) The window shows the translation from proxy to off-screen object by animation.



(h) Finally, animation stops when reaching to the off-screen target.

Figure 2.12: The process of WinHop to access off-screen objects.

2.4.2 Multiscale Zoom

Multiscale Zoom [18, 25] is another hybrid technique that combines the properties of Zooming and proxy based approaches. The mechanism is such that in zoomed-out view (overview) objects are presented bigger than their real size (proxy). This presentation shows the full detail of the objects (Figure 2.13).

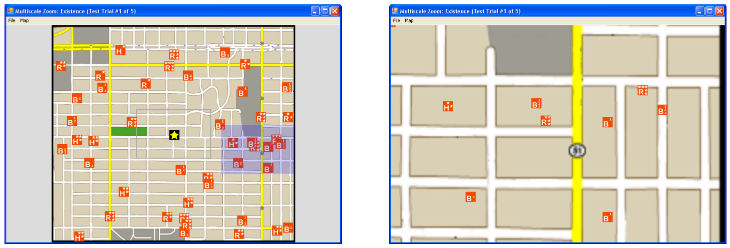
(a) Zoomed-out view with conventional Zoom.

(b) Zoomed-out view with Multiscale Zoom.

Figure 2.13: Unlike traditional zoom, in Multiscale Zoom objects maintain their original size. Image from [18, 25] © 2007 authors

Figure 2.14 illustrates how Multiscale Zoom works; by clicking on the workspace, users can switch between zoom in (b) and zoom out (a) view of the workspace. When the number of objects increases, the technique leads to a cluttered overview. The reason is that the proxies take more space than their corresponding real objects. Multiscale Zoom tries to address this issue by bringing the hovered proxy to the top of the stack so that users can recognize the details of the hovered proxy. However, it does not allow users to see the details of the clustered objects at the same time.

Table 2.4 summarizes the properties as well as pros and cons of the two above-



(a) Multiscale Zoom provides the overview of the workspace while the size of the objects remains the same.

(b) Clicking on the workspace goes to the zoom-in view of the workspace using animation.

Figure 2.14: Multiscale Zoom shows the full detail of objects in zoomed out view.

	WinHop	**Multiscale Zoom**
Principle	Based on Hop, portal, and Zoom.	Based on proxy and Zoom.
Advantage	No disorientation	Fast Easy
Drawback	Time consuming due to using animation Unfixed arrangement of proxies No distance information by proxies arrangement	Overlapping proxies

Table 2.4: Principles, advantages, and disadvantages of the two hybrid techniques.

mentioned hybrid techniques.

Chapter 3

Design and Implementation

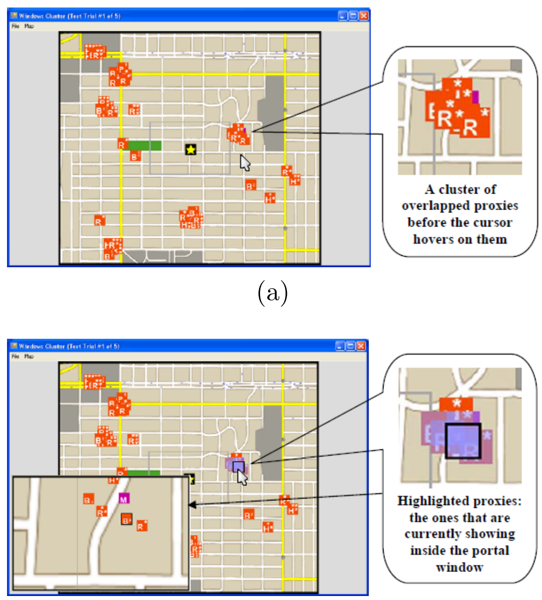
I implemented three novel navigation techniques. I called them Multiscale Window, Spring, and Crystal Ball. Multiscale Window and Spring are improvements to Multiscale Zoom technique. Like Multiscale Zoom, they provide an overview of the content and use proxies to present bigger and detailed-views of the real objects. However, they try to solve the problem of overlapping proxies. Crystal Ball is based on the WinHop technique. It similarly employs halos, proxies, and a portal to locate off-screen targets. The advantage of Crystal Ball is that it facilitates the interaction with off-screen content.

3.1 Multiscale Window

As I mentioned earlier, the Multiscale Window technique was designed to address the problem of overlapping proxies in the Multiscale Zoom technique. To solve the issue, this technique adds a portal-based approach to Multiscale Zoom in order to provide access to off-screen content without actually leaving the current main view.

The Multiscale Window technique can recognize the overlapped proxies such that whenever the cursor hovers on them a portal window automatically pops up showing the zoomed-in view of the corresponding objects of those overlapped proxies. The technique highlights the proxies that currently show their corresponding objects in entirety, in both full width and length inside the portal window. Furthermore, to provide users with feedback, the technique draws a black square around the hovered proxy as well as its corresponding object inside the window. This facility helps users to recognize which proxy/object is currently under their cursor and to help select the correct target.

In order to prevent overlapping of the portal window and hovered overlapped proxies, the program divides the viewport into four equal regions: top left, top right, bottom right, and bottom left. The portal window appears in the opposite direction of the cursor location. In other words, if the cursor hovers on any overlapped proxies on the top left hand side of the viewport, the portal window appears on the bottom right of the viewport. The portal window disappears automatically when the user moves the cursor away from the overlapped proxies. The function of the Multiscale Window technique is shown in Figure 3.1 where (a) no overlapped proxies are under the cursor and (b) the cursor hovers on a cluster of overlapped proxies.



(a)

(b)

Figure 3.1: The process of the Multiscale Window technique to access the details of overlapped proxies. (a) before the cursor hovers on an instance of overlapped proxies and (b) after that.

3.2 Spring

Like Multiscale Window, the Spring technique was designed to address the problem of overlapping proxies in the Multiscale Zoom technique. To address the problem, whenever the cursor hovers on any overlapped proxies, the Spring technique auto-

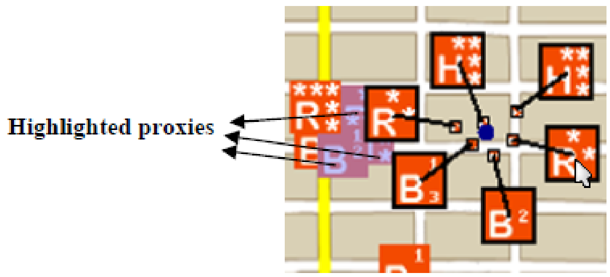
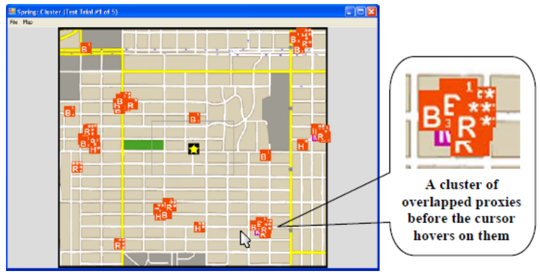


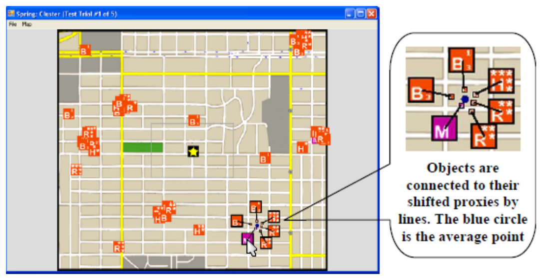
Figure 3.2: The background proxies that overlap with the shifted proxies are highlighted in the Spring technique.

matically moves those proxies around in a circular manner. This feature enables users to easily recognize the details of the overlapped proxies as well as the locations of the corresponding objects. The algorithm first calculates the average coordinate from the corresponding real objects of the hovered overlapped proxies. This average point is presented in the form of a small blue circle in the program. Afterwards, for each of those objects, the program obtains the angle between the horizontal axis and the straight line passing through the object and the calculated average point. Then every proxy is shifted to a new location with a specific distance from the average point and with its own calculated corresponding angle. Finally, the program shows the real objects connected to their corresponding shifted proxies by straight lines. Additionally, in cases that the shifted proxies cover any other proxies outside the hovered overlapped cluster, those proxies are highlighted to separate them from the shifted proxies (Figure 3.2). The function of the Spring technique is illustrated in Figure 3.3 where (a) no overlapped proxies are under the cursor and

(b) the cursor hovers on a cluster of overlapped proxies.



(a)



(b)

Figure 3.3: The process of the Spring technique to access the details of overlapped proxies. (a) before the cursor hovers on an instance of overlapped proxies and (b) after that.

There were two main challenges to implement the Spring technique. The first challenge was overlapping the shifted proxies with each other or with the real objects. This can occur when the number of overlapped proxies in a cluster increase or when the angles of some proxies are close to each other. To solve the problem, I used an algorithm that rotated the shifted proxies. It rotated the overlapped proxy until there was no overlapping with other shifted proxies or real objects. If there is not such a space the algorithm adds the distance from the average point until the overlapping removes.

The second challenge was shifting the overlapping proxies which were close to the edge of the screen. The reason was that some of the shifted proxies went outside the viewport so that users hardly could see them. To address the issue, I once again used the rotation algorithm; it rotated the outside-of-view proxies until they came to the view and after that again checked for overlapping with other proxies similar to solution for the first challenge.

3.3 Crystal Ball

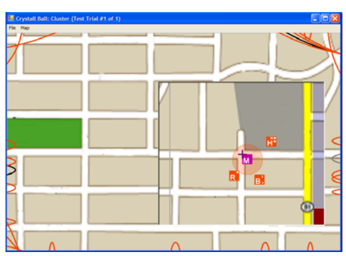
Crystal Ball was designed to overcome the shortcoming of the WinHop technique to access off-screen content. As I mentioned earlier, it employs halos, proxies, and portal-based techniques to locate off-screen targets similar to WinHop. The differences include the shape of halos, the method to invoke halos, and the way the off-screen content is shown in a portal window. First, I implemented the version of halos used in Hop [22] (i.e. circle shape). In addition, to be persistent and remove the bias, I changed the shape of halos in WinHop to circles, too. Second, instead of using the "laser beam" (the method used in WinHop) to invoke halos, Crystal Ball enables users to directly select their desired halo by clicking on them. Lastly, unlike WinHop which uses animation, my new developed technique directly investigates the content of the selected halo without any delay.

(a) Crystal Ball uses the same shape of halos as Hop.

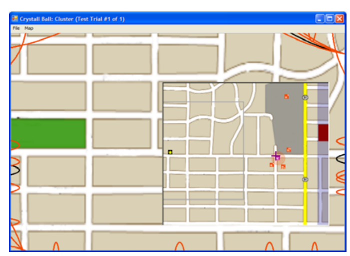
(b) Clicking on a halo, shows its proxy.

(c) Dragging the halo or its proxy, grows a window.

(d) The window keeps growing until it reaches its maximum size.



(e) The window stays on the screen when reached its maximum size.



(f) Inside the window, users can zoom using the red scroll bar in the right hand side of the window.

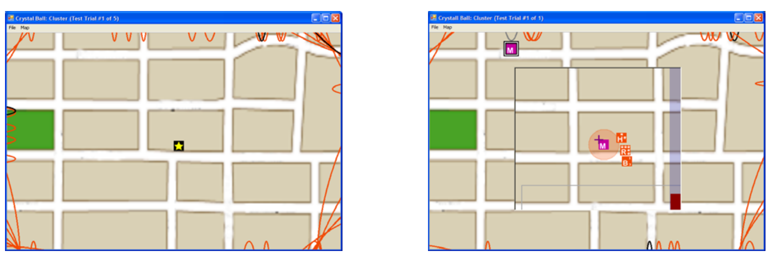
Figure 3.4: The procedure of the first version of Crystal Ball.

