LINKING FOCUS AND CONTEXT IN 3D MULTISCALE ENVIRONMNETS

BY

MATTHEW D. PLUMLEE
B.S. in Computer Science, University of New Hampshire, 1995

DISSERTATION

Submitted to the University of New Hampshire
In Partial Fulfillment of
The Requirements for the Degree of

Doctor of Philosophy
In
Computer Science

May, 2004

This dissertation has been examined and approved.

Dissertation Director, Colin Ware,
Professor of Computer Science

R. Daniel Bergeron,
Professor of Computer Science

James L. Weiner,
Associate Professor of Computer Science

Ted M. Sparr,
Professor of Computer Science

Larry A. Mayer,
Professor of Earth Science and Ocean Engineering

4/27/2004
Date

ACKNOWLEDGEMENTS

Alone I am nothing; together, we know no limit.

The man I owe the greatest debt of gratitude toward completing this dissertation is my advisor, Colin Ware. Colin, you have taught me to harness my creativity, focus my effort, and be more articulate in scholarly writing. You were patient with my tandems and lack of focus, you have treated me with respect in a spirit of desire for mutual understanding, and you have given me every opportunity to grow and to make my own mistakes. Thank you for guiding and encouraging me through this process!

I also owe a debt of gratitude to the members of my dissertation committee: Dan Bergeron, Larry Mayer, Jim Weiner, and Ted Sparr. Thank you for taking the time to critique my methods and my writing, and for giving me guidance as to how to improve. Dan Bergeron, I especially want to thank you for helping Colin and myself navigate the dissertation process peculiar to UNH and the Computer Science department. Larry Mayer, thank you for the wonderful work environment you have established at CCOM that has made me feel very comfortable and has made available all the resources I have needed to learn and work efficiently.

I want to thank all the people that supported me through this process at various times. For their help in running experiments, I thank Jon Gilson and Hannah Sussman. For moral support, I thank Linda Spring-Andrews, Craig Smith, and all my friends at CCOM. For help through my particularly difficult final year, I especially wish to thank Arnold and Diane Gustafson, Duane Kellogg, Mike Leo, Roland Arsenault, Michelle Morton, and Roland Curit for their friendship and support. Finally, I would like to thank my wife Carolyn. Carolyn, thank you for supporting me financially through the long

process of my studies, always with good cheer and understanding, and for encouraging me to stick it out when I began to second-guess myself.

I also wish to thank NSF for funding me throughout my graduate studies.

Over the course of my graduate studies, I have left a trail of unfinished projects that have been picked up and carried by others. I wish first of all to thank Pilar de la Torre, my first advisor. Pilar, I greatly appreciate the time and effort you spent to mold me into an academic, and regret that I could not complete my studies under your advice. You taught me the value and beauty of thinking deeply and carefully about problems and ideas, which made palatable some of the long and difficult work required for this dissertation. I also wish to thank Bill Plant for picking up part of the project I started with Pilar, and bringing to it a sense of closure in his thesis. I wish to thank CJ Clark at Intellitech Corp. for providing the opportunity to hone my programming and design skills, and also to thank Justin Robison for picking up the reigns on my responsibilities when I left to finish my degree. Finally, I wish to thank Briana Sullivan for following through on the development of the exhibit for the Seacoast Science Center. Briana, without your wholehearted effort, either the exhibit would have been left to wither or my dissertation would have yet to be completed.

Finally, I wish to thank the person I owe the deepest debt of gratitude, our Creator. Father, I thank you for bringing my path across the paths of so many wonderful people. I thank you for gifting me and guiding me, and for walking with me throughout this process even when I did not recognize the sound of your footsteps. You have led me on a great adventure that has taught me the importance of relying on your Word, your Spirit, and on the people in whose midst you place me. Alone I am nothing; together, we know no limit.

TABLE OF CONTENTS

CHAPTER PAGE

APPENDICES

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

LINKING FOCUS AND CONTEXT IN 3D MULTISCALE ENVIRONMENTS

by

Matthew D. Plumlee

University of New Hampshire, May, 2004

The central question behind this dissertation is this: In what ways can 3D multiscale spatial information be presented in an interactive computer graphics environment, such that a human observer can better comprehend it? Toward answering this question, a two-pronged approach is employed that consists of practice within computer user-interface design, and theory grounded in perceptual psychology, bound together by an approach to the question in terms of focus and context as they apply to human attention. The major practical contribution of this dissertation is the development of a novel set of techniques for linking 3D windows to various kinds of reference frames in a virtual scene and to each other—linking one or more focal views with a view that provides context. Central to these techniques is the explicit recognition of the frames of reference inherent in objects, in computer-graphics viewpoint specifications, and in the human perception and cognitive understanding of space. Many of these techniques are incorporated into the GeoZui3D system as major extensions. An empirical evaluation of these techniques confirms the utility of 3D window proxy representations and orientation coupling. The major theoretical contribution is a cognitive systems model that predicts when linked focus and context views should be used over other techniques such as zooming. The predictive power of the model comes from explicit recognition of locations where a user will focus attention, as well as applied interpretations of the limitations of visual working memory. The model's ability to predict performance is empirically validated, while its ability to model user error is empirically founded. Both the model and the results of the related experiments suggest that multiple linked windows can be an effective way of presenting multiscale spatial information, especially in situations involving the comparison of three or more objects. The contributions of the dissertation are discussed in the context of the applications that have motivated them.

CHAPTER 1

INTRODUCTION

As technology progresses, new data is continually streaming in to scientists and analysts at ever-increasing resolutions and ever-accelerating speeds. The data comes from every direction: sensors make basic measurements such as surface temperature, rainfall amounts, or the shape of the ocean floor; this information is used in turn to generate higher-order statistics such as averages, trends, and confidence bounds. Such a growing deluge of information presents an increasing challenge for scientists in interpreting and understanding all the pertinent information necessary to make effective decisions.

This dissertation addresses that challenge with respect to improving the ability of the desktop display to represent pertinent, spatially oriented information. More specifically, this dissertation is built around one central question: How can 3D multiscale spatial information be presented so that people can better comprehend it? To clarify terms, *multiscale information* has relevant detail at both small and large scales, and *better comprehension* is realized through the ability of a person to make faster, more reliable decisions.

1.1 <u>Approach</u>

The key to addressing this central question is the realization that people do not comprehend a collage of information all at once. Human attention is partitioned into immediately relevant *focus* information and potentially relevant *context* information. From a purely perceptual standpoint, the big picture is stitched together by first focusing in on details, such as locating features and determining their color, and then *linking* (or relating) the details to each other, such as determining that the darker features seen close up align with dark features seen at a larger scale. The notions of focus and context are useful in refining the central question so that it can be stated it in terms of a problem with quantifiable goals and constraints.

The *focus-in-context problem* is a perceptual optimization problem: with regard to the display and to attentional resources, what balance of focus and context information should be presented to provide the best user comprehension? Stated another way, the goal is to maximize user performance (speed, accuracy) on a given task, under the constraints of limited computer display space, extremely limited attentional capacity, and a limited ability to interact with the computer display. When multiple tasks are to be performed simultaneously, the goal for each subtask must also include minimization of display and attentional requirements without sacrificing task performance.

Solutions to the focus-in-context problem come in the form of specific interface techniques and more general systems of interaction. Such solutions generally organize a "large" number of information items in a reduced form, and concentrate display resources on only a "few" important focus items. The balance is struck to retain enough display and attentional resources to fulfill two purposes:

1. To allow the user to properly interpret the focus item(s) and discern how the focus item(s) relate to other information items.
2. To allow non-focus information items to rapidly become part of the focus.

The tradeoffs are apparent in the nature of the resources in question: a person can only attend to a few items at a given time, but can quickly attend to new information by moving the eye to a new focus; a desktop display is constrained by screen size and resolution, but can be dynamically updated as a user requests new perspectives on the information. Leverage comes in the dynamic allocation of resources throughout the course of an information-intensive task.

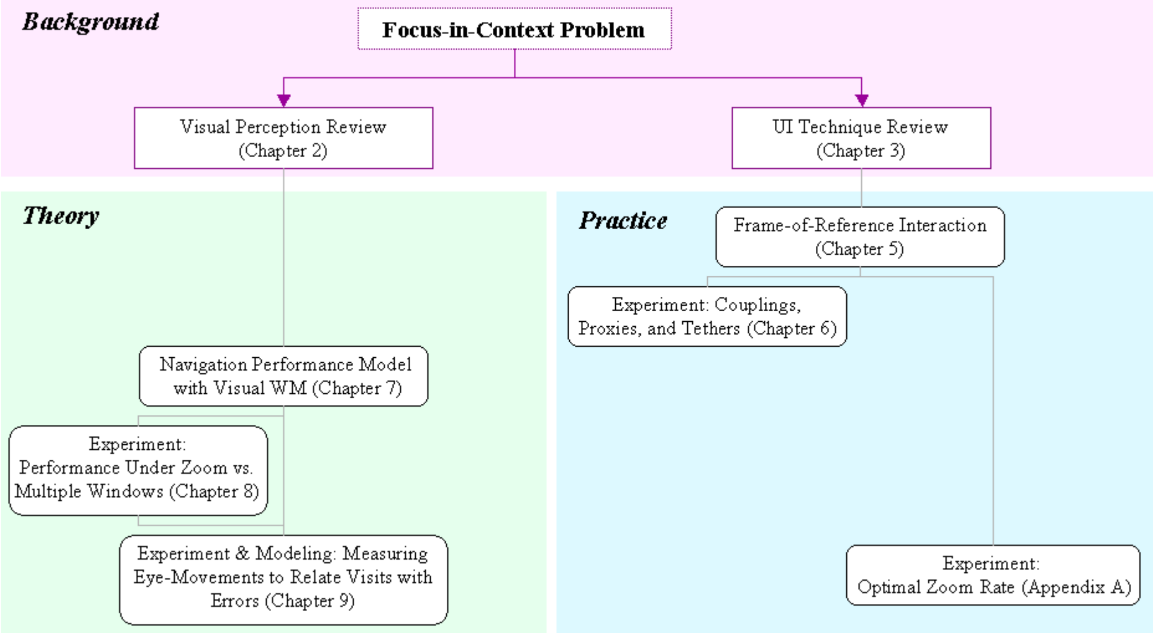

Background
Focus-in-Context Problem
Visual Perception Review (Chapter 2)
UI Technique Review (Chapter 3)
Theory
Practice
Frame-of-Reference Interaction (Chapter 5)
Experiment: Couplings, Proxies, and Tethers (Chapter 6)
Navigation Performance Model with Visual WM (Chapter 7)
Experiment: Performance Under Zoom vs. Multiple Windows (Chapter 8)
Experiment & Modeling: Measuring Eye-Movements to Relate Visits with Errors (Chapter 9)
Experiment: Optimal Zoom Rate (Appendix A)

Figure 1.1: Outline of the two-pronged dissertation approach.

In this dissertation the focus-in-context problem is addressed for the specific case of 3D, multiscale, spatial information. A two-pronged approach is employed, as illustrated in Figure 1.1. One prong (*Theory*) is based on work in perceptual psychology. Prior work in this field illuminates the abilities and limitations of the human visual system, providing as well the visual cues to which the visual system responds. This is applied to create a model of user performance that can be used to decide when one class of navigation interface should be used over another. The other prong (*Practice*) consists of work in computer user-interface design. Prior work in this area provides a large collection of techniques that address the focus-in-context problem and have the potential to improve user efficiency and reliability. These techniques are improved in this dissertation, and new techniques are developed and described. In addition, guidelines are provided as to how and when these techniques should be used. The handle uniting the prongs of theory and practice is the focus-in-context problem. It provides the perspective from which to look at existing literature, as well as the skeletal structure on which the contributions of the dissertation are built.

The concepts of the dissertation are made concrete by applying them to multiscale 3D spatial data related to underwater mapping. The operations on this data that are to be enhanced include mission planning, monitoring, exploration and interpretation of data, and presentation. The practical contributions of the dissertation are made concrete through their inclusion in an interactive 3D display system called GeoZui3D. It is also through GeoZui3D that the techniques are developed and tested for linking the focus and context information inherent in the underwater mapping data.

1.2 Organization

The dissertation proceeds according to the two-pronged strategy illustrated in Figure 1.1. The first two chapters review contributions from related work in the fields of perceptual psychology and human computer interaction. The review of visual perception in Chapter 2 identifies relevant capabilities and limitations of the human visual system, as well as relevant empirical evidence for how these capabilities and limitations impact operations that humans perform in the world. The review of user interface techniques in Chapter 3 describes the compatibility of the human visual system with existing human-computer interaction methodologies, demonstrating some “dos and don’ts” for displaying and providing interaction with various kinds of data. Chapter 4 spells out some motivating applications and states the strategy used in this dissertation to extend the knowledge presented in the review.

Chapters 5 and 6 and Appendix A pertain to the practical side of the two-pronged approach. Chapter 5 describes a new family of 3D focus-in-context interaction techniques involving zooming and multiple windows, as well as a coherent software framework with which the techniques are implemented. The techniques and software framework are built around the use of geometric reference frames and certain relationships between them. Chapter 6 presents experiments that contrast the relative utility of three devices for linking focus and context information from two 3D views. Appendix A presents an auxiliary experiment that provides guidelines for how zoom rate should be regulated.

Chapters 7-9 provide theoretical underpinnings to the practical work in the form of a cognitive systems model. Chapter 7 presents a visit-based model for comparing navigation mechanisms with respect to user performance, and describes how the model can be applied to contrast the utility of zooming and multi-windowed interfaces for a specific task. Chapter 8 describes an experiment that tests the predictions of the model, with results that are generally supportive, but that identify a shortcoming with respect to handling errors. Chapter 9 describes an experiment that provides insight into the source of user errors, and then proposes a model to account for error based on the empirical evidence and a reinterpretation of prior work in the field of cognitive psychology.

Finally, Chapter 10 summarizes the contributions and concludes with an executive summary in the form of an answer to the dissertation's central question.

CHAPTER 2

RELEVANT CHARACTERISTICS OF VISUAL PERCEPTION

The single most relevant property of the human visual system with regard to the focus-in-context problem is that of attention. Attention determines the current focus of cognition—what the mind can manipulate in making a decision. When a computer display is used in decision-making, the focus of attention is usually related to what the eyes are fixated on. Context is comprised of elements previously perceived or non-fixated elements on a display that can either aid in appropriately interpreting the focus or draw attention to a new focus.

This section briefly explores three aspects of the human visual system in terms of attentional focus and context, each of which is relevant to how information on a computer display is perceived and used. The first aspect is feature and object perception within the *visual field*, which defines what properties of visual objects stand out and become available for attention, what can distract or lead attention, and how context can be associated with a focus item. This aspect provides guidance as to how objects should be rendered on the display. The second aspect is *visual working memory* (visual WM), which provides the storage necessary for visual cognition, but can only hold a few objects. This aspect provides guidance as to how many objects should be given prime display space at a time. The third aspect is *navigation and wayfinding*, which can be regarded as seeking information in order to bring it into the focus of attention. This last aspect provides guidance for the design of interaction techniques that allow users to navigate through an information space.

2.1 The Visual Field

At the earliest stage of vision, we encounter the idea of focus in terms of raw visual resolution. While light receptors cover a visual field about 160º wide by 135º high [Card et al. 1999], there is a small, less than 2° high-resolution area called the *fovea* where the highest concentration of cones is found. Within the fovea, an even tighter ½º field has the absolute highest resolution [Ware 2000]. Figure 2.1 illustrates how visual acuity decreases with distance to the fovea. A pattern that is recognized at one size near the fovea must cover an increasing amount of visual field to be as recognizable further out, varying roughly with the square of the distance to the fovea. Muscles move the eye so that the fovea covers areas on which one wishes to focus attention while the rest of the field of view contains visual context. During a visual search, fixations on focus items generally average around 300 milliseconds, while saccades (rapid linear eye movement) take only about 30 milliseconds [Palmer 1999].

Receptors on the retina detect the presence and intensity of light coming from a particular direction, but high-level decision-making requires the detection of features and objects. Signals from the receptors are processed in a variety of ways to detect such high-level items. Pre-attentive processing alerts the perceptual system to the existence of certain features as candidates for focus. Processing characterized by the Gestalt laws assembles features into visual objects and groupings of objects. Depth cues are recognized by the visual system to determine 3D structure of objects and their relative positions. Finally, memory and recognition act to identify potential focal items and to provide context with regard to the situations in which an item has been seen before. Relevant aspects of each form of visual processing are described below.

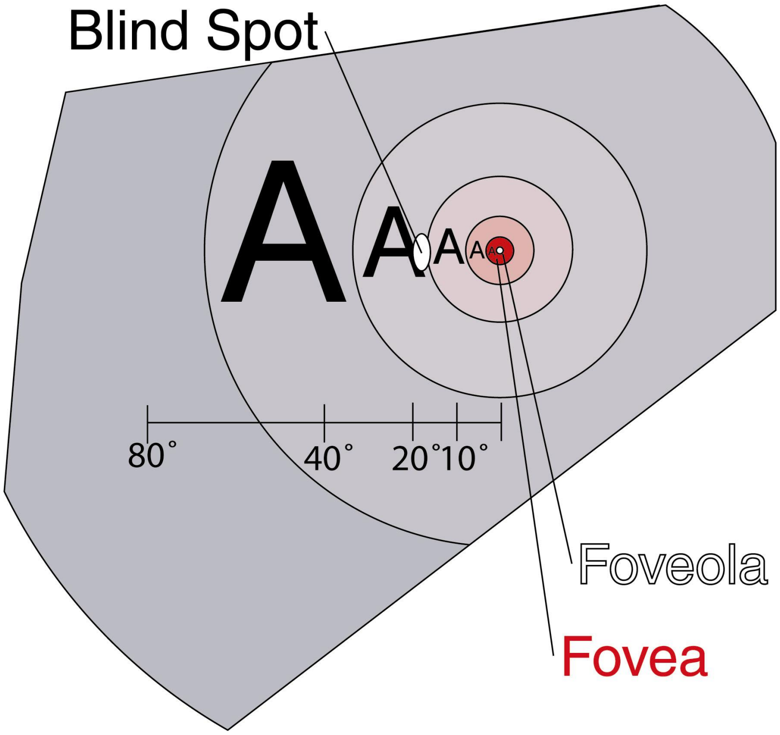


Figure 2.1: Distribution of light receptors in the eye.

2.1.1 Pre-Attentive Processing

Pre-attentive processing causes certain kinds of items to stand out from their surroundings. This is useful to display design because it indicates how items can be purposefully highlighted, but it also indicates what might distract a user from the task at hand. Items identified by pre-attentive processing are more readily accessed by focal attention than other items. Pre-attentive processing distinguishes simple features such as form (line orientation, curvature, grouping, etc.), color, texture, motion, and position [Triesman and Gormican 1988; Ware 2000]. Such processing makes it easy for one to separate different kinds of visual items, for instance discovering a circle in a field of straight lines (Figure 2.2), or a few red squares in a field of blue squares. These pre-attentive features cannot generally be combined, making it difficult to rapidly find a light circle among light squares and dark circles, for instance. The exceptions that do exist [Ware 2000] seem mostly to involve conjunctions of spatial location or motion.

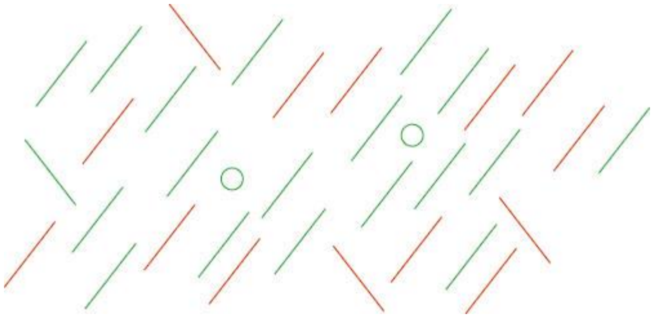


Figure 2.2: Example of pre-attentively processed shape and orientation information.

There is one technique that has been most commonly used by researchers to identify what is pre-attentively processed. This technique involves measuring subjects' response times in locating target objects in a field of distractor objects. The distractor and target objects generally resemble each other in every feature except for the feature or features being tested. For example, consider this task: determine whether or not any circles appear in a field of lines on a white background. If the time it takes a subject to locate a target object remains nearly constant regardless of the number of lines (distractors) present, then the feature that separates the target from the distractors is considered to be pre-attentively processed. For example, the time to identify that there are circles (targets) in Figure 2.2 does not depend on the number of lines (distractors) that are present.

2.1.2 Detection of Objects and Groups

In order to link items of focus with their context, we must know what mechanisms the visual system uses to detect objects, as well as to infer relationships and groupings of objects. Many of these mechanisms are summed up by the Gestalt laws and related perception principles [Palmer 1999, Card et al. 1999]: continuity, connectedness, symmetry, closure, figure vs. ground, and familiarity contribute to the detection of an object; proximity, similarity, familiarity, and similar motion (*common fate*) contribute to the grouping of objects. Relative size contributes to seeing smaller components of an image as objects, while the Gestalt law called *pragnanz* suggests a tendency to perceive as simple a structure as possible.

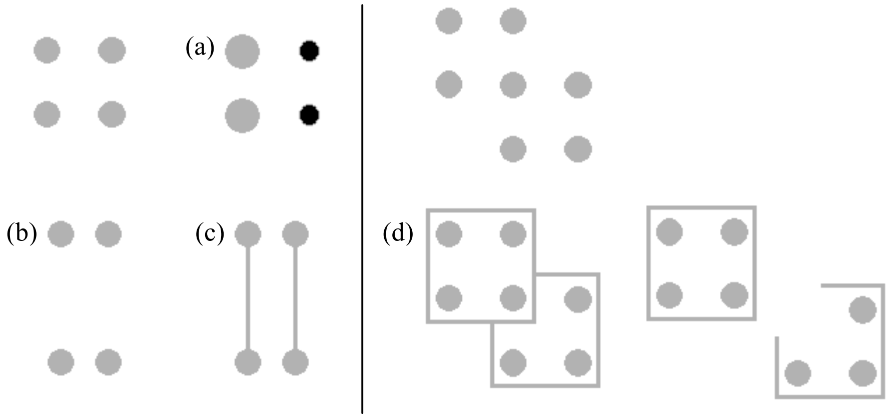


Figure 2.3: An illustration of some of the gestalt principles applied to grouping objects. Objects tend to be grouped according to (a) similarity, (b) proximity, (c) connectedness, and (d) closure.

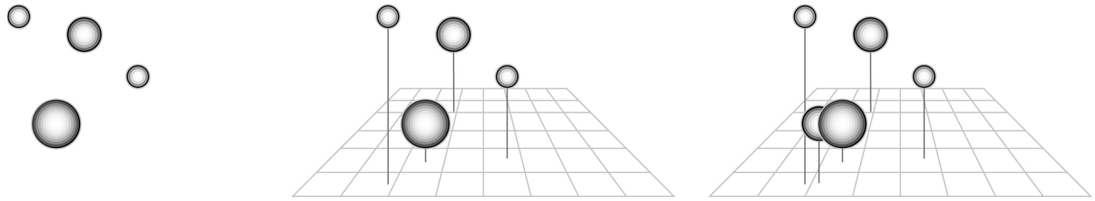
Figure 2.3 illustrates (a) similarity, (b) proximity, (c) connectedness, and (d) closure. Out of these principles, the use of windows and linkage devices in this dissertation make the most direct use of closure and connectedness. *Connectedness* is the tendency to see two visual items connected by a continuous contour as an object or grouping. *Closure* is the tendency to see closed contours as objects, and to close contours that are simple to close. The lower portion of Figure 2.3(d) demonstrates this in the fact that we tend to see two closed boxes surrounding the points, rather than one closed box and one open box. Closure is a strong segmentation cue that helps establish distinct frames of reference in the visual field [Ware 2000].

2.1.3 Depth Cues

Depth cues provide spatial context both in terms of the relative positions of focal objects, and in terms of the proximity of the user to a focal object. The three basic depth cues of most relevance to this dissertation are as follows [Palmer 1999, Wickens and Hollands 2000]:

- Occlusion (or interposition): When one object overlaps another in our field of view in the physical world, it is seen as being closer. This is arguably the strongest depth cue, although it only provides ordinal depth—Object A is in front of object B, is in front of object C. This depth cue relies on the identification of objects from the earlier stages of vision.
- Linear perspective: This is the way normal 3D images project through a pinhole or the eye onto a surface such as a board or the retina of the eye. As distance increases from the observer, object projections become smaller, textures are compressed, and parallel lines converge toward a single point.
- Stereoscopic vision (or binocular disparity): This is the information that comes from combining the visual information from both eyes. Objects or image features close to an observer appear in different positions relative to each other in the images projected onto the retina. As distance increases from the observer, these relative differences decrease, making depth harder to perceive based on stereo alone.

The human visual system also responds to “artificial” cues, such as lines dropped to a ground plane in a figure, as shown in Figure 2.4(b & c), or proximity luminance covariance (fog) [Dosher et al. 1986]. Some cues have precedence over others. For instance, in the absence of dropped lines in Figure 2.4(a), the effects of the relative 2D sizes of the balls causes one to see the "smaller" balls as being the same size as the others, but further back. The addition of the ground plane and dropped lines causes the brain to give a much different interpretation of the sizes and locations of the balls. On the other hand, occlusion overrides dropped lines, as can be seen in Figure 2.4(c) where occlusion puts the new ball behind an old one, even though the dropped line suggests otherwise.



(a) (b) (c)

Figure 2.4: An illustration of the depth cues and precedence: relative size vs. dropped lines vs. occlusion.

2.1.4 Memory and Recognition

Perception is a phenomenon in which the same image entering the eye may result in different interpretations of the image at different times. What we perceive can be affected by what has been perceived before and by what we expect to see. One example of this is the ability of the visual system to be *primed* for later recognition of an object, pattern, or visual behavior. Even a short exposure of an image can be enough to enable someone to recognize that image later, or to identify it more quickly. Another example is the effect that context has on the interpretation of an object: objects are more quickly and accurately recognized in the presence of other objects normally seen with it than they are among objects from another context [Palmer 1999].

Recognition of often-viewed objects is often created through repeated exposure to them from different orientations. Such repeated exposure can allow a person to build up canonical views of an object, according to a theory put forward by Palmer, Rosh, and Chase [1981] and further work by Edelman and Buelthoff [Edelman and Buelthoff 1992; Edelman 1995]. These views allow the recognition of an object even if it is slightly distorted, for instance by perspective, translation, scaling, or rotation. However, too much distortion away from a canonical view foils recognition. This is especially true of rotation, for instance when the face of an acquaintance is displayed upside down [Rhodes 1995]. The limitations of recognition under rotation play an important negative role when considering ways of linking differently oriented 3D views in a virtual environment.

2.2 Visual Working Memory

Once an object is identified as being of interest, visual working memory (visual WM) acts as the substrate that allows such objects to be compared, contrasted, and otherwise processed [Miyake and Shah 1999]. Several models of visual WM treat it in a way that might best be described as a sandbox with a few pails. You can put a number of visual features in each pail, such as shape and color, and you can arrange the pails however you like in the sandbox. Features are placed into a pail by actively attending to a visual item, and the features in a pail can be manipulated with respect to longer-term memory and whatever is currently in the visual field. However, there are only a few pails—if all the pails are full and you want to bring in another visual item, at least one of the pails will get emptied first.

Numerous models of working memory exist, each with its own set of components and constructs for its own particular focus and purpose (for an introduction to leading models, see Miyake and Shah [1999]). The models share a great deal in common, however, including the separation of working memory into components of limited capacity. As a basis for discussion of the relevant concepts, consider the *multiple-component* model of Baddeley and Hitch [1974], recently updated by Baddeley and Logie [Miyake and Shah 1999]. The white area in Figure 2.5 depicts the core components of the multiple-component model of visual working memory, while the gray areas indicate how visual working memory might interface with sensory input and long-term memory. Selected sensory input, in the form of sounds and images, enters working memory where it can reside in the phonological loop (referred to as verbal WM) or the visuo-spatial sketchpad (visual WM). The central executive acts as the regulator for information flow between these components and long-term memory. For visual tasks, focus can be said to reside in visual WM, with context coming from direct visual input and long-term memory.

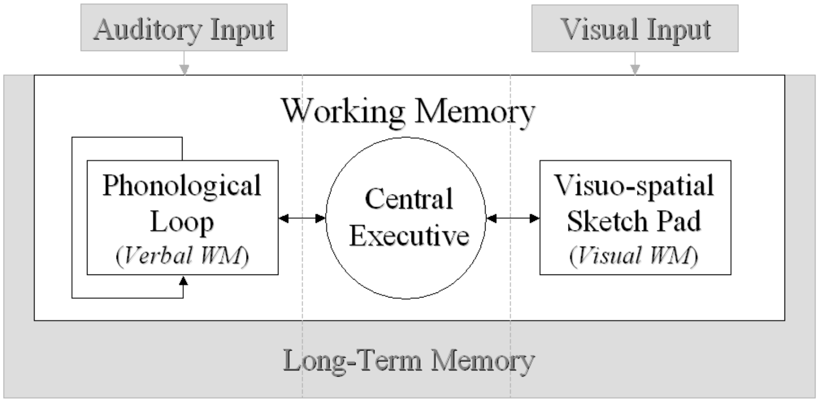


Figure 2.5: The multiple-component model of working memory from [Baddeley and Hitch 1974; Miyake and Shah 1999] (central white area) in relation to sensory input and long-term memory (peripheral gray zones).

Before the multiple-component model was presented, the limit on working memory was considered to be 7 items, plus or minus 2, as put forth by Miller [1956]. Since then, it has become apparent that verbal WM and visual WM are separate, and each has its own limit [Miyake and Shah 1999]. More recent research suggests that Miller's number is more closely related to limits only on verbal WM, and that this limit is based more on the phonological length of items than on the number of items themselves [Baddeley et al. 1975].

For a limit on the number of items that can be held in visual WM, consider recent work using sequential comparison tasks. Vogel et al. [2001] structure the task in the following way, as illustrated in Figure 2.6. First, a sample set of visual objects is displayed to a subject very briefly. A blank field is then displayed for roughly a second, which is long enough to ensure that visual WM would be tested, rather than the shorter-term but higher-capacity iconic memory. Then a probe set is displayed (either the sample set again or the sample set with one object changed in some way), and the subject is asked whether or not the probe set matches the sample set.

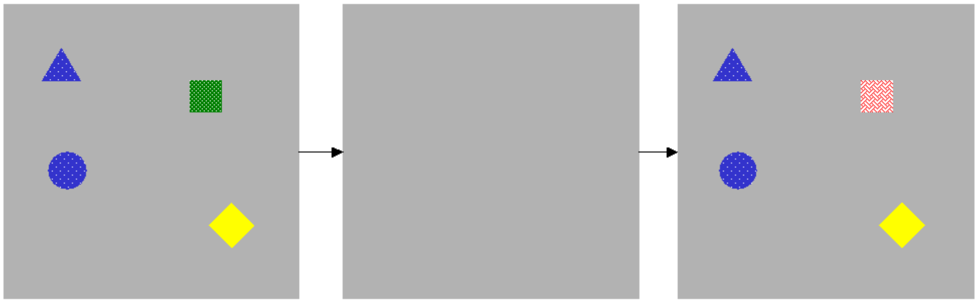


Figure 2.6: Sequence of displays for the sequential comparison task

Vogel's experiments [Vogel et al. 2001] use this sequential comparison design to contrast responses across differing set sizes. The large drop-off in accuracy between sets of sizes 3 and 4, evident in the results depicted in Figure 2.7, suggests that the capacity of visual WM in humans is limited to 3-4 objects at a time. This limit is confirmed by more rigorous analyses of the results that treat the task as a signal detection problem, properly treating false alarms using a formula reported by Pashler [1988]. It should be noted that this formula is based on an assumption that whole objects are stored—no partial information is remembered about any items. Vogel's results also show that decay of visual WM is negligible, at least for blanking periods lasting on the order of about 5 seconds (found by varying the length of time the blank field remained). These experiments, along with those of Jiang et al. [2000], show that objects held in visual WM can have a number of attributes, including color, line orientation, and shape.

Experiments by Jiang et al. [2000] also suggest that our memory of objects is strongly tied to the objects' spatial configuration—their positions relative to one another. In other words, if the objects are laid out differently between the sample set and the probe set, accuracy decreases in determining whether an object has changed or not. Interestingly, it appears that configuration memory may have a higher capacity than the focal object memory heretofore referred to as visual WM (see [Simons 1996] and [Jiang et al. 2000]). There is also some evidence that the memory of the configuration of objects is highly dependent upon the orientation(s) in which the configuration has been viewed [Roskos-Ewoldsen et al. 1998]. Accuracy declines when people are asked to make judgments on information based on a different perspective from what they might have encountered. Configuration memory potentially provides contextual linkage to the visual field—a context that can relate the few objects in visual WM to other related objects that cannot fit, but are easily accessible.

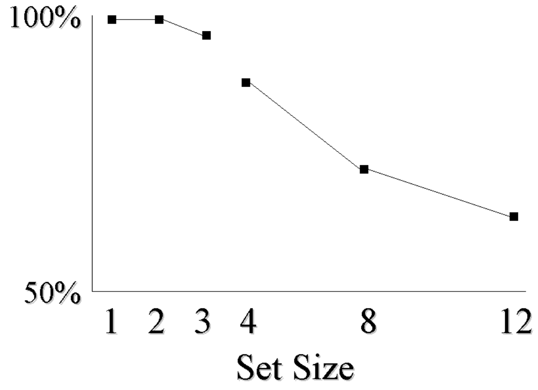


Figure 2.7: Depiction of accuracy results from sequential comparison task experiments performed by Vogel et al. [2001].

To summarize, approximately three objects can be held in visual WM at a time; each object can have several attributes including color, texture, and orientation; the memory of these objects is strongly tied to their spatial configuration; and there is negligible decay for time scales on the order of several seconds. These properties of visual WM are central to the modeling and experiments presented in Chapters 7 through 9 of this dissertation.

2.3 Navigation and Wayfinding

Most decision-making tasks require some searching for information that is not immediately available. Navigation and wayfinding are the processes of information seeking that bring desired information to the focus of attention. While wayfinding and navigation are often used interchangeably, wayfinding generally emphasizes the accumulation of spatial knowledge in an unfamiliar or partially familiar environment, and navigation emphasizes the actual traversal of a space. To illustrate this distinction, consider the task of grocery shopping. This task requires the shopper to search from place to place in a store for specific groceries to put in a cart, and then find a short checkout line to purchase the cartload of groceries. *Wayfinding* describes the process of performing this task when the performer has not been to the particular store before.

Meanwhile, *navigation* describes the process if the store is already familiar. Often the shopping task requires not only finding an item, but also attending to it to make sure it is acceptable. In this light, navigation and wayfinding can be seen as a way of finding an appropriate focus within any spatial context larger than the immediate visual field. This section describes some of the relevant literature on wayfinding, navigation, and other issues related to large spatial contexts.

2.3.1 Building Up Spatial Context

Seigel and White [1975] describe the process of coming to understand a particular environment as involving three stages of knowledge, gathered in the following order: landmark knowledge, route knowledge, and survey knowledge. Landmark knowledge is sometimes referred to as *declarative knowledge*, and is the recognition of distinguished features in a scene from the observer's point of view, such as a tall building or a statue. Route knowledge is often called *procedural knowledge*, and it consists of observer-centered instructions for getting from one place to another, often using landmarks as reference points. Survey knowledge is a mental map of an area, often created over time by navigating between different destinations in the environment.

External aids can provide wayfinding knowledge out of order, such as a postcard of a monument (landmark knowledge), directions from a friend (route knowledge), or a map (survey knowledge). The most important of such aids are maps. Thorndyke and Hayes-Roth [1982] performed experiments to compare the utility of maps and wayfinding (route) experience in a number of situations. Their findings show that 20 minutes of studying a map can equal one year of wayfinding experience for determinations of Euclidean distance and relative object location. Conversely, knowledge gained through wayfinding is better for determinations of orientation (the angle from one object to another).

Without external aids, wayfinding knowledge is prone to error. This is especially true of survey knowledge, which tends to succumb to rectilinear normalization: the straightening of curved paths, the orthogonalization of awkward corners, and the alignment of paths and corners to north-east-west-south grid lines [Wickens and Hollands 2000, Milgram and Jodelet 1976, Chase and Chi 1979].

2.3.2 Using Spatial Context for Orientation

When traversing space (versus gaining knowledge about it), the human visual system responds to a number of orienting cues that help provide spatial context. During forward navigation, the images of objects on the retina move outward from the center region to the periphery. This movement helps to provide a sense of one's motion through space. Perception of self-motion is strengthened both by a larger moving visual field, and the perception of static foreground objects or frames to contrast with a moving background [Howard and Heckman 1989, Howard and Childerson 1994]. The existence of a ground plane, and the sense of an "up" direction (perpendicular to the ground plane) both help to provide a sense of orientation. Even in the absence of a ground plane, an "up" direction can be implied by the orientation of familiar objects that have perceived "up" orientations (such as a table or signpost) [Howard and Childerson 1994].

When spatial context is being provided by external aids such as maps, the relationship between such context and surroundings are often made best when the aids and the world are aligned. For example, when navigating through the world, we often turn our map to match the direction we are facing at the moment. If we keep our map static as we navigate, we risk errors in decoding the orientation of features on the map with respect to current position; if we keep our forward direction as forward or up on the map, left on the map always matches left in the world.

A display that mimics such constant re-orienting of a map is called a track-up or forward-up display. Levine et al. [1984] were the first to demonstrate experimentally the importance of map alignment to successful task completion. Some later studies [Aretz 1991; Eley 1988] suggest that track-up displays are less confusing for novice users, but experts prefer the north-up display as it matches the perspective of a remembered canonical view. Other studies [Darken and Cevik 1999; Aretz and Wickens 1992] suggest that track-up displays are best for search tasks that involve finding something already indicated on the map, while north-up displays are best for organizing experience, such as keeping track of where one has been while searching for something.

2.4 Discussion

This chapter has highlighted key aspects of current knowledge of perception and cognition. From knowledge about the construction of the visual field and limits on working memory, we learn that the human visual system is designed to operate on only a very few things at a given time. From knowledge about pre-attentive processing, Gestalt laws, and depth perception, we learn that the visual system can also interpret key features that help to group focal items, and understand certain relationships between them in the context of a 3D space. From knowledge about wayfinding and navigation, we see that the visual system is able to weave a coherent whole from patches of focus, stitched together by context. It then uses this coherent whole to enable interaction with the world at large.

CHAPTER 3

FOCUS-IN-CONTEXT USER INTERFACE TECHNIQUES

The visual display is the central part of user-interfaces on general-purpose computers today, as it provides a rich medium for communicating information to computer operators. Yet, because of technological limitations, this display is generally small and of poor resolution relative to the human operator's visual capacity. Thus, many researchers have taken up the challenge of fitting the most pertinent information onto the display at once while excluding or reducing irrelevant information. Most of the solutions developed fall into three broad categories based mainly on how context information is treated.

1. *Distortion techniques* assign screen space to information *partly* by its spatial proximity to a focus item, and partly according to its task relevance. Non-uniform magnification is used to emphasize focus information deemed to be most valuable, while less-immediately valuable context information is minified. This causes severe distortion of the overall image.
2. *Zooming techniques* assign screen space to information *wholly* by spatial proximity to the focus. The user is given the ability to scale the virtual scene, zooming in to get focus details and zooming out to regain context. These techniques rely on the user's memory to keep track of distant

context information while uniform magnification causes focus information to grow until it fills the display.

3. *Multiple reference-frame techniques* treat the display as if it has more than one reference frame, often providing magnified focus information in one reference frame and larger-scale context information in another. They provide the tools necessary for assigning reference frames to information and manipulating these reference frames on the screen.

In this chapter, representative contributions from each category are listed in light of current knowledge of visual perception. In addition, some guidelines are investigated that suggest when and how these techniques should be used.

3.1 Distortion Techniques

For the purpose of this discussion, distortion techniques are defined as those techniques that involve selective, localized magnification or minification of information in the display, without any duplication of information. While not always the case, distortion techniques generally use smooth transitions in magnification and maintain a complete (though degraded) overview of the information space at all times. This definition includes the obvious techniques related to fisheye views, but also includes some techniques that use 3D perspective to render 2D data. Distortion techniques tend to be very good at visually emphasizing focus information and keeping necessary context information in view. However, most distortions present inconsistent spatial layouts that may hamper the effectiveness of recognition and visual working memory (visual WM).

It should be noted that many distortion techniques are often referred to as *focus+context* techniques. Focus+context techniques attempt to combine focus and context information into a single, seamless display, while keeping an overview of the entire information space on the display. Focus+context techniques thus comprise a subset of solutions to the focus-in-context problem. While none of the contributions of this dissertation involve distortion, these techniques help elucidate some difficult areas and provide useful formulations relating to focus-in-context.

George Furnas was the first to crystallize the important concepts central to many distortion techniques with his formulation of generalized fisheye views [1986]. A generalized fisheye view explicitly assigns value to pieces of information. This value is assigned through a degree of interest (DOI) function, and consists of two components. First, every piece of information x is assigned an *a priori* importance, denoted API(x), that essentially ranks the information as to its likelihood of providing good context in the general case. Second, every piece of information is evaluated according to its distance from a given focus item y, according to a distance function denoted D(x, y). The distance function modifies the *a priori* ranking to estimate the likelihood that x provides good context for y in particular. The final form of the degree of interest function is as follows:

$$\mathrm{DOI}(x|.=y) = \mathrm{API}(x) - \mathrm{D}(x, y).$$

Furnas creates a fisheye view of a data space by selecting a threshold value (T), and displaying only the information for which the DOI function evaluates to a value above T for the current focus. For instance, consider Furnas's construction of a fisheye view of a tree data structure, as shown in Figure 3.1. Each node's API is defined to be highest for nodes nearest the root, with API values becoming successively lower toward the leaves. Then distance (D) is defined as the number of arcs between two given nodes. The effect of these two functions is that for any node in the tree, the path from that node to the root is always visible (for DOI thresholds that make the focus node visible). Successive relaxations of the threshold bring more children, siblings, and other close relatives into view.

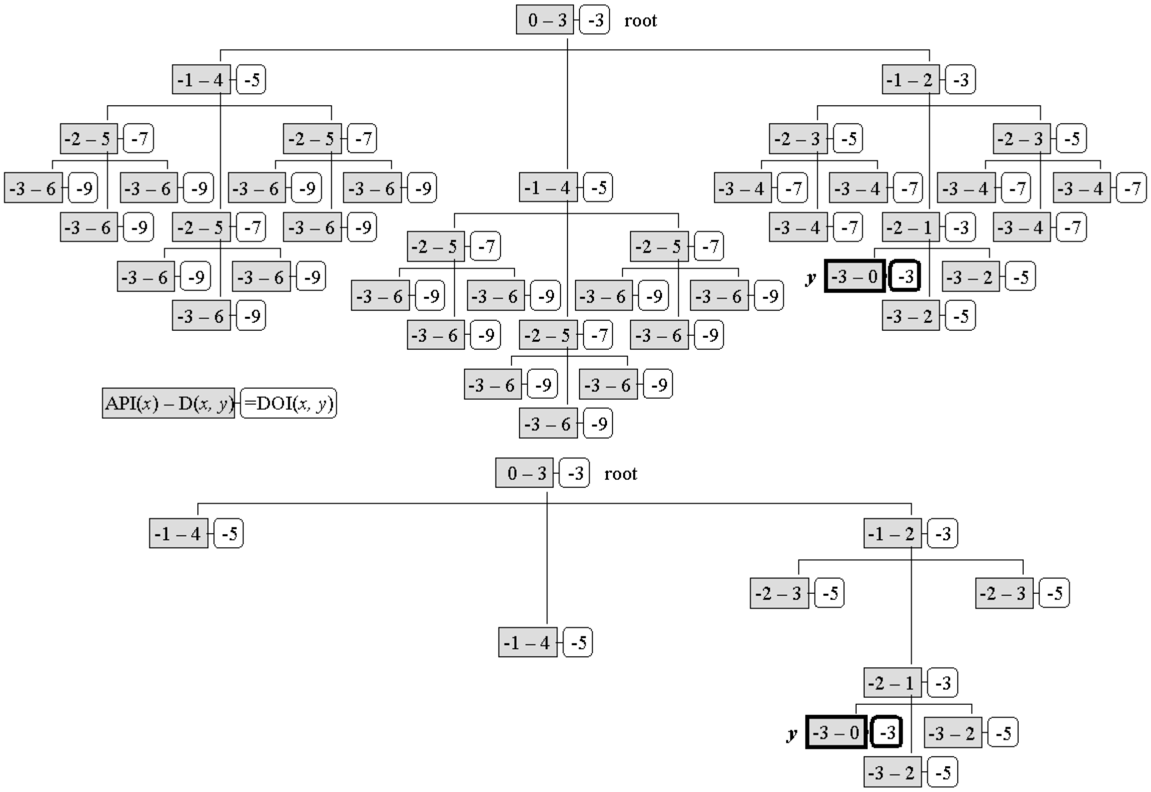


Figure 3.1: Weights in the construction of a fisheye view for a tree data structure (top), and the creation of a fisheye view at threshold –5 (bottom). Adapted from Furnas [1986].

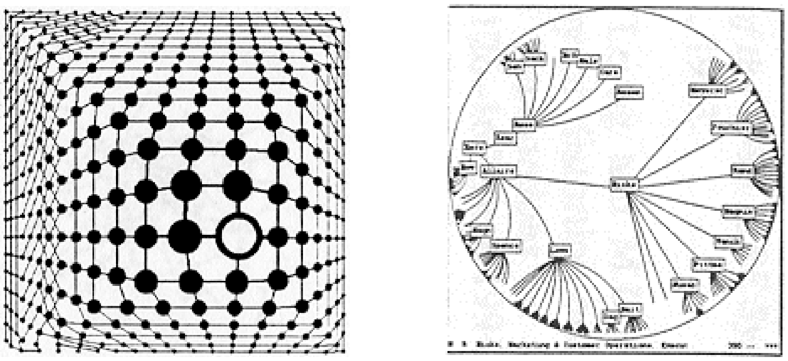


Figure 3.2: Representative examples of fisheye views from Sarkar and Brown [1994] and Lamping et al. [1995] (©1994 ACM and ©1995 ACM, respectively, reprinted by permission).

