First publ. in: International Journal of Human-Computer Studies 66 (2008), 8, pp. 605-627

Zoom Interaction Design for Pen-Operated Portable Devices

Thorsten Büring[1], Jens Gerken, Harald Reiterer

HCI group, Department of Computer and Information Science, University of Konstanz, Universitätsstrasse 10, Box D 73, 78457 Konstanz, Germany

Abstract

Maps are currently the most common application domain for ZUIs. Standard techniques for controlling such interfaces on pen-operated devices usually rely on sequential interaction, i.e. the users can either zoom or pan. A more advanced technique is speed-dependent automatic zooming (SDAZ), which combines rate-based panning and zooming into a single operation and thus enables concurrent interaction. Yet another navigation strategy is to allow for concurrent, but separate, zooming and panning. However, due to the limitations of stylus input, this feature requires the pen-operated device to be enhanced with additional input dimensions. We propose one unimanual approach based on pen pressure, and one bimanual approach in which users pan the view with the pen while manipulating the scale by tilting the device. In total, we developed four interfaces (standard, SDAZ, pressure, and tilting) and compared them in a usability study with 32 participants. The results show that SDAZ performed well for both simple speed tasks and more complex navigation scenarios, but that the coupled interaction led to much user frustration. In a preference vote, the participants strongly rejected the interface and stated that they found it difficult and irksome to control. This result enhances previous research, which in most cases found a high user preference for SDAZ, but focused solely on simple speed tasks. In contrast, the pressure and tilt interfaces were much appreciated, which, considering the novelty of these approaches, is highly encouraging. However, in solving the test tasks the participants took hardly any advantage of parallel interaction. For a map view of 600x600 pixels, this resulted in task-completion times comparable to those for the standard interface. For a smaller 300x300 pixels view, the standard interface was actually significantly faster than the two novel techniques. This ratio is also reflected in the preference votes. While for the larger 600x600 pixels view the tilt interface was the most popular, the standard interface was rated highest for the 300x300 pixels view. Hence, on a smaller display, precise interaction may have an increased impact on the interface usability. Overall, we believe that the alternative interaction techniques show great potential for further development. In particular, a redesign should encourage parallel interaction more strongly and also provide improved support for precise navigation.

Key words: Zoom, SDAZ, tilt, pressure, map navigation, semantic zoom, automatic zoom, small-screen devices, PDA

1 Introduction

A major challenge that interface designers for mobile applications often face is known as the *presentation problem* (Spence (2007)), i.e. information spaces must be presented that are considerably larger than the screen. The most common workaround is to display a fraction of the data and then let the users linearly navigate the off-screen space via scrolling. However, for identifying important information in a large data set this approach is tedious, error-prone, and particularly slow. In contrast, the concept of zoomable user interfaces (ZUIs) has been found to improve the user performance for a variety of exploration scenarios. ZUIs facilitate data presentation based on the assumption that navigation in information spaces is best supported by tapping into our natural spatial and geographical ways of thinking (Perlin and Fox (1993)). To implement this approach on a computer screen, data objects must be organized in space and scale. Users can navigate this space by performing zooming (changing the scale) and panning (movement at constant scale) operations. Since ZUIs make more effective use of limited screen real estate, they thus are considered to have a great potential on small screens (Bederson and Hollan (1994); Bederson and Meyer (1998); Khella and Bederson (2004); Shneiderman and Plaisant (2005)). However, previous research on ZUIs usually focused on standard input devices for desktop computers like mice and keyboards, while many current mobile devices such as PDAs and smartphones are operated by a stylus.

In this work we want to investigate the usability of ZUIs on pen-operated devices, and how the user performance and satisfaction may be improved by advanced interaction techniques such as speed-dependent automatic zooming (SDAZ). SDAZ, as proposed by Igarashi and Hinckley (2000), combines zooming and panning into a single operation where the displacement of a pointing device is mapped to the direction and velocity of a continuous pan movement (rate-based scrolling, see Zhai et al. (1997)). At the same time, the faster the users pan the view, the more the system zooms out, and vice versa. SDAZ has been found to improve the performance for basic navigation tasks (e.g. Cockburn and Savage (2003); Savage and Cockburn (2005)), and unlike many

Email addresses: thorsten.buering@ifi.lmu.de (Thorsten Büring), gerken@inf.uni-konstanz.de (Jens Gerken), reiterer@inf.uni-konstanz.de (Harald Reiterer).

[1] Present address: Ludwig-Maximilians-Universität, Institut für Informatik, LFE Medieninformatik, Amalienstrasse 17, 80333 München, Germany.

other interaction techniques developed for desktop computers, SDAZ may be well suited for application on pen-operated mobile devices. Pen input is good for freeform note-taking and sketching, but due to its limited states and events it does not cover the full set of interface primitives (click, track, drag, double-click, and right-click) that most desktop software is built on (Buxton et al. (1985); Buxton (1990); Hinckley (2002)). In contrast, SDAZ only requires a dragging state with two linear input dimensions and thus can be conveniently controlled with a regular stylus. However, there may also be two potential drawbacks related to automatic zooming:

- While SDAZ has been proposed for accelerating map browsing, the tasks tested across studies consisted mostly of rather abstract speed tasks. For more real-life navigation tasks, which require users to constantly read and analyze map information, the binding of zooming and panning may turn out to be rather counter-productive.
- Most studies on SDAZ have been conducted in a desktop environment with a large screen. For smaller screens such as featured by PDAs, a recent work by Jones et al. (2005) indicates that the effectiveness of SDAZ may be decreased. While the experiment did not include a control group, the authors assumed that the unexpected performance results were due to the lower number of contextual cues provided by the smaller display.

Considering map-based ZUI navigation, there are two alternatives to the binding of zooming and panning as featured by SDAZ: the first one is to allow for sequential navigation only, i.e. the users can either zoom or pan, but cannot perform both operations at the same time. This type of interface is very common on pen-operated devices, and its performance strongly depends on the time it takes the users to switch between the navigation styles. For instance, a comparably slow technique is to provide a tool palette, e.g. the hand and magnifying tools in Adobe Acrobat. Such a widget forces the users to frequently move the pen back and forth between the palette and the focus area, which adds up to a significant amount of travel time (Li et al. (2005); Lank et al. (2006)). However, more advanced techniques such as tracking menus proposed by Fitzmaurice et al. (2003) can reduce switching costs to a minimum.

The second option is to support concurrent, but separate, control of zooming and panning. While this type of interface provides the most flexible navigation, it also assumes the most complex input device. Apart from the 2D position tracking of the pen, which we want to use for controlling rate-based panning, an additional input mechanism is needed to manipulate the scale. A straightforward strategy would be to enhance the device with buttons. These can be placed on the pen, on the bezel of the device display, or as virtual buttons on a touchscreen as in Matsushita et al. (2000) and Yee (2004), for instance. However, providing a button always means making an assumption about how the users will hold the device and push the button (Bartlett (2000)). In consequence, there are many situations in which buttons are awkward to press (Zeleznik and Miller (2006); Grossman et al. (2006)) or may even result in an unstable grip of the device (Karlson et al. (2005)). So instead, we want to focus on two interaction techniques that we believe can integrate more seamlessly with the ergonomics of pen-operated mobile devices, namely pen pressure and inertial sensors. With a pressure-sensitive pen, the users can manipulate all three navigation parameters simultaneously with one hand. Inertial sensors such as accelerometers and gyroscopes may be attached to, or incorporated into, the device. Assuming that panning is controlled by the pen, the users can simultaneously zoom by moving or rotating the handheld. Hence, in contrast to pen pressure, this interface would require bimanual interaction.

Based on the different interface options discussed above we implemented a map viewer application and carried out a formal experiment with the following objectives:

(1) Evaluate and compare the usability of (i) automatic zooming, (ii) sequential zooming and panning, and (iii) concurrent but separate zooming and panning for navigating a map on a pen-operated device. For concurrent but separate navigation, the experiment tests one unimanual approach employing pen pressure and a bimanual approach based on both pen and sensor input.
(2) Analyze the effect that different task types of varying navigation complexity would have on the usability results for each interface.
(3) Analyze the effect that different screen sizes (600x600 pixels versus 300x300 pixels) would have on the usability results for each interface.

In the following sections we will first review related work, and then explain the settings and hypotheses of our experiment. Later, we present the results and interpret the findings. Issues for further research are also highlighted.

2 Related Work

This section summarizes previous research in the fields of SDAZ, pressure interfaces, and inertial sensors. It also highlights some design guidelines that we found helpful for implementing the experimental application.

2.1 Speed-Dependent Automatic Zooming

As discussed, SDAZ couples rate-based scrolling with scaling and was developed to avoid visual blur when navigating large information spaces at high speed (Igarashi and Hinckley (2000)). The users control the velocity of a continuous panning operation by dragging the pointing device. To keep the visual flow constant, the system automatically zooms out when the scrolling speed increases and zooms back in when the scrolling speed decreases. This behavior also maps the design intuition that the users desire a high level of detail when moving slowly, while for covering large distances quickly, overview information becomes more important.

Previous research reports a number of experiments investigating the usability of SDAZ on desktop computers. A preliminary informal study by Igarashi and Hinckley (2000) with 7 participants focused on basic 1D and 2D navigation. The first task required the users to locate target images in a long web document using SDAZ and vertical scrolling. For SDAZ, the custom web browser was enhanced with semantic zooming, i.e. with an increasing scrolling speed (and thus a decreasing magnification) the headings of the document became salient to guide navigation. The results showed that the user performance was approximately equal for both navigation techniques. Nevertheless, six out of the seven participants preferred to use the automatic zooming technique. The second task was to navigate to a target point in a flat map view. The latter consisted of an abstract image showing non-representational shapes. The location of the target point was indicated by a small radar window in a corner of the map view. To reach the target, the participants used SDAZ and a conventional zoom and pan interface, where both interaction techniques were controlled via a joystick. Again, the difference in task-completion times between the two interfaces was found to be non-significant. Four participants preferred SDAZ, and three the scrolling interface.

In contrast to the moderate results discussed above, some follow-up studies indicate that SDAZ can dramatically improve user performance. Cockburn and Savage (2003) presented an OpenGL implementation of SDAZ that employed hardware acceleration to provide a smooth animation performance at a high frame-rate. A usability study with 12 participants was conducted to evaluate SDAZ when compared to a variety of manual zooming, panning and scrolling facilities as provided by standard commercial applications (Acrobat Reader and Paint Shop Pro). The test tasks required the users to find images and headings in a PDF document, and to navigate to named locations on a road map. The target locations were cued by compass direction or route descriptions such as highways or rivers to follow. For both document and map navigation, the participants solved the tasks significantly faster using SDAZ. Subjective preferences and workload measures also strongly favored the automatic zooming technique. Similar results were obtained in a study by Cockburn et al. (2005), who compared SDAZ for 1D document navigation to scrolling, rate-based scrolling and a variant of automatic zooming that implemented optimal pan-zoom trajectories based on the work of van Wijk and Nuij (2003). SDAZ was found to significantly improve the user performance and preference values.

In some cases, the powerful coupling of navigation in SDAZ can also be considered a potential drawback. For instance, the users are not able to zoom out without increasing the pan speed, or to maintain a constant scale without moving. With regard to this limitation, Savage and Cockburn (2005) conducted an experiment in which SDAZ was compared to an interface that allowed concurrent but separate navigation control: the users could drag the mouse to pan the view via rate-based scrolling, and press two keyboard keys for stepwise zooming in and out with 5 discrete scale levels. Apart from browsing documents and maps, a globe ZUI was also tested. During the evaluation, 35 participants were asked to navigate to a target object continuously cued by an arrow positioned in the window center. It was found that automatic zooming significantly reduced task-completion times, that it was preferred by the users, and that it caused less subjective workload. However, this finding does not correspond to the results of a study by Appert and Fekete (2006), in which the authors proposed a 1D multiscale navigation technique called OrthoZoom Scroll. With OrthoZoom the users can control rate-based scrolling by moving the mouse cursor along the Y dimension, while the granularity of the information space is manipulated by dragging the cursor orthogonally. Hence this approach allows concurrent but separate view navigation. A controlled experiment with 12 participants was conducted to compare OrthoZoom to SDAZ for pointing tasks with indices of difficulty up to 30 bits (see Fitts' law, Fitts (1954)). During navigation, the target was continuously cued by arrows and concentric circles. The researchers found that OrthoZoom performed twice as fast as SDAZ and that all participants preferred to use the OrthoZoom technique.

