Zooming, Multiple Windows, and Visual Working Memory

Matthew Plumlee
Department of Computer Science
University of New Hampshire
Durham, NH, 03824, USA
E-mail: mdp@cs.unh.edu

Colin Ware
Data Visualization Research Lab
Center for Coastal and Ocean Mapping
University of New Hampshire
Durham, NH, 03824, USA
E-mail: colinw@cisunix.unh.edu

ABSTRACT
Zooming and multiple windows are two techniques designed to address the focus+context problem. We present a theoretical model of performance that models the relative benefits of these techniques when used by humans for completing a task involving comparisons between widely separated groups of objects. The crux of the model is its cognitive component: the strength of multiple windows comes in the way they aid visual working memory. The task to which we apply our model is multiscale comparison, in which a user begins with a known visual pattern and searches for an identical or similar pattern among distracters. The model predicts that zooming should be better for navigating between a few distant locations when demands on visual memory are low, but that multiple windows are more efficient when demands on visual memory are higher, or there are several distant locations that must be investigated. To evaluate our model we conducted an experiment in which users performed a multiscale comparison task using both zooming and multiple-window interfaces. The results confirm the general predictions of our model.

Keywords: Multiple windows, Zooming, visual working memory, interaction design, experiment, multiscale, multiscale comparison

INTRODUCTION
How does one work effectively where nuggets of useful information are widely separated in space? This is the motivating question of this paper and the heart of the focus+context problem. While techniques such as fisheye views, zooming, and multi-instance techniques (such as multiple windows) all address the focus+context problem, we have chosen to concentrate on methods that preserve spatial relationships: zooming and multiple-windows. We are concerned with understanding the characteristics of tasks for which multiple windows are more effective than a zooming interface and vice-versa.

Wang Baldonado et al [14] provide a number of design guidelines for how a given interface should use multiple windows. Of particular relevance to our task and methods 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"). An experiment performed by Plaisant et al [11] suggests a more specific guideline, that a maximum scale difference of 25x be present between a focus window and its parent. Experiments by North and Shneiderman [9] additionally suggest that coordination among multiple windows is essential to efficient performance for 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.

Meanwhile, zooming techniques have been developed as an alternative solution to the focus+context problem. Pad++ [3] made it easier for designers to integrate zoomable user interfaces (ZUIs) into their designs. Experiments performed by Guo et al [5] suggest that a zoom rate of 8x per second is optimal for ZUIs.

A few studies have been carried out that compare zooming and other focus+context techniques for certain tasks. Schaffer et al [12] compare zooming with a multi-focus fisheye technique they call *variable zoom*. Their experimental task is a directed multiscale search in a nested hierarchy of constant size, for which the variable zoom excels. Combs and Bederson [4] compare a 2D zooming image browser with two 3D image browsers (using carousel and landscape metaphors) and a traditional browser interface using a scrollbar. Their experimental task asks subjects to browse images in variously sized sets to find a target image. Their results suggest that the 2D browsers are more effective than the 3D browsers for larger image set sizes.

We believe that while there are a number of means by which multiple windows can aid users, these means fall into two general categories:

1. Spatial management of distinct contexts (as is applicable to window managers), and

2. Extension to visual and verbal working memory, either for alternate views of the same data, or for similar views of distinct locations—finding structure across scales, comparison at a distance, etc. (as is applicable to overview windows or split-screen views, respectively).

This paper addresses one subset of tasks in the latter category, namely 2D multiscale visual-comparison tasks. Our approach is to use known limits on human visual working memory (visual WM) to model expected performance for users in both zooming and multi-window interfaces. We then use this model to create an experiment that demonstrates the effects these limits have on user efficiency in performing a 2D multiscale comparison task using the two interfaces. Finally, we make inferences on how we expect our results would extend into other multiscale comparison tasks, both visual and verbal.

Cognitive Model of Working Memory

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 [8]). The models share a great deal in common, however, including the separation of working memory into components of limited capacity. To illustrate the concepts we are concerned with, we have chosen the *multiple-component* model of Baddeley and Hitch [1], recently updated by Beddeley and Logie in [8].

Figure 1 illustrates the parts important to our discussion. Selected sensory input, in the form of sounds and images, enters working memory where it can reside in the phonological loop (which we refer 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 components within long-term memory.