Users can select any halos by directly clicking on them. This causes the corresponding proxy to appear on the cusp of the halo. To see the related off-screen content of the selected halo, users need to drag either the halo or its proxy toward the center of the workspace. Alongside the dragging, a portal window, which contains the content of the selected object, starts growing. I designed Crystal Ball to have a feature that enabled the portal window to slide in and out from the screen's edges. It means that dragging toward the viewport' center causes the window to enlarge to a maximum size and dragging toward screen edges makes the window to shrink and finally disappear. Once being maximized, the window can stay on the screen; otherwise it slides out toward the edge that the selected halo is located. Similar to WinHop, panning and zooming are allowed inside the portal window; users can pan by dragging the cursor and zoom by moving a slider to the right side of the portal window. Figure 3.4 demonstrates how Crystal Ball uses the sliding feature of the portal window.

3.4 Design Desicions

I ran a pilot experiment to evaluate the three developed techniques and to obtain the total time for the experiment. The Spring technique did not show any improvement over the Multiscale Zoom technique speciallay when the number of overlapping proxies increased or overlapped proxies located on the border region of the workspace. Also because of the total time it took participants to run the experiment, I decided to exclude the Spring technique from the real experiments.

In addition, I decided to exclude the sliding feature of the Crystal Ball technique. The reason was that the results showed a slight difference between the Crystal Ball and WinHop techniques. Furthermore, participants complained about accidentally loosing the portal window and difficulty leaving the portal in the center of the screen. As a result, I changed the first version of Crystal Ball in order to simplify the technique. In the second/final version, whenever users click on any halos, they can simultaneously see both the related proxy and its surroundings in a fixed-size window. Figure 3.5 shows the procedure of Crystal Ball to provide access to a desired object outside the viewport.



(a) Crystal Ball uses the same shape of halos as Hop.

(b) Clicking on a halo shows both its proxy and content in a fixed-size window

Figure 3.5: The procedure of the second/final version of Crystal Ball.

Chapter 4

Evaluation and Results

As I mentioned in chapter 3, I excluded the Spring technique from the real experiment after conducting a pilot test. Two experiments were designed to evaluate the effectiveness of the Multiscale Window and Crystal Ball techniques. In the first experiment, I evaluated the techniques on a Tablet PC using a digital pen as an input device. The results from the first experiment led to the conduction of the second experiment which evaluated the techniques on a normal PC using a mouse device.

4.1 Hypotheses

In my research I investigated the validity of the following hypotheses.

- Hypothesis 1: Using the Multiscale Window technique, users will access off-screen objects faster than the Multiscale Zoom technique, because they will not need switching between different views of the workspace (zoomed-in and zoomed-out views).

- Hypothesis 2: Users will find the targets more accurately with Multiscale Window rather than the Multiscale Zoom technique, because of using one view.

- Hypothesis 3: Using the Crystal Ball technique, users will access off-screen objects faster than the WinHop technique, because they will not need to wait untill animation takes them to any target.

- Hypothesis 4: Users will find the targets more accurately with Crystal Ball rather than the WinHop technique, because they will not be distracted by the layout of the proxies.

4.2 Experiment One - Using a Tablet PC

The purpose of the first experiment is to evaluate the effectiveness of Multiscale Window and Crystal Ball in comparison to Multiscale Zoom and WinHop, respectively. In this experiment, subjects were required to locate targets in every task.

4.2.1 Participants

Fourteen undergraduate students (12 males and 2 females) participated in the experiment and were assigned to one of the conditions. Subjects were volunteers from a computer science course in human-computer interaction. Their ages ranged from 20 to 24 years. All of the participants were familiar with the Windows software as well as map browsing applications; however, only some of them had experience using a digital pen. Moreover, none of them had any previous experience with the four techniques used in the experiment, although they were familiar with the zooming concept.

4.2.2 Materials

I built the experimental system in C#.NET platform and deployed it on an HP Pavilion tx2500 Entertainment Notebook PC running Windows Vista. The Tablet's Display resolution was set to 800×600 pixels. The system presented a map browsing application simulated from the city of Saint Paul, Minnesota. In order to provide an off-screen environment, I chose a visual workspace which contained a 2600×2400 pixel map. In other words, the workspace was larger than the viewport so that there was equally a 900 pixels distance from the workspace's edges to the viewport. In a zoomed-in view, users could see only the viewport while they could see the entire workspace in zoomed out view.

The system also displayed several icons on top of the city map. Icons were squares of 24×24 pixels and provided users with three types of information: their category, colour, and details. Four different categories were used that included hotels, restaurants, bus stops, and metro stations. They were presented in the form of capital letters H, R, B, and M respectively. Two different colours that were used included orange for hotels, restaurants, and bus stops and purple for metro stations. Finally, small symbols were used to represent further information of an icon. This information contained the number of stars (from 1 to 5) for hotels and restaurants and the stop number for bus stops (e.g. 3, 23, and 123). There was no detailed information for metro stations.

The system randomly presented fifty icons on the workspace; forty five outside the viewport and five inside the viewport. The arrangement of icons was such that on the one hand they did not overlap each other in a zoomed-in view. On the other hand, six clusters including six overlapped icons were shown in a zoomed-out view. All of these clusters were located outside of the viewport. I also placed a 24×24 pixel star icon in the center of the map in order to prevent disorientation while navigating.

4.2.3 Tasks

In the experiment, users were presented with three different tasks which were Cluster, Proximity Between Objects, and Proximity from Reference. Since the potential targets were located in one of the six clusters outside the viewport, off-screen navigation was required for all the tasks. There were only three potential targets and only one of them was the correct target in all of the tasks. For Crystal Ball and WinHop, I highlighted the corresponding halos of potential targets in black so that users could recognize them fast. Subjects could complete every task by only clicking on the correct target specified for that task. The details of each task are described as follows:

Cluster: Before the task started, a cluster of four icons were displayed on the screen accompanied by a written instruction to accomplish the task. The icons included a four-star hotel, a four-star restaurant, bus stop number 123, and finally a metro station (Figure 4.1). The participants were asked to find such a cluster with exactly those four targets. The program randomly placed three clusters on the workspace from which only one contained the correct targets.

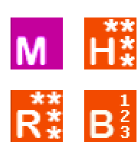


Figure 4.1: Users had to find a tight group of landmarks with exactly the targets shown above.

Proximity Between Objects: Before the task started, the program displayed a metro station and a four-star hotel on the screen and asked users to find the metro station that was closest to a four-star hotel. Three metro-hotel pairs were randomly placed on the map from which only one pair was the closest one. The task was designed to be simple so that users did not need to use any precise measurement to accomplish it.

Figure 4.2: Users had to find the metro station that was closest to a four-star hotel.

Proximity from Reference: Before the task started, the program displayed a four-star hotel and the star icon on the screen and asked users to find the closest four-star hotel to the star icon. Similarly, the system randomly created three potential targets and only one of them was the correct target.

Figure 4.3: Users had to find the closest four-star hotel to the star in the center of the map.

4.2.4 Design and Procedure

The study used a 4×3 within-participants factorial design. The factors were:

- Navigation technique: Multiscale Window, Crystal Ball, WinHop, and Multiscale Zoom
- Task: Cluster, Proximity Between Objects, and Proximity form Reference

To reduce learning effects, the navigation technique and task were fully counterbalanced using a Latin square. The experiment was designed such that it included one training trial, two practice trials, and five test trials. Participants were asked to perform the test trials as fast and as accurately as possible. For each technique, users had to perform all three tasks one after another. After finishing the third task, the technique was changed and the procedure was repeated. After ending each trial, the program reset the workspace to its initial state; i.e. the viewport was in the center. The study averaged half an hour per user, and participants could rest between the trials.

With 14 participants, 4 navigation techniques, 3 tasks, and 5 trials per condition, the system recorded a total of $14 \times 4 \times 3 \times 5 = 840$ trials. The system collected completion times and error rates for each target.

4.2.5 Results and Discussion

To test the above-mentioned hypotheses in section 4.1, I measured participants' performance on the given tasks with respect to the completion time and their accuracy to find and click on the correct target. The study averaged half an hour per user. After collecting the data, I separated the same tasks from the data into the same files. Therefore, three different files were obtained for the three tasks. I then removed the influence of outliers from the data for every task; 7 from Cluster, 6 from Proximity Between Objects, and 3 from Proximity from Reference tasks, respectively. The analysis was performed on the average values and an alpha level of 0.05 was used for all statistical tests. For each task, I carried out a one-way ANOVA to look for effects of navigation technique.

Completion Time

- For Cluster task, figure 4.4 shows the average completion times by navigation technique. A One-Way ANOVA showed a significant effect for technique on time ($F(3,269)=8.293$, $P<0.001$). WinHop was the slowest technique among

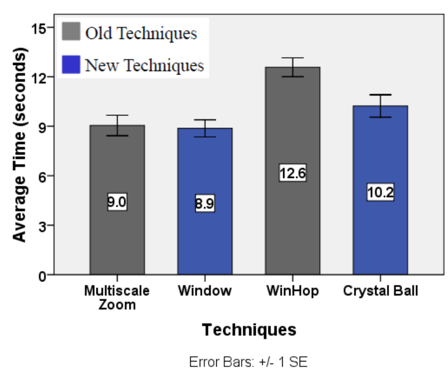


Figure 4.4: Experiment One: Average completion time in seconds for the Cluster task.

the four techniques (12.6 seconds) followed by Crystal Ball (10.2 seconds), Multiscale Zoom (9 seconds), and Multiscale Window (8.9 seconds). The post-hoc tests revealed a significant difference between Crystal Ball and WinHop, but not between Multiscale Window and Multiscale Zoom (SPSS output of the analysis is in Appendix A).

- For Proximity Between Objects task, figure 4.5 shows the average completion times by navigation technique. A One-Way ANOVA showed a significant effect for technique on time ($F(3,270)=23.813$, $P<0.001$). WinHop was again

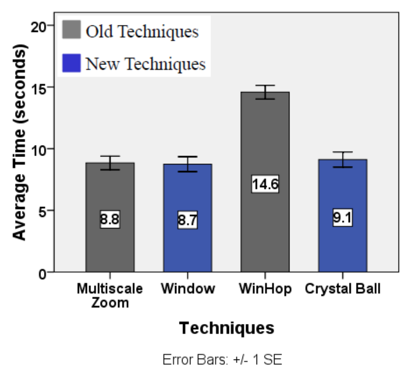


Figure 4.5: Experiment One: Average completion time in seconds for the Proximity Between Objects task.

the slowest technique among the four techniques (14.6 seconds) followed by Crystal Ball (9.1 seconds), Multiscale Zoom (8.8 seconds), and Multiscale Window (8.7 seconds). The post-hoc tests revealed a significant difference between Crystal Ball and WinHop, but not between Multiscale Window and

Multiscale Zoom (SPSS output of the analysis is in Appendix A).

- For Proximity from Reference, figure 4.6 shows the average completion times by navigation technique. A One-Way ANOVA showed a significant effect for technique on time ($F(3,273)=57.621$, $P<0.001$). This time Crystal Ball was the slowest technique among the four techniques (20.9 seconds) followed by WinHop (15.2 seconds), Multiscale Window (7.7 seconds), and Multiscale Zoom (6.8 seconds). The post-hoc tests revealed a significant difference between Crystal Ball and WinHop, but not between Multiscale Window and Multiscale Zoom (SPSS output of the analysis is in Appendix A).

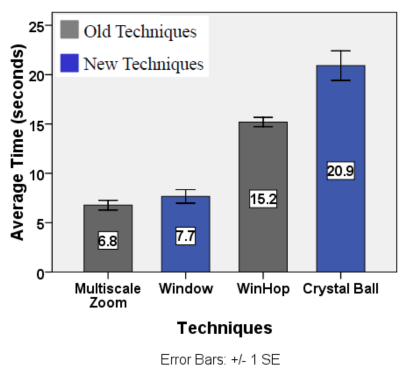


Figure 4.6: Experiment One: Average completion time in seconds for the Proximity from Reference task.

Error Rate

During the experiment, each trial finished whenever users clicked on a target object. The system calculated error rate as the number of clicks that users missed to hit the target for each trial (mis-clicks).

- For Cluster task, figure 4.7 shows the average number of mis-clicks by navigation technique. Although it seemed that users were more precise with Multiscale Window than other techniques, this difference was not statistically significant (F(3,269)=0.605, P=0.612). Moreover, the post-hoc tests confirmed that there was no significant difference between the two compared pairs; Multiscale Window vs Multiscale Zoom and Crystal Ball vs WinHop (SPSS output of the analysis is in Appendix A).

