Techniques for Interacting with Off-Screen Content

Pourang Irani	Carl Gutwin	Grant Partridge	Mahtab Nezhadasl
University of Manitoba	University of Saskatchewan	University of Manitoba	University of Manitoba
Computer Science	Computer Science	Computer Science	Computer Science
Winnipeg, MB	Saskatoon, SK	Winnipeg, MB	Winnipeg, MB
irani@cs.umanitoba.ca	gutwin@cs.usask.ca	umpartr3@cc.umanitoba.ca	umnezhad@cs.umanitoba.ca

Abstract

Many systems – such as map viewers or visual editors – provide a limited viewport onto a larger graphical workspace. The limited viewport means that users often have to navigate to objects and locations that are off screen. Although techniques such as zooming, panning, or overview+detail views allow users to navigate off-screen, little is known about how different techniques perform for different types of off-screen tasks, and whether one technique works well for all tasks. We carried out two studies to explore these issues. The first study compared the performance of three classes of techniques (zoom, overview+detail, and proxy) in six types of off-screen tasks. We found that the techniques show substantial differences across different tasks and that no one technique is suitable for all types of off-screen navigation. This study led to the design of two novel hybrid navigation techniques – WinHop and Multiscale Zoom – that combine properties of multiple simpler approaches in an attempt to broaden support for off-screen navigation. We carried out a second study to assess the hybrid techniques, and found that they do provide reliable performance on a wide range of tasks. Our results suggest that integrating complimentary properties from different approaches can significantly improve performance in off-screen navigation tasks.

Author Keywords

Navigation techniques, offscreen navigation, small displays.

1 Introduction

In many applications such as map browsers or visualization systems, the workspace can be larger than the user's viewport. In order to retrieve and inspect content in these systems, users spend a substantial amount of time and effort navigating to off-screen locations. Researchers have developed a variety of different navigation techniques to alleviate some of the problems with navigating large workspace on small displays. However, most studies have investigated the performance of these navigation techniques with only a limited range of tasks. For example, several studies have investigated scrolling, but primarily on tasks related to navigating to known off-screen content [7,10]; panning and zooming have been investigated primarily in the context of off-screen targeting and navigation [9,13], and focus+context techniques have been studied primarily with reading, targeting, and steering [3,8].

This approach has successfully demonstrated the performance benefits of different systems in particular situations, but it does not provide much information about what would be the best technique in a real-world setting. Although we have evidence about individual techniques with individual tasks, little is known about how off-screen navigation techniques perform across a wide range of tasks. This knowledge is crucially important for designers, who must choose techniques that can adequately support a range of user activities, rather than just a few tasks.

In this paper, we explore several off-screen navigation techniques and tasks within one main type of activity – that is, finding and making decisions about off-screen objects in a 2D workspace with a clear spatial reference frame. The canonical application for these activities is a map browser with specific objects of interest that appear as annotations. Within this domain, we explore several questions:

- Which techniques are best suited to which task types, and is there one technique that performs well on all tasks?
- What are the characteristic properties of the techniques that lead to success with particular tasks?
- Can we combine these properties in new techniques, to increase the range of off-screen tasks that are supported?

To investigate these questions, we first built a framework that describes three different classes of off-screen navigation techniques (time-multiplexing, space-multiplexing, and proxy-based), and two different classes of off-screen tasks: spatially relative and spatially absolute tasks. Relative tasks involve identifying relationships between objects in a workspace, and absolute tasks involve interpreting the relationship of an object to the workspace.

We then conducted two studies. In the first, we compared the performance of representative techniques from each class, on six different off-screen tasks (three spatially relative, and three spatially absolute). We found that none of these basic techniques perform well on all tasks: time-multiplexing techniques such as zooming perform better on spatially-relative tasks, and proxy-based techniques such as hop perform better on spatially-absolute tasks.

The results of the first study led to the development of two hybrid techniques that combine different principles in order to better support a range of tasks. Our second study compared these new techniques–WinHop and Multiscale Zoom–to the 'pure' techniques used in the first study. The results show that both hybrid techniques improved on the originals, and that one hybrid, Multiscale Zoom, performed as well as the best 'pure' techniques for all task types. Overall, this work demonstrates the importance of breadth in evaluating navigation techniques, and suggests that narrow techniques can be broadened by incorporating elements from other approaches.

2 Related literature

A number of existing navigation techniques can be used to interact with off-screen content. These techniques can be organized into three groups: time-multiplexing, space-multiplexing, and proxy-based techniques.