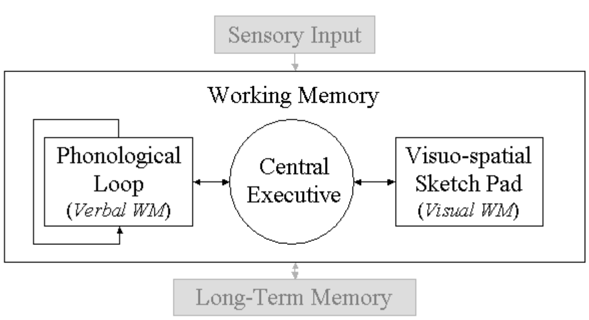

Figure 1: Adaptation of the multiple-component model of working memory from [8]. The model focuses on the components within the box labeled "Working Memory". The gray items above and below this box indicate the context in which working memory operates.

Before the multiple-component model, the limit on working memory was considered to be 7 items, plus or minus 2, as put forth by Miller [7]. Since then, it has become apparent that verbal WM and visual WM each has its own limit. It turns out 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 [2]. For a limit on the number of items that can be held in visual WM, we turn to recent work done using sequential comparison tasks.

Sequential Comparison Tasks

The sequential comparison experimental paradigm was introduced by Phillips [10] to discover the limits on visual working memory. The objective of a sequential comparison task is to determine whether or not a *probe* set of objects is the same as a *sample* set, or whether they differ in some way (for instance, the color of one of the objects). In experiments performed by Vogel et al [13], the sample set was displayed to a subject first, followed by a blank field and then the probe set. The blank field was displayed for roughly a second, which is long enough to clear iconic memory. Upon display of the probe set, the subject was asked to answers whether or not the probe set matched the sample set.

Vogel's experiments [13] used the sequential comparison design to contrast responses across differing set sizes. The drop-off in accuracy between sets of sizes 2, 3, and 4 suggests that visual WM in humans is limited to 3-4 objects at a time. The exact number is likely not an integer due to both between-subject differences and the imperfect nature of visual working memory. These experiments used colors to differentiate objects, but further experiments used attributes such as color, line orientation, and shape (see Jiang et al [6]), as well as certain conjunctions. The results indicate that the attribute does not affect the limit on visual WM, even when attributes are used in conjunction. Further, experiments performed by Jiang et al [6] suggest that our memory of objects is strongly tied to the objects' relative configuration. In other words, if the objects are arranged differently between the sample set and the probe set, then accuracy decreases in determining whether an object has changed or not.

To summarize, approximately 3 objects can be held in visual WM at a time, each object can be composed of several attributes, and the memory of these objects is strongly tied to their configuration. Based on these properties visual WM, we have developed a cognitive model of user performance for tasks that involve searching for visual patterns.

MODEL OF EXPECTED PERFORMANCE

In this section we first present a more general model, and then focus it on the particular task of multiscale comparison. Then we apply this model to two classes of interface: zooming and multiple-window. After drawing some conclusions from these models and presenting some caveats, we show how the model can be used to make actual predictions on the expected time to complete a task.

General Performance Model

Our general model for navigation-intensive information seeking is as follows:

$$T = S + V \cdot (B + D) \tag{1}$$

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 (task-dependent and interface-dependent),
>
> V is the expected number of visits made to different focus locations during the course of the task (task-dependent),
>
> B is the expected time to transit between visits (mostly interface-dependent, but also depends on distance and scale between visits in task), and
>
> D is the expected amount of time that a user will spend during a particular visit (task-dependent).

This model assumes a fairly homogenous task, where distances between focus locations are similar, and the amounts of time a user spends at a particular location are similar. It deals primarily in *expected* times, which allows it to be used for a wide range of purposes, from very general to very specific. The model basically asserts that the time it takes to complete a task is a linear function of the number of visits to various focus locations 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). An effective interface should have a low value for both B and S. Along these lines, the relative size of B and D also indicates 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.

Cognitive Model of Visual WM: The Number of Visits

Visual information seeking requires that a user identify particular patterns in an information display. This implies that these patterns are held (in some form) in visual WM so that a comparison can be made against patterns that become visible during the search process. The work of Vogel et al [13], reviewed in the introduction, suggests that there are strict limits in the number of visual objects that can be held in visual WM. It is our contention that these limits can be critical in the relative effectiveness of different interfaces.

For example, consider two patterns consisting of multiple objects. If navigation between them can be reduced to eye movements, there should be no distractions to using the maximum capacity of visual WM for making the necessary comparisons. In cases where the pattern is complex and contains more than about 3 objects (the capacity of visual WM), several trips back and forth must be made in order to confirm that the two patterns are the same or different. Interfaces that allow rapid eye-movement comparisons should therefore become superior as the complexity of the visual comparison increases. Interfaces that require more involved navigation may require some of the visual WM resource (thereby increasing the number of trips required), but they would certainly require more time to use than saccades of the eye.

