Peephole Displays: Pen Interaction on Spatially Aware Handheld Computers

Ka-Ping Yee

Group for User Interface Research
University of California, Berkeley
ping@zesty.ca

ABSTRACT

The small size of handheld computers provides the convenience of mobility at the expense of reduced screen space for display and interaction. Prior research [5, 6] has identified the value of spatially aware displays, in which a position-tracked display provides a window on a larger virtual workspace. This paper builds on that work by suggesting two-handed interaction techniques combining pen input with spatially aware displays. Enabling simultaneous navigation and manipulation yields the ability to create and edit objects larger than the screen and to drag and drop in 3-D. Four prototypes of the Peephole Display hardware were built, and several Peephole-augmented applications were written, including a drawing program, map viewer, and calendar. Multiple applications can be embedded into a personal information space anchored to the user's physical reference frame. A usability study with 24 participants shows that the Peephole technique can be more effective than current methods for navigating information on handheld computers.

Keywords

Mobile computing, spatially aware displays, 3-D drag-and-drop, two-handed interaction, personal information spaces.

INTRODUCTION

Recent years have shown an explosion of interest in handheld computing devices such as palm-size digital assistants and increasingly smart mobile phones. Their small form factor has the advantages of portability, low power consumption, and instant-on responsiveness, but also limits the size of the display. A key limitation of these devices is the user's inability to view and interact with a large amount of information at once.

Handheld computers employ various scrolling mechanisms to provide access to more information on their small displays, including buttons for moving up and down, a thumbwheel on the side of the device, or scroll bars on a touch-sensitive screen. A standard technique for viewing maps and photographs on touch-screens is to drag a pen to grab and pan the image.

Relative scrolling methods such as buttons and wheels can be slow for navigating long documents, since users may have to press a button or roll a wheel many times to cover large distances. Using a scroll bar or dragging to pan the view is disruptive because it forces users to interrupt the current pen interaction, divert their attention to the scrolling manoeuvre, and switch back.

On current devices, pen interactions cannot span distances beyond the screen unless the display automatically scrolls when the pen reaches the edge of the screen. However, auto-scrolling behaviour is notoriously difficult to control. The screen regions that trigger auto-scrolling are usually invisible, and often the view scrolls too quickly or slowly.

CONCEPT

One way to provide access to more information is to track the position of the display so it can be physically moved around to see different parts of a large workspace. This idea was proposed by Fitzmaurice [5] in 1993. This work takes that idea and explores what happens when we combine it with pen input and other interaction ideas such as the Toolglass [3] and the zooming UI [2, 14].

Though Fitzmaurice's prototypes displayed views on 3-D scenes, our starting point is a 2-D version of the spatially aware display. The information is spread out on a flat virtual workspace larger than the display, and the display shows a movable window (or "peephole") on the space.
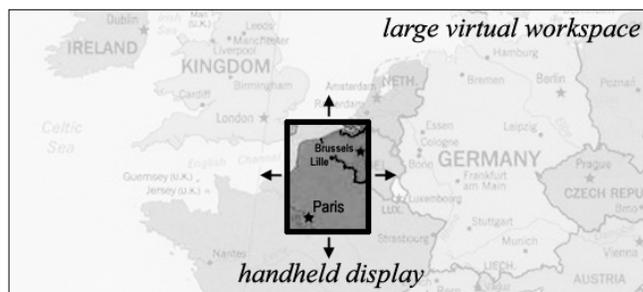

Figure 1. A Peephole Display on a larger workspace.

To create the illusion of a fixed workspace, the handheld computer scrolls the display opposite to the direction of its movement just enough to cancel its physical displacement.

Figure 2 shows an example of this method being used to view a large image. Panning typically involves both hands, but this method lets the user browse with one hand. Thus, the user can also manipulate information or other user interface widgets, using both hands together to

Figure 2. The images on the right were made by blending two photographs taken from the same viewpoint. The position of the device is tracked and the display scrolls to produce the illusion of a movable view on a large street map floating in space. Notice how Gravier St., visible in both views, maintains a fixed position with respect to the outside world.
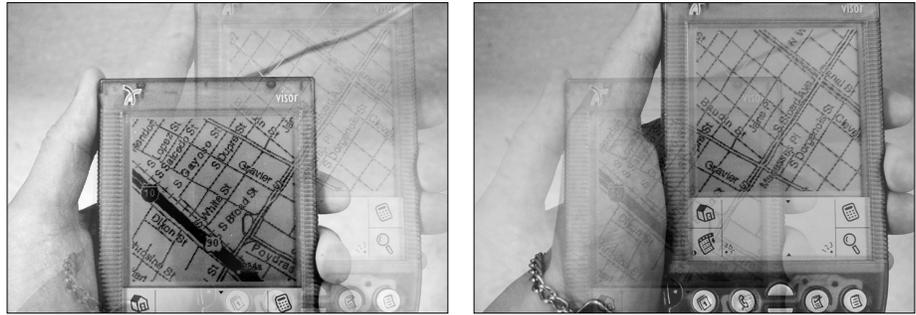
simultaneously navigate and interact within the workspace. In this way, we can augment the space around a user with information and user interface elements; [6] describes a similar concept. The handheld computer becomes a portable gateway to the user's personal information console.

We'll now continue with a survey of the related work, then a description of the 2-D prototypes and applications, the 2-D usability study, the 3-D prototype and applications, and finally a discussion of future directions.