In previous research there have also been attempts to examine the effectiveness of automatic zooming on small screens. Patel et al. (2004) applied a variant of SDAZ to vertically navigate a photo collection in a small window on a desktop PC. Unlike the original design, the reference point for scale and velocity was not the initial mouse-down position but the center of the screen. In an evaluation with 72 participants, the technique was compared to a scrollable thumbnail view and an interface called GestureZoom, which maps vertical dragging of the mouse to rate-based scrolling while horizontal dragging controls the zoom level of the view. Across different search tasks the performance of SDAZ and GestureZoom for searching images turned out to be at least equal to, or better than, that for the scrolling interface. Less promising results were found in a study reported by Jones et al. (2005), in which 12 participants used SDAZ on a simulated PDA screen to search documents and city maps. Again, all test applications were operated by a mouse. Surprisingly, even though the control interface was limited to sequential mode-based navigation, SDAZ produced longer task-completion times. In the case of the 1D document navigation, the difference was significant. The authors assumed that the discrepancy of the results compared to the success of SDAZ in desktop-based studies was due to the smaller view size. However, the study did not comprise a control group for screen size and thus more research must be conducted to clarify this point.

2.2 Pressure Sensing

When talking about pen-operated devices, we mean devices that are controlled by a touchscreen or an electromagnetic digitizer located behind a LCD.[2] Touchscreens are usually found in PDAs and an increasing number of smartphones; they can sense any object (stylus or finger) touching the surface. The electromagnetic digitizer is commonly used in Tablet PCs and graphics tablets such as a Wacom display[3]. While they can only be operated with a special pen, many digitizers provide advanced sensing features including measuring the pressure of the pen tip applied to the display.

One of the earliest works on controlling user interfaces with pressure was done by Herot and Weinzapfel (1978) almost thirty years ago. The device setup consisted of a glass pane with piezoelectric transducers, with the pane being mounted on a display tube with strain gauges. The researchers conducted an informal experiment to investigate the users' ability to perform basic interface interaction by applying pressure to the screen. The tasks were to manipulate a cursor's position and speed, as well as to push, pull, disperse and rotate virtual objects. The authors found that pressure sensing provided a rich channel for direct manipulation and multidimensional interaction.

Pressure input can be used to control a continuous parameter or to produce discrete signals. A discrete design may be used, for instance, to trigger click events while pointing. Buxton et al. (1985) found that just two levels of pressure already provide a stylus with a set of states comparable to a one-button mouse, and that the input can thus be used to control direct manipulation interfaces. Another example of a discrete pressure interface is the pen-operated 2.5D virtual desktop that was presented in Agarawala and Balakrishnan (2006). The users applied maximum pen pressure to switch from a dragging mode to manipulating desktop objects (e.g. a pile of files).

The most common application for mapping pressure to a continuous parameter is to control the width or color opacity of a brush tool in a drawing program. This functionality, which was originally suggested by Buxton et al. (1985), is supported by Adobe Photoshop and Illustrator, for instance. A more advanced scenario for employing continuous pressure is the TwistLens slider widget presented by Ramos and Balakrishnan (2003). In the example, the slider is used to navigate the frames of a video sequence. Frames in focus are enlarged via

[2] Another pen technology is the light pen, but since it only works with CRT monitors it is hardly used any longer.

[3] http://wacom.com

a fisheye zoom, while the users can morph the linear layout via pressure to an s-shape to avoid occlusions among thumbnails. The amplitude of the lens grows with an increasing force.

Surprisingly, to date there have been only a few evaluations of pressure input. Nacenta et al. (2005) present an interface, in which pen pressure was used to define a throwing distance for a virtual object in a multi-display reaching task. In an experiment with 18 participants, the authors found that the technique was inferior in both performance and user preference to several non-pressure techniques (e.g. *pick-and-drop* Rekimoto (1997) and *push-and-throw* Hascoët (2003)). They concluded that one potential reason for the result was the poor control of pressure with the pen. They also acknowledged the need to study pressure-based input devices more carefully to make them easier to control.

Li et al. (2005) investigated pressure input for rapid mode switching between ink and command gestures on a Tablet PC. The pressure space was divided into two levels, where pen strokes with light to medium pressure were interpreted as ink, and strokes with strong pressure were treated as commands. A significant problem detected during a usability evaluation was that some participants showed difficulties in adapting to the uniform pressure threshold. The threshold had been selected based on a preliminary evaluation, but could not compensate for the high variance of individual differences in applying pressure.

An experimental application whose design requirements were quite similar to the ones for the present work has been introduced by Ramos and Balakrishnan (2005). A pressure-sensitive stylus was used as an integrated device for zooming a parameter's scale space while concurrently manipulating the parameter's value within that scale space. The authors conducted a usability study, in which 12 participants were asked to vertically scroll and zoom an abstract document to reach targets of different widths and located at varying distances from the viewport. The pressure-based widget was compared to two bimanual interaction techniques, where the position sensing of the pen input was used in conjunction with either an isometric button or two discrete keyboard keys controlled by the non-dominant hand. The results showed no significant difference in task-completion times between the techniques. Preference ratings were also mixed, but participants commented that 'it felt right' and 'natural' to zoom using a pressure-sensitive pen as an integrated device. In contrast, the participants who favored one of the bimanual techniques liked the fact that zooming and scrolling were decoupled by using two hands. However, other users found it demanding to coordinate the two-handed navigation.

Overall, previous research seems to indicate that pressure interfaces, though promising, are difficult to implement in such a way that they can be controlled accurately and with ease. The following list summarizes some common implementation and usability issues:

- *Lifting the pen*
  Depending on the speed with which the pen is lifted from the display, the pressure decreases gradually, which may trigger unwanted operations (Raisamo (1999)). In the case of multidimensional input, the lifting movement can also cause an unintentional fluctuation of the pen's spatial position just before it loses contact with the device (Buxton et al. (1985); Ramos and Balakrishnan (2005)). Appropriate filtering of the signals should eliminate or at least reduce both problems.
- *Number of pressure levels*
  The human ability to reliably discriminate discrete levels within the pressure space is rather limited. The proposed maximum numbers of levels are 6 (Ramos et al. (2004)), between 5 and 7 (Mizobuchi et al. (2005b)), and between 3 and 7 (Buxton et al. (1985)).
- *Friction*
  For finger input, an increasing pressure force also increases the friction between the finger and the display, which can hamper smooth, sweeping gestures (Herot and Weinzapfel (1978); Buxton et al. (1985); Raisamo (1999)). To a less significant degree this problem also applies to pen input.
- *Maintaining a pressure level*
  Moving a pen decreases the users' ability to maintain a constant pressure level, which in the case of multidimensional input can hamper precise parameter manipulation. In such cases a filter mechanism for signal stabilization is recommended (Ramos and Balakrishnan (2005)).
- *Low-pressure spectrum*
  Users have less pressure control at low levels of pressure (Ramos et al. (2004)). The transfer function in a continuous mapping, or the pressure thresholds in a discrete mapping should outweigh this effect.
- *Individual differences*
  Users have been reported to vary a lot in the range and values of their pressure spaces (Li et al. (2005)), i.e. some people use a lower or greater overall pressure for drawing with a pen. Calibration was suggested to adjust the pressure sensitivity of the interface to the individual pressure space.

Another important factor for easing the control of pressure input is the provision of continuous visual feedback (Buxton et al. (1985)). With respect to this, Ramos et al. (2004) carried out a study comparing continuous feedback with more limited feedback features, which required participants to rely primarily on their haptic memory. Results suggest that limited feedback can significantly decrease user performance.

The design of appropriate feedback is strongly dependent on the task to be performed and the type of pressure mapping used. If applicable, however, the feedback should not only indicate the current amount of pressure applied, but also indicate the consequences of varying the pressure level (Mizobuchi et al. (2005a)). An example of such an approach was proposed by Ramos and Balakrishnan (2003) for the previously discussed TwistLens slider. The slider provides continuous visual feedback with preview icons reflecting the consequences of varying the pen pressure with respect to the s-shape of the video sequence. Apart from visual feedback, auditory feedback may also be beneficial in some scenarios, e.g. Ramos and Balakrishnan (2005).

2.3 Sensor-Augmented Mobile Devices

Sensor-based interaction has become a continually growing area of research during the last ten years. An example of the rich input potential provided by this approach is the video console Wii[4]. The users play games using a wireless controller that, as well as serving as an infrared pointing device, also detects motion and rotation in three dimensions. Since its release in late 2006 the console has been a great success, with millions of units sold worldwide. Other devices that provide tilt-control are recent mobile phones such as the Nokia 5500 Sport[5] and the Samsung SCH-S310[6].

An early research prototype investigating sensor-based interaction was the Chameleon system (Fitzmaurice et al. (1993); Fitzmaurice (1993)). The main idea was that a small palmtop should act as a viewport to a large virtual workspace. By moving the display in 3D, the users could navigate the workspace accordingly. Chameleon consisted of a 4-inch color monitor simulating a future palmtop device, and a back-end workstation for doing all the processing and graphics rendering. The monitor was equipped with a 6-degree-of-freedom input device, whose signals were used by the workstation to detect user gestures in terms of position and tilt. The visual output produced by the workstation was fed back to the palmtop via a video camera. Ten years later a more advanced prototype of a spatially aware display was presented by Yee (2003). Based on a peephole metaphor, the system allowed users to navigate lists and maps, draw images larger than the screen, or navigate a calendar application via semantic zooming. Unlike Chameleon, the peephole display was intended for bimanual usage combining the position-tracking of the device with pen input.

Though not implemented, the developers of Chameleon had also discussed how their interaction model could benefit from tilting gestures. Apart from triggering automatic view shifts in 3D navigation, they saw a great potential

[4] http://wii.nintendo.de/
[5] http://www.nokia.com
[6] http://www.samsung.com

for this type of interaction in controlling 2D rate-based scrolling (Fitzmaurice et al. (1993)). Since then many experimental systems have investigated tilt-control, with scrolling probably being the most common application domain (e.g. Harrison et al. (1998), Small and Ishii (1997)). A way to control both zooming and panning by tilting alone was examined by Eslambolchilar and Murray-Smith (2004). To navigate a text document on a PDA, the authors coupled tilt input around the X axis with speed-dependent automatic zooming. The results of an informal usability study suggested that the handling of tilt-based SDAZ was comparable to that of a pen-based SDAZ implementation.

Rekimoto (1996) presented a palmtop display that employed tilting around the vertical and horizontal axes as continuous input to navigate cylindrical and pie menus, and to browse a 3D map. In the case of the map, when dipping the device edge to one side, the view was panned, rotated, and zoomed to provide a perspective overview of the current location. As another example application, the author had implemented a 3D object viewer, in which the orientation of the view was coupled to the orientation of the display. Thus users could inspect the virtual object from different directions by rotating the device.

An advanced prototype of a small-sized appliance with built-in sensors was presented by Bartlett (2000). Users could control the device by performing tilting and fanning gestures in different directions and around the two axes of the device. During an informal study with 15 participants, tilt-control was used for various navigation tasks. On the one hand, it was found that the majority of participants rapidly learned to master the interaction. On the other hand, when comparing the technique to a four-way rocker switch button, the preferences of the users were mixed. The participants who favored tilt-control found this technique more natural and intuitive to use. The other group preferred the button because of the greater precision it provided.

Using tilt-control for navigating a spreadsheet was proposed by Hinckley et al. (2000) and Hinckley et al. (2005). The researchers experimented with different transfer functions for controlling 2D rate-based scrolling via tilt angles, and found single axis control to be the most promising, i.e. scrolling was limited to either vertical or horizontal scrolling at a time. Though this approach hampers diagonal movements, it allows for rapid scrolling across a long distance without drifting off-axis. Another result was that the 5 participants in an informal usability test seemed to like tilt-to-scroll better than using a rocker switch button. In particular, they preferred tilting for its intuitiveness, and because it was easier to control with one hand. Negative user statements were that it was easy to overshoot a target with tilting, and that it took time to get used to it. Similar results were gained by Cho et al. (2007), who compared tilt-to-scroll with using navigation buttons and an iPod wheel for searching images in a photo collection. Most effective with regard to overshooting, travel distance, and performance time was the button-based input. Nevertheless, the 7 participants slightly preferred tilting to the other techniques, though they also rated it more difficult to control.