Many others ([Sarkar and Brown 1994; Carpendale et al. 1997; Lamping et al. 1995] for instance) have built on Furnas's framework to produce fisheye displays that look more like the information was passed through a photographer's fisheye lens. Figure 3.2 shows some representative examples. While such fisheye-lens displays do show more detail about focal information, they stray from Furnas's original ideas in that they do not explicitly hide information other than, perhaps, a label.

In contrast, the Intelligent Zoom of Bartram et al. [1994] changes the representations of information items as they grow and shrink over time. This technique supports multiple scale-dependent representations for items, so that as an item grows and becomes more focal, it can provide more information; as it shrinks, it can provide representations that are legible enough to provide useful context. The Intelligent Zoom is well suited for monitoring tasks because it updates the *a priori* importance for each item continuously, and takes into account the history of recent focus selections in computing the DOI (as shown in Figure 3.3). This allows potentially important events to attract the user's attention, and provides context for where the user's attention has been focused recently. All of these attributes make Intelligent Zoom an excellent 2D solution to the focus-in-context problem within its particular domain: the monitoring of a nested hierarchy in which spatial relationships are not important.

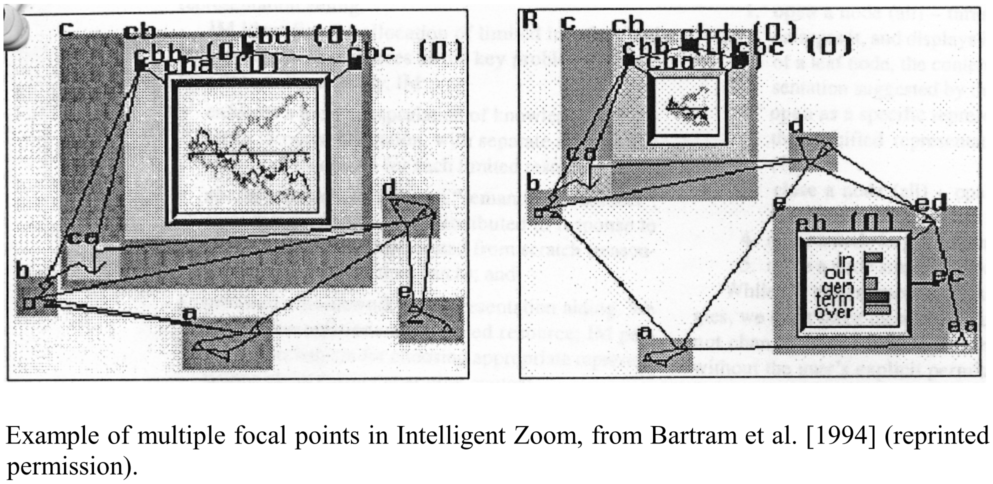


Figure 3.3: Example of multiple focal points in Intelligent Zoom, from Bartram et al. [1994] (reprinted by permission).

Other distortion techniques make more natural use of perspective by displaying 2D information on billboards, and placing more focal billboards “closer” to the user [Card et al. 1996; Robertson and Mackinlay 1993, Robertson et al. 2000]. The Web Forager of Card, Robertson, and York [1996], illustrated in Figure 3.4, uses this approach for web browsing. New focus items are placed in the focus area, while older items can be shrunk back (as if they were being moved away from the user) or placed on a table at the bottom of the workspace for easy access at a later time. A third location is available for placement of information that is not needed immediately. The main insight behind this technique is that it considers context as a sort of visual buffer of relevant items that are easy to access, and provides ways to trade ease-of-access with screen real estate.

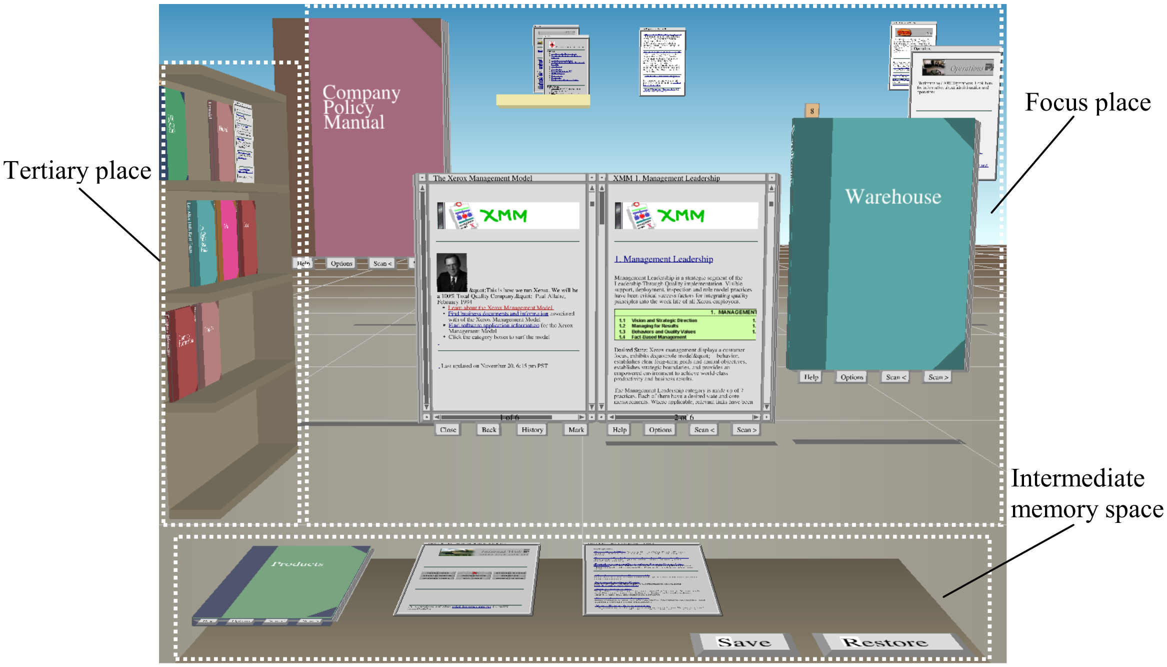
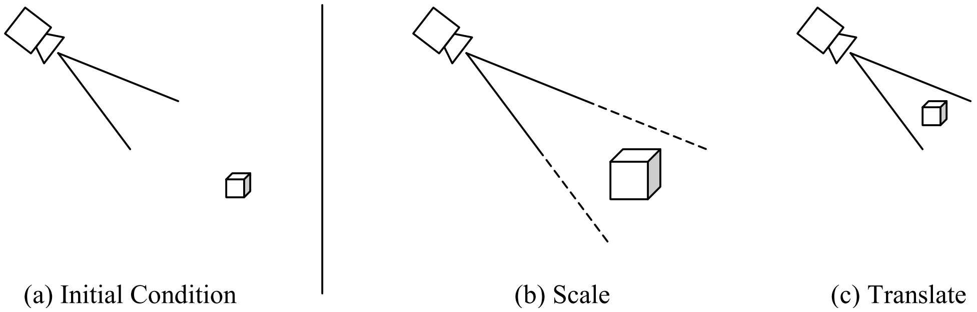


Figure 3.4: A recent incarnation of the Web Forager, courtesy of PARC.

Distortion techniques generally achieve the goal of displaying as many currently relevant focus and context “items” as possible, but they make it difficult or impossible to use information inherent in the arrangement of the items. Changing the focus changes relative distances, directions, and sizes of the information items most relevant to the task at hand. Even if there is no spatial information directly applicable to the task at hand, the distortions that occur can easily slow recognition of groups of items, and lengthen times required to search for previously encountered items of interest.

3.2 Zooming Techniques

Zooming techniques provide a physically undistorted view of information and rely on the user to organize focus and context mentally. These techniques build on certain intuitions of moving between focus and context through user-directed scaling of display space over time. At one point in time, the user zooms in to reach a focus, temporarily discarding context. At the next, focus is discarded by zooming out to review the context. Zooming techniques often provide interfaces that feel natural, are easy to use, and allow efficient completion of certain tasks. However, zooming techniques are not always well suited for applications that require the integration of information across locations and scales, because of the limitations of human memory.


(a) Initial Condition

(b) Scale

(c) Translate

Figure 3.5: The effects of magnifying a focus object and moving towards it look the same on a flat monoscopic display device. Although the relationship of the camera to the object is different after the effects of the operations in (b) and (c), they both result in the same image.

The term *zoom* is used exclusively in this dissertation to mean the scaling of a scene rather than the optical zoom one would encounter using a camera. In the former, the angle of perspective remains constant and the viewpoint moves with respect to the scene; in the latter, the angle of perspective changes while the viewpoint remains fixed in scene coordinates. For simple monoscopic displays, zooming about a point in space is isomorphic to moving the viewpoint toward that point, as illustrated in Figure 3.5. However, this isomorphism breaks down for stereo displays or any other display that provides cues as to the proper depth of focus of the item.

Interfaces in which zooming is an integral component of navigation are called zooming user interfaces, or ZUIs. The Pad++ toolkit created by Bederson and Hollan [1994] has been used to create excellent examples of such interfaces in 2D. Two examples are illustrated in Figure 3.6. The first is a web browser that displays previously visited pages off to the side in ever decreasing scale. Another is a file browser that visually nests directories within each other. The file browser uses scale-dependent representations when displaying textual and graphical files in a spirit similar to Bartram et al. [1994]. Each item first appears as a thumbnail image, but renders in complete detail once the user zooms in close enough.

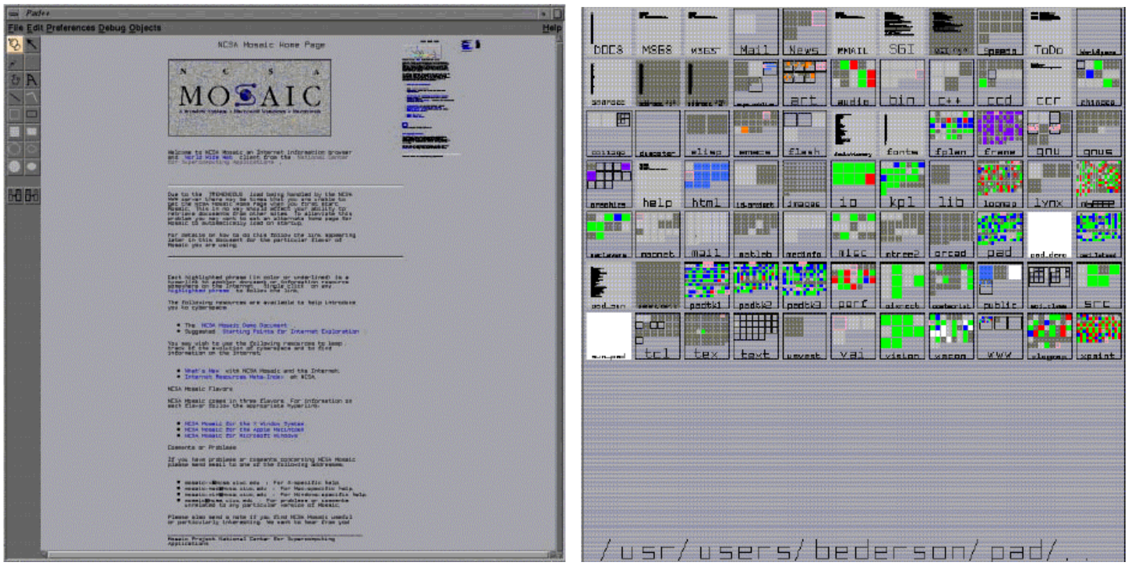


Figure 3.6: Two ZUI’s: a web browser and a file browser created with Pad++ from [Bederson and Hollan 1994] (©1994 ACM, reprinted by permission).

In three dimensions, zooming user interfaces are complicated by the need for rotation. First attempts used camera-oriented approaches: techniques that controlled one's view by specifying the position, direction, and magnification of the camera. For instance, Point of Interest (POI) movement as developed by Mackinlay et al. [1990] allows a user to select a point on the surface of a target object and change the distance of the camera from that point. Upon selection of a point of interest, the camera automatically orients itself to the selected point, as illustrated in Figure 3.7. Once oriented, zooming is simulated by moving the camera closer to the surface at a logarithmic pace (zooming out occurs at an exponential pace).

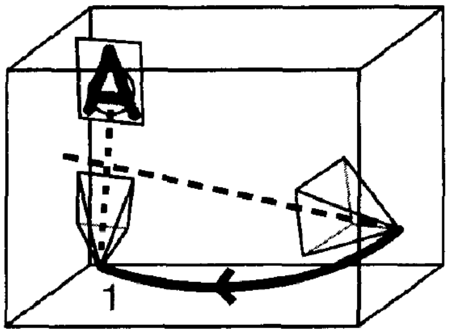


Figure 3.7: Illustration of POI movement from Mackinlay, Card, and Robertson [1990] (reproduced by permission of PARC).

Parker, Franck, and Ware [1998] reduce the complexities of dealing with camera movement by performing the visual near-equivalent: scaling about the selected object. Their NestedVision3D (NV3D), illustrated in Figure 3.8, does this by bringing focus objects that the user selects to a designated focus point in the display (the center of workspace), and providing mechanisms for rotating and scaling the view relative to the objects' center. NV3D also provides for attaching the center of the workspace to a certain object that moves through the environment such that translation of the object is mimicked by the center of workspace.

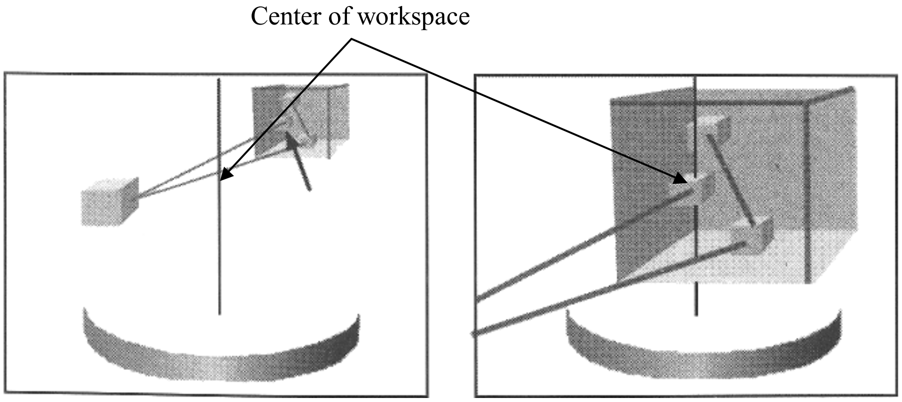


Figure 3.8: Illustration of center-of-workspace movement from Parker, Franck, and Ware [1998] (reprinted by permission).

Through their invention of the space scale diagram, Furnas and Bederson [1995] introduced a way of easily deciding questions involving ZUI's, such as when it would be more efficient for an interface to zoom than to pan. The space scale diagram flattens the scene space to one or two dimensions and adds a dimension of scale, as illustrated in Figure 3.9. The scale dimension represents a scaling factor on the scene that ranges continuously from zero to infinity. The viewing window (Figure 3.9(a)) is represented as a special object that has a constant size throughout all of scale-space. As the viewing window moves in scale space through higher scale factors (from (c) to (d) to (b) in Figure 3.9), it covers less and less of the scene, but what it does contain is magnified according to the current scale factor. Such movement constitutes a zoom operation. Movement of the window perpendicular to the scale dimension results in a pan through the scene.

Space scale diagrams can be useful for illustrating interaction history in a ZUI, but their power comes from the cost model for window movement. The cost model is based on an analysis of the amount of information that enters or leaves the display as the window moves in each dimension. In an environment with information uniformly distributed, panning a window has a cost linear to distance traveled in the scene because it causes a linear amount of information to appear and disappear on the screen. In an environment with roughly the same amount of information at every scale, zooming has a cost logarithmic to the distance traveled in the scale dimension. This is perhaps easier to see if the vertical dimension is defined to be the log of the scale, as in the adaptation on the right in Figure 3.9. In this adaptation, window movement has roughly equal cost in any dimension (or, at least the relationship between dimensions is linear). For every unit moved in scene coordinates, one can expect a corresponding unit of new information to appear on one side of the window. However, new information appears as a constant factor for every unit moved in the scale dimension (exponential)—appearing on the borders when scale decreases, and appearing "between pixels" when scale increases.

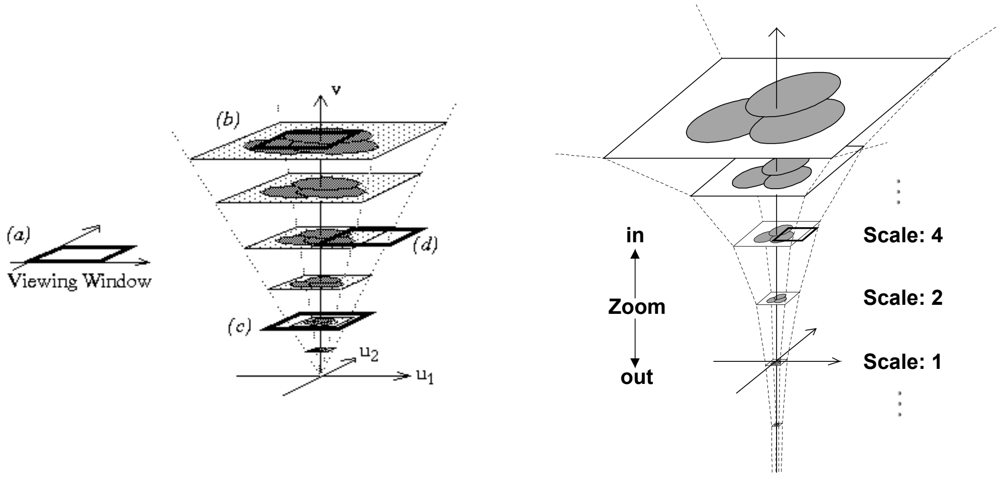


Figure 3.9: A space scale diagram from Furnas and Bederson [1995] (©1995 ACM, reprinted by permission) and a homogeneous-cost adaptation.

3.3 Multiple Reference-Frame Techniques

For the purpose of this discussion, the category of multiple reference-frame techniques is defined as those techniques that treat the display as having more than one frame of reference. Often these techniques create two or more visual instances of the same information. Simple examples include multiple-window interfaces, also referred to as multiple views, and techniques like worldlets [Elvins et al. 1997] that create a miniature copy of part of a scene (in this case to help in recognition of landmarks from other vantage points—see Figure 3.10). However, this category also includes techniques that superimpose multiple reference frames in the same location. This section describes some representative multiple reference-frame techniques, and then discusses specific aids for reducing the cognitive load of fusing context from multiple reference-frames.

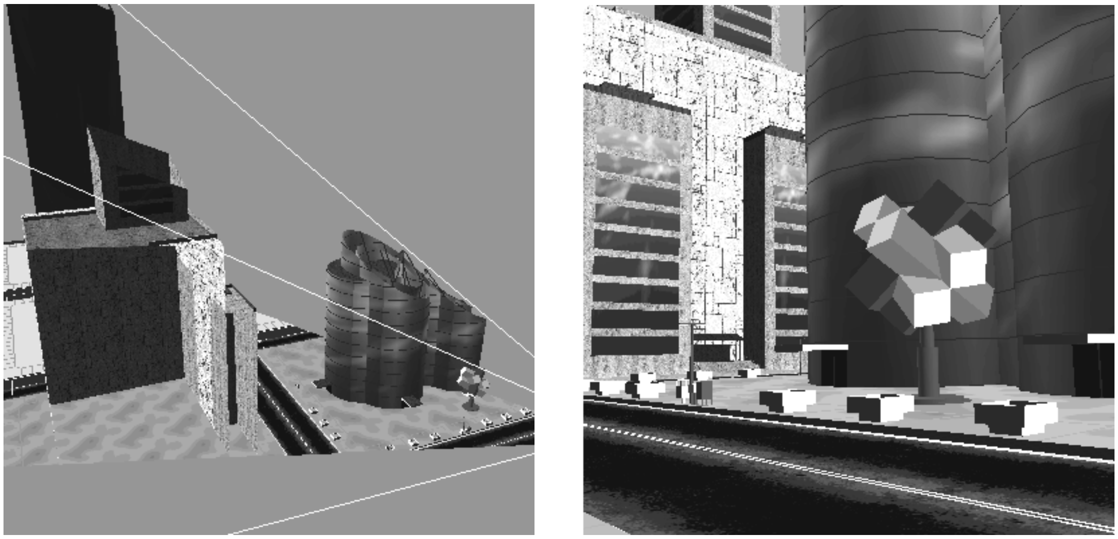


Figure 3.10: A worldlet and the view that generated it from Elvins et al. [1997] (©1997 ACM, reprinted by permission).

3.3.1 Representative Multiple Reference-Frame Techniques

Arguably, the most popular class of multiple reference-frame technique is that of multiple windows, of which the DragMag technique of Ware and Lewis [1995] is a classic 2D example. As illustrated in Figure 3.11, the DragMag consists of a base view (labeled "Base Image") and several zoomed views. The base view provides context by displaying an overview of the entire scene, while the zoomed views provide a mechanism for focusing on finer details. Lines, or tethers, connect the zoomed views to a proxy box (labeled "Mag Window") in a base view. The proxy provides context for the zoomed views by indicating the locations of their content relative to the overview. The user can drag the proxy to a new location in order to update the contents of the associated view.

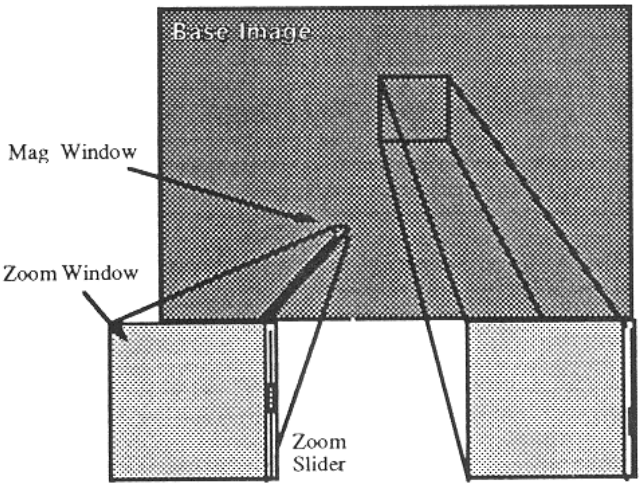


Figure 3.11: The DragMag multiple-window technique from Ware and Lewis [1995] (©1995 ACM, reprinted by permission).

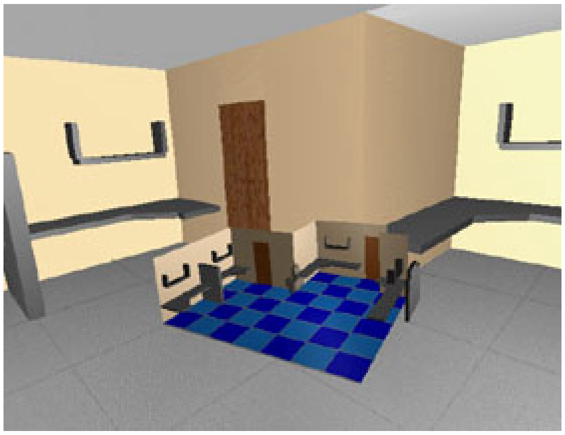


Figure 3.12: Worlds in Miniature from Stoakley el al. [1995] (©1995 ACM, reprinted by permission).

Another class of technique is one that creates a copy of some portion of the scene and duplicates it at a new scale and location [Chuah et al. 1995; Elvins et al. 1997; Stoakley et al. 1995]. Worlds in Miniature (WIM) by Stoakley, Conway, and Pausch [Stoakley et al. 1995], illustrated in Figure 3.12, is a good representative technique. A user can hold and manipulate a miniature replica of the surrounding environment, bringing it up close to get a better look, and spinning it around for a different perspective.

The user can also move things around within the replica to affect the objects in the main scene, including a representation of the user's eyepoint in the main scene. Some researchers have extended this class of technique to apply in more than three dimensions [Becker and Cleveland 1987; Goldstein et al. 1994].

Other multiple reference-frame techniques involve the superposition of reference frames on one another [Beshers and Feiner 1993, Peirce et al. 1997]. For instance, consider the image-plane techniques of Pierce et al. [1997] designed for the purpose of object selection. These techniques take advantage of the ambiguity created when the 2D reference frame of the screen (or image-plane) is superimposed over the 3D reference frame of the virtual scene. One example is the Head Crusher technique, illustrated in Figure 3.13. To use the Head Crusher technique, a user surrounds the image of a target object with the fingers. (Note that the user is wearing a head-mounted display and data gloves, and that virtual fingers appear in the display to align with the user's fingers.) The insight here is that due to the nature of the human visual system, a number of 3D problems may have simple 2D solutions, and that overlapping reference frames is one way to find these solutions.

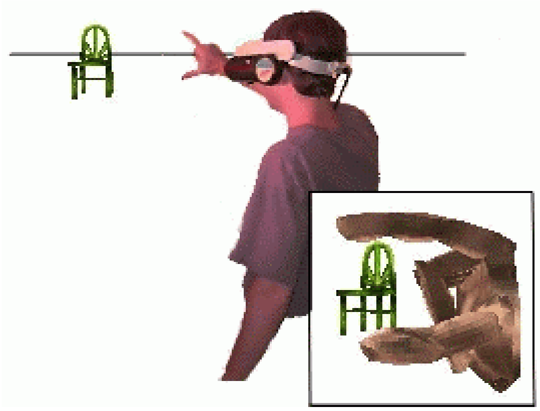


Figure 3.13: The Head Crusher technique from Pierce et al. [1997] (©1997 ACM, reprinted by permission).

Multiple view techniques generally do not suffer from the spatial distortion and memory problems associated with distortion and zooming techniques, respectively. However, multiple views generally carry additional overhead due to their creation and maintenance, and to the cognitive load of integrating their information.

3.3.2 Linking Aids

The cognitive load imposed on users can be high when it is necessary to integrate information seen from each of several views that are simultaneously displayed. In addressing this problem, researchers have most often turned to three kinds of artificial aids for linking frames of reference. Two types of these aids visually express an existing relationship, while the other enforces a visual relationship that would not otherwise exist:

1. View proxy—the explicit representation of one view (or point of interest) within another. Figure 3.14 shows a view proxy indicating location and orientation of the viewpoint in two dimensions along with a box indicating the front portion of the view.
2. Tethers—explicit lines connecting one view (or point of interest) to its location in another. Figure 3.14 demonstrates a two-tether interface that connects corners of the focal view closest to the proxy with the corners in the contextual view.

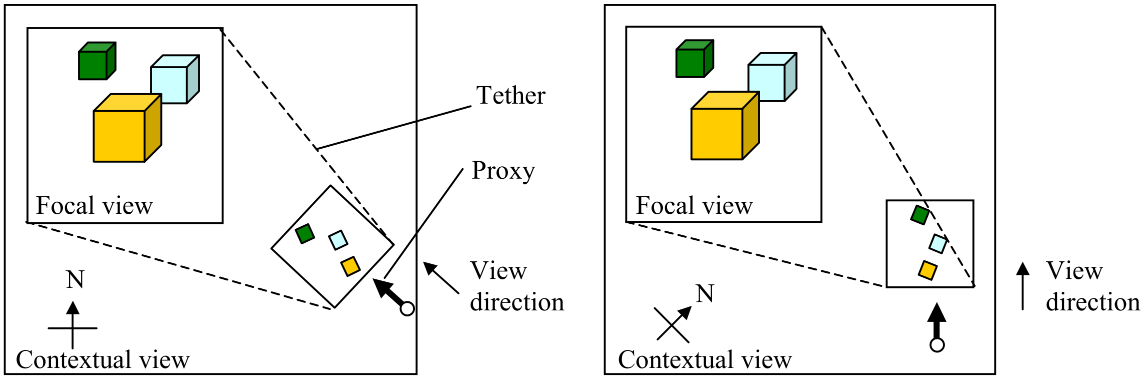


Figure 3.14: Examples of tethers, proxies, and orientation coupling.

3. Orientation coupling—an implicit algorithmic aid that keeps two views oriented in similar directions. The right-hand image in Figure 3.14 demonstrates a track-up coupling: a contextual view that keeps its "up" direction aligned with the "forward" or "up" direction of the focal view. This contrasts with the more traditional north-up contextual view (or *overview*) in the left-hand image.

We have already seen instances of proxies in DragMag (the 2D Mag Window boxes) and Worlds in Miniature (the 3D camera representation) [Ware and Lewis 1995; Stoakley et al. 1995]. 2D Proxies are also pervasive in video games and urban maps through the ubiquitous "You Are Here" markers. Yamaashi et al. [1996] demonstrated that a "linked" proxy (one that can be used for navigating its represented window) could reduce the time needed to perform a multiscale identification task. Their task required that subjects monitor a wider-angle video display for the presence of a character in the scene, and then cause a second video display to zoom in so as to identify the character when it appeared. The number of user operations was reduced because moving a linked proxy in two dimensions was a single operation, while zoom, pan, and tilt operations had to be composed to move a detail view around in the absence of a linked proxy. The time reduction (roughly 45%) was accounted for by the reduced number of operations.

Tethers appear commonly in printed illustrations. Illustrations in magazines such as National Geographic "blow-up" a portion of an image and connect this blown-up portion to a proxy on the overview image, similar to DragMag [Ware and Lewis 1995]. The Spiral Calendar of Mackinlay, Robertson, and DeLine [1994] uses semitransparent, planar tethers (see Figure 3.15) to connect calendars on successively longer time scales. These tethers help to point out one calendar's location in the context of the next wider-scaled calendar. The Starlight system described by Risch et al. [1997] uses tethers in a different way, through a construct they call tie-nodes. Tie nodes connect different representations of the same object in respective views or contexts as illustrated in Figure 3.16.

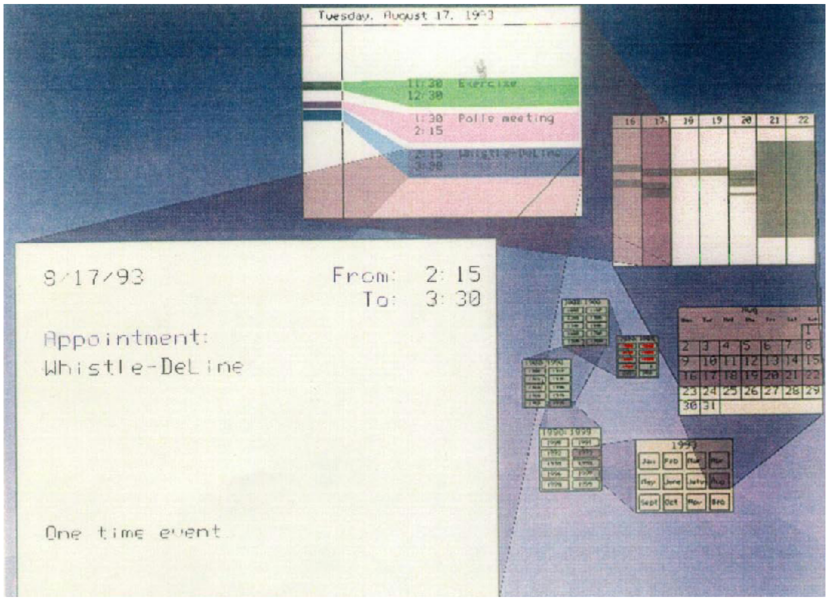


Figure 3.15: Tethers in the Spiral Calendar [Mackinlay et al. 1994] (reproduced by permission of PARC).

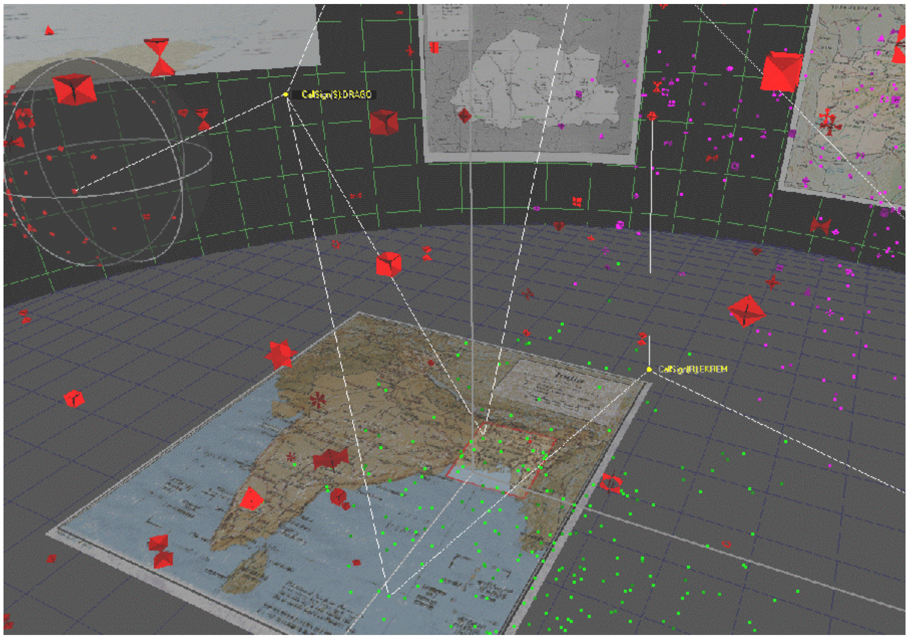


Figure 3.16: Tethers called tie-nodes connect instances of the same objects in different contexts in Risch et al. [1997], © 1997 IEEE.

Orientation coupling is perhaps the best studied of the three linking aids, if not as widely used in virtual environments. Orientation coupling causes a view to update its orientation so that it always matches the orientation of another view. Some of the work related to orientation coupling was covered in the discussion about north-up versus track-up maps in Chapter 2—a track-up map is one kind of orientation coupling. Other research involves virtual environments. For instance, Darken and Cevik [1999] investigated the utility of orientation coupling in several applied search tasks. For each task, subjects were asked to navigate a virtual world with the aid of an overview map. The results confirm for these applied tasks what Aretz and Wickens [1992] had found in more abstract situations. They suggest that track-up map displays are best when the search target is clearly marked on the map: the user can essentially navigate using just the track-up overview display. The results further suggest that north-up map displays are best when the destination is either unknown, or has been previously visited but is not shown: the consistency of a north-up map allows the user to remain oriented with respect to an organized search strategy.

3.4 Usage Guidelines

This section briefly lists the guidelines of relevance to the research presented in this dissertation, with particular emphasis on empirical results and results regarding multiple-window techniques.

Wang Baldonado et al. [2000] provide design guidelines for multiple windows with regard to how they should be applied in a given interface. Of particular relevance are their rules of complementarity ("Use multiple views when [they] bring out correlations and/or disparities"), space/time resource optimization ("Balance the spatial and temporal costs of presenting multiple views with the spatial and temporal benefits of using the views"), and self-evidence ("Use perceptual cues to make relationships among multiple views more apparent to the user").

Empirically derived guidelines exist for both zooming and multiple-window interfaces. Experiments performed by Guo et al. [2000] suggest that a zoom rate of 8x per second is optimal for ZUIs. Regarding multiple-window interfaces, results from an experiment performed by Plaisant et al. [1992] suggest that a maximum scale difference of 25x be present between a focal window and its contextual source window. Results from experiments by North and Shneiderman [2000] suggest that coordination among multiple windows is essential for efficient performance of tasks that require information at multiple scales. Such coordination includes simultaneous updating or highlighting of corresponding bits of information among the views and making clear the relationship between overview and detail views.

A few studies have been carried out that compare zooming and other focus-in-context techniques for certain tasks. Schaffer et al. [1996] compared zooming with a multi-focus fisheye technique they call *variable zoom* (a close relative of Intelligent Zoom [Bartram et al. 1994]). Their experimental task was a directed multiscale search in a nested hierarchy of constant size. The variable zoom excelled at this task. Combs and Bederson [1999] compared image browsers with various ways of navigating an image set: a 2D zooming interface, two 3D interfaces (using carousel and landscape metaphors), and a traditional scrollbar-driven interface. Their experimental task asked subjects to browse images in variously sized sets to find a target image. Their results suggest that 2D zooming navigation is more effective than the 3D methods for larger image set sizes, although other 3D methods such as 3D zooming were not considered.

3.5 Discussion

This chapter has reviewed a number of focus-in-context techniques in the categories of distortion, zooming, and multiple reference-frame techniques. From the distortion techniques, we learn the importance of assigning screen space according to user attention, but we also see the incompatibility of distortion with inherently spatial information. From the ZUI's we learn how to allocate attention without distortion, but we also see something of the burden zooming puts on the memory of the user. From the multiple reference-frame techniques, we learn how to retain views at multiple scales without distortion, but we also come up against the cognitive costs of integrating these views into a meaningful whole. We have seen some linking aids for reducing these costs, but more work needs to be done in this area. Finally, from the guidelines and comparison regarding the various focus-in-context techniques, it becomes obvious that many successful comparisons between techniques are done in the context of applied tasks.

Although the guidelines that have been discussed suggest when to use which techniques, they are generally vague and/or unprincipled. They do not take into account the perceptual and cognitive issues discussed in the previous chapter. This problem is addressed by the cognitive systems model in Chapters 7 and 9.

CHAPTER 4

MOTIVATING APPLICATIONS

In this chapter, a concrete application area is described that provides motivation for the main goal of the dissertation: to increase the comprehensibility of multiscale 3D spatial information by taking into account the need to display focal information in its larger context. The application area is 3D geospatial visualization as applied to mapping the seafloor and objects in the water column above the seafloor. Tasks in this application area include discovery tasks, vehicle control tasks, and presentation tasks. The chapter concludes with the strategy the dissertation takes in light of the motivating applications.

4.1 Motivating Applications

Underwater mapping involves vast quantities of 3D multiscale data that is fundamentally geospatial in nature. This data tends to span large geographic areas with ever-increasing detail, and the sheer volume of data tends to conceal complex 3D relationships. Researchers who wish to make sense out of the data often turn to geographic information systems (GIS) and scientific visualization for help. At their most basic, such systems enable researchers to see information in a more intuitive form than raw, abstract numbers. With the proper interactive tools and animation capabilities, these systems can enable researchers to perceive 3D relationships that would otherwise be obscured or invisible.

Underwater mapping also involves the use of technologies such as ROVs (remotely operated vehicles) and AUVs (autonomous underwater vehicles) to collect the information that researchers want to investigate. Proper operation of these technologies often requires that an operator be able to integrate information from multiple viewpoints, potentially in situations of limited visibility. The reference frame of an underwater vehicle must be understood in the contextual reference frame of the larger survey area in order to properly guide the vehicle to areas of interest and interpret the (potentially incomplete) imagery and information being collected. An operator would benefit greatly from a visual display environment enhanced with interactive tools that help to integrate information from these frames of reference.

It is the development of such interactive tools and guidance for their use that this dissertation seeks to address. In some cases, it is easy to develop tools and guidelines with generic applicability, but in many cases more work is required. In order to design innovative sets of interactive tools and evaluate each set's effectiveness against the other, it is often necessary to identify the key representative tasks that require improved support. With this in mind, let us consider three types of tasks that have motivated development within this dissertation: *discovery*, *vehicle control*, and *presentation*. Each of the tasks involves the interactive display of local information within a wider context, which is the central problem addressed in this dissertation. These tasks provide a sort of ground-truth against which the contributions of the dissertation can be applied and evaluated.

4.1.1 Discovery Tasks

Discovery tasks involve the detection of relationships and trends in underwater data that were not previously known. For example, a biologist may be interested in understanding the life cycle and habitat of scallops or other species. The biologist could use information on scallop population densities and combine that with the bathymetric, temperature, and bottom-type information in selected representative areas for similarities with each other, and differences from areas of low population. Proper identification of scallop habitat may involve comparing the small-scale information from many locations, and seeing how these pieces of information relate to one another. For instance, a researcher could identify the local conditions necessary for reproduction as well as the conditions necessary for healthy growth. The researcher could then look at the prevailing currents over larger scales to determine the most likely progression of scallop populations through generations of growth and reproduction. The multiscale comparison that occurs while identifying the reproduction and growth habitats for scallops is the same sort of comparison that is important for many visualization applications (not just oceanography).

4.1.2 Vehicle Control Tasks

Vehicle control tasks include planning, monitoring, and real-time control. These tasks could be part of a simple mission such as navigating a single vessel through a channel, or a complex mission such as orchestrating several AUVs and surface vessels in a mapping expedition. First, in planning tasks, it may be necessary to specify waypoints across widely varying scales. For instance, when planning the route for an AUV in a mapping expedition, several waypoints at which video and measurements should be taken may be within a few tens of meters of each other, but another cluster of waypoints may be several kilometers away. It would be useful to be able to link the reference frames of the clusters of waypoints within the reference frame of the entire expedition so as to see the detail of the local clusters within the wider context.

Second, in monitoring tasks, it may be necessary to have several vantage points available so that incoming information can be properly interpreted and acted upon. For instance, an operator viewing a video recording from a past ROV dive may spot features that should be investigated further. The reference frame of the video feed in a new dive must be understood in the wider context of information already known about the area in order to be able to efficiently return the ROV to scattered locations of interest.

Third, in real-time control tasks, a situation may arise that requires swift intervention, and it may be necessary to have views that update themselves appropriately. For instance, when monitoring a group of AUV's, it may become apparent that a collision is imminent. Instructions must be quickly issued to the appropriate vehicles to avoid the catastrophe. An operator must be able to quickly manipulate the vehicle in its own reference frame using knowledge gained from the context of the larger reference frame.

4.1.3 Presentation Tasks

Presentation tasks involve communication of information to scientific colleagues, students, investors, or the general public. In these tasks, the creator of the presentation is usually pointing out important features and relationships between them, as well as processes involving these features. For instance, in explaining why scallop populations appear to migrate over the years, a biologist might point out particular regions with the right local bottom features and water temperature to support scallop reproduction, and then show how ocean currents carry developing scallops to the locations in which they mature. It might be beneficial to have one window showing an area suited for reproduction and another for areas suited for growth, all in the context of a third that shows the currents leading from one to the other. The presentation may involve explanation at multiple scales, with processes and relationships involving all three dimensions. During such a presentation, it is necessary to direct the attention of the audience to various features in an orderly manner, while maintaining enough context to avoid audience confusion and enable audience members to see complex relationships that are not easily discerned.

4.2 Research Strategy

In light of the applications just described, the goal of the dissertation is met by developing new techniques to make a 3D geospatial visualization system more effective and by determining how existing techniques can be used more effectively. This dissertation provides an integrated set of techniques aimed at creating an effective multi-scale interface, together with a predictive cognitive model that can be used to guide key design decisions. Two empirical evaluations serve as the basis for this model, which predicts when one navigation mechanism should be used over another as well as providing some measure of how many errors can be expected. Additional experiments provide guidance in the use of the specific interface techniques developed in this dissertation.

CHAPTER 5

FRAME-OF-REFERENCE INTERACTION[1]

As discussed in the introduction, interfaces that address the focus-in-context problem generally have the goal of maximizing user performance (speed, accuracy) on a given task, while minimizing display and attentional requirements. In this dissertation, the problem is addressed primarily by establishing appropriate frames of reference, then linking these reference frames together in meaningful ways. This chapter describes frame-of-reference interaction (FoRI), an integrated set of techniques for the presentation and manipulation of reference frames, along with a software framework for realizing these techniques. After briefly outlining what capabilities frame-of-reference interaction supports, this chapter defines the *frame-of-reference* (FoR) concept, and how this concept applies to focus and context. It then provides a brief overview of the software framework that realizes the FoRI techniques. Following that are the details of exactly what forms reference frames take in FoRI, how a user can interact with them, by what mechanisms reference frames can be linked, and how new reference frames can be created to aggregate other reference frames.

The core interaction techniques of frame-of-reference interaction support the following capabilities:

[1] Parts of this chapter have been published in modified form in [Plumlee and Ware 2003b], [Arsenault et al. 2003], [Plumlee and Ware 2002b], and [Plumlee et al. 2001].

1. view manipulation with respect to the current point of attention;
2. object manipulation that feels natural regardless of the current view;
3. the ability to link two or more views together to work in tandem, usually with one providing focus information and the other providing a contextual overview;
4. the ability to couple a view to navigation in another view, or to a moving object; and
5. the ability to couple a view to abstract objects such as aggregate collections or the closest pair among a group of objects.

Central to providing these capabilities is the appropriate use of frames of reference.

5.1 Frames of Reference

For the purposes of this dissertation, a *frame of reference* (FoR, or reference frame) is defined as a collection of position, orientation, and scale information (seven quantities: x, y, z; roll, pitch, yaw; and magnification factor). In any spatial system, there is an inherent coordinate system on which all reference frames are based, often referred to as the *world* reference frame or *world coordinates* or *scene coordinates*. In many 3D interfaces, two additional kinds of reference frames are commonly used. One is for viewpoint control. A common technique is to fly a virtual camera around the scene, using a viewpoint-centered FoR: the origin is at the viewpoint and orientation is relative to the direction the camera is facing. The second kind is for object manipulation. When objects are to be moved within the scene, an object-centered FoR is adopted that specifies how the object rotates and scales. These two kinds of reference frames are not immediately compatible, in the sense that it can be difficult to properly align views with objects of interest.

5.1.1 Interaction Reference Frames (I-FoRs)

The reference frames of views and objects are unified in frame-of-reference interaction through the concept of an *interaction reference frame* (I-FoR). An I-FoR serves as the reference frame for a view, with its origin situated at a designated "look-at" point within the view. In general, the user's attention is predominantly directed to objects at or near the origin of an I-FoR. During user interaction, objects are brought near this point to be investigated, manipulated, or edited. I-FoRs are also useful for linking views to moving objects and interlinking multiple views.