RELATED WORK

Browsing Information on Small Screens
Previous work has proposed many compelling interaction techniques based on physical manipulation of a small-screen device, including contact, pressure, tilt, and motion.

Specifically with regard to navigation, Rekimoto [15] used tilt input for navigating menus, maps, and 3-D scenes, and Harrison et al. [7] and Hinckley et al. [9] have used tilt for scrolling through documents and lists. Peephole Displays fall into the category of spatially aware displays, which differ from the aforementioned work in that they create a positional mapping between the virtual space and the real world, enabling the use of spatial memory for navigation.

Another approach to fitting information on a small screen is to provide zoom control. Techniques for improving navigation on small screens include zooming UIs [2, 14] and speed-dependent automatic zooming [10].

Spatially Aware Displays
Fitzmaurice's Chameleon [5] and Ishii and Ullmer's activeLENS [11] are motion-tracked displays based on positional mapping, like this one. The Chameleon was the original Wizard-of-Oz implementation of a spatially aware display, in which a handheld colour TV showed an image from a camera pointed at a graphics workstation; the workstation rendered a 3-D scene with the view controlled by the position of the TV. The activeLENS is an armature-mounted full-size LCD screen, tracked in space by the joint angles on the arm. The Chameleon provides a single button for input; the activeLENS does not read input other than its position. Small and Ishii also experimented with tracking displays to control translation or zooming [17].

As the Peephole is also a spatially aware handheld display, some of its applications resemble ideas that Fitzmaurice suggested but did not implement for the Chameleon – such as interaction on a whiteboard mediated by a spatially aware display [5], a "virtual cubic spreadsheet" and its use

as a calendar [6], or an information space wrapped around a user [6]. The difference in approach is that, while in [5] Fitzmaurice proposes 3-D, virtual-reality-style mediation of an office environment, the Peephole designs and studies are more rooted in the rich heritage of interaction techniques for the desktop. Even when 3-D position information is used, the purpose is not to achieve depth perception (as is the focus of the experimental study in [6]).

Two-Handed Interaction
The advantages of two-handed interaction have been well studied [4]. In many asymmetric two-handed operations, the non-dominant hand provides a reference frame to situate the dominant hand's actions: for example, users can orient the work piece in the non-dominant hand while specifying operations with the dominant hand [8, 12, 16], or hold tools in the non-dominant hand for precise activation by the dominant hand [2, 12, 18]. This idea is extended here by using the non-dominant hand for navigating information spaces. Two-handed Peephole interaction benefits from a unified kinaesthetic reference frame [1].

Contributions of This Work
The Peephole Display is probably best described as a direct descendant of the Chameleon and the Toolglass [2]. In some ways, a Peephole Display is a physical realization of a Toolglass. This paper extends the previous work in a few new directions by contributing: (a) the use of *pen input* on a movable display as a new form of two-handed interaction; (b) an emphasis on, and implementation of, more typical handheld computer applications and *desktop-style metaphors*; (c) a *user study* to determine the validity of these techniques for PDA-like applications; (d) a more *portable* spatially aware display, in which the display is generated directly by the handheld computer itself; and (e) a working *implementation* of multiple applications arranged around the user in a personal information space.

Recent New Work
Between the submission and publication of this paper, work was published on two other spatially tracked displays with pen input. The Boom Chameleon [19] is an armature-mounted flat-panel screen for viewing 3-D scenes, just like the activeLENS, but with the addition of pen and audio input for making annotations. The Interaction Lens [13] is a handheld computer used as a lens for augmenting paper documents with digital ink annotations. The variety of recent work on spatially aware displays suggests that there are many exciting possibilities to be explored in this area.

2-D IMPLEMENTATION

The first prototype implementations of this technique were built using a Handspring Visor with a monochrome, 160-by-160-pixel LCD. The position of the Visor was tracked in two dimensions using three different methods.

Optical Mouse Tracking

For this implementation, the innards of an optical mouse were affixed to the handheld computer so that its motion could be tracked on a flat surface. Mouse technology is mature and cheap; this technique gives fast, reliable, and precise position data. However, it adds the limitation that the handheld computer has to be put down on an available surface before Peephole interaction is possible.

Ultrasonic Tracking

This method employed the commercially available Mimio whiteboard-capture system. The ultrasound transmitter from a Mimio marker was attached to the handheld computer. Position was computed from distance measurements obtained by the ultrasound receivers. This had the advantage that it allowed the handheld computer to move freely. However, the position readings were too slow and noisy to be effective, and tracking only worked while the transmitter was held exactly in the plane of the receivers.

Two-Tethered Tracking

This method used two lengths of monofilament fishing line and a mechanical mouse. In mechanical mice, the mouse ball contacts two plastic shafts, one horizontal and one vertical, and an optical encoder measures the rotation of each shaft. As shown in Figure 3, the mouse was anchored to a platform, with two screws as reference points at the left and right ends of the platform. Each length of fishing line ran from the handheld device to a reference point, then into the mouse, around one of the plastic shafts, back out of the mouse, and finally to a small weight that maintained tension on the line.