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 a formal analysis of the number of trips needed based on visual WM limitations alone, assuming perhaps that verbal WM is already engaged for other purposes.

The capacity of visual WM plays a key role in the number of visits that a person makes over the course of information seeking. To calculate the number of visits that might occur in general, we first determine V_p: how many visits we would expect to occur during the comparison of two sets of objects. The key factors are n, the number of objects each set, 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.

If the sets of object 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.

$$V_{match} = 1 + \left\lceil \frac{n}{M} \right\rceil \tag{2}$$

The addition of one is required for the initial visit to the first set of objects.

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. We can find this probability by partitioning the objects into groups of size M, with one group having a remnant $r \leq M$. The probability of any partition containing the differing object is M/n, with the remnant partition having a probability of r/n. This partitioning is demonstrated in Figure 2. To find the expected number of visits when the sets differ by one, which we denote by V_{differ}, we sum the products of each probability with its associated visit number. Again, we add one to the final result for the initial visit to the sample set.

$$V_{differ} = 1 + \left(\sum_{i=1}^{\left\lfloor \frac{n}{M} \right\rfloor} \frac{M}{n} \cdot i \right) + \frac{r}{n} \cdot \left(\left\lfloor \frac{n}{M} \right\rfloor + 1 \right) \tag{3}$$

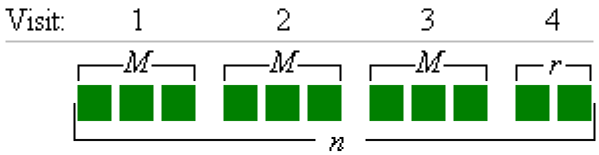

Figure 2: Determining the expected number of visits by partitioning the probability of finding the differing object during a visit.

Substituting for r and reducing, we obtain the following.

$$V_{differ} = 2 + \left\lfloor \frac{n}{M} \right\rfloor - M \cdot \frac{\left\lfloor \frac{n}{M} \right\rfloor + \left\lfloor \frac{n}{M} \right\rfloor^2}{2n} \tag{4}$$

If n is a multiple of M, this reduces to

$$V_{differ} = \frac{3}{2} + \frac{n}{2M} \qquad | \, n = kM, k \in \mathrm{I}\,. \tag{4b}$$

Multiscale Comparison Tasks

To demonstrate how to use our performance model, we have chosen to apply it to task we have termed the *multiscale comparison* task. A multiscale comparison task is similar to a sequential comparison task in that it asks a user to compare probe object sets to sample object sets. However, in this task there may be more than one probe set, and the sample and probe sets are 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 the scale of their individual objects to make the comparison directly. Whereas in the sequential comparison task, the user has no control over visits to the sample or probe sets, the performer of a multiscale comparison task is allowed to revisit sample and probe sets as often (and as long) as necessary to make a match determination. The multiscale comparison task is intended to bear more 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.

$$T = S + f(P, V_p) \cdot (B + D) \tag{5}$$

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 (we would expect these visits to be between the sample set and some probe set),

f is a function that calculates the total number of expected visits given P and V (based on cognitive modeling),

D is the expected time for the user to make a match determination.

In order to better define the visit-function f, we must choose a strategy for completing the multiscale comparison task. The strategy we have chosen to investigate consists 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 the last M objects remembered from a differing set can be carried to the next. 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 times the number of visits for each of these sets, plus the number of visits required for a set that matches the sample set:

$$f(P, V_p) = \left(P \cdot (V_{differ} - 1) + V_{match} \right). \tag{6}$$

By using Formulas 2 and 4b, and assuming n is a multiple of M, but leaving P in place for notational convenience, we get

$$f(P, V_p) = \frac{(2 + P) \cdot (M + n)}{2M} \qquad | \, n = kM \tag{6b}$$

Applying the Model to Specific Interfaces

To this point, then, we have constructed a performance model based on parameters that account for both the interface and the task. We have further refined the task parameters for the multiscale comparison task, taking into account limits on visual WM. We now turn our attention to refining the parameters for the individual interfaces.