The previous literature indicates that sensor-based interaction, and especially tilting, may provide an intuitive input dimension for controlling handheld devices. Unlike position-tracking in 3D, tilting requires less physical activity and less space, which makes gesturing more comfortable. As with pen pressure, there have been only very few quantitative studies on tilting, but problems have been reported with regard to input precision. Furthermore, the following issues should be considered when implementing tilt-control:

- *Physical limitations of tilting input*
  The users' physical ability to tilt around the two device axes may not be equal for each direction. For instance, the results of an evaluation by Crossan and Murray-Smith (2004) examining PDA tilt-control in a target-acquisition task indicated a lower performance and accuracy for upward pitch motions. Mantyjarvi et al. (2006), who used discrete tilt events to navigate a virtual museum guide on a PDA, reported that the majority of the participants in an informal user test found tilting around the X axis of the device more difficult than tilting around the vertical axis.
- *Toggle tilt-control*
  Tilt-sensitive devices must provide a functionality to disable and engage sensor input, otherwise accidental gestures can be difficult to avoid. One solution is to provide a simple toggle mechanism such as pressing a button (Mantyjarvi et al. (2006)), squeezing the device (Harrison et al. (1998)), or performing a tilting gesture (Bartlett (2000)). Even then, the users may simply forget to turn the sensitivity off. Hence other designs propose a clutch button that must be kept depressed during navigation (e.g. Fitzmaurice et al. (1993); Rekimoto (1996); Small and Ishii (1997)). Obviously, this method may become tedious if the navigation continues over a longer time span. An alternative technique that does not require muscle tension is to use a touch sensor on the bezel of the display (Hinckley et al. (2000)). The tilt-sensitivity is enabled until the users release contact. However, as acknowledged by the authors, inadvertent contact can be a problem.
- *Neutral angle*
  A touch sensor (Hinckley et al. (2000)) or a gesture (Bartlett (2000)) may also provide a flexible way to set a device's neutral orientation, relative to which tilt interaction takes place. In contrast, a pre-defined angle as proposed by Harrison et al. (1998), though derived from in-laboratory testing, may not be appropriate for different user postures.
- *Display visibility*
  Several researchers have experienced difficulties with display visibility due to reflections and extreme viewing angles, e.g. (Harrison et al. (1998); Eslambolchilar and Murray-Smith (2004); Crossan and Murray-Smith (2004);

Bartlett (2000)). Hinckley et al. (2000) proposed to compensate for the loss of apparent contrast during tilting by continuously adjusting the contrast settings of the device. However, as noted by Cho et al. (2007), the introduction of wide viewing angle LCD and OLED technology may significantly reduce the problem in the near future.

3 User Evaluation

Based on SDAZ, pen pressure, and tilt-control, we implemented three interaction interfaces for controlling a ZUI on pen-operated mobile devices. A standard interface relying on sequential interaction was also developed. We conducted a formal usability study, in which we compared the approaches in terms of task-completion times, user preference, and workload. The study required the participants to solve navigation and search tasks by zooming and panning a Munich city map. The map consisted of three semantic layers and was displayed in a viewport of either 600x600 pixels or 300x300 pixels. The following sections discuss the experimental settings and our hypotheses.

3.1 Experimental Interfaces

In a first attempt we implemented our experimental map viewer application using C# and the .NET framework. However, for the large map bitmaps used, the graphics performance was unsatisfactory. Thus, we reimplemented the application using managed DirectX 9.0, which allowed us to take advantage of hardware acceleration. Since the map viewer is essentially a 2.5D interface, we simply wrapped the map bitmap in a sprite object, which is translated and scaled during navigation according to the user input. This approach provided us with smooth zooming and panning at 40 to 50 frames per second. We were also able to reuse some of the code from the previous prototype.

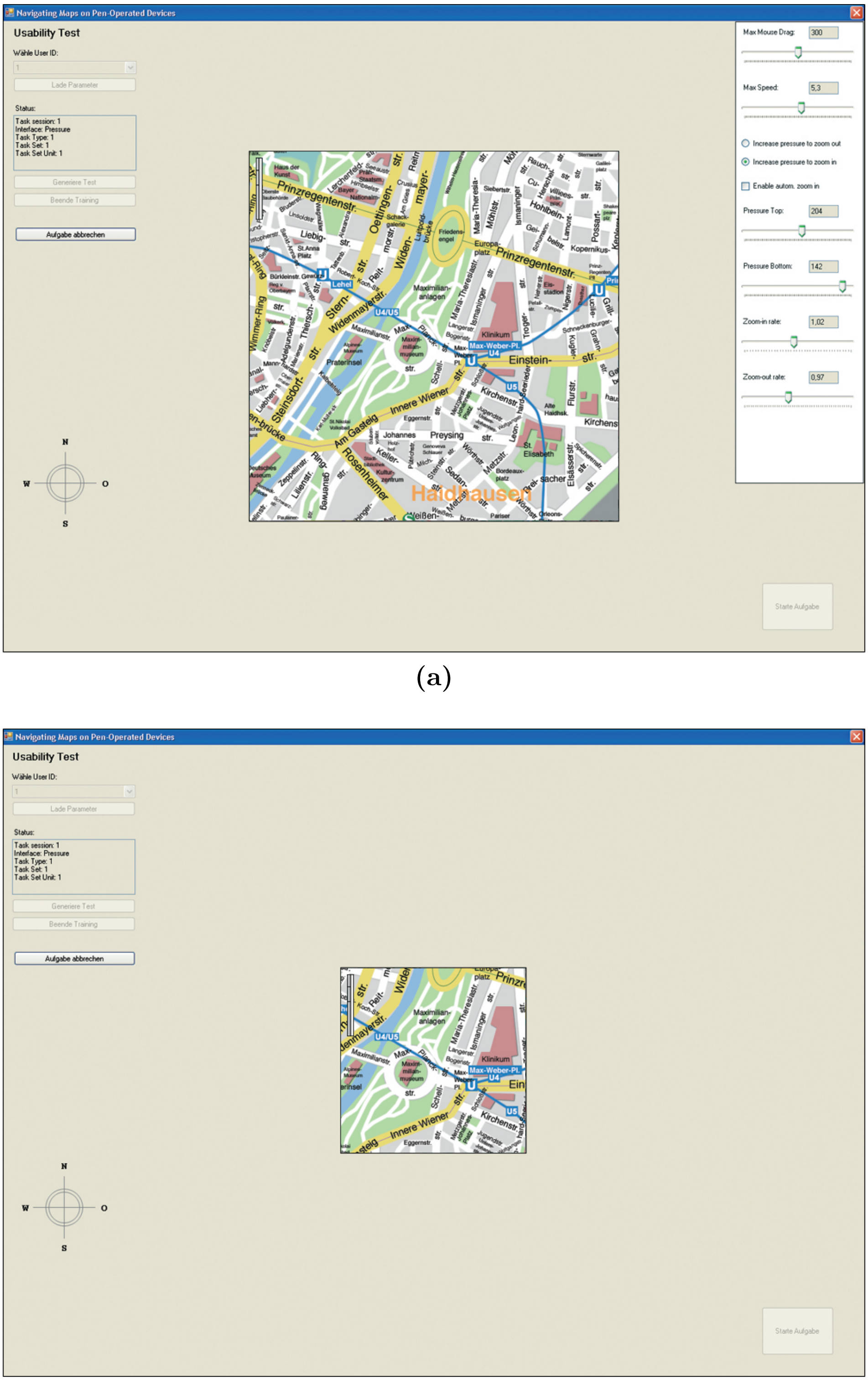
Each of the different interaction techniques is implemented as custom event handlers. Apart from zooming and panning, all interfaces also support a click mechanism by quickly tapping the screen with the stylus. Except for the standard interface, panning is implemented as rate-control, i.e. the speed of an animated and continuous pan movement increases with the distance the pen is dragged on the display. For rate-control, we defined a maximum pan speed along with a maximum drag distance at which the maximum speed would be reached. The drag range was set to be a fraction of the smaller dimension of the view size because the users quickly reach the end of the display when dragging a pen on a PDA or a smartphone screen. Being stuck at the bezel, the pen is significantly impaired in the way it can be moved. By setting the maximum drag distance to a fraction of the view size, the users can reach the maximum speed and, without decreasing the speed of the movement, still have sufficient space left before reaching the bezel to freely change the pan direction.

Figure 1 shows the general layout of the map-viewer application based on the example of the pressure interface. The GUI consists of controls for the test administrator, a compass rose, and the viewport to the map information space. A cross-hair is drawn in the center of the map view. Visual feedback, apart from the implicit map transformation, is provided by each event handler individually. Furthermore, the rate-controlled interfaces provide the users with a visual reference point of the pen displacement. Dragging the pen over the display causes the system to draw an arrow with its origin in the center of the view. The direction of the arrow is equal to the direction of the pen movement and the length mirrors the current drag distance with respect to the initial pen position. While interacting, the arrow is continuously updated. For measuring the effect of display size, the map view could be set to either 600x600 pixels or 300x300 pixels. The smaller view corresponds roughly to the display size of a PDA-like device.

The parameters used for the interaction techniques (e.g. maximum pan speed, zoom rate, etc.) were determined in a preliminary test with 6 participants. For this purpose, the variables were mapped to interface controls so they could be dynamically adjusted during testing. Such a control panel is shown in Figure 1a on the right hand side of the screenshot. This panel was only visible in the preliminary test.

3.1.1 SDAZ Interface

The SDAZ interface maps both rate-based 2D scrolling and zooming to the pen position. For the relationship between pan speed and scale, the prototype implements the original design as proposed by Igarashi and Hinckley (2000). In the preliminary user test, however, the smooth decrease in scale proved to be a problem for more complex navigation tasks. For instance, given that the view was set to an area of interest, the users wanted to gain an overview of the region at a low magnification level, but when zooming out the implicit panning caused the initial location to quickly move into the offscreen space. Thus the users frequently had to backtrack to the start position, which they found rather annoying and confusing. To minimize this problem, we manipulated the rate of change of the scale to be more significant at first, leading to a lower magnification level when panning slowly, i.e. in Igarashi and Hinckley (2000) the scale is calculated based on the equation: $scale = s0^{D_{norm}}$, where $s0$ is the minimum scale, D_{norm} is the normalized displacement of the pointing device with respect to the minimum device movement (before that no zoom-



Navigating Maps on Pen-Operated Devices

Usability Test

Wähle User ID: 1

Status:

Task session: 1
Interface: Pressure
Task Type: 1
Task Set: 1
Task Set Unit: 1

Max Mouse Drag: 300

Max Speed: 5,3

○ Increase pressure to zoom out

● Increase pressure to zoom in

□ Enable autom. zoom in

Pressure Top: 204

Pressure Bottom: 142

Zoom-in rate: 1,02

Zoom-out rate: 0,97

(a)

(b)

Fig. 1. Layout of the experimental map viewer application: (a) the 600x600 pixels view (with the manipulation panel used in the preliminary test), (b) the 300x300 pixels view.

ing occurs) and the maximum device movement. In our implementation we manipulate D_{norm} (lies between 0 and 1) by raising it to the power of a manipulation factor set to 0.8. In a second step, we then calculate the pan speed based on the scale, as described by Igarashi and Hinckley (2000).

As recommended by Igarashi & Hinckley, we defined a maximum falling rate to ease the transition between zoomed-out and zoomed-in views. Otherwise, when lifting the pen or reversing the drag direction, the map would abruptly zoom in to full size because the pan rate defined by the pen displacement drops to zero or crosses zero. This has a very disorienting effect, in particular when dealing with semantic zooming. Given the maximum falling rate the interface takes 1.5s to zoom from the minimum magnification level of 12.5% to full scale. The minimum and maximum pen displacement thresholds (between which the scale manipulation takes place) are set to 39% and 95% of the side length of the view size. The maximum pan speed is 330 pixel/s.

To provide the users with a notion of the zoom level, a scale bar is drawn within an outlined box in the upper-left corner of the view. The size of the white bar reflects the current zoom level proportional to the scale range and is continuously updated during navigation (see Figure 2). When the box is all white, the map is shown at the highest magnification level.

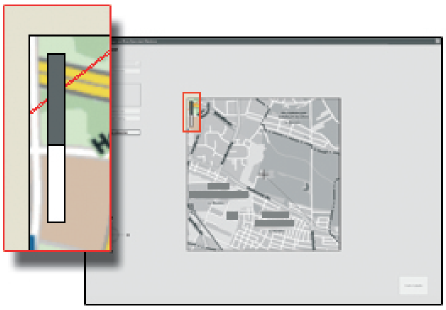


Fig. 2. Scale bar feedback for SDAZ. The white bar grows with an increasing magnification level.

3.1.2 Pressure Interface

With the pressure interface, the users pan via rate-control and manipulate the scale by adjusting the pressure of the pen on the display. The test device used in the evaluation provides a pressure space from 0 to 255. Considering that the users were expected to zoom and pan in parallel, the application required a robust mapping that would resist significant but unintentional fluctuations of the pen pressure. Hence, we avoided a continuous mapping and defined two thresholds that would tile the pressure space into three discrete intervals. Each interval corresponds to a zoom command: when the input value lies within one of the outer intervals, the scale is continuously increased or decreased by a constant factor; when within the range of the middle interval, the current scale is maintained. Such an all-or-none response for parallel zooming and panning has already been found beneficial by Bourgeois and Guiard (2002). Moreover, the design of only three command intervals is a good match for the limited human ability to discriminate a larger number of pressure levels.