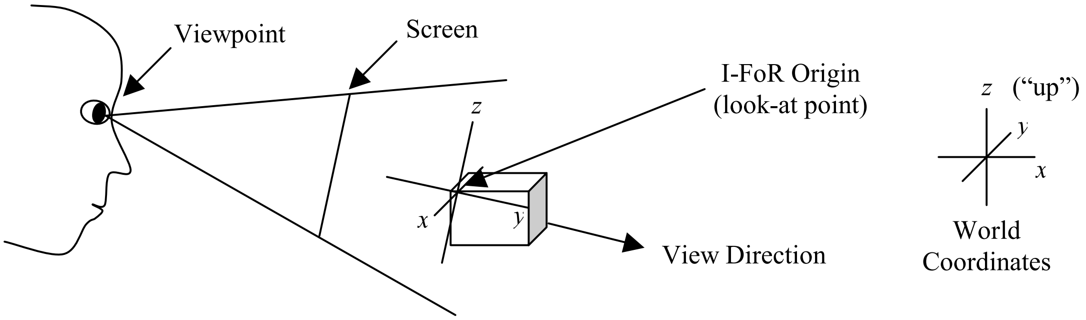


Figure 5.1: Defining an interaction reference frame in terms of an observer and a point of interest. The position, orientation, and scale of an I-FoR are given in terms of translation, rotation, and scale with respect to world coordinates.

An I-FoR is illustrated in Figure 5.1. The origin is situated at the center of a view, conceptually at about arm's length from the user. One axis (*y*) of the interaction I-FoR is pointing directly away from the observer, another axis (*z*) is vertical with respect to the observer, and the third (*x*) is horizontal with respect to the observer.

The orientation of an I-FoR is designed such that simple viewpoint control corresponds well to the way a person normally investigates the world, even though the center of rotation is not at the viewpoint, but at a look-at point. In everyday locomotion through a real-world environment, the orientation of an observer's egocentric view is primarily determined by heading—rotation of the body or head parallel to the ground plane, as illustrated in Figure 5.2(a). As such, the primary orientation component within an I-FoR (the one applied first in the matrix algebra of viewpoint control) is heading. Changes in heading cause the scene to rotate about the look-at point within the *x*-*y* plane of world coordinates, or equivalently, changes in heading cause the viewpoint to orbit the look-at point in the *x*-*y* plane of the world as illustrated in Figure 5.2(b). This corresponds well to the kind of investigation done by humans in the real world: if the object is small enough to be picked up, it is rotated in place; if the object is large, a person will often move around the object while maintaining their gaze on the object.

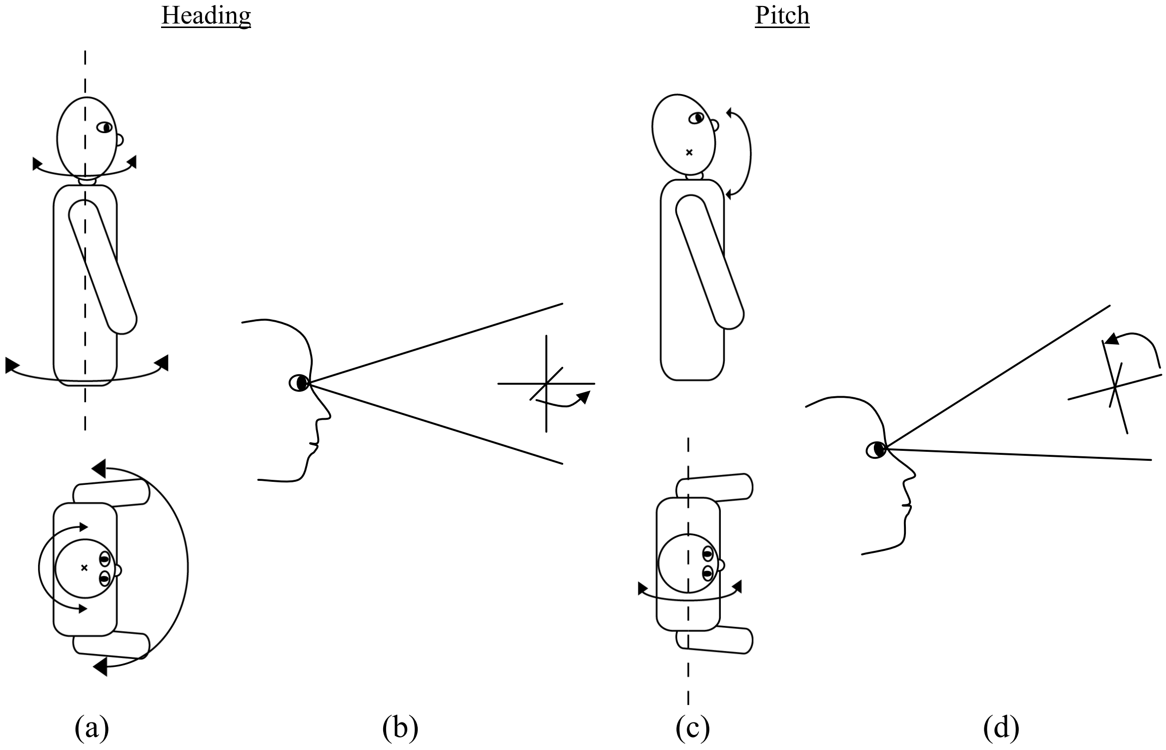
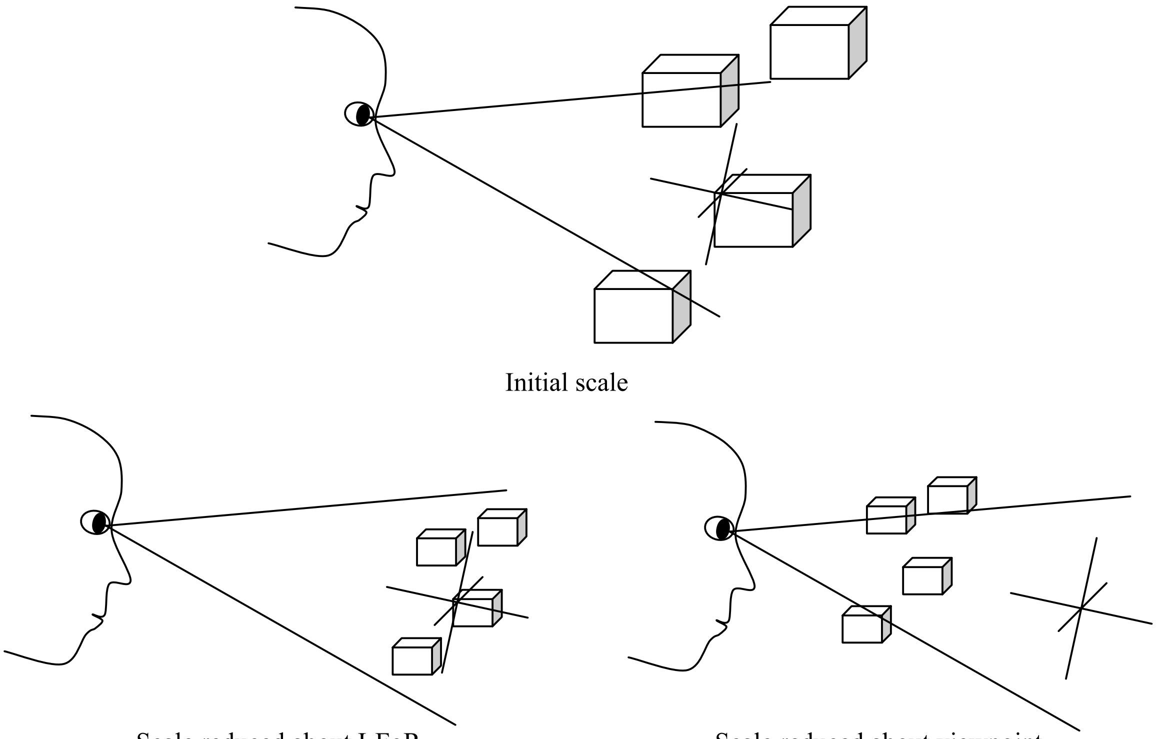


Figure 5.2: The effects of orientation on a viewpoint. (a) The primary rotations people make are in heading, either with their bodies or their heads. (b) Changes in I-FoR heading cause the viewpoint to rotate about the I-FoR origin in the world's *x*-*y* plane. (c) The other common rotation people make is in pitch—looking up or down with their heads. (d) Changes in I-FoR pitch cause the viewpoint to rotate about the *x*-axis of the transformed I-FoR.

A secondary component of observer orientation in the real world is pitch—people tilt their heads forward and backward as illustrated in Figure 5.2(c). As such, the secondary orientation component within an I-FoR is pitch (applied second in the matrix algebra). Changes in pitch occur with respect to the *x*-axis of the I-FoR after any rotation due to heading, and similarly cause rotation to occur about the look-at point as illustrated in Figure 5.2(d). By applying rotations in this order, the observer always keeps the "up" vector of world coordinates within the *y*-*z* plane of the I-FoR. In other words, the "up" vector always projects to either a vertical line on the display, or it points toward or away from the observer.

For completeness, I-FoRs also contain a roll component, corresponding to tilting one's head sideways. However, people rarely tilt their heads sideways, and the interaction techniques described in this chapter do not make use of the roll component. With this roll component included, the full calculation that an I-FoR applies to every point v in the scene before it is rendered to the display is as follows: $v' = SRPHTv$, where S is a scale matrix, R, P, and H are roll, pitch, and heading rotation matrices, and T is a translation matrix, each using the corresponding I-FoR information. For comparison, the view of a human moving around in the real world might be characterized as $v_{world}' = RPHTv_{world}$ (same order, but no scale). The key considerations are that points are translated to the observer before any rotation or scaling occurs, and that rotation occurs in the same order in FoRI as it does in the real world.

The scale of an I-FoR controls the size of the scene relative to the observer (or equivalently, the effective size of the observer with respect to the scene). Because the origin of an I-FoR is at the look-at point and not at the eye, objects at the look-at point remain there after scaling. The difference between scaling about the viewpoint and I-FoR scaling is illustrated in Figure 5.3. Scale affects the amount of the 3D virtual environment that is visible around the look-at point. Users perceive zooming in (increasing scale) as getting closer to an object and zooming out (decreasing scale) as getting further away.


Initial scale

Scale reduced about I-FoR

Scale reduced about viewpoint

Figure 5.3: The effect of scale on perceived distance and the extent of spatial context provided to the observer.

5.1.2 Object Reference Frames (O-FoRs)

Just as an interaction reference frame provides a useful way to deal with view control, an *object reference frame* (O-FoR) provides a way to describe object orientation and position. An O-FoR is defined as a reference frame used to describe a potential target of attention, usually an object or group of objects. The placement and orientation of an O-FoR with respect to an object often comes from the symmetries inherent in that object. For example, the O-FoR of a vehicle might have its origin situated at its centroid or at the place a driver might sit, such as the bridge of a ship, with orientation pointing in the direction of motion and with a scale of one. Every object has an associated O-FoR, but O-FoRs can exist on their own and be used for other purposes. One alternative use is in designing a focus target specifically for viewing in the context of an I-FoR. For example, an O-FoR might be created for a group of vehicles with its origin at the group's center (as illustrated in Figure 5.4), orientation pointing in the average direction of motion, and scale indicating the spatial extents of the group. Frame-of-reference interaction provides a way of linking a view's I-FoR to a group's O-FoR such that interaction can occur with the object *group*, as opposed to just the individual objects.

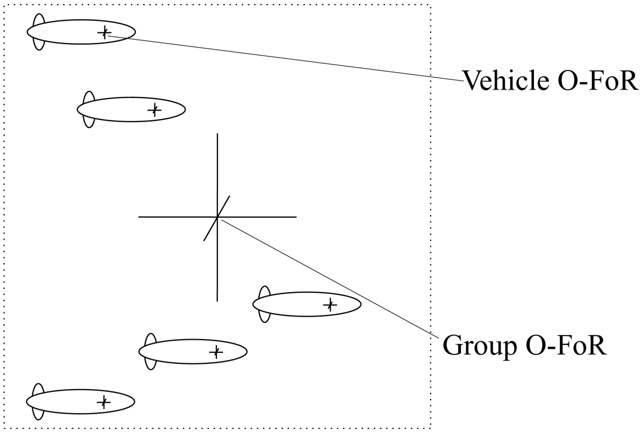


Figure 5.4: Examples of O-FoRs—vehicle O-FoRs and a group O-FoR.

One distinct difference between I-FoRs and O-FoRs, besides their usage, is the order in which components are applied. Whereas an I-FoR applies the translation matrix T (for its x, y, and z components) before rotation or scale, an O-FoR applies translation last. In other words, an I-FoR applies its transformations such that it rotates and scales about the look-at point ($v' = SRPHTv$); an O-FoR applies its transformations such that it rotates and scales about its own origin ($v' = TSRPHv$). In addition, O-FoRs may not apply certain transformations. For example, objects that are serving as glyphs may not need to be rotated or scaled. These objects may use O-FoRs that do not use any of the rotation or scale information. Such behavior can be useful when using general purpose manipulation widgets—operations that do not apply to the object can have their widget representations removed.

5.2 Frame-of-Reference Interaction Software Framework

The basic software framework that enables the frame-of-reference interaction techniques to be implemented effectively is illustrated in Figure 5.5. The core components of the software framework are as follows:

- **FoR**. A FoR encapsulates translation, rotation, and scale information.
- **Zoomport**. Zoomports are windows that use I-FoRs to control the view.
- **Object**. Objects include anything that can be rendered and moved around in the scene. Each object has an O-FoR.
- **Interactor**. Interactors translate user input into operations on FoRs. They include things like navigation widgets and object manipulation tools.
- **FoR Relationship**. FoR relationships are all fundamentally geometric operations on FoRs, and include couplings and FoR-ops.
    - **Coupling**. Couplings cause two FoRs to change in tandem. Any changes to coupled components of one FoR are reflected in the other.
    - **FoR-Op**. FoR-Ops are frame-of-reference operations that summarize or transform information from other FoRs into a resultant O-For. FoR-Ops act as abstract Objects, and therefore each FoR-Op has an O-FoR.

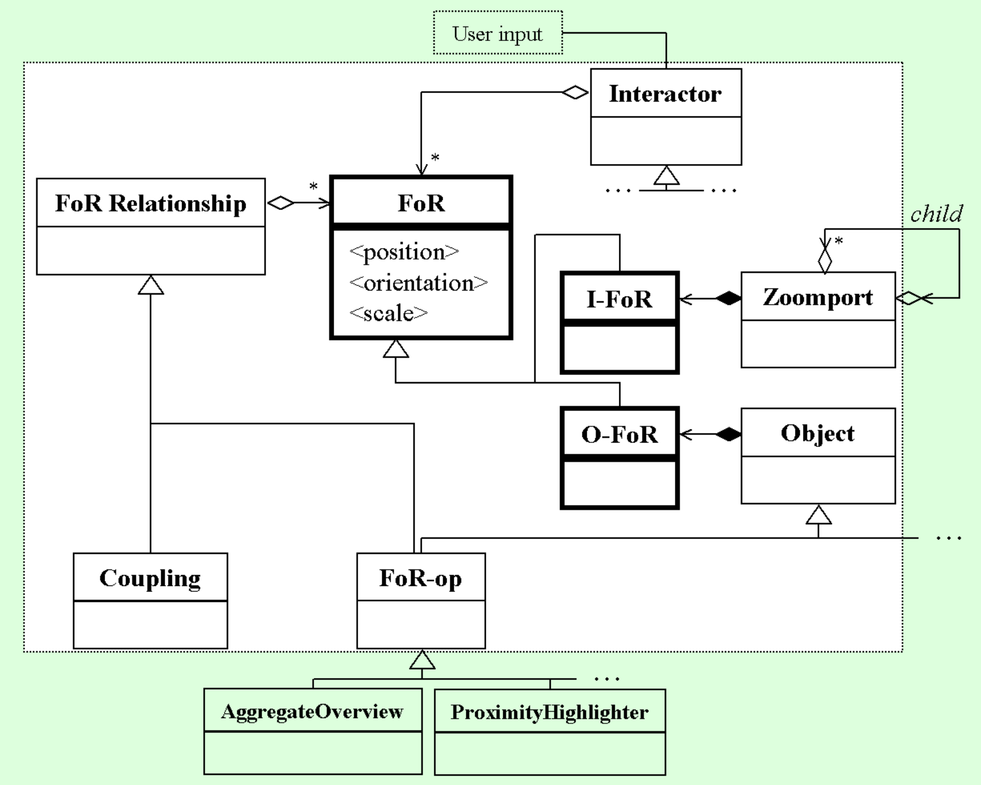


Figure 5.5: Diagram of the software framework that supports frame-of-reference interaction. Core components are within the white dashed box.

The sections that follow describe in detail the frame-of-reference interaction techniques, as well as how the software framework is used to implement each of the capabilities listed above.

5.3 Capability 1: View Manipulation

At the core of frame-of-reference interaction is center-of-workspace interaction, which is how view manipulation occurs within a single window. Center-of-workspace interaction designates a point in the 3D display as the central focus for an interaction reference frame. This point is called the *center of workspace*, and is positioned just behind the screen, conceptually at about arms length. It corresponds to the "look-at" point for a virtual camera on the workspace.

In center-of-workspace interaction, the center of workspace is the place where parts of a scene can be brought rapidly and where the interface is optimized for further interaction. As center-of-workspace interaction is implemented in GeoZui3D, a point in the scene is moved to the workspace center (or equivalently, the view is navigated to a different point) when the user clicks on that point with the middle mouse button. This input causes a smoothly animated translation of the selected point to the center of workspace. While still holding the button, the user can move the mouse forward or backward to zoom in or out, respectively, about the center of workspace. The zoom rate is held at a constant scale factor of 8 times magnification per second (8 x/s). The constant rate of scale change was chosen on the basis of the careful empirical study described in Appendix A.

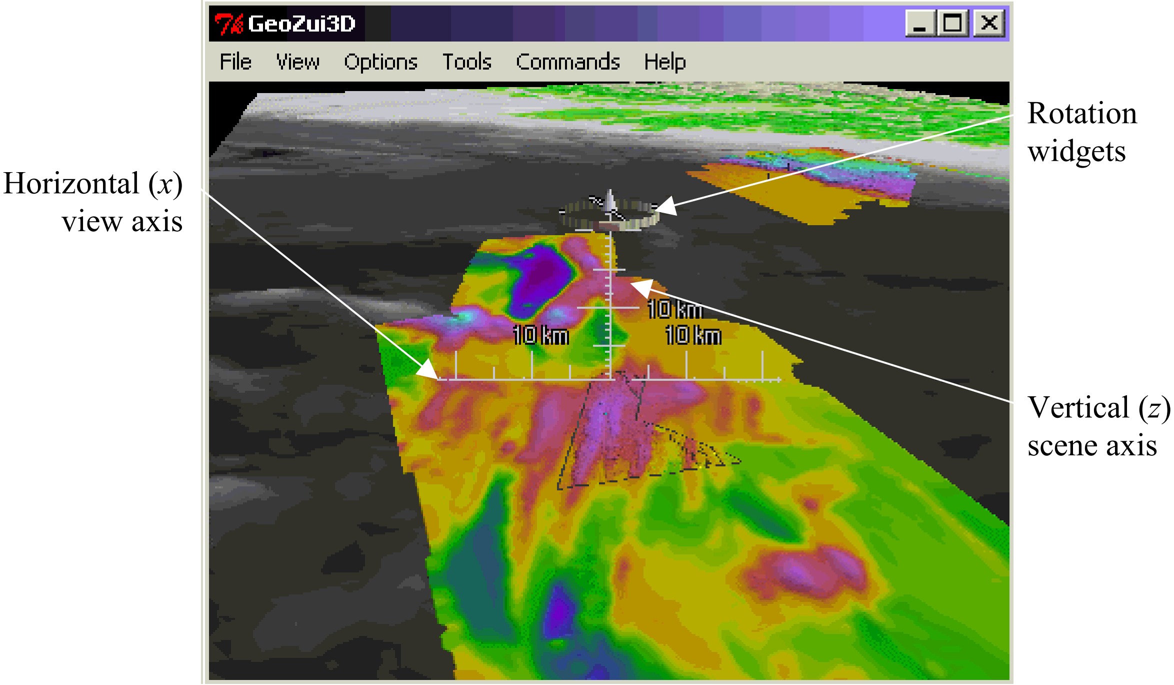


Figure 5.6: 3D widgets at the center of workspace, as implemented in GeoZui3D. The *y*-axis of the I-FoR at the center of workspace points directly into the page.

The user can also rotate and scale the view using a set of 3D widgets, as shown in Figure 5.6. These widgets allow for direct manipulation of the FoR's orientation and scale by grabbing various handles. Pitch is controlled directly by clicking on the yellow ring or white cone and dragging vertically. Heading is controlled by clicking and dragging on the pink button found on the yellow ring. This causes the scene to rotate about the center of workspace with a rate of rotation proportional to the distance the pink button has been dragged. The widgets can be quickly hidden or restored by pressing the *w* key on the keyboard.

The widgets provide an alternative way of manipulating the overall scale: a user can click and drag a scale tic-mark on the horizontal axis (away from the center to scale in, or toward the center to scale out). The scale factor is changed in such a way that the tic-mark is always in the same vertical line as the mouse cursor. Height exaggeration (scaling only in the *z*-dimension) is controlled in a similar way when a user clicks and drags a scale tic-mark on the vertical axis.

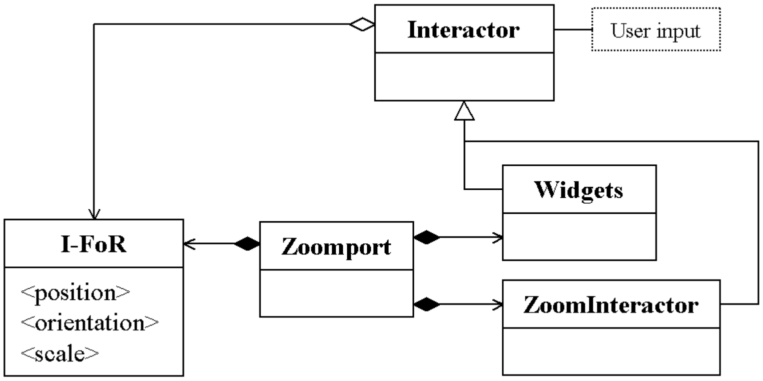


Figure 5.7: Diagram of the relationship between a zoomport (window) and a FoR.

Figure 5.7 shows how the software framework implements center-of-workspace interaction. A *zoomport* is a window that displays a 3D view of the scene. Its view is controlled by an I-FoR—any changes to a component of the I-FoR are reflected by changes in the zoomport’s view. A zoomport has interactors that handle user input to the zoomport and modify the zoomport’s I-FoR to affect changes in the view. The ZoomInteractor transforms middle-mouse-button input into translation and zooming by modifying the position and scale components of the zoomport’s I-FoR. An instance of Widgets is an Interactor that transforms certain click-and-drag operations into rotation (and zooming) by modifying the orientation (and scale) components of the zoomport’s I-FoR.

Center-of-workspace interaction has several advantages. Two of these are exceptional suitability to stereoscopic viewing and a natural region in which special rendering can be done.

Humans are used to investigating objects directly in front of them, within arms’ reach—approximately where the center of workspace is located. This location maps especially well to stereo display environments because it is also exactly where stereoscopic depth perception works best. This is in contrast with flying interfaces where the viewpoint is usually a long way from objects in the scene, resulting in a minimal stereoscopic depth effect.

Rendering of 3D data often runs into various problems like occlusion and high computation demands. Having a designated point of attention in a view makes it easier to decide how rendering might best be done. For instance, in the field of oceanography, there may be important information both on the seabed, and in the structure of the sediment layers beneath that surface. In order to focus on information beneath the surface, it is necessary to remove occluding information. The center of workspace provides a natural place to make the surface of the seabed transparent and reveal the subsurface information. Likewise, using the workspace center to determine degree of interest can inform rendering as to what is most important to render in high resolution when resolution is at a premium.

5.4 Capability 2: Object Manipulation

Direct view manipulation is sufficient for navigating through static scenes, but many of the motivating applications of this dissertation involve moving objects. Some of these objects move on their own, such as vessel objects representing real-world vessels or AUVs replaying a simulation. Other objects must be moved by the user, for example when creating a presentation about scallop populations or planning a mission to map out a particular area.

One interactive tool with which objects can be moved is the object movement tool illustrated in Figure 5.8. This tool makes it possible for the user to click and drag an object along a plane in a direct-manipulation fashion, as well as providing a way to move this plane of interaction up and down. The plane of interaction is made evident by a regular grid that indicates metric distances in the most visible power-of-2, while vertical arrows act as handles for moving the plane up and down.

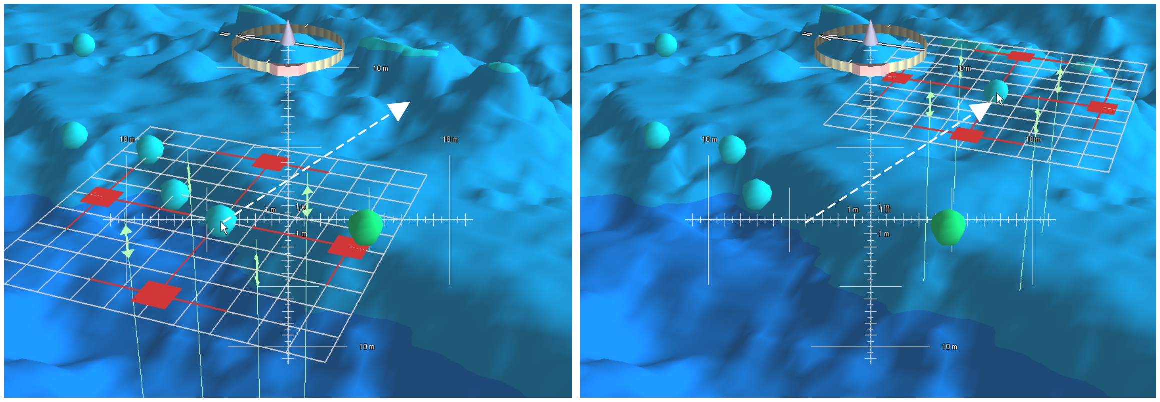


Figure 5.8: Illustration of the object movement tool in GeoZui3D.

The object movement tool is used as follows. The user first selects an editable object by clicking on it with the left mouse button. This causes the object movement tool to appear, including the regular grid, the vertical arrows, and a central control point (which may often be hidden by the object being moved). When the user clicks and drags the central control point for an object (or any part of the grid), the tool moves the object within the grid plane such that the picked point always appears under the cursor (if the cursor position can be mapped to the plane). When the user clicks and drags one of the green control arrows vertically, the tool moves the object in the *z* dimension, along with the regular grid. When the object movement tool is active, the user can also use the up and down arrow keys on the keyboard to nudge the object up and down in the *z* dimension.

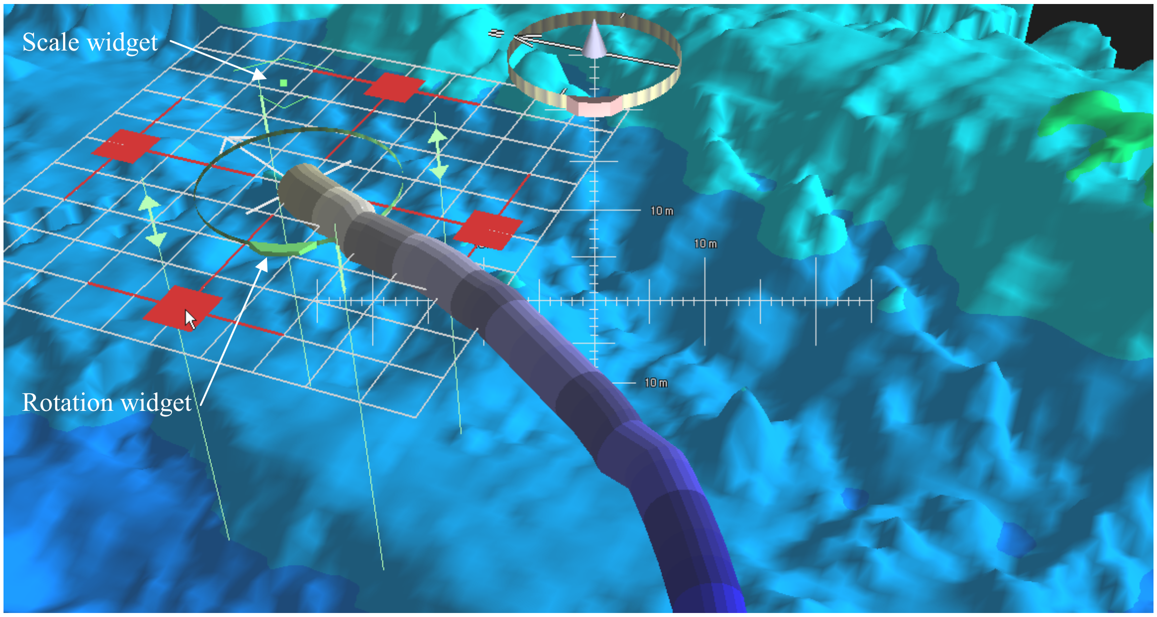


Figure 5.9: The object movement tool applied to an object that supports rotation and scale.

Additional widgets appear for modifying heading and scale, as illustrated in Figure 5.9, when the object being moved uses heading and scale. The heading rotation widget looks and works like the widget for controlling the view: the bright green button is activated by a click-and-drag using the left mouse button, and the rate of rotation depends on the horizontal distance the mouse has traveled since the mouse was clicked. The green button always faces the user so as to keep the interaction simple. The scale widget is the square green dot with a wide arrowhead above and a narrower arrowhead below. To scale the object, the user clicks on this widget and drags up to scale the object up, or drags down to scale the object down. The scale widget always appears perpendicular to the user to keep interaction simple.

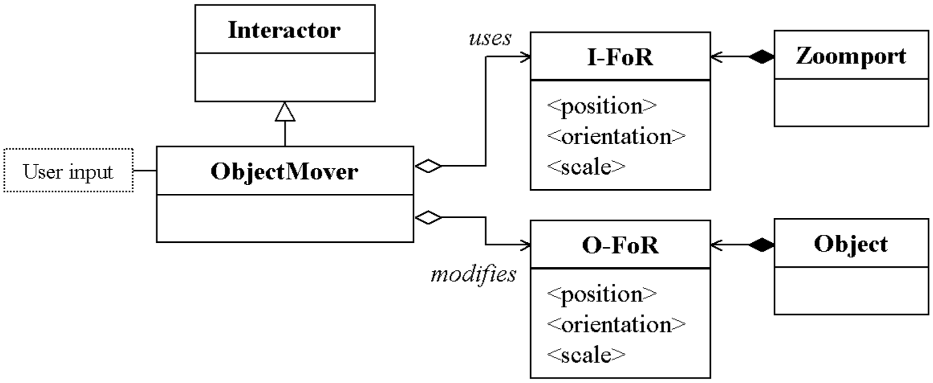


Figure 5.10: Diagram illustrating the relationships involved in using an ObjectMover to move an object.

The ObjectMover implements the object movement tool in the FoRI software framework as illustrated in Figure 5.10. The ObjectMover transforms user input into the appropriate changes in the target object's O-FoR, according to the current state of the I-FoR of the zoomport in which the interaction is occurring. Object motion parallel to the ground plane is achieved as follows. First, the cursor position is inverse-projected (using the viewing matrices associated with the zoomport I-FoR) to find the world coordinates corresponding to the hot spot of the cursor. Then, a ray is cast from the viewpoint through this cursor point, and the intersection is found with the *x-y* plane of the object O-FoR ($z = z_{object}$). When the user first clicks, this operation is performed to record the world-coordinate offset between the intersection and the object. During subsequent dragging of the mouse, this operation is performed to update the *x* and *y* components of the object's FoR, keeping the selected part of the widget under the cursor. Vertical (*z*) object motion is achieved by scaling vertical mouse input by the current scale factor in the zoomport FoR. The rotation and scale widgets affect the object O-FoR with little regard for the zoomport I-FoR.

5.5 Capability 3: Linking Multiple Windows

The third capability that frame-of-reference interaction supports is the ability to link two or more views together to work in tandem, often with one providing focus information and the other providing a contextual overview. This capability is supported in several ways, all of which concern displaying or enforcing certain geometric relationships between views. The geometric relationships involve either the 3D geometry of the scenes being viewed, the layered 2D geometry of the screen on which the views appear, or some combination of both geometries.

5.5.1 Zoomports

Under FoRI, 3D views take the form of an entity called a zoomport. A *zoomport* is a window that displays a 3D view of the scene, with the view specified by an I-FoR. Each zoomport has its own center of workspace and its own navigation widgets, and certain zoomports have decorative borders for moving and resizing the zoomport on the screen as illustrated in Figure 5.11. While the contents of a zoomport are fully 3D, zoomport themselves are treated as flat 2D windows for most operations. For instance, the user can click and drag on the title bar to move the zoomport, or click and drag on parts of the thin border to resize the zoomport. Treating zoomports as screen-bound 2D objects has the effect of assigning the focus (or context) of a zoomport to a particular region of the screen.

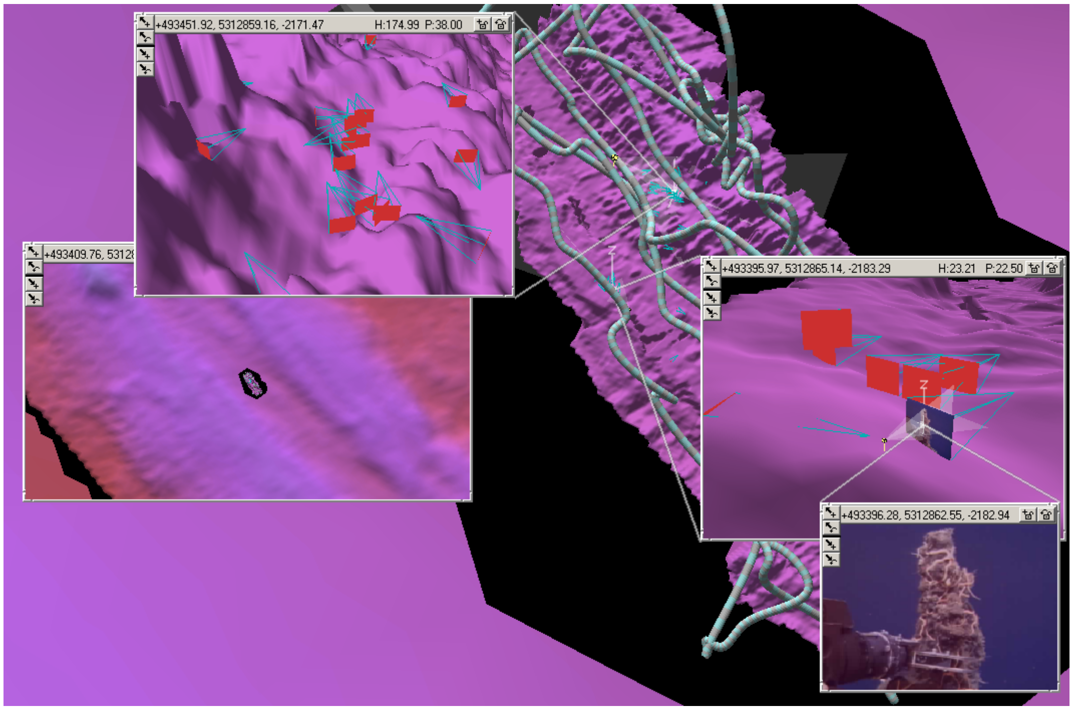


Figure 5.11: Zoomports as overlapping, flat 2D windows displaying 3D views. Navigation widgets have been hidden.

In initial design discussions for zoomports, alternative treatments of zoomports as 3D entities were considered (similar to the Worlds in Miniature approach [Stoakley et al. 1995]). If zoomports are embedded as part of the scene in another zoomport, they become rotated and scaled during navigation, making it hard to keep them visible. If they are placed in the scene space, but kept in-place with respect to the user viewing frustum, they can be more easily managed, but they can also be punctured or occluded by the scene itself. By placing zoomports in the image plane (with their clipping planes and decorative borders in the plane of the display) and treating their contents as having no depth, all of these problems are avoided. Furthermore, maintenance and implementation are simplified, and user experience from conventional 2D window management systems can transfer to a 3D FoRI environment.

5.5.2 Zoomport Proxies

Often, the most important relationship to show between two reference frames is where one is in relation to the other. FoRI represents an interaction reference frame through the use of a *zoomport proxy*, and visually connects each proxy to the I-FoR it represents through the use of *tethers*, as illustrated in Figure 5.12 and Figure 5.13. The zoomport proxy highlights the location of a zoomport's center of workspace by rendering to-scale cross hairs and a scale-independent post. The zoomport proxy also provides information about the orientation of its zoomport through the depiction of the viewpoint (yellow box) and viewing angle ("spotlight") emanating from the viewpoint. A line is dropped from the camera through the surface to provide further depth cues that aid the user in distinguishing 3D orientation. Tethers link a zoomport's closest corners (on the 2D screen) to the center of the proxy (in the 3D scene) using two lines. The tethers provide a way of quickly determining which zoomport belongs with which proxy.

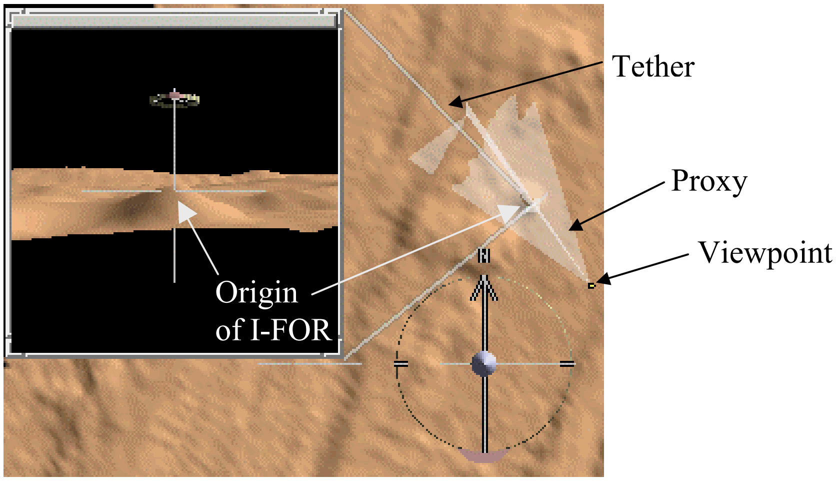


Figure 5.12: A zoomport and its proxy representation within another zoomport.

The use of proxies with tethers is inspired both from artist renditions in magazines such as National Geographic, and from the use of similar devices in the DragMag system [Ware and Lewis 1995]. One significant innovation over these 2D precursors is that these proxies and tethers work in 3D. Another innovation is the identification of the point of interest in the proxy itself, and then using this point to tie tethers to the corresponding view. In two-dimensional settings, the convention has been to draw a box representing the extent of the child 2D view, and connect the box to the child view at the corners.

5.5.3 Zoomport Hierarchy

Under FoRI, zoomports are organized in a parent-child hierarchy. A *subwindow* is a child zoomport (that may parent child zoomports of its own), and the *main zoomport* or *root zoomport* is the parent zoomport at the base of the hierarchy. Conceivably, every zoomport could have a proxy representation in every other zoomport, but this could quickly clutter the display with relationships that the user does not care about. By enforcing a hierarchy, each parent zoomport provides a common context for all of its children, and at most one proxy is displayed for any given zoomport. In its simplest form, this hierarchy naturally designates the root zoomport as the common context in which all child windows display their proxies. Figure 5.13 illustrates such a situation, where a zoomport has three children, two of which are minimized.

FoRI provides a mechanism to change the focus of a zoomport within the context of its parent: the user can click on the child's zoomport proxy and drag it along any visible surface in the parent zoomport. As the child's proxy is dragged through the parent zoomport, the child's center of workspace correspondingly animates. Rather than staying in an *x-y* plane (as with the manipulation of objects), the child's center of workspace is moved to the intersection of a ray from the virtual camera through the cursor tip with the nearest surface along that ray. In this way, the center of workspace as represented by the proxy always appears at the surface immediately under the cursor. Such direct manipulation of the zoomport proxy provides an affordance for dragging the focus of attention for the child zoomport in a way that is metaphorically like telling the computer, "I want *this* zoomport to look at *that*."

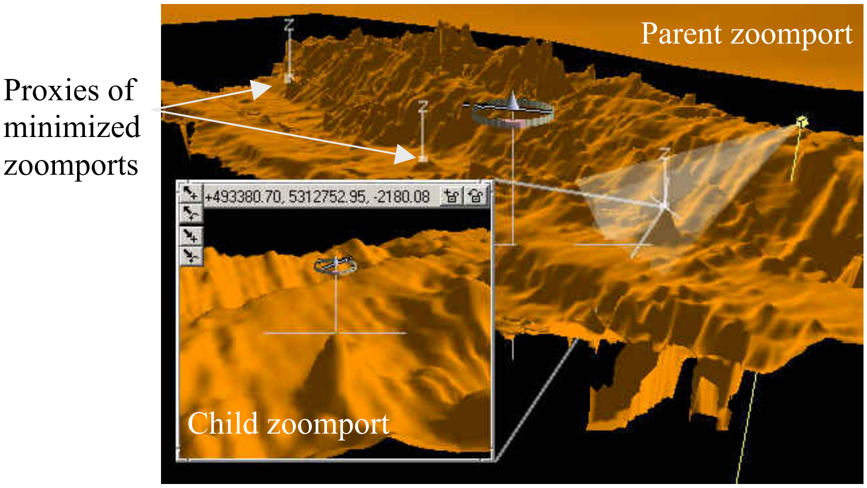


Figure 5.13: A zoomport, with proxy and tethers linking its center of workspace to its parent's view. Two minimized zoomports are shown as well.

5.5.4 Zoomports in the FoRI Software Framework

Figure 5.14 illustrates the way in which zoomports, zoomport proxies, and the zoomport hierarchy are implemented in the FoRI software framework. Each Zoomport has its own I-FoR, and has references to its parent and children. Each child Zoomport (subwindow) also has a ZoomportProxy, which is an Interactor that displays a zoomport proxy and implements the dragging behavior for that proxy. When the user clicks and drags the proxy representation, the depth buffer is checked under the mouse cursor and is reverse-projected using the parent's I-FoR to find the 3D world coordinates of the surface point under the cursor. The ZoomportProxy then sets the child's I-FoR to these coordinates, causing the child zoomport to update its view. If the depth buffer is empty, dragging occurs such that it leaves the child's scene-*z* coordinate alone, and movement occurs within an *x*-*y* plane in a way similar to object movement.

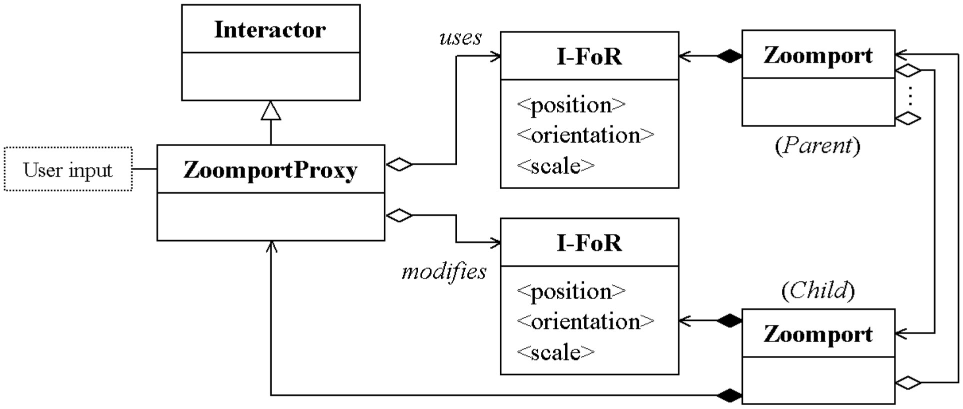


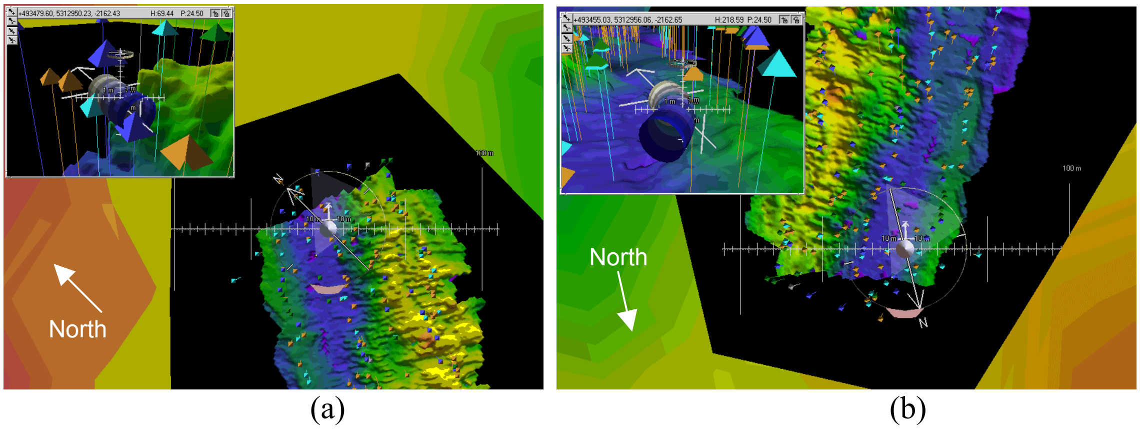
Figure 5.14: Diagram illustrating the implementation of direct manipulation of a zoomport center-of-workspace by dragging its proxy. The hierarchical relationship between parent and child zoomports is also illustrated.