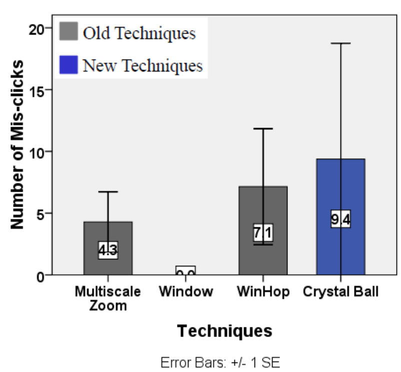


Figure 4.7: Experiment One: Average number of mistakes by navigation technique for the Cluster task.

- For Proximity Between Objects task, figure 4.8 shows the average number of mis-clicks by navigation technique. A One-Way ANOVA did not show any significant effect for technique on error rate (F(3,270)=1.107, P=0.347), and the post-hoc tests confirmed the result (SPSS output of the analysis is in Appendix A).

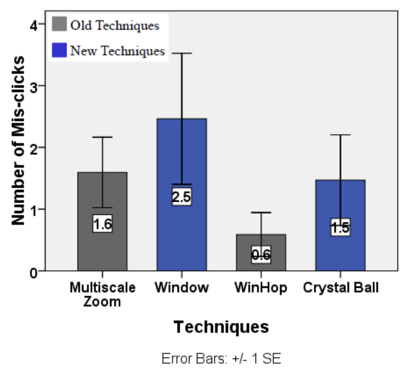


Figure 4.8: Experiment One: Average number of mistakes by navigation technique for the Proximity Between Objects task.

- For Proximity from Reference task, figure 4.9 shows the average number of mis-clicks by navigation technique. A One-Way ANOVA showed a significant effect for technique on error rate (F(3,273)=3.373, P=0.019). Users were more accurate with WinHop followed by Multiscale Zoom, Multiscale Window, and Crystal Ball. The post-hoc tests revealed that there was a significant difference between Crystal Ball and WinHop (SPSS output of the analysis is in Appendix A).

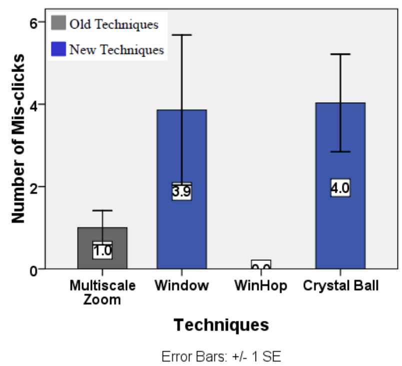
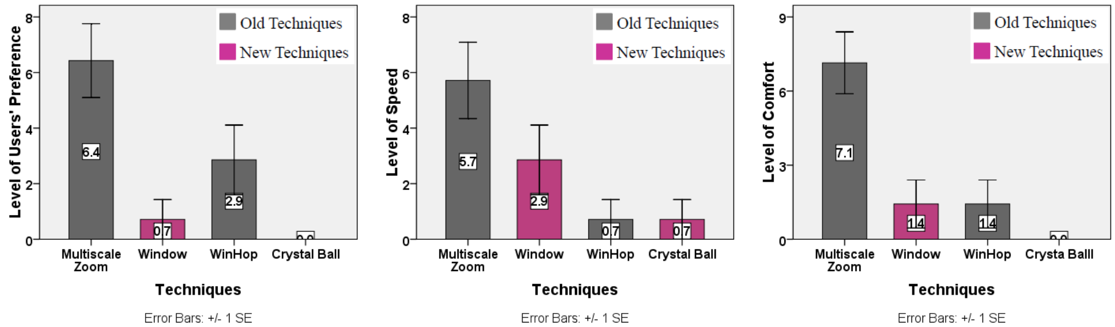


Figure 4.9: Experiment One: Average number of mistakes by navigation technique for the Proximity from Reference task.

Users' Feedback

Participants were asked to fill out a brief questionnaire (Appendix B) about the techniques at the end of the experiment. The questionnaire included three questions. The first question asked them which technique they liked the most. The second question asked them which technique was fastest. Finally the last question asked them which technique was easiest to use (SPSS output of the analysis is in Appendix A).

Figure 4.10 presents the obtained result. The post-hoc tests showed that subjects liked Multiscale Zoom and found it easier than the Multiscale Window technique. There was not any significant difference between the CrystalBall and WinHop techniques regarding preference, speed, and comfort to use the techniques.



(a) (b) (c)

Figure 4.10: Experiment One: (a) Users' preference, (b) Level of speed for users, and (c) Level of comfort for users.

Users commented that they had difficulty clicking on halos with the digital pen in Crystal Ball. This is not surprising as a pen input is subject to parallax errors. Most users complained about the arrangement of the proxies in WinHop, since it was not consistent (i.e. the arrangement changed every time that users swept the laser beam) and it did not provide relative distance information (i.e. users could not obtain the distance between two objects just by looking at the locations of their related proxies). Also one user mentioned that he expected the size of the halos to be associated with their distance in both Crystal Ball and WinHop. Another user said that Crystal Ball was faster because the potential targets were highlighted. He mentioned that without providing such clues the technique would not be fast. Moreover, one participant stated that his arm was blocking the portal window in the Multiscale Window technique. As a result, he had to move his arm to see the content inside the window.

Discussion

The completion time for the tested techniques indicated that Crystal Ball was faster than WinHop in all the tasks except for Proximity from Reference. The reason was that WinHop provided users with a facility to show the exact distance of an off-screen target from the center of the map, while Crystal Ball lacked this clue; i.e. in Crystal Ball users had to estimate the shortest distance to the center of the map themselves. As a result, my third hypothesis was confirmed in all tasks except in Proximity from Reference. From another view, Multiscale Window and Multiscale Zoom performed approximately the same in all tasks with no significant difference. Thus, my first hypothesis was rejected. Regarding accuracy, no significant difference was seen between the two compared pairs (Multiscale Window vs Multiscale Zoom and Crystal Ball vs WinHop) except in the Proximity from Reference task; WinHop had less error rates than Crystal Ball. As a result, my second and forth hypotheses were rejected.

In both the Crystal Ball and Multiscale Window techniques, some of the user's interaction area was overlapped due to using a window/portal to access off-screen content. In some cases also the user's arm covered the window so that users need to move their arm to see the content inside the window. Furthermore, users claimed that they had difficulty using a digital pen as an input device to work with the two techniques. Therefore, I decided to run the second experiment using a different device. Besides, I decided to add a distance indicator ability to the Crystal Ball technique to improve the efficiency of the technique in the Proximity from Reference task.

4.3 Experiment Two - Using a PC with a Mouse Device

The results from the first experiment indicated that the new developed techniques were not ideal for a pen-based environment. Therefore, the second experiment was designed to evaluate the previous tested techniques using a PC with a mouse device. The design was very similar to the experiment one with a 4×3 within-participants design (4 navigation techniques and 3 tasks).

4.3.1 Participants

Fourteen male students participated in the experiment and were assigned to one of the conditions. Subjects were 12 volunteer undergraduate students from a computer science course in human-computer interaction, and 2 volunteer graduate students from the Computer Science Department. Their ages ranged from 20 to 30 years. All of the participants were familiar with mouse-and-window software as well as map browsing applications. None of them participated in the first experiment or had any previous experience with the four techniques used in the experiment, although they were familiar with the zooming concept.

4.3.2 Materials

The materials used for this experiment were similar to the ones in experiment one; I used the same program including the same map, icons, and their arrangements. However, this time I ran the experiment on a P4 Windows XP PC system. The display was a 17" monitor set to 800×600 resolution and a normal mouse was used as an input device.

In addition, I added a feature to the Crystal Ball technique in order to provide distance information. Providing such information is helpful especially in the Proximity from Reference task. The system added this information whenever the user clicks on any desired halo. A black rectangle appears on the tip of the halo and shows the distance of the corresponding object to the center of the map (Figure 4.11).

Figure 4.11: Showing the distance information when clicking on a halo.

4.3.3 Tasks

The tasks for this experiment were exactly the same as the ones in experiment one: Cluster, Proximity Between Objects, and Proximity from Reference. Only this time, users selected the targets by using the left button of the provided mouse.

In experiment one, users could only click on the target object inside the portal window in Crystal Ball. To make the Crystal Ball technique faster, this time I enabled users to select the target by either clicking on its proxy or its object inside the portal window.

4.3.4 Design and Procedure

The experimental set up was also similar to experiment one: The study used a 4×3 within-participants factorial design. The factors were:

- Navigation technique: Multiscale Window, Crystal Ball, WinHop, and Multiscale Zoom
- Task: Cluster, Proximity form Reference, and Proximity Between Objects

To reduce learning effects, the navigation technique and task were fully counterbalanced using a Latin square. The experiment was designed such that it included one training trial, two practice trials, and five test trials. Participants were asked to perform the test trials as fast and as accurately as possible. For each technique, users had to perform all three tasks one after another. After finishing the third task, the technique was changed and the procedure was repeated. After ending each trial, the program reset the workspace to its initial state; i.e. the viewport was in the center. The study averaged half an hour per user, and participants could rest between the trials.

With 14 participants, 4 navigation techniques, 3 tasks, and 5 trials per condition, the system recorded a total of $14 \times 4 \times 3 \times 5 = 840$ trials. The system collected completion times and error rates for each target.

4.3.5 Results and Discussion

To test the four hypotheses mentioned in section 4.1, I measured participants' performance on the given tasks with respect to completion time and their accuracy to click on the right target. Like Experiment One, the study averaged half an hour per user. After collecting data, I separated the same tasks from the data into the same files. Therefore, three different files were obtained for the three tasks. I then removed the influence of outliers from the data for every task; 2 from Cluster, 2 from Proximity Between Objects, and 7 from Proximity from Reference tasks, respectively. The analysis was performed on the average values and an alpha level of 0.05 was used for all statistical tests. For each task, I carried out a one-way ANOVA to look for effects of navigation technique.

Completion Time

- For Cluster task, figure 4.12 shows the average completion times by navigation technique. A One-Way ANOVA showed a significant effect for technique on time (F(3,274)=24.364, P<0.001). WinHop was the slowest technique among

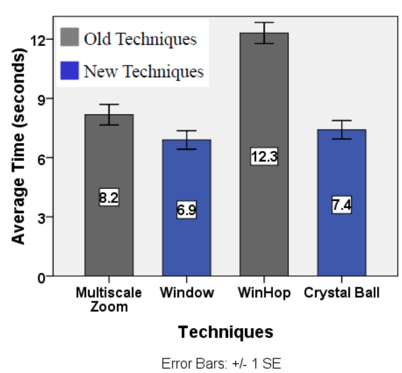


Figure 4.12: Experiment Two: Completion time in seconds for the Cluster task.

the four techniques (12.3 seconds) followed by Multiscale Zoom (8.2 seconds), Crystal Ball (7.4 seconds), and Multiscale Window (6.9 seconds). The post-hoc tests revealed a significant difference between Crystal Ball and WinHop, but not between Multiscale Window and Multiscale Zoom (SPSS output of the analysis is in Appendix A).

- For Proximity Between Objects task, figure 4.13 shows the average completion times by navigation technique. A One-Way ANOVA showed a significant effect for technique on time ($F(3,274)=17.986$, $P<0.001$). WinHop was again

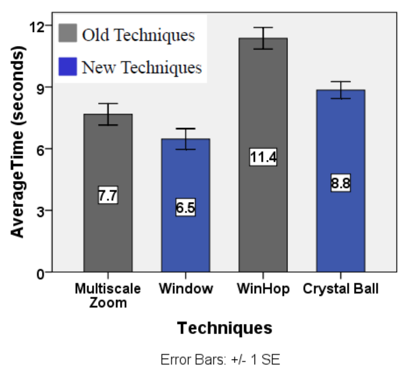


Figure 4.13: Experiment Two: Completion time in seconds for the Proximity Between Objects task.

the slowest technique among the four techniques (11.4 seconds) followed by Crystal Ball (8.8 seconds), Multiscale Zoom (7.7 seconds), and Multiscale Window (6.5 seconds). The post-hoc tests revealed a significant difference between Crystal Ball and WinHop, but not between Multiscale Window and

Multiscale Zoom (SPSS output of the analysis is in Appendix A).