Time-Multiplexing Navigation

Time-multiplexing techniques allow users to interact with different regions of the workspace at different times – as a result, different views of the workspace are available in a serial fashion. Scrolling, panning, and zooming are the three most common techniques in this group.

Scrolling and Panning allow users to adjust the viewport without changing the scale of the view. Scrolling and panning have been studied extensively (e.g., [7,10]). However both require considerable effort and several variations have been developed to facilitate navigation [12,17]. *Zooming* allows people to navigate by changing the scale of the viewport. With a zoom technique, 'off-screen' is relative to the current zoom level – and if required, any amount of the workspace can be brought into view, albeit at the cost of detail [5]. The overviews that result from zooming out provide awareness of off-screen content to users, and these can perform better than regular scrolling systems [14]. However, to find a particular off-screen object from a set of candidates, the user may have to perform multiple zoom operations.

Space-Multiplexing Navigation

Space-multiplexing techniques allow users to concurrently view different regions of the workspace. The main method of showing multiple regions is to divide the viewport into two or more windows; as a result, these techniques use more display space than time-multiplexing techniques. Common space-multiplexing techniques include overview+detail systems, focus+context views, and portals.

Overview+detail techniques present a miniature view of the entire workspace in a small inset window [4,16]; the main display shows a zoomed-in view. Users move the detail view either by panning or by dragging a viewfinder in the overview. DragMag [20] is an overview+detail technique, but in which the main window shows the overview, and a smaller inset window shows a detail region. The detail window follows the viewfinder in the overview. Overview+detail views have been shown to be effective [3]; but they require additional cognitive overhead to switch between the different scales of the two views and occlude some of the context in the main window.

Focus+context techniques such as fisheye views [21] eliminate the need for multiple windows by presenting a distorted view of the entire workspace. They provide a smooth transition between an enlarged focus region and the surrounding context. The drawback with many focus+context views is that they can make tasks that require targeting more difficult [11], and the distortion caused by fisheye views can degrade performance in tasks that have a clear spatial component.

Portals allow the user to view remote content of large displays through a window that is overlaid on top of the user's viewport. With Frisbees [15], for example, users pan and zoom into the off-screen space using a porthole metaphor. WinCuts [19] allows users to interact with off-screen regions by providing a local replica of the off-screen content. Unlike most other space-multiplexing techniques, portals do not provide users access to the entire workspace. As a result, additional operations such as zooming in and out of portals are necessary to view the overall context.

Proxy-Based Techniques

The emergence of large screens and multi-display systems has led to proxy-based navigation techniques that bring distant objects closer to the user's interaction space. For example, drag-and-pop [2] and the vacuum filter [6] create local copies of distant objects, in response to a gesture from the user. These forms of interaction have shown significant savings in the time required to select distant objects in comparison to conventional dragging. However, neither drag-and-pop nor the vacuum filter was designed as off-screen navigation techniques. Hop [13] is a proxy-based technique that was developed to adapt the proxy approach to the needs of off-screen navigation. Hop uses halos [1] to provide awareness of off-screen objects, and allows users to create proxies by interacting with the visible halos.

3 Off-Screen navigation tasks

Within the general scenario defined earlier (objects of interest located on a spatially-organized workspace), there are a wide variety of tasks that users carry out with off-screen objects. There are many ways to categorize tasks, but in 2D workspaces a main distinguishing factor is the spatial relationship between objects and the space. From this perspective, tasks can either be spatially relative, involving relationships between two or more objects; or spatially absolute, involving relationships between an object and the underlying workspace. The list of tasks in each group is not exhaustive and many other spatial tasks are performed on graphical workspaces. However, the following list captures the types of tasks that are routinely performed in canonical object-workspace settings (such as those seen in a map browser application).

Spatially Relative Tasks

These tasks require people to determine and understand spatial relationships between objects in the workspace. Main types of spatially relative tasks include determining the proximity of an object to a reference point, the proximity of objects to one another, or identification of clusters of objects that match certain criteria.

Proximity to Point of Reference. Users often need to locate an object that is closest to a point of reference such as finding the bus stop that is closest to the user's current position. These tasks are carried out by first locating the point of reference, and then by searching outwards to locate candidate objects. For each candidate that is located, the user needs to remember the current best candidate; when all likely candidates are checked, the user can determine which was closest to the reference.

Proximity between Pairs of Objects. A number of tasks involve finding a pair of items that are close together – but location in the workspace is not important – such as locating a 3-star hotel that is close to a railway station. To complete this task, the user must locate all pairs of items in the workspace, perform distance comparisons to determine which candidate pair are closest together (or below some 'close enough' threshold), and remember the best pair.