The x and y movement readings from the mouse were used to track the distance from the handheld device to each reference point and thereby triangulate the position of the device. This method obtained fairly accurate position data while still permitting the device to move freely in space.
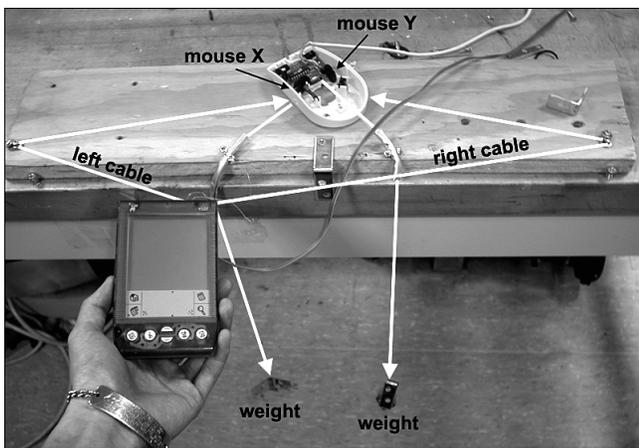

Figure 3. Position tracking with tethers and mouse.

2-D APPLICATIONS

I wrote four simple applications in order to investigate the effectiveness of Peephole interaction techniques.

One-Handed Selector

Selecting an item from a vertically scrolling list is a very common operation in mobile phone user interfaces. This program provides an alternate way to perform that operation. To simulate conditions on a mobile phone, the program is operated using one hand and only about half of the display is used (five items are visible at once). The item nearest the middle of the view is highlighted; the user holds the display in the non-dominant hand and moves it along the list to highlight different items. When the desired item is highlighted, the user selects it by pressing a button with the thumb of the non-dominant hand.

Two-Handed Selector

This program also allows selection from a long scrolling list, but under slightly different conditions, approximating those of a palm-sized computer. The whole display is used, so that ten items are visible at once. Unlike in the one-handed selector, moving the display does not affect the selection. The user selects items by tapping on them with a pen in the dominant hand, while holding and moving the display in the non-dominant hand.
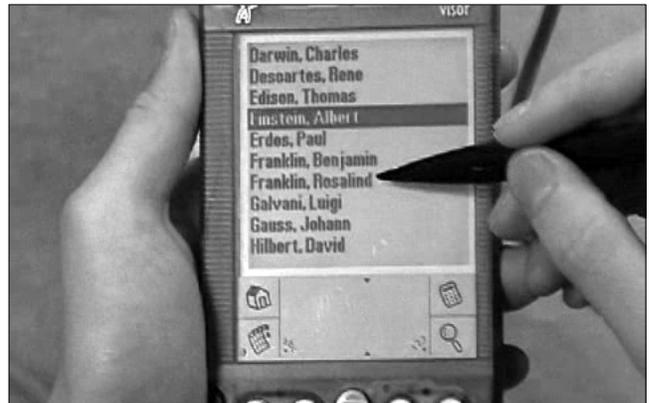

Figure 4. The two-handed list selection program.

Peephole Image Viewer

This program enables the user to view a large image on a handheld computer by physically moving the display around to see different parts of the image. Such an application might be useful on the street for viewing a map, or for reading a large document like a Web page. This is the program depicted in Figure 2.

Peephole Doodle Pad

This is a simple drawing program where the pen is used to draw in digital ink on a large canvas, and the display provides a movable Peephole on the canvas. This allows the user to draw on a potentially unlimited surface using just a small handheld device. At typical handwriting sizes, very little text fits on a palm-size display, so having a large canvas can aid note-taking. Using the Doodle Pad, one can keep writing past the edge of the display without interruption, by bringing the display along while writing.
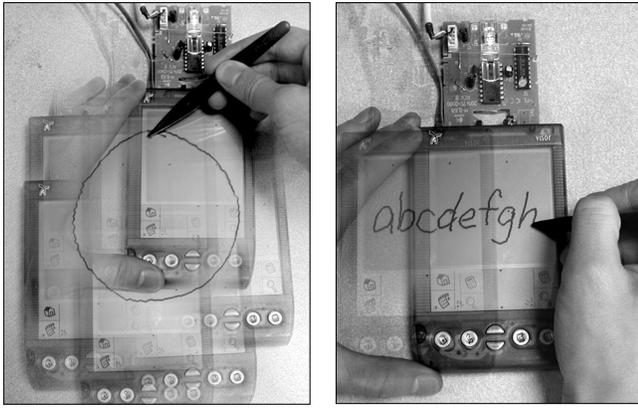
Figure 5. Drawing on a large workspace. (Photos were blended together but not otherwise edited.)

The effectiveness of spatial memory becomes apparent when trying to draw figures bigger than the screen. For example, when using the optical mouse implementation on a table surface, as in Figure 5, it is straightforward to draw a circle larger than the screen in a single continuous, natural stroke – an operation that would be impossible on existing devices. The user simply draws with respect to the table instead of the screen. Kinaesthetic memory and spatial memory make it easy to close the circle accurately. In effect, putting down the handheld computer on a table augments the entire table surface to make it drawable.

2-D USABILITY STUDY
Participants
I conducted a usability study to compare the effectiveness of Peephole techniques with more conventional interfaces for common tasks on mobile computers. The study had 24 participants, all of whom were familiar with handheld computers, though not necessarily owners of such devices. None had previously seen or used a Peephole Display.