- For Proximity from Reference, figure 4.14 shows the average completion times by navigation technique. A One-Way ANOVA showed a significant effect for technique on time ($F(3,269)=95.800$, $P<0.001$). Similar to both previous tasks, WinHop was the slowest technique among the four techniques (13.6 seconds) followed by Crystal Ball (9.1 seconds), Multiscale Window (4.8 seconds), and Multiscale Zoom (4.4 seconds). The post-hoc tests revealed a significant difference between Crystal Ball and WinHop, but not between Multiscale Window and Multiscale Zoom (SPSS output of the analysis is in Appendix A).

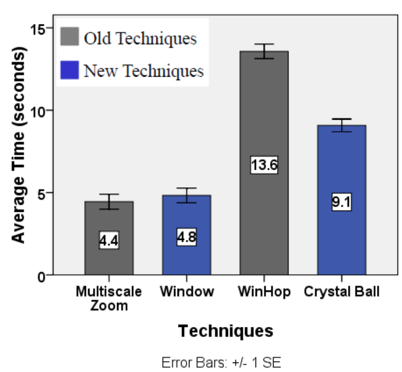


Figure 4.14: Experiment Two: Completion time in seconds for the Proximity from Reference task.

Error Rate

Like experiment One, The system calculated error rate as the number of clicks that users missed to hit the target for each trial (mis-clicks).

- For Cluster task, figure 4.15 shows the average number of mis-clicks by navigation technique. Although it seemed that users were more precise with Multiscale Window than other techniques, this difference was not statistically significant (F(3,274)=2.349, P=0.073). Moreover, the post-hoc tests confirmed that there was no significant difference between the two compared pairs; Multiscale Window vs Multiscale Zoom and Crystal Ball vs WinHop (SPSS output of the analysis is in Appendix A).

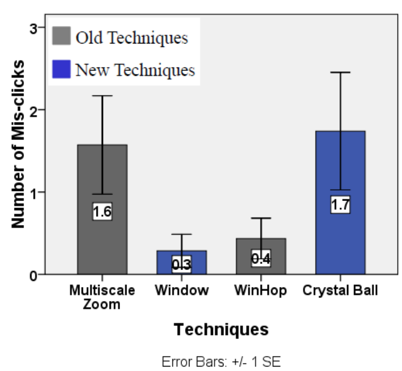


Figure 4.15: Experiment Two: Average number of mistakes by navigation technique for the Cluster task.

- For Proximity Between Objects task, figure 4.16 shows the average number of mis-clicks by navigation technique. A One-Way ANOVA did not show any main difference for technique on error rate (F(3,274)=1.380, P=0.249), and the post-hoc tests confirmed the result (SPSS output of the analysis is in Appendix A).

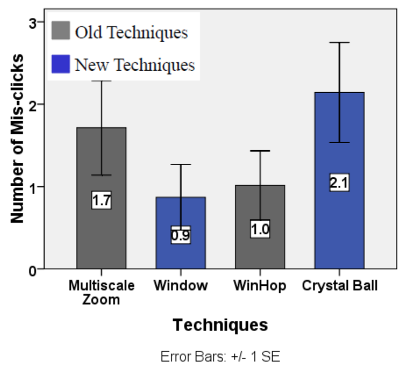


Figure 4.16: Experiment Two: Average number of mistakes by navigation technique for the Proximity Between Objects task.

- For Proximity from Reference task, figure 4.17 shows the average number of mis-clicks by navigation technique. Like previous tasks, a One-Way ANOVA did not show any main difference for technique on error rate (F(3,269)=1.161, P=0.325), and the post-hoc tests confirmed the result (SPSS output of the analysis is in Appendix A).

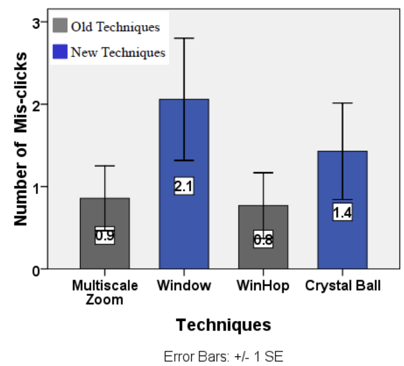


Figure 4.17: Experiment Two: Average number of mistakes by navigation technique for the Proximity from Reference task.

Users' Feedback

This time, I used a Likert-scale questionnaire (Appendix B) to ask users to rank the tested techniques based on their preference, speed, and convenience of the techniques.

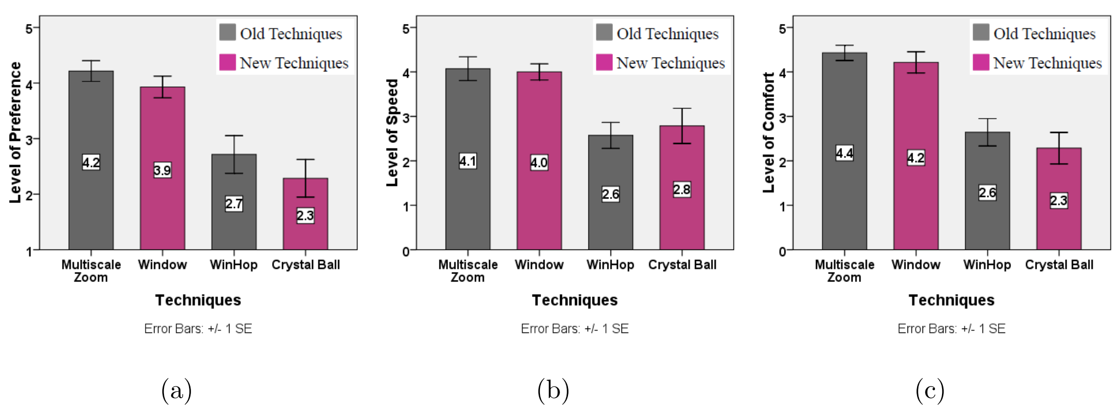


Figure 4.18: Experiment Two: (a) Users' preference, (b) Level of speed for users, and (c) Level of comfort for users.

Figure 4.18 presents the obtained result. This time, the post-hoc tests did not show any significant difference between the two compared pairs. Users found the Multiscale Window and Crystall Ball techniques almost the same as the Multiscale Zoom and WinHop techniques correspondingly in preference, speed, and comfort. Multiscale Zoom was ranked as the most desirable, fastest, and easiest technique among all the four techniques (SPSS output of the analysis is in Appendix A).

Like the first experiment, users had difficulty in finding and selecting halos where there was a cluster of them overlapping each other specially on the corners. A user mentioned that he was quickly able to focus on the objects that he wanted to look at in the Multiscale Window technique. Similarly another subject liked Multiscale Window, but mentioned that sensitivity in selecting a target should be decreased.

Discussion

The completion time for the tested techniques indicated that Crystal Ball was faster than WinHop in all the tasks and this supported my third hypothesis. However, like experiment One, the Multiscale Window and Multiscale Zoom techniques performed approximately the same in all the tasks with no significant difference. Thus, like experiment One, my first hypothesis was rejected. Regarding accuracy, no significant difference was seen between the two compared pairs (Multiscale Window vs Multiscale Zoom and Crystal Ball vs WinHop), and this rejected my second and forth hypotheses as well.

Crystal Ball showed improvement to WinHop and even it was as fast as Multiscale Window and Multiscale Zoom in most tasks. The Proximity from Reference task was the only task in which the efficiency of Crystal Ball was not as good as Multiscale Window and Multiscale Zoom. The reason was that in the overview users could compare the distance of clusters to the center of the map just by one glance; however, in Crystal Ball they needed to look at the written distances and then compare them to find the best case that was more time consuming.

I think Crystal Ball can be improved significantly by using different approaches to prevent overlapping halos and by employing different methods to select them. Using the Wedge [10] algorithm is an example of the former improvement to prevent overlapping halos. For the later one, showing proxies instead of halos or showing halos and proxies at the same time can be efficient so that users could select proxies. In my implementation, after clicking on halos, the corresponding proxy appeared.

Chapter 5

Conclusion and Future Work

Nowadays, navigation is a common problem associated with large computer-based information systems which are too expansive to be displayed in their entirety. A huge amount of research has been done to facilitate navigation and quick access to off-screen content on both small screens and large displays. However, each developed technique offers some drawbacks besides its advantages.

In this thesis, I introduced two novel navigation techniques, Multiscale Window and Crystal Ball, to improve the efficiency of two existing navigation techniques, WinHop and Multiscale Zoom. Multiscale Window was based on Multiscale Zoom, and Crystal Ball inherited from the WinHop technique. However, the developed techniques did not include the drawbacks of the two existing techniques; Multiscale

Window tried to solve the problem of overlapping proxies, while Crystal Ball made accessing to off-screen content faster.

I described in detail the design and implementation of my developed techniques and the results of an empirical evaluation intended to test the effectiveness of these techniques in real world tasks. I evaluated the Multiscale Window and Crystal Ball techniques in two experiments. In the first experiment a Tablet PC with a digital pen as an input device was used. Results showed that there was no significant difference between Multiscale Window and Multiscale Zoom. However, Crystal Ball showed improved effects over WinHop in most tasks. The second experiment compared the same techniques as in experiment one, on a PC with a mouse as input device. The results showed that Crystal Ball was faster than WinHop to access off-screen areas for all given tasks, while Multiscale Window did not show any obvious advantage over Multiscale Zoom.

5.1 Contributions

The main contributions of this research are:

- A detailed summary of the related work conducted in this field on both small screen devices such as GPS and large screens such as interactive walls and multiple displays.

- Introducing three new navigation techniques to access off-screen content; Multiscale Window, Crystal Ball, and Spring to improve the efficiency of the two existing hybrid navigation methods, WinHop and Multiscale Zoom.

- Evaluation of the new interaction methods with three spatially relative tasks. The evaluation was carried out on both a small screen with a digital pen as an input device and a normal PC using a mouse.

5.2 Future Work

I intend to investigate the applicability of my developed techniques to other kinds of large virtual spaces such as API (Document Application Programming Interface), web pages, GPS (Geographical Information Systems), spreadsheets, and image browsing. Also I would like to test my developed techniques in a real world scenario where more factors may affect the results. Moreover, investigating the solutions to overcome the problem of covering the user's interaction space by the portal window in the Multiscale Window and Crystal Ball techniques is my next step. Finally, more work must be done to help users to better select halos when using a digital pen or touch pad in the Crystal Ball technique.

Bibliography

[1] P. Baudisch, E. Cutrell, D. Robbins, M. Czerwinski, P. Tandler, B. Bederson, and A. Zierlinger. Drag-and-pop and drag-and-pick: techniques for accessing remote screen content on touch-and pen-operated systems. In *Proceedings of the Interact Conference*, pages 57–64, 2003.

[2] P. Baudisch, N. Good, V. Bellotti, and P. Schraedley. Keeping things in context: A comparative evaluation of focus plus context screens, overviews, and zooming. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 259–266, 2002.

[3] P. Baudisch and R. Rosenholtz. Halo: a technique for visualizing off-screen locations. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 481–488, 2003.

[4] B. B. Bederson and J. D. Hollan. Pad++: a zooming graphical interface for

exploring alternate interface physics. In *Proceedings of the 7th annual ACM symposium on User interface software and technology*, pages 17–26, 1994.

[5] B. B. Bederson, J. D. Hollan, J. Stewart, D. Rogers, and D. Vick. A zooming web browser. `http://www.cs.umd.edu/hcil/pad++/papers/bookchap-98-webbrowser/`, Last visited May 2006.

[6] A. Bezerianos and R. Balakrishnan. The vacuum: Facilitating the manipulation of distant objects. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 361–370, 2005.

[7] A. Bezerianos and R. Balakrishnan. View and space management on large displays. *IEEE Computer Graphics and Applications*, 25(4):34–43, 2005.