Clusters. Cluster tasks are a more complex variation of the proximity between objects. These tasks involve locating an object in the vicinity of other objects. For example, a user may wish to locate a 4-star hotel that is near a supermarket, a bus route and a seafood restaurant. In this task, the user has to perform a visual query over the entire workspace to locate the required cluster of objects.

Spatially Absolute Tasks

Spatially absolute tasks involve determining the relationship of an object to the workspace that contains the item. Some examples of spatially absolute tasks include determining whether an item is in the workspace at all, the number of occurrences of a certain type of item, or the location of an object in the workspace.

Existence. A common question when browsing a graphical workspace is to ask whether a specific type of object exists. For instance, a user may want to determine whether a zoo, an art gallery, or a library exists on a map. In such tasks, the user scans the workspace until the desired object is found. In the worst, case the user needs to scan the entire workspace to locate the object.

Object Count. Another common task is to determine the number of objects of a specific type. For example, counting the number of 5-star hotels or cinemas may be necessary, if the user wishes to compare hotels or decide where to see a movie. In counting tasks, the user scans the entire workspace and mentally maintains a tally (and possibly the locations) of the objects found.

Location. Location tasks involve determining the position of an object in the workspace. This task is carried out at a particular level of granularity – that is, sometimes a user will need to know only the quadrant of the city in which an object appears, and sometimes a more detailed location is needed. In this task, the user scans the workspace until the object has been found, and then establishes the location of the target with respect to the entire workspace.

4 STUDY 1: a comparison of representative navigation techniques

The goal of the first experiment was to compare different classes of off-screen navigation techniques on a variety of different tasks. The results from experiment 1 also serve as a baseline for understanding the features of various techniques that make them suitable to different tasks.

Study 1 Methods

Participants

Eighteen volunteers (12 men, 6 women) were recruited from a local university. Ages ranged from 22 to 32 years (mean of 24.5 years). All participants were familiar with mouse-and-windows software (more than 8 hrs/wk); 10 were also familiar with mapping applications.

Navigation Techniques

We selected one navigation technique from each of the categories described above (time-multiplexing, space-multiplexing, and proxy-based).

Zooming was chosen to represent the time-multiplexing class, since prior results show that zooming is superior to panning and scrolling for off-screen navigation [13]. We implemented a two-level zoom, where users move from overview to full detail by clicking the barrel button on the tablet pen.

DragMag was selected from the space-multiplexing class; a pilot study with three participants suggested that this technique performs better than either an overview+detail display or a fisheye view. In our implementation, an inset detail window magnified the area below a viewfinder that users could drag in the overview. Users could select items in either the overview or the detail view.

Hopping was picked from the proxy class, since it is the only proxy-based technique that is designed for off-screen navigation [13]. Hop shows halos of off-screen objects; the user can invoke proxies by sweeping a ‘laser beam’ across the halos. Users can then ‘teleport’ to the actual location of any object by clicking on its proxy.

Apparatus

An experimental system was built in C#.NET, and deployed on a Windows Tablet PC with 1024×768 screen resolution. The system presented a simulated map-browsing application with a visual workspace that was larger than the viewport. The workspace contained a 2600×2400-pixel map and the viewport was 800×600 pixels; therefore, the map extended 900 pixels past the viewport in all directions.

The system also displayed several icons on top of the city map. Icons were 24×24-pixel orange squares that represented items of interest such as hotels, restaurants, and bus stops.

On each icon, a capital letter indicated the object category, and smaller symbols represented further information such as the number of stars for the hotel or the bus routes servicing a particular stop. Twenty-four icons were randomly placed in the workspace, including the particular target icons used in different tasks.

Tasks

Participants completed all of the six tasks described earlier. Off-screen navigation was required for all tasks. Trials were completed, depending on the task, either by clicking a target in the workspace, or pressing a button on the tablet.

- *Existence*. Participants were asked to determine if there was a four-star hotel icon on the map. There was a 50% chance of the target being present.
- *Location*. Lines were added to the map to divide it into a 3×3 grid. Participants were asked to determine which section of the map contained the four-star hotel.
- *Object Count*. Participants were asked to count the number of four-star hotels on the map. The system randomly placed 2-6 targets for each trial.
- *Proximity between Objects*. Participants were asked to find the four-star hotel that was closest to a metro station. The system randomly placed three metro-hotel pairs on the map. One pair was always clearly closer together than the others, so that no precise measuring was required.
- *Proximity from Reference*. Participants were asked to find the closest four-star hotel to the centre of the map. The system randomly placed three targets; one of these was clearly closer upon inspection.
- *Cluster*. Participants were presented with a set of targets (e.g., a four star hotel, a four star restaurant, and a metro station), and were asked to find a cluster of exactly these targets. The system randomly placed three clusters, of which only one contained the correct targets.