Remembering back to the descriptions of Formulas 1 and 5, 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:

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

where B_{zoom} is the expected cost of using the zooming interface to get from set to set, and S_{zoom} includes little more than the cost of a user orienting him or herself to the initial configuration of the sets. By substituting Formula 6b in for the visit-function, we get

$$T_{zoom} = S_{zoom} + \frac{(2+P)(M+n)}{2M}(B_{zoom}+D) \quad |\, n = kM \qquad (7b)$$

For interfaces relying 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. 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 possibly be held in memory all at once. Here is how our model applies to a multiple-window interface:

$$\begin{aligned} T_{multi} = {} & S_{eye} + f(P,V_p)\cdot(B_{eye}+D) + \\ & S_{multi} + f'(P,V_p)\cdot(B_{multi}+D') \end{aligned} \qquad (8)$$

We 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 we assume that the setup cost S_{multi} includes situating the first two windows over their respective targets, then $f'(P,V_p) = P$, since there is no need to situate a window over subsequent probe sets more than once. Therefore, formula 8 can be reduced to

$$T_{multi} = S_{multi} + P\cdot B_{multi} + f(P,V_p)\cdot(B_{eye}+D) \qquad (9)$$

By substituting Formula 7b in for the visit function, we get

$$\begin{aligned} T_{multi} = {} & S_{multi} + P\cdot B_{multi} + \\ & \frac{(2+P)(M+n)}{2M}\cdot(B_{eye}+D) \quad |\, n = kM \end{aligned} \qquad (9b)$$

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. V_p and $f(P,V_p)$ can also be calculated, based on the number of items in each sample and probe set, as well as the number of items that can be held in visual WM at a given time. Because this calculation is non-trivial, we explain how it can be done presently.

Comparing Performance Models

We now have all the components in place to begin making theoretical comparisons between zooming and multiple window interfaces, as they apply to the multiscale comparison task. The extra terms in formula 9 beyond those in formula 7 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 these trips to be taken with the eye (B_{eye}). If we look at each S as the intercept of a line, and the slope as proportional to (B+D), we see that the slope of Formula 7 is steeper than the slope of Formula 9. 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 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.

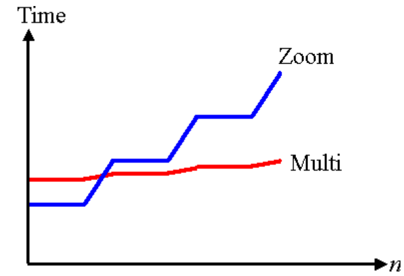

Figure 3: Expected relationship between performances in completing a multiscale comparison task when using zoom and multiple window techniques.

Model Caveats

Even before validating these models with experiments, there is some analysis that can be done relating to omissions from the models based on our incomplete understanding of memory. For instance, there is no term for errors in memory, errors in discernment, or use of sub-optimal strategies. Without knowing how to model these, we are left to postulate that such errors might manifest themselves as higher than expected numbers of visits, $f(P,V_p)$. This would serve to further increase the apparent differences in slope between the two techniques.

A potentially more interesting omission is the effect of these techniques on visual WM, especially since most of these omissions would detract from the usefulness of the zooming technique. Either method might use a "slot" within visual WM, for example to remember which probe set is currently being compared (with a zooming interface); visual memory might decay over the time period of a zoom; intermediate images seen during zooming might interfere with visual WM. Again, 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 possible that the linear relationship between n and $f(P,V_p)$ would become quadratic, or worse.

SAMPLE MODEL APPLICATION

The specific application of the model that we lay out here serves two purposes. First, it provides an example of the way the model can be used to predict performance with real numbers. Second, the values we choose for parameters to reach this level of specificity are those that correspond to the experiment that we will describe in the next section. From the work of Vogel et al [13] we know that we should estimate the capacity of visual working memory, M, at 3 (if we are forced to pick 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, we assume that D is a constant 0.8 seconds.

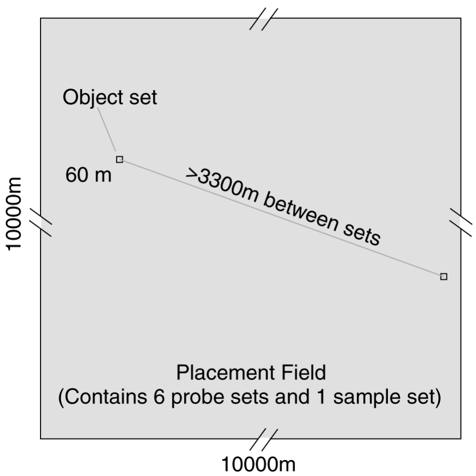

Figure 4: Schematic of the multiscale comparison task used in the sample model application and the experiment.

Consider the following scenario for our multiscale task, as diagramed in Figure 4. Let each object be a size that fits within a circle of 15-meter diameter. Let the size of the object sets be 60 meters, and let the minimum amount of space between them 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). With these parameters in place, the scales at which the objects in a set can be made out are roughly between 10 cm/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).