[8] S. Bjrk. Hierarchical flip zooming: Enabling parallel exploration of hierarchical visualizations. In *Proceedings of the Working Conference on Advanced Visual interfaces*, pages 232–237, 2000.

[9] A. Cockburn and J. Savage. Comparing speed-dependent automatic zooming with traditional scroll, pan, and zoom methods. In *People and Computers XVII: British Computer Society Conference on Human Computer Interaction*, pages 87–102, 2003.

[10] S. Gustafson, P. Baudisch, C. Gutwin, and P. Irani. Wedge: Clutter-free visualization of off-screen locations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*, pages 787–796, 2008.

[11] C. Gutwin. Improving focus targeting in interactive fisheye views. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 267–274, 2002.

[12] C. Gutwin and C. Fedak. Interacting with big interfaces on small screens: a comparison of fisheye, zoom, and panning techniques. In *Proceedings of Graphics Interface (GI '04)*, pages 145–152, 2004.

[13] K. Hinckley, E. Cutrell, S. Bathiche, and T. Muss. Quantitative analysis of scrolling techniques. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 65–72, 2002.

[14] K. Hornbk, B. B. Bederson, and C. Plaisant. Navigation patterns and usability of zoomable user interfaces with and without an overview. *ACM Transactions Computer-Human Interaction*, 9(4):362–389, 2002.

[15] K. Hornbk and E. Frkjaer. Reading of electronic documents: The usability of linear, fisheye, and overview+detail interfaces. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 293–300, 2001.

[16] T. Igarashi and K. Hinckley. Speed-dependent automatic zooming for browsing large documents. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, pages 139–148, 2000.

[17] P. Irani. Presentation. Human-Computer Interaction 2 (COMP 4020) course notes, University of Manitoba, Department of Computer Science, 2005.

[18] P. Irani, C. Gutwin, G. Partridge, and M. Nezhadasl. Techniques for interacting with off-screen content. In *Proceedings of the IFIP International Conference on Human-Computer Interaction (Interact '07)*, pages 234–249, 2007.

[19] P. Irani, C. Gutwin, and X. D. Yang. Improving selection of off-screen targets with hopping. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 299–308, 2006.

[20] J. A. Johnson. A comparison of user interfaces for panning on a touch-controlled display. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 218–225, 1995.

[21] V. Kaptelinin. A comparison of four navigation techniques in a 2d browsing task. In *Conference Companion on Human Factors in Computing Systems*, pages 282–283, 1995.

[22] A. Khan, G. Fitzmaurick, D. Almeida, N. Burtnyk, and G. Kurtenbach. A remote control interface for large displays. In *Proceedings of the 17th Annual ACM Symposium on User interface Software and Technology*, pages 127–136, 2004.

[23] Y. K. Leung and M. D. Apperley. A review and taxonomy of distortion-oriented presentation techniques. *ACM Transactions on Computer-Human Interaction*, 1(2):126–160, 1994.

[24] T. Moscovich and J. F. Hughes. Navigating documents with the virtual scroll ring. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, pages 57–60, 2004.

[25] G. Partridge, M. Nezhadasl, P. Irani, and C. Gutwin. A comparison of navigation techniques across different types of off-screen navigation tasks. In *Proceedings of the IFIP International Conference on Human-Computer Interaction (Interact '07)*, pages 716–721, 2007.

[26] D. C. Robbins, E. Cutrell, R. Sarin, and E. Horvitz. Zonezoom: Map navigation for smartphones with recursive view segmentation. In *Proceedings of the Working Conference on Advanced Visual interfaces*, pages 231–234, 2004.

[27] M. Sarkar and M. H. Brown. Graphical fisheye views of graphs. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 83–91, 1992.

[28] B. Shneiderman and C. Plaisant. *Designing the User Interface: strategies for effective human-computer interaction.* Addison–Wesley, 4st edition, 2005.

[29] A. Skopik and C. Gutwin. Improving revisitation in fisheye views with visit wear. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 771–780, 2002.

[30] E. Smalley. Screen arcs widen view. `http://www.trnmag.com/Stories/2003/050703/Screen_arcs_widen_view_050703.html`, May 2003.

[31] G. M. Smith and M. C. Schraefel. 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*, pages 53–56, 2004.

[32] R. Spence. *Information Visualization.* ACM Press, Pearson Education Limited, 1st edition, 2001.

[33] D. S. Tan, B. Meyers, and M. Czerwinski. Wincuts: Manipulating arbitrary window regions for more effective use of screen space. In *Extended Abstracts on Human Factors in Computing Systems*, pages 1525–1528, 2004.

[34] P. T. Zellweger, J. D. Mackinlay, L. Good, M. Stefik, and P. Baudisch. City lights: Contextual views in minimal space. In *Extended Abstracts on Human Factors in Computing Systems*, pages 838–839, 2003.

[35] S. Zhai, B. A. Smith, and T. Selker. Improving browsing performance: a study of four input devices for scrolling and pointing tasks. In *Proceedings of the IFIP TC13 Interantional Conference on Human-Computer Interaction (INTERACT '97)*, pages 286–293, 1997.

Appendix A

SPSS Analysis from the Experiments

A.1 Experiment One

Cluster Task - Completion Time

Descriptives

Time

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean Lower Bound	95% Confidence Interval for Mean Upper Bound	Minimum	Maximum
Crystal Ball	64	1.02E4	5432.366	679.046	8872.39	11586.32	3448	22448
Multiscale Zoom	70	9049.20	5223.235	624.296	7803.76	10294.64	2652	25943
Window	69	8875.97	4317.947	519.820	7838.69	9913.26	2090	19001
WinHop	70	1.26E4	4793.201	572.897	11436.29	13722.08	5116	26177
Total	273	1.02E4	5146.617	311.487	9573.98	10800.44	2090	26177

ANOVA

Time

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	6.099E8	3	2.033E8	8.293	.000
Within Groups	6.595E9	269	2.452E7		
Total	7.205E9	272			

Experiment One - Completion time for the Cluster task.

Post Hoc Tests

Multiple Comparisons

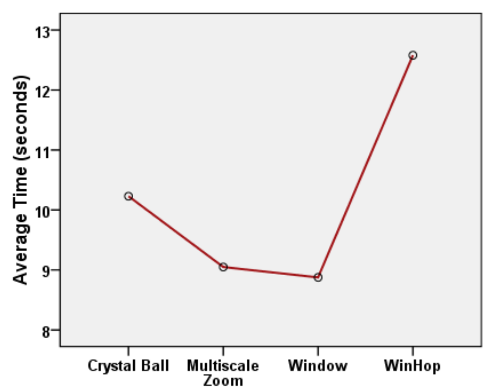
Time

Bonferroni

(I) Techniques	(J) Techniques	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	Upper Bound
Crystal Ball	Multiscale Zoom	1180.159	856.320	1.000	-1095.86	3456.18
	Window	1353.388	859.278	.699	-930.50	3637.27
	WinHop	-2349.826*	856.320	.039	-4625.85	-73.80
Multiscale Zoom	Crystal Ball	-1180.159	856.320	1.000	-3456.18	1095.86
	Window	173.229	839.956	1.000	-2059.30	2405.76
	WinHop	-3529.986*	836.929	.000	-5754.47	-1305.50
Window	Crystal Ball	-1353.388	859.278	.699	-3637.27	930.50
	Multiscale Zoom	-173.229	839.956	1.000	-2405.76	2059.30
	WinHop	-3703.215*	839.956	.000	-5935.74	-1470.69
WinHop	Crystal Ball	2349.826*	856.320	.039	73.80	4625.85
	Multiscale Zoom	3529.986*	836.929	.000	1305.50	5754.47
	Window	3703.215*	839.956	.000	1470.69	5935.74

*. The mean difference is significant at the 0.05 level.

Means Plots



Experiment One - Completion time for the Cluster task.

Cluster Task – Number of mis-clicks

Descriptives

Mistakes

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
Crystal Ball	64	.09	.750	.094	-.09	.28	0	6
Multiscale Zoom	70	.04	.204	.024	.00	.09	0	1
Window	69	.00	.000	.000	.00	.00	0	0
WinHop	70	.07	.393	.047	-.02	.17	0	3
Total	273	.05	.426	.026	.00	.10	0	6

ANOVA

Mistakes

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	.330	3	.110	.605	.612
Within Groups	48.952	269	.182		
Total	49.282	272			

Experiment One - Number of mis-clicks for the Cluster task.

Post Hoc Tests

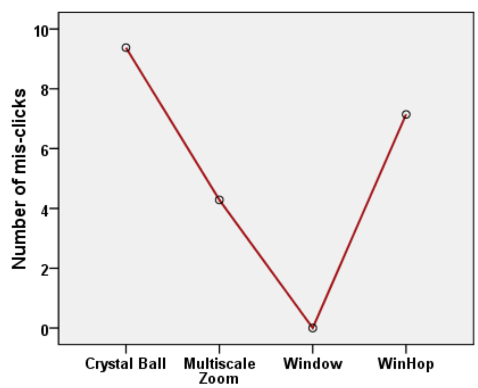
Multiple Comparisons

Mistakes

Bonferroni

(I)Techniques	(J)Techniques	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
Crystal Ball	Multiscale Zoom	.051	.074	1.000	-.15	.25
	Window	.094	.074	1.000	-.10	.29
	WinHop	.022	.074	1.000	-.17	.22
Multiscale Zoom	Crystal Ball	-.051	.074	1.000	-.25	.15
	Window	.043	.072	1.000	-.15	.24
	WinHop	-.029	.072	1.000	-.22	.16
Window	Crystal Ball	-.094	.074	1.000	-.29	.10
	Multiscale Zoom	-.043	.072	1.000	-.24	.15
	WinHop	-.071	.072	1.000	-.26	.12
WinHop	Crystal Ball	-.022	.074	1.000	-.22	.17
	Multiscale Zoom	.029	.072	1.000	-.16	.22
	Window	.071	.072	1.000	-.12	.26

Means Plots



Experiment One - Number of mis-clicks for the Cluster task.

Proximity Between Objects Task - Completion Time

Descriptives

Time

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean Lower Bound	95% Confidence Interval for Mean Upper Bound	Minimum	Maximum
Crystal Ball	68	9116.71	5086.351	616.811	7885.55	10347.87	2481	27940
Multiscale Zoom	69	8841.13	4545.325	547.193	7749.22	9933.04	2418	19609
Window	69	8746.91	5009.165	603.033	7543.58	9950.25	1779	28938
WinHop	68	14579.91	4582.567	555.718	13470.69	15689.13	5055	25740
Total	274	10310.02	5381.265	325.094	9670.01	10950.03	1779	28938

ANOVA

Time

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1.654E9	3	5.514E8	23.813	.000
Within Groups	6.251E9	270	2.315E7		
Total	7.906E9	273			

Experiment One - Completion time for the Proximity Between Objects task.

Post Hoc Tests

Multiple Comparisons

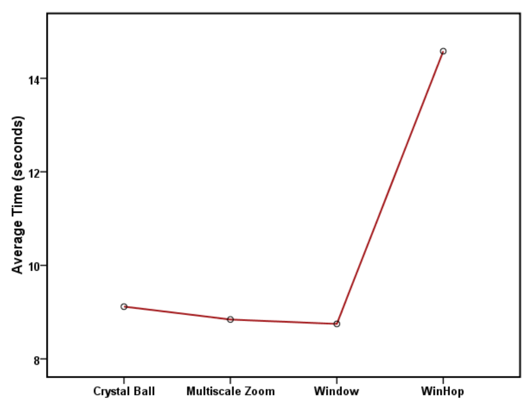
Time

Bonferroni

(I) Techniques	(J) Techniques	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	Upper Bound
Crystal Ball	Multiscale Zoom	275.575	822.224	1.000	-1909.76	2460.91
	Window	369.793	822.224	1.000	-1815.55	2555.13
	WinHop	-5463.206*	825.220	.000	-7656.51	-3269.91
Multiscale Zoom	Crystal Ball	-275.575	822.224	1.000	-2460.91	1909.76
	Window	94.217	819.218	1.000	-2083.13	2271.57
	WinHop	-5738.781*	822.224	.000	-7924.12	-3553.44
Window	Crystal Ball	-369.793	822.224	1.000	-2555.13	1815.55
	Multiscale Zoom	-94.217	819.218	1.000	-2271.57	2083.13
	WinHop	-5832.999*	822.224	.000	-8018.34	-3647.66
WinHop	Crystal Ball	5463.206*	825.220	.000	3269.91	7656.51
	Multiscale Zoom	5738.781*	822.224	.000	3553.44	7924.12
	Window	5832.999*	822.224	.000	3647.66	8018.34

*. The mean difference is significant at the 0.05 level.