Study 1 Procedure and Design

The study used a 3×6 within-participants factorial design. The factors were:

- *Navigation technique*: Zoom, DragMag, Hop
- *Task*: Existence, Location, Object-Count, Proximity between Objects, Proximity from Reference, Cluster.

Navigation technique and task were fully counterbalanced using a Latin square. Within each condition, participants carried out one demonstration trial, one practice trial, and five test trials. The workspace was reset to its initial state (viewport centred) after each trial. Participants completed all six tasks in a condition before moving to a new technique (rests were provided between conditions).

With 18 participants, 3 navigation techniques, 6 tasks, and 5 test trials, the system recorded a total of 1620 trials. The study system collected completion times and error information for each target.

Study 1 Results

Completion Time

A repeated-measures ANOVA showed significant main effects of both *navigation technique* ($F_{2,34}$=15.17, p<0.001) and *task* ($F_{5,85}$=29.67, p<0.001). However, there was a significant interaction between navigation technique and task ($F_{10,170}$=37.88, p<0.001), and so our analysis is organized below by task.

For each task, we carried out a one-way ANOVA to look for effects of navigation technique. Figure 1 shows the average completion times by task and technique. There are obvious differences between the two main classes of tasks (absolute and relative), and so we organize the results by class (note, however, that we cannot collapse the data into these classes since there are different tasks in each group).

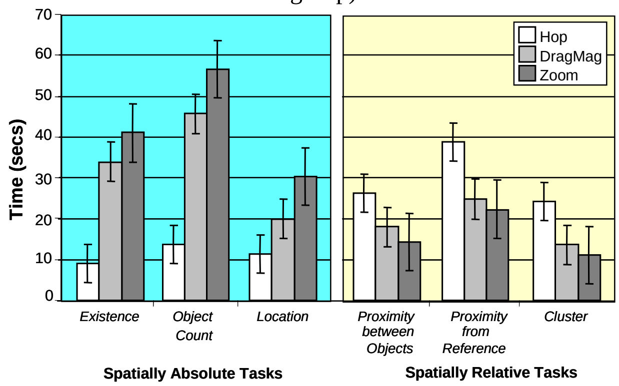

Fig. 1. Average completion time for each technique with each task. Error bars show standard error.

Spatially Absolute Tasks. For all three absolute tasks, one-way ANOVA found significant main effects of *navigation technique*: for Existence, ($F_{2,53}$=54.06, p<0.001); for Object Count, ($F_{2,53}$=50.635, p<0.001); for Location, ($F_{2,53}$=23.875, p<0.001). Tukey's post-hoc tests for all three tasks showed significant differences between all three techniques (Hop was significantly faster than DragMag, which was significantly faster than Zoom; all p<0.05).

Spatially Relative Tasks. For all three relative tasks, we also found significant main effects of navigation technique using a one-way ANOVA: for Proximity between Objects, ($F_{2,53}$=7.724, p<0.001); for Proximity from Reference, ($F_{2,53}$=9.544, p<0.001); for Cluster, ($F_{2,53}$=3.975, p=0.025). Post-hoc Tukey's tests show consistent significant differences between the fastest two techniques (Zoom and DragMag) and the slowest (Hop) (all p<0.05), but no differences between Zoom and DragMag.

The completion time results show that for the spatially absolute tasks, Hop was consistently fastest, followed by DragMag and Zoom (all differences significant). In spatially relative tasks, however, there was an exact reversal: Zoom was fastest, followed by DragMag and Hop.

Errors

A repeated measures ANOVA on error rates did not show a significant main effect of *navigation technique* ($F_{2,34}$=3.102, p=0.058) but did show a main effect of *task* ($F_{5,85}$=19.105, p<0.001). There was an interaction between navigation technique and task ($F_{10,170}$=4.114, p<0.001). Overall the error rates reaffirm the performance measure collected from the completion time data.

Study 1 Discussion

The main result of the first study is that there are strong differences in the effectiveness of the different techniques. For absolute tasks, Hop was best and Zoom was worst, and for relative tasks, the opposite ordering occurred. The limitations and strengths of the techniques in absolute and relative tasks provide guidelines for designing new off-screen navigation techniques. Our goal is to add elements together from different techniques to produce a hybrid that can potentially be effective for a wider range of tasks. In the next sections, we describe two hybrids: WinHop, which combines the proxy-based interaction of Hop with a space-multiplexing inset window; and Multiscale Zoom, which combines the time-multiplexing character of Zoom with the full-detail view of objects from proxy techniques.