With this task scenario in place, we now choose operating parameters for each individual navigation techniques. For the zooming technique, we select a zoom rate of approximately 7x/s (7 times magnification per second). We estimate that a person can inspect a set at around 45 cm/pixel, and can zoom out to about 15 m/pixel to see the entire placement field. The average 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. We estimate the time to move the cursor this distance and press a mouse button to start a new zoom at about 1.5 seconds. Then we can expect that B_{zoom}=2[$\log_7$ (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 we assume is roughly 2 seconds. Using all this information, and letting the number of items be n=3, we can use Formula 7b to get an estimate:

$$T_{zoom} = 2 + \frac{\left(2+\frac{6-1}{2}\right)(3+3)}{6}(5.2+.8) = 29.0 \qquad (10)$$

For the multiple-window technique, we choose fixed scales for the overview and focus windows at about 17.5 m/pixel and the equivalent of 10 cm/pixel, respectively. We say *equivalent* of 10 cm/pixel for the focus window because the user can resize a focus window arbitrarily; if the user resizes this window to one-fourth the size of the overview (in at least one linear dimension), then the true scale would be 40 cm/pixel. We estimate the overhead to create, resize, and maintain proper positions of these focus windows at roughly 10 seconds per window. If we assume that there are two focus windows to be used (the optimum strategy for our 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 6). 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, we estimate the expected time to move a proxy from probe to probe at about 2 seconds per visit. This translates into a B_{multi} of 2 seconds. The final parameter we require 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 between .02 and .1 seconds. If we estimate our B_{eye} = .1 second, and remember that we have already chosen D = .8, we can use Formula 9b to get an estimate when n=3:

$$T_{multi} = 22 + \frac{6-1}{2} \cdot 2 + \frac{\left(2+\frac{6-1}{2}\right)(3+3)}{6} \cdot (.1+.8) = 31.05 \qquad (11)$$

Looking at the resulting numbers of Formulas 10 and 11, based on a number of estimates, does not tell us much. To get a better idea of what to expect, one should at least pick several values of n, and perhaps choose a couple values for M. Figure 5 gives the results of such a plot, for n between 1 and 8, and for M between 2 and 4. The plot suggests that we should expect zooming to become less efficient than using multiple windows at between 3 and 5 items.

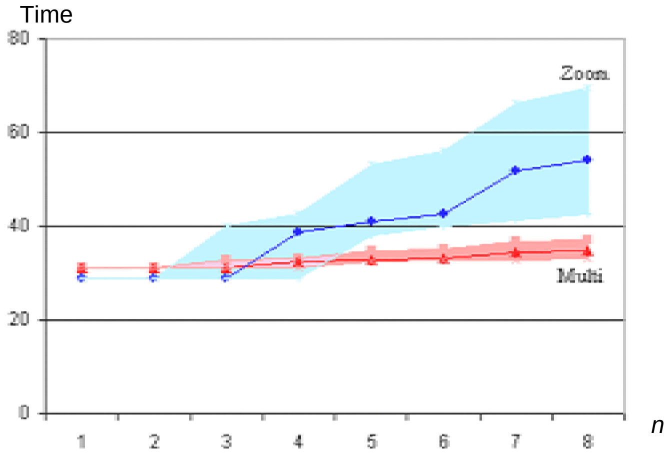

Figure 5: A refinement of Figure 3 using estimated parameters for each model variable. The heavy lines represent the values calculated for M=3. The light lines below and above the heavy lines represent the values calculated for M=2 and M=4, respectively.

EXPERIMENTAL VALIDATION OF MODEL

In order to provide some empirical evidence to support our model, we conducted an experiment contrasting the performance of subjects using both zooming and multiple-window interfaces to complete a multiscale comparison task. Our hypothesis was that, as illustrated in Figure 5, multiple windows would be slower than zooming when the number of items per set was low, and faster than zooming as the number of items increased past the maximum capacity of visual WM.

Experimental Design

The layout of the experiment consisted of a textured 2D background upon which seven clusters of objects were randomly placed, as shown in Figure 6. One cluster was the sample set. The sample set had a random arrangement of *n* objects, and was identifiable by its yellow border. The other clusters were the probe sets, each of which had a gray border. While every probe set had the same number and arrangement of objects as the sample set, 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 7). The sizes and placement parameter values for object sets were as described in the sample model application.

During each treatment, the subject was given one of two mechanisms for navigating the layout. The first was a zooming mechanism, which we refer to as *zoom*. 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.