Apparatus
Tests were performed using the two-tether implementation in a lab setting. I decided that this implementation was adequate for user testing because it allowed freedom of movement in the air while still providing fairly accurate and fast position data. Naturally, a deployed product would be quite different from this prototype; the goal was to determine the feasibility of the Peephole concept.

Design and Procedure
This study used a within-subjects design. For each of four tasks, a conventional scrolling interface was compared to a Peephole interface. Each participant did all tasks using both interfaces. Participants were given a dummy data set with which to practice using each interface before proceeding with each timed task. Each participant used the Peephole interface first in half the tasks and the conventional interface first in the other tasks. For each task, half the participants used the Peephole interface first and half used the conventional interface first. Two different data sets were used for each task, to reduce learning effects. For each data set, half the participants saw it first and half saw it second; half used it with the Peephole interface and half used it with the conventional interface.

Tasks
The four tasks were designed to test a common operation on handheld devices (menu selection) and two typical uses for mobile computers – map viewing and note taking.

1. **One-handed selection:** Using only the non-dominant hand, find a name in an alphabetized list of 50 names, where 5 names are visible at a time. Repeat this for 10 names, using the same list but prompted for a different name each time. The Peephole one-handed selector was compared with a conventional interface operated by pressing physical "up", "down", and "select" buttons with the thumb.

2. **Two-handed selection:** Using both hands, find a name in an alphabetized list of 50 names, where 10 names are visible at a time. Repeat this for 10 names, using the same list but prompted for a different name each time. The Peephole two-handed selector was compared with a conventional interface operated by pressing physical "page up" and "page down" buttons with the thumb of the non-dominant hand and selecting items with the stylus in the dominant hand.

3. **Map viewing:** Given a fictional subway map, find two stations by name, and then plan a route between them. With the same map, find two more stations by name and plan a second route. The Peephole image viewer was compared with a conventional interface operated by using the pen to drag the image around the screen.

4. **Drawing:** Copy down a simple diagram consisting of labelled boxes and arrows. The Peephole doodle pad was compared with a conventional interface that had a pencil tool and a panning tool, with two small onscreen buttons for switching tools. To ensure that both interfaces provided the same screen area for drawing, the vertical space taken up by the tool buttons in the conventional interface was left unused in the Peephole interface.
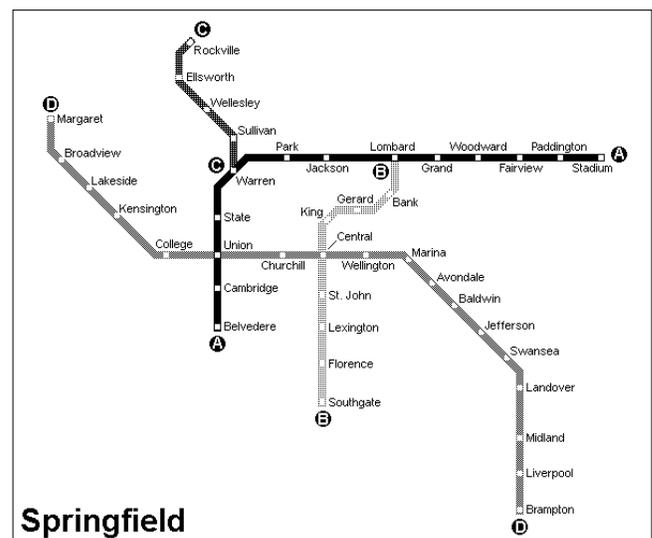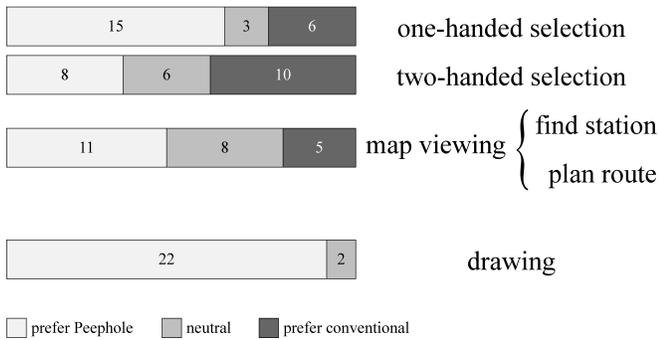

Figure 6. One of the maps used in the study. Maps were 600 by 500 pixels; the screen size was 160 by 160 pixels.

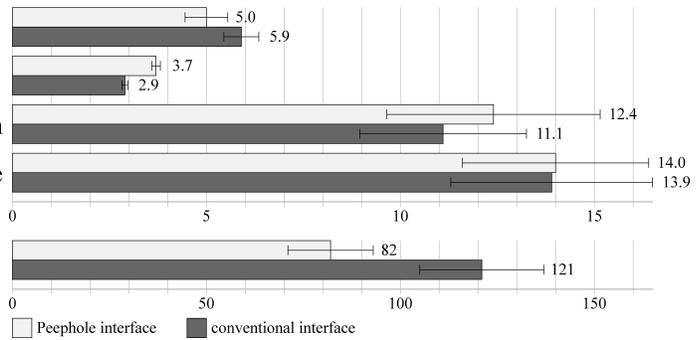Subjective preferences | Task times in seconds

Figure 7. Experimental data from the usability study. Differences in task times were significant (p < 0.01) for one-handed and two-handed selection, significant (p < 0.001) for drawing, and not significant for both map-viewing tasks.