Which outer interval to map to which zoom direction proved to be a difficult decision. When developing the interface, the first design intuition was that the map should zoom out when applying strong pressure (the input value enters the top interval). This corresponds to pushing a physical map away from the body. Accordingly, applying only little pressure would cause the map to rise, i.e. to zoom in. We found, however, that the users in the preliminary test expected the opposite mapping, and when trying out this opposite mapping they also found it more natural to control. They associated the high pressure with diving deeper into the information space and thus with magnifying the map. We also experimented with different pressure thresholds and eventually settled on the following values:

- *Zoom in with constant speed:* [205,255]
- *Maintain the current scale:* [142,204]
- *Zoom out with constant speed:* [0,141]

These thresholds agree with the previous literature reviewed in that our users showed more difficulties in controlling the low-pressure spectrum. This resulted in a zoom-out interval that is almost three times the size of the zoom-in interval in the high-pressure spectrum. The remaining interface parameters are set as follows: the pen displacement threshold for reaching the maximum panning speed (330 pixel/s) is 50% of the side length of the view size, zooming in from the minimum magnification level (12.5%) to full scale takes 2.45s, and zooming out from maximum magnification to the minimum magnification level takes 1.5s.

It is important to note that, compared to SDAZ, the pressure interface does not force the users to zoom and pan in parallel. If desired, they can control each navigation parameter independently. Another feature is that when the users lift the pen from the display the current view freezes. This allows examination of a portion of the map in a more relaxed state. With SDAZ, the users must permanently circle over an area of interest to avoid losing the current scale and position of the view.

An implementation issue we experienced with freezing the view was that, when lifting the pen, the pressure input inevitably crosses the lowest pressure spectrum. When the users had previously zoomed in or maintained the scale, this then caused a slight but annoying drop in scale. A similar problem occurred when initially pressing the pen on the display to either pan at the current scale or to zoom in. The input value would gradually rise from 0 to above 141 and thus temporarily decrease the scale. To overcome these effects we introduced a timer that is set each time the pressure input enters the lowest pressure interval. The corresponding zoom-out operation is not triggered until the timer executes after 100 milliseconds. If the pressure value exceeds the low-pressure interval within that timeframe, the operation is cancelled. The delay of 100 milliseconds is too short to be noticed by the users, but sufficient to reliably filter out the pressure fluctuation when lifting or setting the pen.

The visual feedback of the pressure interface as shown in Figure 3 enhances the scale feedback of SDAZ by an additional box in which the horizontal lines mirror the two pressure thresholds proportional to the pressure space. Within the box outline, the current pressure value is denoted as a red dot that rises with a growing pressure. The interval that currently contains the pressure dot is further highlighted by a blue background color.

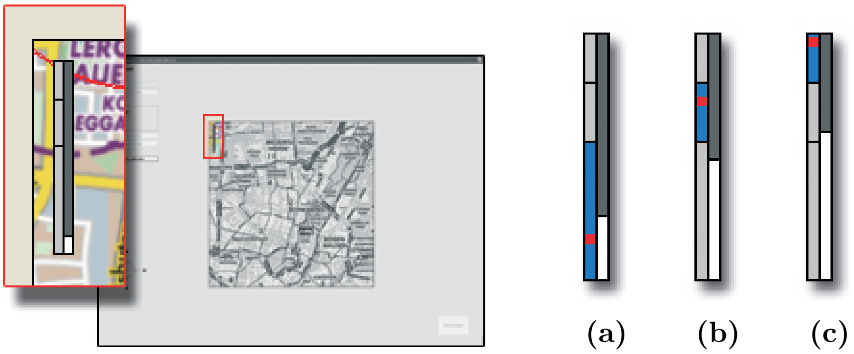


Fig. 3. Visual feedback for the pressure interface, consisting of a pressure bar and a scale bar. The tiles of the pressure bar mirror the three pressure intervals, and the red dot represents the current pressure input: (a) low pressure interval, the interface zooms out, (b) middle pressure interval, the current scale is maintained, (c) high pressure interval, the interface zooms in. The white scale bar indicates the current zoom level.

3.1.3 Tilt Interface

The tilting interface offers the same functionality as the pressure interface but it is controlled by two-handed interaction based on Guiard's model of the Kinematic Chain (Guiard (1987)). According to the model, in real-life human action the non-dominant hand sets a frame of reference in which the dominant hand operates. Moreover, the action of the non-dominant hand precedes that of the dominant hand and the level of detail at which the dominant hand works is finer than that of the non-dominant hand. We applied this design to the map viewer application by assigning the comparably coarse task of controlling the scale (the frame of reference) to the non-dominant hand via device tilting.

Accordingly, the fine granular task of rate-based panning (view pointing) is left to the dominant hand via pen input.

For controlling the zoom, only a single tilt dimension is needed. We limited the interaction to rotation around the X axis of the device, because tilting the device around the Y axis with the display pointing away from the dominant hand makes it ergonomically difficult to operate the device using the pen. In contrast, tilting around the X axis ensures a relatively constant level of ease-of-access for pen input.

The tilt-control is based on the data from the accelerometer of the test device. For mapping the input to the scale of the view, we used the same threshold strategy as for the pressure interface. However, an important difference is that the thresholds are not defined as absolute values, but as angular distances relative to a neutral orientation of the device. The neutral orientation is set to the current tilt input value each time the users put the pen on the display. Hence the neutral orientation is frequently and automatically initialized without requiring any extra action by the users.

Based on the preliminary study, we set the relative thresholds to be 6 degrees from the neutral orientation when tilting the device upwards (rotating the upper edge of the device away from the body) and 2 degrees from the neutral orientation when tilting it downwards. The difference in the angular distances is due to the special settings of the test device, in which downward tilting decreases the visibility of the display more significantly than upward tilting. Given the two thresholds, the users have an angular range of 8 degrees in which the current scale is maintained. When the tilt angle exceeds the range, the zoom operation corresponding to the respective tilt direction is triggered. Apart from being easy to control, the threshold design with all-or-none response also helps to avoid extreme device angles. The values for the remaining parameters (drag distance threshold, maximum panning speed and zoom speed) are the same as for the pressure interface.

When assigning the zoom commands to the two tilt directions, we again faced the problem that there seems to be no definite natural mapping. However, most participants of the preliminary test preferred to zoom in when tilting upwards, and to reduce the magnification when tilting the device downwards. Zooming is at constant speed.

As with the pressure interface, the display freezes when the pen is lifted from the display. While this functionality forces the users to place the pen on the display even when they only want to zoom without panning, it has the benefit of providing an effective clutch mechanism for toggling tilt-control.

The same visual feedback as shown in Figure 3 is provided, but in this case the left box visualizes the angular thresholds along with the current tilt input.

3.1.4 Standard Interface

In the standard interface the users can control the scale by dragging a slider thumb or by tapping plus and minus buttons (see Figure 4). 11 discrete zoom levels are provided. While the slider design is visually similar to the one used in Google Maps, for instance, one difference is that it provides direct feedback, i.e. the view is updated while dragging the thumb and not just when it is released. Panning is implemented as conventional drag&drop, i.e. the users drag the information space with the pen. Compared to the other interaction techniques, the standard interface is limited to discrete and, in particular, sequential interactions. Advantages may be the more precise navigation control, and the fact that the users can perform larger changes in scale more quickly. Since the slider widget already indicates the current scale level, no further visual feedback is provided for the standard interface. To reduce occlusion by the user's hand when operating the slider, we placed the control on the right side of the map view (the study included right-handed participants only).

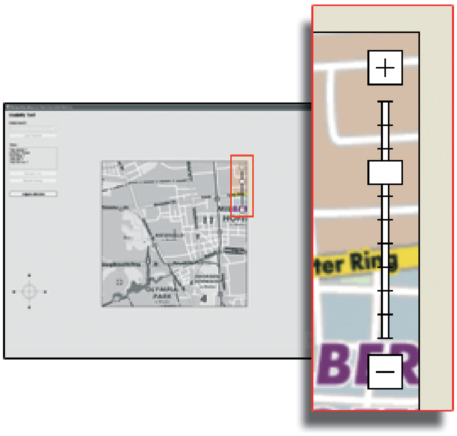


Fig. 4. Zoom slider of the standard interface with 11 discrete levels. The users can also increase or decrease the current level stepwise by tapping the plus and the minus buttons.

3.2 Map Information Space

Most geographic ZUIs make use of semantic zooming, in which the granularity of the map information changes with the scale. For the evaluation, we therefore created a simple semantic information space consisting of three bitmaps that show the same city map of Munich at different scales[7]. The first map layer has a size of 6000x6000 pixels and shows the city at a high level of detail (scale: 1:10,000). The layer includes a complete labeling of main roads, small streets, public transport lines, tourist landmarks, official buildings, parks, etc.

[7] For the maps see http://www.stadtplandienst.de.

The second layer shows the map at a scale of 1:20,000. The granularity is reduced in that small streets and buildings are omitted. The public transport lines are still visible, but without labels. In contrast, the labels of the city districts are highlighted. The third layer has a scale of 1:40,000 and accordingly the amount of detail information is further reduced. Only some of the main roads are labeled, and the landmarks and transport lines have been completely removed.

For the standard interface, the semantic layers are swapped according to the scale values of the distinct zoom level, i.e. when zooming out from the highest magnification of the map, the first layer will be scaled down with each zoom level until it is replaced by the second layer, which in turn is scaled down until the third layer is eventually displayed.

For the three alternative interfaces that support continuous zooming, we were not satisfied with the abrupt swapping effect, given the otherwise fluent zoom. Thus we enhanced the transitions between the layers with fluent alpha blending. For instance, the transition between the first layer and the second layer starts at a scale of 0.8 and ends at scale 0.6. During this scale range the second layer is blended onto the first layer with a continuously decreasing transparency that is proportional to the value of the current scale within the transition range. Hence when the scale has reached a value of 0.6, the second layer is shown at full opacity while the first layer is no longer visible.

The alpha blending effect resulted in a smooth semantic zoom, which was found to be very appealing and natural by all of the participants in the preliminary test. In fact, the users did not perceive the map as three distinct bitmaps, but as a single, dynamically changing information space. Initially, we were also concerned that if the users were, by chance, to stop zooming exactly midway between a transition of two layers, the blending effect might disturb them. However, as it turned out, this transition-situation happened too seldom to cause a serious problem.

3.3 Tasks

When developing the following four task types, our objective was to cover a variety of use cases ranging from simple speed tasks to complex search and map navigation tasks. Each task starts with the view being set to the maximum magnification level of the map.

(1) *Halo-navigation*
The users were asked to navigate to an offscreen target as quickly as possible. The target consisted of a circle (radius 20 pixels at maximum scale), which was continuously cued by a red halo (i.e. off-screen objects are surrounded

by an ellipse (halo) just large enough to reach into the viewport; based on the curvature of a halo, users can judge not only the direction but also the distance of the corresponding off-screen object with respect to the current viewport). Targets were only tappable at the highest magnification level, which was further indicated by the circle changing its color from gray to red. Upon a successful hit, the system produced a beep of 150 milliseconds, the circle was removed from the display, and a new halo-enhanced offscreen target was generated. In total the users had to reach ten targets in succession. The targets were located at a distance of 2500 pixels from each other and the directions were set at random but within the bounds of the map. Unlike in the original design by Baudisch and Rosenholtz (2003), the halos in our application did not disappear once the target objects entered the viewport. Instead they became an outer concentric circle (radius 25 pixels) of the onscreen target. We found that this approach improved the identification of the target, and particularly so when the circle was drawn in a highly cluttered map region.

The main purpose of halo-navigation was to measure speed. It may correspond to the rather rare scenario in which the users are highly familiar with a given map and know in which direction and distance a point-of-interest is located. The most effective strategy to reach the target is to zoom out, pan, and, once the target object appears on the screen, to zoom in on it. While the halos are rendered without any relation to the underlying information space, the map was nevertheless displayed to provide additional visual feedback to the users.

(2) *Follow-route*

The users had to find and tap a named station by following an underground line. When starting the task, the view was automatically positioned such that the relevant underground line crossed the view. The label of the target station was highlighted with a red rectangle, which was only visible (and tappable) at the highest magnification level. Due to the fact that the underground lines in Munich are of similar length, we were able to keep the distance between the initial view position and the target station rather equal for all tasks. To support the users in case they became lost (e.g. accidentally switched to another line at an intersection), the start location was marked by a red halo. Moreover, we enhanced the 1:10,000 map with additional labeling to better distinguish between the individual transport lines.