5.6 Capability 4: View Coupling

Frame-of-reference interaction supports the ability to couple a view to a moving object or to another view. Such couplings are useful when the focus of attention is not a static object, but is either a moving object or a changing view. For example, suppose a user wishes to see a detailed view of a remotely operated vehicle in the context of an overview map. As demonstrated by the child zoomport of Figure 5.15, a "wingman" view or an "over-the-shoulder" view could be set up by coupling a zoomport to the vehicle such that the view maintains its position relative to the vehicles reference frame. Furthermore, the overview zoomport could be coupled to the child zoomport such that it acts as a forward-up map, as illustrated by the parent zoomport of Figure 5.15.

The mechanism FoRI supplies to enable such behaviors is *frame-of-reference coupling*. A FoR coupling is a mathematical constraint on one or more components of a

FoR, such as position or heading, such that a change in one frame of reference induces a change in the other. Users can couple zoomports to objects to set up couplings like an "over-the-shoulder" view by using the middle mouse button to click on the moving object. This causes the zoomport FoR to become coupled to the object's FoR in a particular manner described shortly. Without further interaction, the zoomport rotates as the object rotates and its center of workspace remains fixed in the reference frame of the moving object. Interaction with zoomport widgets can rotate the view around the moving object as if the object was static, and selection of other points on the object with the middle mouse button provide a sense of navigating with respect to the moving object. To return to a static view or transfer to another moving object, it is only necessary for the user to click on the appropriate object with the middle mouse button. Thus, navigation with the middle mouse button always places the center of workspace into the reference frame of the selected focus object.



(a) (b)

Figure 5.15: The overview zoomport is coupled with the inset zoomport to yield a forward-up map view, while the inset zoomport is coupled to a moving vehicle using a localized coupling and a relative coupling in heading to implement a tethered view. (a) and (b) show how both zoomports translate and rotate as the vehicle moves and turns.

Users can couple a zoomport to another zoomport in order to allow the two zoomports to be used in tandem, such as by making one a forward-up map. Under FoRI, the interface only supports couplings between a child zoomport and its parent. Two buttons are provided in the upper-right-hand corner of subwindows to couple the child and parent in either position or heading attributes. Depressing both of these buttons instantiates a coupling between the child and parent such that any change in position or heading in either zoomport causes the same change to occur in the other. If their headings are aligned, and the parent is pitched to be looking straight down, the parent acts as an overview forward-up map for the child.

5.6.1 Couplings in the FoRI Software Architecture

Figure 5.16 illustrates broadly the way in which the FoRI software architecture implements FoR couplings. Each Zoomport has two Couplings reserved for the interaction behaviors just described. The first is a Coupling to the Zoomport's parent. This Coupling is inactive until the user presses one or both of the buttons in the top right corner of the zoomport, at which point the appropriate geometric coupling or couplings are enabled. The second is a Coupling intended for Objects. Whenever the middle mouse button is pressed on a moving object, this Coupling is detached from any previous Object and is attached to the selected Object and activated. Conversely, when the middle mouse button is pressed on a static object or empty space, this Coupling is detached from any previous Object and is deactivated.

Couplings as implemented in the FoRI software architecture do not discriminate between I-FoRs and O-FoRs, making it possible to update objects as well when a zoomport or another object moves. For instance, an object could be coupled to a zoomport such that it always appears at the workspace center. This might be useful in games or interactive exhibits, where the widgets would be undesirable, but some proxy for user control is needed. An object could also be coupled to another object, for instance a vector might be attached to a ship to indicate the most likely course a few minutes out. However, these capabilities are byproducts of the bi-directional nature of couplings—they are not core capabilities of frame-of-reference interaction, and they therefore are not discussed further.

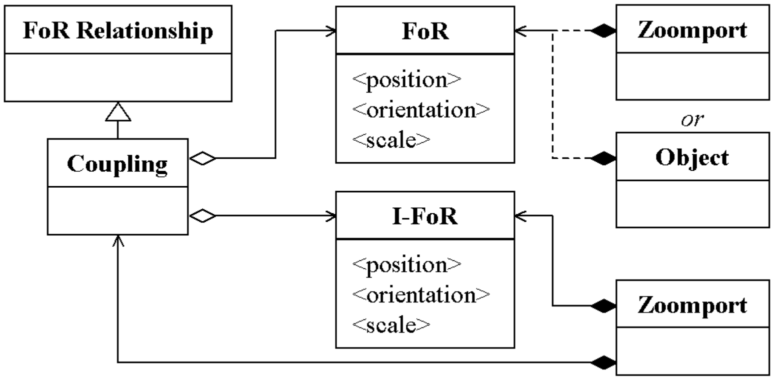


Figure 5.16: Diagram illustrating the implementation of coupling a zoomport to an object or another zoomport.

5.6.2 Types of Coupling

To support the coupling behaviors available in FoRI, as well as many other possibilities, FoRI defines three basic types of coupling: absolute, relative, and localized. An instance of Coupling is capable of supporting multiple such couplings, although some combinations of are prohibited because they conflict.

Absolute coupling is the simplest. If an absolute coupling exists between reference frames *P* and *Q* on an attribute *a*, then whenever *P.a* changes value, *Q.a* is updated to have the same value. For instance, for a coupling on heading between two zoomports, if the heading of a zoomport characterized by *P* changes to 45˚, the heading of the zoomport associated with *Q* is also set to 45˚. Figure 5.17(a) shows a zoomport (*Q*) and object (*P*) at the initialization of a coupling, while Figure 5.17(b) shows what would happen if no coupling were enabled and the object moved. The effect of an absolute coupling on position between the zoomport and object is illustrated in Figure 5.17(c).

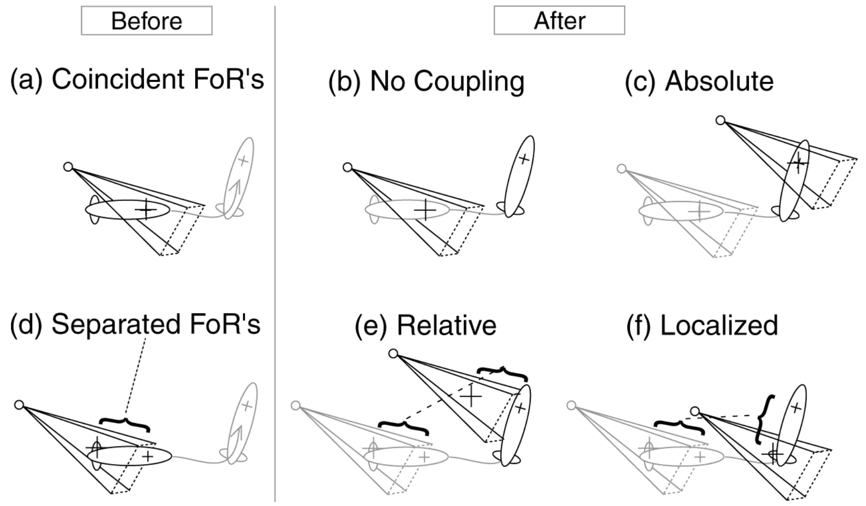


Figure 5.17: Illustration of various couplings of position between zoomport and object in the position attribute. In the first row, the zoomport and object share the same position to begin with (a). After the object moves, the effects of no coupling (b) and an absolute coupling (c) are shown. In the second row, the zoomport is positioned behind and slightly above the object initially (d). After the object moves, the effects of relative coupling (e) and localized coupling (f) are shown.

Relative coupling is more general than absolute coupling. If a relative coupling exists between reference frames *P* and *Q* on an attribute *a*, then whenever *P.a* changes by some amount δ, *Q.a* changes by δ as well. For instance, consider what happens if the heading of the zoomport characterized by *P* starts at 45˚ and the zoomport associated with *Q* starts at 130˚. If the heading of *P* changes to 25˚ (δ = -20˚), then the heading of *Q* changes to 110˚. Figure 5.17(e) shows the effect of a relative position coupling between a zoomport (*Q*) and object (*P*), after movement of the object from its initial position as seen in Figure 5.17(d). To avoid accumulation of errors, the implementation maintains the δ present at the instantiation of the coupling, rather than calculating δ at each change.

Whereas absolute and relative couplings operate over a single attribute and are valid for any single attribute of a reference frame (in position, orientation, or scale), a localized coupling operates over all attributes. The purpose of a localized coupling is to "fix" one reference frame with respect to another, just as would occur if one were to rigidly attach a camera to a moving object. Localized coupling works as follows. If a localized coupling exists between reference frames *P* and *Q*, then whenever *P* changes in position, orientation, or scale, *Q* is updated so that its position remains unchanged with respect to *P*. More formally, $Q.position = P.position + \delta_{orientation} \bullet \delta_{scale} \bullet \delta_{position}$, where each δ_a is the original difference between *P* and *Q* in attribute *a*[1]. The effect of a localized coupling is to "fix" the position of *Q* into *P*, as if *Q* were rigidly attached to the origin of *P*. Figure 5.17(f) shows the effect of a localized coupling between a zoomport (*Q*) and object (*P*), after movement of the object from its initial position as seen in Figure 5.17(d). The position of the zoomport is always in the same place on the tail of the object, regardless of how the object moves (or how it is scaled).

5.6.3 Practical Uses of Couplings

Couplings are generally most useful in combinations. For instance, when the user clicks the middle mouse button on an object, the coupling behavior triggered is established using a localized coupling between the zoomport and the object, combined with a relative coupling in orientation (illustrated in the child zoomport of Figure 5.15). The forward-up overview map behavior (also illustrated in Figure 5.15) is achieved by absolute-coupling two zoomports in position and heading attributes. As another example, a magnified view can be created by absolute-coupling two zoomports in position and

[1] When dealing with orientation in terms other than quaternions, the formula would be more properly written as $Q.position = P.position + P.orientation \bullet \delta_{scale} \bullet (\delta_{orientation})^{-1} \bullet \delta_{position}$.

orientation, and relative-coupling them in scale. Such a coupling arrangement would provide a way of rapidly operating at two disparate scales (see Figure 5.18).

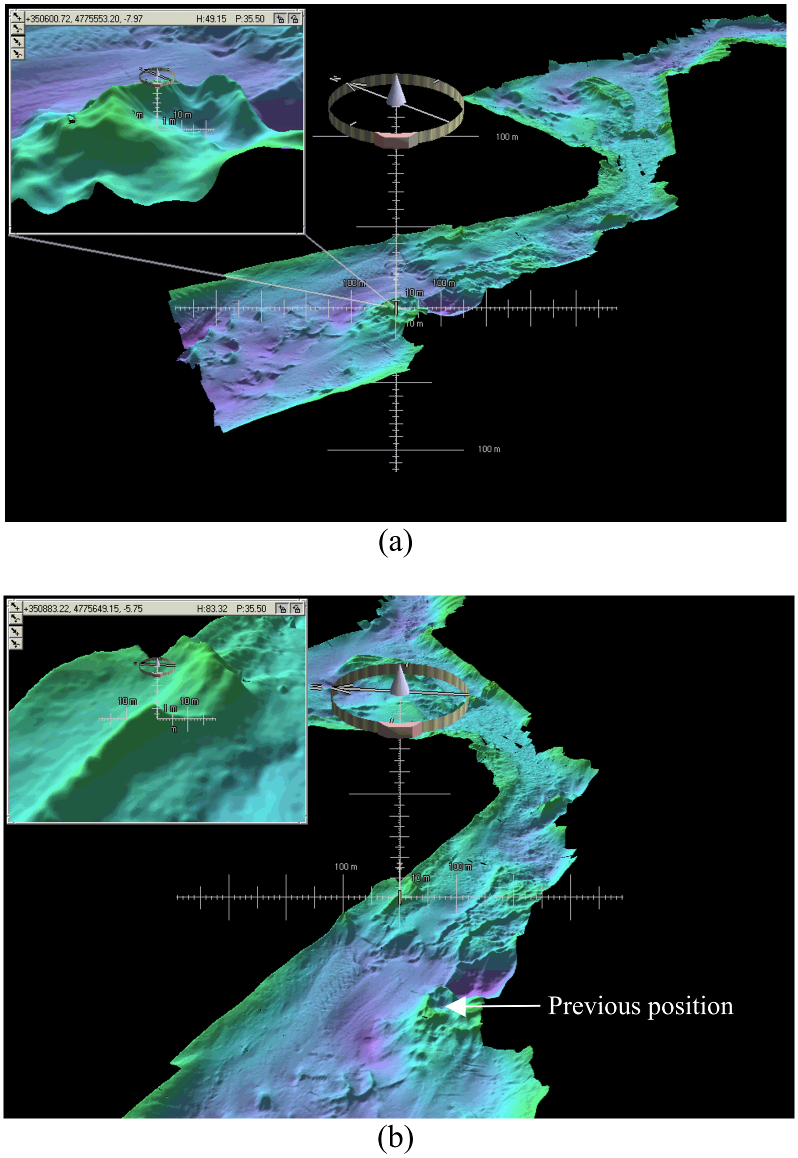


Figure 5.18: Zoomports coupled in a magnified view arrangement. Any movement in one is matched in the other, as in the translation, rotation, and scaling from (a) to (b), allowing the inset zoomport to act as a magnifying glass for the center of workspace in the main zoomport.

Careful suspension of couplings is useful in giving the user the feeling that they can navigate on a moving object just as they would navigate through the static virtual world. The localized coupling between a zoomport and object is suspended at each animation frame during active user navigation (so that new δ's are recorded). Likewise, the relative coupling in orientation is suspended at each animation frame during which the user is interacting with the zoomport's heading widget.

5.6.4 Identifying the Most Useful Coupling Combinations

An informal assessment of various combinations of couplings between a zoomport and its parent was done, leading to the set of buttons on the zoomport borders for instantiating couplings. Originally, buttons existed for instantiating relative couplings in translation, heading, pitch, and scale. In addition, buttons existed for setting each attribute in the child to be equal to the value in the parent and vice versa. These buttons made it possible to simulate an absolute coupling (by setting the values equal before instantiating the relative coupling). Various combinations of the couplings were tried in different situations to see how they might aid the user. The situations included general exploration of a multiscale scene, path identification in simple 3D entangled paths, monitoring of a vehicle in transit, and target identification in a multiscale scene.

One result that emerged from this informal experimentation indicated that azimuth coupling is more useful than elevation coupling—there was no situation that required two views to move together in elevation. Other results indicated that absolute couplings are useful in both heading and position, especially for providing forward-up views. Absolute couplings in position appear to be useful because they provide a way of aligning two views at different scales. Relative couplings in position, on the other hand, are not as useful because the zoomports are not focused on the same item, and the directional distance between the FoR's does not generally have any useful meaning. Absolute couplings in heading appear to be useful because they provide a way of maintaining orientation when the zoomports are at different scales. The uses of relative couplings in heading are limited to things like investigating an item from two angles at once or maintaining multiple views from a single vantage point. Coupling in scale did not seem to provide general benefit, probably because our scenarios tended to have important information at fixed scales—a relative coupling in scale would only be useful if levels of detail changed from place to place but were always at the same relative scale.

Localized coupling between zoomports was also investigated. Localized coupling causes a child zoomport's proxy to always appear in the same location in the parent, and makes it possible to rotate the parent about the child's center of workspace. However, the child's zoomport proxy tends to get obscured during navigation in the parent zoomport, and there are no situations in which the special rotation capability makes sense. Localized coupling was abandoned in favor of absolute coupling in position and heading.

The selection of buttons adorning zoomport borders reflects the results of the informal assessment, as evidenced in the presence of a button for relative-coupling in heading and one for absolute-coupling in position (upper right corner of zoomports in Figure 5.13), supplemented by position-setting and heading-setting buttons (upper left corner). In general, necessity for other coupling combinations would appear to be quite task specific, useful only under very specialized conditions.

5.7 <u>Capability 5: Coupling to Abstract Objects</u>

Sometimes it is useful to couple a view to something that is neither an individual object nor another view. For example, a user may want to monitor the extent of a fleet of vessels for their progress in a survey mission or a search-and-rescue mission. Alternatively, a user may wish to be alerted to potential collision conditions in the management of a busy port, and have views update dynamically to monitor close calls that may require intervention. It is these sorts of situations that have motivated the development of frame-of-reference operations.

A *frame of reference operation* (FoR-op) is a generalization of a reference-frame coupling. A FoR-op is an abstract object that maintains a reference frame that geometrically aggregates position, orientation, and/or scale information from one or more other frames of reference. The reference frame that the FoR-op maintains is referred to as the *resultant* reference frame, and the frames of reference that it aggregates are referred to as *operand* reference frames. The resultant FoR is an O-FoR that represents a focus of attention related to some aspect of the collection of objects, rather than any individual object. The resultant FoR can be coupled to, or it can alternatively be used as an operand of another FoR-op, just like any other O-FoR.

5.7.1 Example FoR-Ops

Two examples of FoR-ops from GeoZui3D help to illustrate the concept of FoR-ops. The first is the *aggregate-overview* FoR-op, and the second is the *closest-proximity* FoR-op.

The aggregate-overview FoR-op maintains its resultant FoR so that its origin is at the center of its operand FoRs, its scale corresponds to the furthest extents of its operands, and its heading is the average of its operands' headings. A zoomport can be coupled to this resultant (in position and scale attributes), making it possible to monitor a group of vehicles and follow them no matter where they go, or how far apart they stray from each other. For example, the zoomport shown in the lower right corner of Figure 5.19 illustrates such a coupling where the aggregate-overview has five operands corresponding to the five vehicles. The zoomport widens its view as the vehicles spread apart, and zooms back in as vehicles move closer together. If a zoomport coupling in orientation is enabled, the heading of the zoomport points in the direction that is the average of the directions in which the all the vehicles are heading.

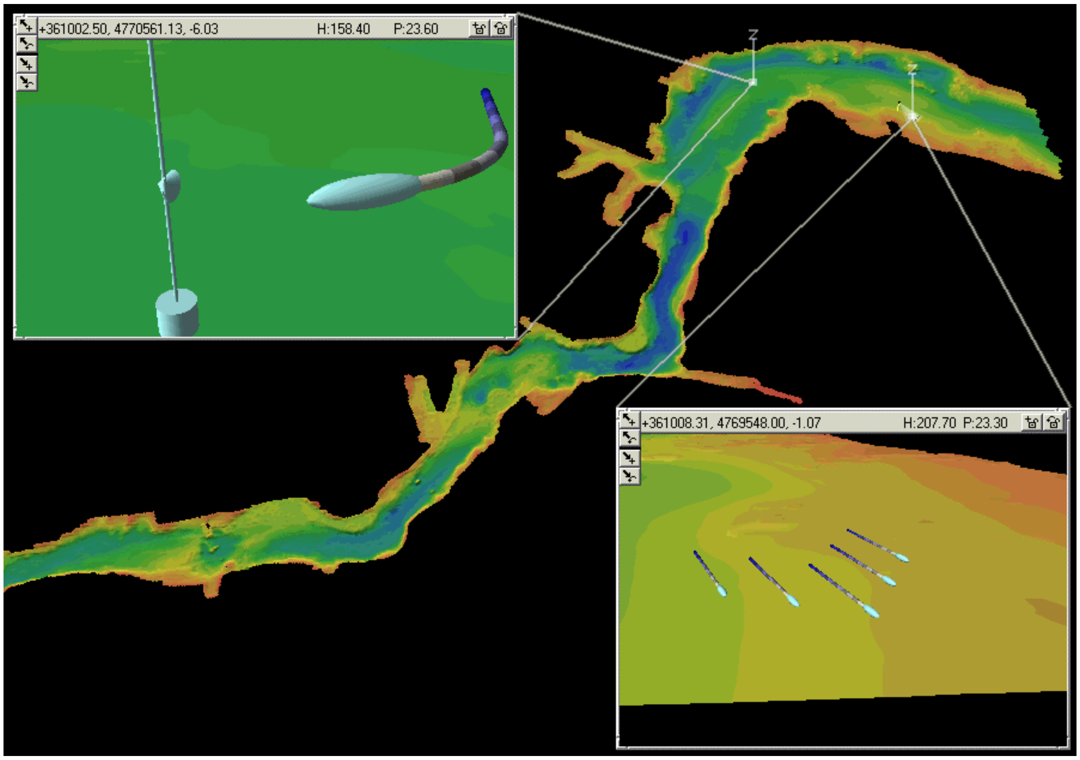


Figure 5.19: Zoomport coupled to a closest-proximity FoR-op (top-left) and another coupled to an aggregate-overview FoR-op (bottom-right).

The closest-proximity FoR-op tracks its operands to find the two closest to each other at each moment. It maintains its resultant FoR so that its origin is at the center of the two closest operand FoRs and its scale corresponds to the extent of these operands. The closest-proximity FoR-op generates system events when this extent crosses certain thresholds. A zoomport can be coupled to a closest-proximity resultant (in position and scale attributes), making it possible to monitor the two closest operand objects at any given instant. For example, consider a situation in which two fleets are passing near each other and two of the member vehicles are coming dangerously close, as shown in

Figure 5.22. The closest-proximity FoR-op constantly updates the view to show the two closest vehicles in the group and generates warning events when certain proximity thresholds are crossed. When the vehicles get too close, the closest-proximity FoR-op alerts the user and tracks the situation. The zoomport in the upper left corner of Figure 5.19 illustrates another example in which the operands include all of the vehicles plus docking stations. This zoomport allows docking operations to be monitored, regardless of the number of docking stations (assuming no simultaneous docking operations).

5.7.2 Implementation of FoR-ops in the FoRI Software Architecture

Figure 5.20 illustrates how FoR-ops are implemented in the FoRI software framework. A FoR-Op can be given responsibility for monitoring any number of operand FoRs. The FoR-Op updates its resultant O-FoR whenever any one of these operand FoRs changes. Most often, a zoomport is coupled to the resultant O-FoR so that the FoR-op can control its view. However, Objects can also be coupled to resultants.

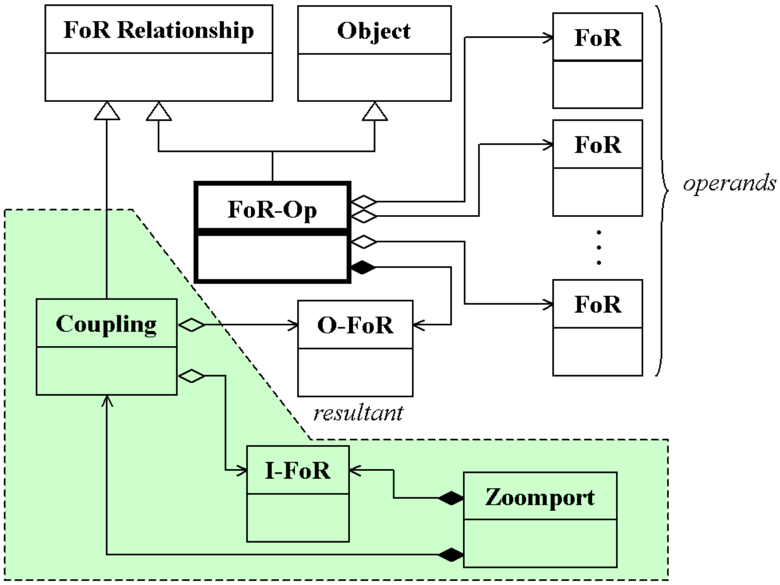


Figure 5.20: Diagram of a zoomport coupled to a FoR-op. The part of the diagram inside the colored region represents what is required for coupling a zoomport to a FoR-op, while everything outside the region represents what is required for the FoR-op.

Figure 5.21 illustrates how the aggregate-overview FoR-op is implemented. The resultant of the overview FoR-op contains all the information needed to control a view intended to monitor the operand FoRs as a group, namely the center and scale of a bounding box and average direction. Every time any operand position changes, a bounding rectangular solid is generated using the minimum and maximum extents in world *x*, *y*, and *z* coordinates. The origin of the resultant is then updated so that it is situated at the center of this solid. The longest dimension of the rectangular solid is used to indicate the scale of the aggregate-overview FoR-op; a constant factor times the longest dimension is stored in the scale component of the resultant FoR. The average orientation of the operands is stored as the orientation of the resultant, as well.

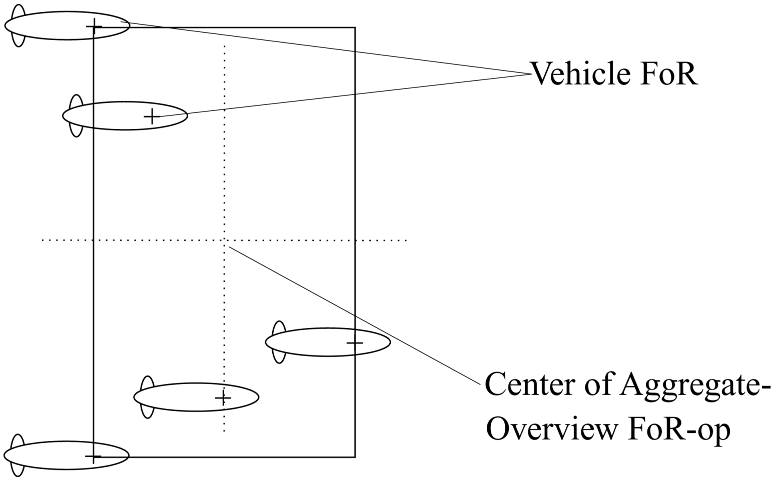


Figure 5.21: 2-dimensional representation of the geometric operations involved in the aggregate overview FoR-op, as applied to five vehicles.

Figure 5.22 illustrates how the closest-proximity FoR-op is implemented. The resultant of the closest-proximity FoR-op summarizes all the information needed to control a view intended to monitor the two closest operand reference frames at a given time and produce alerts when two operand FoRs get within a certain range. Every time any operand position changes, the closest-proximity FoR-op finds the two closest operand FoRs (Euclidean distance). Then the minimum and maximum extents are calculated in scene *x*, *y*, and *z* coordinates, generating a bounding box for these two closest FoRs. The origin of the resultant is situated at the center of this bonding box and the scale is set to the largest dimension of the bounding box, just as the aggregate-overview does. In addition the closest-proximity FoR-op checks the Euclidean distance between the two closest FoRs and compares them against danger threshold parameters. If the distance crosses a danger threshold, the FoR-op generates a system event that can be used to alert the user to the condition.

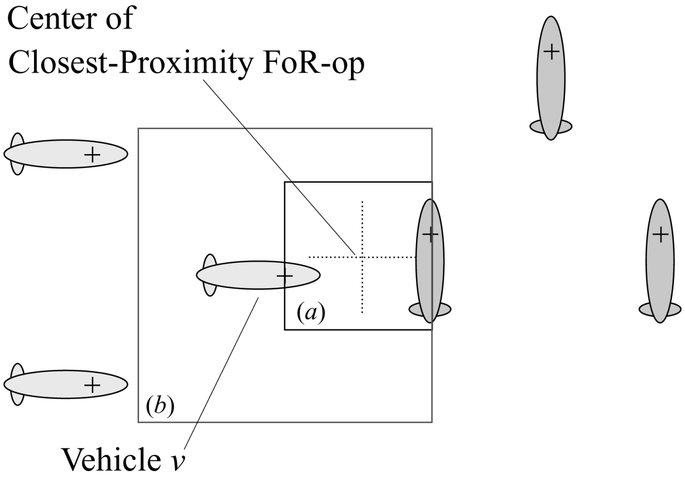


Figure 5.22: 2-dimensional representation of the closest-proximity FoR-op, applied to six vehicles. (a) Extent of zoomport coupled directly to the resultant (absolute coupling). (b) Extent of zoomport position-coupled to vehicle *v* and scale-coupled (relative coupling) to the resultant.

The actual coupling of a zoomport to a closest-proximity FoR-op can be done a couple of different ways depending on the task. If a zoomport is coupled in both position and scale attributes to the closest-proximity FoR-op, the zoomport will always be centered between the two closest objects, usually in empty space as illustrated in Figure 5.22(a). This may be ideal when the user can monitor the situation, but cannot take corrective action. However, if corrective action is possible, it would be better to couple the zoomport to the closest-proximity FoR-op in scale only, and allow the zoomport to be coupled to one of the vehicles in heading and position. This situation is illustrated in Figure 5.22(b), where vehicle *v* is the target of the coupling.

5.7.3 Future Directions for FoR-ops

The guiding principle in the design and use of a FoR-op is that it should directly map to a higher-level relationship between objects that the user is likely to have interest in. When used in this way, the FoR-op resultant acts as a sort of "chunk", aggregating the information and relationships of the operands into a single focus of interest. Because the FoR-op's resultant is like any other FoR, it can also be the target of even higher-level relationships represented by FoR-ops. Ideally, each level of FoR-op should reduce cognitive load on the user by automatically monitoring important relationships and alerting the user only when specific conditions requiring user attention are met.

As an example of using FoR-ops in this way, consider a situation in which an unexpected event occurs during a semi-automated survey mission. Cooperative fleets of autonomous vehicles are already being tracked by aggregate-overview FoR-ops, as they perform routine mapping missions in unmapped areas. A family of whales has also had tracking devices planted on their bodies. A biologist that has been monitoring the whales detects strange behavior in their movement and asks the survey scientist to send a team of vehicles over to investigate. The survey scientist could create an aggregate-overview FoR-op to track the family of whales, then use a specialized proximity-highlighter FoR-op to automatically detect and track the fleet FoR-op closest to the whale family FoR-op. If the closest fleet was too busy or there was an obstacle between the fleet and the whale family, it could be dropped from consideration and the next closest fleet would automatically be detected and tracked.

A possible extension to FoR-ops would be to add a dimension of time. Such an extension would be able to monitor geometric behavior over time, such as relative velocities and accelerations, as well as angular velocity. If combined with non-geometric information, momentum and forces among objects could be tracked. Such extensions would also allow a FoR-op to implement “dynamic tethers,” implemented by Wang and Milgram [2002], which treat the viewpoint as if it were attached to the object of interest with a contraption built of springs. One advantage the FoR-op dynamic tether would have over the original is that a zoomport need not be completely constrained to the resulting position and orientation. By using a relative coupling in attaching a zoomport to the resultant of such a FoR-op, the user can modify the view using some of the normal navigation techniques, looking more left or right, or moving higher or closer in, for example.

5.8 <u>Conclusion</u>

Frame-of-reference interaction provides a solution to the focus-in-context problem for multiscale 3D environments by providing the tools necessary to efficiently assign focus information to parts of the display, as well as the linking mechanisms necessary to relate focus information to the appropriate context. The various core capabilities of FoRI work with each other to create a consistent interface with the following properties:

1. *Selection of focus just under the cursor*. Whether it is in navigation, object movement, or zoomport proxy movement, selection of the current item of interest occurs with respect to the surface beneath the cursor.

2. *Display of focus at a center of workspace.* During navigation, the focus is brought to the center of workspace in the zoomport where interaction is occurring. During zoomport proxy movement, the focus is brought to the center of workspace of the zoomport associated with the proxy.
3. *Assignment of display resources to a semantic focus.* Zoomports can not only have static locations at their workspace centers, but also can be coupled to moving objects, and (through FoR-ops) coupled to groups of objects and even particular semantically important properties of groups of objects.
4. *Linkage of focus to a common context.* Focus and context can be linked visually through zoomport proxies and tethers, behaviorally through couplings between zoomports and between zoomports and FoR-ops, and logically through the zoomport hierarchy.

On top of these properties, FoRI also combines zooming with multiple windows in an effective way, works well with stereoscopic views, and leverages the common user experience-base with 2D window management interaction concepts. The chapters that follow provide empirical evidence and theoretical explanations for how FoRI environments can be employed to improve user efficiency and/or accuracy.

CHAPTER 6

EVALUATION OF METHODS FOR LINKING 3D VIEWS[1]

In the previous chapter, several devices for linking focus and context in multiple views were discussed. In this chapter, three of these devices (directional proxies, tethers, and absolute couplings in orientation and position) are investigated in order to determine their effectiveness in helping users to make faster or more reliable decisions. The investigation takes the form of two experiments based on a new task called the multi-perspective identification task.

6.1 The Multi-Perspective Identification Task

When performing a task such as guiding a remotely operated underwater vehicle to a new point of interest, it is common to combine information from several sources (such as sonar maps of the area, a video feed from the vehicle, and estimates of vehicle position and orientation). Often it is necessary to see aspects of the same geographic space at different scales, for example, a local view of a vehicle together with a contextual overview. This is the motivation behind the multi-perspective identification task.

The multi-perspective identification task (MP-ID) requires a subject to combine the information from two or more different perspectives to identify a target object. More specifically, a subject must select from among identical-looking objects on an overview

[1] The contents of this chapter have also been published in Plumlee and Ware [2003a] in a modified form.

map given distinguishing information in a local-perspective. This task might be considered the inverse of a virtual-world search such as Darken and Cevik's task [1999]: rather than using an indication on the map display to guide oneself to the target, one must indicate on the map display where one sees the target from a local perspective. In other words, MP-ID requires the user to use local view information to interpret a global map. This contrasts with prior studies that required users to use map information to guide local actions.

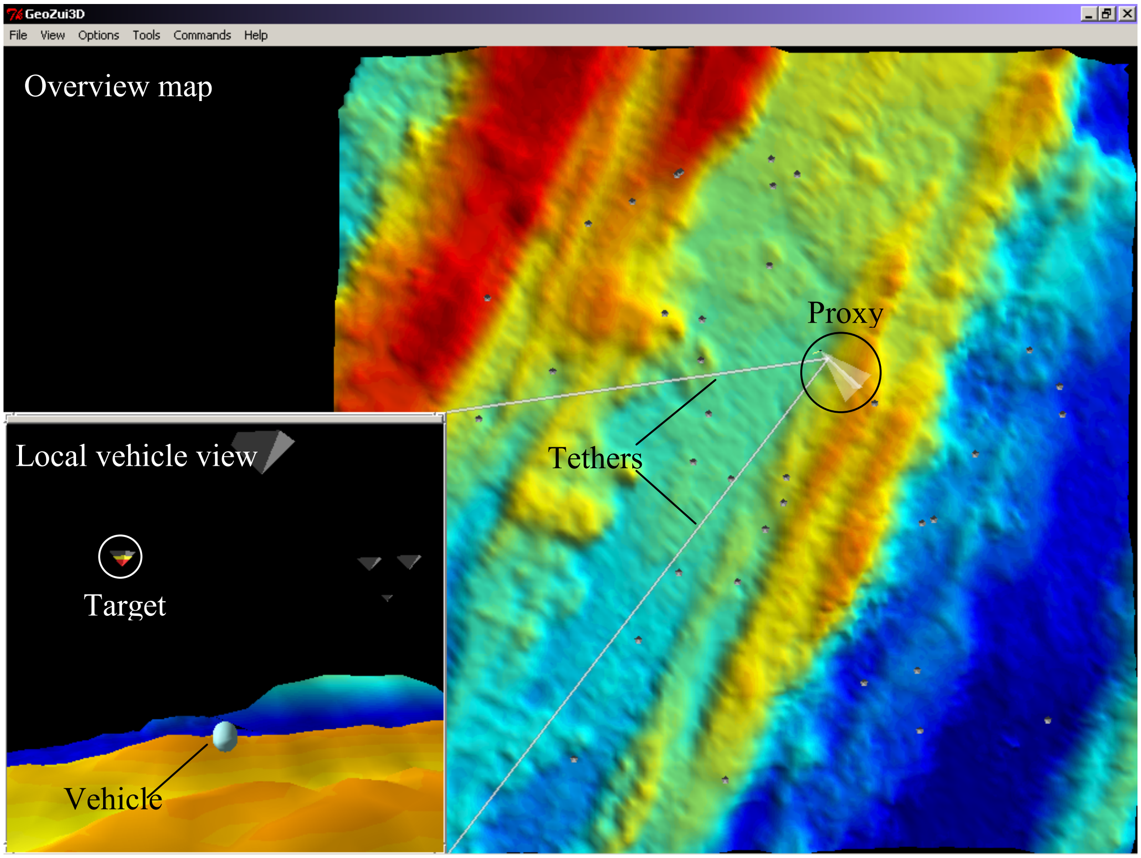


Figure 6.1: Example of the display presented to subjects in Experiment 1.

For the particular instance of MP-ID used in the experiments in this chapter, consider a situation in which one or more autonomous vehicles are exploring an undersea landscape. It is the job of an operator to identify the position of certain objects spotted by one of the vehicles. As illustrated in Figure 6.1, the operator is given an overview map in one window and in sub-windows is given local forward-looking views from just above and behind each vehicle. In this instance, the multi-perspective identification task consists of monitoring one or more local, vehicle-centric views for a distinctive target within a field of distracters.

6.2 Common Experimental Method

Two experiments were designed around the MP-ID task. The goal of the experiments was to determine how effective various linking mechanisms were in helping subjects to complete the task faster or more accurately. In both experiments, subjects were presented with a display similar to that shown in Figure 6.1 (Experiment 1 used one local view, while Experiment 2 used two). Each local view smoothly[1] followed a small vehicle as it wandered randomly through the environment. Once the target appeared in a local view, the operator identified it in the overview map by clicking on its representation with the mouse. Subjects were told that they could make their decisions based on any available information, including changes in vehicle heading, distracter layout, and surface cues such as form and color.

The design for the two experiments had several aspects in common, including initial trial conditions, two of the linking aids, the apparatus, and the methods of measurement.

6.2.1 Initial Trial Conditions

Each trial started with a new random layout of 35 distracters and a new random path for each vehicle to follow. The target was placed along the line of sight of one of

[1] The vehicle maintained a heading tangential to a spline, as did the "camera" for the local view. The effect was that the vehicle always appeared centered toward the bottom of the local view.

the latter path segments such that it would be encountered from a north-looking (< 90˚ from north) or south-looking (< 90˚ from south) vantage point. The path always started near the center of the overview and continued in a constrained random manner. The path was always a cubic spline interpolating 8 points. Each successive point was generated by selecting a random distance and a random heading relative to the previous point, as well as a random height above the surface. The distance was constrained to be between roughly 2-5% of the width of the screen from its predecessor, while the direction was constrained such that each base segment was within 90˚ degrees of its predecessor. If the path did not provide a way for the target to be encountered in the chosen direction for the trial, the path was regenerated from scratch.

6.2.2 Linking Aids

The following conditions were common to both experiments:

- Proxy vs. no proxy
- Tethers vs. no tethers

The third linking method, using track-up orientation coupling, only applied in the first experiment.

In the overview map, a small (roughly 5-pixel-wide), semitransparent box appeared at the position of each vehicle. When tethers were present, they appeared as semi-transparent lines as shown in Figure 6.1 and Figure 6.2(b), connecting two corners of a local view to the center of the corresponding small box (the box was not readily visible with proxies showing). When proxies were present, they appeared as semi-transparent triangles as in Figure 6.1 and Figure 6.2(b). A small dot at the apex of the triangle represented the viewpoint for its corresponding local view.

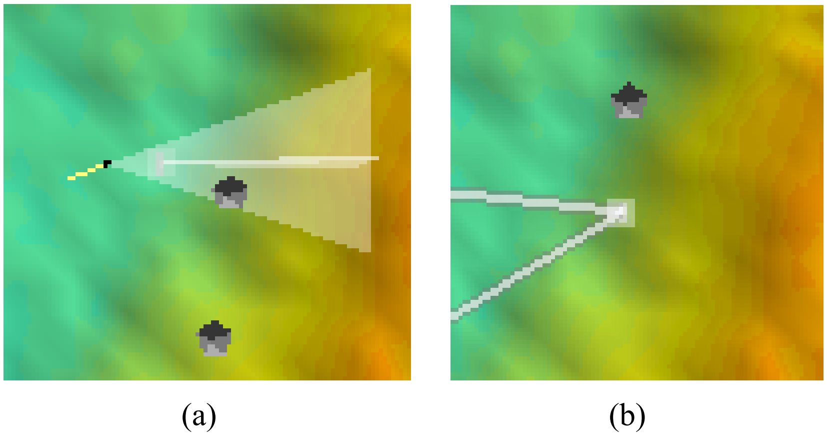


Figure 6.2: Enlarged view of proxies: (a) directional proxy with semitransparent triangles emanating from a small black dot; (b) semi-transparent box indicating the position of the vehicle, with tethers leading to the local view.

6.2.3 Apparatus

Both experiments were run on a Windows 2000 (Professional) system configured with a Pentium 4 processor, a Wildcat II 5110 graphics card, and a 19" monitor running at a resolution of 1024x768 pixels. The animation rate of each experiment was roughly 30 frames per second. Subjects were provided a standard Microsoft mouse for controlling the on-screen cursor. All references in this chapter to a click or a selection using the mouse refer to the clicking of the left mouse button only (input from the other two buttons were ignored by the experiment software).

6.2.4 Measurements

In both experiments, decision time and errors were measured. Decision time was measured as the amount of time that elapsed between when the target was first visible in a local view and when the subject moved the mouse cursor out of a local view to make a selection on the overview map[1]. Errors were recorded whenever the subject made an

[1] If the target was visible in a local view for less than a second, the timer was reset under the assumption that the target disappeared too quickly for the subject to make use of it. In practice, this rarely happened.

incorrect selection, and the position of that selection was recorded for later determination of the magnitude of the error (in angular degrees).

It was possible for a subject to simply timeout by not responding to any target. If the target appeared for less than a second, the trial was repeated. Otherwise, the trial was recorded as an error (without a position), and the next trial was begun. Also, if the subject made an incorrect selection before the target ever appeared, the trial was repeated.

One additional error condition was the result of an aspect of the interface design. In the orientation-coupled conditions, it was necessary to stop all vehicle movement when the subject was ready to make a selection so that they did not have to chase the desired point with the mouse cursor. Movement was therefore stopped whenever the subject placed the cursor into the overview region, and subjects were instructed not to move the mouse cursor out of a local view until they were ready to make a selection. If the subject spent more than 5 seconds making a selection, or if the subject moved the cursor in and out of a local view too many times, the trial was recorded as an error (without a position), and the next trial was begun.

At the end of each experiment, each subject was asked to fill out a short questionnaire. The questionnaire asked subjects to rate the usefulness of each linking aid encountered in the experiment. The scale went from 1 (counterproductive) to 5 (not useful) to 10 (extremely useful). The questionnaire also asked subjects which combination of aids they preferred the most.

6.3 Experiment 1

The first experiment compared user performance under all combinations of the three linking methods, with only a single local view present (as illustrated in Figure 6.1):

- Proxy vs. no proxy
- Tethers vs. no tethers
- Track-up orientation coupling vs. no coupling

When coupling was enabled, it was an absolute coupling in both position and orientation. The center of the overview display always corresponded to the position of the randomly wandering vehicle, and was constantly rotated to track the forward direction of the vehicle—the "up" direction of the overview was always parallel to the vehicle's heading.

6.3.1 Design

Each subject was trained on a block of 40 trials before being presented with 4 experimental blocks of 32 trials each. Each experimenteal block was divided into sub-blocks of 8 trials each as shown in Figure 6.3. Subjects were asked to take a five to ten minute break between blocks 2 and 3.

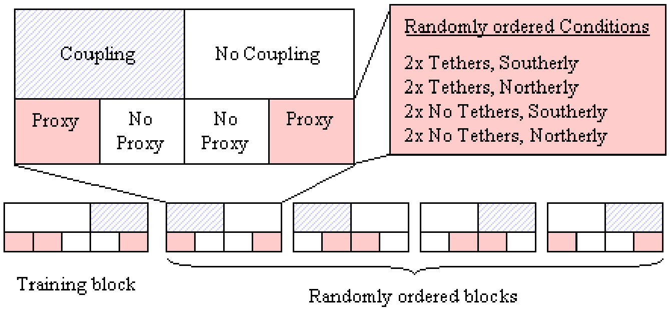


Figure 6.3: Experimental design of Experiment 1.

The experiment was set up as a 2x2x2x2 within-subjects factorial design. Within each 2x2 sub-block, the presence or absence of *tethers* was varied, as was the expected *direction* that the local view would have to face for the target to be present (northern semicircle vs. southern semicircle). Each combination of these variables appeared twice in each sub-block in a random order. Between sub-blocks, the presence or absence of the directional *proxy* was varied, as was the state of the track-up *coupling* (enabled vs.

disabled). Sub-blocks were organized such that all conditions with the same value for *coupling* were grouped together, and that the order of each combination of *coupling* and *proxy* were counterbalanced across blocks.

6.3.2 Subjects

Experiment 1 was run on 17 subjects: 10 male and 7 female. We discarded all trials in which the subject failed to make a selection. This amounted to less than 3% of the total. The results for the remainder of the data are summarized in Figure 6.4 through Figure 6.7.

6.3.3 Mean Error Rate Results

For each subject, a mean error rate was calculated for each cell in the 2x2x2 matrix of *coupling* versus *proxy* versus *tethers*. An analysis of variance performed on the mean error rate revealed both *coupling* and *proxy* to be highly significant. *Coupling* reduced errors by 27% (**F**(1, 16) = 7.03, **p** < 0.001), while use of a *proxy* reduced errors by 52% (**F**(1, 16) = 33.10, **p** < 0.001). The mean error rates for *proxy* and *coupling* conditions are summarized in Figure 6.4. The interaction between *proxy*, *coupling*, and *subject* was significant (**F**(16, 16) = 3.42, **p** < 0.01). This interaction effect is likely due to the sparsity of data (two samples) in each cell at this level of interaction. There was no main effect for *tethers* and there were no other interactions.

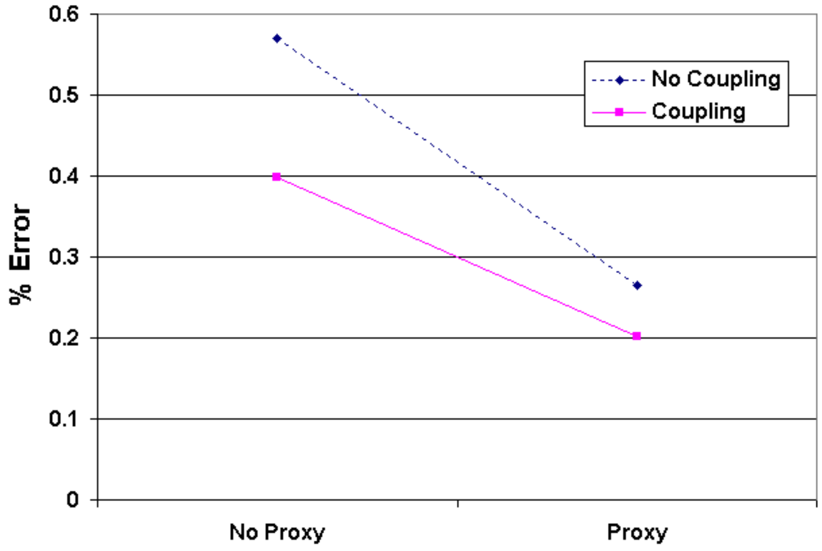


Figure 6.4: The effects of *proxy* and *coupling* on percentage of errors made by subjects during Experiment 1.