5 Winhop

WinHop is an off-screen navigation technique that adds space-multiplexing and time-multiplexing elements to a proxy-based approach. Like Hop, it uses halos and proxies as the mechanism for finding off-screen objects. WinHop is space-multiplexing as it provides a view of off-screen regions through an inset window and is time-multiplexing as it allows zooming and panning in the portal.

Proxy-Based Characteristics

WinHop is an extension of Hop, and so shares that technique's basic characteristics (see Figure 2): (i) WinHop uses modified halos to inform users of the distance and direction to off-screen objects; (ii) users invoke proxies with a 'laser beam': the user places the pen on the workspace background and drags toward the edge of the screen, which draws a line to the screen edge. The user then sweeps the beam around with a circular motion, and as the laser beam touches a halo, the corresponding proxy is created near the pen; (iii) proxies are identical to the objects they represent, with the addition of a thin black and white border to distinguish them from true objects. A layout algorithm positions the proxies near the cursor, without occluding existing objects in the workspace; (iv) when the user moves the cursor over a proxy, WinHop provides additional information. To show direction, the system draws a thick arrowed line from the proxy to the actual off-screen object.

Space- and Time-Multiplexing Properties

In Hop, tapping on a proxy would teleport the user to the off-screen region. However, this approach is relatively slow (teleportation involves an animated scene transition), and participants sometimes felt that they got lost or disoriented through repeated teleportation. WinHop introduces a space-multiplexing inset window to let users explore the distant region without actually leaving their current location. When the user taps a proxy (Figures 2.a and 2b), a secondary viewport 'grows' out of the proxy (Figure 2.c. and 2.d); this new window teleports to the off-screen location, but nothing changes in the main view. The user may pan and zoom in the portal: panning by dragging the cursor; and zooming by moving a slider at the side of the portal window. The user can close the WinHop window by clicking on any region outside the window. Proxies that were previously displayed remain visible, so that the user can inspect other off-screen regions without re-sweeping the laser.

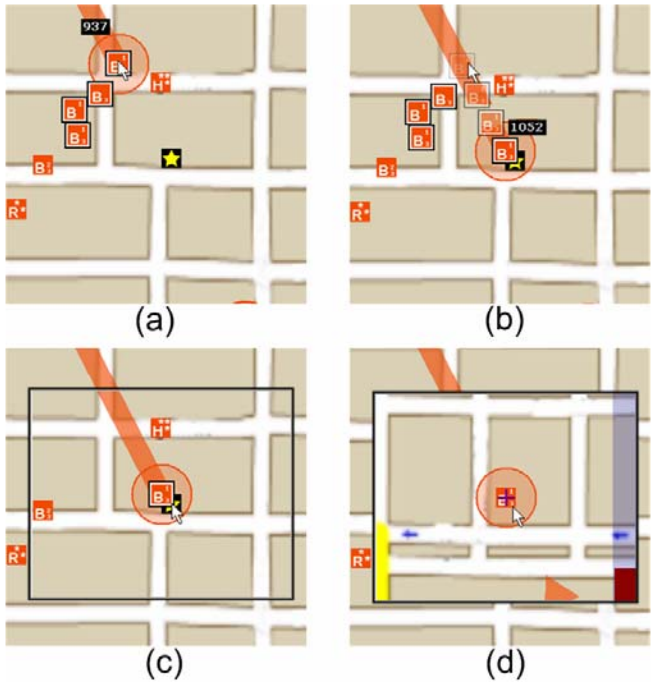

Fig. 2. The appearance of the WinHop window and translation from proxy to off-screen object. Clicking on a proxy (a), shifts the proxy to the center (b) and then opens a portal into the off-screen region around the object represented by the proxy (c & d)

6 Multiscale zoom

The main problem with the Zoom technique in the first study was that object details could not be seen in the overview (i.e., the zoomed-out view). Our second hybrid addresses this problem by incorporating full-detail object representations that are fundamental to proxy techniques.

The technique works by using different zoom functions for different elements in the workspace. In particular, object data has a greater endpoint, so that when the user zooms out to the overview, objects are not reduced in scale as much as the rest of the map (Figure 3). The end result is that objects remain above the threshold of visibility and readability in the overview. This idea is an extension of semantic zooming, which presents different representations at different scales (e.g., more detailed representations at larger scales). In our technique, it is the scaling rate of the zoom function that is 'semantic' – different data zooms at different rates. In the two-level zoom of the experimental system, we change the scale rates so that icon details remain visible in the overview, but are normally sized (with respect to the surrounding map) when zoomed in. We note that a simpler version of this idea has been seen in previous commercial applications: for example, mapping and GIS systems often lock the minimum size of the text tags and place names so that they remain readable at any zoom level.