Means Plots



Experiment One - Completion time for the Proximity Between Objects task.

Proximity Between Objects Task – Number of mis-clicks

Descriptives

Mistakes

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
Crystal Ball	68	.15	.605	.073	.00	.29	0	4
Multiscale Zoom	69	.16	.474	.057	.05	.27	0	2
Window	69	.25	.881	.106	.03	.46	0	6
WinHop	68	.06	.293	.036	-.01	.13	0	2
Total	274	.15	.604	.036	.08	.23	0	6

ANOVA

Mistakes

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1.210	3	.403	1.107	.347
Within Groups	98.352	270	.364		
Total	99.562	273			

Experiment One - Number of mis-clicks for the Proximity Between Objects task.

Post Hoc Tests

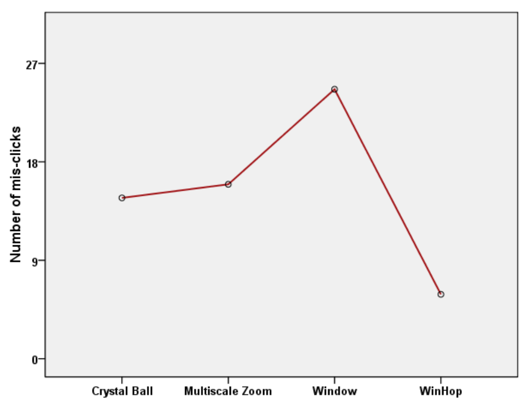
Multiple Comparisons

Mistakes

Bonferroni

(I) Techniques	(J)Techniques	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
Crystal Ball	Multiscale Zoom	-.012	.103	1.000	-.29	.26
	Window	-.099	.103	1.000	-.37	.17
	WinHop	.088	.104	1.000	-.19	.36
Multiscale Zoom	Crystal Ball	.012	.103	1.000	-.26	.29
	Window	-.087	.103	1.000	-.36	.19
	WinHop	.101	.103	1.000	-.17	.37
Window	Crystal Ball	.099	.103	1.000	-.17	.37
	Multiscale Zoom	.087	.103	1.000	-.19	.36
	WinHop	.188	.103	.420	-.09	.46
WinHop	Crystal Ball	-.088	.104	1.000	-.36	.19
	Multiscale Zoom	-.101	.103	1.000	-.37	.17
	Window	-.188	.103	.420	-.46	.09

Means Plots



Experiment One - Number of mis-clicks for the Proximity Between Objects task.

Proximity from Reference Task – Completion Time

Descriptives

Time

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean Lower Bound	95% Confidence Interval for Mean Upper Bound	Minimum	Maximum
Crystal Ball	67	20914.57	12289.514	1501.403	17916.92	23912.22	4414	55474
Multiscale Zoom	70	6771.07	4097.337	489.725	5794.10	7748.05	2230	21076
Window	70	7665.96	5757.014	688.095	6293.25	9038.67	1622	33400
WinHop	70	15200.06	3984.199	476.203	14250.06	16150.06	5227	28205
Total	277	12548.27	9265.702	556.722	11452.31	13644.24	1622	55474

ANOVA

Time

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	9.187E9	3	3.062E9	57.621	.000
Within Groups	1.451E10	273	5.315E7		
Total	2.370E10	276			

Experiment One - Completion time for the Proximity from Reference task.

Post Hoc Tests

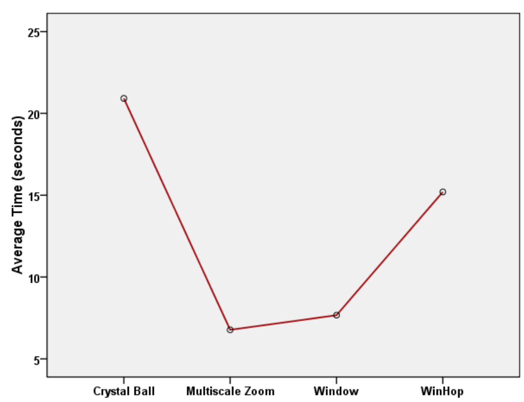
Multiple Comparisons

Time
Bonferroni

(I) Techniques	(J) Techniques	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Crystal Ball	Multiscale Zoom	14143.496*	1245.967	.000	10832.19	17454.80
	Window	13248.610*	1245.967	.000	9937.30	16559.92
	WinHop	5714.510*	1245.967	.000	2403.20	9025.82
Multiscale Zoom	Crystal Ball	-14143.496*	1245.967	.000	-17454.80	-10832.19
	Window	-894.886	1232.249	1.000	-4169.74	2379.97
	WinHop	-8428.986*	1232.249	.000	-11703.84	-5154.13
Window	Crystal Ball	-13248.610*	1245.967	.000	-16559.92	-9937.30
	Multiscale Zoom	894.886	1232.249	1.000	-2379.97	4169.74
	WinHop	-7534.100*	1232.249	.000	-10808.95	-4259.25
WinHop	Crystal Ball	-5714.510*	1245.967	.000	-9025.82	-2403.20
	Multiscale Zoom	8428.986*	1232.249	.000	5154.13	11703.84
	Window	7534.100*	1232.249	.000	4259.25	10808.95

*. The mean difference is significant at the 0.05 level.

Means Plots



Experiment One - Completion time for the Proximity from Reference task.

Proximity from Reference Task – Number of mis-clicks

Descriptives

Mistakes

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean Lower Bound	95% Confidence Interval for Mean Upper Bound	Minimum	Maximum
Crystal Ball	67	.40	.970	.119	.17	.64	0	6
Multiscale Zoom	70	.10	.347	.041	.02	.18	0	2
Window	70	.39	1.526	.182	.02	.75	0	12
WinHop	70	.00	.000	.000	.00	.00	0	0
Total	277	.22	.932	.056	.11	.33	0	12

ANOVA

Mistakes

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	8.562	3	2.854	3.373	.019
Within Groups	231.005	273	.846		
Total	239.567	276			

Experiment One - Number of mis-clicks for the Proximity from Reference task.

Post Hoc Tests

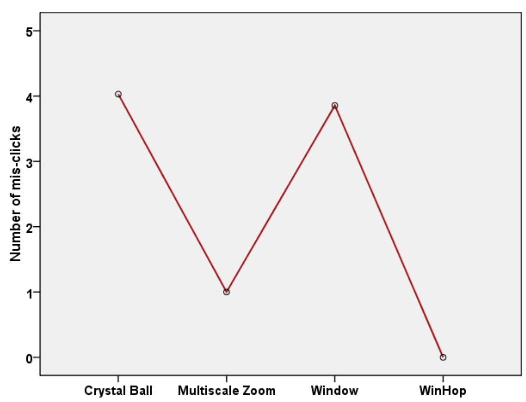
Multiple Comparisons

Mistakes
Tamhane

(I) Techniques	(J)Techniques	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Crystal Ball	Multiscale Zoom	.303	.126	.104	-.04	.64
	Window	.017	.217	1.000	-.56	.60
	WinHop	.403*	.119	.007	.08	.72
Multiscale Zoom	Crystal Ball	-.303	.126	.104	-.64	.04
	Window	-.286	.187	.568	-.79	.22
	WinHop	.100	.041	.106	-.01	.21
Window	Crystal Ball	-.017	.217	1.000	-.60	.56
	Multiscale Zoom	.286	.187	.568	-.22	.79
	WinHop	.386	.182	.207	-.11	.88
WinHop	Crystal Ball	-.403*	.119	.007	-.72	-.08
	Multiscale Zoom	-.100	.041	.106	-.21	.01
	Window	-.386	.182	.207	-.88	.11

*. The mean difference is significant at the 0.05 level.

Means Plots



Experiment One - Number of mis-clicks for the Proximity from Reference task.

Users' Feedback - Preference

Descriptives

Preference

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean Lower Bound	95% Confidence Interval for Mean Upper Bound	Minimum	Maximum
Crystal Ball	14	.00	.000	.000	.00	.00	0	0
Multiscale Zoom	14	.64	.497	.133	.36	.93	0	1
Window	14	.07	.267	.071	-.08	.23	0	1
WinHop	14	.29	.469	.125	.02	.56	0	1
Total	56	.25	.437	.058	.13	.37	0	1

ANOVA

Preference

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	3.500	3	1.167	8.667	.000
Within Groups	7.000	52	.135		
Total	10.500	55			

Post Hoc Tests

Multiple Comparisons

Preference
Bonferroni

(I) Techniques	(J) Techniques	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Crystal Ball	Multiscale Zoom	-.643*	.139	.000	-1.02	-.26
	Window	-.071	.139	1.000	-.45	.31
	WinHop	-.286	.139	.266	-.67	.09
Multiscale Zoom	Crystal Ball	.643*	.139	.000	.26	1.02
	Window	.571*	.139	.001	.19	.95
	WinHop	.357	.139	.077	-.02	.74
Window	Crystal Ball	.071	.139	1.000	-.31	.45
	Multiscale Zoom	-.571*	.139	.001	-.95	-.19
	WinHop	-.214	.139	.770	-.59	.17
WinHop	Crystal Ball	.286	.139	.266	-.09	.67
	Multiscale Zoom	-.357	.139	.077	-.74	.02
	Window	.214	.139	.770	-.17	.59

*. The mean difference is significant at the 0.05 level.

Experiment One - Level of users' preference.

Users' Feedback - Speed

Descriptives

Speed

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean Lower Bound	95% Confidence Interval for Mean Upper Bound	Minimum	Maximum
Crystal Ball	14	.07	.267	.071	-.08	.23	0	1
Multiscale Zoom	14	.57	.514	.137	.27	.87	0	1
Window	14	.29	.469	.125	.02	.56	0	1
WinHop	14	.07	.267	.071	-.08	.23	0	1
Total	56	.25	.437	.058	.13	.37	0	1

ANOVA

Speed

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	2.357	3	.786	5.018	.004
Within Groups	8.143	52	.157		
Total	10.500	55			

Post Hoc Tests

Multiple Comparisons

Speed
Bonferroni

(I) Techniques	(J) Techniques	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Crystal Ball	Multiscale Zoom	-.500*	.150	.009	-.91	-.09
	Window	-.214	.150	.948	-.62	.20
	WinHop	.000	.150	1.000	-.41	.41
Multiscale Zoom	Crystal Ball	.500*	.150	.009	.09	.91
	Window	.286	.150	.370	-.12	.70
	WinHop	.500*	.150	.009	.09	.91
Window	Crystal Ball	.214	.150	.948	-.20	.62
	Multiscale Zoom	-.286	.150	.370	-.70	.12
	WinHop	.214	.150	.948	-.20	.62
WinHop	Crystal Ball	.000	.150	1.000	-.41	.41
	Multiscale Zoom	-.500*	.150	.009	-.91	-.09
	Window	-.214	.150	.948	-.62	.20

*. The mean difference is significant at the 0.05 level.

Experiment One - Level of speed.