For each subject, a mean error rate was also calculated for either *direction*. An analysis of variance performed on this mean error rate indicated no main effect for *direction*.

6.3.4 Error Magnitude Results

An analysis of variance was performed on *coupling, proxy, direction,* and *tethers* with magnitude of errors as the dependent variable, over all trials that ended in an error. Magnitude of error was measured in degrees as the angle between the ray from the vehicle to the target and the ray from the vehicle to the selected distracter (in the X-Y plane). The analysis again revealed *coupling* and *proxy* to be significant. *Coupling* reduced the magnitude of error by 38%, or 15˚ (**F**(1, 45) = 7.47, **p** < 0.01), while a *proxy* reduced error magnitude by 67%, or 26˚ (**F**(1, 22) = 17.63, **p** <= 0.001). The interaction between *coupling* and *proxy* was significant as well (**F**(1, 32) = 9.71, **p** < 0.01), with the effects illustrated in Figure 6.5. No other variables or combinations of variables were significant.

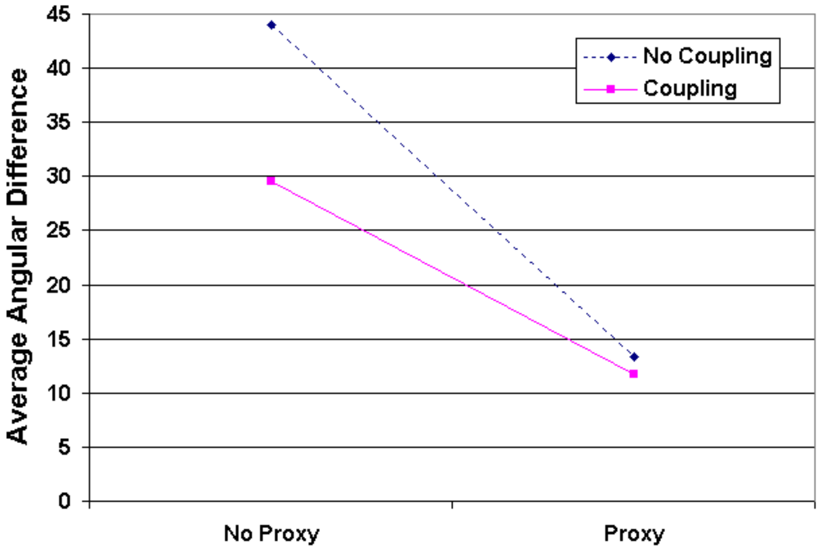


Figure 6.5: Average angular difference in degrees between the target and the subject's selection in Experiment 1.

6.3.5 Decision Time Results

An analysis of variance was run on all four variables again with respect to decision time over all valid trials (ending in either success or error). As described earlier, decision time was measured as the time elapsed between the first appearance of the target and when the subject moved the cursor to make a selection on the overview map. This analysis showed that *coupling* reduced decision time by 15%, or 1 second (**F**(1, 16) = 12.44, **p** < 0.01), and a *proxy* reduced decision time by 18%, or 1.4 seconds (**F**(1, 16) = 48.77, **p** < 0.001). Their interaction was also significant (**F**(1, 32) = 13.51, **p** < 0.01), with the effects illustrated in Figure 6.6. No other variables or combinations of variables were significant.

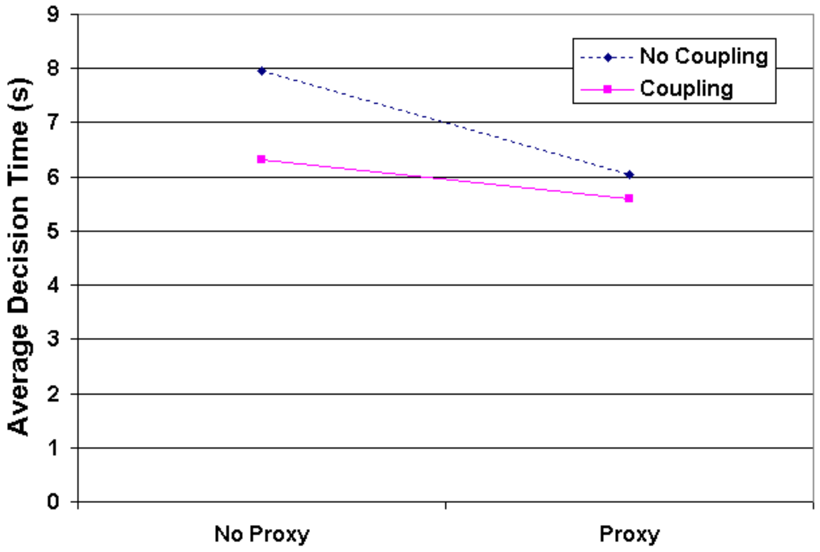


Figure 6.6: The average time elapsed in seconds from when the target first appeared in the local view and the subject's selection in Experiment 1.

6.3.6 Subject Preference Results

Figure 6.7 summarizes the opinions of subjects collected in the questionnaire. The range of responses for both the proxy and coupling aids were between 7 and 10 (extremely useful), with averages of 9.1 and 8.4, respectively. The range of responses for the use of tethers was between 1 (counterproductive) and 7, with an average of 4.6. On the question of which combination of features was best, 8 answered with the *proxy/coupling* combination, 4 answered with all aids, and 1 answered with just the proxy. Four subjects misinterpreted the question, apparently answering instead which aid was most important. To this question, 2 answered with the *proxy*, and 2 answered with *coupling*.

6.3.7 Discussion

The results of Experiment 1 indicate that the *proxy* and *coupling* devices individually contribute to performance improvement in both accuracy and decision time, while the *tethers* do not. The results further show that these two linking aids can be combined for further improvement in terms of reducing the number of errors. The results also indicate that the *direction* of the local view has no significant effect on accuracy or decision time. Subject preferences are in line with the performance data.

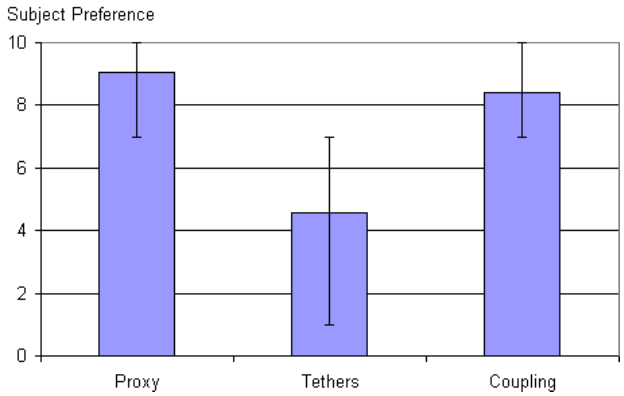


Figure 6.7: The average and range of subject opinions on the utility of each linking aid in Experiment 1.

While tethers do not have a significant effect on this task with a single local view, it was hypothesized that tethers would benefit a task with multiple local views, at least when coupling was not in use. This appeared to be a plausible hypothesis because adding another vehicle would introduce ambiguity: Which view belongs to which proxy? This was the motivation for the second experiment.

6.4 Experiment 2

The second experiment compared user performance under combinations of only tethers and proxies, with two local views present (as illustrated in Figure 6.8). Each local view followed its own vehicle, and each vehicle followed its own constrained random path. Orientation coupling was inappropriate for this experiment because the overview could only have been coupled to one local view. The purpose of this experiment was to determine whether or not tethers could be useful when ambiguity exists in the identity of the representations of independent views.

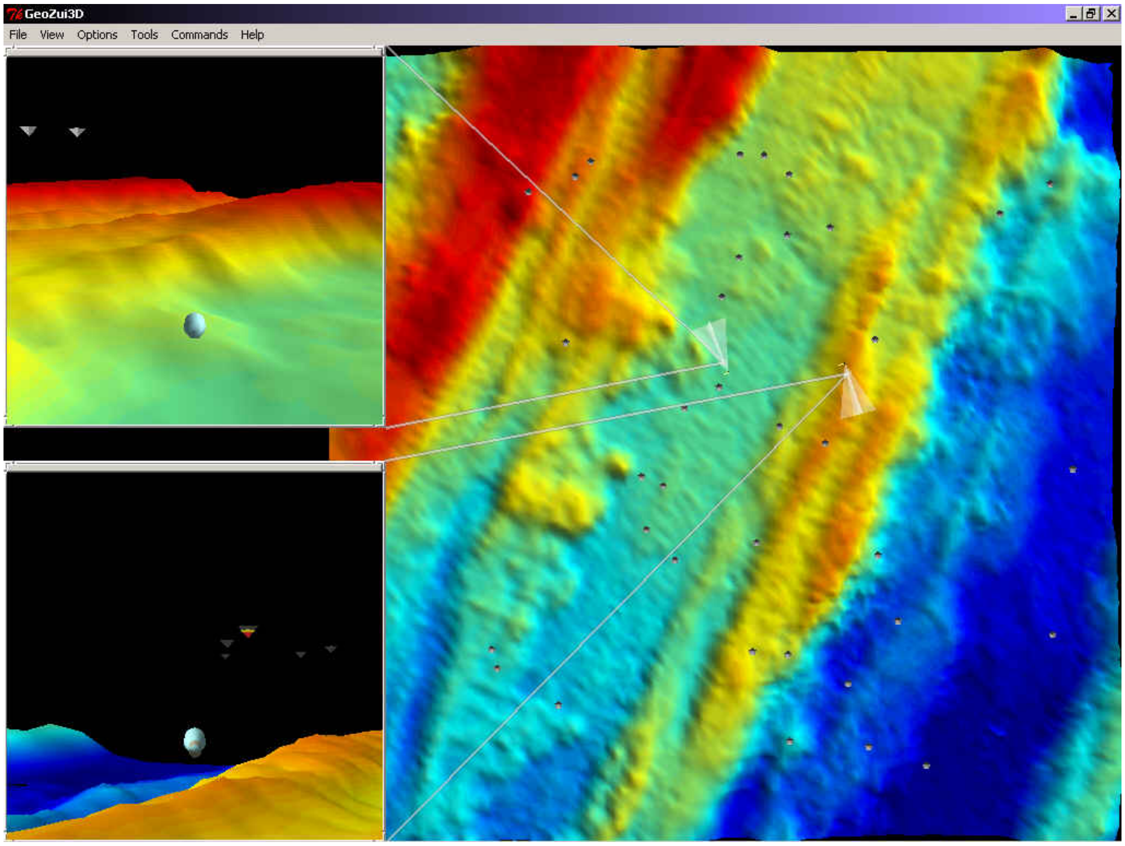


Figure 6.8: Example of the display presented to subjects in Experiment 2.

6.4.1 Design

Each subject was trained on a block of 40 trials before being presented with 4 experimental blocks of 32 trials each. Each block was divided into sub-blocks of 8 trials each as shown in Figure 6.9. Subjects were asked to take a five to ten minute break between blocks 2 and 3.

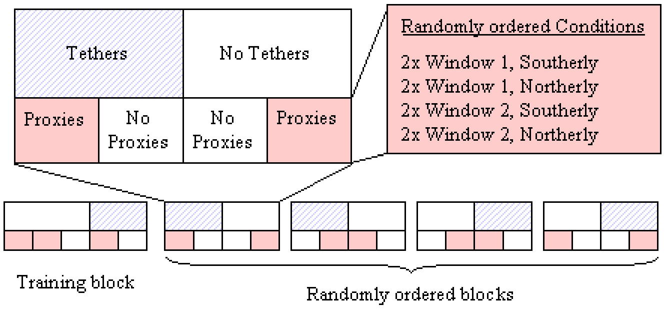


Figure 6.9: Organization of treatments in Experiment 2.

Within each sub-block, the *window* in which the target was expected to appear first was varied, as was the expected *direction* that the local view would have to face for the target to be present (northern semicircle vs. southern semicircle). Each combination of these variables appeared twice in each sub-block in a random order. Between sub-blocks, the presence or absence of the directional *proxies* was varied, as was the presence or absence of *tethers*. Sub-blocks were organized such that all conditions with the same value for *tethers* were grouped together, and such that the order of each combination of *tethers* and *proxies* were counterbalanced across blocks.

6.4.2 Subjects

Experiment 2 was run on 17 subjects: 11 male and 6 female. All trials in which the subject failed to make a selection were discarded. This amounted to about 5% of the total. The results of one male subject were discarded completely; his comments and data indicated he did not even try to make a valid selection when there were no linking aids in place.

6.4.3 Mean Error Rate Results

For each subject, a mean error rate was calculated for each cell in the 2x2 matrix of *proxies* and *tethers*. An analysis of variance performed on this error rate revealed *proxies* to be significant. Use of *proxies* reduced errors by 47% (**F**(1, 15) = 105.52, **p** < 0.001), from an error rate of 69% to an error rate of 37%. The interaction between *proxies* and *subject* was significant (**F**(15, 15) = 3.07, **p** < 0.05). This interaction effect was in the amount that the use of proxies helped; there were no subjects for which proxies degraded performance. There was no main effect for *tethers* and no other interactions.

For each subject, a mean error rate was also calculated for each cell in the 2x2 matrix of *direction* and *window*. An analysis of variance performed on the mean error rate revealed that *direction* was significant: heading in a southerly direction when discovering the target decreased the error rate by 10%, from a rate of 56% to a 50% error rate (**F**(1, 15) = 9.25 and **p** < 0.01). There was no main effect for *window* and no interactions. Figure 6.10 illustrates the relative effects of *proxies* and *direction* on error rate.

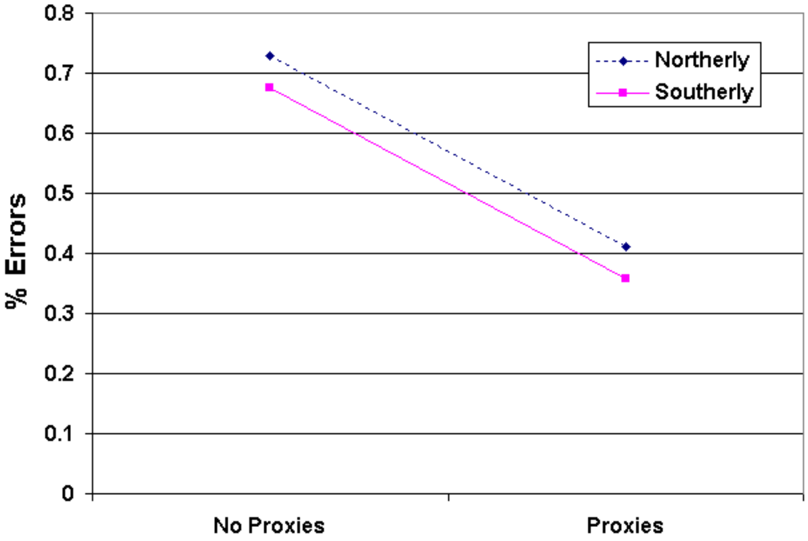


Figure 6.10: The effects of *proxies* and *direction* on percentage of errors made by subjects during Experiment 2.

6.4.4 Error Magnitude Results

An analysis of variance on *proxies, tethers, direction*, and *window* was performed with angular difference as the dependent variable, for all trials that ended in an error. This analysis revealed only *proxies* to be significant. The presence of *proxies* reduced the average angle of error by 39%, from 51˚ to 31˚ (**F**(1, 17) = 14.94, **p** < 0.01). There were no main effects for any of the other variables, and no interactions.

6.4.5 Decision Time Results

An analysis of variance on the same four variables was performed with respect to decision time. This analysis found *proxies* and *window* to be the most significant individual factors. Presence of *proxies* decreased average decision time by 13%, from 13.9 seconds to 12.1 seconds (**F**(1, 15) = 10.89, **p** < 0.01). Having the target appear in the upper *window* as opposed to the lower decreased decision time by 11%, from 13.7 to 12.3 seconds (**F**(1, 15) = 6.67, **p** <= 0.05). The relative contributions of *proxies* and *window* to decision time are shown in Figure 6.11.

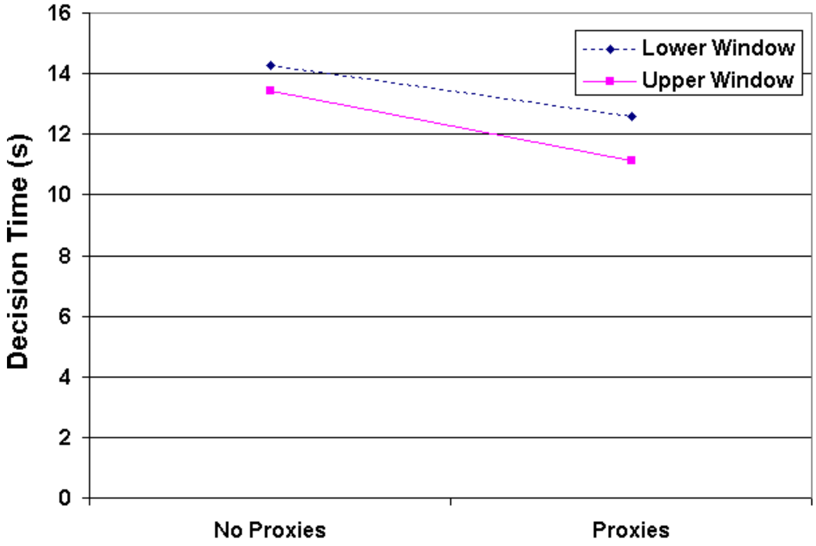


Figure 6.11: Average decision times in Experiment 2.

The presence of tethers reduced decision time by roughly 6%, from 13.4 seconds to 12.6 seconds. However, this failed to reach statistical significance (**F**(1, 15) = 3.78, **p** ≈ 0.07). There was a significant interaction between *proxies*, *tethers*, and *direction* (**F**(1, 16) = 5.67, **p** <= 0.05). The means for this interaction are given in Table 6.1. This cause of this interaction does not elicit a ready explanation.

Table 6.1: Interaction of variables on decision time. The interaction is most apparent in the "No *Tethers*" column under "*Proxies*".

	No *Proxies*		*Proxies*	
	No *Tethers*	*Tethers*	No *Tethers*	*Tethers*
Northerly *Direction*	14.1	13.4	11.2	11.5
Southerly *Direction*	14.4	13.6	13.9	11.8

6.4.6 Subject Preference Results

Figure 6.12 summarizes the subjects' opinions collected from the questionnaire. The range of responses for the aid of proxies was between 8 and 10 (extremely useful), with an average of 9.8. The range of responses for the use of tethers was between 1 (counterproductive) and 8, with an average of 6.6. On the question of which combination of features was best, 10 answered with both proxies and tethers, and 6 answered with just the proxies.

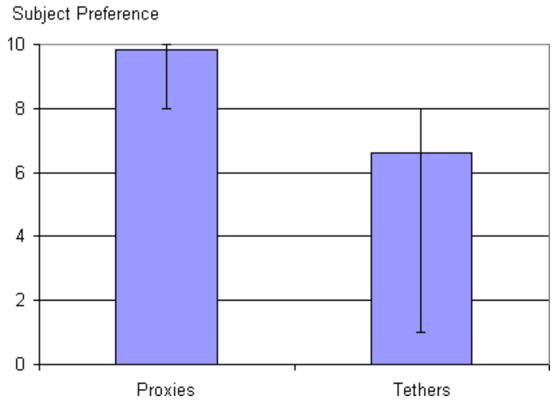


Figure 6.12: The average and range of subject opinions on the utility of each linking aid in Experiment 2.

6.4.7 Discussion

The results of Experiment 2 indicate that the use of a *proxy* device contributes in both accuracy and decision time.

Although tethers failed to provide a statistically significant benefit, it is still possible that they could be useful in cases involving many local views or when there is a higher cognitive load on the user. Even in these cases, however, it would appear that tethers are only useful for a brief period of time. This indicates that a better implementation might involve a strategy for having tethers appear briefly upon certain user actions. Such behavior has been implemented and is an option available in the GeoZui3D system.

In contradiction to the results of Experiment 1, *direction* had a small but significant effect on the number of errors. This may have been due to the increased difficulty of Experiment 2 over Experiment 1, and a consequential increase of attention paid to the detail of the land underneath. The land to the south was more distinctive than the land to the north, and therefore may have provided better contextual cues.

Another result is that subjects responded faster when the target appeared in the upper *window* than when it appeared in the lower window, by about 1.4 seconds. It appears that most subjects were paying attention primarily to the upper window, incurring a time penalty if the target appeared in the lower window. Four subjects appeared to pay attention primarily to the lower window. If the absolute values of differences between response times were taken across subjects, the average difference is much closer to 2 seconds, although the analysis does not indicate an interaction between *window* and *subjects*.

6.5 General Discussion

The strongest result from both experiments is the utility of directional proxies in reducing errors, with a reduction in error rates of around 50%. Orientation track-up coupling worked nearly as well for use with a single local view, but was not straightforward with multiple views. Tethers contributed little, even when two local views were present. These results suggest some guidelines for interface design, and provide inspiration for new possibilities.

1. Use directional proxies. The results suggest that, for an MP-ID task such as the one employed in the experiments, a directional proxy should be made available for each local view. It appears that directional proxies aid in the mental transformation of an angle from a plane along the line of sight (the local view) to one perpendicular (the global overview). The results do not indicate what precise characteristics a directional proxy should have, but experience suggests that the proxy should be as minimal as possible while still providing information about the extents of the viewing angle.

2. Use track-up coupling or a similar aid to enhance the proxy. The results agree with and support Aretz [1991] and Eley [1988] in this matter. Track-up coupling simplified the experimental task, at least in part, by reducing the area of the overview that subjects had to consider—it was always between the middle and top of the screen in a relatively narrow area. Furthermore, if the target crossed directly in front of the local view, the subject could simply select the target in a straight line above the center of rotation (since there were rarely more targets along that line). Without a proxy, coupling does not provide guidance as to how to map an angle from the plane along the line of sight to one perpendicular. However, it does make consistent the notions of left and right within these two planes, and it does provide a line of reference for the forward direction.

The results showed that tethers tended to add clutter without apparent benefit. Subjects were able to associate windows with proxies in their absence, presumably by using terrain matching and motion cues. However, this may have been partly due to the nature of the task, which gave subjects ample time to visually associate windows with proxies before the target appeared. It is quite possible that in a dual task situation requiring an operator to using the three-dimensional display to intermittently monitor some situation, the tethers could be more useful. Also, it might be useful to display tethers only when the user needs them for making the decision as to which proxy (or moving object) belongs with which local view.

Alternatives to tethers include things such as color-coding the proxies to match the window borders of local views. The results say nothing for or against the use of tethers in static images (such as magazine illustrations), nor in environments with many moving objects—these situations may still benefit from the constant use of tethers.

In the second experiment subjects tended to be faster when the target appeared in the upper of the two local view windows. This result suggests that when there are multiple local views, the interface should designate one as being primary, especially if there is a higher probability of needing information from that view. This might be done by simple placement (for instance, the upper-left corner for use in western cultures), or by a distinctive border around the focal window.

There was an interesting result that almost reached significance in the first experiment. This result was that track-up coupling helped male subjects more than female subjects. If the result were to reach significance in a wider sample, it could be that men and women tend to approach the task of integrating the two views differently, in terms of which perceptual and cognitive resources are employed. If this is the case, the second guideline regarding track-up coupling may require amendment to consider sex differences.

6.6 Conclusion

The experiments in this chapter have quantified the utility of three devices for linking focus and context in multiple views. The results show that both proxies and track-up coupling are effective devices for helping people to understand the relative spatial arrangement of the views. Counter to expectations, tethers proved not to have a measurable benefit. However, this may have been partially due to the particular task constraints of the study.

The fact that tethers approached significance in the second experiment implies that there may be situations in which tethers could provide a benefit. One of the problems with tethers was that they introduced visual clutter, which suggests that some strategy for intermittent display might be beneficial. Possibilities for implementing just-in-time display of tethers include making the tethers appear only when the mouse cursor is over a proxy, or when the user actively selects the proxy (by clicking on it, for instance). It may also be useful to make the tethers appear for one or two seconds when the mouse cursor first enters a local view. Some of these techniques have been implemented in GeoZui3D, but they have not been found preferable to static tethers in any situations so far.

Track up coupling might also benefit from a more flexible approach. For instance, track-up coupling could become active between a particular local view and the overview only when the user's cursor is within the local view. When the user exits, the coupling would be deactivated and the overview would either stop moving or animate back to a canonical orientation.

More important than improving these techniques is determining just why some are especially effective while others are not. Let it be conjectured, then, that proxies and view coupling are extremely useful because they do some of the work that the human visual system is not well adapted to, namely rotation in the image plane. A proxy visually transforms context information about the local view into the context of the overview, making it easier to correlate between the two and find the focus. With a slightly less powerful effect, track-up coupling behaviorally transforms the context of the local view to the context of the overview by re-orienting the overview's context. On the other hand, tethers offer what is essentially tracking information—assigning an identity to a proxy as to which window it belongs to. For one or two items, the human brain is already well equipped to do such tracking. Once the identities are discovered, the tethers become redundant, as long as the mental resources for tracking are not required for performing another task. The best leverage for linking focus and context through multiple views may be in transforming information from one view into the visible context of another when such a transformation is not a natural capability of the human visual system.

CHAPTER 7

A MODEL OF NAVIGATION PERFORMANCE FOR MULTISCALE COMPARISON TASKS[1]

Navigation methods, such as zooming or the use of multiple windows, provide a mechanism to link focal and contextual reference frames at different locations and scales in a virtual space. In this chapter, a theoretical model of performance is presented that identifies the relative benefits of different navigation methods when used by humans for completing a task involving comparisons between widely separated groups of objects. While the model is general enough to be applied to any navigation interface, this chapter concentrates on a comparison between zooming and multiple windows. The crux of the applied model is its cognitive component: the strength of multiple windows comes in the way they aid visual working memory.

The task to which the model is applied in this chapter is multiscale comparison. In such a task, a user begins with a known visual pattern and searches for an identical or similar pattern among distracters. This task is similar to scientific tasks such as identifying combinations of geologic features that might suggest certain sediment types or habitats.

[1] The contents of this chapter have also been published in [Plumlee and Ware 2002a] in a modified form.

7.1 General Performance Model

The applied model that is developed in this chapter is a simplified case of a more general model, which is also original to this dissertation. A brief introduction to the general model is given to make the assumptions of the applied model more obvious. The general model for human performance in a navigation-intensive task is as follows:

$$T = S + \sum_{i=1}^{V} (B_i + D_i) \qquad \textbf{(7.1)}$$

where

T is the expected time to complete the task,

S is the expected overhead time for constant-time events such as setup and user-orientation,

V is the expected number of visits to be made to different focus locations during the course of the task,

B_i is the expected time to transit to the location corresponding to visit i, and

D_i is the expected amount of time that a user will spend at the location corresponding to visit i.

The general model essentially breaks a task up into three time categories based upon a specific notion of a visit. For the purpose of the model, a *visit* to a particular location includes the transit (navigation) to the location and the work done at that location before any visits to another location. Time spent navigating *to* a location during visit i is accounted for by B_i. D_i accounts for time spent *at* that location, performing work such as making comparisons or editing objects. Time spent on anything unrelated to any visit is accounted for in the overhead or setup time (S).

Breaking a task up in this way is beneficial because there are two major ways in which a user interface can have an effect on user performance. First, it can make transitions between locations happen faster, which is manifested by a reduction in the *B* terms. An effective interface can be characterized by low values for *B*, with minimal contribution to *S* (for interface-dependent setup tasks such as resizing windows). The relative size of *B* and *D* terms also indicate the impact a change in interface might have with respect to the amount of work that would occur independently of the interface chosen. If *B* is already low with respect to *D*, a change in interface is unlikely to have a large impact on the overall efficiency with which a task is completed.

The second way a user interface can have an effect on user performance is by reducing *V*, the number of visits required. An effective interface can be characterized as one that reduces *V* without increasing the *B* or *D* terms too much. However, if *S* is already high with respect to the sum of the time spent on visits, a change in interface is unlikely to have a large impact on the total time required to complete the task. How an interface can have an effect on *V* will be described later.

7.2 Multiscale Comparison Tasks

In this section, the general performance model is made specific to the *multiscale comparison* task through the application of some simplifying assumptions. A multiscale comparison task is similar to a sequential comparison task (used by Vogel et al. [2001]) in that it asks a user to compare probe object sets to sample object sets, where each set has the same number of objects. However, in this task there are several probe sets, and the object sets are all separated by distance rather than by time. The sample and probe sets are sufficiently far away from each other that traversal of distance or scale must take place; the sets are too far apart relative to their scale to make the comparison directly. Whereas in a sequential comparison task, the user has no control over visits to the object sets, the performer of a multiscale comparison task may revisit sample and probe sets as often (and as long) as desired to make a match determination. The multiscale comparison task is intended to bear some resemblance to problems that may arise in real applications.

For a multiscale comparison task, the number of visits V is dependent upon the number of probe sets in the task, as well as the number of visits required to determine whether or not a probe matches the sample. Both the expected transit time for a visit B_i and the expected time spent during a visit D_i are considered to be invariant, making it possible to replace the sum by a multiplication by the number of visits:

$$T = S + f_V(P, V_p) \cdot (B + D) \qquad \textbf{(7.2)}$$

where

P is the expected number of probe sets that will be visited before the task is completed,

V_p is the expected number of visits made for each probe,

f_V is a function that calculates the total number of expected visits given P and V_p,

B is the expected time to make a transit on any given visit, and

D is the expected time for the user to make a match determination on any given visit.

For a given task instance, all of these parameters are static; the use of *expected* values means that the model only addresses average behavior. If one affects a change on the number of visits across task instances (by changing either P or V_p), the model basically asserts that the time it takes to complete a multiscale comparison task is a linear function of the number of visits made during the course of the task. The model also characterizes the effectiveness of an interface in terms of its ability to get a user from place to place (*B*) and the amount of setup time required (*S*).

In order to better define the visit-function f_V, a strategy for completing the multiscale comparison task must be chosen. Consider, for the sake of simplicity, the obvious strategy of making a match determination for each probe set before moving on to the next probe set. If only a subset of the objects can be remembered on each visit, the same probe might be visited a number of times before a determination is made. This strategy eliminates one trip to the sample for each probe set that differs from the sample set, since some objects remembered from a differing set can be carried to the next probe set. If there are *p* probe sets, then the expected number of differing sets visited is $P = (p\text{-}1)/2$. The total number of visits would then be the expected number of differing sets ($V_{differ} - 1$) times the number of visits for each of these sets, plus the number of visits required for a set that matches the sample set (V_{match}):

$$f_V(P, V_p) = \left(P \cdot (V_{differ} - 1) + V_{match}\right). \qquad \textbf{(7.3)}$$

7.3 <u>Cognitive Model of Visual WM: The Number of Visits</u>

In order for the model for multiscale comparison to be used to make predictions, it is necessary to estimate the values of V_{match} and V_{differ}. The capacity of visual WM plays a key role in estimating these values. To see why this is so, consider what must occur for the successful comparison of two sets of objects. In order to make a comparison, the task performer must remember objects from one set, then transit to the other set and compare the objects seen there with the ones remembered. The work of Vogel et al. [2001] suggests that there are strict limits in the number of visual objects that can be held in visual WM. If only a fixed number of objects can be remembered, the task performer must transit back and forth between the two sets a number of times inversely proportional to the limit on visual WM.

According to this argument, the important factors are n, the number of objects in each set to be visited, and M, the maximum number of objects that can be held in visual WM. With relatively few objects to be compared ($n \leq M$), it would be reasonable to expect someone to remember all of the objects from the first set, and a match determination could be made with a single reference to each set. However, as the number of objects increases, it is only possible to remember *some* of the objects ($n > M$). In this case, a match determination requires several visits between each set, with the optimal strategy consisting of attempts to match M items per visit.

It should be noted that fewer trips would be necessary if verbal WM were to be used concurrently with visual WM. This is because the information seeker could verbally rehearse some information, such as “red cube, blue sphere”, while visually remembering information about another two or three objects, thereby increasing total capacity. What follows is an analysis of the number of trips needed based on visual WM limitations alone, assuming that verbal WM is already engaged for other purposes.

If the sets of objects being compared do indeed match, then the number of visits V_{match} that must be made is proportional to the number of objects in each set. If the subject executes the optimal strategy (and if this strategy does not require additional resources from visual WM), the following equality holds.

$$V_{match} = 1 + \left\lceil \frac{n}{M} \right\rceil \tag{7.4}$$

The addition of one is required for the initial visit to the first set of objects, where items are first loaded into visual WM and no comparisons can yet be made.

If the sets do not match, and they differ in only one object, then there is a specific probability that the remembered subset will contain the differing object on any given visit. This probability can be found by partitioning the objects into groups of size *M*, with one group having a remnant $r \leq M$. The probability of a full group containing the differing object is M/n, with the remnant group having a probability of r/n. This partitioning is demonstrated in Figure 7.1. To find the expected number of visits when the sets differ by one, V_{differ}, the probability of finding a differing object on a given visit is multiplied by the rank of that visit, and these products are summed to the maximum possible number of visits.

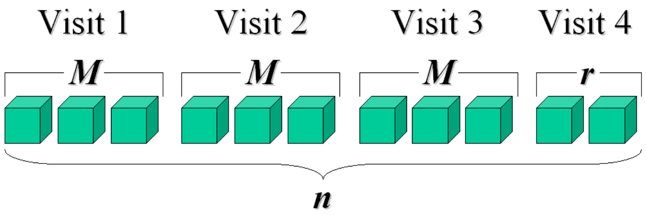


Figure 7.1: Determining the expected number of visits by partitioning the probability of finding the differing object during a visit.

Again, one is added to the final result for the initial visit to the sample set.

$$V_{differ} = 1 + \left(\sum_{i=1}^{\left\lfloor \frac{n}{M} \right\rfloor} \frac{M}{n} \cdot i \right) + \frac{r}{n} \cdot \left(\left\lfloor \frac{n}{M} \right\rfloor + 1 \right) \tag{7.5}$$

Substituting for *r* and reducing, the following is obtained.

$$V_{differ} = 2 + \left\lfloor \frac{n}{M} \right\rfloor - M \cdot \frac{\left\lfloor \frac{n}{M} \right\rfloor + \left\lfloor \frac{n}{M} \right\rfloor^2}{2n} \tag{7.6}$$

If *n* is a multiple of *M*, this reduces to

$$V_{differ} = \frac{3}{2} + \frac{n}{2M} \qquad | \, n = kM, k \in \mathrm{I}\,. \tag{7.6b}$$

With estimates for V_{match} and V_{differ} in hand, it is possible to restate the expression of the number of visits from Formula 7.3 in terms of known or empirically determined quantities. Assuming n is a multiple of M,

$$f_V(P, V_p) = \frac{(2+P) \cdot (M+n)}{2M} \quad | \, n = kM, k \in \mathrm{I}\,. \tag{7.7}$$

7.4 Applying the Model to Specific Interfaces

To this point, then, a performance model has been constructed based on parameters that account for both the interface and the task. The task parameters have been further refined for the multiscale comparison task, taking into account limits on visual WM. Now the parameters for individual interfaces can be refined, namely zooming and multiple windows.

Recalling the descriptions of Formulas 7.1 and 7.2, the key variables that change between different interfaces are B and S—the transit time between visits, and the setup and overhead time. For zooming interfaces the application of the model is trivial:

$$T_{zoom} = S_{zoom} + f_V(P, V_p) \cdot (B_{zoom} + D), \tag{7.8}$$

where B_{zoom} is the expected cost of using the zooming interface to get from set to set, and S_{zoom} includes little more than the cost of a user orienting him or herself to the initial configuration of the sets. By substituting Formula 7.7 for the visit-function f_V, it follows that

$$T_{zoom} = S_{zoom} + \frac{(2+P)(M+n)}{2M}(B_{zoom} + D) \quad | \, n = kM, k \in \mathrm{I}\,. \tag{7.8b}$$

For interfaces that rely on multiple windows, the model must be applied twice, since there are actually two ways to transit between visits. The first way, of course, is by situating a window over a desired focus point using whatever method the multiple-window technique supplies. This occurs when the user wishes to visit a new set for comparison. The second way is by performing a saccade of the eyes between windows that have already been situated in this way. This is an important distinction for tasks like these that require operations on information from more than one location. It is especially important when that information cannot all be held in memory all at once. Here is how the model applies to a multiple-window interface:

$$\begin{aligned} T_{multi} = {} & S_{eye} + f_V(P,V_p)\cdot(B_{eye}+D) + \\ & S_{multi} + f_V'(P,V_p)\cdot(B_{multi}+D') \end{aligned} \qquad \textbf{(7.9)}$$

One can simplify this formula by recognizing that S_{eye} = 0, since there is no setup related to using our eyes, and D' = 0 since the work being done during a visit from a window is accounted for in the terms contributed from use of the eye. If the assumption is made that the setup cost S_{multi} includes situating the first two windows over their respective targets, then $f_V'(P,V_p) = P$, since there is no need to situate a window over subsequent probe sets more than once. Therefore, Formula 7.9 can be reduced to

$$T_{multi} = S_{multi} + P\cdot B_{multi} + f_V(P,V_p)\cdot(B_{eye}+D) \qquad \textbf{(7.10)}$$

By substituting Formula 7.7 in for the visit function f_V, we get

$$T_{multi} = S_{multi} + P\cdot B_{multi} + \frac{(2+P)(M+n)}{2M}\cdot(B_{eye}+D) \quad | n = kM, k \in \mathrm{I} \qquad \textbf{(7.10b)}$$

For a given technique and task, the various forms of *B*, *D*, and *S* can all be determined empirically. Such a determination requires establishing parameters such as zoom rate and distance between probe sets. Similarly, *P* can easily be calculated based on the number of probe sets present in the task. Once all the parameters are determined, the model can be used to compare expected user performance times under the two different interfaces.

7.5 Comparing Performance Models

Now the analytic tools are at hand to compare zooming and multiple window interfaces as they apply to the multiscale comparison task. The extra terms in Formula 7.10 beyond those in Formula 7.8 might cause one to think that zooming would always have the better completion time. This would be strengthened by the expectation that S_{multi} should be larger than S_{zoom} due to the added overhead of creating and managing the additional windows. However, as n increases beyond what can be held in visual WM, zooming requires more time to navigating back and forth between sample and probe sets (B_{zoom}), whereas multiple windows allow comparisons to be made by means of eye movements (B_{eye}).

If one considers each S as the intercept of a line, and the slope as proportional to ($B + D$), it follows that the slope of Formula 7.8 is steeper than the slope of Formula 7.10. And with a P term as a factor, one would expect the difference in slopes to be exaggerated as the number of probe sets increased. Thus, as illustrated in Figure 7.2, there must be a point at which the overhead of multiple windows is justified by the ability to make visits by quick saccades of the eye. The next section looks at a particular instance of a multiscale comparison task to illustrate how this modeling might be applied to a particular application.

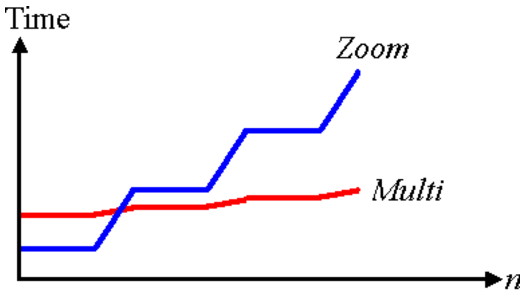


Figure 7.2: Expected relationship between performances in completing a multiscale comparison task when using zoom and multiple window techniques.

7.6 Sample Model Application

In order to evaluate the model, detailed predictions were constructed for the use of zooming and multiple-window interfaces in completing a specific multiscale comparison instance. The details are given in the following paragraphs, based on specifications for the multiscale instance and hypothetical interfaces. The specifics discussed here are used to guide the construction of an experiment in the next chapter that empirically tests the model.

From the work of Vogel et al [2001], a good estimate of the capacity of visual working memory, *M*, is 3 (assuming an integer value). The time, *D*, to determine whether or not the objects in a probe match those remembered, is a bit more elusive. From informal experience, this number should be between a half-second and a full second. While informal experience also shows that *D* is smaller for smaller *n*, let us assume that *D* is a constant 0.8 seconds.

Consider the following scenario for a multiscale task, as diagrammed in Figure 7.3. Let each object have a size that fits within a circle with a 15-meter diameter. Let the size of the object sets be 60 meters to a side, and let the minimum amount of space between sets be 3.3 kilometers (on center). Further, let the valid field of placement be a square 10 kilometers to a side. Let there be six probe sets in this field, along with the sample set (p = 6).

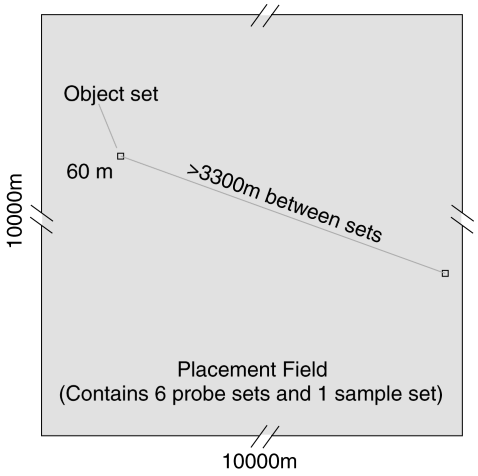


Figure 7.3: Schematic of the multiscale comparison task used in the sample model application and the experiments in the next two chapters.

With these parameters in place, the scales at which the objects in a set can be visually identified are roughly between 0.1 m/pixel and 2 m/pixel. The scales at which more than one cluster can be seen range from 3.4 m/pixel (at the very least), to 15 m/pixel (to see all of the object sets at once), to 60 m/pixel (where a set is the size of a pixel).

The task instance is now specified. Now, consider possible operating parameters for performing this instance using a zooming interface, and for using a multiple-window interface.

7.6.1 Zooming Interface

For the zooming technique, consider a zoom rate of approximately 7x/s (7 times magnification per second—this is a little less than the 8x/s rate indicated in Appendix A, chosen to accommodate less expert users). It seems reasonable to estimate that a person will normally inspect a set at a scale of roughly 0.45 m/pixel, and can zoom out to about

15 m/pixel to see the entire placement field. Thus the cost of zooming in or out can be estimated at $\log_7$ (15/0.45). The distance covered between visits is between 3.3 kilometers and 14.1 kilometers, which is between 220 pixels and 940 pixels at 15 m/pixel. A good estimate of the time to move the cursor this distance and press a mouse button to start a new zoom is about 1.5 seconds. This leads to the following conclusion: $B_{zoom} = 2 \bullet [\log_7 (15/0.45)] + 1.5 = 5.2$ seconds. S_{zoom} should be small, since the only overhead to account for is the initial user-orientation period, which can be considered to be about 2 seconds. Using all this information, and letting the number of items be $n = 3$, Formula 7.8b can be used to get an estimate on the total task time:

$$T_{zoom} = 2 + \frac{\left(2 + \frac{6-1}{2}\right)(3+3)}{6}(5.2 + .8) = 29.0 \qquad \textbf{(7.11)}$$

7.6.2 Multiple-window Interface

To model the multiple-window technique, consider a fixed scale for the overview windows of about 17.5 m/pixel and assume the user resizes the focus window to a scale of about 0.45 m/pixel. The estimated overhead time required to create, resize, and maintain proper positions of the focus windows is roughly 10 seconds per window. If it is assumed that there are two focus windows to be used (the optimum strategy for the task), and that the user will require 2 seconds for orientation, then $S_{multi} = 22$ seconds. Let us assume that one navigates the focus windows from place to place by clicking on and dragging its proxy representation within the overview (see Figure 8.1). In such a case, the optimum strategy is to park one window on the sample set, and continually drag the proxy of the other window around to each probe set. Let us further assume that these proxies are just slightly larger than the object sets, around 70 meters. With this information, and knowing that it is more difficult to properly place a proxy than select a zooming location, the expected time to move a proxy from probe to probe can be estimated at about 2 seconds per visit. This translates into a B_{multi} of 2 seconds. The final parameter required is the time for saccadic eye movements between the window over the sample and the window over the current probe. This is known to take about .03 seconds on average, with 0.1 second as a good upper bound. If the estimate of B_{eye} = 0.1 second is used (remembering that D = .8), then Formula 7.10b can be used to get an estimate when n = 3:

$$T_{multi}=22+\frac{6-1}{2}\cdot 2+\frac{\left(2+\frac{6-1}{2}\right)(3+3)}{6}\cdot(.1+.8)=31.05 \tag{7.12}$$

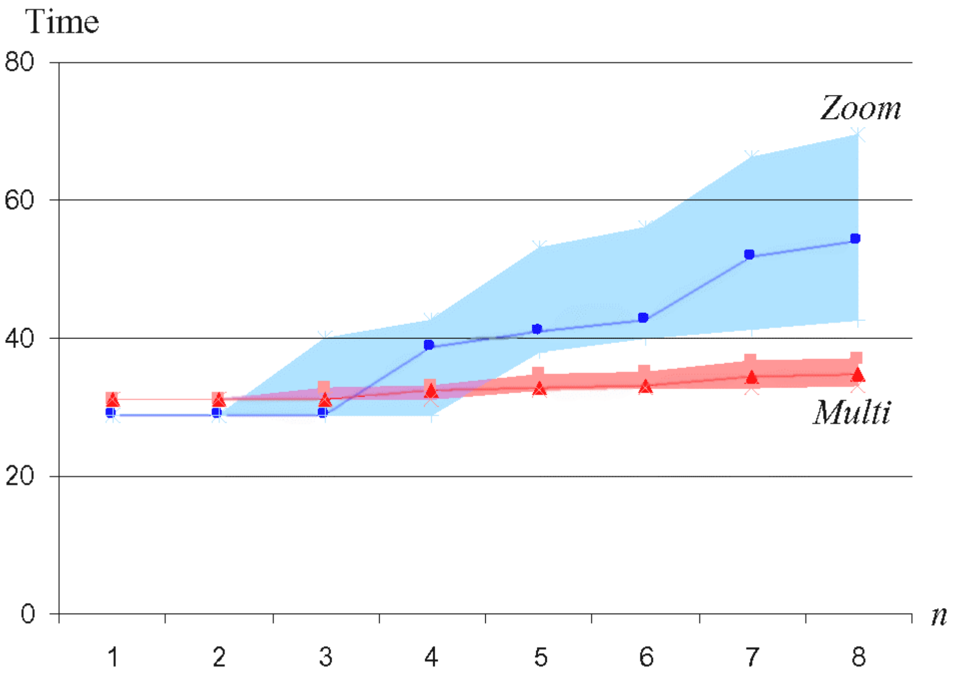


Figure 7.4: A refinement of Figure 3 using estimated parameters for each model variable. The heavy lines represent the values calculated for M=3. The borders above and below the heavy lines represent the values calculated for M=2 and M=4, respectively.