Fig. 3. Overview (zoomed-out view) with conventional Zoom (left), and with Multiscale Zoom (right). In multi-scale zoom the objects maintain their original size.

Multiscale Zoom still preserves spatial relationships between targets (almost as well as regular Zoom), but also ensures that object details will be visible. This approach, however, leads to certain challenges. Since objects take up more space in the overview than they should, some maps will appear cluttered, and objects will occlude each other. To overcome the occlusion problem, our multiscale zoom brings objects to the top of the stack when the user hovers the cursor over them.

7 STUDY 2: Hybrid versus 'pure' navigation

We carried out a second experiment to determine whether the combination of elements allowed the two hybrid techniques to support a wider range of tasks. To compare the new techniques to those used in the first study, we asked participants from the first study to come back for the second, and then analyzed data from both studies together. (We recognize that there is a possible learning effect between the two studies that could improve relative performance on the new techniques; nevertheless, this method does at least allow the identification of large differences).

Study 2 Methods

Participants. Twelve subjects (8 male and 4 female) who participated in the first study volunteered to return.

Navigation Techniques and Tasks. WinHop and Multiscale Zoom were used as described above; the analysis also incorporated data from the earlier techniques (Hop, Zoom, and DragMag). Study two used the same six tasks described earlier, and participants also carried out the same number of trials.

Apparatus. The same tablet computer was used as that of Experiment 1; the custom study system was extended with implementations of the two new techniques.

Procedure and Design. Including data from the earlier study results in a 5×6 within-participants factorial design. The factors were Navigation technique (WinHop, Multiscale Zoom, Hop, Zoom, and DragMag), and Task (Existence, Location, Object Count, Proximity Between Objects, Proximity From Reference, and Cluster).

The second study, as mentioned above, gathered data from only the two new techniques (WinHop and Multiscale Zoom). For these two conditions, navigation technique and task were counterbalanced using a Latin square. With 12 participants, 2 navigation techniques, 6 tasks and 5 test trials, the system recorded a total of 720 trials.

Study 2 Results

Completion Time

The 12 participants from experiment one also participated in experiment two and therefore the analysis was performed across all techniques. A repeated-measures 5×6 ANOVA showed significant main effects of both *navigation technique* ($F_{5,55}$=14.738, p<0.001) and *task* ($F_{4,44}$=31.326, p<0.001). There was a significant interaction between navigation technique and task ($F_{20,220}$=23.315, p<0.001), and so our analysis is organized below by task.

For each task, we carried out a one-way ANOVA to look for effects of navigation technique. Figure 4 shows average completion times for task and technique by task category.

Spatially Absolute Tasks. For all three absolute tasks, one-way ANOVA found significant main effects of *navigation technique*: for Existence, ($F_{4,55}$=62.922, p<0.001); for Object Count, ($F_{4,55}$=44.265, p<0.001); for Location, ($F_{4,55}$=34.679, p<0.001). Post-hoc Tukeys's test for all three tasks showed significant differences between the fastest technique (Multiscale Zoom) and the slowest techniques (Zoom and DragMag; all p<0.001). Performance with Multiscale Zoom was also significantly faster than the two proxy-based techniques (WinHop and Hop; all p<0.05) in the Location task but there is no significant difference between Multiscale Zoom, WinHop and Hop for the Existence and Object Count tasks. sWinHop was significantly faster than both Zoom and DragMag in all tasks (all p<.001). There was no significant difference between WinHop and Hop.

Spatially Relative Tasks. For all the relative tasks, we found significant main effects of navigation technique: for Proximity Between Objects, ($F_{4,55}$=8.187, p<0.001); for Proximity from Reference, ($F_{4,55}$=12.196, p<0.001); for Cluster, ($F_{4,55}$=3.084, p=0.023). Post-hoc Tukey's tests show that Multiscale Zoom is significantly faster than all the other techniques in the Proximity between Objects and Proximity from Reference tasks (all p<0.05), but in the Cluster task Multiscale Zoom is only significantly faster than Hop (p<0.01). Interestingly, WinHop was significantly faster than Hop in the Proximity from Reference (p<.001) but not in the other two tasks. However, there were no significant differences between WinHop, DragMag and Zoom across all spatially relative tasks.