Users' Feedback - Comfort

Descriptives

Comfort

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean Lower Bound	95% Confidence Interval for Mean Upper Bound	Minimum	Maximum
Crystal Ball	14	.00	.000	.000	.00	.00	0	0
Multiscale Zoom	14	.71	.469	.125	.44	.98	0	1
Window	14	.14	.363	.097	-.07	.35	0	1
WinHop	14	.14	.363	.097	-.07	.35	0	1
Total	56	.25	.437	.058	.13	.37	0	1

ANOVA

Comfort

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	4.214	3	1.405	11.621	.000
Within Groups	6.286	52	.121		
Total	10.500	55			

Post Hoc Tests

Multiple Comparisons

Comfort
Bonferroni

(I) Techniques	(J) Techniques	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Crystal Ball	Multiscale Zoom	-.714*	.131	.000	-1.07	-.35
	Window	-.143	.131	1.000	-.50	.22
	WinHop	-.143	.131	1.000	-.50	.22
Multiscale Zoom	Crystal Ball	.714*	.131	.000	.35	1.07
	Window	.571*	.131	.000	.21	.93
	WinHop	.571*	.131	.000	.21	.93
Window	Crystal Ball	.143	.131	1.000	-.22	.50
	Multiscale Zoom	-.571*	.131	.000	-.93	-.21
	WinHop	.000	.131	1.000	-.36	.36
WinHop	Crystal Ball	.143	.131	1.000	-.22	.50
	Multiscale Zoom	-.571*	.131	.000	-.93	-.21
	Window	.000	.131	1.000	-.36	.36

*. The mean difference is significant at the 0.05 level.

Experiment One - Level of comfort.

A.2 Experiment Two

Cluster Task - Completion Time

Descriptives

Time

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean Lower Bound	95% Confidence Interval for Mean Upper Bound	Minimum	Maximum
Crystal Ball	69	7408.55	3868.117	465.667	6479.33	8337.77	1843	18625
Multiscale Zoom	70	8176.07	4357.646	520.838	7137.03	9215.12	2078	20703
Window	70	6894.86	3945.191	471.541	5954.16	7835.55	1562	19687
WinHop	69	12311.14	4422.031	532.350	11248.86	13373.43	4531	22594
Total	278	8689.29	4651.571	278.983	8140.10	9238.49	1562	22594

ANOVA

Time

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1.262E9	3	4.207E8	24.364	.000
Within Groups	4.731E9	274	1.727E7		
Total	5.993E9	277			

Experiment Two - Completion time for the Cluster task.

Post Hoc Tests

Multiple Comparisons

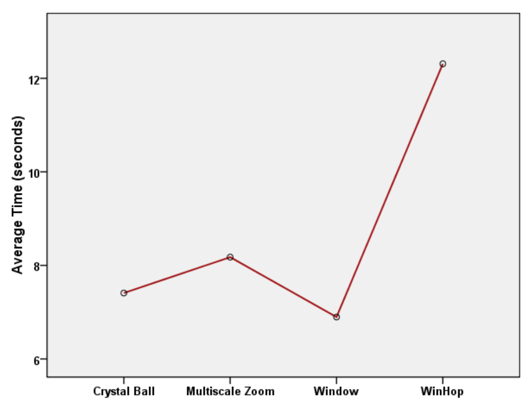
Time

Bonferroni

(I)Techniques	(J)Techniques	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Crystal Ball	Multiscale Zoom	-767.521	704.937	1.000	-2640.93	1105.89
	Window	513.694	704.937	1.000	-1359.71	2387.10
	WinHop	-4902.594*	707.468	.000	-6782.73	-3022.46
Multiscale Zoom	Crystal Ball	767.521	704.937	1.000	-1105.89	2640.93
	Window	1281.214	702.397	.415	-585.44	3147.87
	WinHop	-4135.073*	704.937	.000	-6008.48	-2261.67
Window	Crystal Ball	-513.694	704.937	1.000	-2387.10	1359.71
	Multiscale Zoom	-1281.214	702.397	.415	-3147.87	585.44
	WinHop	-5416.288*	704.937	.000	-7289.69	-3542.88
WinHop	Crystal Ball	4902.594*	707.468	.000	3022.46	6782.73
	Multiscale Zoom	4135.073*	704.937	.000	2261.67	6008.48
	Window	5416.288*	704.937	.000	3542.88	7289.69

*. The mean difference is significant at the 0.05 level.

Means Plots



Experiment Two - Completion time for the Cluster task.

Cluster Task – Number of mis-clicks

Descriptives

Mistakes

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
Crystal Ball	69	.17	.593	.071	.03	.32	0	3
Multiscale Zoom	70	.16	.500	.060	.04	.28	0	3
Window	70	.03	.168	.020	-.01	.07	0	1
WinHop	69	.04	.205	.025	.00	.09	0	1
Total	278	.10	.413	.025	.05	.15	0	3

ANOVA

Mistakes

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1.183	3	.394	2.349	.073
Within Groups	45.997	274	.168		
Total	47.180	277			

Experiment Two - Number of mis-clicks for the Cluster task.

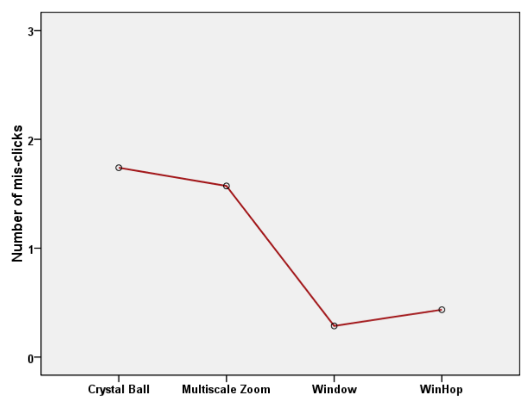
Post Hoc Tests

Multiple Comparisons

Mistakes
Bonferroni

(I) Techniques	(J) Techniques	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Crystal Ball	Multiscale Zoom	.017	.070	1.000	-.17	.20
	Window	.145	.070	.225	-.04	.33
	WinHop	.130	.070	.375	-.05	.32
Multiscale Zoom	Crystal Ball	-.017	.070	1.000	-.20	.17
	Window	.129	.069	.387	-.06	.31
	WinHop	.114	.070	.619	-.07	.30
Window	Crystal Ball	-.145	.070	.225	-.33	.04
	Multiscale Zoom	-.129	.069	.387	-.31	.06
	WinHop	-.015	.070	1.000	-.20	.17
WinHop	Crystal Ball	-.130	.070	.375	-.32	.05
	Multiscale Zoom	-.114	.070	.619	-.30	.07
	Window	.015	.070	1.000	-.17	.20

Means Plots



Experiment Two - Number of mis-clicks for the Cluster task.

Proximity Between Objects Task - Completion Time

Descriptives

Time

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
Crystal Ball	70	8848.47	3433.676	410.403	8029.74	9667.20	1938	19781
Multiscale Zoom	70	7669.84	4367.081	521.966	6628.55	8711.14	2093	20235
Window	69	6468.99	4178.894	503.080	5465.11	7472.87	1641	19297
WinHop	69	11367.57	4350.903	523.787	10322.36	12412.77	4203	23282
Total	278	8586.35	4460.368	267.515	8059.72	9112.97	1641	23282

ANOVA

Time

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	9.067E8	3	3.022E8	17.986	.000
Within Groups	4.604E9	274	1.680E7		
Total	5.511E9	277			

Experiment Two - Completion time for the Proximity Between Objects task.

Post Hoc Tests

Multiple Comparisons

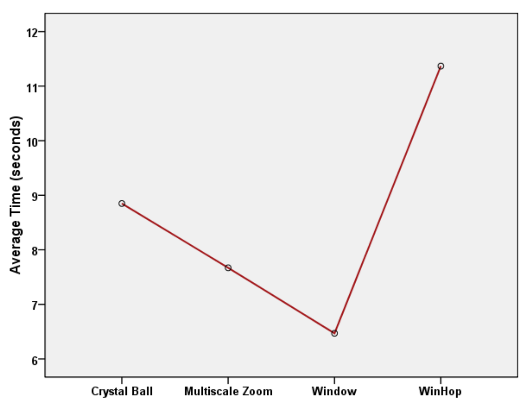
Time

Bonferroni

(I)Techniques	(J)Techniques	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
Crystal Ball	Multiscale Zoom	1178.629	692.896	.540	-662.78	3020.03
	Window	2379.486*	695.402	.004	531.42	4227.55
	WinHop	-2519.094*	695.402	.002	-4367.16	-671.03
Multiscale Zoom	Crystal Ball	-1178.629	692.896	.540	-3020.03	662.78
	Window	1200.857	695.402	.512	-647.21	3048.92
	WinHop	-3697.722*	695.402	.000	-5545.79	-1849.66
Window	Crystal Ball	-2379.486*	695.402	.004	-4227.55	-531.42
	Multiscale Zoom	-1200.857	695.402	.512	-3048.92	647.21
	WinHop	-4898.580*	697.899	.000	-6753.28	-3043.88
WinHop	Crystal Ball	2519.094*	695.402	.002	671.03	4367.16
	Multiscale Zoom	3697.722*	695.402	.000	1849.66	5545.79
	Window	4898.580*	697.899	.000	3043.88	6753.28

*. The mean difference is significant at the 0.05 level.

Means Plots



Experiment Two - Completion time for the Proximity Between Objects task.

Proximity Between Objects Task – Number of mis-clicks

Descriptives

Mistakes

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
Crystal Ball	70	.21	.508	.061	.09	.34	0	2
Multiscale Zoom	70	.17	.481	.057	.06	.29	0	2
Window	69	.09	.332	.040	.01	.17	0	2
WinHop	69	.10	.349	.042	.02	.19	0	2
Total	278	.14	.426	.026	.09	.19	0	2

ANOVA

Mistakes

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	.748	3	.249	1.380	.249
Within Groups	49.497	274	.181		
Total	50.245	277			

Experiment Two - Number of mis-clicks for the Proximity Between Objects task.

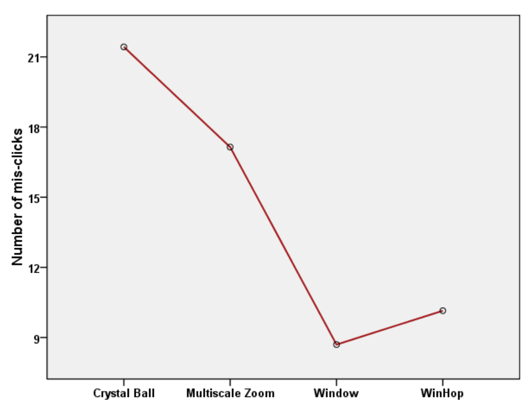
Post Hoc Tests

Multiple Comparisons

Mistakes
Bonferroni

(I)Techniques	(J) Techniques	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Crystal Ball	Multiscale Zoom	.043	.072	1.000	-.15	.23
	Window	.127	.072	.471	-.06	.32
	WinHop	.113	.072	.712	-.08	.30
Multiscale Zoom	Crystal Ball	-.043	.072	1.000	-.23	.15
	Window	.084	.072	1.000	-.11	.28
	WinHop	.070	.072	1.000	-.12	.26
Window	Crystal Ball	-.127	.072	.471	-.32	.06
	Multiscale Zoom	-.084	.072	1.000	-.28	.11
	WinHop	-.014	.072	1.000	-.21	.18
WinHop	Crystal Ball	-.113	.072	.712	-.30	.08
	Multiscale Zoom	-.070	.072	1.000	-.26	.12
	Window	.014	.072	1.000	-.18	.21

Means Plots



Experiment Two - Number of mis-clicks for the Proximity Between Objects task.

Proximity from Reference Task – Completion Time

Descriptives

Time

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean Lower Bound	95% Confidence Interval for Mean Upper Bound	Minimum	Maximum
Crystal Ball	70	9078.06	3217.935	384.617	8310.77	9845.35	3813	20329
Multiscale Zoom	70	4446.27	3790.742	453.080	3542.40	5350.14	1344	23813
Window	68	4826.47	3678.741	446.113	3936.03	5716.92	1312	20859
WinHop	65	13570.15	3572.835	443.156	12684.85	14455.46	5047	24937
Total	273	7900.96	5106.940	309.086	7292.46	8509.47	1312	24937

ANOVA

Time

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	3.664E9	3	1.221E9	95.800	.000
Within Groups	3.430E9	269	1.275E7		
Total	7.094E9	272			

Experiment Two - Completion time for the Proximity from Reference task.

