Zooming Versus Multiple Window Interfaces: Cognitive Costs of Visual Comparisons

MATTHEW D. PLUMLEE and COLIN WARE
University of New Hampshire

In order to investigate large information spaces effectively, it is often necessary to employ navigation mechanisms that allow users to view information at different scales. Some tasks require frequent movements and scale changes to search for details and compare them. We present a model that makes predictions about user performance on such comparison tasks with different interface options. A critical factor embodied in this model is the limited capacity of visual working memory, allowing for the cost of visits via fixating eye movements to be compared to the cost of visits that require user interaction with the mouse. This model is tested with an experiment that compares a zooming user interface with a multi-window interface for a multiscale pattern matching task. The results closely matched predictions in task performance times; however error rates were much higher with zooming than with multiple windows. We hypothesized that subjects made more visits in the multi-window condition, and ran a second experiment using an eye tracker to record the pattern of fixations. This revealed that subjects made far more visits back and forth between pattern locations when able to use eye movements than they made with the zooming interface. The results suggest that only a single graphical object was held in visual working memory for comparisons mediated by eye movements, reducing errors by reducing the load on visual working memory. Finally we propose a design heuristic: extra windows are needed when visual comparisons must be made involving patterns of a greater complexity than can be held in visual working memory.

Categories and Subject Descriptors: H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Evaluation/methodology, theory and methods.*; H.1.2 [**Models and Principles**]: User/Machine Systems—*Human information processing, human factors*

General Terms: Design, Experimentation, Human Factors

Additional Key Words and Phrases: Multiple windows, zooming, visual working memory, interaction design, multiscale, multiscale comparison, focus-in-context

1. INTRODUCTION

In visualizations of large information spaces, such as detailed maps or diagrams, it is often necessary for a user to change scale, zooming in to get detailed

This research was funded in part by NSF grant 0081292 to Colin Ware and NOAA grant NA170G2285 to the UNH Center for Coastal and Ocean Mapping (CCOM).

Authors' address: Data Visualization Research Lab, Center for Coastal and Ocean Mapping, University of New Hampshire, Durham, NH 03824; email: mdp@ccom.unh.edu, colinw@cixunix.unh.edu.

information, and zooming out to get an overview before inspecting some other detail. We work with applications in oceanography where photographic imagery may be situated in the context of a much larger terrain map. A scientist might see a group of starfish in one part of the environment and become curious about similarities to a group previously seen in another region. Similarly, geologists may wish to spot similarities and differences in geological morphology between regions. In a very different problem domain, a network analyst may be interested in comparisons between localized subnets of a much larger system. These are all examples of exploratory data analysis where visual comparisons can be used to address a stream of informal queries issued within the enquiring mind of the scientist or engineer. An important aspect of such exploratory comparisons is that the objects of study may not easily be categorized or labeled with verbal descriptions.

Our purpose in this article is to report on an investigation we carried out to develop principled design heuristics that tell us what kind of interface is likely to be most effective for a given visual comparison task. We present a model of multiscale comparison tasks that has visual working memory capacity as a central component. This model is evaluated in an experiment comparing a zooming interface with a multiwindow interface and refined by means of a second experiment in which we measure the number of eye movements made by observers as they compare patterns.[1]

1.1 Interfaces that Support Multiscale Visual Comparisons

There are several interface design strategies that can be used to support visual comparisons in multiscale environments. One common method for supporting multiscale visual comparison tasks is to provide extra windows. One window is used to provide an overview map and one or more other windows show magnified regions of detail. The overview map usually contains visual proxies showing the positions and area of coverage of the detail maps. A number of studies have shown that overviews can improve performance on a variety of tasks. Beard and Walker [1990] demonstrated an advantage to having an overview in a tree navigation task, and North and Shneiderman [2000] found a substantial improvement in performance for text navigation with an overview, compared to a detail only interface. It is claimed that the overview + detail map can be used for a relative scale factor of up to 25 [Plaisant et al. 1995] or 30 [Shneiderman 1998] between overview and detail maps.

A second method of supporting multiscale visual comparison tasks is to use a fisheye technique. When a user selects a point of interest in fisheye views, this point expands spatially while other regions contract [Sarkar and Brown 1994; Carpendale et al. 1997; Lamping et al. 1995]. This means that both the focus region and the surrounding regions are available in the same display. There have been many variations on this basic idea. Some take semantic distance from the point of interest into account [Furnas 1986; Bartram et al. 1994]; others have concentrated more on simple geometric scaling around points of

[1] A prior version of the model and the first experiment were reported in Plumlee and Ware [2002]. We provide a more refined analysis here. The second experiment has not been previously reported.

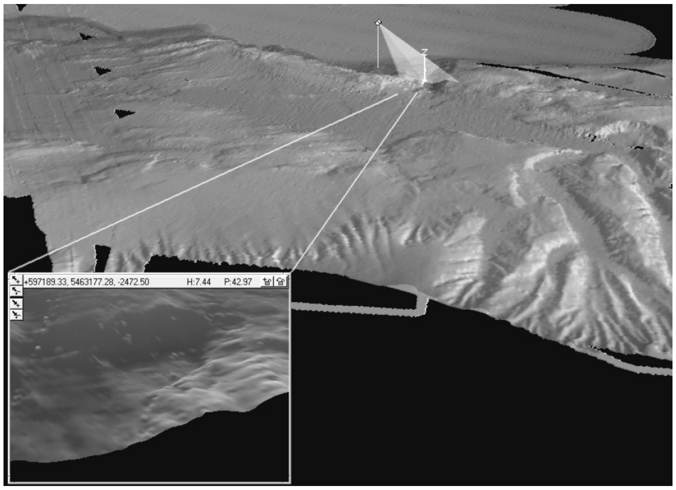


Fig. 1. A scene viewed from our GeoZui3D system [Plumlee and Ware 2003], illustrating the use of an extra window to focus on a detail. Linking mechanisms are used to situate this window in the context of the overview in the background window: a proxy of the viewpoint indicates the position, orientation, and relative scale of the detail window, and lines link the detail window to its proxy at the designated focus of user attention.

interest. Fisheye views suffer from the limitation that when large scale changes are applied, the distortion is such that the spatial information can no longer be recognized. Skopik and Gutwin [2003] found large decreases in subject's ability to remember the locations of targets as the distortion factor increased up to a scale factor of 5. Above this scale factor, fisheye views can become so distorting that shapes become unrecognizable.

A third way of dealing with the problem of transitioning between an overview and a detailed region is to make scale changes much faster and more fluid. In some systems called ZUIs, for Zoomable User Interfaces, zooming in and out can be accomplished rapidly with single mouse clicks [Perlin and Fox 1993; Bederson and Hollan 1994]. This means that the user can navigate between overview and focus very quickly and, arguably, use visual working memory to keep context (i.e., overview) information in mind when examining details.

1.2 Comparing Multiscale Navigation Interfaces

The particular application area that motivates our research is geospatial visualization. We have developed a system that we call GeoZui3D [Ware et al. 2001], which incorporates a zooming user interface and supports extra windows [Plumlee and Ware 2003], illustrated in Figure 1. Because quite large scale changes are often required in navigating our data spaces, we do not think that fisheye views would be useful in this application. Thus, the remainder of this article only deals with the tradeoffs between zooming and employing extra windows. However, we believe that the analysis we perform could be readily adapted to fisheye views as well as other spatial navigation methods designed to support visual exploration with occasional comparisons.

Clearly there can be advantages to using extra windows in a display, but there can also be drawbacks. They take up space on the screen, and if the user has to manage them they will take time and attention to set up, position, and adjust the scale. This amounts to a considerable extra complexity. In a study that evaluated zoomable interfaces with and without an overview, Hornbaek et al. [2002] found, somewhat paradoxically, that subjects preferred an overview but were actually faster without it.

Some authors have suggested guidelines for when extra windows should be provided [Wang Baldonado et al. 2000; Ahlberg and Shneiderman 1994; Plaisant et al. 1995]. These suggest that overview and detail windows must be tightly coupled to be effective; this is a feature of our multi-window system [Plumlee and Ware 2003], in which we use both linking lines and a proxy for the viewpoint to link the views (see Figure 1). Most relevant to our present work is Wang Baldonado et al.'s [2000] suggestion that we should "use multiple views when different views bring out correlations and/or disparities" according to their rule of *space-time resource optimization*. They suggested that the interface designer must "balance the spatial and temporal costs of presenting multiple views with the spatial and temporal benefits of using the views." We agree, but note that Wang Baldonado et al. provide little guidance as to how to achieve such a balance. In this article we present a quite simple model, which has visual working memory as a core component, and show how it can be used to model the tradeoffs of using a multiple view interface with an alternative zooming interface.