With the follow-route tasks, our main interest lay in evaluating the panning functionality. Zooming did not provide a performance benefit, as the station labels were hardly readable or simply not visible at scales other than the highest magnification level. Adjacent stations were also positioned very close to each other on the map, which would have required users to constantly zoom in and out to check the name for every station. Example task: *Follow the underground line number 5 from Odeonsplatz (start) to Laimer Platz (destination). The destination is located west of the start position.*

(3) *Find-route*

This task type asked the users to find a route between two given landmarks (e.g. a slip road, a park, etc.) that, due to their distance apart, could not be displayed in a single view. The route had to be submitted orally as driving directions based on street names. The start and destination landmarks were marked by differently colored halos. The participants were instructed to choose the shortest route based on the main streets (colored yellow on the map). The task ended when the participants tapped a red target circle positioned at the destination.
Routing is probably the most common real-life task when reading maps. In the experiment it was also the most complex task type to solve. It usually required the participants to frequently zoom out to maintain the overview of the route, and then to zoom to a higher magnification level to be able to read the street names more easily. Example task: *Find a route from Theodor-Heuss-Platz to the Ostbahnhof.*

(4) *Find-landmark*
The users were given a printout showing a map clipping of 225x225 pixels. On this clipping a target position (e.g. a street crossing) was marked by a red dot. The users had to locate and tap the target position, which on the digital map was highlighted by a red circle. The target was only visible at the highest magnification level.
For each interface the users had to solve three find-landmark tasks in succession, with each task being based on a clipping of another semantic layer of the map interface. All clippings contained an item of symbolic or alphanumeric information that was represented on all three map layers. This task type may correlate to a typical map exploration, in which the users browse an information space in search of certain visual information.

3.4 Apparatus

For the evaluation we used a Toshiba Portégé Tablet PC running Windows XP Tablet PC Edition. The device is equipped with a Pentium M 1.6 GHz processor, 1GB of RAM, a 12.1 inch LCD display, a built-in dual axis accelerometer and a digitizer with 255 levels of pressure. The experiment was conducted at a resolution of 1400x1050 pixels. For pen input the participants used a standard Tablet PC pen.

While the Tablet PC allowed us to simulate two different display sizes, it is too heavy (2 kilo) to be held for a longer period with one hand, and thus would have rendered tilt-control unusable. To overcome this problem, we used a metal rack equipped with springs that held the Tablet PC in an upright position and counterbalanced its weight (see Figure 5a, b). In this way, the users could rest their non-dominant arm on a table and tilt the device with a minimum of force and arm movement. We believe that the apparatus provided us with a reasonable approximation of tilting a smaller, lightweight device such as a PDA. Nevertheless, the differences in ergonomics and the constrained range of device movement have to be taken into account when interpreting the results.

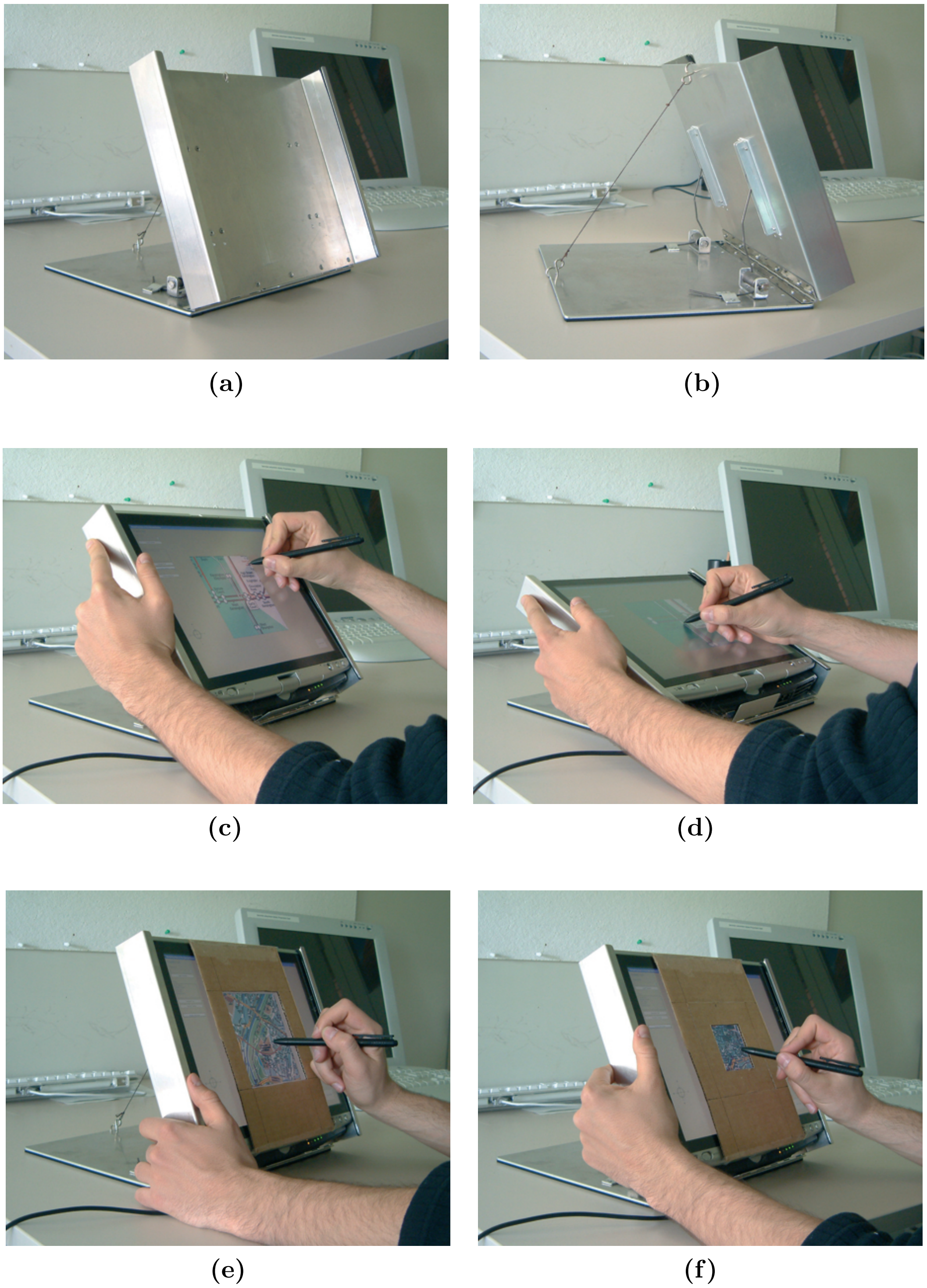
The rack enables a total tilting range of 41 degrees. In the default position (Figure 5a), the Tablet PC is fixed at an orientation of 24 degrees (measured from a vertical axis). Since this is not the most convenient angle for reading the display, we hoped that it would motivate the users to tilt the device to an individual orientation right from the start (image c). This was especially important for the tilt interface, since for zooming in and out the users needed to pitch the device in two directions from the current orientation when setting the pen on the display. Tilting the device all the way upwards resulted in a maximum angle of 65 degrees (image d). The metal rack was used for all interaction techniques. To simulate a display bezel we used two differently sized passepartouts cut out of cardboard and attached to the Tablet PC screen (images e, f).

3.5 Hypotheses

We defined the following hypotheses to test with our experiment:

(1) *The task-completion times for halo-navigation would be equal for the 600x600 pixels and the 300x300 pixels interfaces.*
Halo-navigation tasks are identical to what has been termed multiscale pointing tasks: the participants have to reach and tap a continuously cued target object located in a ZUI. No semantic information needs to be processed; the users mechanically follow the cues as quickly as possible. The implication of view size on such tasks has been investigated by Guiard et al. (2004) and Guiard and Beaudouin-Lafon (2004). The authors found that display miniaturization has a cost in terms of navigation time, but the study also indicated a ceiling effect at a rather low level, i.e. for displays smaller than about 80x60 pixels. This constraint may therefore be more applicable to wrist-watch interfaces, for instance, than to PDAs and smartphones. With regard to the larger view settings tested in our evaluation, we expected that display size would not have an effect on task-completion time for halo-navigation tasks.

(2) *The total task-completion time for semantic tasks (task types 2 to 4) would be better for the 600x600 pixels interface compared to the 300x300 pixels interface.*
In typical multiscale pointing tasks, little space is needed to effectively cue the target object, for instance by using a halo (Baudisch and Rosenholtz (2003))



(a) **(b)**

(c) **(d)**

(e) **(f)**

Fig. 5. Experimental apparatus: (a) the metal rack, (b) two springs are fixed to the back of the rack to counterbalance the weight of the Tablet PC; the string is used to hold the device in an upright position, (c) the rack with the Tablet PC inserted, (d) the user tilts the device upward to the maximum angle, (e) the apparatus with a 600x600 pixels cardboard passepartout attached to the display, (f) the 300x300 pixels passepartout.

or concentric circles (Guiard et al. (2004)). Given a certain minimum size of the display, the users therefore do not benefit from a larger view. This is different for semantic tasks in which the content of the information space is essential for solving the task. The larger the view, the more context information is displayed, which may reduce unnecessary navigation (Schaffer et al. (1996); Büring et al. (2006)). Hence we expected that for the semantically more demanding task types the smaller display size would increase the task-completion time.

(3) *The users would be faster using the tilt interface compared to the pen pressure interface.*
The tilt and pressure interfaces provide equal functionality to the users in the sense that they both allow for concurrent, but independent, zooming and panning. While we are not aware of any previous research comparing the usability of these two input techniques, there has been extensive work on investigating unimanual versus bimanual interface control. Two-handed input has been found beneficial in various scenarios (e.g. positioning and scaling (Buxton and Myers (1986)), document scrolling (Zhai and Smith (1999)), and image alignment (Latulipe et al. (2005))), but its performance appears to be strongly dependent on the type of task to be performed (Kabbash et al. (1994)). Owen et al. (2005) recommends that bimanual input should be used for tasks that are visually and conceptually integrated, which is the case for ZUI navigation (see also Jacob et al. (1994) for zooming and panning as an integral task). In fact, many studies on ZUI interaction have successfully employed bimanual input. For instance, in the experiments conducted by Guiard et al. (2001), Guiard et al. (2004) and Guiard and Beaudouin-Lafon (2004) users controlled zooming by a joystick with their non-dominant hand and panning via a mouse or a stylus on a tablet with their dominant hand. Bourgeois et al. (2001) and Bourgeois and Guiard (2002) specifically examined the effect that bimanual and parallel input may have on user performance in a ZUI. The results showed that multiscale pointing performance strongly depends on the degree of pan-zoom parallelism, and that a higher degree of parallelism is better supported by two-handed input. Hence we expected the bimanual tilt interface to outperform the unimanual pressure-based interface. Another reason for this hypothesis was our impression that, due to the separate input devices (device tilting and pen position), the tilt interface required less fine-motor skills compared to the integrated pen input.

(4) *Both the tilt and pressure interfaces would be faster than the standard interface.*
This hypothesis was again based on the results of Bourgeois et al. (2001) and Bourgeois and Guiard (2002), which indicate that users are able to perform zooming and panning in parallel, and that such parallelism improves task-completion times in multiscale pointing tasks. While parallel actions are supported by the tilt and the pressure interface, the standard interface is limited

to sequential navigation. However, we were also aware that the users' unfamiliarity with the non-standard interfaces may reduce the performance benefit; for instance, when the users failed to take advantage of a parallel strategy as described by Buxton and Myers (1986).

(5) *SDAZ would be faster for halo-navigation compared to the other interfaces.*
As found in previous literature, the binding of zooming and panning can provide an effective solution for basic navigation and browsing tasks (Cockburn and Savage (2003); Patel et al. (2004); Cockburn et al. (2005); Savage and Cockburn (2005)). We expected to observe the same effect for our halo-navigation tasks, in which the underlying map information can be ignored by the users.

(6) *SDAZ would be slower for semantic tasks (task types 2 to 4) compared to the other interfaces.*
We believed that there is a mismatch between the demands of supporting real-life map-navigation tasks and the interaction provided by SDAZ. Reading cluttered semantic information such as map symbols or small-sized street names in different orientations requires the users to temporarily reduce the pan speed or to bring the movement to a complete halt. However, doing so will cause the SDAZ interface to increase the zoom level, which changes the view and thus requires the users to reorient. Moreover, with semantic zoom (Perlin and Fox (1993)), some data may only be present at a certain scale level. To read such information, the users must pan back and forth, but with each change in direction the current zoom factor is lost and the users need time to readjust it. Another serious drawback of SDAZ is that when releasing the pointing device the current view state is not locked; instead, the interface falls back to the highest magnification level. This is particularly troublesome when using pen-operated mobile devices since (a) in a mobile context the users may frequently have to interrupt the navigation (e.g. when being spoken to), and (b) having located an area of interest at a lower magnification level, the users may want to view the map clipping without the physical effort and occlusion of constantly pressing the pen on the display. To sum up, SDAZ is a highly unsteady interface for focusing on detail information within a limited navigation radius at a lower magnification level. The enforced need for continuous panning to maintain a constant scale level can easily lead to time-consuming disorientation (Jones et al. (2005)). Hence, the feature of independent zooming and panning provided by the other interfaces may be more effective for solving the semantic tasks.