Post Hoc Tests

Multiple Comparisons

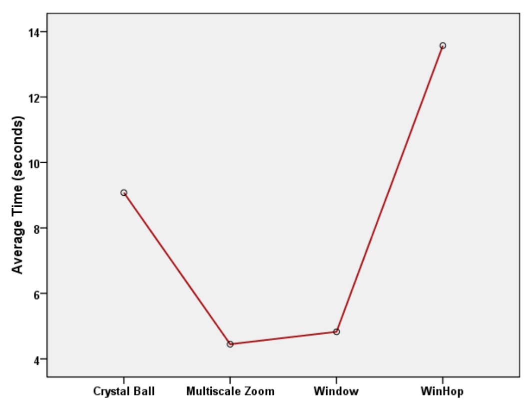
Time

Bonferroni

(I)Techniques	(J) Techniques	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
Crystal Ball	Multiscale Zoom	4631.786*	603.557	.000	3027.59	6235.99
	Window	4251.587*	607.978	.000	2635.63	5867.54
	WinHop	-4492.097*	615.054	.000	-6126.86	-2857.34
Multiscale Zoom	Crystal Ball	-4631.786*	603.557	.000	-6235.99	-3027.59
	Window	-380.199	607.978	1.000	-1996.15	1235.75
	WinHop	-9123.882*	615.054	.000	-10758.64	-7489.12
Window	Crystal Ball	-4251.587*	607.978	.000	-5867.54	-2635.63
	Multiscale Zoom	380.199	607.978	1.000	-1235.75	1996.15
	WinHop	-8743.683*	619.394	.000	-10389.98	-7097.39
WinHop	Crystal Ball	4492.097*	615.054	.000	2857.34	6126.86
	Multiscale Zoom	9123.882*	615.054	.000	7489.12	10758.64
	Window	8743.683*	619.394	.000	7097.39	10389.98

*. The mean difference is significant at the 0.05 level.

Means Plots



Experiment Two - Completion time for the Proximity from Reference task.

Proximity from Reference Task – Number of mis-clicks

Descriptives

Mistakes

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean Lower Bound	95% Confidence Interval for Mean Upper Bound	Minimum	Maximum
Crystal Ball	70	.14	.490	.059	.03	.26	0	3
Multiscale Zoom	70	.09	.329	.039	.01	.16	0	2
Window	68	.21	.612	.074	.06	.35	0	3
WinHop	65	.08	.322	.040	.00	.16	0	2
Total	273	.13	.456	.028	.07	.18	0	3

ANOVA

Mistakes

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	.723	3	.241	1.161	.325
Within Groups	55.790	269	.207		
Total	56.513	272			

Experiment Two - Number of mis-clicks for the Proximity from Reference task.

Post Hoc Tests

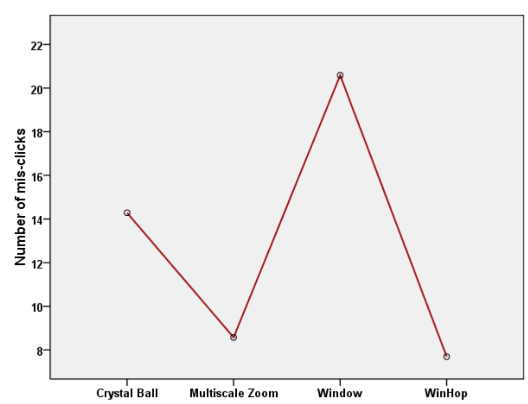
Multiple Comparisons

Mistakes

Bonferroni

(I) Techniques	(J) Techniques	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
Crystal Ball	Multiscale Zoom	.057	.077	1.000	-.15	.26
	Window	-.063	.078	1.000	-.27	.14
	WinHop	.066	.078	1.000	-.14	.27
Multiscale Zoom	Crystal Ball	-.057	.077	1.000	-.26	.15
	Window	-.120	.078	.734	-.33	.09
	WinHop	.009	.078	1.000	-.20	.22
Window	Crystal Ball	.063	.078	1.000	-.14	.27
	Multiscale Zoom	.120	.078	.734	-.09	.33
	WinHop	.129	.079	.623	-.08	.34
WinHop	Crystal Ball	-.066	.078	1.000	-.27	.14
	Multiscale Zoom	-.009	.078	1.000	-.22	.20
	Window	-.129	.079	.623	-.34	.08

Means Plots



Experiment Two - Number of mis-clicks for the Proximity from Reference task.

Users' Feedback - Preference

Descriptives

Preference

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean: Lower Bound	95% Confidence Interval for Mean: Upper Bound	Minimum	Maximum
Crystal Ball	14	2.29	1.267	.339	1.55	3.02	1	5
Multiscale Zoom	14	4.21	.699	.187	3.81	4.62	3	5
Window	14	3.93	.730	.195	3.51	4.35	2	5
WinHop	14	2.71	1.267	.339	1.98	3.45	1	5
Total	56	3.29	1.289	.172	2.94	3.63	1	5

ANOVA

Preference

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	36.429	3	12.143	11.481	.000
Within Groups	55.000	52	1.058		
Total	91.429	55			

Post Hoc Tests

Multiple Comparisons

Preference
Bonferroni

(I) Techniques	(J) Techniques	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval: Lower Bound	95% Confidence Interval: Upper Bound
Crystal Ball	Multiscale Zoom	-1.929*	.389	.000	-2.99	-.86
	Window	-1.643*	.389	.001	-2.71	-.58
	WinHop	-.429	.389	1.000	-1.49	.64
Multiscale Zoom	Crystal Ball	1.929*	.389	.000	.86	2.99
	Window	.286	.389	1.000	-.78	1.35
	WinHop	1.500*	.389	.002	.43	2.57
Window	Crystal Ball	1.643*	.389	.001	.58	2.71
	Multiscale Zoom	-.286	.389	1.000	-1.35	.78
	WinHop	1.214*	.389	.017	.15	2.28
WinHop	Crystal Ball	.429	.389	1.000	-.64	1.49
	Multiscale Zoom	-1.500*	.389	.002	-2.57	-.43
	Window	-1.214*	.389	.017	-2.28	-.15

*. The mean difference is significant at the 0.05 level.

Experiment Two - Level of users' preference.

Users' Feedback - Speed

Descriptives

Speed

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean Lower Bound	95% Confidence Interval for Mean Upper Bound	Minimum	Maximum
Crystal Ball	14	2.79	1.477	.395	1.93	3.64	1	5
Multiscale Zoom	14	4.07	.997	.267	3.50	4.65	2	5
Window	14	4.00	.679	.182	3.61	4.39	3	5
WinHop	14	2.57	1.089	.291	1.94	3.20	1	5
Total	56	3.36	1.271	.170	3.02	3.70	1	5

ANOVA

Speed

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	26.143	3	8.714	7.226	.000
Within Groups	62.714	52	1.206		
Total	88.857	55			

Post Hoc Tests

Multiple Comparisons

Speed
Bonferroni

(I) Techniques	(J) Techniques	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Crystal Ball	Multiscale Zoom	-1.286*	.415	.019	-2.42	-.15
	Window	-1.214*	.415	.031	-2.35	-.08
	WinHop	.214	.415	1.000	-.92	1.35
Multiscale Zoom	Crystal Ball	1.286*	.415	.019	.15	2.42
	Window	.071	.415	1.000	-1.07	1.21
	WinHop	1.500*	.415	.004	.36	2.64
Window	Crystal Ball	1.214*	.415	.031	.08	2.35
	Multiscale Zoom	-.071	.415	1.000	-1.21	1.07
	WinHop	1.429*	.415	.007	.29	2.57
WinHop	Crystal Ball	-.214	.415	1.000	-1.35	.92
	Multiscale Zoom	-1.500*	.415	.004	-2.64	-.36
	Window	-1.429*	.415	.007	-2.57	-.29

*. The mean difference is significant at the 0.05 level.

Experiment Two - Level of speed.

Users' Feedback - Comfort

Descriptives

Comfort

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
Crystal Ball	14	2.29	1.326	.354	1.52	3.05	1	5
Multiscale Zoom	14	4.43	.646	.173	4.06	4.80	3	5
Window	14	4.21	.893	.239	3.70	4.73	2	5
WinHop	14	2.64	1.151	.308	1.98	3.31	1	5
Total	56	3.39	1.384	.185	3.02	3.76	1	5

ANOVA

Comfort

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	49.500	3	16.500	15.361	.000
Within Groups	55.857	52	1.074		
Total	105.357	55			

Post Hoc Tests

Multiple Comparisons

Comfort
Bonferroni

(I) Techniques	(J) Techniques	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
Crystal Ball	Multiscale Zoom	-2.143*	.392	.000	-3.22	-1.07
	Window	-1.929*	.392	.000	-3.00	-.85
	WinHop	-.357	.392	1.000	-1.43	.72
Multiscale Zoom	Crystal Ball	2.143*	.392	.000	1.07	3.22
	Window	.214	.392	1.000	-.86	1.29
	WinHop	1.786*	.392	.000	.71	2.86
Window	Crystal Ball	1.929*	.392	.000	.85	3.00
	Multiscale Zoom	-.214	.392	1.000	-1.29	.86
	WinHop	1.571*	.392	.001	.50	2.65
WinHop	Crystal Ball	.357	.392	1.000	-.72	1.43
	Multiscale Zoom	-1.786*	.392	.000	-2.86	-.71
	Window	-1.571*	.392	.001	-2.65	-.50

*. The mean difference is significant at the 0.05 level.

Experiment Two - Level of comfort.

Appendix B

Material from the Experiments

Participant Information

Participant Number: ____________ Participant Name: ___________________________

Informed Consent Agreement

Please read this consent agreement carefully before you decide to participate in the study.

Purpose of the study: The purpose of this experiment is to test different map navigation techniques.

Time required: The experiment will last a maximum of 30 minutes.

What you will do in the study: You will perform three tasks using four navigation techniques.

Risks: There are no anticipated risks in this study.

Confidentiality: The information that you give in the study will be handled confidentially. It will be viewable only by researchers working on this project, which may involve faculty members and graduate research assistants.

Voluntary participation: Your participation in the study is completely voluntary.

Right to withdraw from the study: You have the right to withdraw from the study at any time without penalty.

How to withdraw from the study: If you want to withdraw from the study, please inform the experimenter and leave the room. There is no penalty for withdrawing. You will still receive full credit for the study. If you would like to withdraw after your materials have been submitted, please contact the researcher below:

Mahtab Nezhadasl
umnezhad@cs.umanitoba.ca

Agreement:
I agree to participate in the research study described above.

Signature: ___________________________ Date / Time: _______________________

Consent form for the experiments.

Map Navigation Survey (First Experiment)

Date / Time: ________________________

Participant Number: _____________ Participant Name: __________________________

1. Which technique did you like more?

MultiScale Zoom ☐ Window ☐ WinHop ☐ Crystal Ball ☐

2. Which technique did you find faster?

MultiScale Zoom ☐ Window ☐ WinHop ☐ Crystal Ball ☐

3. Which technique did you find easier?

MultiScale Zoom ☐ Window ☐ WinHop ☐ Crystal Ball ☐

4. Comments:

____

____

____

____

Experiment one's Questionnaire.

Map Navigation Survey (Second Experiment)

Date / Time: ________________________

Participant Number: _____________ Participant Name: __________________________

1. Please rank the tested techniques based on your preference?

	Strongly agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree
• MultiScale Zoom	☐	☐	☐	☐	☐
• Window	☐	☐	☐	☐	☐
• WinHop	☐	☐	☐	☐	☐
• Crystal Ball	☐	☐	☐	☐	☐

2. Please rank the tested techniques based on their speed?

	Strongly agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree
• MultiScale Zoom	☐	☐	☐	☐	☐
• Window	☐	☐	☐	☐	☐
• WinHop	☐	☐	☐	☐	☐
• Crystal Ball	☐	☐	☐	☐	☐

3. Please rank the tested techniques based on their ease?

	Strongly agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree
• MultiScale Zoom	☐	☐	☐	☐	☐
• Window	☐	☐	☐	☐	☐
• WinHop	☐	☐	☐	☐	☐
• Crystal Ball	☐	☐	☐	☐	☐

4. Comments:

____

____

____

____

Experiment Two's Questionnaire.