It should be noted that while the cited literature employs a background window for the detailed view and a smaller window for the overview, we do the reverse. The major reason we use smaller windows to display detail views is that users sometimes wish to display detail from two disparate locations at once. As long as the relative scales of the windows involved are taken into account and there is enough screen space available for the level of detail required for either the overview or the detail view, the choice of which gets assigned to the background window is not material to the analysis carried out in this article.

1.3 Visual Working Memory

The key insight that motivated the work we present here is that visual working memory may be the most important cognitive resource to consider when making decisions about when extra views are needed to support multiscale visual comparisons.

There is an emerging consensus among cognitive psychologists that there are separate working memory stores for visual and verbal information as well as for cognitive instruction sequencing [Miyake and Shah 1999]. These temporary stores can hold information for several seconds but are generally employed for less than a second. Recent studies of visual working memory (visual WM) have shown it to be extremely limited in capacity. Vogel et al. [2001] carried out a series of experiments in which they showed a few simple shapes to subjects (a *sample set*), followed by a blank screen for about a second, followed by a second group of shapes, which were either identical to the first or differed in a single object (a *comparison set*). These experiments, and those of other researchers, revealed visual WM to have the following properties. (Note that we speak of visual WM in terms of it being able to hold objects, when it would be more precise to speak of mental representations of perceived objects.)

- Only three objects can be held reliably at a time.
- As new objects are acquired other objects are dropped.
- Objects can only be held for several seconds, over which time they do not appreciably decay [Zhang and Luck 2004]. More time than that requires either a conscious act of attention or recoding into verbal working memory.
- Visual WM objects can have several attributes, such as color, simple shape, and texture. Thus it is not the case that only three colors, or three shapes, or three textures can be stored. All attributes can be stored so long as they are bound to only three objects.
- Visual WM object attributes are simple. It is not possible to increase information capacity by having, for example, three objects, each of which has two colors. Furthermore, an object that has a complex shape may use the entire capacity of visual working memory [Sakai and Inui 2002].

When visual comparisons are made between groups of objects, visual working memory is the cognitive facility used to make those comparisons. An observer will look at the first group, store some set of objects and their attributes, then look at the second group and make the comparison. If both groups are simultaneously visible on a single screen, eye movements are made back and forth between the two patterns. On each fixation, objects are stored in visual WM for comparison with objects picked up on the next fixation. If, on the other hand, both groups are small in size and spread out in a larger information space, then visual working memory can still mediate comparison when a rapid zooming interface is provided. However, now the objects must be held longer, while the user zooms out and back in to make the comparison.

But consider the case where the groups are larger, or the objects complex. Since only a part of a group can be stored in visual working memory, the user will have to navigate back and forth many times to make the comparison. This will become very time consuming, and at some point adding extra windows becomes beneficial. With extra windows, both groups can be displayed simultaneously and visual comparisons can be made using eye movements.

It is straightforward to infer a design heuristic from this analysis: If the groups of objects to be compared are more complex than can be held in visual working memory, then extra windows will become useful. The exact point where adding windows will become worthwhile will depend on design details concerning the following: how much effort is needed to set new windows up, the speed and ease of use of the zooming interface, the ease or difficulty of the visual comparisons required in the task, and the probability of occurrence of different classes of patterns. Note that we are only considering visual working memory in our analysis. If a pattern can be named, then the burden of remembering its presence may be transferred to verbal working memory, with a corresponding increase in the loading on that resource.

The remainder of this article takes this descriptive heuristic and elaborates it into a more detailed model. The model is then tested via a formal experiment comparing a zooming interface with a multiwindow interface. A second, follow-up experiment provides data on the number of eye movements made in visual comparisons of groups of objects. This allowed us to compare "visits" made via eye movements between windows with "visits" made by zooming, and thereby to further test and suggest refinements to the model.

To place our effort in the context of other modeling efforts, we briefly contrast our approach to others. Some models, such as those based on GOMS [Card et al. 1983] concern themselves with an intricately detailed task analysis, assigning time values to each mouse-click and key press, and building up estimates of the time it would take for a user to employ an interface for a given task. Cognitive resources such as working memory may be considered during the development of a GOMS model, but they are not included in a way that provides flexibility in applying the model to tasks that might require varying amounts of such resources. Other models, such as EPIC, ACT-R, or SOAR [Miyake and Shah 1999] concern themselves with an intricately detailed model of cognition, and rely on simulations to estimate how a user will perform on a given interface (for example, Bauer and John [1995]). These models account for cognitive resources such as visual working memory either explicitly or as a byproduct of deeper model processes. Our modeling effort is much more focused, taking the approach of highlighting the most important factors for visual comparison tasks, and accounting for visual working memory without attempting to develop a complete model. In addition, our results could be incorporated as a refinement to any cognitive model that has visual working memory as a component.

2. PERFORMANCE MODEL

In this section, we first present a general performance model for navigation-intensive tasks that lays the foundation for our analysis of comparison tasks. We then apply the model to a particular type of comparison task and tie performance to the limits of human visual working memory. Finally, we apply this more specific model to both a zooming interface and a multiple-window interface to make some rough predictions about when one interface would be more effective than the other.

2.1 General Performance Model

We propose the following general performance model for human performance in navigation-intensive tasks:

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

where

T is the expected <u>time</u> to complete the task,

S is the expected overhead time for constant-time events such as <u>setup</u> 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 between the prior location and the location corresponding to visit i, and
- D_i is the expected amount of time that a user will spend at the focus location during visit i.

This 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, performing mental arithmetic, rotating puzzle pieces into place, 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 indicates the impact a change in interface might have with respect to the amount of work that would occur regardless 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 facilitating a task strategy that reduces V, the number of visits required. In this sense, 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 in more detail later.

2.2 Applying the Model to 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 a *sample* set of objects to *comparison* sets, where each set has the same number of objects, and if a comparison set differs from the sample set, it differs in only one object. However, in our multiscale comparison task, there are several comparison sets rather than one, they are separated by distance rather than by time, and there is always exactly one comparison set that matches the sample set (as illustrated in Figure 2). The object 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

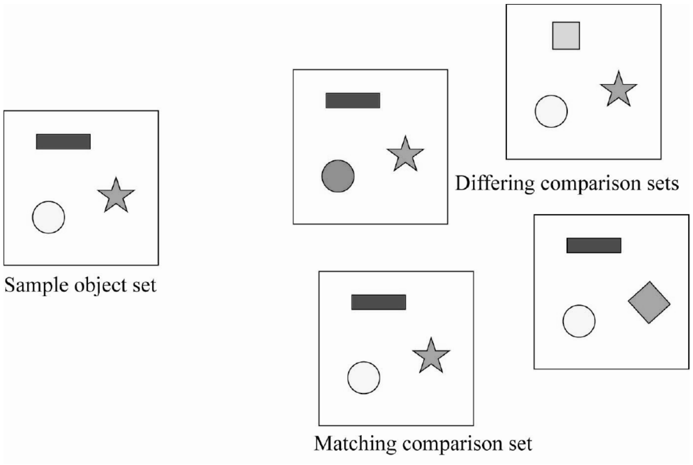


Fig. 2. An illustration of objects sets in a multiscale comparison task. Note that the actual locations of the sets would be much more spread out—so much so that only one set could be distinguished at a time.

scale to make the comparison directly. Whereas in Vogel et al.'s sequential comparison task, the user had no control over visits to the object sets, the performer of a multiscale comparison task may revisit sample and comparison 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. It is worthy of note that the task ends when the user determines that a comparison set under investigation matches the sample set, so that a user only visits about half of the comparison sets on average.