7.6.3 Putting Together Multiple Predictions

The resulting predicted times from Formulas 7.11 and 7.12 provide a comparison of zooming and multiple-window interfaces for one set of assumptions, but the power of the model comes in being able to vary some of the assumptions. For instance, what should be expected for 2 items per set (n = 2) or 5 items (n = 5)? Or, what if different people have slightly different capacities of visual WM (M = 2 or M = 4). Figure 7.4 plots the results reapplying the model while varying n between 1 and 8, and varying M between 2 and 4. The plot suggests that one should expect zooming to become less efficient than using multiple windows at between 3 and 4 items.

7.7 Model Caveats

The model described so far makes several assumptions worthy of note. The model assumes perfect accuracy of visual WM. It also assumes people that a person has the ability to remember which objects and probe sets have been visited already, and furthermore that this ability does not burden visual WM. The model contains no provisions for error, such as might occur if someone mistakenly identifies a mismatched object as matching an object in the sample set, or identifies a matching object as differing from an object in the sample set. Invalidations of assumptions or the presence of errors might manifest themselves as either lower than expected values of M, or higher than expected numbers of visits, $f_V(P,V_p)$. Either effect would serve to further increase the apparent differences in slope between the two techniques. On the other hand, careless errors may also *decrease* the expected number of visits, sacrificing accuracy for decreased task completion time. The effects of errors are explored further in Chapter 9.

Another important factor not included in the model is the amount of visual WM required by the user interface, i.e., how much the user interface decreases the capacity available to be applied to the task. Either the zooming interface or the multiple-window interface might use a "slot" within visual WM. For example, a slot in visual WM might be used to remember which probe set is currently being compared (with a zooming interface). Alternatively, visual memory might decay over the time period of a zoom (although there is evidence in the literature against decay [Vogel et al. 2001]), or intermediate images seen during zooming might interfere with visual WM. All of these effects would most likely increase the expected number of visits, thereby increasing the slope for the effected technique. If the effect is dependent upon the number of probes already visited, it is also possible that the linear relationship between n and $f_V(P,V_p)$ would become quadratic, or worse.

7.8 Conclusion

In this chapter, a theoretical model has been presented, and it has been applied to predict user performance under two focus-in-context techniques on a multiscale comparison task. The model is general enough that it could be applied to other focus-in-context techniques as well, such as fisheye views or Intelligent Zoom. It can also be serve to predict performance on similar tasks.

Because the predictions of the model rest particularly on the capacity of a visual WM, and because *verbal* WM appears to have similar capacity restrictions, the model should extend beyond purely visual tasks to include textual tasks. While verbal tasks are less likely to take the form of multiscale comparison, spreadsheets and large documents present their own multiscale challenges in contrasting or integrating data across large distances. If the data is more than can be held in verbal WM, it may be a situation for which multiple windows would be more efficient than other methods in use. In a sense, some packages already provide a restricted form of windowing, either in terms of split-screens or panes, or in terms of overview windows.

While the model was applied here to a simple multiscale comparison task, its results should also extend to complementary or higher-level tasks in which working memory plays a key role. An example complementary task is one in which object sets are dissimilar from each other, and the goal is to find the regions that share an object in common. An example higher-level task is one in which it is not known what visual pattern various distinct locations might share, if any, and the goal is to find and classify repeated patterns. A concrete example of these tasks might be in looking for similarities in different locations on the sea bottom: as a complementary task, a scientist might be looking for a particular similarity among some pair of locations; as a higher-level task, a scientist may be trying to find characteristics that are shared between regions that are already known to support the same kinds of marine fauna.

Beyond the applied form of the model used here, contrasting interfaces based on visual WM capacity, the general model exhibits a level of abstraction that could be beneficial in other comparisons of navigation interfaces. The key notion to the general model is one of a higher-level operation (a *visit*) that is common to many tasks, and is supported by many kinds of interfaces. In essence, a visit is an operation that links navigation *to* a location with the focusing of user attention *on* that location. The model therefore puts the efficiency of an interface in terms of its efficiency in allowing a user to navigate from one focus (or context) location to the next, relative to cognitive and task requirements. Whether that location is in physical 3D space, a sentence in a hypertext, or a construct in an abstract data space is immaterial to the model.

CHAPTER 8

EMPIRICAL TEST OF THE PERFORMANCE MODEL: ZOOMING VERSUS MULTIPLE WINDOWS[1]

In the previous chapter, a model of navigation performance in a multiscale comparison task was presented, and was applied in a detailed comparison of two interfaces for their efficiency in aiding users to complete the task. In this chapter, an experiment is described that tests the model. The experiment contrasts the performance of subjects using both zooming and multiple-window interfaces to complete a multiscale comparison task. The hypothesis is that, as determined by the analysis in the previous chapter, multiple windows should be slower than zooming when the number of items per set is low, and faster than zooming as the number of items increases past the maximum capacity of visual WM.

8.1 Experimental Method

The task for the experiment required subjects to perform a 2D multiscale comparison task in which they would search among six probe sets for one that matched the sample set. On the screen of a computer display, subjects were presented with a textured 2D background upon which the seven sets of objects were randomly placed, as shown in Figure 8.1. The sample set had a random arrangement of *n* objects, and was identifiable by its yellow border. The probe sets each had a gray border and the same

[1] The contents of this chapter have also been published in [Plumlee and Ware 2002a] in a modified form.

number and arrangement of objects as the sample set, but only one matched the sample set exactly. The other five probe sets differed in exactly one object, either in shape, in color, or in both aspects. The background texture camouflaged the clusters and their contents at intermediate scales—enough to cause a subject to zoom in or out by a significant amount so as to see individual objects, or spot the clusters in relation to one another, respectively (see Figure 8.3). The sizes and placement parameter values for object sets were as described in the previous chapter.

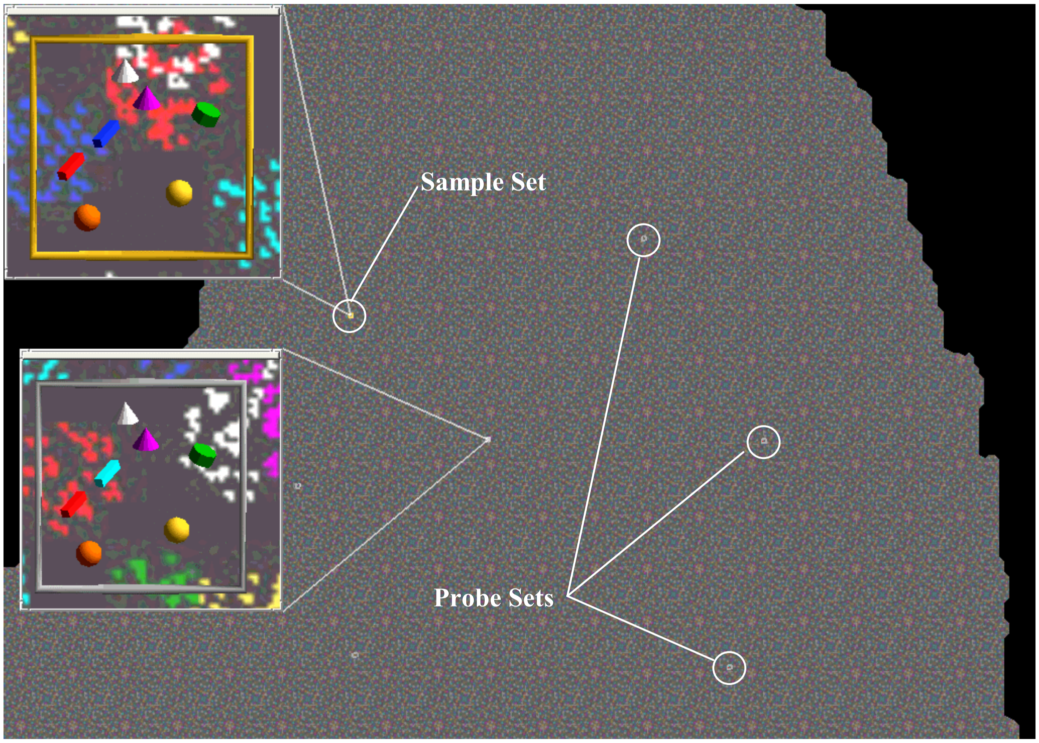


Figure 8.1: Example of the *multi* condition with two windows created. One window is focused on the sample set, while the other is focused on its match.

For each trial, the location and composition of each set was randomized according to certain constraints. In creating each set of objects, there were 5 shapes (see Figure 8.2) and 8 colors to choose from. No color or shape could appear more than twice in any object set, and objects could not overlap significantly. The configuration of the objects matched exactly for every set in a given trial. The sets themselves were considered 60 units wide, and were randomly placed in a 10,000 by 10,000 unit area such that they were never closer than 3,300 units, center to center.

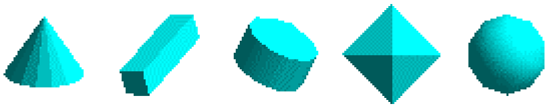


Figure 8.2: The 5 shapes that were available for creating each object set.

8.1.1 The Zooming Navigation Mechanism

During each trial, the subject was given one of two mechanisms for navigating between object sets. The first was a zooming mechanism, referred to as *zoom* for short. When the subject pressed the middle mouse button, the screen centered on the point under the cursor. If the subject then pushed the mouse forward, the scene zoomed in (at roughly 7x/s) about the new center point. If the subject pulled the mouse backward, the scene similarly zoomed out (at about 8x/s). There were no limits placed on the scale that the subject could achieve in either direction.

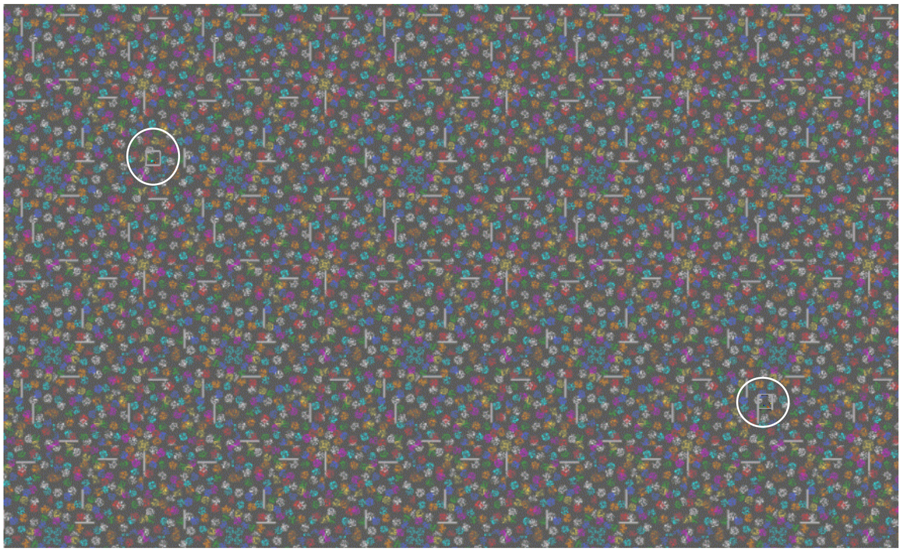


Figure 8.3: Two object sets camouflaged in the texture of the background at an intermediate scale during a *zoom* condition. To see individual objects, the subject must zoom in. To pre-attentively spot the clusters, the subject must zoom out.

8.1.2 The Multiple-window Navigation Mechanism

The second mechanism for navigating the layout was through multiple windows, referred to as *multi* for short. The scale of the main view was fixed, and there were initially no other windows. To create a window, the user first pressed the ‘z’ key on the keyboard, and then clicked the left mouse button to select a location for the center of the new window. The window was created in the upper left corner of the screen at a size too small to be useful. The subject then used the mouse to resize the window to a usable size, and was free to place it elsewhere on the screen (using common windowing techniques). The windows were brought up very small to compensate for the fact that they were automatically set to the optimal scale for viewing the object clusters. A maximum of two windows was allowed. Each window had two tethers linking it to its proxy in the main view, as shown in Figure 8.1. The proxy marked the area in the main view that the associated window was magnifying. Once a window was created, the subject could click and drag the window’s proxy through the main view to change its focus. The contents of the window were updated continuously without perceptible lag.

8.1.3 Blocking Verbal Working Memory

In order to determine whether or not verbal WM played a role in completion of the task, trials were further varied according to whether or not users had to perform a secondary task that blocked verbal working memory. In half of the trials, verbal working memory was *blocked* by requiring subjects to subvocally or mentally repeat the list “cat giraffe mouse mole” throughout the course of the trial. This secondary task precluded subjects from verbally rehearsing some information, such as “red cube, blue sphere”, while visually remembering information about another two or three objects. Such rehearsal might increase the effective total capacity, potentially decreasing the number of visits required for a subject to complete the task. In the remaining, *unblocked* trials, subjects were not asked to repeat anything.

8.1.4 Remaining Interface Details

Prior to each trial, the subject was shown a screen that told the subject how many objects to expect in each cluster, what navigation method was to be used (the other was disabled), and whether or not to repeat the list of words for blocking verbal WM. Once the subject clicked the mouse, timing began for the trial and the subject was presented with the layout at such a scale that all seven sets of objects could be located. The subject was instructed to press the spacebar on the keyboard when he or she believed that a probe set matched the sample set (the probe set had to be visible on the screen). If the subject pressed the spacebar on the correct probe set, the experiment proceeded to the next trial. Otherwise, the subject was informed of the incorrect choice and the trial was repeated with a new random layout and selection of objects.

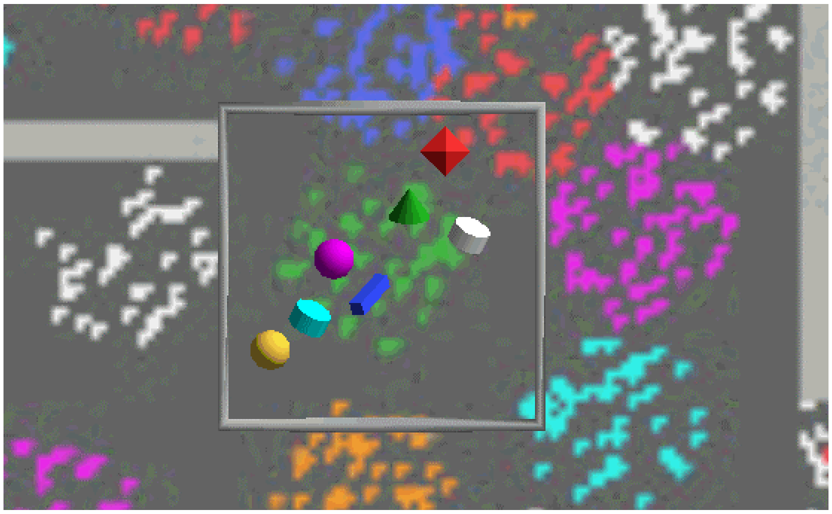


Figure 8.4: A visit to a probe set during a *zoom* condition.

8.1.5 Design

Each subject was trained using 8 representative trials, and was then presented with 4 blocks of 16 different trials in a 4x2x2 factorial design. All trials varied in three parameters:

- *n*, the number of object in each set, chosen from {1, 2, 3, 4} for the first 8 subjects, but changed to investigate the larger range {1, 3, 5, 7} for the additional 12 subjects,
- *m*, whether the navigation mechanism was *zoom* or *multi* (multiple windows), and
- *b,* whether verbal WM was *blocked* or *unblocked*.

To reduce user confusion in switching between mechanisms, each experimental block was split into two groups such that all *zoom* conditions were grouped together and all *multi* conditions were grouped together within the block. Each of these groups was again split into two subgroups such that all conditions in which visual WM was *blocked* were grouped together and all *unblocked* conditions were grouped together. The groups and subgroups were counterbalanced across the four experimental blocks and the order of the four values for *n* varied randomly within each subgroup.

At the end of the experiment, each subject was given a brief questionnaire. Subjects were asked which interface was preferred when searching for a single matching object, which was preferred when searching for the most number of objects (either 4 or 7, as appropriate), and which interface was preferred overall. The questionnaire also asked for additional comments.

8.1.6 Subjects

The experiment was run on 20 subjects: 10 male and 10 female. 8 subjects were run with *n* confined to {1, 2, 3, 4} and 12 subjects were run with *n* confined to {1, 3, 5, 7}.

8.2 Results

In total, data was collected from 1451 trials. Trials that ran longer than 90 seconds were discarded (26 from *zoom* conditions, 6 from *multi* conditions), leaving 1419 trials.

8.2.1 Completion Times

The completion-time results are summarized in Figure 8.5 for trials ending in successful completion. As predicted by the model in the previous chapter, there was a crossover in efficiency between the two navigation methods between 3 and 4 items per set.

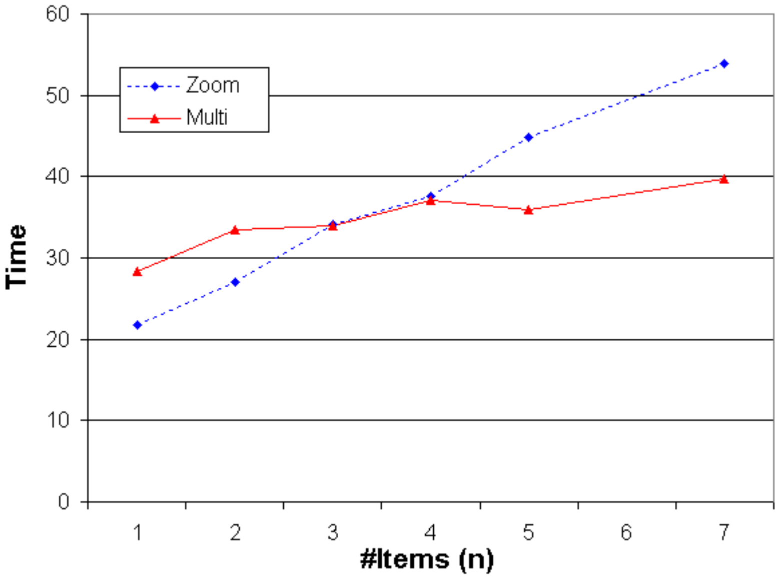


Figure 8.5: Completion-time results of the experiment, plotting the average time to successfully complete a task for various values of *n*. The *zoom* condition exhibits a greater slope than the *multi* condition.

An analysis of variance revealed that the number of objects in each set (*n*) and the interaction between the number of objects and the navigation mechanism (*n* x *m*) contributed significantly to task completion time (**F**(5, 56) = 72.41 and **F**(5, 56) = 12.16, respectively, both with **p** < .001).

In addition, the same analysis revealed that the interaction of verbal WM blocking with navigation mechanism (*b* x *m*) was significant: **F**(1, 26) = 10.91 (**p** < .01). Further ANOVAs run within each navigation method revealed that the blocking of verbal WM was not significant in the *multi* condition (**F**(1, 25)=.79), but it was in the *zoom* condition: **F**(1, 23)=7.62, (**p** < .05). This is illustrated in Figure 8.6.

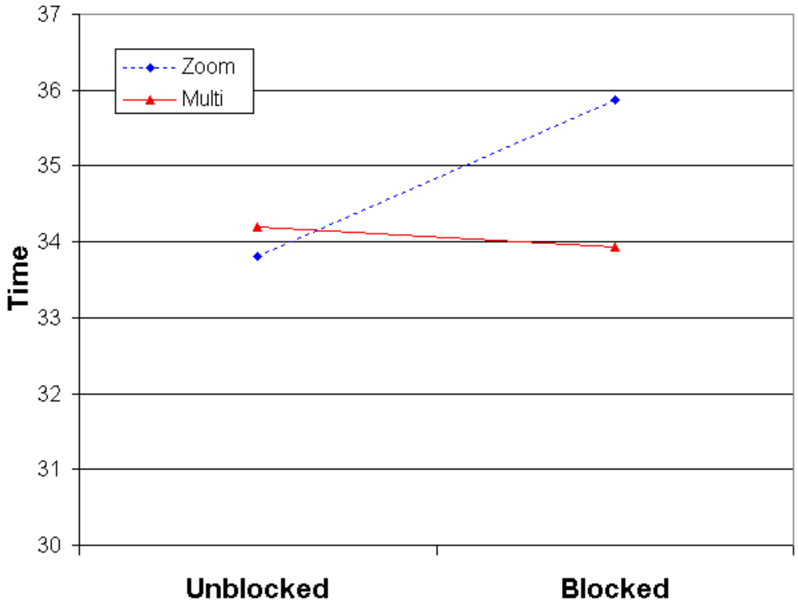


Figure 8.6: The effect of verbal WM blocking on task completion time is significant for the *zoom* condition but not for the *multi* condition.

A linear regression was performed on the summary means for each value of *n*, corresponding to fitting a line to the data displayed in Figure 8.5. This was done to get some sense of the general slope and intercept of the experimental results so that they could be compared with regressions on the predictions of the model. It should be noted that a linear regression was not wholly appropriate for this situation because of the expectation of a piecewise-linear result, among other violated assumptions of linear regression. For comparison, regressions were run on the corresponding model predictions assuming a visual WM capacity of two ($M = 2$). For the observed data from the *zoom* condition, the intercept is 16.7 seconds, and the slope (coefficient of n) is 5.41 seconds per item, with an $\mathbf{R}^2$ value of 0.99. The corresponding model regression yields an intercept of 20.3 seconds and a slope of 6.09 seconds per item. For the observed data from the *multi* condition, the intercept is 28.7 seconds, and the slope is 1.65 seconds per item, with an $\mathbf{R}^2$ value of 0.86. The corresponding regression on the model yields an intercept of 29.7 seconds and a slope of 0.91 seconds per item.

8.2.2 Error Rates

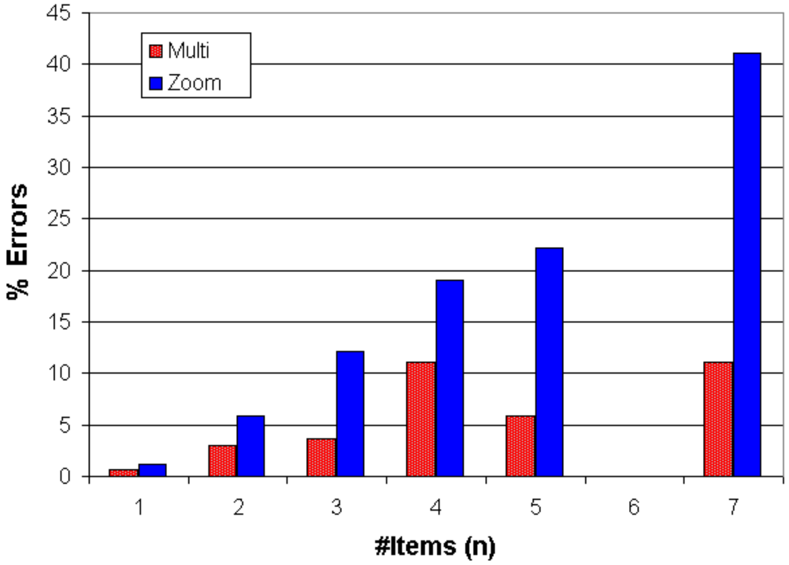


Figure 8.7: The percentage of errors for various values of *n*. The *zoom* condition exhibits a greater number of errors than the *multi* condition.

Figure 8.7 presents the average percentage of errors generated by subjects, calculated as the number of trials ending in error divided by the total number of trials for a given value of *n*. For instance, an error rate of 25% for a given value of *n* would indicate that there was one incorrect match for every three correct matches. As the figure shows, the percentage of errors generally increased with *n*, and this error rate was much greater for the *zoom* condition than the *multi* condition. It should be noted that false-positives—cases in which a subject signaled a match for a non-matching probe set—were the only kind of error readily detectable by the experimental design, and are therefore the only kind reported.

8.2.3 Subject Preference

Subject responses to the questionnaire revealed that interface preference generally reflected performance. When there was only one object in a set (*n* = 1), zooming was preferred over multiple windows (12:8). When there were significantly more objects per set (*n* = 4 or *n* = 7), all subjects preferred multiple windows. When asked which interface was preferred in general, the multiple-window interface was preferred (18:2). Many of the subjects commented on additional difficulty they had in using the zooming interface during the *blocked* conditions.

8.3 Discussion

The results of this experiment support the predictions of the model put forth in the previous chapter, namely that multiple windows are slower than zooming when the number of items per set is low, and faster than zooming when the number of items increases past *M*, the maximum capacity of visual WM. The results also show that when verbal WM was not blocked, subjects made use of it in the *zoom* condition, resulting in lower completion times. If subjects used verbal WM in the *multi* condition, it did not have any significant impact on subject performance. The model can account for this phenomenon by encompassing both visual and verbal WM in its parameter *M* for storage capacity.

There were large differences between the two interfaces in terms of the numbers of errors that occurred, as shown in Figure 8.7. The errors reported are false positives. False negatives are not reported (cases in which the user initially treated a matching set as if it differed from the sample, likely showing up as a longer task completion time). Since most of the errors occurred in the *zoom* condition, the question arose as to why the zooming interface generated so many more errors than the multiple-window interface.

One way to account for the observed differences in error rates is to assume that errors occurred because subjects made fewer visits than necessary to probe sets in order to guarantee a correct response. This assumption says that subjects essentially *guessed* that the last probe set they investigated matched the sample—perhaps after they had matched enough items that they felt it would be quicker just to guess than make any further visits. Under this assumption, there must have been something about the zooming interface that caused subjects to make fewer visits than they did with the multiple-window interface.

8.4 Post-hoc Error Analysis

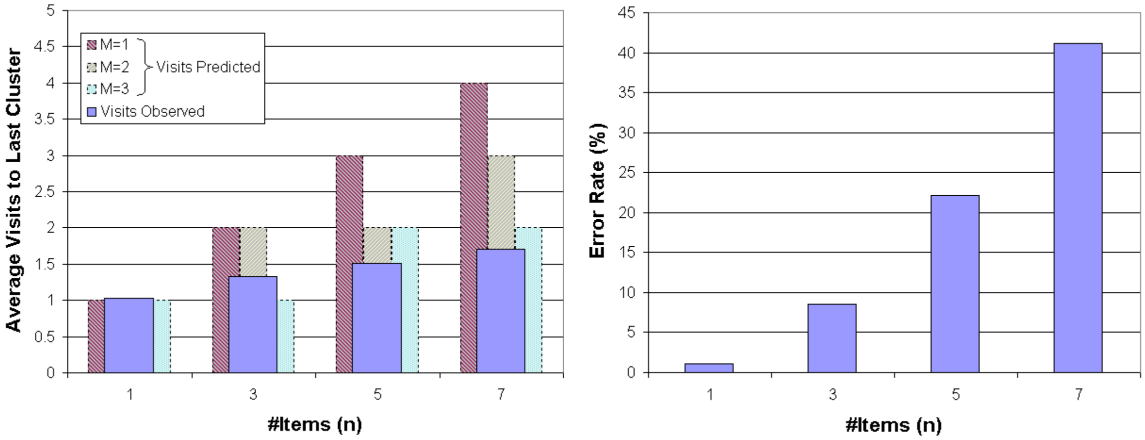
To test this assumption, a post-hoc analysis of the data was carried out to see how the numbers of visits observed compared with those predicted by the model. It was possible to do this analysis for the *zoom* condition because the necessary data was collected, but visits in the multiple-window interface were made with the eye and were not measured. Thus, a post-hoc analysis was performed on some of the *zoom* data for this experiment and a separate experiment was planned to collect additional data.

For the post-hoc analysis, data was only used from the 12 subjects who had *n* chosen from {1, 3, 5, 7}, 4 of whom were male and 8 of whom were female. This was done to maintain consistent conditions between this analysis and the analysis run on the experiment described in the next chapter. The analysis focused on how many visits subjects made to the last probe set—the set under investigation when the subject made the "match" decision and pressed the space bar.

Plotted in the background of Figure 8.8(a) are the predicted number of visits required to achieve perfect performance, assuming capacities of visual working memory at 1, 2, and 3 objects. The predicted values were calculated by modifying Formula 7.4 to count only the number of visits to the matching probe set (Formula 7.4 includes visits to both the probe and sample sets):

$$V_{matching-probe} = \left\lceil \left(1+\left\lceil \frac{n}{M} \right\rceil\right) \Big/ 2 \right\rceil \tag{8.1}$$

The foreground bars in Figure 8.8(a) illustrate the average number of visits subjects actually made to this last probe set for each level of *n*. The number of visits observed match the model when there is 1 item per cluster, but subjects seem to have "under-visited" the final set when it contained 5 or 7 items.



(a) (b)

Figure 8.8: The number of visits to the last probe set investigated and the number of errors made, versus the number of items in the sample set: (a) actual number of visits to the last probe set plotted in front of the expected number of visits for perfect performance at visual working memory capacities M={1,2,3}; (b) the actual error rates observed.

Figure 8.8(b) illustrates the error rates for each level of *n*. The large increase in error rate at 5 and 7 items appear to correspond well with the difference in predicted and measured numbers of visits at these levels of *n*. More development of the model is necessary to establish the relationship between the number of visits and error rate.

8.5 Conclusion

A practical implication of the results of the experiment is summed up in the following guideline:

> *Use multiple windows when visual comparisons must be made that would otherwise encourage storage of more items in visual working memory than its capacity. For the general population, this means multiple windows should be used when three or more items must be compared at a time.*

The results presented in this chapter additionally lend support to the ideas underlying the performance model presented in the previous chapter. One of these underlying ideas is that of breaking a task up according to visits. Another is the link between these visits and visual WM, which was given further support in the role that blocked verbal WM played in reducing performance in the *zoom* condition.

The results also highlight a deficiency in the model with regard to errors, namely that it does not predict the likelihood of error in the performance of a task. However, further analysis has shown that error rates increase as the difference increases between numbers of visits subjects made and numbers of visits predicted by the model, at least in the *zoom* condition. The next chapter investigates how linking the number of visits to expected error rates might extend the model, and uses special eye-tracking hardware to measure the number of visits in the *multi* condition.

CHAPTER 9

MEASURING EYE MOVEMENTS TO ESTABLISH A RELATIONSHIP BETWEEN VISITS AND ERRORS

The experiment in the previous chapter provided empirical support for the model of navigation performance presented in Chapter 7, but it also highlighted a key weakness: an inability to account for errors. The model assumes error-free performance, yet error rates in the experiment reached as high as 40%. Furthermore the error rate differed between *zoom* and multiple-window (*multi*) conditions. To account for the errors actually measured, it was hypothesized that subjects were making fewer visits than necessary to guarantee perfect performance. However, it was impossible to know how many visits were made in the *multi* condition because visits between windows were made with the eye rather than the mouse.

In this chapter, a modified version of the experiment was run with an eye-tracking device, making it possible to rerun the *multi* conditions with a new set of subjects and observe the number of visits subjects made by moving their eyes from window to window. The goal was to determine how many eye movements (eye-visits) were actually made and use this data to refine the performance model described in Chapter 7.

The results of the new experiment indicated that subjects made more eye-visits under the multi-window interface than zoom-visits under the zooming interface. Subjects also made fewer errors using the multi-window interface, roughly in line with the number of errors subjects made with the same interface in the prior experiment. These results led to the development of an extension to the performance model described in Chapter 7 that allows the model to account for the observed error rates and the expected number of visits, in addition to task completion time. The key to this extension is the development of a new cognitive model for how visual information is stored in visual WM based on two components:

1. a reinterpretation of the data from the work of Vogel et al. [2001] that allows for partial memory of objects, and
2. a cost for remembering which locations have been compared during the course of the task.

The performance model, extended using these concepts, is applied in a post-hoc comparison with the empirical results.

9.1 Eye-Tracked Experiment

An experiment was designed to measure the number of eye-visits subjects made during a multiscale comparison task using multiple windows. Of most interest was the number of times each user glanced between the window displaying the sample set and the window displaying a probe set. In order to make such measurements, an eye tracker was employed that had the capability of resolving where a subject's gaze was with respect to the various windows on the screen.

9.1.1 Method

The task for the new experiment was the same as the prior experiment: a 2D multiscale comparison task with one sample set and six probe sets, with the object being to find the lone probe set that matched the sample set. The biggest differences between the two experiments were the use of the eye-tracker and the elimination of the *zoom* condition in the new experiment. The remainder of this section provides the details of these differences.

9.1.2 The Multiple-window Navigation Mechanism

The basic navigation mechanism for this experiment was the same as for the multi-window condition of the previous experiment, however window creation was different for most of the subjects. Window creation occurred exactly as before for the first two subjects, with newly created windows appearing in the upper left corner of the screen at a size too small to be useful. However, for the remaining subjects, each window was created at a usable size and location so that no window management was necessary.

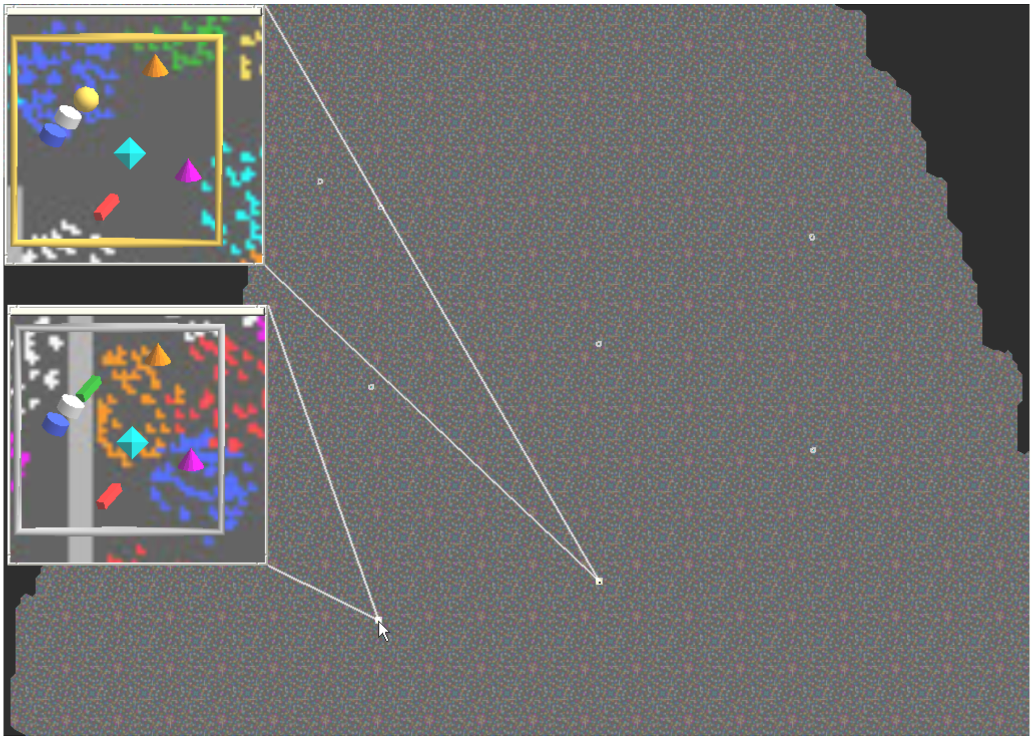


Figure 9.1: The default window sizes presented to subjects 3 through 10 in the experiment, relative to the rest of the screen.

The change in method of window creation was made for two reasons. First, it was done to speed the rate at which useful data could be obtained, because window management took a lot of the subjects' time, and overall task completion time was not an important measurement for this experiment. Second, the eye-tracking device had limited accuracy that required about 40 pixels of space between the windows in order to be certain as to which window was being visited. The layout of the windows as they appeared upon creation is illustrated in Figure 9.1.

9.1.3 Design

Each subject was trained on 8 representative trials, and was then presented with 6 blocks each containing 8 different trials in a 4x2 factorial design. The two factors were

- *n*, the number of objects in each set, chosen from {1, 3, 5, 7}, and
- *b*, whether verbal WM was *blocked* or *unblocked*.

As in the prior experiment, each experimental block was split into two groups such that all *blocked* trials were grouped together and all *unblocked* trials were grouped together within the block. The groups were counterbalanced across the six trial blocks and the order of the four values for *n* varied randomly within each subgroup. If a subject were to complete every trial without error, that subject would have encountered six trials for each of the eight conditions, for a total of 48 trials. Subjects generally completed more trials because trials that ended in error were repeated.

For each trial, the location and composition of each sample and probe set was randomized according to the constraints described in Section 8.2.

9.1.4 Apparatus

The eye tracker used was a Quick Glance 2S model from EyeTech Digital Systems. This system required that the subject's head remain still, so a chair modified with a specialized headrest was used for this purpose. Figure 9.2 illustrates how the equipment was arranged. The chair was located such that a subject's eye was between 60cm and 69cm from the screen. The visible area on the screen was between 36cm and 40cm. This produced a horizontal field of view subtending 33˚± 4˚.

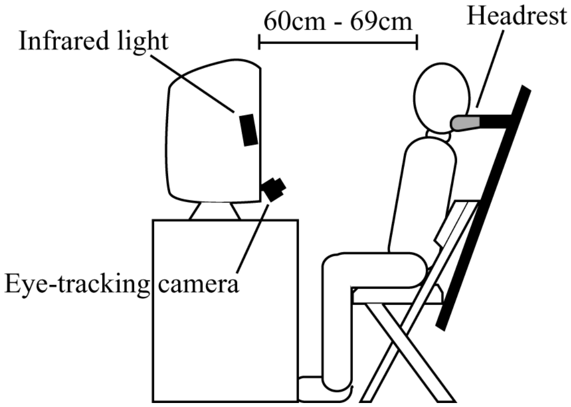


Figure 9.2: Eye-tracking equipment, monitor, and chair with headrest (not drawn to scale).

The EyeTech Digital Systems tracker delivered eye gaze information at a rate of about 25 Hz with a precision of roughly 20 pixels (about ½˚), although tracking tended to drift more than ½˚ throughout a session, reducing precision to approximately 1˚. To compensate, the eye tracker was calibrated to each subject before training and between blocks 3 and 4 (to maintain accuracy within ½˚). More accurate calibration was not critical to the study because it was only necessary to determine which window a subject was looking in, and the windows could be spaced far enough apart so as to eliminate ambiguous measurements.

9.1.5 Measurement

For the purposes of measurement, an eye-visit (or just *visit*) to the object set viewed by a subwindow was defined as the detection of a subject fixating on (or very near) that subwindow after either

1. The subject had just been fixating on the other subwindow, or
2. The subject moved the focus of the subwindow to a new object set.

In other words, a visit was recorded whenever the subject's eye made a saccade from the one subwindow to the other, or whenever the probe-set subwindow was moved to a different probe set. Eye movements back and forth between a subwindow and the overview did not count as visits unless the subject navigated the subwindow to a new probe-set.

If during a trial eye-tracking information was lost for more than two seconds at a time, the trial was summarily terminated, and the trial was repeated. Trials terminated in this fashion were considered incomplete and were not considered in the analysis.

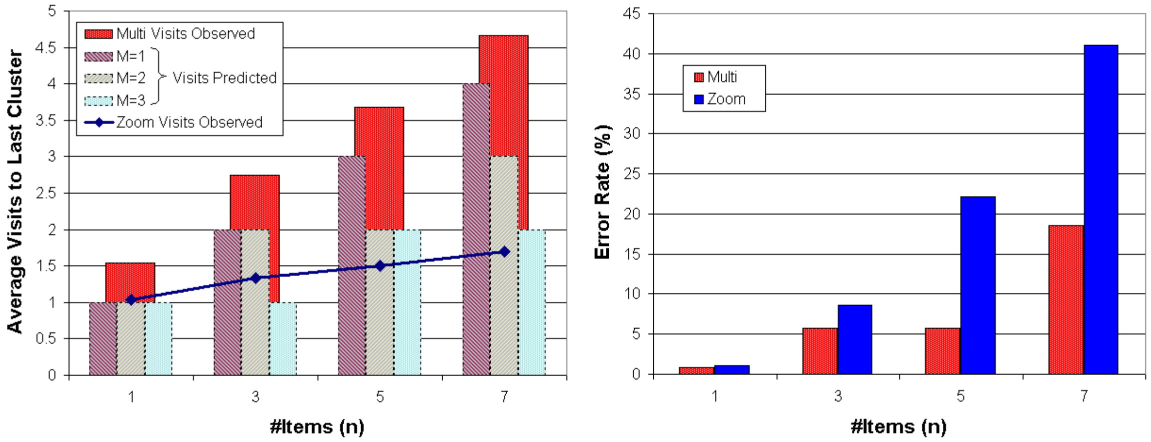
9.1.6 Subjects

The experiment was run on 10 subjects: 5 male and 5 female.

9.2 Results

A total of 523 trials were completed, of which 497 produced data deemed valid for analysis. Blocks 4 through 6 (24 successful trials and 2 error trials) of one subject were discarded due to poorly calibrated tracking. This left 480 – 24 = 456 successfully completed trials, plus 41 completed trials in which the subject made an error and had to repeat the condition.

Figure 9.3 summarizes the results. The background bars in Figure 9.3(a) illustrate the average number of visits made (with the eye) to the last probe set for each probe-set size. The foreground bars show the predicted number of visits required to achieve perfect performance assuming capacities of working memory at 1, 2, and 3 objects, calculated using the method described in the previous chapter. For comparison, the foreground line illustrates the average number of visits made in the *zoom* condition of the prior experiment.



(a) (b)

Figure 9.3: The number of visits to the last probe set investigated and the number of errors made, versus the number of items in the sample set: (a) actual number of visits made with the eyes to the last probe set plotted behind the expected number of visits for perfect performance at visual working memory capacities M={1,2,3}, with visits made in the *zoom* condition shown as a line on top of everything else; (b) the actual error rates observed for both conditions.

The results show that for the multi-window condition, subjects *over-visited* the last probe set—the average observed number of visits exceeded the model prediction in all cases. Even assuming that subject only held a single object in working memory as they looked back and forth between the sample- and probe-set windows, they made more eye movements than necessary.

Figure 9.3(b) illustrates the error rates for each level of *n* in the multi-window condition alongside the same error rates for the *zoom* condition of the prior experiment. Even though it appears that over-visiting has occurred in the multi-window condition, there are still significant errors with 7 items. However, the error rate in the multi-window condition is still much lower than that of the *zoom* condition.

Figure 9.4 illustrates how the new error rates for the multi-window condition compare against the error rates from the prior experiment. The results are relatively close at all set sizes except 7. One possible reason for the large difference is the large error contribution of two subjects who took less time (and perhaps less care) than the rest of the subjects did in looking at the contents of the lower window when 7 items were in a set: 6.2 seconds and 8.4 seconds, respectively, where the average was 11.7 seconds. Without these two subjects, the error rate for the current experiment at 7 items would have been 13.5%.

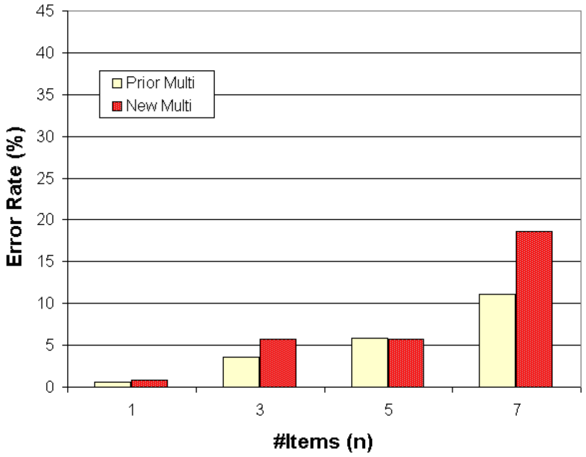


Figure 9.4: Comparison of error rates between the multi-window condition of the prior experiment and current experiment.

To determine whether or not verbal WM was a significant factor in error rates, an analysis of variance was performed with observed number of visits as the dependent variable and verbal WM blocking as an independent variable. No significant difference in error rates was found.

9.3 <u>Discussion</u>

The results show that subjects made dramatically more visits back and forth with the eye between windows containing the last cluster and the sample set than they made with the zooming interface. In addition, subjects made more eye-visits (in the multi-window condition) than the model predicted would be necessary to achieve perfect performance, yet still made errors.

The increase in visits in the multi-window condition over the *zoom* condition inversely correlates with errors to an extent, but why did subjects make so many more visits in the *multi* condition and what were the sources of error? Were subjects utilizing the same amount of visual WM capacity in both conditions, or did they make more visits in the *multi* condition because they were using less of their visual WM capacity?

Questions such as these drove the development of a new model of visual WM as well as the extension of the time-cost prediction model from Chapter 7 to account for errors. The key is to move from an error-free, capacity-based view of visual WM to one in which memory is not always accurate and the amount remembered depends in part on the number of objects attended to for memory.