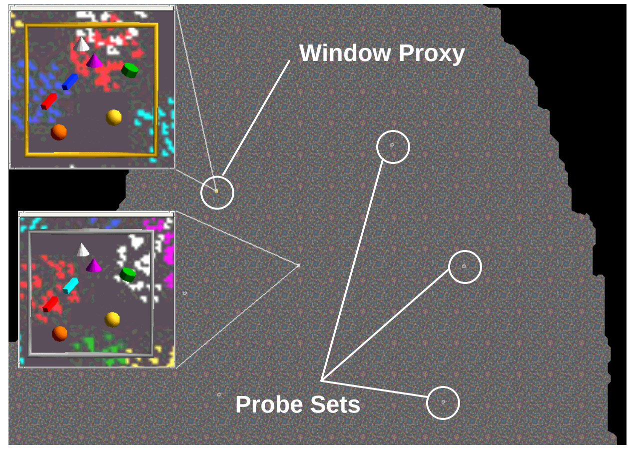

Figure 6: Example of the *multi* condition with two windows created. One window is focused on the sample set, while the other is focused on its match.

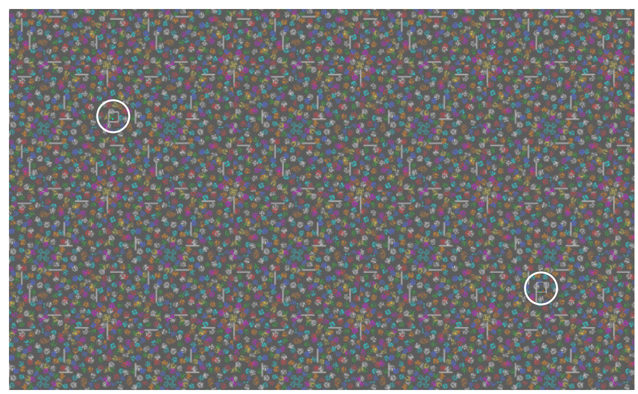

Figure 7: Two object sets camouflaged in the texture of the background at an intermediate scale during a *zoom* condition. To see individual objects, the subject must zoom in. To pre-attentively spot the clusters, the subject must zoom out.

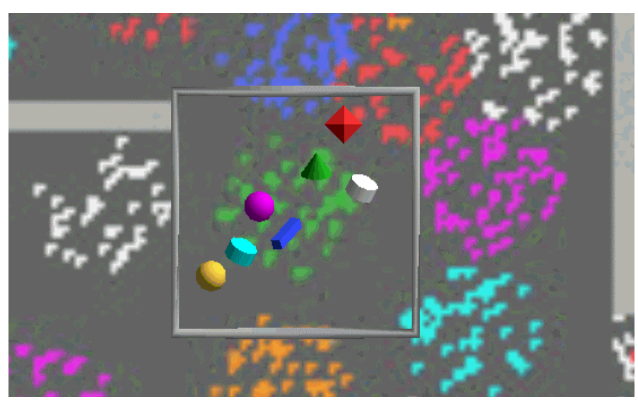

Figure 8: A visit to a probe set during a *zoom* condition.

The second mechanism for navigating the layout was through multiple windows. We refer to this mechanism as *multi*. 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 6. 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.

Prior to each treatment, 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 a simple list of words (for blocking verbal WM). Once the subject clicked the mouse, timing began for the treatment and the subject was presented with the layout at a scale such that the location of all seven sets of objects could be seen. 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 treatment. Otherwise, the subject was informed of the incorrect choice and the treatment was repeated with a new random layout and selection of objects.

Treatments

The experiment presented each subject with 8 representative training conditions and 4 blocks of experimental conditions, each containing 16 different treatments. All treatments varied in three parameters. We initially used {1,2,3,4} for the number of objects in each set (*n*) for the first 8 subjects, but later decided to investigate a larger range of values and changed this to {1, 3, 5, 7} for an additional 9 subjects. The navigation mechanism (*m*) was either *zoom* or *multi*. The verbal WM blocking condition (*b*) was either *blocked* or *unblocked*. So as to reduce user confusion in switching between mechanisms, each block was split into two groups such that all *zoom* treatments were grouped together and all *multi* treatments were grouped together within the block. Each treatment group was again split into two subgroups such that all *blocked* treatments were grouped together and all *unblocked* treatments were grouped together. The groups and subgroups were counterbalanced across the four treatment blocks and the order of the four values for *n* varied randomly within each subgroup.

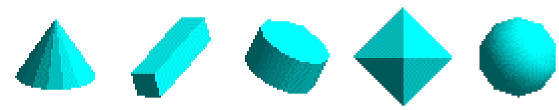

Figure 9: The 5 shapes that were available for creating each object set.