For a multiscale comparison task, the number of visits V is dependent upon the number of comparison sets in the task, as well as the number of visits required to determine whether or not a comparison set matches the sample. Both the expected transit time for a visit B_i and the expected time spent during a visit D_i are approximated as constants representing average behavior, making it possible to replace the sum in Formula 1 with a multiplication by the number of visits:

$$T = S + f_V(P, V_p) \cdot (B + D) \tag{2}$$

where

P is the expected number of nonmatching comparison sets that will be visited before the task is completed,
V_p is the expected number of visits made for each comparison set,
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 between sets on any given visit, and
D is the expected time for the user to make a match determination during a 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 effects 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 still characterizes the effectiveness of an interface in terms of the time it takes 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 assumed. Consider, for the sake of simplicity, the obvious strategy of making a match determination for one comparison set before moving on to the next comparison set. If only a subset of the objects can be remembered on each visit, the same comparison set might be visited a number of times before a determination is made. This number of visits is represented below by the term $V_{differ.}$ Theoretically, the strategy eliminates one trip to the sample for each comparison set that differs from the sample set, since some objects remembered from a differing set can be carried to the next comparison set. We assume this is true in our model (yielding $V_{differ} - 1$). If there are p comparison sets, then the number of comparison sets differing from the sample is $p - 1$; if each differing set is just as likely as the next to be detected as differing, then the expected number of differing sets visited is half of that, yielding $P = (p - 1)/2$. The total number of visits would then include the first visit to the sample set (when items are first loaded into visual WM and no comparisons can yet be made), plus ber of differing sets (P) times the number of visits for each of these sets ($V_{differ} - 1$), plus the number of visits required for a set that matches the sample set (V_{match}):

$$f_V(P, V_p) = (1 + P \cdot (V_{differ} - 1) + V_{match}). \tag{3}$$

2.3 Estimating the Number of Visits: Visual Working Memory

The capacity of visual WM plays a key role in estimating the values of V_{match} and V_{differ}. 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. If only a fixed number of objects can be remembered, as suggested by the work of Vogel et al. [2001], then the task performer must transit back and forth between the two sets a number of times inversely proportional to the limit on visual WM.

The important factors here 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 ($n \leq M$), a person could be expected 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 ($n > M$), it is only possible to remember *some* of the objects. 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 an optimal strategy (and if this strategy does not require additional resources from visual WM), the following equality holds.

$$V_{match} = \left\lceil \frac{n}{M} \right\rceil \tag{4}$$

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. Thus, when n is an integral multiple of M $(n = kM, k^{\text{TM}}\ \text{N})$, V_{differ} is as follows.

$$V_{differ} = \frac{3}{2} + \frac{n}{2M} \qquad | n = kM, k \in \mathbb{N}. \tag{5}$$

A derivation for Formula 5 is given in [Plumlee and Ware 2002], where formulas are also given for situations in which n is not a multiple of M.

With estimates for V_{match} and V_{differ} in hand, it is possible to restate the expression of the number of visits from Formula 3 in terms of known or empirically determined quantities. Assuming n is a multiple of M,

$$f_V(P, V_p) = \frac{(2 + P) \cdot (M + n)}{2M} \qquad | n = kM, k \in \mathbb{N}. \tag{6}$$

2.4 Applying the Specific Model to Navigation 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 1 and 2, the key variables that change between different interfaces are B and S—the transit time between focus locations, and the setup and overhead time. For zooming interfaces, the application of the model is trivial:

$$T_{zoom} = S_{zoom} + f_V(P, V_p) \cdot (B_{zoom} + D), \tag{7}$$

where B_{zoom} is the expected cost of using the zooming interface to get from set to set, and S_{zoom} includes the cost of a user orienting him or herself to the initial configuration of the sets. By substituting Formula 5 for the visit-function f_V, it follows that

$$T_{zoom} = S_{zoom} + \frac{(2 + P)(M + n)}{2M}(B_{zoom} + D) \qquad | n = kM, k \in \mathbb{N}. \tag{8}$$

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 at once. Here is how the model applies to a multiple-window interface:

$$\begin{aligned} T_{multi} \; = \; & S_{eye} + f_V(P, V_p) \cdot (B_{eye} + D) \\ & + S_{multi} + f'_V(P, V_p) \cdot (B_{multi} + D'). \end{aligned} \tag{9}$$

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 comparison sets more than once. Therefore, Formula 9 can be reduced to

$$T_{multi} = S_{multi} + P \cdot B_{multi} + f_V(P, V_p) \cdot (B_{eye} + D). \tag{10}$$

By substituting Formula 6 in for the visit function f_V, we get

$$T_{multi} = S_{multi} + P \cdot B_{multi} + \frac{(2+P)(M+n)}{2M} \cdot (B_{eye} + D) \qquad |n = kM, k \in \mathbb{N}. \tag{11}$$

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 comparison sets. Similarly, P can easily be calculated based on the number of comparison 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.

2.5 A Rough Model Comparison of Navigation Interfaces

Now the analytic tools are at hand to make a rough comparison of zooming and multiple window interfaces as they apply to the multiscale comparison task. The extra terms in Formula 11 beyond those in Formula 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 navigate back and forth between sample and comparison 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 8 is steeper than the slope of

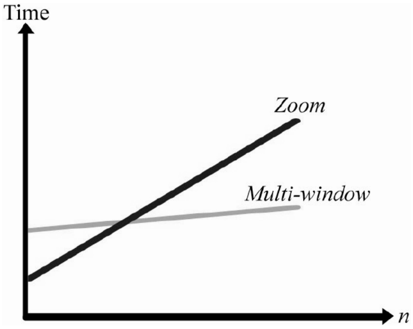


Fig. 3. Expected relationship between performances in completing a multiscale comparison task when using zoom and multiple window techniques.

Formula 11. Thus, as illustrated in Figure 3, 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. In Section 3, a particular instance of a multiscale comparison task is used to illustrate how this modeling might be applied.

2.6 Model Caveats

The model described so far makes several assumptions worthy of note. The model assumes perfect accuracy of visual WM. It also assumes that a person has the ability to remember which objects and comparison 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 Sections 4 and 5.

Another important factor not included in the model is the amount of visual WM required by the user interface—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 comparison set is currently being compared (with a zooming interface). Alternatively, visual objects might be dropped from visual WM over the time period of a zoom, or intermediate images seen during zooming might interfere with visual WM. All of these effects would either render the task infeasible, or increase the expected number of visits and thereby increase the slope for the effected technique. If the effect is dependent upon the number of comparison sets already visited, it is also possible that the linear relationship between n and $f_V(P,V_p)$ would become quadratic, or worse.

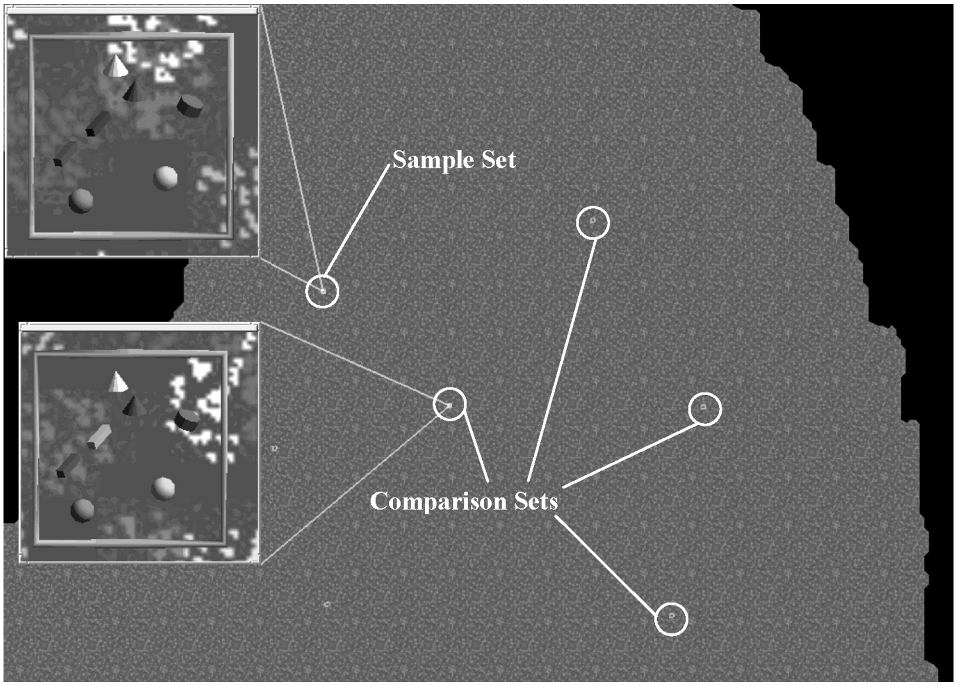


Fig. 4. Example of the *multi-window* condition with two windows created. One window is focused on the sample set, while the other is focused on its match.