Results

User error rates were negligible ($< 2\%$) for all the tasks, independent of data set or interface. The data confirmed that there was no significant difference between the two data sets for each task (all $|t|$'s < 2.2, all p's > 0.05). Figure 7 presents a summary of the experimental data.

For each task, after trying both interfaces, users were asked which one they preferred. The Peephole interface was preferred for the one-handed selection and map viewing tasks and strongly preferred for the drawing task.

For the one-handed selection task, the Peephole interface was 15% faster ($t(23) = -2.57$, $p < 0.05$); for the two-handed selection task, the conventional interface was 21% faster ($t(23) = 2.94$, $p < 0.01$). For the map-viewing task, there was no significant difference in performance between the two interfaces, for either finding stations or planning routes (both $|t|$'s < 1, both p's > 0.3). Note however that the Peephole interface required only one hand to operate, while the conventional interface required both hands. For the drawing task, Peephole drawing was about 32% faster than the conventional interface, and this difference was highly significant ($t(23) = 8.27$, $p < 10^{-7}$).

Many participants made much smaller drawings with the conventional paint program than with the Peephole paint program. In no case was the drawing produced with the conventional interface ever larger than that produced with the Peephole interface. This suggests that participants felt less space-constrained when using the Peephole interface, even though the actual canvas sizes were the same – only the method of scrolling differed between the two interfaces.

Of the 24 participants, 17 were observed to use both hands together during the drawing task, panning and drawing concurrently to make long pen strokes. All 17 attempted this technique without prompting, which suggests that this type of two-handed interaction is natural and effective.

The most frequent complaint about the Peephole interface was that the display was blurry while it was in motion. In fact, all five participants who preferred the conventional map viewer explained that they preferred it because the blurry text in the Peephole viewer was too hard to read. I believe this factor also accounts for the poor speed of the Peephole interface for two-handed list selection, as it is a

text-intensive task. In the one-handed condition, this deficiency is overwhelmed by the constraints of the conventional one-handed interface: it takes 49 steps to traverse the entire list in the conventional one-handed selector, but only 4 steps (a page at a time) in the two-handed selector.

The Handspring Visor prototype has an LCD that responds quite slowly. Personal experience and these user comments suggested that the Peephole techniques would work much better on a faster and brighter display. It was very encouraging to obtain positive results despite the sub-optimal screen and crude position-tracking hardware.

3-D IMPLEMENTATION

Based on user feedback and my own experiences with the 2-D prototypes, I developed a fourth Peephole prototype. By this time, better hardware was available for both display and tracking: this prototype used a Sony CLIÉ with a 320-by-480-pixel colour screen and an Ascension Bird receiver. This hardware had the advantages of better resolution and contrast, faster screen response, improved tracking precision, and the ability to track positions in 3-D.

3-D APPLICATIONS

I wrote three applications to exploit the improved screen and experiment with ways to use 3-D tracking.

Peephole Zoom Viewer

Perhaps the most obvious mapping for motion along the depth axis is zooming. This is an enhancement of the 2-D image viewer that zooms out when the screen is lifted and zooms in when it is lowered. Each point in 3-D space corresponds to a particular zoom level and panning offset, giving continuous control over both panning and zooming. With a single arcing gesture, the user can smoothly zoom out to see context and dive into a new region of interest.

Peephole Calendar

The standard Palm DateBook application accommodates the small size of the display by offering multiple views at different scales. A toolbar provides buttons for switching between the day view, the week view, and the month view. Only the day view shows the descriptions of appointments and allows them to be edited, but it also gives the least context. When looking for available time to schedule a meeting, for example, the month view can be more useful.
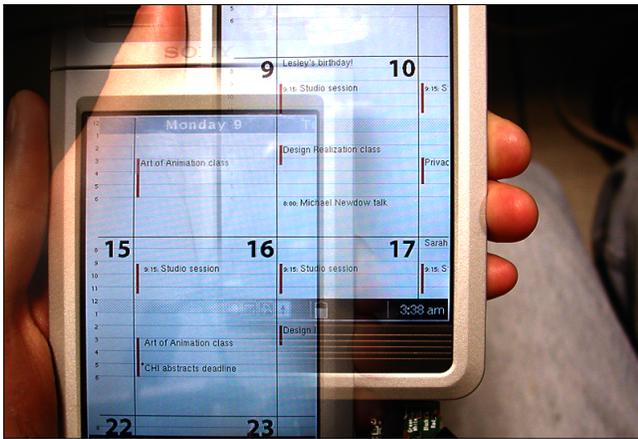
Figure 8. Peephole Calendar month view. (Photos were blended together but not otherwise edited.)

The Peephole Calendar tries to combine the strengths of all three views into a single modeless Peephole view. It reads data from the standard Palm DateBook application and lays out the entire month on the workspace like a page of a wall calendar. The box for each day shows all the appointments for that day, just like the standard full-screen day view. The user can easily scan horizontally to view events in the coming week, scan vertically to look at a particular weekday, or browse through the entire month.