For each treatment, 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 9) 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 treatment. 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.

Results

The experiment was run on 17 subjects: 9 male and 8 female. 8 subjects were run with *n* confined to {1, 2, 3, 4} and 9 subjects were run with *n* confined to {1, 3, 5, 7}. We discarded as outliers all 23 trials that ran longer than 90 seconds (20 of which were *zoom* conditions), leaving us with 1064 data points. The results are summarized in Figure 10. As predicted, there was a crossover in efficiency between the two navigation methods between 3 and 5 items per set, and some of the stepwise flavor is evident in the shape of the graph.

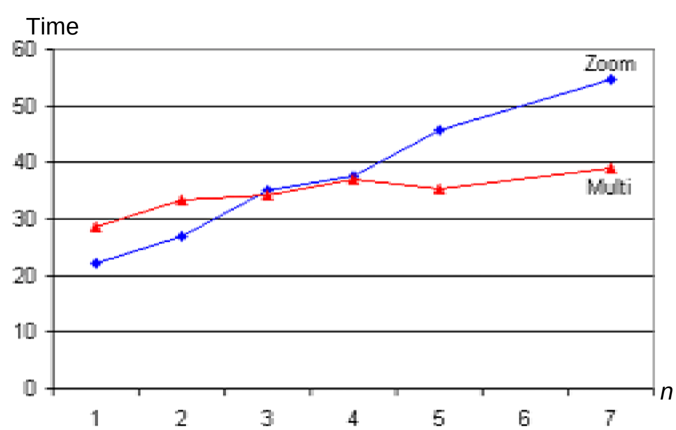

Figure 10: Results of the experiment, plotting the average time to complete a task for various values of *n*. The *zoom* condition exhibits a greater slope than the *multi* condition.

Errors, in which the subject selected the wrong probe set as a match for the sample set, are not included in Figure 10 but are represented in Figure 11. Figure 11 presents the percentage of errors generated by subjects in the context of trials being repeated until a correct match was indicated. For instance, an error rate of 25% for a given value of *n* would indicate that there was one false match for every three correct matches. As the figure shows, the percentage of errors generally increased with *n*, but the percentage was much greater for the *zoom* condition than the *multi* condition.

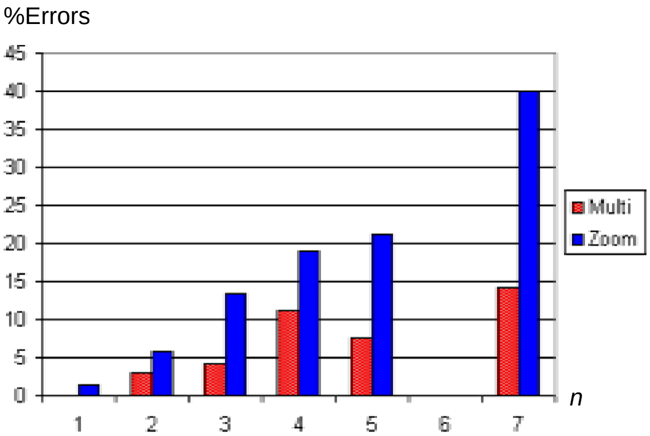

Figure 11: The percentage of errors for various values of *n*. The *zoom* condition exhibits a greater number of errors than the *multi* condition.

An analysis of variance based on our treatment variables 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 the variance in task completion time ($\mathbf{F}(5, 47) = 58.131$ and $\mathbf{F}(5, 47) = 10.428$, respectively, both with $\mathbf{p} < .001$). In addition, the interaction of verbal WM blocking with navigation mechanism (*b* x *m*) was significant: $\mathbf{F}(1, 20) = 9.55$ ($\mathbf{p} < .01$). Further ANOVAs run within each navigation method revealed that the blocking of verbal WM was not significant in the *multi* condition ($\mathbf{F}(1, 19)=1.607$), but it was in the *zoom* condition: $\mathbf{F}(1, 19)=5.746$, ($\mathbf{p} < .05$).

While a linear regression was not wholly appropriate for this situation (because we were expecting a piecewise-linear result, among other violated assumptions of linear regression), we performed a linear regression on the means for each value of *n* to get some sense of the general slope and intercept of the experimental results. For the *zoom* condition, the intercept is at 14.8 seconds, and the coefficient of *n* (the number of items in each set) is 6.34 seconds per item. For the *multi* condition, the intercept is 28.6 seconds, and the coefficient of *n* is 1.71 seconds per item. The $\mathbf{R}^2$ values for each regression were .981 and .818, respectively. The slope and intercept values obtained this way are similar to the values we would obtain by fitting a line to the predictions of our model.