3. APPLYING THE MODEL TO A SPECIFIC INSTANCE

The model as applied in the previous section predicts that, for any multiscale comparison task, zooming should outperform multiple-window interfaces when relatively few items must be compared, but that a multiple-window interface should outperform zooming interfaces once the number of items to be compared crosses some critical threshold. Toward validating the model in light of this prediction, this section describes an instance of the multiscale comparison task and of the zooming and multiple-window navigation techniques that are then analyzed with the model. The task we are interested in is visual comparisons between patterns. To facilitate a formal analysis and empirical evaluation we chose to use patterns of discrete geometric colored shapes. Section 4 presents an experiment based on the same task and interface instances for comparison against the model predictions.

3.1 An Instance of Multiscale Comparison

The task instance is a 2D multiscale comparison task in which a person (hereafter referred to as a *subject*) must search among six comparison sets for one that matches the sample set. These seven sets of objects are randomly placed over a textured 2D background as shown in Figure 4. The sample set has a random arrangement of n objects, and is identifiable by its yellow border. The comparison sets each have a gray border and the same number and arrangement of objects as the sample set, except that only one matches the sample

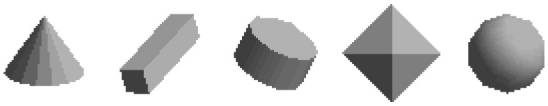


Fig. 5. The five shapes that were available for creating each object set.

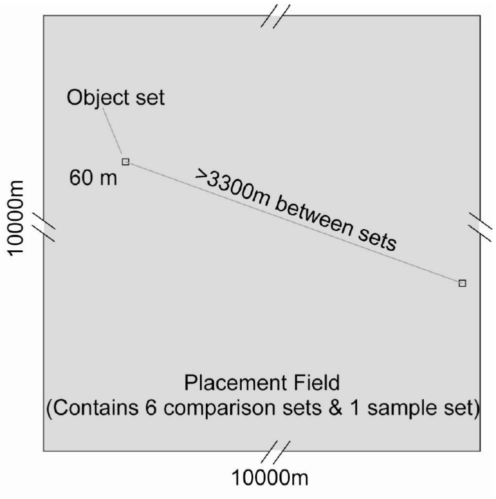


Fig. 6. Schematic of the constraints on random placement in the multiscale comparison task instance.

set exactly. The other five comparison sets differ in exactly one object, either in shape, in color, or in both aspects. The background texture camouflages the clusters and their contents at intermediate scales—enough to require a subject to zoom in by a significant amount so as to see individual objects, and to zoom out enough to spot the clusters in relation to one another.

The layout of objects and object sets are random under certain constraints. Each object fits within a circle with a 15-meter diameter (in the virtual world of the task). Each object set is created by random selection from 5 shapes (see Figure 5) and 8 colors. No color or shape appears more than twice in any object set, and objects cannot overlap significantly. The relative locations of objects are invariant in a task instance (during a given experimental trial), even though an individual object may differ from set to set in shape and/or color. The scales at which the objects in a set can be visually identified are roughly between 0.1 m/pixel and 2 m/pixel. As illustrated in Figure 6, the size of an object set is 60 meters to a side, and the minimum amount of space between any two sets is 3.3 kilometers (on center). Further, the valid field of placement on the textured background is a square 10 kilometers to a side. 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).

3.2 The Navigation Mechanisms

In order to perform a proper analysis or implement an experiment, certain characteristics of the two navigation mechanisms must be nailed down. The zooming mechanism, referred to as *zoom* for short, is activated when a subject presses the middle mouse button. When the button is pressed, the screen first animates so that the point under the cursor begins moving toward the center of the screen. This panning operation occurs very quickly, advancing roughly a quarter of the distance to the target location each animation frame, or 99.4% each second. If the subject then pushes the mouse forward, the scene zooms in (at roughly 7×/s) about the new center point. If the subject pulls the mouse backward, the scene similarly zooms out (at about 8×/s). A subject may zoom in or out without bound, as many times as is desired. The subject uses this interface to zoom back and forth between the sample set and the various comparison sets, potentially zooming back and forth a few times for each comparison set.

The multiple-window mechanism, referred to as *multi-window* for short, retains a main view at a fixed scale of about 17.5 m/pixel, initially with no other windows present. To create a window, the user first presses the 'z' key on the keyboard, and then clicks the left mouse button to select a location for the center of the new window. The window is created in the upper left corner of the screen at a size too small to be useful. The subject then uses the mouse to resize the window to a usable size, and is free to place it elsewhere on the screen (using common windowing techniques). The windows are brought up very small to compensate for the fact that they are automatically set to the optimal scale for viewing the object clusters. They are automatically set to the optimal scale so as not to introduce any elements of the zooming interface into the multiple-window interface. A maximum of two windows is allowed by this interface. Each window has two semi-transparent lines (tethers) linking it to a proxy representation in the main view, as shown in Figure 4. The proxy marks the area in the main view that the associated window is magnifying. Once a window is created, the subject can click and drag the window's proxy through the main view to change its location. The contents of the window are updated continuously without perceptible lag. The subject establishes one window over the sample set, and another over a comparison set, and then uses the proxy for this second window to navigate it to each of the other comparison sets as needed.

3.3 Model Analysis

Before running an experiment based on the task and interface instances just described, we estimated model parameters to determine what our performance model would predict for subject performance. Note that in a practical situation, the values of D, B, and S could be determined empirically, but here we made estimates without recourse to an existing prototype. From the work of Vogel et al. [2001], a good estimate of the capacity of visual working memory, M, was 3 (assuming an integer value). The time, D, to determine whether or not the objects in a comparison set match those remembered, was a bit more elusive.

From informal experience, we determined that this number should be between a half-second and a full second. While informal experience also showed that D would be smaller for smaller n, we assumed that D was a constant 0.8 seconds. Finally, because there are six comparison sets, $P = (6 - 1)/2 = 2.5$. The remaining parameters depend on the navigation interface.

3.3.1 *Zooming Interface.* For simplicity, we assumed that the zooming rate was 7×/s in both directions. It seemed reasonable to estimate that a subject would inspect an object set at a scale of about 0.45 m/pixel, and might zoom out to about 15 m/pixel to see the entire field of object sets. Thus, the cost of zooming in or out was estimated at $\log_7(15/0.45)$. The distance covered between visits was seen to be between 3.3 kilometers and 14.1 kilometers, which is between 220 pixels and 940 pixels at 15 m/pixel. We estimated the average time to move the cursor this distance and press a mouse button to start a new zoom at about 1.5 seconds. This led to the following conclusion: $B_{zoom} = 2 \cdot [\log_7(15/0.45)] + 1.5 = 5.2$ seconds. We believed S_{zoom} should be small, since the only overhead to account for was the initial user-orientation period, which we estimated to be about 2 seconds. Using all this information, Formula 8 can be used to get an estimate on the total task time:

$$T_{zoom} = 2 + \frac{(2 + 2.5)(3 + n)}{6} \cdot (5.2 + .8) = 15.5 + 4.5 \cdot n. \qquad (12)$$

3.3.2 *Multiple-Window Interface.* To model the multiple-window technique, we assumed that subjects would resize the focus windows 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 was estimated at 10 seconds per window. We assumed that both of the allowed focus windows would be created, and that a subject would require 2 seconds for orientation as we assumed with the zooming interface, leading us to estimate $S_{multi} = 22$ seconds. We assumed that subjects would navigate the focus windows from place to place by clicking and dragging their proxy representations within the overview (see Figure 4). In such a case, the optimum strategy would be to park one window on the sample set, and continually drag the proxy of the other window around to each comparison set. With this information, and expecting that it would be more difficult to properly place a proxy than to select a zooming location, the expected time to move a proxy from set to set was estimated at about 2 seconds per visit. This translates into a B_{multi} of 2 seconds. The final parameter estimate required is the time for saccadic eye movements between the window over the sample and the window over the current set of objects. Such eye movements are known to take about .03 seconds on average [Palmer 1999], so we took 0.1 second as a good upper bound. With our estimate of $B_{eye} = 0.1$ second, Formula 11 can used to get an estimate of the total task time:

$$T_{multi} = 22 + 2.5 \cdot 2 + \frac{(2 + 2.5)(3 + n)}{6} \cdot (.1 + .8) = 29.025 + 0.675 \cdot n. \qquad (13)$$

3.3.3 *Comparing Predictions.* Formulas 12 and 13 provide simple linear estimates for how long a subject might take in performing the multiscale