(7) *If constrained to use the smaller view of 300x300 pixels, the performance decrease for semantic tasks (task types 2 to 4) would be most distinct for SDAZ compared to the other interfaces.*
In hypothesis 2 we assumed that a smaller screen increases task-completion times for semantic tasks due to insufficient context. Since the lack of context

is likely to lead to otherwise unnecessary navigation, we expected that the decrease in performance would be most notable for SDAZ, where each change in pan direction brings the further disadvantage of losing the current zoom level. This would also partly conform to the results of Jones et al. (2005).

(8) *The participants would prefer the standard interface and reject the SDAZ interface.*
The standard interface benefits from the users' familiarity with sliders and drag&drop, which we believed would increase user confidence and thus result in a comparatively positive user rating. While SDAZ has also been found easy to learn (e.g. Cockburn and Savage (2003); Patel et al. (2004)), its potential inadequacy for semantic tasks (see hypothesis 6) would cause a high amount of user frustration, leading to the lowest rating. The tilt and pressure interfaces offer the most flexible navigation features, but require the users to control an additional input dimension by a fairly unusual interaction mechanism. Moreover, in previous research novice users have been repeatedly reported to have difficulties with accurately controlling pen pressure and device tilting. We assume that these results are at least partially due to insufficient training, which, given the limited time frame of an evaluation, can hardly be avoided. In summary, we expected our tilt and pressure interfaces to result in a rather moderate rating.

(9) *Across the different task types, the subjective workload would be lower for the standard interface compared to SDAZ.*
In previous research SDAZ has been found to produce less subjective workload than conventional scroll and zoom-pan approaches (Cockburn and Savage (2003); Savage and Cockburn (2005); Cockburn et al. (2005)). However, with respect to the semantic task types tested, we expected that the participants would find the decoupled interaction of the standard interface less demanding than SDAZ. This assumption corresponds to hypothesis 6.

3.6 Participants

For the study we selected 32 subjects, 14 male and 18 female. All of them were students or Ph.D. students at the University of Konstanz. Their ages ranged from 20 to 33 years, with 24.47 years as the mean value and a standard deviation of 3.25 years. Their fields of study varied greatly. No participant was a computer science student and they had been students for a mean time of 6.52 semesters. The pre-test questionnaire showed that two of our subjects actually owned a PDA, and 13 more had at least tried one and were therefore familiar with the general pen-interaction concept. All of our users were regular PC and internet users. Since we used a map of Munich for our experiment, we also asked our participants about their knowledge of Munich. None of them actually came from Munich and on a 5-point scale (1 = no familiarity at all, 5 perfect familiarity) they rated their familiarity with a mean of 1.59 and a standard deviation of 0.76. As expected, later analysis confirmed that prior Munich knowledge did not have an effect on task performance.

3.7 Experimental Design

We used a 2x4x4 split-plot design, the first being a between-subjects factor and the latter two being within-subjects factors. The between-subjects factor was *view size* (300x300 pixels and 600x600 pixels) resulting in two different groups, each of 16 participants. The within-subjects factors were *interface type* (four different levels) and *task type* (four different levels). For each interface, a different task set was developed to reduce learning effects. We used a latin square design to counterbalance the interface types and the task sets, resulting in four groups (with four subjects each) per interface order and four groups per task-set order. We randomly assigned each participant to one of the resulting 16 groups. The dependent variables were task-completion time (in seconds), interface preference, and subjective workload. The workload was measured with the NASA Task Load Index questionnaire (NASA TLX).

3.8 Procedure

The session started with a short written introduction and the pre-test questionnaire. After that, the users were introduced to the pen handling on the TabletPC. During this process, the pen was recalibrated by the participants themselves. Next, the test administrator explained the general procedure and handed out the first interface explanation. The explanation was a written description of the interface functionality. Participants were allowed and encouraged to ask questions. Subsequently, they were given time to try the application with a London Underground map. Each participant had to complete two halo-navigation tasks before advancing to the test interface. When participants showed that they had understood the interface, the Munich map was loaded and each of the eleven tasks was presented to them successively as a printout. The first two tasks were halo-navigation tasks, followed by three follow-route tasks, then three find-route tasks, and finally three find-landmark tasks. Since every task type required a different strategy for each interface type, we excluded the first task of each type from the analysis and marked them as training tasks, without the knowledge of the participants. Participants read all questions aloud and then pressed a start-task button on the lower right of the screen. To complete a task, a marked target had to be tapped. The participants could also cancel a task by pressing a button on the left hand side. Moreover, a time-out routine was defined, which interrupted the task automatically after 5 minutes. Upon completion of a task the screen immediately turned black. After finishing all tasks for the first interface type, the participants had to fill out a paper version of the NASA TLX translated into German, the participants' native language. The questionnaire measured the average workload on six scales, each weighted through a pairwise comparison procedure. Subsequently, the same procedure was repeated in succession for the remaining three interfaces and the corresponding task sets. As the last step, the participants completed a preference questionnaire and were then handed a movie theatre voucher worth EUR 10,-. Experiment sessions lasted about 90 to 120 minutes, and were recorded and video taped for documentation purposes.

4 Results

For the analysis of the task-completion times we used mainly RM-ANOVA, a priori contrast analysis, and post-hoc pairwise comparisons, the latter two with Bonferroni adjusted significance levels when needed. For our main 2x4x4 design including two levels for the between-subjects factor view size and four levels each for the within-subjects factors interface type and task type, we could assume sphericity for our within-subjects factor interface type but not for task type, which is why we used the Greenhouse-Geisser degree of freedom adjustment in this case. Throughout the analysis we used a significance level of 5 percent. For four participants we logged a cancellation (3) or a timeout (1) of a task, but each time only a single task within a task type for an interface was affected. Hence, in those cases the generated mean was equal to the task-completion time of the corresponding second task, which was properly completed.

H1: Influence of different view sizes for halo-navigation tasks.
Our results show that it took our participants about 92.6 seconds on average to complete the halo-navigation tasks on the large display, but only 84.25 seconds on average on the small display (standard error: 3.14 seconds). However, an RM-ANOVA analyzing the main effect of the between-subjects factor view size shows that this difference is not significant on the 5 percent significance level ($F(1, 30) = 3.532, p = 0.07$). While this result seems at first glance to agree with previous research (Guiard et al. (2004); Guiard and Beaudouin-Lafon (2004)), it must be noted that, due to the relatively low observed power of 0.444 (resulting in a type 2 error of 56.6 %), the test cannot provide clear empirical evidence that halo-navigation is not affected by view size.

H2: Influence of different view sizes for semantic tasks.
To analyze the main effect for our between-subjects factor, we used a 2x4x3

design, excluding the halo-navigation tasks. The results show that there is indeed a difference between the large and small view sizes, resulting in an average task-completion time of 34.25 seconds for the 600x600 pixels view compared to 42.35 seconds for the 300x300 pixels view (standard error: 2.07). Furthermore, this difference is significant, the significance being backed up by a relatively large effect size ($F(1,30) = 7.662, p = 0.01, hp^2 = 0.203$). Hence, we can accept the experimental hypothesis and reject the null hypothesis. An in-depth analysis of the pairwise comparisons reveals that task-completion times for task types 2 and 3 differ significantly, but not for task type 4 (one-sided, Bonferroni adjusted t-tests, see Table 1 and Figure 6).

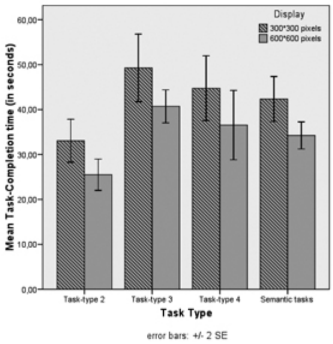


Fig. 6. The difference in task-completion time between the two view sizes is significant for the task types 2 and 3 for semantic tasks.

H3: Comparing the performance of the tilt interface and the pressure interface. It took the participants about 46.33 seconds on average to complete the tasks with the pressure interface (standard error: 2.42 seconds) and 45.90 seconds on average with the tilt interface (standard error: 1.73 seconds). This difference is non-significant (analyzing simple contrasts: $F(1,30) = 0.019, p = 0.891$). Also, further in-depth pairwise comparisons do not show any significant effect for one of our four task types. Therefore we have to reject the hypothesis and confirm the null hypothesis for the time being.

With regard to navigation behavior, we logged the time the users spent for panning and zooming in parallel while using these two interfaces. The analysis reveals that the participants spent more time panning and zooming in parallel while using the pressure interface (3.58 seconds on average per task, standard error: 0.232 seconds) than while using the tilt interface (2.92 seconds on average per task, standard error: 0.26 seconds). This difference is significantly in favor of the pressure interface ($F(1,30) = 8.587, p = 0.006, hp^2 = 0.223$) and indicates that, in contrast to our expectations, the integrated pen input seems to have encouraged parallel zooming and panning more than the bimanual approach of the tilt interface. However, considering the mean task-completion times of about 46 seconds, our participants relied strongly on separate zooming and panning regardless of which of the two interfaces they used.

H4: Comparing the performance of tilt and pressure with the standard interface.
Since the previous analysis already indicated that our participants made only very limited use of concurrent zooming and panning, it is not surprising that H4 does not hold. Moreover, our analysis shows that both the pressure and the tilt interfaces are slower than the standard interface (46.33 seconds for pressure and 45.90 seconds for the tilt interface compared to 43.15 seconds and a standard error of 1.72 seconds for the standard interface). A contrast analysis (simple contrasts) shows that this difference is not significant (standard vs. pressure: $F(1, 30) = 1.714, p = 0.4$, standard vs. tilt: $F(1, 30) = 1.22, p = 0.556$, Bonferroni adjusted). Investigating a possible influence of the view size, it seems that the difference in task-completion time results from the small view size (see Table 2 and Figure 7).
However, a contrast analysis does not show a significant main interaction effect for interface x view size (standard vs. pressure: $F(1, 30) = 3.68, p = 0.128$; standard vs. tilt: $F(1, 30) = 1.43, p = 0.484$, Bonferroni adjusted). Nevertheless, we performed single sided pairwise comparisons comparing the standard interface with the pressure and the tilt interfaces for the small view size. The results are significant only for the standard interface compared to the pressure interface (standard vs. pen-pressure: $p = 0.03$; standard vs. tilt: $p = 0.114$; Bonferroni adjusted). Furthermore, the results show that the standard interface does not benefit at all from the larger view size and even performs slightly worse than the other two interfaces for the 600x600 pixels view, though this difference is not significant.

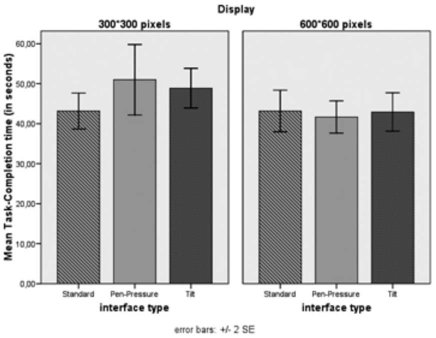


Fig. 7. Unlike the pressure and the tilt interfaces, the standard interface does not benefit from the larger view size of 600x600 pixels.

H5: Comparing the SDAZ interface to the three other interfaces for halo-navigation.
The mean task-completion times for this task type are listed in Table 3 and displayed in Figure 8. An RM-ANOVA reveals that the four interface types differ significantly $((F3, 90) = 12.764, p = 0.000, hp^2 = 0.198)$. Further in-depth single-sided pairwise comparisons for the SDAZ interface show that the observed difference is significant compared to the pressure and the tilt interfaces (in each case $p = 0.000$), but not to the standard interface, although it

is close to being significant ($p = 0.55$, Bonferroni adjusted alpha level).

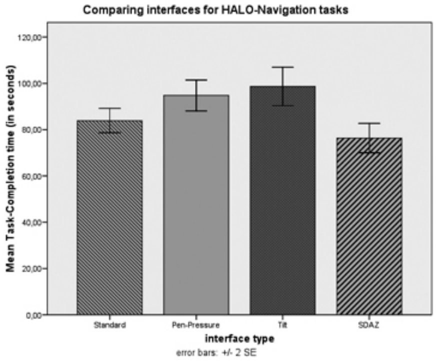


Fig. 8. SDAZ is significantly faster for halo-navigation compared to the pressure and the tilt interfaces.