9.4 Modeling Visual WM by Allowing for Partial Memory of Objects

The model of visual WM on which the work of Vogel et al. [2001] is based uses a formula derived from Pashler [1988] to transform errors observed during a sequential comparison task into a capacity for visual WM. This formula assumes "complete" memory of items. In other words, it assumes a constant, integer number of "slots" in visual WM.

But there is no evidence to indicate that visual WM operates in such a discrete manner. What if there is no "complete" memory of any items? "Partial" memory of items would help explain why errors occurred even when there was only one item to remember in the simple experiments of Vogel et al. [2001]. Could it be possible that we remember a little bit of everything we see? The results of the experiments of Vogel et al. make it clear that there is some limit on the number of objects about which information can be held, but they do not make it clear that this information is discrete.

This section lays out an alternative model of visual WM based on the idea of the partial memory of objects. It first defines what is meant by partial memory, and then shows how this definition was applied to reinterpret the results from Vogel et al. This reinterpretation yields a new cognitive model with a concrete mathematical representation that can be used for modeling. The section that follows shows how to apply the new cognitive model to account for both the observed error rates and number of visits made in both experiments.

9.4.1 Partial Memory of Objects

Partial memory of an object is defined as follows: A person has remembered x_m of an object if the object can be correctly identified or distinguished from other objects (x_m • 100)% of the time beyond chance in the context of a given task. This can be calculated from empirical evidence by noting how often a subject correctly identifies an object. If a subject has made correct identifications (x_i • 100)% of the time over a number of identifications, and x_c is the probability of making a correct identification by chance, then the x_m is more formally defined as follows:

$$x_m = \frac{x_i - x_c}{1 - x_c},\ 0 \le x_c < 1. \qquad \textbf{(9.1)}$$

x_m is undefined if x_c is 1.

To understand why Formula 9.1 has the form it does, consider the following possibilities. First, if a subject were to perform at chance, then $x_i = x_c$, and $x_m = 0$. In other words, if the subject is using any memory to perform the task, it is not having any effect on the ability to correctly identify the objects presented. Conversely, if a subject were to perform perfectly, then $x_i = 1$, and $x_m = 1$. In other words, the subject's memory is perfectly affecting identification. Note that for simplicity this notion of memory includes every part of the identification from perception of the object to registering the appropriate identification.

Consider an example where a subject can only correctly identify three of the four objects presented, and answers randomly when presented with the fourth. When each object has an equal chance of appearing, the subject answers correctly 81.25% of the time (x_i = 0.8125). This is because the probability of a correct identification is 1.0 three quarters of the time and 0.25 1/4th of the time (1 • 0.75 + 0.25 • 0.25 = 0.8125). The chance of guessing right by chance is simply x_c = 0.25. Using the formula, this would mean that 0.75 of an object is remembered: (0.8125 - 0.25) / (1 - 0.25) = 0.75. This correlates well with an intuitive understanding of how the subject is performing: the subject is perfect in 75% percent of the cases, but performs at chance otherwise. If instead the object that the subject guesses about appears half of the time, x_i = 1 • 0.5 + 0.25 • 0.5 = 0.625, and x_c does not change, meaning that only half of an object is remembered: (0.625 - 0.25) / (1 - 0.25) = 0.5. This demonstrates how the definition covers a broader range of possibilities in a uniform way that can be applied to a wider variety of tasks than the multiscale comparison instance considered in this chapter.

The definition of *partial memory of n objects* is simply the sum of the partial memories of each of the *n* objects involved. If it is assumed that an equal amount is remembered about each object, then the amount remembered can be written as follows:

$$x_m(n) = n \cdot \left(\frac{x_i - x_c}{1 - x_c} \right), \ 0 \le x_c < 1, \ \ n \ge 1 \,. \tag{9.2}$$

Several modeling advantages come from defining partial memory of objects in this way. It is easier to talk about the amount stored in visual WM without having to deal with “capacity” in terms of “complete” objects. The amount stored in visual WM can now be a non-integer value for individuals (rather than for average capacity over several individuals). In addition, by speaking of partial memory of objects in terms of *some* *number of objects* (*n*), it becomes possible to parameterize visual WM storage according to the number of objects over which it is spread—the amount stored in visual WM can now vary with the number of objects that are the focus of attention.

9.4.2 Storage in Visual WM Parameterized by Amount Attended

In order to develop an error model for the multiscale comparison task, the notion of partial memory of objects was used to reinterpret the results of Vogel et al. [2001]. Under a partial-memory interpretation, the amount stored in visual WM and error rate are no longer independent quantities, but are instead two ways of looking at the same basic capability. Visual WM becomes a "fuzzy" system that can hold a few items with high fidelity, or can hold 3-4 items-worth of lower fidelity information about more objects.

Figure 9.5(a) presents a summary of results from experiments of Vogel et al. [2001] (experiments 1, 2, 4, 10-14, and the test-cue condition of experiment 5), given as an error rate rather than an accuracy rate. The bold, dashed line indicates the average error rate over these experiments for each set size (1—4, 6, 8, and 12), while the boundaries of the shaded area indicate the maximum and minimum error rates across experiments. The lack of shaded areas at one and three objects appear because there were only two data points for these set sizes.

To have a way of interpolating data to model error rates for memory of partial objects, a continuous function was fit to the data. The most appropriate way to do this for situations in which the dependent variable is dichotomous (for instance, correct versus incorrect identification) is to perform logistic regression[1] on the original data. Logistic

[1] For more information on logistic regression, see Tabachnick and Fidell [2001] and Hosmer and Lemeshow [1989].

regression fits a curve over the range (0, 1) on the dependent variable (incorrect versus correct). However, the expected response did not fit the range of the identification results, but instead the range (0, 0.5)—the range of the associated dichotomy of perfect performance ("memory") and chance. In order to properly fit this range, the data was transformed according to a modified logit function before regression. The logit function is defined as logit(p) = ln(p/(1 – p)), where p is a probability and $0 < p < 1$. The function represents the probability of an event over the probability of its opposite. If this is interpreted as the probability of answering due to working memory versus the probability of answering due to chance, then a slightly different function can be defined: tlogit(p) = ln(p/(.5 – p)). The results of this transformation are shown in Figure 9.5(b).

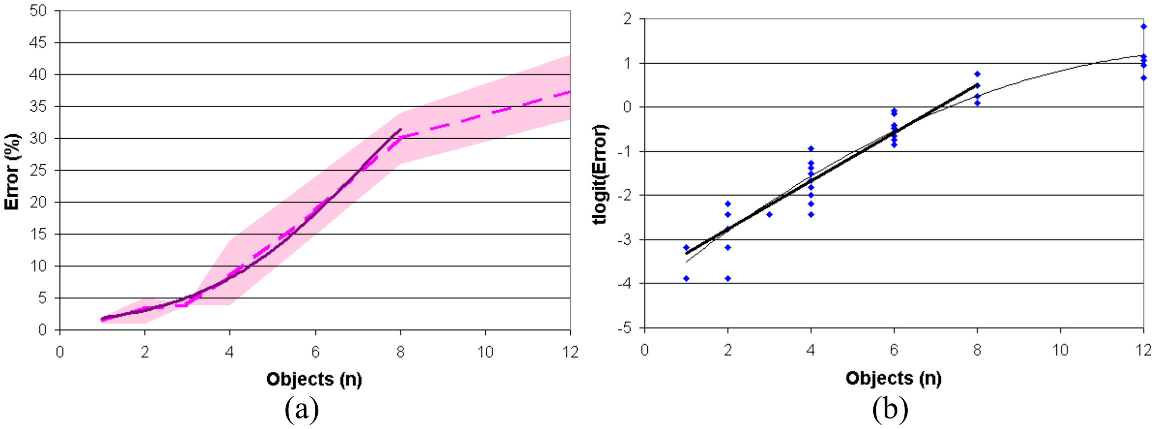


Figure 9.5: A partial-object re-interpretation of results from Vogel et al. [2001]: (a) likelihood of error when presented with a varying number of objects—the dashed line indicates average error and the solid curve indicates the fit from a linear regression of the logit for $n \leq 8$; (b) logit of the likelihood of error for each result from Vogel et al. [2001]—the bold line is a best fit line for $n \leq 8$, while the lighter curve is a best-fit quadratic on all the data.

Figure 9.5(b) highlights a problem with using the data from results where $n = 12$. Notice that the best fit is a quadratic (thin curve) rather than a line when all the data is plotted, but a line is more appropriate when the $n = 12$ results are excluded. Since a quadratic fit on the tlogit would transform back in to a curve that tended toward 0% error as n went to infinity, it seemed best to stick to a linear fit. A linear fit seems appropriate for the data with $n \leq 8$, but seems less appropriate with all the data. There is evidence (such as from Jiang et al. [2000]) that we cannot even remember the position of twelve objects well, let alone what they looked like, so it may be that there is a significant change in the way visual WM performed above eight items. In addition, the experiments described in this chapter presented a maximum of seven items. For these reasons, the results involving twelve objects were dropped from further consideration in the reinterpretation.

A linear regression was performed on the data where $n \leq 8$, resulting in the bold line in Figure 9.5(b). The regression intercept was roughly –3.8 and the regression coefficient for n was roughly 0.54. This regression line was then transformed back into range of measured error as appropriate for logistic regression:

$$\hat{x}_i = 1 - \frac{0.5 \cdot e^{(-3.8+0.54n)}}{1 - e^{(-3.8+0.54n)}}. \qquad \textbf{(9.3)}$$

This equation is illustrated by the bold curve in Figure 9.5(a).

With a continuous estimate for x_i, it was possible to use Formulas 9.2 and 9.3 to estimate x_m (the amount that can be stored in visual WM) given the number of items attended to. All that was required was the following simplifying assumption: subjects in the sequential comparison experiments tried to remember a little about every object they were presented. This allowed the substitution of Formula 9.3 into Formula 9.2. With the realization that $x_c = 0.5$, the following estimate for x_m was made.

$$\hat{x}_m(n) = n \cdot \left(1 - \frac{e^{(-3.8+0.54n)}}{1 - e^{(-3.8+0.54n)}}\right), \; n \geq 1. \qquad \textbf{(9.4)}$$

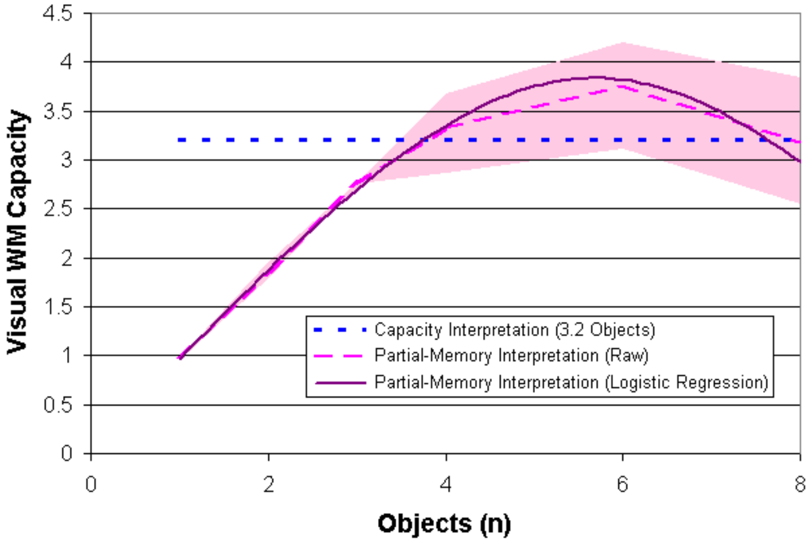


Figure 9.6: The amount that can be stored in visual WM when presented with a varying number of objects. The dashed line indicates the new partial-memory interpretation using the raw data, flanked by the bounds of the raw data. The solid curve indicates the new partial-memory interpretation using the estimate from Formula 9.4. The horizontal dotted line indicates a whole-object interpretation with a 3.2 object capacity.

Figure 9.6 shows how the amount stored visual WM varies with the number of items according to the new model, as manifested in Formula 9.4. For comparison, the horizontal dotted line represents a constant capacity of 3.2 objects (one of the capacities given by Vogel et al. [2001]), and the dashed line represents the averages of the raw data run through Formula 9.2 (bordered on either side by the minimum and maximum data values run through the same formula). Note that if this function were believed to be a reasonable approximation of the amount that can truly be held in visual WM, then there would rarely be reason to design a system that required the user to remember more than 6 items from one fixation to the next.

In summary, the new interpretation of the results of Vogel et al. [2001] match well with the "capacity" interpretation when errors can be disregarded yet it also provides a way to account for error. As can be seen from Figure 9.6, the new interpretation of visual WM still behaves as though there is a capacity of 3-4 items as long as one does not try to remember more than 8 items at a time. In addition, it is easy to extract from Formula 9.3 an expression for the probability of error (E_M) in a given use of visual WM, because $E_\mathrm{M} = 1 - x_i$.

$$E_\mathrm{M} = \frac{0.5 \cdot e^{(-3.8+0.54n)}}{1 - e^{(-3.8+0.54n)}} . \qquad \textbf{(9.5)}$$

9.4.3 The Cost of Remembering What Has Been Done

While the partial-memory interpretation of visual WM is important in accounting for errors, it is not the whole story. The multiscale comparison task places higher demands on visual WM than the sequential comparison task used by Vogel et al. [2001]. The sequential comparison tasks allowed only one "visit" to each of the sample and probe sets and did not require any manual navigation. However, the multiscale comparison task allows multiple visits between sample and probe sets using manual navigation techniques, and furthermore involves multiple probe sets. In order to perform the task efficiently, a subject must remember two or three additional pieces of information regarding the location of items:

1. which probe sets have already been seen and discounted as possible matches;
2. which probe is the one currently under investigation (*zoom* condition only); and
3. which objects have already been compared in the current probe.

Note that the objects under investigation in the current probe are already accounted for by the sequential comparison task. In addition, there may be demands on visual WM due to the particular manual navigation technique used.

No work was found in the literature to provide guidance as to how any of the additional demands on visual WM should be accounted for, however the error results illustrated in Figure 9.3(b) provide some guidance. For trials in which object sets contained only one item, resulting error rates were small (< 1.1%) regardless of the interface. These error rates resembled what one could expect from a simple sequential comparison task with one item (≤ 2%). Therefore, it seems reasonable to ignore information that deals with remembering which probe sets have been seen, which one is currently under investigation, and even what demands the interface might incur on visual WM.

The remaining possibility for contribution to error in the model is a cost for remembering the objects that have already been compared in the current probe (item 3 in the above list). The way that was chosen to treat the memory load was to use an additive factor equal to some proportion *w* of the objects in a set past the first one. In other words, if there are *n* items in a set, then the cost n_p of remembering one's place in the set is given as the equivalent amount of storage it requires in visual WM:

$$n_p = w \cdot (n - 1) \qquad \textbf{(9.6)}$$

The value of *w* should obviously be between zero and one, but there is no basis on which to choose its value other than choosing it to fit the data from the multiscale comparison experiments. A reasonable fit was found around $w = 0.46$.

9.5 Accounting for Error in the Multiscale Comparison Task

The new interpretation of visual WM storage provides two major benefits for modeling. First, it helps to explain why subjects might choose to make more visits in the multi-window condition than in the *zoom* condition. If the chance of error increases with the number of objects remembered, and if subjects were somewhat aware of this effect, then it would make sense to remember as few items as possible. However, with the desire to finish the task quickly, there is a disincentive to making many visits in the *zoom* condition because each visit takes a significant amount of time—there is a sort of point of diminishing returns from the perspective of being more careful. Thus, it makes sense to assume that subjects might remember only about one object at a time and make more (inexpensive) visits in the multi-window condition in order to minimize errors.

The second benefit to modeling is that it provides a way to account for errors: visual WM is not perfect, and its accuracy depends on how much is remembered. The new interpretation makes it possible to model the likelihood of error on a particular probe set based on the number of items being remembered per visit. This section describes a way to model error rates on the multiscale comparison task and then applies them post-hoc to the experimental results.

9.5.1 Propagating Within-Probe Error to Task Error

One way to model the error for the multiscale comparison task is to sum the likelihood for errors at each major decision point in the course of the task. The major decision points come when a subject identifies a cluster as matching (by pressing the space bar, for example) or as differing from the sample set (by moving on to investigate another cluster). Such decisions can then be put in terms of the likelihood of error in the use of visual WM. In other words, it becomes possible to derive an expression for the expected error rate involving E_{M} (from Formula 9.5).

Starting from the top, the first thing to do is to model the multiscale comparison task in terms of the major decision points. For any probe set visited, there is a chance that the probe set is a match, and a chance that it is not. In either case, the subject can be right or wrong. Let each case be denoted as follows:

- *a* for a matching set,
    - *c* for a correct match,
    - *f* for a false rejection of a match;
- *b* for a set that does not match,
    - *r* for a correct rejection, and
    - *e* for an erroneous acceptance of a set that does not match.

Furthermore, let x_σ be the probability of case σ occurring, given all the preceding cases, where $\sigma \in \{c, e, f, r\}$.

Figure 9.7 illustrates the dependence of cases and their probabilities of occurring given the cases preceding them. This illustration makes clear several properties of the task. Both *c* and *e* result in immediate termination of the task, while *f* and *r* allow it to continue. An *f* case (false rejection) cannot be followed by an *a* case (presentation of a matching set), because the matching set has already been rejected. Therefore it is always followed by a *b* case. Notice also that there are multiple *b* cases at every cluster between 2 and *p* (where *p* is the total number of object clusters). Some of these can be combined because they have the same probability of occurring and the same fate. These cases have been circled with an indication of how many identical cases are represented. At the bottom of Figure 9.7 is a special case (coded as a remnant possibility) that represents the case in which the subject has rejected all clusters. Presumably, if this case were to be reached in reality, the subject would assume they missed the matching set and would essentially start the task over.

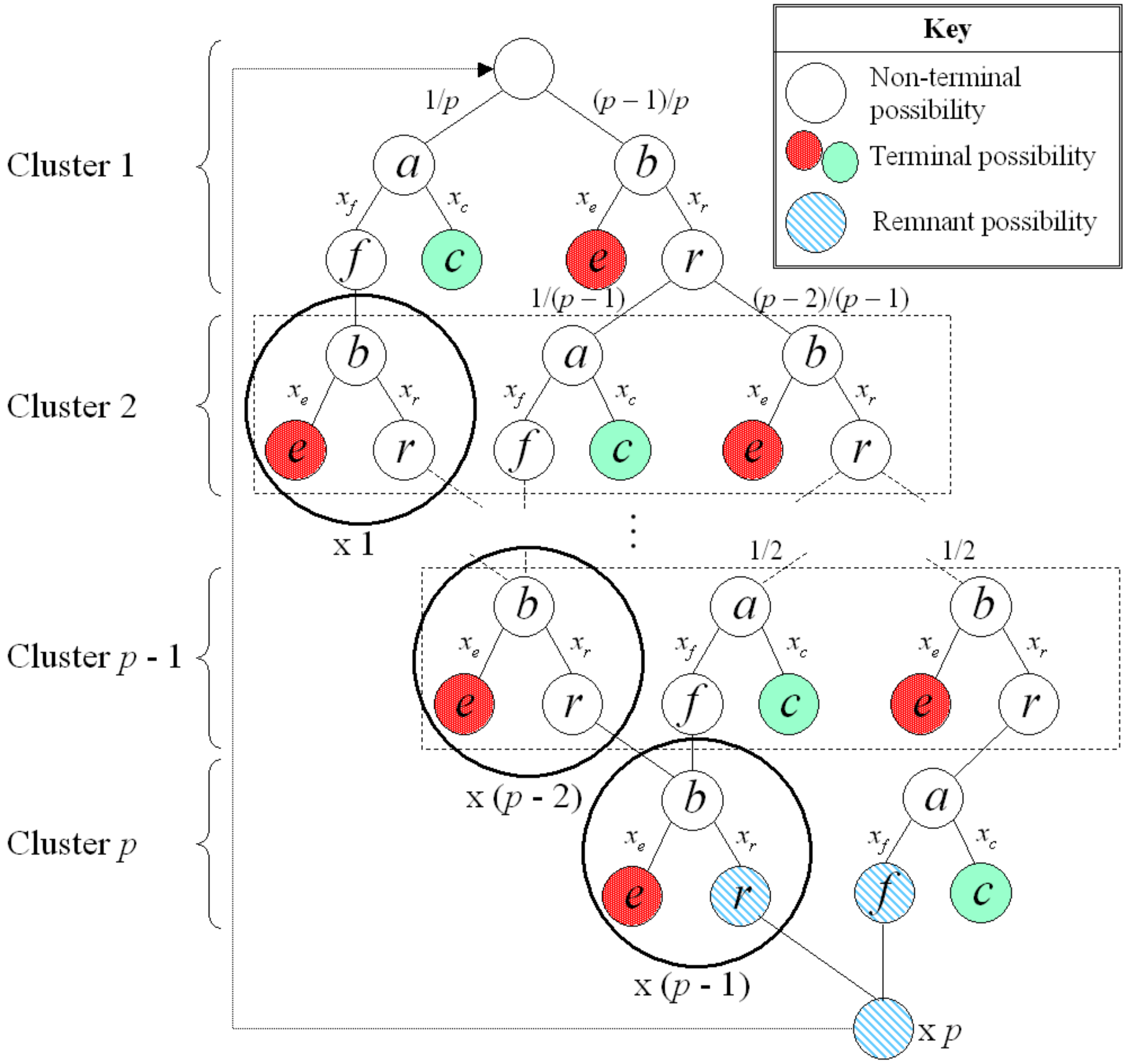


Figure 9.7: Possible states in the multiscale comparison task, and their local probabilities for a task involving *p* object clusters.

With this picture of the probabilities in place, it becomes relatively easy to sum up the probabilities. The probability of a given case can be calculated by multiplying along each link to the root. This works even when a case has multiple parents because it represents a combined case in which the probabilities are the same along any path. While it would be possible to add up the probabilities for all the *e* cases individually, the easiest route to find the total error is to find the remnant (*R*) and the sum of the probabilities of the *c* cases (*C*), and then subtract these from one. The remnant *R* is the easiest to compute: $R = x_f \bullet (x_r)^{p-1}$. The probability of a correct response is $C = \sum_{i=1}^{p} \frac{x_c (x_r)^i}{p}$. Thus, the probability of error without any rejected cluster coming under investigation again is $1 - C - R$. If it assumed that the ratio between correct responses and erroneous responses is the same in the remnant (further passes through the task) as it was in the first pass, then the probability of error E in the multiscale comparison task can be estimated as follows:

$$E = 1 - \frac{C}{C + (1 - C - R)} = 1 - \frac{C}{1 - R} = 1 - \frac{\sum_{i=1}^{p} \frac{x_c x_r{}^i}{p}}{1 - x_f \cdot x_r{}^{p-1}}. \qquad \textbf{(9.7)}$$

By filling the details into Formula 9.7, a model can be built for errors in the specific case of the task as configured in the experiments. The number of probe sets p is 6. For simplicity[1], let it be assumed that $x_f = x_e = E_\mathrm{M}$, and that $x_c = x_r = 1 - E_\mathrm{M}$. Thus, for the specific case of the experimental setting, E can be reduced essentially to a ratio of polynomials:

$$E = 1 - \frac{6 - 21E_\mathrm{M} + 35E_\mathrm{M}{}^2 - 35E_\mathrm{M}{}^3 + 21E_\mathrm{M}{}^4 - 7E_\mathrm{M}{}^5 + E_\mathrm{M}{}^6}{6 - 6E_\mathrm{M} + 30E_\mathrm{M}{}^2 - 60E_\mathrm{M}{}^3 + 60E_\mathrm{M}{}^4 - 30E_\mathrm{M}{}^5 + 6E_\mathrm{M}{}^6}. \qquad \textbf{(9.8)}$$

With this formula in place, all that remains is to substitute the appropriate E_M according to Formula 9.5.

[1] This assumption tends to overestimate error for smaller values of n. If more were known about the types of error that occurred in the experiments of Vogel et al. [2001], then the error could be split into E_MM, the probability of error when the sets match, and E_MD, the probability of error when they differ. Then $x_f = E_\mathrm{MM}$, $x_e = E_\mathrm{MD}$, $x_c = 1 - E_\mathrm{MM}$, and $x_r = 1 - E_\mathrm{MD}$. If $E_\mathrm{MD} < E_\mathrm{M}$ as estimated in Formula 9.5 (Vogel et al. informally reported that errors were three times as likely when the sets differed than when they were the same), then the model would predict lower values, especially at $n = 1$.

9.5.2 Modeling Error Rates for Zooming and Multiple Windows

At the beginning of this section (Section 9.5), an argument was made for assuming that subjects chose to remember multiple items per visit during *zoom* conditions in the experiment and one item during multi-window conditions. In order to find the value of m that best approximated the observed error rates in the *zoom* condition, the data for this condition was modeled twice: once with $m = 3$, and once with $m = 2$. When only one object was present in each cluster, the value of $m = 1$ was used for the obvious reason. The observed error rates for the multi-window condition were modeled using $m = 1$.

Table 9.1: Calculation of the expected error rates for the task at various set sizes under *zoom* and multi-window conditions.

	All	*Zoom* (m = 3)			*Zoom* (m = 2)			*Multi*		
n	1	3	5	7	3	5	7	3	5	7
m	1	3	3	3	2	2	2	1	1	1
n_p	0	0	1.84	2.76	0.92	1.84	2.76	0.92	1.84	2.76
n_s	1	3	4.84	5.76	2.92	3.84	4.76	1.92	2.84	3.76
E_M	1.8%	5.1%	11.7%	16.7%	4.8%	7.6%	11.3%	3.0%	4.7%	7.3%
E	4.7%	12.9%	29.4%	40.7%	12.4%	19.2%	28.5%	7.5%	12.0%	18.5%

Table 9.1 lists the results of each step required to compute the task error rate E from the number of items remembered per visit, m, and the number of items per cluster, n. First, m is added to the memory cost of remembering one's place in an object set n_p, given in Formula 9.6, to yield n_s—the effective amount of storage used in visual WM to perform the task. Then n_s is substituted into Formula 9.5 in the place of that formula's n to yield E_M. Finally, E_M is inserted into Formula 9.8 to yield E, the expected error rate for the task.

Figure 9.8 illustrates how these modeled values compare with the observed error rates and with each other. All the models grossly overestimate errors when there is only one item per cluster. The m = 1 model remains within seven percentage points of the observed error rates, while the m = 2 is off by as much as thirteen percentage points (at n = 7) and the m = 3 model is off by as much as just over seven percentage points (at n = 5). While the models do not match the measured results closely, they do provide a good estimate for how the error rates relate to each other between the *zoom* and multi-window conditions. In other words, the error model predicts that the error rates should be roughly equal for one object, but that error rates should rise faster for the *zoom* condition than for the multi-window condition.

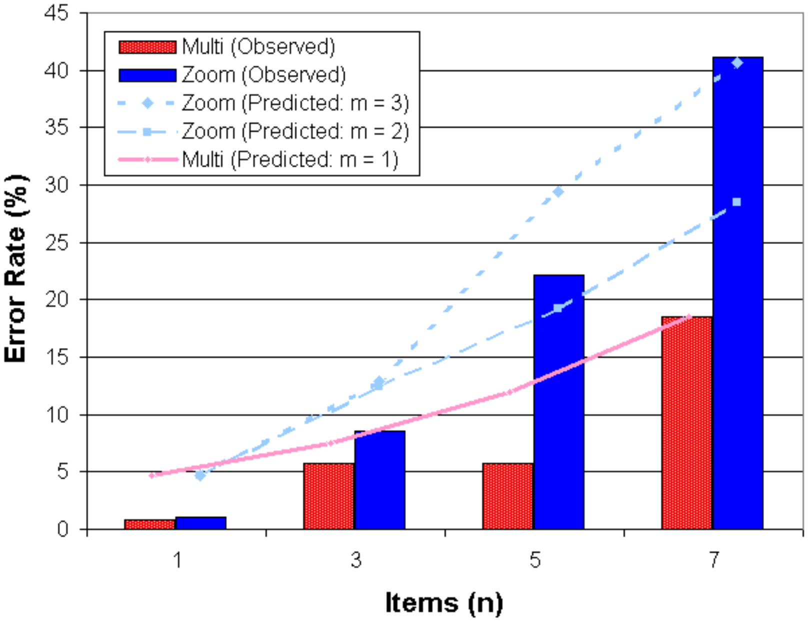


Figure 9.8: Modeled error rates versus observed error rates.

9.5.3 Worthwhile Direction for Further Model Development

The model presented to this point is as complete as is reasonable given the empirical evidence at hand. However, further analysis points to a fruitful direction for further development. The ability of the model to predict error would be strengthened if it were better known how subjects chose the amount to store in visual WM at each visit. It seems reasonable that there should be a tradeoff between total task time and allowance for error: as the number of objects increases, the subject may try to remember more objects per visit so as not to increase by much the number of visits required to make comparisons. Remembering more objects per visit would increase the likelihood of error but decrease the total time required to make the visits.

To investigate how much the error model would benefit if the precise nature of this tradeoff was known, the model was run with a different assumption for *m*. The assumption for the proper value of *m* is based on the numbers of visits observed in the experiments: subjects remembered just enough in each visit to the final cluster to cover all *n* items in the cluster, and the same amount was remembered on each visit. This assumption was applied by solving Formula 8.1 for *M* while ignoring ceiling operations in light of the possibility of remembering partial objects:

$$m = M = \frac{n}{2 \cdot V_{matching-probe} - 1}, \qquad \textbf{(9.9)}$$

Table 9.2 lists the results of using Formula 9.9 to model the observations.

Table 9.2: Calculation of the expected error rates for the task at various set sizes under *zoom* and multi-window conditions using *m* inferred from the empirical data.

Condition:	*Zoom*				*Multi*			
Set Size:	1	3	5	7	1	3	5	7
$V_{matching\text{-}probe}$	1.03	1.33	1.51	1.70	1.54	2.75	3.68	4.66
m	**0.94**	**1.80**	**2.48**	**2.92**	**0.48**	**0.67**	**0.79**	**0.84**
n_p	0	0.92	1.84	2.76	0	0.92	1.84	2.76
n_s	0.94	2.72	4.32	5.68	0.48	1.59	2.63	3.60
E_M	1.8%	4.4%	9.4%	16.2%	1.4%	2.5%	4.2%	6.8%
E	4.5%	11.3%	23.8%	39.7%	3.6%	6.3%	10.8%	17.2%

Figure 9.9 illustrates how this model compares with the observed data. The model still severely overestimates error in the single-item condition. However, differences between modeled error and observed error are reduced to within five percentage points across conditions. This indicates that it might be possible to make better predictions if it could be determined how subjects choose the number of objects to remember on a visit.

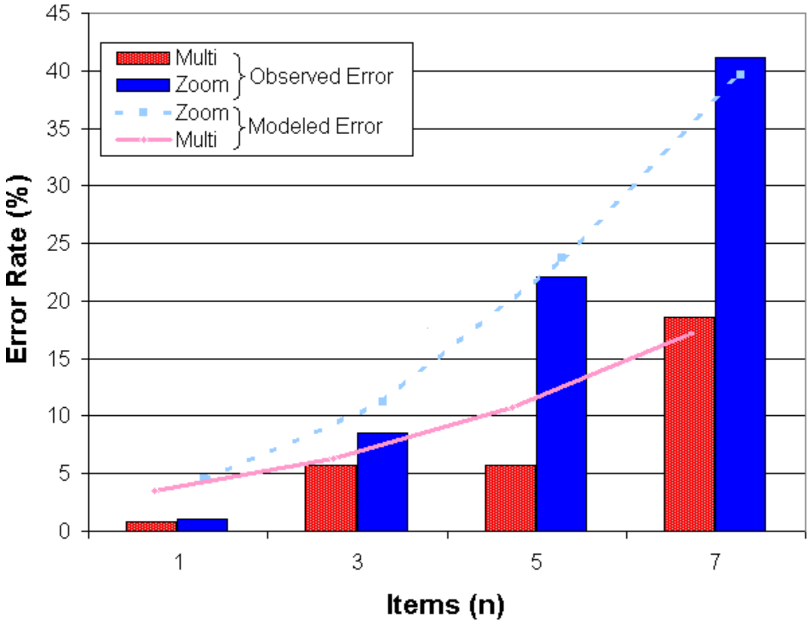


Figure 9.9: Modeled error rates based on observed items per visit, versus observed error rates.

9.6 Conclusion

This chapter has presented an experiment intended to shed some light on the error rates seen in the previous experiment, as well as a new cognitive modeling concept inspired by the results of the experiment. Conclusions of the experiment and analysis using the new modeling concepts are as follows:

1. At a given set size, people who made more visits tended to make fewer errors.

2. The model suggests that storage in visual WM is not based on a capacity for distinct objects but instead accommodates storage of more “objects” at the cost of higher error. Those who took more visits apparently remembered fewer items per visit at higher fidelity than those who took fewer visits, and therefore made fewer errors.
3. Subjects’ behavior seems to take into account the tradeoff between the number of items attended to for storage and the accuracy of that storage in order to balance overall task time. The data suggests that people remember one item or less when making comparisons using eye movements between windows, while people remember more when making comparisons using a zooming interface. For modeling time performance with the model from Chapter 7, it should therefore be assumed that $M = 1$ for multiple windows and $M = 2$ or $M = 3$ for zooming. Note that a choice of $M = 1$ for the multiple-window condition does not significantly impact the time predictions for that condition.

The results and the new cognitive theory together suggest that the multiple-window interface is superior to the zooming interface in terms of accuracy because visits can be made more quickly, and therefore more visits can be made in which fewer items are stored in visual WM with greater accuracy.

A further implication of the new cognitive model and the use of visits to model user performance is that new operational procedures or new interfaces could be designed to decrease error rates. Consider a procedure example. If one wanted to limit user error rates on the multiscale task used for the experiments, then one could specify (or build into the interface) that a minimum number of visits be made to a probe set before it could be deemed a match. This might be accompanied with a directive to concentrate on only one object at a time. Such a procedure, if properly enforced, would skew the user's time-error tradeoff to produce more accurate performance. As an example of an interface concept, consider how useful it might be to mark and/or mask the items have already been compared or are not even under consideration yet (making the items of interest most prominent). Such an artifact might aid the memory of the items of interest to the point that the half-item penalty discussed in Section 9.4.4 need no longer apply. Another interface example would be to automatically add extra windows when comparisons are to be performed, or to provide such windows permanently if the application primarily supports comparison tasks.

In summary, the key finding of this chapter is that subjects store less information in visual WM when making eye movements between locations (using a multiple-window interface) than when navigating between locations by zooming. As a consequence, the subjects make fewer errors when using the multiple-window interface than they do with the zooming interface. This suggests that other lightweight navigations (such as mouse-hovering, short-distance mouse scrolling, or instant navigation to a desired location like hypertext) may also encourage users to store less visual information in visual WM at a time and therefore make fewer errors. Conversely, it suggests that interfaces requiring more time required to navigate from place to place (as with zooming, large-scale scrolling, or menu-driven navigation interfaces) may encourage users to store more visual information in visual WM to reduce overall task time, thereby increasing the likelihood of error.

Zooming is extremely limited as a solution to the focus-in-context problem, because it works best when the mental image of only one or two objects is all that must be retained from one visit to the next. When more complex comparisons must be made, it is essential to display focus and context simultaneously, as with multiple windows. This allows a user to make rapid eye movements back and forth between them, comparing only a small amount of visual information at a time. The results reported here show that subjects make an eye movement for every object in a multiple-window display, even when these objects are very simple. This results in reduced errors, and is the reason that the multi-window display is superior for all but the simplest of visual comparison tasks.

CHAPTER 10

CONCLUSION

In the introduction, the question was asked, “How can 3D multiscale spatial information be presented so that people can better comprehend it?” This question was refined by rephrasing it in terms of the focus-in-context optimization problem: How can user speed and accuracy be maximized for a given task under the constraints of limited computer display space for focus and context information, extremely limited attentional capacity, and a limited ability to interact with the computer display?

This dissertation has attacked the focus-in-context problem on two fronts: practice and theory. On the side of practice, a set of interaction techniques and a software framework for realizing those techniques has been presented, collectively termed frame-of-reference interaction. Frame-of-reference interaction provides a new way of designing views to display focus and context information and linking these views in ways the user can readily understand. On the side of theory, a cognitive systems model has been developed that takes into account both key cognitive limitations, such as visual working memory, and key interface capabilities. An example interface capability is navigation: must the user navigate from one focus location to another with the mouse or can the user simply look back and forth with the eyes? This model provides guidelines for designers so that interfaces can be created that are more efficient and less prone to user error

Through the process of answering the central question in terms of focus and context, two substantial contributions have been made to the field of human-computer interaction. The first is the development of the *frame-of-reference interaction* techniques and software framework, which provide an interface designer with ways of partitioning focal and contextual information into multiple windows, as well as interaction techniques for linking the information in those windows. The second major contribution is a *cognitive systems model* that provides rigorous guidance as to when multiple windows are more or less appropriate than other navigation techniques.

10.1 Frame-of-reference Interaction: Techniques for Linking Multiple Windows

The central idea behind frame-of-reference interaction is to assign a geometric frame of reference (FoR) to each 3D view and to objects in the 3D scene, and then establish meaningful relationships among them (through couplings and frame-of-reference operations) to create linked focal and contextual views. A frame of reference encapsulates components of translation, rotation, and scale with respect to some origin. For 3D views, this origin is located at a place called the *center of workspace*, conceptually at arms length from the viewer. Meaningful relationships are established either by constraining certain components of two FoRs to change in tandem (through couplings) or by performing operations on them to create new frames of reference (through FoR-ops). The rest of frame-of-reference interaction consists of visually representing these reference frames and the meaningful relationships between them.

The techniques of frame-of-reference interaction provide a novel way of integrating zooming and multiple windows into a single interface, as well as providing a unified approach to linking focus with context through multiple 3D windows. Of particular novelty is the way a view can be attached to a moving object simply by indicating a desire to navigate to it (by clicking the middle mouse button on it), and the ability to track groups of objects while still providing the user with degrees of freedom in the viewpoint taken on a group.

The frame-of-reference interaction software framework supports the frame-of-reference interaction techniques by implementing geometric frames of reference, and by providing ways to couple these reference frames together and perform operations on them to create new reference frames. The couplings and frame-of-reference operations make it possible for windows to feature not just static locations in the scene, but moving objects, groups of objects, and groups of objects that meet certain interest criteria. Such capabilities make it possible to unify all of the interface techniques presented while providing a rich medium for exploring new techniques.

Frame-of-reference interaction has been implemented in GeoZui3D, providing support for a number of practical applications. The applications to which it has already contributed include the following:

- *Planning, monitoring and control of underwater robotic vehicles*. Work with Autonomous Undersea Systems Institute (AUSI) has been done to develop multi-windowed interfaces for AUV planning and monitoring. GeoZui3D has also been integrated into separate projects with the Woods Hole Oceanographic Institute (WHOI) and Johns Hopkins University for monitoring and controlling ROV progress as a tool in underwater archaeology and exploration. In every case, the approach is to link local vehicle perspectives with contextual overview perspectives in a way that facilitates better planning, monitor, and control tasks.

- *Outreach exhibit.* The frame-of-reference interaction software framework provides the interaction engine in a 3D interactive exhibit at the Seacoast Science Center at Odiorne Point State Park in New Hampshire. The exhibit allows novices to explore various areas near New Hampshire's Seacoast. The approach is to create locations of interest (represented by software reference frames), and provide the level of interactivity most appropriate to the exhibit-user for navigating between these locations.
- *"Chart of the Future" Project.* Frame-of-reference interaction is central to the development of new visualization and interaction techniques related to safety of navigation and planning for mariners and marine scientists in a project with the National Oceanic and Atmospheric Administration (NOAA). The goal is a revolutionary advance in the stat-of-the-art in electronic charting technology. Part of the approach involves identifying the most important and most commonly needed perspectives on the available data to carry out navigation, planning, and scientific tasks, and then linking these perspectives in such ways as to provide a seamless context for each task.

The facet of frame-of-reference interaction that is ripest for further development is in the discovery of new useful operations on multiple reference frames (FoR-ops). This is because a FoR-op is a useful way of codifying a complex geometric relationship between objects into a single reference frame that can then have a window attached to it.

One obvious operation that could be developed is one that mimics the "dynamic tethers" implemented by Wang and Milgram [2002], but allows for more interactive flexibility in the view. Other possibilities include operations that identify autonomous vehicles surveying the furthest frontiers of a particular survey area, operations that identify the largest gaps between vehicles in a survey fleet (to indicate where more vehicles may be needed), and operations that allow a human operator to monitor the area through which a vehicle is planned to transit over the next 20 minutes.

Another facet of frame-of-reference interaction that deserves further investigation is the use of tether lines to link a focus window with its proxy representation in the wider context. Experiments should be carried out in a wide array of situations to determine what situations merit the use of tethers, and in what situations it suffices to use proxy representations without tethers. Good candidates include situations involving many focus windows, and situations in which the user must occasionally monitor a focus window, but whose primarily task does not involve the focus window at all.

10.2 Cognitive Systems Model: When to Use Multiple Windows

The cognitive systems model presented in Chapters 7 and 9 makes it possible to predict how quickly and accurately users can complete particular multiscale tasks using a given interface. This model applies well to tasks that require a user to synthesize information from a number of focus locations in the context of a larger multiscale environment. To account for the speed of completion on such a task, the model requires that the task be broken into *visits*—navigations to key focus locations in the virtual space and the work done at those locations before any further navigation. Three parameters involve expected times to perform some part of the task:

1. A constant overhead factor indicating what must be done regardless of the number of visits. This factor includes interface setup time, such as is required to resize additional windows.
2. A cost of navigation on each of these visits, such as the average time to zoom out from one location and into the next.
3. An expected amount of time required to do work at each visit.

The model combines these factors with the expected number of visits in a given task instance to calculate an expected total task time. It is the calculation of the expected number of visits that has proved most crucial in the comparison of zooming and multiple-window interfaces—the limited capacity of visual working memory implies that several visits between the same sets of locations are required in non-trivial comparison tasks, and the multiple-window interface incurs much lower costs than the zooming interface for such visits.

To account for errors in the completion of a given task, the model allows the task to be broken up according to major decision points—places where choices can be made that would result in the occurrence of an error. The essence of the model is simply to sum up the probability of error at each of these major decision points. However, many tasks can reach a point where the user realizes that a mistake (potentially recoverable error) has been made, and the task is essentially started over. In these situations, there is no clear termination point at which to stop the sum, and it is necessary to take a slightly different approach. In the absence of the possibility of a simple sum, the model breaks up the probability of error into three components:

1. The remnant probability of realizing an error and starting the task over again;
2. The probability of successfully completing the task before that point; and
3. The probability of making an unrecoverable error before that point.

The model combines these factors to estimate the likelihood of error in a given task instance. The best estimates are obtained from the model when it is known how many visits were involved with each decision point—because visual working memory is more error-prone as more items are stored, more visits leads to fewer items remembered, which leads to fewer errors.

The cognitive systems model is novel because of the level of abstraction that it uses to make its time and accuracy estimates. GOMS modeling (Goals, Operators, Methods, Selection Rules) begins at a low level (such key strokes and mouse movements) and builds a task automaton from selection rules and goals [Card et al. 1983]. Cognitive models like EPIC, SOAR, and ACT-R similarly build automata to describe how the human perceptual and visual system might interact with an interface toward a particular goal [Miyake and Shah 1999]. The cognitive systems model presented here requires some assumptions to be made about the overall user strategy, but uses high-level operations that allow the details of strategy execution to be hidden. The key to hiding the details is identifying the key high-level operations, and then performing computations on the expected costs of these operations, as opposed to modeling every conceivable detail.

Although the cognitive systems model requires less effort to apply than others, there are many times when a simple guideline may be sufficient to make a sound interface design decision. The model as applied in Chapters 7 and 9 suggest a simple guideline that is further supported by the experiments in Chapters 8 and 9:

> *Use multiple windows when visual comparisons must be made that would otherwise encourage storage of more items in visual working memory than its capacity. For the general population, this means multiple windows should be used when three or more items must be compared at a time.*

The human brain seems capable of only holding one or two simple items with high fidelity. Multiple windows allow visual patterns from distant locations to be compared side-by-side on the display. As the model and experimental results from Chapters 7 through 9 suggest, side-by-side comparison is crucial because it allows comparisons to be made by quick movements of the eye, rather than time-consuming interactions with the hand and mouse. The reduced time cost seems to make users more willing to be more careful in their visual comparisons.

The model has only been applied to one task and two interfaces in this dissertation, but it is applicable to many tasks and most navigation methods. For example, it can be used for tasks involving pattern matching or classification, where it is not known which pattern is the one that will match. While these tasks heavily involve visual working memory and visual comparison, the model should extend just as well to verbal working memory and mental calculation, comparison, or transformation. Examples of other navigation methods include fisheye views, Intelligent Zoom, page scrolling, and hypertext browsers.

A useful mathematical extension to the model would be to use probability distributions rather than expected values as inputs. If model computations were performed on probability distributions rather than single values, the prediction could be given as a distribution rather than a single value. Such a result would be more useful in making comparisons between interfaces because quantifiable confidence measures could be generated.

A useful extension to the model as applied to tasks involving visual working memory would be a way to determine how many visual items a person is likely to store at a time given the circumstances of the task. In Section 9.5.3, estimates were made as to the amount subjects stored in visual working memory based on the average number of visits they made to the last cluster of objects they visited. This only allows post-hoc analyses to be made. It would be more useful if the number of visits were predicted, based on a prediction of how many items a person is likely to store, perhaps based in turn on total task time or time between comparisons. Knowing how to predict the number of visits based on other information would enable better predictions to be made both for the expected amount of time to complete the task and the expected error rate. An experiment that could be used to determine how much a person is likely to store in visual working memory at a given time might take the form of a modified sequential comparison experiment, in which unlimited viewings of two sets of objects are allowed but the amount of time between successive viewings is varied between trials.