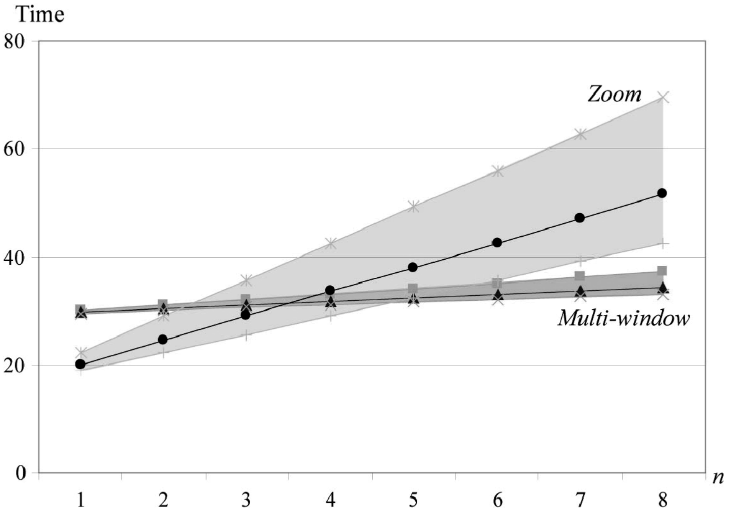


Fig. 7. A refinement of Figure 3 using estimated parameters for each model variable. The heavy lines represent the values calculated for $M = 3$. The borders above and below the heavy lines represent the values calculated for $M = 2$ and $M = 4$, respectively.

comparison task instance involving n items in an object set. The intersection point of these two estimates (the point at which the multi-window interface has faster completion times than the zooming interface) is just under $n = 3.6$.

A range of preditions can be made by choosing a few different estimates for the capacity of visual working memory. Figure 7 plots the results of applying the model in this way while varying n between 1 and 8, and varying M between 2 and 4. This plot also suggests that one should expect zooming to become less efficient than using multiple windows at around 3 or 4 items.

4. EXPERIMENT 1: EVALUATING THE MODEL

We conducted an experiment to directly test the analysis presented in the previous section, and thereby lend support to the overall model. The task and interface instances described in the previous section are exactly what subjects were presented with in a given trial of the experiment. In this section, we describe remaining details regarding the design of the experiment and present the experimental results.

4.1 Design

Each experimental subject was trained using 8 representative trials, and was then presented with 4 experimental blocks of 16 different trials in a 4 × 2 × 2 factorial design. All trials varied in three parameters:

- n, the number of objects 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-window*, and
- b, whether verbal WM was *blocked* or *unblocked*.

Because of the two different sets of values used for n, the result was an unbalanced 6 × 2 × 2 experimental design, with differing numbers of trials for differing levels of n.

To reduce user confusion in switching between mechanisms, each experimental block was split into two groups such that all *zoom* conditions were grouped together within an experimental block, separate from all *multi-window* conditions in that block. The groups were counterbalanced across the four experimental blocks and the order of the four values for n varied randomly within each subgroup.

Prior to each trial, a screen was displayed that told the subject how many objects to expect in each cluster and what navigation method was to be used (the other method was disabled). 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 comparison set matched the sample set (the comparison set had to be visible on the screen at a reasonable scale for the spacebar to register). If the subject pressed the spacebar on the correct comparison set, the experiment proceeded to the next trial. Otherwise, the subject was informed of the incorrect choice and the condition was repeated in a new trial with a new random layout and selection of objects. A condition could be repeated a maximum of 5 times (this occurred only once in practice).

In order to determine whether or not verbal working memory played a role in the execution of the task, subjects were required to subvocally repeat the list "cat, giraffe, mouse, mole" throughout the course of the trials on trials in which verbal WM was blocked.

4.2 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}. Subjects ranged in age between 18 and 37, with most of them at the bottom of that range. All subjects had normal or corrected-to-normal vision, and informal questioning indicated that some had experience with virtual worlds (particularly gaming) but many did not.

4.3 Results

Data was collected from 1451 trials, including 1279 successful trials and 166 that ended in an error and triggered a new trial on the same condition. Trials that ran longer than 90 seconds were discarded (26 from *zoom* conditions, 6 from *multi-window* conditions), leaving 1419 trials. 90 seconds was chosen because it was the beginning of a gap in the distribution of time results that appeared just inside three standard deviations from the mean.

4.3.1 *Completion Times.* The completion-time results are summarized in Figure 8 for trials ending in successful completion. An analysis of variance revealed that the number of objects in each set (n) contributed significantly to

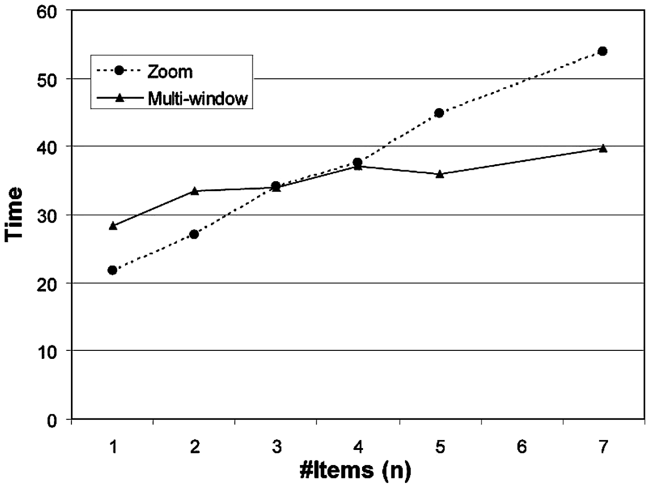


Fig. 8. Completion-time results of Experiment 1, plotting the average time to successfully complete a task for various values of *n*. The *zoom* condition exhibits a greater slope than the *multi-window* condition.

Table I. Results of Individual Analyses of Variance on Task Completion Time for Each Level of *n*

n	ANOVA Result
1	**F** (1, 19) = 46.62; **p** < 0.001
2	**F** (1, 7) = 5.67; **p** < 0.05
3	**F** (1, 19) = 0.008; NS
4	**F** (1, 7) = 0.002; NS
5	**F** (1, 11) = 11.22; **p** < 0.01
7	**F** (1, 11) = 15.73; **p** < 0.005

task completion time (**F** (5, 56) = 72.41; **p** < 0.001). Most relevant to our model however, was an interaction between the number of objects and the navigation mechanism ($n \times m$) that also contributed significantly to task completion time (**F** (5, 56) = 12.16; **p** < 0.001). As predicted by the model, there was a crossover in efficiency between the two navigation methods between 3 and 4 items per set. This was substantiated by individual analyses of variance for each level of *n* as summarized in Table I.

There was a small but significant interaction between blocking of verbal working memory and the navigation mechanism (**F** (1, 26) = 10.91; **p** < 0.01). This is illustrated in Figure 9. This interaction suggests that verbal working memory is used as an additional resource in the *zoom* condition, but not in the *multi-window* condition.

4.3.2 *Error Rates.* Figure 10 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*. 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-window* condition. It should be

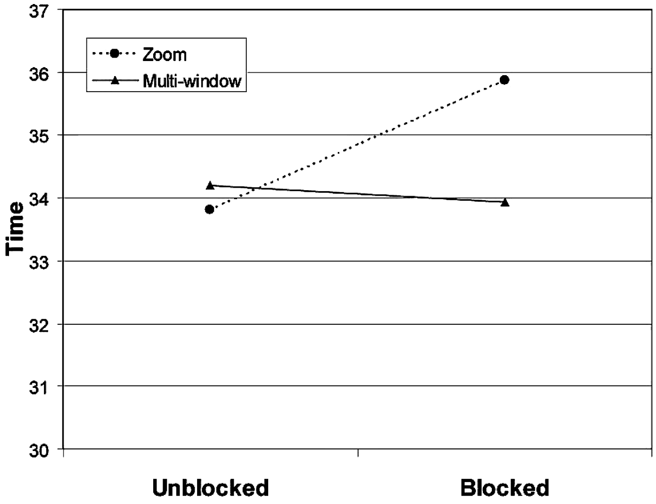


Fig. 9. The effect of blocking verbal WM on task completion time is small but significant for the *zoom* condition but is not significant for the *multi-window* condition. Note the non-zero origin.