H6: Comparing the SDAZ interface to the three other interfaces for semantic tasks.
Our results indicate that, despite the SDAZ interface being the slowest, this hypothesis does not hold. The task-completion times using the different interfaces for the task types 2 to 4 are listed in Table 4 and displayed in Figure 9. A contrast analysis comparing the SDAZ mean task-completion time with the three other interfaces shows that the difference is not significant (SDAZ vs. standard: $F(1, 30) = 3.26, p = 0.83$; SDAZ vs. pressure: $F(1, 30) = 1.58, p = 0.218$; SDAZ vs. tilt: $F(1, 30) = 1.84, p = 0.185$)).

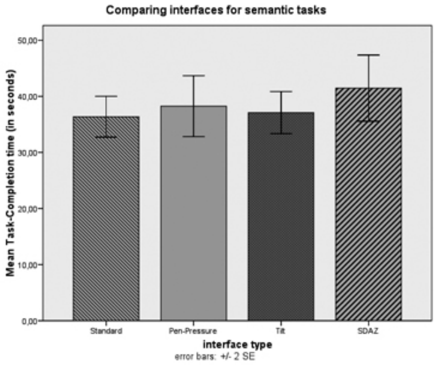


Fig. 9. The difference in performance between the SDAZ interface and the other experimental interfaces for semantic tasks is non-significant.

H7: Influence of view size on solving the semantic tasks with the SDAZ interface in comparison to the other interfaces.
To analyze the hypothesis, we first had to create a new variable reflecting the difference between the mean task-completion time and the mean for the corresponding view size. We then performed an RM-ANOVA with this variable as the within-subjects factor. Since the view size was already represented in this

variable, no additional between-subjects factor was included in the analysis. An RM-ANOVA shows that there is no significant main effect for the four different interfaces ($F(3, 93) = 1.255, p = 0.294$). Contrast analysis furthermore reveals that the SDAZ interface is not influenced more than any of the other interfaces (SDAZ vs. standard: $F(1, 31) = 2.272, p = 0.142$; SDAZ vs. pressure: $F(1, 31) = 0.125, p = 0.726$; SDAZ vs. tilt: $F(1, 31) = 0.193, p = 0.663$). Table 5 and Figure 10 display the differences from the mean of the two view sizes. The high error values give further support for the contrast analysis being non-significant. We must therefore reject the hypothesis.

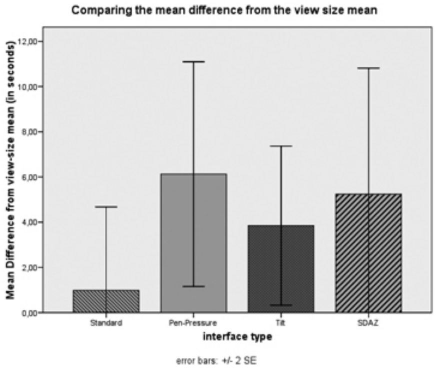


Fig. 10. For semantic tasks the SDAZ interface is not influenced more than the other experimental interfaces by the different view sizes.

H8: Interface Preferences.
Our participants ranked the four interfaces after using them from 1 (best) to 4 (worst). Across the two view sizes, the frequencies for being voted into first place by the 32 participants were: standard: 13, pressure: 9, tilt: 9, and SDAZ: 1. Standard, pressure, and tilt received about the same number of second- and third-place votes, but the latter two were ranked more often in the last place than the standard interface (four times for the pressure interface and five times for the tilt interface). A Chi-Square analysis shows that the standard, pressure and tilt interface received significantly more first place votes compared to the SDAZ interface (standard vs. SDAZ: $X^2(1, N = 14) = 10.286, p = 0.001$; pressure vs. SDAZ: $X^2(1, N = 10) = 6.4, p = 0.011$; tilt vs. SDAZ: $X^2(1, N = 10) = 6.4, p = 0.011$).
Analyzing the results for the two view sizes separately gives the distribution of first-place votes as listed in Table 6. These votes give the impression that, while for the 600x600 pixels view the participants slightly preferred the novel pressure and tilt techniques even more than the highly familiar standard interface, the latter interface proved to be more resilient to the constraints of a smaller view size. However a chi-square analysis shows that the differences in first place votes between the two novel approaches and the standard interface are not significant in both cases. As assumed, the SDAZ interface was strongly rejected for both view sizes.
We also analyzed the overall rankings using t-tests for dependent samples. The

mean ranks are listed in Table 7. Analysis indicates that SDAZ was ranked significantly lower compared to the other three interfaces (SDAZ vs. standard: $T(31) = -6.334, p = 0.000$; SDAZ vs. pressure: $T(31) = 4.322, p = 0.000$, SDAZ vs. tilt: $T(31) = 3.745, p = 0.003$, Bonferroni adjusted). However, none of the other comparisons did reveal any further significant differences between the interfaces. The complete distribution of votes across view sizes is given in Table 8.

H9: Subjective Workload.
The total mean scores for the four interfaces as measured by the TLX questionnaire are shown in Table 9. An RM-ANOVA shows a significant main effect for interface type ($F(3, 90) = 7.456, p = 0.000, hp^2 = 0.199$) and a significant interaction effect for interface type x view size ($F(3, 90) = 3.897, p = 0.011, hp^2 = 0.115$). We therefore analyzed the simple effects through pairwise comparisons. The results indicate that the standard interface caused significantly lower workload than each of the other three interfaces (pressure: $p = 0.009$, tilt: $p = 0.008$, and SDAZ: $p = 0.001$). The differences between the other interfaces are not significant. Moreover, analyzing the simple interaction effects, we found that the differences only occur for the 300x300 pixels view size (standard: 31.40, pressure: 48.96, tilt: 45.69, SDAZ: 60.90). Again, only the differences compared to the standard interface are significant. In contrast, the 600x600 pixels view resulted in more or less equal workloads for all four interfaces (standard: 40.19 points, pressure: 44.10 points, tilt: 44.67 points, SDAZ: 44.96 points).

While the TLX results seem to support the experimental hypothesis, an unexpected result was that the workload for the standard interface increased when using the larger view. In contrast, the other three interfaces seem to have benefited from the large display size in terms of a reduced workload. In our opinion, this effect borders on the inexplicable, since one must assume that a larger view size would be more convenient to work with, regardless of the interaction technique used. We are therefore rather sceptical of the validity of the TLX results. One aspect that further supports this doubt is the observation that many participants felt bothered by the frequent questionnaires and particularly so by the weighting procedure. In some cases this may have resulted in more or less random scoring.

5 Discussion

The experiment conducted led to a variety of interesting results. First, we found that the performance for tasks that require the users to read semantic map information was significantly decreased by a smaller view size of a

300x300 pixels view compared to a 600x600 pixels view. The task types responsible for this effect were follow-route and find-route, for which context information seems to be most important. The find-location tasks, which focused on exploration, were less affected by the change of the view size. Previous research indicates that display miniaturization also increases navigation time for multiscale pointing tasks (similar to halo-navigation), but with the performance loss taking effect at a much lower view size, i.e. for displays smaller than about 80x60 pixels (Guiard et al. (2004); Guiard and Beaudouin-Lafon (2004)).

The results for the SDAZ interface were mixed. As expected, for simple halo-navigation tasks, automatic zooming provided a better performance than the other interfaces, with significant differences to the pressure and the tilt interfaces. This result partly corresponds to previous research. In terms of task-completion times, SDAZ also provided a reasonable solution for solving more realistic map navigation tasks, which required the users to constantly read and process semantic information. Though SDAZ was slower than the other interfaces, the differences were non-significant. However, the predicted inappropriateness of SDAZ for such tasks was nevertheless indicated by the preference ratings and user comments. The participants clearly rejected automatic zooming compared to the other interfaces. Frequent comments were that it was difficult to solve tasks using SDAZ because of the coupled zooming and panning, which for the semantic task types was found tedious, imprecise and hard to control (9 users). Positive comments were that SDAZ was found to be particularly effective for halo-navigation, which corresponds to the performance results regarding that task type (8 users).

We did not find evidence that SDAZ may be less effective than other interfaces for a smaller view size. The moderate results of SDAZ on a simulated PDA screen as reported by Jones et al. (2005) may be due more to implementation issues. As has already been suggested by Cockburn and Savage (2003), the usability of SDAZ depends strongly on the smoothness of zooming and panning, which may not have been supported by the Java prototype used by Jones et al. (2005).

Both the pressure and the tilt interfaces provided the users with concurrent but separate control of zooming and panning. During the experiment, however, the users hardly ever took advantage of parallel movement, which corresponds to a similar observation by Buxton and Myers (1986). It seems that, in the test scenario, the users felt more comfortable solving the tasks with the easier-to-control discrete zoom and pan operations. We assume that more extensive training would change the navigation behavior, as was also the case, for instance, with our expert users, i.e. the colleagues who accompanied the development of the interaction techniques over weeks and tested them frequently. With the navigation strategy used by the participants, both interfaces resulted in comparable task-completion times across task types. When compared to the standard interface, the performance was significantly decreased when using the small view, but on the larger view the mean task-completion times were fairly equal for the three interfaces.

The preference ratings indicate that the participants liked the pressure and tilt control almost as much as the standard interface. For the larger view of 600x600 pixels, the tilt interface was actually the most preferred technique. The user comments suggest that tilting was fun, as well as fast and easy to use (14 users), while the pressure interface was mostly appreciated for its intuitiveness and the low effort it required (14 users). Considering the users' high familiarity with the standard interface, both the preference and the performance results for the novel pressure and tilt techniques are highly encouraging. However, particularly in case of the tilt interface, the influence of the metal rack as part of the experimental apparatus has to be considered. It is up to further research to clarify the effects a truly mobile setting would have on the interface performance. Another aspect one has to keep in mind is that the usability of all three experimental map browsers strongly depends on the values of the various interface parameters. A generalization of the reported results to a wide range of interface settings may not be feasible.

The standard interface proved to give a reliably high performance across the different task types compared to the other three interfaces. This is despite the accumulated travel time enforced by the need to frequently move the pen back and forth between the zoom slider and the focus area. Moreover, the separation of zooming and panning was not perceived as a limitation by many participants, but as a feature to enable more accurate view movement and facilitate control (9 users). Another interesting property of the standard interface is that the preference and performance values turned out to be particularly good for the small view size of 300x300 pixels. This may indicate that, on a smaller screen such as featured by mobile devices, precise control becomes an increasingly important usability factor.

6 Outlook

While the standard interface proved to significantly reduce task-completion times for the small 300x300 pixels view, we still believe that the results gained for the alternative interfaces strongly encourage further development. Significant potential for improvement lies in optimizing interface parameters and the overall interaction design. Regarding the handling of SDAZ for map navigation, the interaction may for instance be improved by introducing a mechanism that allows the users to view a scaled subset of the map without movement, e.g. the view could freeze when the users lift the pen from the display, with the last position of the pen being marked by an icon. If the users want to automatically zoom back in they just tap anywhere on the display (this could also be implemented as point-directed zooming). If they want to continue the previous zoom and pan movement, they tap and drag the icon. While this strategy means a more complex interaction, it may result in a more stable interface. Another option to ease SDAZ interaction may be to introduce a fixed reference point for scale and velocity as proposed by Patel et al. (2004). That way the interface can provide navigation shortcuts.

In case of the novel pressure and tilt interfaces, the more critical user comments that were collected indicate strong individual differences in how well participants coped with each of the two interfaces. Whilst ten users explicitly stated that the pressure-based zoom was difficult to control, many more users were observed to frequently trigger zoom-in or zoom-out operations by accident. In a real-life application, this problem may be reduced by individual calibration of the pressure threshold settings. As expected, the tilt-controlled interface seemed to require less fine-motor skills, but the mapping of the tilt movement to the zoom direction was found less intuitive compared to the pressure interface. Some participants were observed to frequently tilt the device in the wrong direction. One solution for this issue may be to extend the visual feedback to emphasize more clearly the effect a certain tilt movement will have. Overall, we think that the results of the tilt interface also provide a positive feedback for the apparatus setup. In the future we would like to verify the findings for the tilt interface with a PDA device.

An issue we observed for both the pressure and the tilt interfaces was that some participants initially became confused by the fact that the pen did not serve as a device for point-directed zooming. For instance, in a situation in which the target object is already visible in the view, the users seemed to assume that by tapping the target, they would automatically zoom in on it. However, as previously described pen strokes are interpreted as relative navigation commands for steering the view. This mismatch of navigation style and user expectation may be worth investigating in further research. Another aspect to explore is to equip the pressure and the tilt interfaces with more concise navigation elements that may better match the users' needs on PDA-like screens. One method, for instance, could be to employ drag&drop panning instead of the quicker, but harder to control, rate-based approach.