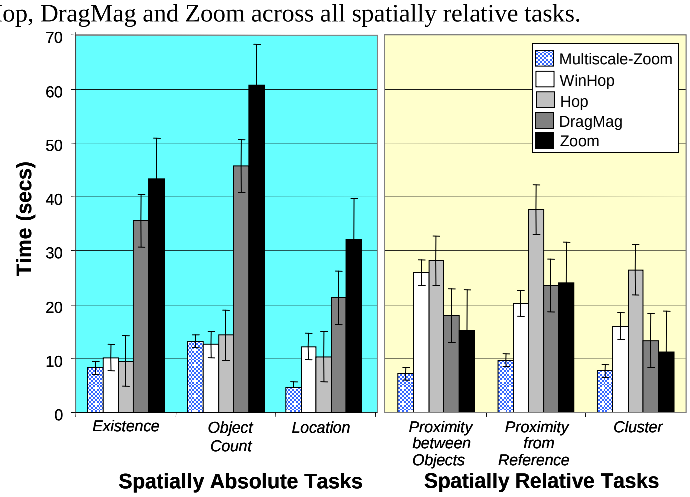

Fig. 4. Average completion time for each technique.

8 Discussion

The two studies described above provide answers to the questions posed at the beginning of the paper, and serve as an example of how a broader understanding of the characteristics of navigation techniques can be achieved by examining them over a wider range of tasks. The main findings are:

- No single basic technique performed well on all of the tasks, and performance was strongly affected by the characteristics of the task;
- WinHop offers improved performance on tasks where Hop performed poorly (relative tasks), and Multiscale Zoom dramatically improved performance on tasks where Zoom performed poorly (absolute tasks);
- Overall, Multiscale Zoom was the fastest technique, and its performance was consistent across all tasks;

In the following sections we reflect on the underlying goals of the two studies (testing breadth, and hybridization), and summarize the main lessons for practitioners.

Broader testing of navigation techniques

The exploration of the first study – to determine the performance of representative navigation techniques on six different tasks – provides an initial perspective of the uncharted territory between the known performance peaks for the three different navigation styles.

Whereas 'point studies' with carefully-chosen tasks allow researchers to establish that a novel technique is advantageous at a single location, survey studies of the kind carried out here help to show the regions of task space where a particular technique will be valuable. Comparing techniques on a range of tasks provided insight into some of the key strengths of different classes of techniques: Zooming is good for seeing spatial relationships, Hop provides key information on object details, and DragMag falls between Zoom and Hop across the range of our tasks. In addition, the first study identified characteristics of the techniques that led to the development of the new techniques tested in the second study.

However, the first study does not cover all possible tasks in visual workspaces – we focused on object-and-map systems and two main classes of tasks – and so further work is clearly needed. We plan to extend the investigation to activities such as measuring, steering, route-finding, and revisitation. Even though we did not test an exhaustive list of tasks, however, the first study provides an example of a methodological approach that can be used for further work.

Hybridizing navigation techniques

Identifying the strengths of different techniques in the first study led to two hybrid designs that combine elements from two or more techniques. The second study provides evidence that this process was successful: both WinHop and Multiscale Zoom appeared to make up for deficiencies in the techniques from which they were derived. In spatially-relative tasks, WinHop improved the performance of the proxy-based approach to a level that is comparable to Zooming and DragMag. Similarly, Multiscale Zoom provided an enormous improvement over regular Zoom in spatially-absolute tasks.

In particular, Multiscale Zoom worked well across the entire range of tasks in our study. However, this result must be tested across even more tasks – just as we cautioned that individual techniques can be over-fitted to particular tasks, it is also possible that Multiscale Zoom is over-fitted to the two classes of tasks that we studied. The potential limitation of

Multiscale Zoom is that in environments where many objects have been identified, the oversized overview icons can occlude the map, and can reduce object visibility when they overlap one another. Although there are solutions to these problems (as described above), we need to test this technique in other classes of tasks.

Nevertheless, augmenting navigation techniques by combining features is a promising design approach. We note that other researchers have also had similar success by integrating features of multiple techniques to improve performance [19]. However, the idea of hybridization also has limitations. Embedding a large number of features into a system creates overhead, increases training time, and can make simple tasks harder to accomplish. Clearly, there is a threshold beyond which adding new features to a technique will result in reduced performance. This threshold may have been crossed, in fact, in the design of WinHop: performance on the Location and Existence tasks actually decreased in comparison to Hop, and one explanation is that the new features required more effort (e.g., users had to open a portal before initiating a move toward the object). At the same time, however, we note that for this same task users made fewer errors with WinHop. This example suggests that integrating multiple features may enhance some performance aspects but at the expense of others.

