A Comparison of Navigation Techniques Across Different Types of Off-Screen Navigation Tasks

Grant Partridge[1], Mahtab Nezhadasl[1], Pourang Irani[1],and Carl Gutwin[2]

[1] University of Manitoba
Computer Science
{umpartr3, umnezhad, irani}@cs.umanitoba.ca
[2] University of Saskatchewan
Computer Science
gutwin@cs.usask.ca

Abstract. In many systems such as PDAs, users access data through a limited viewport. This means that users have to frequently navigate to regions that are off-screen to view important content. Many techniques exist for moving to off-screen regions; However, none of these have been evaluated across a range of different types of off-screen tasks. In this video, we demonstrate the effectiveness of several major off-screen navigation techniques across a variety of tasks. We also include two newly developed techniques – WinHop and Multiscale Zoom – that were based on complementary features of existing systems. Our results suggest that integrating complementary properties from different approaches can significantly improve performance on a wide range of off-screen navigation tasks.

1 Introduction

In many ubiquitous environments, the *workspace* can be much larger than the user's *viewport*. To perform retrieval and inspection tasks in these systems, users spend a substantial amount of time and effort navigating to locations that are outside the viewing space [6], commonly referred to as *off-screen locations*. Researchers have developed a variety of different navigation techniques to improve the performance of working with large workspaces on small devices. These navigation techniques include scrolling, zooming or panning. Such techniques are very common and are heavily embedded in map browsers such as Google Maps™. However, the performance of these navigation techniques have been explored with only a limited range of tasks [3,4,5,6].

Evaluating the performance of different techniques on a limited range of tasks has successfully demonstrated the performance benefits of different navigation systems in particular situations. However such an approach does not provide much information about what would be the best technique in a real-world setting. Therefore very little is known about how different techniques perform across a wider range of tasks. This knowledge is crucially important for software designers, who must choose techniques that can adequately support a range of user activities, rather than just a few tasks.

C. Baranauskas et al. (Eds.): INTERACT 2007, LNCS 4663, Part II, pp. 716 – 721, 2007.

In this video, we present the results of a study that explores the effectiveness of different off-screen navigation techniques across a variety of common tasks within a particularly common environment – a map browser with specific landmarks as semantic icons.

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. However, each of these techniques are clearly deficient for performing certain common tasks. Scrolling requires considerable effort for most location tasks. Panning can work well for small shifts in view, but degrades with larger workspaces. Unlike scrolling or panning, zooming allows users to view off-screen content in a non-linear fashion; that is, distant objects can be inspected before those that are nearby). However, to find a particular off-screen object from a set of candidates, the user may have to perform multiple zoom operations [3,6].

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 utilize more display space than time-multiplexing techniques. Common space-multiplexing techniques include overview+detail systems (which typically consist of an overview inset within a detailed view) [2], focus+context views [4], and portals [7].

Proxy-Based Techniques: The emergence of large screens has led to a class of techniques known as proxy-based navigation techniques that bring representations of distant objects closer to the user's interaction space [1]. These forms of interaction have shown significant savings in the time required to select distant objects in comparison to conventional techniques. However, since these systems are relatively new and have been designed for mostly large screens, very little is known about the effectiveness of such techniques.

While all three classes of techniques have been studied to some extent, the focus has been upon examination of performance concerning very specific user tasks, so little is known about how they perform on a wider range of tasks. Even less is known about performance of such techniques in small-screen environments.

3 Off-Screen Navigation Tasks

Numerous taxonomies could be constructed to categorize the wide variety tasks involving off-screen objects in 2D workspaces. One useful distinction involves a classification of tasks as either spatially relative, involving relationships between two or more objects, or spatially absolute, involving relationships between an object and the workspace.

Spatially Relative Tasks

Spatially relative tasks require people to determine and understand spatial relationships between objects in the workspace. The relationship between the objects and the workspace itself is not required to complete the task. Users were asked to complete the following spatially relative tasks:

Proximity between Objects. Participants were asked to find the four-star hotel that was closest to a metro station. The system randomly placed 3 metro-hotel pairs on the map. One pair was always clearly closer together than the others.

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 * hotel, a four * 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.

Spatially Absolute Tasks

Unlike spatially relative tasks, spatially absolute tasks involve determining the relationship of an object to the workspace that contains the item. The participants were required to complete the following tasks:

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.

4 Two New Off-Screen Navigation Techniques

In a preliminary study, we compared performance of three main techniques – Hop, Zoom and DragMag – across the various types of off-screen tasks. We found that 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 for absolute and relative tasks provided guidelines for designing two new off-screen navigation techniques: WinHop and Multiscale Zoom. Both these techniques inherited elements from prior navigation systems to form a hybrid with a purpose of greater effectiveness upon a wider range of tasks.

WinHop

WinHop is an extension of Hop, and so shares that technique's basic characteristics. Like Hop, proxies are made by sweeping a laser beam and intersecting it with various halos representing off-screen objects. However, instead of directly teleporting the user to an off-screen region as is the case in Hop, 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, a secondary viewport 'grows' out of the proxy; 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 (Figure 1).