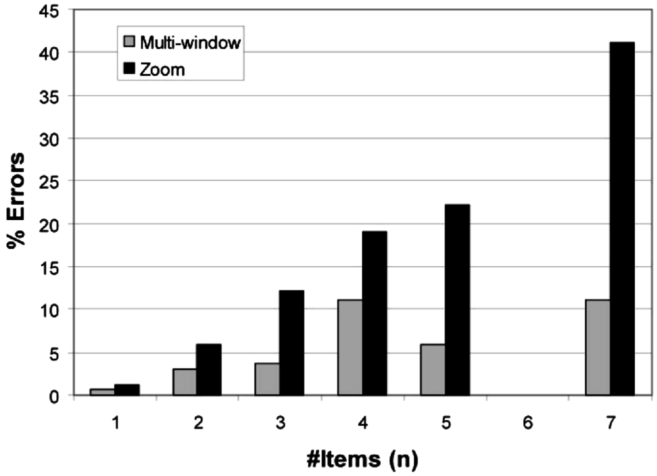


Fig. 10. The percentage of errors for various values of n. The *zoom* condition exhibits a greater number of errors than the *multi-window* condition.

noted that false-positives—cases in which a subject signaled a match for a non-matching comparison set—were the only kind of error readily detectable by the experimental design, and are therefore the only kind reported.

An analysis of variance was performed with average error rate as the dependent variable and with n, navigation method (m), and the blocking of verbal WM as independent variables. Both n and m significantly affected error rates ($\mathbf{F}(5, 55) = 16.30$; $\mathbf{p} < 0.001$ and $\mathbf{F}(1, 22) = 27.00$; $\mathbf{p} < 0.001$ respectively), as did their interaction ($\mathbf{F}(5, 55) = 3.52$; $\mathbf{p} < 0.01$). However, blocking of verbal WM had no significant impact on error rates.

4.4 Discussion

The results of this experiment support the predictions of the model from the previous section, 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 finding that verbal WM was used by subject as resource is interesting although unsurprising. Subjects have to hold information in visual WM far longer when using the zooming interface and passing some of the load to verbal WM would provide an obvious benefit.

There were large differences between the two interfaces in terms of the numbers of errors that occurred, as shown in Figure 10. 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 comparison sets in order to guarantee a correct response. This assumption says that subjects essentially *guessed* that the last comparison 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.

4.5 Post Hoc Error Analysis

To test the assumption that subjects may have made decisions without complete information, 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 Experiment 2 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 later on Experiment 2. The analysis focused on how many visits subjects made to the last comparison set—the set under investigation when the subject made the "match" decision and pressed the space bar. Visits to the other comparison sets were not considered because there was no way to determine when the "no-match" decision was made—it could have been while looking at the sample set or while looking at the non-matching comparison set. It is at first plausible that subjects might base their match decisions on probability of error rather than solely upon information gained from making comparisons. For instance, if the comparison set is the last one to be investigated (all others having been judged not to match), one might expect a subject to guess based on a confident assessment of the prior comparison sets. Alternatively, if it is not the

Table II. Evidence for Judgments Based Solely on Comparisons

	Number of Visits Made...			
	in All Cases, to...		When All Sets are Visited, to...	
n	The Chosen Set	Non-Chosen Sets	The Chosen Set	Non-Chosen Sets
1	1.03	1.00	1.18	1.00
3	1.33	1.08	1.21	1.14
5	1.51	1.13	1.74	1.29
7	1.70	1.25	2.47	1.49

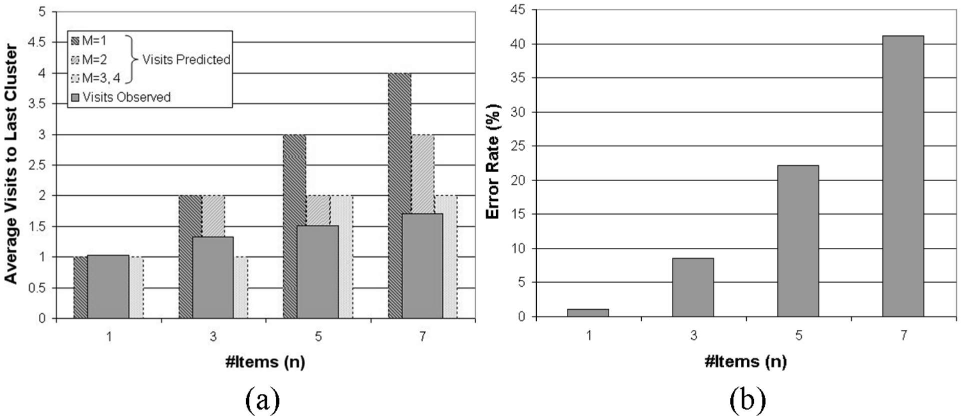


Fig. 11. The number of visits to the last comparison set investigated in the zoom condition and the number of errors made, versus the number of items in the sample set: (a) actual number of visits to the last comparison set plotted in front of the expected number of visits for perfect performance at visual working memory capacities $M = \{1, 2, 3, 4\}$; (b) the actual error rates observed.

last one to be investigated, a subject might be expected to make extra visits to be more confident. However, the data listed in Table II provides evidence against such decision-making behaviors: most visits were made when all comparisons sets were visited, and sets not chosen as the matching one received significantly fewer visits than the chosen set.

Plotted in the background of Figure 11(a), are the predicted number of visits required to achieve perfect performance, assuming capacities of visual working memory at 1, 2, 3 and 4 objects. The predicted values were calculated by modifying formula 4 to count only the number of visits to the matching comparison set (Formula 4 includes visits to both the comparison and sample sets). An addition of one is required (within the outer ceiling) to account for when the user made the match determination while looking at the sample set, but had to navigate back to the matching comparison set in order to record the decision:

$$V_{matching\text{–}comparison\text{–}set} = \left\lceil \left(1 + \left\lceil \frac{n}{M} \right\rceil\right) \Big/ 2 \right\rceil . \tag{14}$$

The foreground bars in Figure 11(a) illustrate the average number of visits subjects actually made to this last comparison 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.

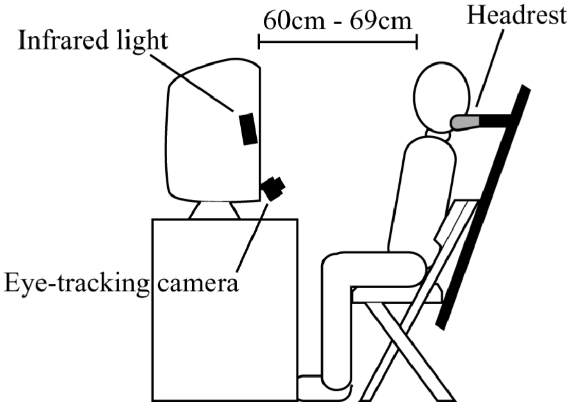


Fig. 12. Eye-tracking equipment, monitor, and chair with headrest (not drawn to scale).

Figure 11(b) illustrates the error rates for each level of n. The large increase in error rate at 5 and 7 items is notable, and appears to correspond roughly with the difference between the measured and predicted numbers of visits when $M = 1$ or $M = 2$. Thus in the zooming condition, subjects visited object clusters far less than needed, even assuming a visual working memory capacity of three or four objects.

5. EXPERIMENT 2: VISITS MADE BY EYE MOVEMENTS

The first experiment revealed that subjects made fewer visits between object clusters than required for the zooming condition. This could plausibly account for the high error rates we observed in these conditions. However, we had no data on visits in the *multi-window* conditions for a comparable analysis. In those conditions, visits were being made with eye movements and we had not measured them. We therefore designed a second experiment using eye-tracking technology to determine the number of number of visits made by eye movement. We predicted that subjects were making more visits than we had observed for zoom conditions.

5.1 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 12 illustrates how the equipment was arranged. The chair was located such that a subject's eye was between 60 cm and 69 cm from the screen. The visible area on the screen was between 36cm and 40cm. This produced a horizontal field of view subtending $33^\circ \pm 4^\circ$.