Lessons for practitioners

This work provides three main lessons for designers of visual-workspace systems:

- Designers of object-and-workspace systems should consider using a hybrid technique – particularly Multiscale Zoom – to support offscreen navigation;
- Off-screen navigation techniques should show both spatial relationships and object detail with minimal navigation effort;
- Designers can develop new techniques by investigating the limitations of prior techniques with a range of tasks, but should add features cautiously to avoid reducing performance in other areas.

9 Conclusion and future work

Many techniques exist for navigating to off-screen content in a visual spatial workspace. However, any particular technique may not be suitable for a wide variety of tasks. We present the results of two studies that explored the effectiveness of different techniques for a wide range of off-screen navigation tasks. In the first experiment, we compared three techniques that represent three different approaches to off-screen navigation. In the second study, we tested two new techniques that are constructed by combining elements from the representative techniques. Both of these new techniques (Multiscale Zoom and WinHop) significantly improved user performance, particularly on tasks that are not easily supported by the earlier techniques.

In practical terms, designers cannot expect to produce a technique that fits all different possible off-screen navigation tasks. Similarly, we cannot expect users to switch between techniques to execute different types of tasks. At best, we can produce new techniques that are effective on many common tasks. With this outlook, our future work will proceed in three directions: studying navigation performance in other classes of tasks such as steering, measuring, and revisitation; studying the performance of hybrid techniques with datasets that can test their limits; and applying the idea of testing task breadth in an entirely different application domain, such as image editing or text browsing.

REFERENCES

1. Baudisch, P. and Rosenholtz, R., Halo: a technique for visualizing off-screen objects. Proc. CHI 2003, 481-488.
2. Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B., and Zierlinger, A., Drag-and-pop and drag-and-pick: techniques for accessing remote screen content on touch- and pen-operated systems. Proc. Interact 2003, 57-64.
3. Baudisch, P., Good, N., Bellotti, V., and Schraedley, P., Keeping things in context: a comparative evaluation of focus plus context screens, overviews, and zooming. Proc. CHI 2002, 259-266.
4. Baudisch, P., Xie, X., Wang, C., and Ma, W., Collapse-to-zoom: viewing web pages on small screen devices by interactively removing irrelevant content. Proc. UIST 2004, 91-94.
5. Bederson, B. and Hollan, J., Pad++: a zooming graphical interface for exploring alternate interface physics. Proc. UIST 1994, 17-26.
6. Bezerianos, A. and Balakrishnan, R., The vacuum: facilitating the manipulation of distant objects. Proc. CHI 2005, 361-370.
7. Cockburn, A. and Savage, J., Comparing Speed-Dependent Automatic Zooming with Traditional Scroll, Pan and Zoom Methods. Proc. CHI 2003, 87-102.
8. Gutwin, C., Improving Focus Targeting in Interactive Fisheye Views. Proc. CHI 2002, 267-274.
9. Gutwin, C. and Fedak, C., Interacting with big interfaces on small screens: a comparison of fisheye, zoom, and panning. Proc. Graphics Interface 2004, 145-152.
10. Hinckley, K., Cutrell, E., Bathiche, S., and Muss, T., Quantitative analysis of scrolling techniques. Proc. CHI 2002, 65-72.
11. Hornbæk, K. and Frøkjær, E., Reading of electronic documents: the usability of linear, fisheye, and overview+detail interfaces. Proc. CHI 2001, 293-300.
12. Igarashi, T. and Hinckley, K., Speed-dependent automatic zooming for browsing large documents. Proc. UIST 2000, 139-148.
13. Irani, P., Gutwin, C., Yang, X., Improving selection of off-screen targets with hopping. Proc. CHI’06, 299-308.
14. Kaptelinin, V., A comparison of navigation techniques in a 2D browsing task. Proc. CHI 1995, 282-283.
15. Khan, A., Fitzmaurice, G., Almeida, D., Burtnyk, N., and Kurtenbach, G., A remote control interface for large displays. Proc. UIST 2004, 127-136.
16. Lam, H. and Baudisch, P., Summary thumbnails: readable overviews for small screen web browsers. Proc. CHI 2005, 681-690.
17. Moscovich, T. and Hughes, J., Navigating documents with the virtual scroll ring. Proc. UIST 2004, 57-60.
18. Sarkar, M., and Brown, M., Graphical Fisheye Views of Graphs. CHI 1992, 83-91.
19. Tan, D., Meyers, B., Czerwinski, M., WinCuts: manipulating arbitrary window regions for more effective use of screen space. Proc. CHI’04, 1525-1528.
20. Ware, C. and Lewis, M., The DragMag image magnifier. Proc. CHI 1995, 407-408.