10.3 Final Remarks

So, back to the central question, "How can 3D multiscale spatial information be presented so that people can better comprehend it?" There are many considerations that must be taken into account to answer this question. Many good partial answers have been contributed by others in the field of human-computer interaction, and this dissertation puts forth its own partial answer intended as a complement to them. The partial answer that this dissertation puts forth can be summarized as follows.

Strongly consider using multiple windows in your interface. Multiple windows allow focal information to be organized on the screen as the user sees fit, and can usually be linked to a common context without too much computational or cognitive overhead. They allow for multiple 3D perspectives to be available simultaneously, making possible rapid switches in user attention between them. Creation, maintenance and deletion of windows should be as simple for the user as possible, potentially even involving some aspects of automatic system management. Efforts in linking windows should be concentrated on situations such as rotation that require great cognitive effort, since linkage mechanisms can usually eliminate the need for such cognitive effort at the cost of minimal additional perceptual effort. Modeling user attention according to geometric frames of reference can aid in the design and implementation of effective linking techniques.

If designing an interface for tasks with a high demand on a cognitive resource such as visual working memory, consider modeling alternative interface designs using the techniques described in Chapters 7 and 9. Keep in mind key cognitive constraints such as the storage capacity and reliability of visual working memory. Consider both efficiency with respect to minimizing time, and effectiveness with respect to minimizing errors, and consider empirically validating these models by running a few subjects on the modeled tasks. If such modeling is cost-prohibitive but the following guideline is applicable, then heed it: Use multiple windows whenever comparisons of 3 or more visual items are involved.

GLOSSARY

Absolute coupling—A coupling (see below) that locks together the values of a component of two reference frames so that they are always the same.

Center of workspace—A point just behind a computer display or display window, conceptually at about arms length, intended to act as the central focus for user interaction. The center of workspace coincides with the origin of an interaction reference frame (I-FoR).

Center-of-workspace interaction—A single-view interaction paradigm that uses the center of workspace as a basis for interaction.

Context information—Information of potential relevance to an observer. Context information often provides cues as to how focus information should be interpreted or combined.

Coupling—A mathematical constraint between components of two reference frames, such that certain changes in one cause certain changes in the other. Couplings can occur in individual components of a reference frame (such as position or heading), or can involve several attributes.

Focus information—Information of immediate relevance to an observer. A user's attention is focused on this information.

Focus-in-context problem—An optimization problem in which the goal is to maximize user performance (in terms of speed and/or accuracy) while minimizing the use of computer display space and human attentional resources.

Frame of reference (FoR)—A geometric construct consisting of position, orientation, and scale information. Used synonymously with *reference frame*.

Frame-of-reference interaction (FoRI)—An integrated set of techniques for the presentation and manipulation of reference frames, along with a software framework for realizing these techniques effectively. FoRI uses center-of-workspace interaction within individual views, and links these views with various linking mechanisms that operate on I-FoRs and O-FoRs.

Frame-of-reference operation (FoR-op)—A reference frame, also referred to as a resultant, that embodies the result of an algorithmic constraint between a group of operand reference frames. Changes in the operands cause changes to occur in the resultant according to the constraints of the FoR-op.

Interaction reference frame (I-FoR)—A reference frame that describes a designated focus of attention ("look-at" point) within a window. Its origin is located at the focus of attention such that its *y*-axis is pointing away from the observer, the *z*-axis is vertical with respect to the observer, and the *x*-axis is horizontal.

Linking—The act of relating information from different sources to obtain information that could not be discerned from any single source.

Localized coupling—A coupling that maintains the position of one reference frame at a fixed offset from another reference frame with respect to postion, orientation, and scale, regardless of how this latter reference frame changes.

Main zoomport—The root zoomport of the zoomport hierarchy.

Multi-perspective identification task (MP-ID)—A task that requires a person to combine information from one or more different forward-looking perspectives with an overview perspective to identify a target object.

Multiscale comparison task—A task that requires a person to compare probe object sets with sample object sets, within a large multiscale environment, to find the probe set that matches the sample set.

Multiscale information—Information that has relevant or important detail at both small and large scales.

Object reference frame (O-FoR)—A reference frame used to describe a potential target of attention, usually an object or group of objects. All objects have an associated O-FoR, but O-FoRs can also exist on their own.

Partial memory of an object—An empirical estimate of the likelihood that a person will correctly identify or distinguish an object from other objects, beyond chance.

Partial memory of *n* objects—The sum of the partial memories of each of the *n* object involved.

Relative coupling—A coupling that maintains a fixed difference between the values of a component of two reference frames so that they always remain at a fixed offset from each other in value.

Reference frame—See *frame of reference* (FoR).

Subwindow—A zoomport that has a parent zoomport.

Tethers—Connecting lines. Within the context of frame-of-reference interaction, these lines connect a zoomport with its zoomport proxy.

Visit—Under the navigation performance model, a transit (navigation) to a given location and the work done at that location before any visits to another location

Zoomport—A window that displays a 3D view of the scene, with the view specified by an I-FoR. Under frame-of-reference interaction, zoomports are organized in a parent-child hierarchy and can be visually and behaviorally linked.

Zoomport proxy—A visual device that represents the reference frame of one zoomport within the visual context of its parent zoomport.

REFERENCES

ARETZ, A. J. 1991. The Design of Electronic Map Displays. *Human Factors*, *33*, 1, 85-101.

ARETZ, A. J. AND WICKENS, C. D. 1992. The Mental Rotation of Map Displays. *Human Performance, 5*, 4, 303-328.

ARSENAULT, R., SMITH, S., WARE, C., MAYER, L. AND PLUMLEE, M. 2003. Fusing Information in a 3D Chart-of-the-Future Display. *U.S. Hydro 2003 CD-ROM Proceedings*.

BADDELEY, A. D. AND HITCH, G. J. 1974. Working Memory. In G. H. Bower (Ed.), *The Psychology of Learning and Motivation: Advances in Research and Theory*, *8*, New York, Academic Press, 47-89.

BADDELEY, A. D, THOMPSON, N., AND BUCHANAN, M. 1975. Word Length and the Structure of Short-Term Memory. *Journal of Verbal Learning and Verbal Behavior*, *14*, 575-589.

BARTRAM, L., OVANS, R., DILL, J., DYCK, M., HO, A., AND HAVENS, W. S. 1994. Contextual Assistance in User Interfaces to Complex, Time-Critical Systems: The Intelligent Zoom. *Proceedings of Graphics Interfaces '94*, Morgan Kaufmann, Palo Alto, 216-224.

BECKER, R. A. AND CLEVELAND, W. S. 1987. Brushing Scatterplots. *Technometrics*, *29*, 2, 127-142.

BEDERSON, B. B. AND HOLLAN, J. D. 1994. Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '94)*, ACM Press, 17-26.

BESHERS, C. AND FEINER, S. 1993. AutoVisual: Rule-based design of interactive multivariate visualizations. *IEEE Computer Graphics and Applications*, *13*, 4, 41-49.

CARD, S. K., MACKINLAY, J. D., AND SHNEIDERMAN, B. 1999. *Readings in Information Visualization: Using Vision to Think.* Morgan Kaufmann Publishers, San Fransisco, 1-34; 307-309.

Card, S. K., Moran, T. P., and Newell, A. 1983. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

CARD, S. K., ROBERTSON, G. G., AND YORK W. 1996. The WebBook and the Web Forager: An Information Workspace for the World-Wide Web. *Human Factors in Computing Systems CHI '96 Proceedings*, ACM Press/Addison Wesley, New York, 111-117.

CARPENDALE, M. S. T., COWPERTHWAITE, D. J., AND FRACCHIA, F. D. 1997. Extending Distortion Viewing from 2D to 3D. *IEEE Computer Graphics and Applications*, July/Aug, 42-51.

CHASE, W. AND CHI, M. 1979. *Cognitive Skill: Implications for Spatial Skill in Large-Scale Environments* (Technical Report No. 1). Pittsburgh: University of Pittsburgh Learning and Development Center.

CHUAH, M. C., ROTH, S. F., MATTIS, J., AND KOLOJEJCHICK, J. 1995. SDM: Selective Dynamic Manipulation of Visualizations. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '95)*, ACM Press, New York, 61-70.

COMBS, T. T. A. AND BEDERSON, B. B. 1999. Does Zooming Improve Image Browsing? *Proceedings of the Fourth ACM Conference on Digital Libraries*. ACM Press, New York, 130-137.

DARKEN, R. P. AND CEVIK, H. 1999. Map Usage in Virtual Environments: Orientation Issues. *Proceedings of IEEE Virtual Reality 99*, 133-140.

DOSHER, B. A., SPERLING, G., AND WURST, S. A. 1986. Trade-offs between Stereopsis and Proximity Luminance Covariance as Detereminants of Perceived 3D Structure. *Vision Research*, *26*, 6, 973-990.

EDELMAN, S. 1995. Representation of similarity in 3D object discrimination. *Neural Computation*, *7*, 407-422.

EDELMAN, S. AND BUELTHOFF, H. H. 1992. Orientation Dependence in the Recognition of Familiar and Novel Views of 3D Objects. *Vision Research*, *32*, 2385-2400.

ELEY, M. G. Determining the Shape of Land Surfaces from Topographical Maps. 1988. *Ergonomics*, *31,* 3, 355-376.

ELVINS, T. T., NADEAU, D. R., AND KIRSH, D. 1997. Worldlets—3D Thumbnails for Wayfinding in Virtual Environments. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '97)*, ACM Press, New York, 21-30.

FURNAS, G. W. 1986. Generalized Fisheye Views. *Proceedings of CHI '86*, ACM Press, 16-23.

FURNAS, G. W. AND BEDERSON, B. B. 1995. Space-Scale Diagrams: Understanding Multiscale Interfaces. *Human Factors in Computing Systems CHI '95 Proceedings*, ACM Press/Addison Wesley, New York, 234-241.

GOLDSTEIN, J. AND ROTH, S. F. 1994. Using Aggregation and Dynamic Queries for Exploring Large Data Sets. *Human Factors in Computing Systems CHI '94 Proceedings*, ACM Press/Addison Wesley, New York, 23-29.

GUO, H., ZHANG, V., AND WU, J. 2000. The Effect of Zooming Speed in a Zoomable User Interface. *Student CHI Online Research Experiments (SHORE)*, http://otal.umd.edu/SHORE2000/zoom/.

HOSMER, D. W., AND LEMESHOW, S. 1989. *Applied Logistic Regression*. John Wiley & Sons, New York.

HOWARD, I. P. AND CHILDERSON, L. 1994. The Contributions of Motion, the Visual Frame, and Visual Polarity to Sensations of Body Tilt. *Perception*, *23*, 753-762.

HOWARD, I. P. AND HECKMAN, T. 1989. Circular Vection as a Function of the Relative Sizes, Distances and Positions of Two Competing Visual Displays. *Perception*, *18*, 5, 657-667.

JIANG, Y., OLSON, I. R., AND CHUN, M. M. 2000. Organization of Visual Short-Term Memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *26*, 3, 683-702.

LAMPING, J., RAO, R., AND PIROLLI, P. 1995. A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. *Human Factors in Computing Systems CHI '95 Proceedings*, ACM Press, 401-408.

LEVINE, M., MARCHON, I, AND HANLEY, G. 1984. The Placement and Misplacement of You-Are-Here Maps. *Environment and Behavior, 16*, 2, 139-157.

MACKINLAY, J. D., CARD, S. K., AND ROBERTSON, G. G. 1990. Rapid Controlled Movement Through a Virtual 3D Workspace. Proceedings of SIGGRAPH '90 in *Computer Graphics*, *24*, 4 (August), ACM SIGGRAPH, New York, 171-177.

MACKINLAY, J. D., ROBERTSON G. G., AND DELINE, R. 1994. Developing Calendar Visualizers for the Information Visualizer. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '94)*, ACM Press, New York, 109-118.

MILGRAM, P. AND JODELET, D. 1976. Psychological maps of Paris. In H. M. Proshansky, W. H. Itelson, and L. G. Revlin (Eds.), *Environmental Psychology*. New York: Holt Rinehart & Winston.

MILLER, G. A. 1956. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *Psychological Review*, *63*, 81-97.

MIYAKE, A. AND SHAH, P. 1999. *Models of Working Memory: Mechanisms of Active Maintenance and Executive Control*. Cambridge University Press, New York.

NORTH, C., AND SHNEIDERMAN, B. 2000. Snap-Together Visualization: A User Interface for Coordinating Visualizations via Relational Schemata. *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI 2000)*, ACM Press, 128-135.

PALMER, S. E. 1999. *Vision Science—Photons to Phenomenology*. MIT Press, Cambridge, Massachusetts.

PALMER, S., ROSH, E., AND CHASE, P. 1981. Canonical Perspective and the Perception of Objects. In J. Long and A. Baddeley, Eds. *Attention and Performance IX*. Erlbaum: Hilldale, N.J., 135-151.

PARKER, G., FRANCK, G., AND WARE, C. 1998. Visualization of Large Nested Graphs in 3D: Navigation and Interaction. *Journal of Visual Languages and Computing*, *9*, Academic Press, 299-317.

PASHLER, H. 1988. Familiarity and Visual Change Detection. *Perception Psychophysics*, *44*, 4, 369-378.

PIERCE, J. S., FORSBERG, S., CONWAY, M. J., HONG, S., ZELEZNIK, R., AND MINE, M. R. 1997. Image Plane Interaction Techniques in 3D Immersive Environments. *1997 Symposium on Interactive 3D Graphics*, ACM Press, New York, 39-43.

PLAISANT, C., CARR, D., AND HASEGAWA, H. 1992. When an Intermediate View Matters: A 2D Browser Experiment. *University of Maryland Technical Report CS-TR-2980*, http://www.cs.umd.edu/Library/TRs/CS-TR-2980/CS-TR-2980.ps.Z.

PLUMLEE, M., KOMERSKA, R., ARSENAULT, R., WARE, C., AND MAYER, L. 2001. Visualization Techniques To Support Monitoring And Control Of Autonomous Platforms. *12th International Symposium on Unmanned Untethered Submersible Technology UUST '01*, CD Proceedings.

PLUMLEE, M. AND WARE, C. 2002a. Zooming, Multiple Windows, and Visual Working Memory. *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI 2002)*, ACM Press, New York, 59-68.

PLUMLEE, M. AND WARE, C. 2002b. Frame of Reference Interaction. *UIST 2002 Companion (Posters and Demos from the 15th Annual ACM Symposium on User Interface Software and Technology)*, 41-42.

PLUMLEE, M. AND WARE, C. 2003a. An Evaluation of Methods for Linking 3D Views. *Proceedings of ACM SIGGRAPH 2003 Symposium on Interactive 3D Graphics*, ACM Press, New York, 193-201.

PLUMLEE, M. AND WARE, C. 2003b. Integrating Multiple 3D Views through Frame-of-Reference Interaction. *Proceedings International Conference on Coordinated & Multiple Views in Exploratory Visualization (CMV 2003)*, IEEE, Los Alamitos, CA, 34-43.

RHODES, G. 1995. Face Recognition and Configurational Coding. In T. Valentine (ed.), *Cognitive and Computational Aspects of Face Recognition*. Routledge, New York.

RISCH, J. S., REX, D. B., DOWSON, S. T., WALTERS, T. B., MAY, R. A., AND MOON, B. D. 1997. The STARLIGHT Information Visualization System. *Proceedings of IEEE International Conference on Information Visualization*, London, 42-49.

ROBERTSON, G., VAN DANTZICH, M., ROBBINS, D., CZERWINSKI, M., HINCHLEY, K, RISDEN, K., THIEL, D., AND GOROKHOVSKY, V. 2000. The Task Gallery: A 3D Window Manager. *Human Factors in Computing Systems CHI 2000 Proceedings*, ACM Press/Addison Wesley, New York, 494-501.

ROBERTSON, G. G., AND MACKINLAY, J. D. 1993. The Document Lens. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '93)*, ACM Press, 101-108.

ROSKOS-EWOLDSEN, B., MCNAMARA, T. P., SHELTON, A. L., AND CARR, W. Mental Representations of Large and Small Spatial Layouts Are Orientation Dependent. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *24*, 1, 215-226.

SARKAR, M. AND BROWN, M. H. 1994. Graphical Fisheye Views. *Communications of the ACM,* 47, 12 (December), ACM Press, 73-84.

SCHAFFER, D., ZUO, Z., GREENBERG, S., BARTRAM, L., DILL, J., DUBS, S., AND ROSEMAN M. 1996. Navigating Hierarchically Clustered Networks Through Fisheye and Full-Zoom Methods. *ACM Transactions on Computer-Human Interaction, 3*, 2, 162-199.

SEIGEL, A. W. AND WHITE, S. H. 1975. The Development of Spatial Representations of Large-Scale Environments. In H. W. Reese (ed.), *Advances in Child Development and Behaviour*, Academic Press, London, 9-55.

SIMONS, D. J. 1996. In Sight, Out of Mind: When Object Representations Fail. *Psychological Science*, *7*, 301-305.

STOAKLEY, R., CONWAY, M. J., AND PAUSCH, R. 1995. Virtual Reality on a WIM: Interactive Worlds in Miniature. *Human Factors in Computing Systems CHI '95 Proceedings*, ACM Press/Addison Wesley, New York, 265-272.

TABACHNICK, B. G, AND FIDELL, L. S. 2001. *Using Multivariate Statistics*. Allyn & Bacon, Needham Heights, Massachusetts.

THORNDYKE, P. W. AND HAYES-ROTH, B. 1982. Differences in Spatial Knowledge Acquired from Maps and Navigation. *Cognitive Psychology*, *14*, 560-589.

TRIESMAN, A. AND GORMICAN, S. 1988. Feature Analysis in Early Vision: Evidence from Search Asymmetires. *Psychological Review*, *95*, 1, 15-48.

VOGEL, E. K., G. F. WOODMAN, AND S. J. LUCK. 2001. Storage of Features, Conjunctions, and Objects in Visual Working Memory. *Journal of Experimental Psychology: Human Perception and Performance*, *27*, 1, 92-114.

WANG BALDONADO, M. Q., WOODRUFF, A., AND KUCHINSKY, A. 2000. Guidelines for Using Multiple Views in Information Visualization. *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI 2000)*, ACM Press, 110-119.

WANG, W. AND MILGRAM, P. 2002. Viewpoint Optimisation for Navigation Using Dynamic Tether. *Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting*, 2164-2168.

WARE, C. 2000. *Information Visualization*. Morgan Kaufmann Publishers, San Francisco.

WARE, C. AND LEWIS, M. The DragMag Image Magnifier. 1995. *Human Factors in Computing Systems CHI '95 Companion*, ACM Press/Addison Wesley, New York, 407-408.

WICKENS, C. D. AND HOLLANDS, J. G. 2000. *Engineering Psychology and Human Performance, Third Edition*. Prentice-Hall, Inc., Upper Saddle River, New Jersey.

YAMAASHI, K., COOPERSTOCK, J. R., NARINE, T., AND BUXTON, W. Beating the Limitations of Camera-Monitor Mediated Telepresence with Extra Eyes. 1996. *Human Factors in Computing Systems CHI '96 Proceedings*, ACM Press/Addison Wesley, New York, 50-57.

APPENDIX A

MEASURING ZOOM RATE PREFERENCES

To better understand the limiting factors that contribute to an ideal zoom rate, an experiment was designed and executed to contrast preferred zoom rates under various conditions. The independent variables of this experiment were frame rate, data density, and scale disparity between target objects. Zoom rate was a dependent variable, chosen by experimental subjects according to what was most comfortable for each configuration of the independent variables. The results of the experiment indicate that scale disparity and frame rate have an effect on preferred rate of zoom, but that the effect is not linear or monotonic. Meanwhile, data density appears to have a very negligible effect, and differences between individual subjects seem to have the greatest effect of all. These results suggest that zoom rate should be held nearly constant, regardless of frame rate, but that users should be given the option to set a preferred rate of zoom. This chapter provides the details of the experiment and the analysis of its results.

A.1 <u>Experimental Task</u>

The experimental task for the experiment consisted of having a subject zoom arbitrarily deep into an infinite hierarchy of squares, illustrated in Figure A.1. At each scale, the hierarchy contained one parent square in which n child squares were randomly placed. Each child's width was related to its parent by $1/r$ ($1/r^2$ for area), where r was the independent variable for scale disparity. Exactly one of these squares was in turn the parent of *n* more squares. Squares at each layer were colored to be pre-attentively distinguishable from the immediate parent, cycling through red, yellow, green, and light blue. Animation during zoom operations occurred at a rigidly controlled frame rate, *f*. The subject's task was to "follow the information" by continuously zooming into squares with child squares nested inside them.

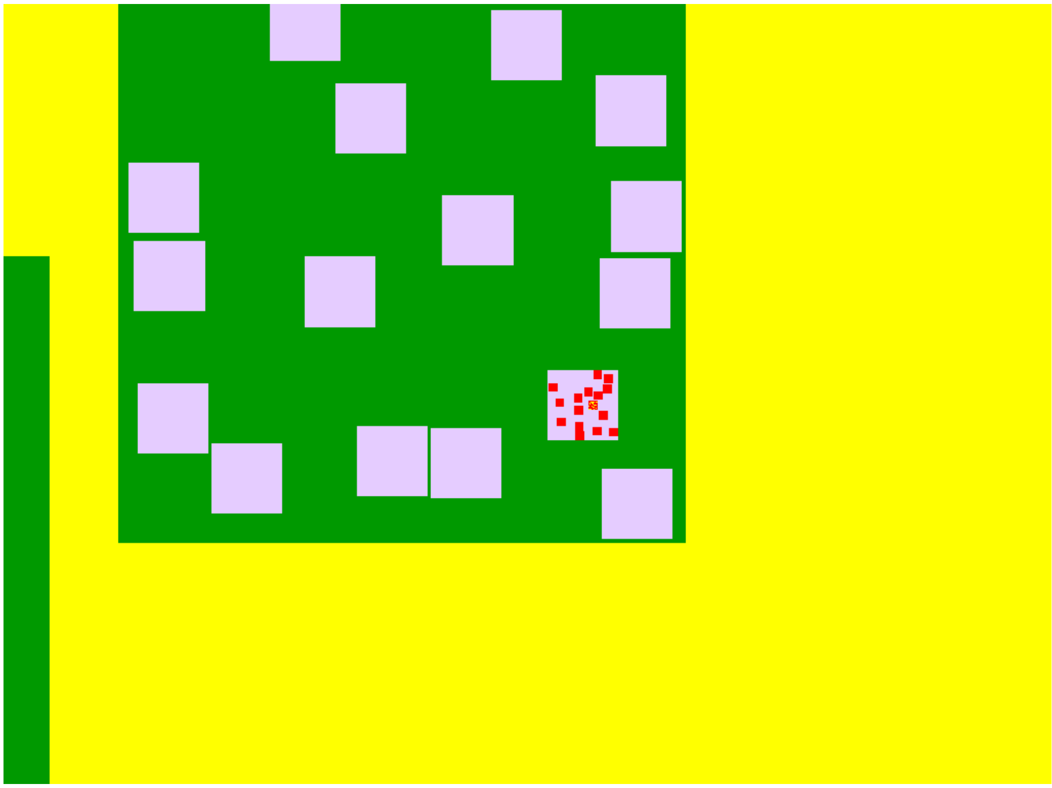


Figure A.1: Infinitely nested hierarchy of squares

The user interface consisted of a mouse with two buttons for zooming in and out, plus two arrow keys (up and down) on the keyboard for controlling zoom rate. To zoom in, the user positioned the mouse over a target location and held down the left mouse button. The center of zoom followed the mouse as long as the button was held, allowing the user to navigate continuously through the hierarchy. Zooming out occurred when the user pressed the right mouse button, but was always done about the center of the screen.

Subjects were required to move the cursor over the nested information as it emerged while adjusting the zoom rate to comfortable levels. Subjects could adjust the zoom rate simultaneously with zooming, or they could stop to make the adjustment. Subjects were first asked to adjust the zoom rate to a *minimum comfort threshold*, where they felt comfortable but not completely bored. They were then asked to raise the zoom rate to an uncomfortably fast rate, and then reduce the zoom rate to a *maximum comfort threshold*, the fastest rate at which they felt they could comfortably operate, indefinitely. A trial ended whenever a maximum comfort threshold was recorded.

A.2 **Experimental Variables**

Trials in the experiments varied in three parameters:

- Frame rate—the number of animation frames per second.
- Scale disparity ratio (r)—the ratio between the width of a square to that of its parent.
- The number (n) of child squares per parent, $n/r^2 \leq \frac{1}{4}$. The limit on n is due to the way the squares were randomly placed—it was possible that after the placement of the first n squares, there would be no room for any additional squares once $n/r^2 > \frac{1}{4}$.

Before running the experiment, the expectation was that a higher scale disparity would lead to faster comfortable zoom rates, because the time elapsed between decisions was greater. It was also originally hypothesized that frame rate and number of child squares would not have an effect on what subjects considered to be a comfortable zoom rate. The reason no change was expected on the number of child squares was simply that the squares were pre-attentively distinguishable from parent squares. The reasons for expectations on frame rate deserve more discussion.

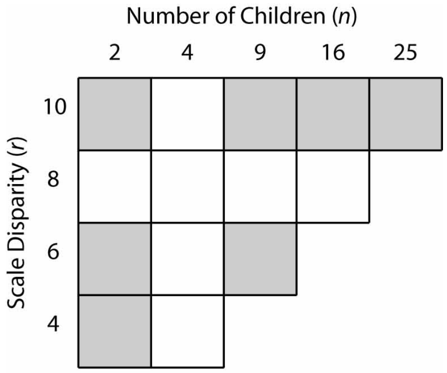
Some of the zooming interfaces that are described in the literature give anecdotal hints as to what a good zoom rate should be, but they put it in terms of frames per second. For instance, Point of Interest (POI) navigation [Mackinlay et al. 1990] does not use zooming per se, but it does use a constant (15% per frame) rate of motion that mimics zooming. However, the zoom rate is given in terms of animation frames without mention of the frame rate(s) maintained. For reference, this constant at 16 frames per second corresponds to a zoom rate of 9.4 x magnification per second (x/s), and at 60 frames per second yields a zoom rate of 4384 x/s. In any case, the authors of POI navigation indicate that a wide range of values is also acceptable to either side of this constant. NV3D, another system with a zooming interface, has a zoom factor of 5% per frame [Parker et al. 1998]. Other systems give the user control over zoom rate, for example with a slider.

The only formal evaluation of zoom rate we have found is a study by Guo et al. [2000] that used JAZZ, a successor to Pad++ to assess task performance with zoom rates between 1.5 x/s and 16 x/s. The study found no significant differences between task performance at 4 x/s and 16 x/s but a task performance times were longer for rates of zoom slower than 4x/s. However, this study only used a single fixed frame rate and it might be the case that having a constant magnification per *frame* would be the correct approach to setting zoom rates because we rely on frame-to-frame coherence for visual continuity. Alternatively, having a specific zoom rate per *second* may be a more valid way of setting zoom rates, because information will then scale at a constant rate. The expectation for the effects of frame rate were that, if frame-to-frame coherence was the main factor, comfortable zooming rates should fall with the log of the frame rate. If information flow was the main effect, frame rate should have little to no effect on comfortable zoom rate.

A.3 Experimental Design

Subjects were presented a set of 21 trials as illustrated in Figure A.2. Frame rate was varied as odd powers of two, (2, 8, and 32), to capture a wide range. Scale disparity varied linearly (4, 6, 8, and 10). The number of children varied quadratically (2, 4, 9, 16) so that the relative amount of space covered by the child squares varied linearly.

Subjects were given representative training trials to diminish learning effects. After training, all trials were presented in a random order such that each appeared to a subject exactly once. Subjects were given unlimited time to make their determinations. Zoom rate was recorded whenever the subject indicated that a minimum or maximum comfort level was reached.



(b) Phase II: $f = \{2, 8, 32\}$

Figure A.2: Values of experimental variables, highlighted in white within the array of possible values.

A.4 Subjects

Eighteen subjects participated in the experiment: 13 male and 5 female. The first five subjects tended to have a great deal of desktop computer experience and were all above the age of 25. The remaining thirteen subjects were generally undergraduate students with varying amounts of computer experience.

A.5 Results

Data was collected from 378 trials. Three trials were thrown out due to outrageously low (< 1 x/s) or outrageously high (> 80 x/s) maximum comfort thresholds, leaving a total of 375 trials.

A.5.1 Maximum Comfort Threshold

Figure A.3 summarizes the results regarding the maximum comfort threshold in terms of frame rate and scale disparity. While both variables were significant according to the analysis that follows, neither variable had a simple effect on the maximum comfort threshold zoom rate. Most notably, the effect of frame rate was not the same shape as the effect that could be expected from a model of optimal zoom rate that assumed the human visual system requires a minimal coherence between frames. This frame-coherence model is illustrated in the figure by a heavy dashed line. For comparison, a heavy solid line has been added to the illustration that indicates the average over all the scale disparities. Note that the range of mean maximum comfort thresholds only varied by about 30% above and below the overall mean. In other words, the maximum comfort thresholds remained remarkably constant given the wide range of frame rates, child square densities, and scale disparities.

Two analyses of variance were performed with the base-two log of the maximum comfort threshold as the dependent variable. The log of the zoom rate was used because of the exponential nature of zoom rate, and was a decision made before the collection of the data. One analysis of variance was performed on trials in which the number of child squares was 4 (the white column in Figure A.2), and the other was performed on trials in which the scale disparity was 8 (the white row in Figure A.2).

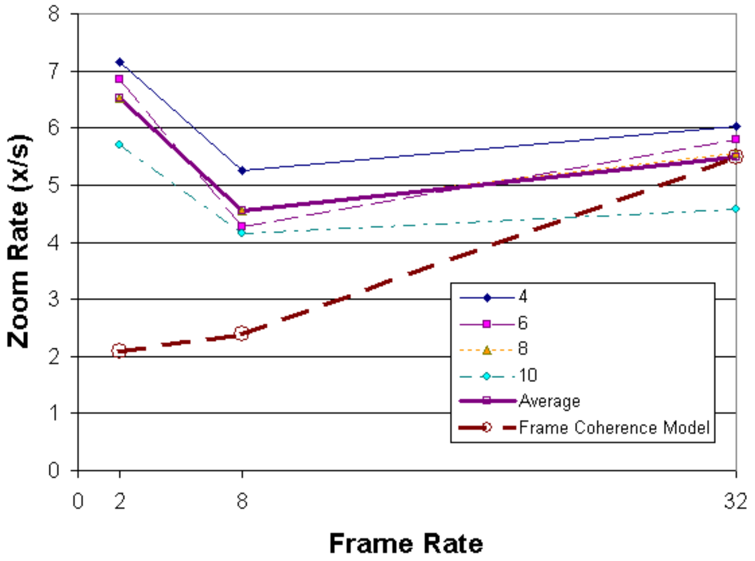


Figure A.3: Maximum comfort threshold zoom rate plotted against frame rate at different scale disparities.

The analysis of variance on all trials with 4 child squares involved frame rate and scale disparity as independent variables. This analysis found both variables to be significant, with the significance of frame rate at **F**(2, 34) = 10.73, **p** < 0.001 and of scale disparity at **F**(3, 51) = 4.09, **p** < 0.05. In addition, individual differences between subjects were highly significant (**F**(17, 45) = 16.31, **p** < 0.001), as were as the interactions of subject with either frame rate or scale disparity: for the interaction of subject and frame rate, **F**(34, 100) = 2.91, **p** < 0.001; for the interaction of subject and scale disparity, **F**(51, 100) = 2.00, **p** < 0.01.

The analysis of variance on trials having a scale disparity of 8 involved frame rate and number of child squares and independent variables. This analysis did not find the number of child squares per parent to be significant, but it again found frame rate, subject, and the interaction of subject and frame rate to be significant at the **p** < 0.001 level (**F**(2, 34) = 11.87, **F**(17, 36) = 20.02, and **F**(34, 100) = 2.94, respectively).

A.5.2 Minimum Comfort Threshold

Figure A.4 summarizes the results regarding the minimum comfort threshold in terms of the frame rate. Again, the effect of frame rate was not simple, and again the effect is not of the same shape as what could be expected from a model based on frame coherence (the heavy dashed line. Note that the range of mean minimum comfort thresholds only varied by about 20% above and below the overall mean.

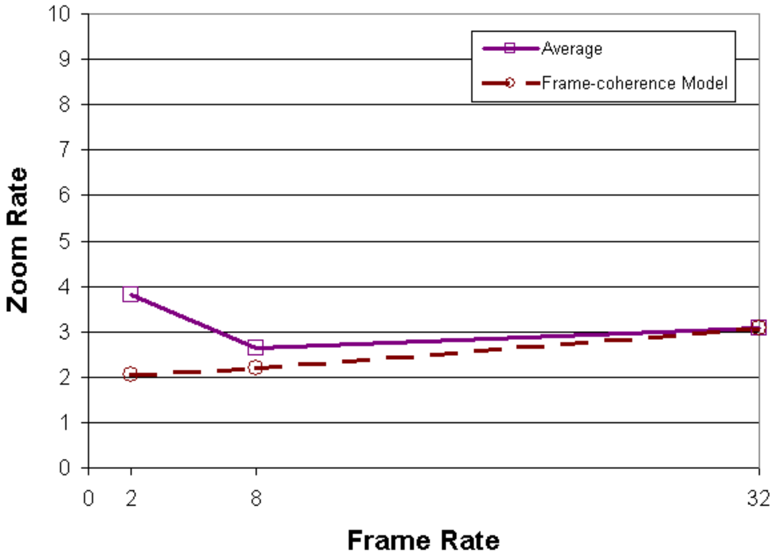


Figure A.4: Minimum comfort threshold zoom rate plotted against frame rate.

Two analyses of variance were performed with the base-two log of the minimum comfort threshold as the dependent variable, similar to the analyses performed on the maximum comfort threshold. Both analyses found only frame rate, subject, and their interaction to be significant. For the analysis involving 4 child squares per parent, frame rate, subject, and their interaction were all significant at the **p** < 0.001. For frame rate, **F**(2, 34) = 15.15; for subject, **F**(17, 32) = 15.68; and for their interaction, **F**(34, 100) = 3.02. For the analysis in which the scale disparity was held constant at 8, frame rate and subject were significant at the **p** < 0.001 level (**F**(2, 34) = 27.52 and **F**(17, 32) = 20.94 respectively), and their interaction was significant at **p** < .05 (**F**(34, 100) = 1.72).

A.5.3 Between Subject Differences

Figure A.5 illustrates the variations in both thresholds by subject. Averages were calculated using a geometric mean due to the exponential nature of zoom rate. Notice that the first five subjects tended to have higher maximum comfort thresholds than the rest. The overall average maximum comfort threshold was 5.5 x/s, while the overall average minimum comfort threshold was 3.1 x/s. Subjects' maximum comfort thresholds varied from a high of 23.0 x/s for one exceptional individual, to a low of 3.0 x/s. Minimum comfort thresholds varied from a high of 7.7 x/s to a low of 1.6 x/s.

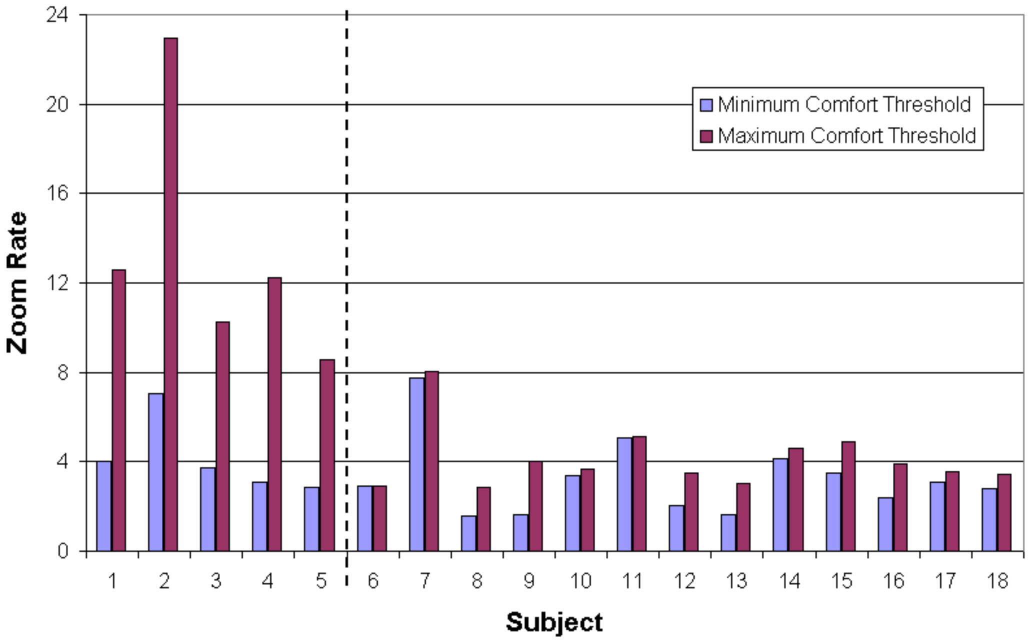


Figure A.5: The variation in comfort thresholds varied significantly by subject. Subjects who were generally older and more experienced with desktop computers appear to the left of the dashed line.

A.6 Discussion

The results suggest that while frame rate has a significant effect on preferred rate of zoom, its effect is not based on coherence between frames. In fact, a line representing a constant rate of zoom would better approximate the effect than a frame-coherence model. This suggests that zooming interfaces would do better to maintain a constant rate of zoom per time period than a constant rate of zoom per frame.

The results also suggest that scale differences between levels of detail play an important role in how quickly users are comfortable zooming. More study would need to be done to determine the nature of the relationship between scale disparity and comfortable zoom rate.

The results make it clear that user preferences in zoom rate vary widely. One-third of the subjects appear that they would be comfortable with a zoom rate of about 8 x/s, while nearly another half appear that they would be comfortable with a zoom rate of about 3 x/s. It would seem that there are distinct populations who prefer different rates of zoom, perhaps due to a certain level of experience with zooming, or to a certain level of more general eye-hand coordination (possibly correlating with age). Whatever the reason, any interface that relies on user-controlled navigation would do well to allow the selection of a zoom rate by the user. For a default value, it might be prudent to choose a zoom rate between 3 and 4 x/s for a novice user base, and a zoom rate between 7 and 8 x/s for an expert user base.

An interesting thing to note that deserves further study is the “dip” in the comfort thresholds in Figures 3 and 4 at eight frames per second. One hypothesis for this phenomenon is that there is a fundamental change in the strategy used for zooming between 2 frames per second and 8 frames per second. At two frames per second and below, the zooming task resembles a rapid-fire sequence of Fitts' Law tasks in which the limiting factor is how well the subject can move the cursor to the target area. At eight frames per second and above, motion is smooth enough that tracking would better describe user behavior, except that the subject must continually acquire new tracking targets at each level of detail in the hierarchy of squares. In this case, the limiting factor may be a combination of how quickly a user can acquire and track new targets with the eye, and how often the user decides that the mouse must be moved in order to keep zooming into the target.

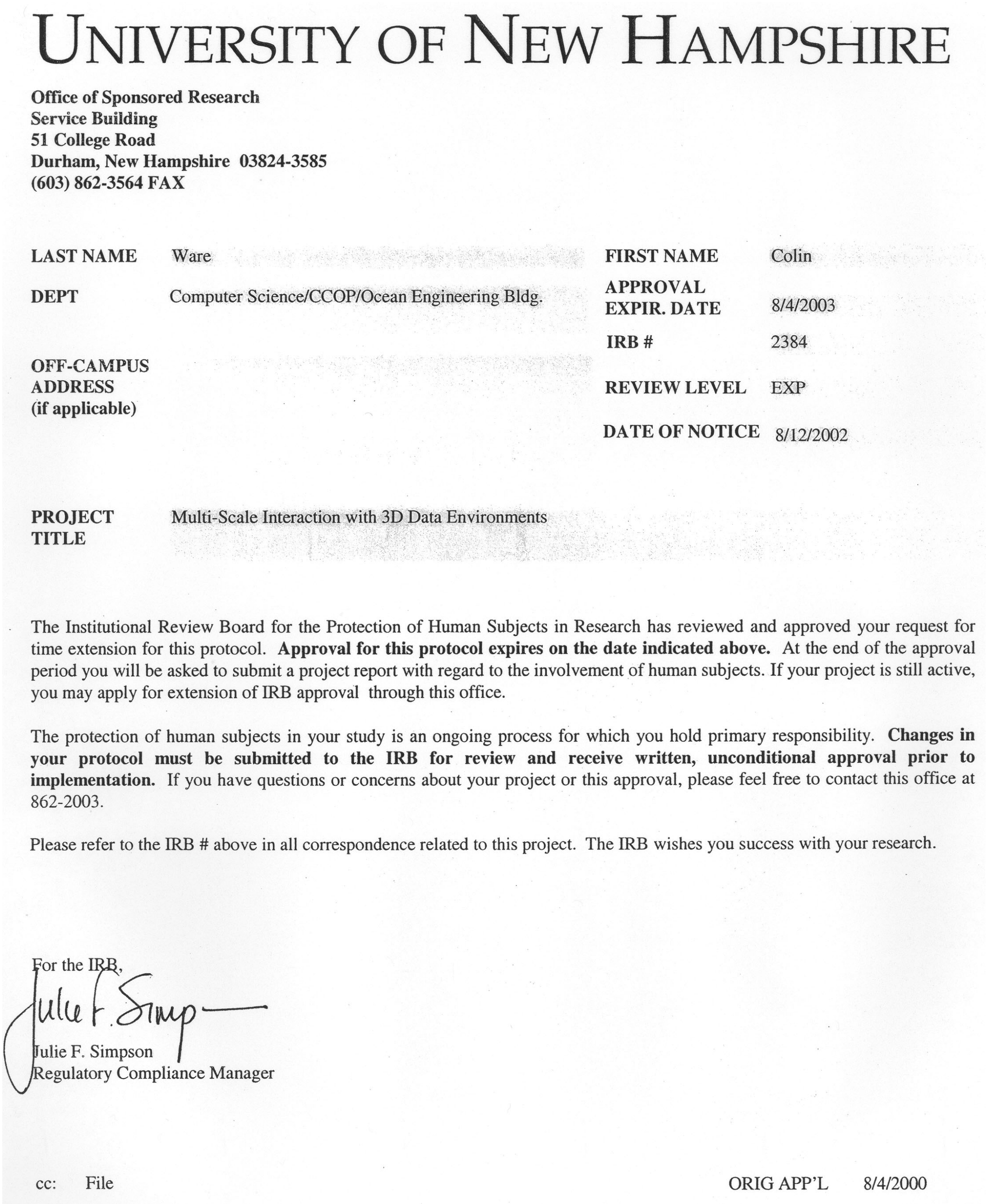
A.7 Conclusion

If memory were the only link between focus and context in a zooming interface, it would make sense to zoom in and out as quickly as possible. However, the results of this chapter indicate that other factors play an important role in zoom rate as well. The weakest factor is scale disparity—the difference in size between successive levels of detail. Frame rate is a more significant consideration. While higher frame rates tended to allow for slightly higher zoom rates past 8 frames per second, they did not increase tremendously. This leads to a guideline for user interfaces, namely that interfaces should strive to maintain a constant zoom rate even under a changing frame rate. The strongest factor was between-subject differences, perhaps due to user experience with computer interfaces or general eye-hand coordination. The maximum comfort threshold is probably the most relevant to setting system zoom rates. This combined with the stark differences between subjects in maximum comfort thresholds suggests that zoom rate should be tuned to the user or user population, with zoom rates of about 3.5 x/s for novices and zoom rates of about 8.0 x/s for experts.

To better understand why user experience or eye-hand coordination might play such a large role, consider the following explanation of the process of zooming. For a user to understand the multiscale information being displayed, he or she must keep track of the locations of focal items when zooming out for context, so that relationships with the focal item can be seen. For example, consider a scientist interacting with a zoomable Geographic Information System (GIS) who spots an interesting pattern of rock outcroppings at a close-up scale, and would like to understand what local geological events may underlie that pattern. The scientist must be able to pinpoint the location of the rock outcroppings by zooming out to get a wider contextual view. A sufficiently slow, smooth animation is required for the focal item to remain tracked while zooming out. In terms of controlling the navigation while zooming in, the user must be able to adequately guide navigation toward a potential focus, and then be able to stop zooming at the right time to keep the view at the proper scale—the scientist must be able to obtain the right scale for context, and be able to rapidly and easily zoom back in to the scale of the outcroppings.

APPENDIX B

IRB APPROVAL DOCUMENTATION



UNIVERSITY OF NEW HAMPSHIRE

Office of Sponsored Research
Service Building
51 College Road
Durham, New Hampshire 03824-3585
(603) 862-3564 FAX

LAST NAME	Ware	**FIRST NAME**	Colin
DEPT	Computer Science/CCOP/Ocean Engineering Bldg.	**APPROVAL EXPIR. DATE**	8/4/2003
		IRB #	2384
OFF-CAMPUS ADDRESS (if applicable)		**REVIEW LEVEL**	EXP
		DATE OF NOTICE	8/12/2002
PROJECT TITLE	Multi-Scale Interaction with 3D Data Environments		

The Institutional Review Board for the Protection of Human Subjects in Research has reviewed and approved your request for time extension for this protocol. **Approval for this protocol expires on the date indicated above.** At the end of the approval period you will be asked to submit a project report with regard to the involvement of human subjects. If your project is still active, you may apply for extension of IRB approval through this office.

The protection of human subjects in your study is an ongoing process for which you hold primary responsibility. **Changes in your protocol must be submitted to the IRB for review and receive written, unconditional approval prior to implementation.** If you have questions or concerns about your project or this approval, please feel free to contact this office at 862-2003.

Please refer to the IRB # above in all correspondence related to this project. The IRB wishes you success with your research.

For the IRB,

Julie F. Simp

Julie F. Simpson
Regulatory Compliance Manager

cc: File

ORIG APP'L 8/4/2000