The EyeTech Digital Systems tracker delivered eye gaze information at a rate of about 25 Hz with a precision of roughly 20 pixels (about $1/2^\circ$), although tracking tended to drift more than $1/2^\circ$ throughout a session, reducing precision to approximately 1°. To compensate, the eye tracker was calibrated to each

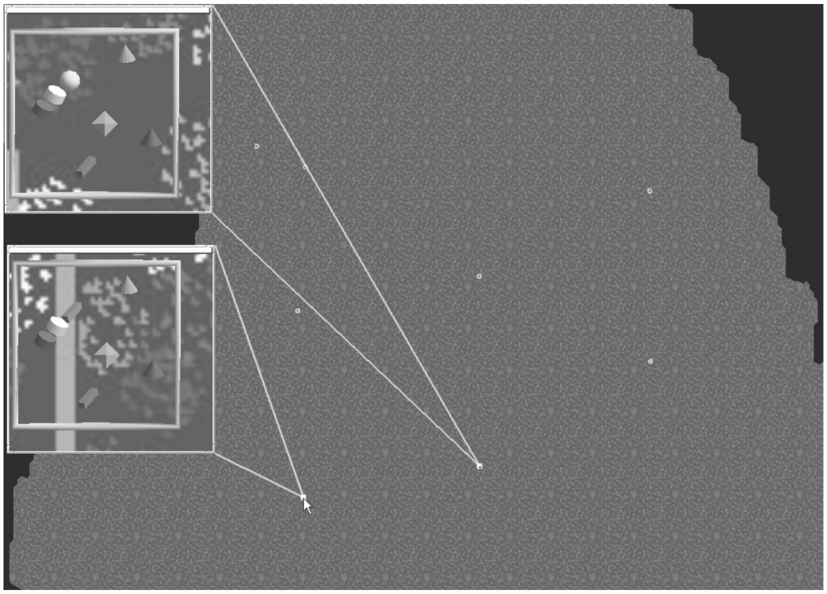


Fig. 13. The default window sizes presented to subjects 3 through 10 in the experiment, relative to the background overview display.

subject before training and between experimental 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 at and the windows could be spaced far enough apart so as to eliminate ambiguous measurements.

5.2 Changes to the Multiple-Window Navigation Mechanism

The basic navigation mechanism for this experiment was the same as for the *multi-window* condition of Experiment 1, 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.

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. This change would not significantly impact the flow of the remainder of the task, and therefore was not expected to impact error rates in task completion. The layout of the windows as they appeared upon creation is illustrated in Figure 13.

5.3 Design

Each subject was trained on 8 representative trials, and was then presented with 6 experimental blocks, each containing 8 trials in a 4 × 2 factorial design. The 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 Experiment 1, each experimental block was split into two groups such that all trials on which verbal WM was *blocked* were grouped together separately from all trials on which verbal WM was *unblocked*. The groups were counterbalanced across the six experimental 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.

5.4 Measurement

For the purposes of measurement, an *eye-movement 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 one subwindow to the other, or whenever the comparison set subwindow was moved to a different comparison 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 comparison set.

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

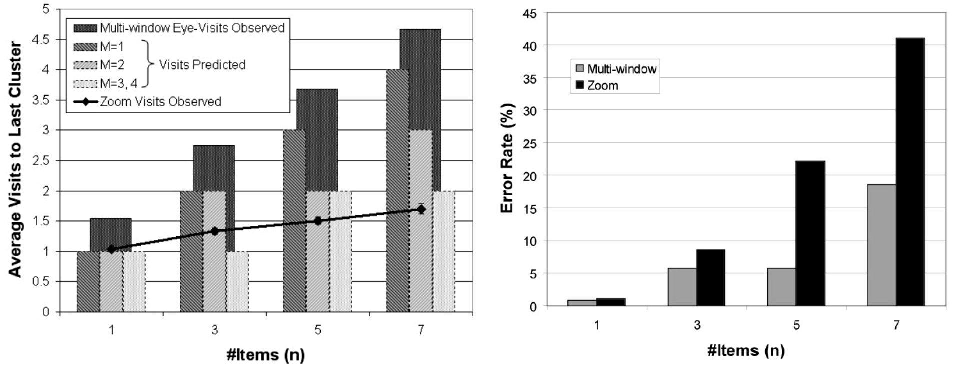
5.5 Subjects

The experiment was run on 10 subjects: 5 female and 5 male. Subjects ranged in age between 18 and 25. All subjects had normal or corrected-to-normal vision, and there was again a mix of those with exposure to virtual environments.

5.6 Results

A total of 523 trials were completed, of which 497 produced data deemed valid for analysis. Experimental 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 14 summarizes the results. The background bars in Figure 14(a) illustrate the average number of visits made (with the eye) to the last comparison set



(a) (b)

Fig. 14. The number of visits to the last comparison 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 comparison set plotted behind the expected number of visits for perfect performance at visual working memory capacities M = {1, 2, 3, 4}, 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.

for each comparison 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 Section 4.5. For comparison, the foreground line illustrates the average number of visits made in the *zoom* condition of Experiment 1.

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

Figure 14(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 Experiment 1. 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 15 illustrates how the new error rates for the *multi-window* condition compare against the error rates from Experiment 1. 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%.

To determine whether or not verbal WM was a significant factor in error rates, an analysis of variance was performed with average error rate as the dependent variable and with *n* and the blocking of verbal WM as independent

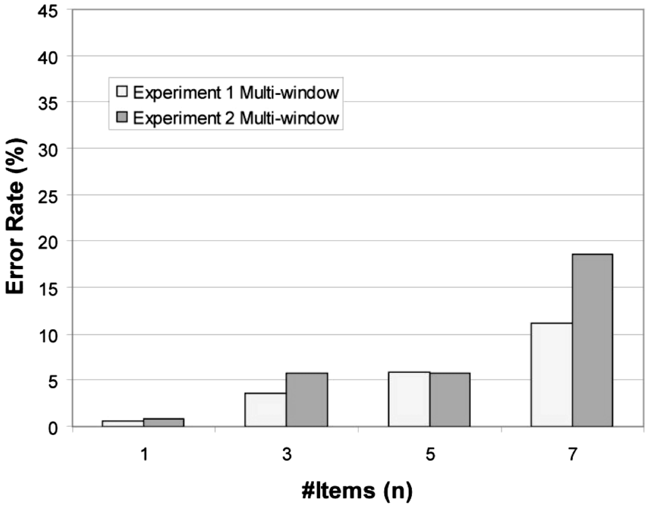


Fig. 15. Comparison of error rates between the *multi-window* conditions of the Experiments 1 and 2.

variables. While n significantly affected error rates ($\mathbf{F}\,(3, 27) = 9.79$; $\mathbf{p} < 0.001$), blocking of verbal WM had no significant impact on error rates.

5.7 Discussion of Experiment 2

The results show that subjects made dramatically more visits with the eye between windows than they made with the zooming interface. In addition, subjects made more eye-visits (in the multiple-window condition) than the model predicted would be necessary to achieve perfect performance.

This suggests a kind of satisficing strategy with visual working memory as a limited-capacity, cognitively critical resource [Simon 1956]. When visits are cheap in time and cognitive effort, for example when they are made via eye movements, they are made frequently and people make a separate eye movement to check each component of the two patterns they are comparing. Thus their visual WM capacity relating to the task is effectively one. However, when visits are expensive in time and cognitive effort, for example when zooming is required, subjects attempt to load more information into visual WM and they also quit the task after fewer visits, which results in many more errors.

Of course, the high error rates we observed have much to do with the participant's level of motivation and ability to maintain attention through a long repetitive experiment. Given a situation with fewer tasks, and a higher penalty on making errors, errors would be lower. Conversely, with more repetitive tasks or a lower penalty on making errors, errors would be more common. In some situations, for example, where an image analyst must visually scan hundreds of images per day, the error rate might be higher.

6. CONCLUSION

Our results support the theory that visual WM capacity is a key resource in visual comparison tasks. However, given the number of visits required to achieve low error rates, they also suggest that an error-free capacity of three objects is an overestimate. This is hardly surprising given the nature of the tasks carried out in our experiments. It is important to note the differences between our study and those carried out by vision researchers such as Vogel et al. [2001]. In most laboratory studies all subjects have to do is remember the target patterns. In a real application (and in our experiments) subjects also had to use and apply visual information about the interface to enable them to navigate from point to point. The navigation task undoubtedly consumed visual working memory capacity. A more reasonable estimate of the remaining capacity that might be applied to the pattern matching task is one, relatively error-free, visual-working-memory object.

How should interface designers take advantage or our results? Very few if any real world tasks map exactly onto the task we designed for our experiments. The kind of modeling we carried out turned out to be surprisingly difficult for even the simple task reported in Experiment 1, and it seems unlikely that many designers would wish to undertake this kind of detailed mathematical modeling. Therefore it is worth discussing the value of a simple design heuristic. If this could be shown to be robust under a wide variety of conditions it would likely be more useful than a detailed model.

The detailed model we built to support the experiment had as its starting point Equation 1:

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

where S is the setup time, B is the time to make a movement, and D is the dwell time at a particular location. Section 2 was devoted to elaborating this model. We now briefly take the opposite approach and consider how it may be simplified for designers.