Further analysis indicated that the crossover point between navigation methods was highly dependent upon the number of object sets that were actually visited during the course of the task. Zooming was effective for higher values of *n* when the number of object sets visited was low, and was ineffective for lower values of *n* when the number of object sets visited was high.

DISCUSSION

The results of our experiment support the predictions of our model, 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 significance of blocking verbal WM in the *zoom* condition and the absence of significance in the *multi* condition offers further support for our model, since the model predicts that an expanded memory capacity would only significantly reduce the amount of navigation in the *zoom* condition.

There were large differences between the two interfaces in terms of the numbers of errors that occurred as shown in Figure 11. The errors we have reported are false positives—cases in which a match was signaled for a non-matching set. False negatives, in which a match initially went undetected, likely showed up as longer task completion times (probably including but not limited to the pruned outliers). Since most of these errors occurred in the *zoom* condition, these errors may be due to interference or decay of visual WM, due to the delay and visual distractions caused by navigation. They may also be due in part to errors in perception (before encoding in visual WM) or other imperfections in the visual WM system.

Because the predictions of our 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 or overview windows.

CONCLUSION

We have presented a theoretical model of performance that contrasts the relative efficiencies of zooming and multiple windows for aiding users in completing a multiscale task. Our model has shown how limits on visual WM play a crucial role in determining these efficiencies, and how individual differences in these limits have an impact. Our experiment has provided empirical support for our model, illustrating the tradeoffs predicted by the model: zooming is more effective for comparing small sets of objects, while multiple windows are more effective when an object set is too large to fit in visual WM. Our work shows the main disadvantage to multiple windows to be the additional setup cost. This suggests that it will be useful to explore ways of streamlining window creation and removal. We have also touched on how the model can be used to explore how results might differ with changes to the parameters of the task or of the interface design. Finally, we have mentioned how our model should extend to textual tasks due to limitation on verbal WM that are similar to those in visual WM. We believe that this work should be helpful to interface designers in deciding when and how to employ various navigation techniques, including zooming and multiple windows, in their interfaces.

ACKNOWLEDGEMENTS

This research was funded by NSF grant 0081292.

REFERENCES

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

2. Baddeley, A. D, N. Thompson, and M. Buchanan. Word Length and the Structure of Short-Term Memory. *Journal of Verbal Learning and Verbal Behavior*, *14*, 1975, pp. 575-589.

3. Bederson, B. B. and Hollan, J. D. 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, New York, 1994, pp. 17-26.

4. Combs, T. T. A. and Bederson, B. B. Does Zooming Improve Image Browsing? *Proceedings of the Fourth ACM Conference on Digital Libraries*. ACM Press, New York, 1999, pp. 130-137.

5. Guo, H., V. Zhang, and J Wu. The Effect of Zooming Speed in a Zoomable User Interface. *Student CHI Online Research Experiments (SHORE)*, http://otal.umd.edu/SHORE2000/zoom/, 2000.

6. Jiang, Y., I. R. Olson, and M. M. Chun. Organization of Visual Short-Term Memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 26(3), 2000, pp. 683-702.

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

8. Miyake, A. and Shah, P. *Models of Working Memory: Mechanisms of Active Maintenance and Executive Control*. Cambridge University Press, New York, 1999.

9. North, C., and Shneiderman, B. Snap-Together Visualization: A User Interface for Coordinating Visualizations via Relational Schemata. *Proceedings of AVI 2000*, Palermo, Italy, May 2000, pp 128-135.

10. Phillips, W. A. On the Distinction Between Sensory Storage and Short-Term Visual Memory. *Perception & Psychophysics, 16*, 1974, pp. 283-290.

11. Plaisant, C., D. Carr, and H. Hasegawa. 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, 1992.

12. Schaffer, D., Z. Zuo, S. Greenberg, L. Bartram, J. Dill, S. Dubs, and M. Roseman. Navigating Hierarchically Clustered Networks Through Fisheye and Full-Zoom Methods. *ACM Transactions on Computer-Human Interaction*, 3(2), June 1996, pp. 162-199.

13. Vogel, E. K., G. F. Woodman, and S. J. Luck. Storage of Features, Conjunctions, and Objects in Visual Working Memory. *Journal of Experimental Psychology: Human Perception and Performance*, 27(1), 2001, pp. 92-114.

14. Wang Baldonado, M. Q., A. Woodruff, and A. Kuchinsky. Guidelines for Using Multiple Views in Information Visualization. *Proceedings of AVI 2000*, Palermo, Italy, May 2000, pp 110-119.