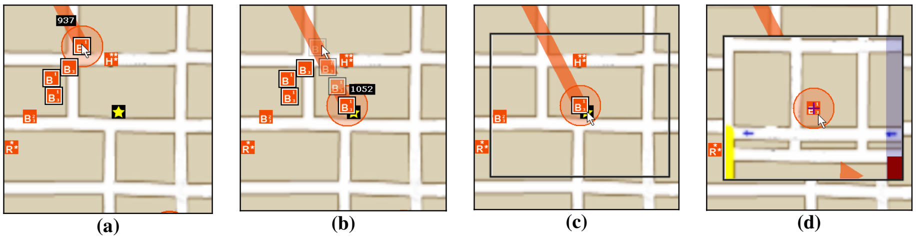


Fig. 1. The appearance of the WinHop window and translation from proxy to off-screen object. Clicking on a proxy (a), shifts the proxy to the center (b) and then opens a portal into the off-screen region around the object represented by the proxy (c & d).

Multiscale Zoom

The main problem with Zoom is that users cannot see sufficient details in the zoomed-out view. Multiscale Zoom 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. The end result is that objects remain above the threshold of visibility and readability

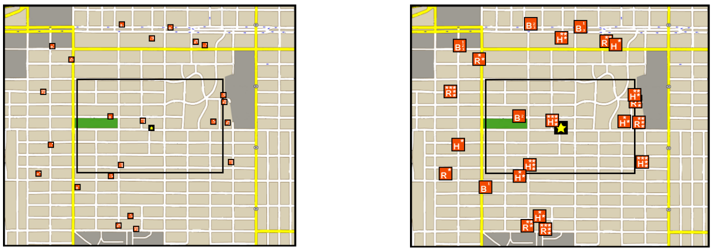


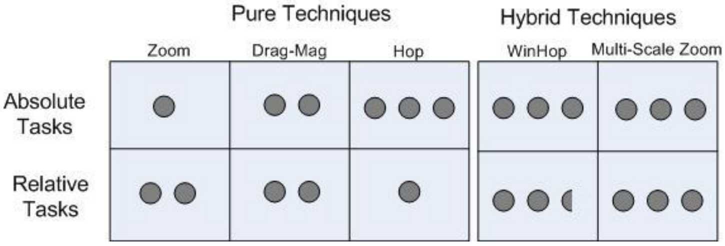
Fig. 2. Overview (zoomed-out view) with conventional Zoom (left), and with Multiscale Zoom (right). In multi-scale zoom the objects maintain their original size.

in the overview. Multiscale Zoom still preserves spatial relationships between targets (almost as well as regular Zoom), but also ensures that object details will be visible (Figure 2).

5 Results of the Study

We carried out an experiment to determine whether the two new hybrid techniques support a wider range of tasks. Twelve participants (8 male and 4 female) were involved. We compared WinHop and Multiscale Zoom to Zoom, Hop and DragMag. The study used the six tasks described earlier. The design consisted of 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). With 12 participants, 5 navigation techniques, 6 tasks and 5 test trials, the system recorded a total of 1800 trials.

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). In the video we provide a summary of the performance of each technique with respect to each of the tasks as shown in figure 3 below. Each circle represents the relative strength of a technique for a category of tasks.



	Pure Techniques			Hybrid Techniques	
	Zoom	Drag-Mag	Hop	WinHop	Multi-Scale Zoom
Absolute Tasks	●	● ●	● ● ●	● ● ●	● ● ●
Relative Tasks	● ●	● ●	●	● ● ◐	● ● ●

Fig. 3. The circles show the relative strength of each technique for the two different classes of tasks

6 Conclusion

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 designed two new techniques based on our observations of how users performed with prior techniques. Results of our experiment show that both of the new techniques (Multiscale Zoom and WinHop) significantly improved user performance, particularly on tasks that are poorly supported by the primitive 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 and select the most appropriate for a given application.

References

1. Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B., Zierlinger, A.: Drag-and-pop and drag-and-pick: techniques for accessing remote screen content on touch- and pen-operated systems. In: Proc. Interact 2003, pp. 57–64 (2003)
2. Baudisch, P., Good, N., Bellotti, V., Schraedley, P.: Keeping things in context: a comparative evaluation of focus plus context screens, overviews, and zooming. In: Proc. CHI 2002, pp. 259–266 (2002)
3. Cockburn, A., Savage, J.: Comparing Speed-Dependent Automatic Zooming with Traditional Scroll, Pan and Zoom Methods. In: Proc. CHI 2003, pp. 87–102 (2003)
4. Gutwin, C., Fedak, C.: Interacting with big interfaces on small screens: a comparison of fisheye, zoom, and panning. In: Proc. Graphics Interface 2004, pp. 145–152 (2004)
5. Igarashi, T., Hinckley, K.: Speed-dependent automatic zooming for browsing large documents. In: Proc. UIST 2000, pp. 139–148 (2000)
6. Irani, P., Gutwin, C., Yang, X.: Improving selection of off-screen targets with hopping. In: Proc. CHI'06, pp. 299–308 (2006)
7. Ware, C., Lewis, M.: The DragMag image magnifier. In: Proc. CHI 1995, pp. 407–408 (1995)