The display has three parts. Most of the screen is occupied by a fully scrolling region that works just like a 2-D Peephole image viewer, except that it also allows direct interaction with the displayed appointments. Along the top of this region is a bar showing the days of the week; this bar scrolls only in response to horizontal movement. Along the left is a column showing the time of day; this bar responds only to vertical movement. These bars, like locked headings in a spreadsheet, help to maintain context when the user is navigating around the workspace.
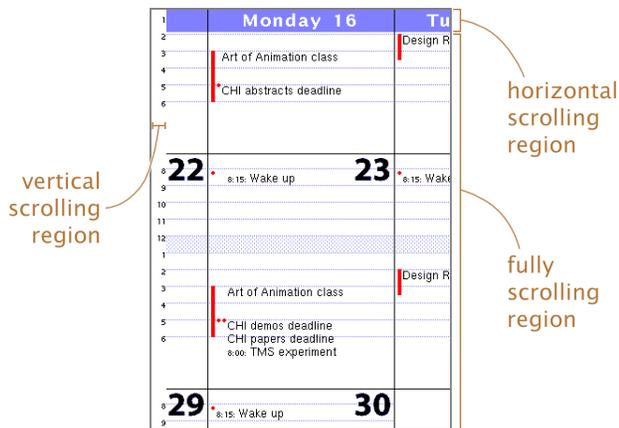

Figure 9. Scrolling behaviour in the month view.

For switching between months, the Peephole Calendar provides a year view. It adopts a model where there are two view planes, one for the overview and one for detail, with the overview (the year view) on the upper plane. The user switches to a different month by lifting the display into the overview plane, moving it to focus on another month, and dropping it back to the detail plane. This is a semantic zoom operation, similar in feel to zooming UIs like Pad [14]. While a desktop ZUI can offer larger visual context, here we have the advantage that all navigation is controlled by the non-dominant hand, leaving the dominant hand free to interact. Dragging an event to a different month is as direct as dragging it to a different day or time. The Peephole Calendar views do not fade or zoom smoothly; a more complete implementation of Pad on a Peephole Display would be an obvious next step.
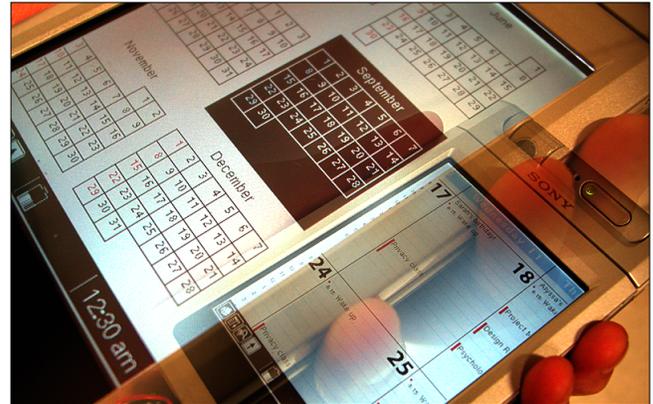

Figure 10. Popping up from the detail plane to the overview plane to navigate to a different month.

Peephole Sketchpad

I wrote a simple object-based drawing program in order to experiment with object manipulation on a 3-D-tracked Peephole Display. A toolbar on the left side of the display lets the user create, select, and move simple shapes. The toolbar is fixed to the display, while the rest of the screen area is a Peephole view on the drawing canvas. As the non-dominant hand moves the view to the region of interest, the tools stay nearby, like a Toolglass would.

Because the Sketchpad responds concurrently to device movement and pen input, the user can easily draw objects larger than the screen and can move objects beyond the edge of the screen in a single operation. In Figure 11, the screen and pen are moved together to drag an object.
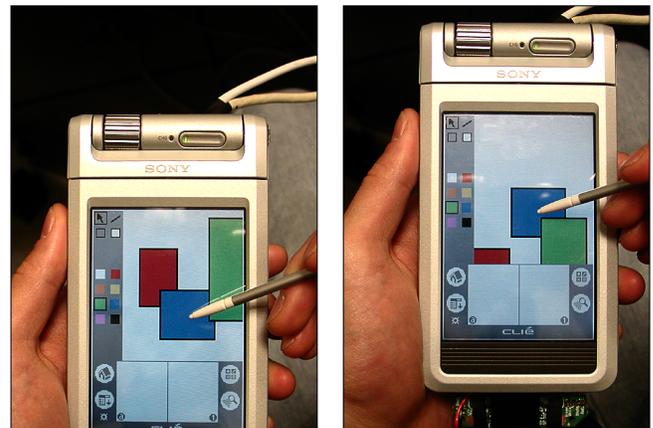

Figure 11. Dragging an object to a new location by moving both hands together. The pen carries along the square while two other objects stay fixed to the canvas.

Figure 12. To copy an object from the canvas to the clipboard, the user holds the pen on the object and lifts it up to the clipboard (left). To paste an object from the clipboard, the user holds the pen on the object and pushes it down onto the canvas (right).
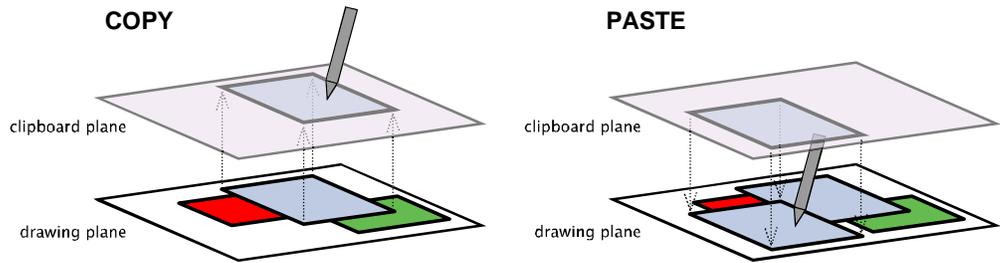
The third dimension is used for situating the clipboard in real space – it resides in a workspace of its own on a plane above the plane of the drawing canvas. The operations for moving objects to and from the clipboard can then be a natural extension of drag-and-drop into 3-D, as shown in Figure 12. To help the user stay oriented, the clipboard and canvas planes have different background colours.