For many interfaces it is reasonable to assume that, to a first approximation, B and D are constants. Thus for example in the case of a ZUI a reasonable approximate value for $B + D$ might be 5 seconds. In the case of eye movements: A rough estimated of $B + D$ might be 1 second (assuming one saccade between patterns and two fixations on each pattern). Note that if a prototype for the interface exists, estimates for B and D can be determined empirically.

Our experiments suggest that the value of V should be based on the assumption that only *one* simple visual object can be held in visual WM. Thus, $V = C_p$ where C_p is an estimate of the pattern complexity in units of visual-working-memory objects. For example, the task used in this experiment required a user to visit 5/2 sets of objects on average, each with n/2 visual-WM objects on average, plus one set with n visual-WM objects. Thus, our task would have a pattern complexity of $C_p = 5/2 \cdot n/2 + n = 2.25n$.

For multiple window design solutions we arrive at

$$T_{win} = S_{win} + C_p \cdot (1.0), \tag{15}$$

and for the ZUI solutions we arrive at

$$T_{zoom} = S_{zoom} + C_p \cdot (5.0). \tag{16}$$

These equations would put a steeper slope on the predictions given in Section 3.3 because we are now considering near-error-free performance, and thus the crossover point moves to between 2 and 3 items.

A major unknown is the overhead cost of setting up a zoom versus the overhead cost involved in setting up extra windows. In many interfaces, setting up a scale change is a slow operation, requiring a menu selection and several clicks. In a ZUI, zooming is a frequent, well learned operation and should be fast. The cost of providing extra windows is also based on how easily they can be created, positioned and sized. Generally, both costs vary inversely with frequency of use. If users need an extra window only very occasionally it may take a minute or more for the user to remember how to set them up. That would be time for a lot of zooming. But if extra windows are very frequently used, and can be setup rapidly—or perhaps are a permanent feature of the user interface—then the cognitive cost should be much lower. These kinds of considerations are difficult to model; they should depend on a task analysis of the particular application. However, the crossover point is very likely to lie somewhere between 2 and 7 visual-working-memory objects. Zooming back and forth to compare more than 7 visual-working-memory objects would be intolerably burdensome no matter how good the ZUI.

A second major unknown is how many errors the user will make. If the task is perceived as boring, repetitive, and with no reward for good performance, application users would be inclined to load more items in visual working memory and guess more, leading to more errors. This is a problem for our model since it has no way of properly accounting for motivation. Conversely, if users are highly motivated and interested in the task, errors should be low and our model would, we expect, be quite accurate.

We would also like to suggest that our result can be used as a general design heuristic, without any specific modeling, for any application where visual comparisons are important. The design heuristic is as follows: if more than a two or three shape features are required for a comparison, adding extra views may be warranted if the alternative is flipping between web pages, zooming, or any other navigation method requiring more than a second or two. This leads to the question of "what is a visual feature?" In our studies we used simple geometric shapes, following the lead of most researchers in perception. However, it seems likely that visual working memory capacity is similarly limited for patterns that we would not normally call objects. For example the exact shape of a crack in a rock, or a particular fork structure in a node-link diagram. Some results from perception research relate to this issue. For example Sakai and Inui [2002] showed that about 4 convex contour bends could be stored in visual WM. And Phillips [1974] showed that a 4 × 4 pattern of random black and white squares could not be stored reliably. (It should be noted that random squares in a 4 × 4 grid normally fuse into a small number of rectangular areas and so their results are roughly consistent with the later studies that suggest a capacity of three items.) In most cases, however, the capacity of visual working memory to hold a particular pattern is unknown. The recourse of the designer, then, would be a judgment guided by the knowledge that visual working memory can hold only two or three quite simple visual components.

There are many applications where visual comparisons are common, including online shopping, information visualization, and geospatial data visualization. In some cases, to be sure, verbal working memory can take over the short-term memory burden. Once identified and named, arbitrarily complex patterns require no visual working memory capacity. In such cases the working memory burden may be moved entirely from visual working memory to become a single chunk in verbal working memory. However, even though a biologist might identify a protuberance on a bacterium as a "flagellum," thereby offloading that feature to verbal working memory, aspects of its shape (curvature and thickness) might also be visually encoded for comparison.

ACKNOWLEDGMENTS

The authors would like to thank Roland Arsenault for his integral support in developing the GeoZui3D system, and to Jon Gilson and Hannah Sussman for their help in running the experiments.

REFERENCES

Ahlberg, C. and Shneiderman, B. 1994. Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In *Human Factors in Computing Systems CHI'94 Proceedings*. ACM Press, 313–317.

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. In *Proceedings of Graphics Interfaces '94*. Morgan Kaufmann, Palo Alto, 216–224.

Bauer, M. I. and John, B. E. 1995. Modeling time-constrained learning in a highly interactive task. In *Human Factors in Computing Systems CHI'95 Proceedings*. ACM Press, 19–26.

Beard, D. and Walker, J. 1990. Navigational techniques to improve the display of large two-dimensional spaces. *Behavior Info. Tech. 9*, 6, 451–466.

Bederson, B. and Boltman, A. 1999. Does animation help users build mental maps of spatial information? In *Proceedings of the 1999 IEEE Symposium on Information Visualization*, 28–35.

Bederson, B. B. and Hollan, J. D. 1994. Pad++: A zooming graphical interface for exploring alternate interface physics. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'94)*. ACM Press, 17–26.

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

Carpendale, M. S. T., Cowperthwaite, D. J., and Fracchia, F. D. 1997. Extending distortion viewing from 2D to 3D. *IEEE Comput. Graph. Appl.*, July/Aug, 42–51.

Furnas, G. W. 1986. Generalized fisheye views. In *Proceedings of CHI'86*. ACM Press, 16–23.

Hornbaek, K., Bederson, B. B., and Plaisant, C. 2002. Navigation patterns and usability of zoomable user interfaces with and without an overview. *ACM Trans. Comput.-Hum. Inter. 9*, 4, 362–389.

Lamping, J., Rao, R., and Pirolli, P. 1995. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Human Factors in Computing Systems CHI'95 Proceedings*. ACM Press, 401–408.

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. In *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.

Perlin, K. and Fox, D. 1993. An alternative approach to the computer interface. In *Proceedings of Computer Graphics and Interactive Techniques (SIGGRAPH'93)*. 57–64.

PHILLIPS, W. A. 1974. On the distinction between sensory storage and short-term visual memory. *Perception and Psychophysics 16*, 283–290.

PLAISANT, C., CARR, D., AND SHNEIDERMAN, B. 1995. Image browsers: Taxonomy, guidelines and informal specifications. *IEEE Softw. 12*, 2, 21–32.

PLUMLEE, M. AND WARE, C. 2002. Zooming, multiple windows, and visual working memory. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI 2002)*. ACM Press, 59–68.

PLUMLEE, M. AND WARE, C. 2003. Integrating multiple 3D views through frame-of-reference interaction. In *Proceedings International Conference on Coordinated & Multiple Views in Exploratory Visualization (CMV 2003)*. IEEE, Los Alamitos, CA, 34–43.

SAKAI, K. AND INUI, T. 2002. A feature-segmentation model of short-term visual memory. *Perception 31*, 579–589.

SARKAR, M. AND BROWN, M. H. 1994. Graphical fisheye views. *Comm. ACM 47*, 12, December, 73–84.

SHNEIDERMAN, B. 1998. *Designing the User Interface*. Addison-Wesley, Reading, Mass.

SIMON, H. A. 1956. Rational choice and the structure of the environment. *Psychological Review 63*, 129–138.

SKOPIK, A. AND GUTWIN, C. 2003. Finding things in fisheyes: Memorability in distorted spaces. In *Proceedings Graphics Interface*. 47–55.

VOGEL, E. K., WOODMAN, G. F., AND LUCK, S. J. 2001. Storage of features, conjunctions, and objects in visual working memory. *J. Exper. Psycho.: Human Perception and Performance 27*, 1, 92–114.

WANG BALDONADO, M., WOODRUFF, A., AND KUCHINSKY, A. 2000. Guidelines for using multiple views in information visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI 2000)*. ACM Press, 110–119.

WARE, C., PLUMLEE, M., ARSENAULT, R., MAYER, L. A., SMITH, S., AND HOUSE, D. 2001. Data fusion for interpreting oceanographic data. *Oceans 2001*, Hawaii, CD ROM Proceedings.

ZHANG, W. AND LUCK, S. J. 2004. Do representations decay in visual working memory? *J. Vision, 4*, 8, 396a.

Received December 2004; revised October 2005, January 2006; accepted January 2006 by Susan Dumais