Since the clipboard is a visible workspace, the user does not have to memorize what was placed there. The user can place multiple objects on the clipboard, arrange them as desired, and then group them into a single object for reuse. This works very much like a Toolglass clipboard, though in this case more clipboard space is available, and the original locations of objects can be preserved if desired.

3-D EVALUATION

A formal usability study has not yet been conducted on the 3-D applications, but some users have been informally surveyed. During a recent public poster session, the prototype was left running the Zoom Viewer on the subway map in Figure 6. Twelve curious visitors picked it up and had no trouble panning the view without instructions. One immediately remarked, "It's like there's a map underneath it." Another brought over a friend and explained, "You just move this in a kind of natural way." Seven of the twelve found the zooming feature on their own; the rest understood it as soon as they were prompted to try vertical motion. These observations suggest that the panning and zooming actions are easy to understand. Three users tried two variants of the Zoom Viewer: one that zooms out when lifted, and one that zooms in when lifted. All three preferred to zoom out by lifting the display.

The most common problem users experienced with the Zoom Viewer is that they would sometimes get lost in the workspace. A distraction could cause them to let their hand drift beyond the edge of the map, leaving them with a blank screen and no indication of where to go. This could be addressed by showing arrows pointing back to objects of interest when the display is moved into empty space.

Three users tried the Peephole Sketchpad. After having used the Zoom Viewer, they already knew how to pan around the workspace. All three inferred that they could pan while drawing and dragging objects. One user, after seeing how to copy items by lifting them to the clipboard, immediately guessed that items could be pasted by pushing them down from the clipboard. The others could successfully copy and paste items once 3-D drag-and-drop was described to them. One commented, "This is a great idea."

The most common problem with the Sketchpad was that users found it startling for their canvas to disappear suddenly upon crossing an invisible depth boundary. Switching planes could be made less jarring by blending the two planes together over a gradual transition region.

MULTIPLE-APPLICATION WORKSPACE

To experiment with the concept of personal information spaces, I embedded two applications concurrently into a single virtual workspace: the Calendar and the Doodle Pad. In this prototype, the user wears the tracking equipment so that the applications are embedded in the user's personal reference frame, just in front of the torso with the Calendar on the left and the Doodle Pad on the right. The combined workspace supports linking between applications: for example, the user can draw a map to a party in the Doodle Pad, and then drag the drawing over to the Calendar to record the date and time of the event.

Selecting the event causes the associated drawing to be brought into the Doodle Pad and a big red arrow to appear at the right edge of the display, directing the user's attention over to the Doodle Pad.
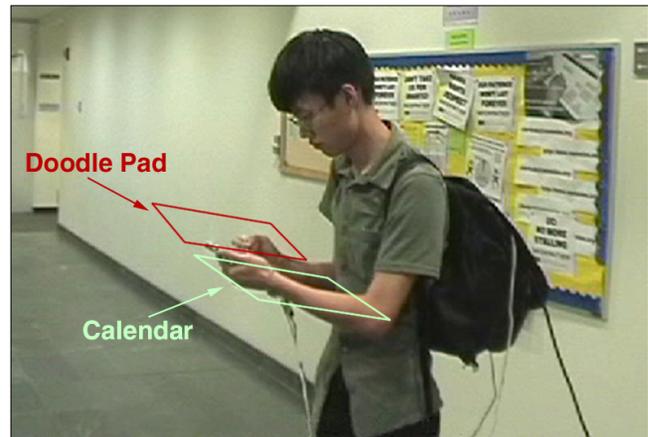

Figure 13. Two applications situated within a personal information space. The apparatus is tethered only by the power cord on the lower right.

DISCUSSION

The fundamental concept here is concurrent navigation and interaction. When the non-dominant hand can take over navigation control, the dominant hand is free to work continuously over boundaries that would previously have forced an interruption. The boundary at the edge of the screen, the structural boundary between months in the year, and the conceptual boundary between the work area and the clipboard are just examples of such boundaries.

Several interaction techniques have been described (1 and 4 are not new, but are listed here to establish a pattern):

1. moving the display in a plane to view a workspace
2. moving the display while drawing on it (draw & pan)
3. moving the display while dragging an item (drag & pan)
4. lifting the display to zoom out, lowering it to zoom in
5. lifting the display to switch from detail to overview
6. lifting the display to switch to a clipboard view
7. lifting or lowering to drag an object to another plane

Any single-button mouse interaction on a desktop computer, as it uses only the dominant hand, can be adapted to an analogous interaction on a 2-D tracked display with a pen in the dominant hand and navigation control in the non-dominant hand. Techniques 1, 2, 3 are instances of such adaptation. Additional mouse buttons can be emulated by providing modifier keys for the non-dominant hand.

A tracked display can also offer some information that a desktop does not: the position of the viewport indicates where the user is looking. The Sketchpad exemplifies how this can be used to keep tools nearby. Tracking in 3-D also yields added input without occupying more hands. The Zoom Viewer, Calendar, and Sketchpad examine various uses for this input – techniques 4, 5, and 6 respectively.

Whereas 1, 2, 3 are "pure desktop" and 4, 5, 6 are "pure Peephole", so to speak, technique 7 (3-D drag and drop) is an example of taking a traditional desktop interaction technique and extending it with Peephole capabilities.

A significant drawback is the loss of peripheral awareness. Peephole interfaces can compensate for this by giving notification of off-screen activity and directional indicators to aid navigation. Using Peepholes is also more fatiguing for the non-dominant arm, so they are probably better suited to short-term interactions (like common PDA tasks).

The Chameleon used its button as a "clutch" to allow the user to move the display while holding onto the workspace. The current Peephole prototypes lack this, and it is evident that a clutch feature is vital for being able to work in a comfortable position. Instead of shifting the entire workspace, however, the button could grab and reposition documents within the user's personal cloud of documents.

CONCLUSION AND FUTURE WORK

This work has combined spatially aware displays with pen input and suggested a family of interaction techniques based on this concept. Two of the techniques have been usability-tested so far, and were shown to be successful in a study of 24 participants. One of the prototypes augments the physical space around a user with an interactive information space. However, the tracking hardware has a long way to go before it is truly robust and portable; inertial tracking is one future possibility.

I believe this work has only scratched the surface of the possibilities for interaction techniques and applications on pen-enabled spatially aware displays. It should be clear that there is a wide range of techniques that can be brought over from the desktop and extended. As for applications, some of the unimplemented ideas include: navigating nested hierarchies of folders and moving items among them with 3-D drag-and-drop; using different planes for comparing versions of a changing document; and a handheld Web browser that lets users save clippings of the current page or open links in alternate planes for later re-use. All of these ideas and more remain to be explored.

REFERENCES

1. R. Balakrishnan, K. Hinckley. The Role of Kinesthetic Reference Frames in Two-Handed Input Performance. In Proc. UIST 1999, p. 171–178.
2. B. Bederson, J. D. Hollan. Pad++: A Zooming Graphical Interface for Exploring Alternative Interface Physics. In Proc. UIST 1994, p. 17–26.
3. E. Bier, M. Stone, K. Pier, W. Buxton, T. De Rose. Toolglass and Magic Lenses: the See-Through Interface. In Proc. SIGGRAPH 1993, p. 73–80.
4. W. Buxton, B. Myers. A study in two-handed input. In Proc. CHI 1986, p. 321–326.
5. G. W. Fitzmaurice. Situated Information Spaces and Spatially Aware Palmtop Computers. Communications of the ACM, vol. 36, no. 7 (July 1993), p. 38–49.
6. G. W. Fitzmaurice, S. Zhai, M. Chignell. Virtual Reality for Palmtop Computers. In ACM TOIS, July 1993, p. 197–218.
7. B. L. Harrison, K. Fishkin, A. Gujar, C. Mochon, R. Want. Squeeze me, hold me, tilt me! An exploration of manipulative user interfaces. In Proc. CHI 1998, p. 17–24.
8. K. Hinckley, R. Pausch, J. C. Goble, N. F. Kassell. Passive Real-World Interface Props for Neurosurgical Visualization. In Proc. CHI 1994, p. 452–458.
9. K. Hinckley, J. Pierce, M. Sinclair, E. Horvitz. Sensing Techniques for Mobile Interaction. In Proc. UIST 2000, p. 91–100.
10. T. Igarashi, K. Hinckley. Speed-Dependent Automatic Zooming for Browsing Large Documents. In Proc. UIST 2000, p. 139–148.
11. H. Ishii, B. Ullmer. Tangible Bits: Towards Seamless Interfaces Between People, Bits, and Atoms. In Proc. CHI 1997, p. 234–241.
12. G. Kurtenbach, G. Fitzmaurice, T. Baudel, B. Buxton. The Design of a GUI Paradigm Based on Tablets, Two-Hands, and Transparency. In Proc. CHI 1997, p. 35–42.
13. W. Mackay, G. Pothier, C. Letondal, K. Bøegh, H. Sørensen. The missing link: augmenting biology laboratory notebooks. In Proc. UIST 2002, p. 41–50.
14. K. Perlin, D. Fox. Pad: An Alternative Approach to the Computer Interface. In Proc. SIGGRAPH 1993, p. 57–64.
15. J. Rekimoto. Tilting Operations for Small Screen Interfaces. In Proc. UIST 1996, p. 167–168.
16. E. Sachs, A. Robers, D. Stoops. 3-Draw: A Tool for Designing 3D Shapes. In IEEE Computer Graphics and Applications, November 1991, p. 18–24.
17. D. Small, H. Ishii. Design of Spatially Aware Graspable Displays. In Extended Abstracts of CHI 1997, p. 367–368.
18. Z. Szalavári, M. Gervautz. The Personal Interaction Panel: a Two-Handed Interface for Augmented Reality. In Proc. EUROGRAPHICS 1997, p. 335–346.
19. M. Tsang, G. Fitzmaurice, G. Kurtenbach, A. Khan, B. Buxton. Boom Chameleon: Simultaneous capture of 3D viewpoint, voice and gesture annotations on a spatially-aware display. In Proc. UIST 2002, p. 111–120.