7 Acknowledgment

This work was supported by the DFG Research Training Group GK-1042 ”Explorative Analysis and Visualization of Large Information Spaces”. We also thank Dominik Morbitzer for his support in carrying out the evaluation.

References

Agarawala, A., Balakrishnan, R., 2006. Keepin’ it real: pushing the desktop metaphor with physics, piles and the pen. In: CHI ’06: Proceedings of the SIGCHI conference on Human Factors in computing systems. ACM Press, New York, NY, USA, pp. 1283–1292.

Appert, C., Fekete, J.-D., 2006. Orthozoom scroller: 1d multi-scale navigation. In: CHI ’06: Proceedings of the SIGCHI conference on Human Factors in computing systems. ACM Press, New York, NY, USA, pp. 21–30.

Bartlett, J. F., 2000. Rock ’n’ scroll is here to stay. IEEE Comput. Graph. Appl. 20 (3), 40–45.

Baudisch, P., Rosenholtz, R., 2003. Halo: a technique for visualizing off-screen objects. In: CHI ’03: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, New York, NY, USA, pp. 481–488.

Bederson, B., Meyer, J., 1998. Implementing a zooming user interface: experience building pad++. Softw. Pract. Exper. 28 (10), 1101–1135.

Bederson, B. B., Hollan, J. D., 1994. Pad++: a zooming graphical interface for exploring alternate interface physics. In: UIST ’94: Proceedings of the 7th annual ACM symposium on User interface software and technology. ACM Press, New York, NY, USA, pp. 17–26.

Bourgeois, F., Guiard, Y., 2002. Multiscale pointing: facilitating pan-zoom coordination. In: CHI ’02: CHI ’02 extended abstracts on Human factors in computing systems. ACM Press, New York, NY, USA, pp. 758–759.

Bourgeois, F., Guiard, Y., Lafon, M. B., 2001. Pan-zoom coordination in multi-scale pointing. In: CHI ’01: CHI ’01 extended abstracts on Human factors in computing systems. ACM Press, New York, NY, USA, pp. 157–158.

Büring, T., Gerken, J., Reiterer, H., September-October 2006. User interaction with scatterplots on small screens - a comparative evaluation of geometric-semantic zoom and fisheye distortion. IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization / Information Visualization 2006) 12 (5), 829–836.

Buxton, W., 1990. A three-state model of graphical input. In: INTERACT ’90: Proceedings of the IFIP TC13 Third Interational Conference on Human-Computer Interaction. North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands, pp. 449–456.

Buxton, W., Hill, R., Rowley, P., 1985. Issues and techniques in touch-sensitive tablet input. In: SIGGRAPH '85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques. ACM Press, New York, NY, USA, pp. 215–224.

Buxton, W., Myers, B., 1986. A study in two-handed input. In: CHI '86: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, New York, NY, USA, pp. 321–326.

Cho, S.-J., Sung, Y., Murray-Smith, R., Lee, K., Choi, C., Kim, Y.-B., 2007. Dynamics of tilt-based browsing on mobile devices. In: CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, New York, NY, USA, pp. 815–820.

Cockburn, A., Savage, J., 2003. Comparing speed-dependent automatic zooming with traditional scroll, pan, and zoom methods. In: People and Computers XVII: British Computer Society Conference on Human Computer Interaction. pp. 87–102.

Cockburn, A., Savage, J., Wallace, A., 2005. Tuning and testing scrolling interfaces that automatically zoom. In: CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, New York, NY, USA, pp. 71–80.

Crossan, A., Murray-Smith, R., 2004. Variability in wrist-tilt accelerometer based gesture interfaces. In: Mobile HCI. pp. 144–155.

Eslambolchilar, P., Murray-Smith, R., 2004. Tilt-based automatic zooming and scaling in mobile devices - a state-space implementation. In: Mobile HCI. pp. 120–131.

Fitts, P. M., 1954. The information capacity of the human motor system in controlling the amplitude of movement. Journal of Experimental Psychology 47 (6), 381–391.

Fitzmaurice, G., Khan, A., Pieké, R., Buxton, B., Kurtenbach, G., 2003. Tracking menus. In: UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology. ACM Press, New York, NY, USA, pp. 71–79.

Fitzmaurice, G. W., 1993. Situated information spaces and spatially aware palmtop computers. Commun. ACM 36 (7), 39–49.

Fitzmaurice, G. W., Zhai, S., Chignell, M. H., 1993. Virtual reality for palmtop computers. ACM Trans. Inf. Syst. 11 (3), 197–218.

Grossman, T., Hinckley, K., Baudisch, P., Agrawala, M., Balakrishnan, R., 2006. Hover widgets: using the tracking state to extend the capabilities of pen-operated devices. In: CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems. ACM Press, New York, NY, USA, pp. 861–870.

Guiard, Y., 1987. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. Journal of Motor Behavior 19 (4), 486–517.

Guiard, Y., Beaudouin-Lafon, M., 2004. Target acquisition in multiscale electronic worlds. Int. J. Hum.-Comput. Stud. 61 (6), 875–905.

Guiard, Y., Beaudouin-Lafon, M., Bastin, J., Pasveer, D., Zhai, S., 2004. View size and pointing difficulty in multi-scale navigation. In: AVI '04: Proceedings of the working conference on Advanced visual interfaces. ACM Press, New York, NY, USA, pp. 117–124.

Guiard, Y., Beaudouin-Lafon, M., Mottet, D., 2001. Beyond the 10-bit barrier: Fitts' law in multi-scale electronic worlds. In: People and Computers XV - Interaction without frontiers. Springer Verlag, pp. 573–587.

Harrison, B. L., Fishkin, K. P., Gujar, A., Mochon, C., Want, R., 1998. Squeeze me, hold me, tilt me! an exploration of manipulative user interfaces. In: CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, pp. 17–24.

Hascoët, M., 2003. Throwing models for large displays. In: HCI'03: British HCI Group. pp. 73–77.

Herot, C. F., Weinzapfel, G., 1978. One-point touch input of vector information for computer displays. In: SIGGRAPH '78: Proceedings of the 5th annual conference on Computer graphics and interactive techniques. ACM Press, New York, NY, USA, pp. 210–216.

Hinckley, K., 2002. Input technologies and techniques, 151–168.

Hinckley, K., Pierce, J., Horvitz, E., Sinclair, M., 2005. Foreground and background interaction with sensor-enhanced mobile devices. ACM Trans. Comput.-Hum. Interact. 12 (1), 31–52.

Hinckley, K., Pierce, J., Sinclair, M., Horvitz, E., 2000. Sensing techniques for mobile interaction. In: UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology. ACM Press, New York, NY, USA, pp. 91–100.

Igarashi, T., Hinckley, K., 2000. Speed-dependent automatic zooming for browsing large documents. In: UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology. ACM Press, New York, NY, USA, pp. 139–148.

Jacob, R. J. K., Sibert, L. E., McFarlane, D. C., M. Preston Mullen, J., 1994. Integrality and separability of input devices. ACM Trans. Comput.-Hum. Interact. 1 (1), 3–26.

Jones, S., Jones, M., Marsden, G., Patel, D., Cockburn, A., 2005. An evaluation of integrated zooming and scrolling on small screens. Int. J. Hum.-Comput. Stud. 63 (3), 271–303.

Kabbash, P., Buxton, W., Sellen, A., 1994. Two-handed input in a compound task. In: CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, New York, NY, USA, pp. 417–423.

Karlson, A. K., Bederson, B. B., SanGiovanni, J., 2005. Applens and launchtile: two designs for one-handed thumb use on small devices. In: CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, New York, NY, USA, pp. 201–210.

Khella, A., Bederson, B. B., 2004. Pocket photomesa: a zoomable image browser for pdas. In: MUM '04: Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia. ACM Press, New York, NY, USA, pp. 19–24.

Lank, E., Ruiz, J., Cowan, W., 2006. Concurrent bimanual stylus interaction: a study of non-preferred hand mode manipulation. In: GI '06: Proceedings of the 2006 conference on Graphics interface. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, pp. 17–24.

Latulipe, C., Kaplan, C. S., Clarke, C. L. A., 2005. Bimanual and unimanual image alignment: an evaluation of mouse-based techniques. In: UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology. ACM Press, New York, NY, USA, pp. 123–131.

Li, Y., Hinckley, K., Guan, Z., Landay, J. A., 2005. Experimental analysis of mode switching techniques in pen-based user interfaces. In: CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, New York, NY, USA, pp. 461–470.

Mantyjarvi, J., Paternò, F., Salvador, Z., Santoro, C., 2006. Scan and tilt: towards natural interaction for mobile museum guides. In: Mobile HCI. pp. 191–194.

Matsushita, N., Ayatsuka, Y., Rekimoto, J., 2000. Dual touch: a two-handed interface for pen-based pdas. In: UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology. ACM Press, New York, NY, USA, pp. 211–212.

Mizobuchi, S., Chignell, M., Newton, D., 2005a. Mobile text entry: relationship between walking speed and text input task difficulty. In: MobileHCI '05: Proceedings of the 7th international conference on Human computer interaction with mobile devices & services. ACM Press, New York, NY, USA, pp. 122–128.

Mizobuchi, S., Terasaki, S., Keski-Jaskari, T., Nousiainen, J., Ryynanen, M., Silfverberg, M., 2005b. Making an impression: force-controlled pen input for handheld devices. In: CHI '05: CHI '05 extended abstracts on Human factors in computing systems. ACM Press, New York, NY, USA, pp. 1661–1664.

Nacenta, M. A., Aliakseyeu, D., Subramanian, S., Gutwin, C., 2005. A comparison of techniques for multi-display reaching. In: CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, New York, NY, USA, pp. 371–380.

Owen, R., Kurtenbach, G., Fitzmaurice, G., Baudel, T., Buxton, B., 2005. When it gets more difficult, use both hands: exploring bimanual curve manipulation. In: GI '05: Proceedings of the 2005 conference on Graphics interface. Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, pp. 17–24.

Patel, D., Marsden, G., Jones, S., Jones, M., 2004. An evaluation of techniques for browsing photograph collections on small displays. In: Mobile HCI. pp. 132–143.

Perlin, K., Fox, D., 1993. Pad: an alternative approach to the computer interface. In: SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques. ACM Press, New York, NY, USA, pp. 57–64.

Raisamo, R., 1999. Evaluating different touched-based interaction techniques in a public information kiosk. In: Conference of the CHI Special Interest Group of the Ergonomics Society of Australia.

Ramos, G., Balakrishnan, R., 2003. Fluid interaction techniques for the control and annotation of digital video. In: UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology. ACM Press, New York, NY, USA, pp. 105–114.

Ramos, G., Balakrishnan, R., 2005. Zliding: fluid zooming and sliding for high precision parameter manipulation. In: UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology. ACM Press, New York, NY, USA, pp. 143–152.

Ramos, G., Boulos, M., Balakrishnan, R., 2004. Pressure widgets. In: CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, New York, NY, USA, pp. 487–494.

Rekimoto, J., 1996. Tilting operations for small screen interfaces. In: UIST '96: Proceedings of the 9th annual ACM symposium on User interface software and technology. ACM Press, New York, NY, USA, pp. 167–168.

Rekimoto, J., 1997. Pick-and-drop: a direct manipulation technique for multiple computer environments. In: UIST '97: Proceedings of the 10th annual ACM symposium on User interface software and technology. ACM Press, New York, NY, USA, pp. 31–39.

Savage, J., Cockburn, A., 2005. Comparing automatic and manual zooming methods for acquiring off-screen targets. In: People and Computers XIX: British Computer Society Conference on Human Computer Interaction. pp. 439–454.

Schaffer, D., Zuo, Z., Greenberg, S., Bartram, L., Dill, J., Dubs, S., Roseman, M., 1996. Navigating hierarchically clustered networks through fisheye and full-zoom methods. ACM Trans. Comput.-Hum. Interact. 3 (2), 162–188.

Shneiderman, B., Plaisant, C., 2005. Designing the user interface, 4th Edition. Pearson/Addison-Wesley.

Small, D., Ishii, H., 1997. Design of spatially aware graspable displays. In: CHI '97: CHI '97 extended abstracts on Human factors in computing systems. ACM Press, New York, NY, USA, pp. 367–368.

Spence, R., 2007. Information Visualization, 2nd Edition. Pearson Education.

van Wijk, J. J., Nuij, W. A. A., 2003. Smooth and efficient zooming and panning. In: INFOVIS.

Yee, K.-P., 2003. Peephole displays: pen interaction on spatially aware handheld computers. In: CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, New York, NY, USA, pp. 1–8.

Yee, K.-P., 2004. Two-handed interaction on a tablet display. In: CHI '04: CHI '04 extended abstracts on Human factors in computing systems. ACM Press, New York, NY, USA, pp. 1493–1496.

Zeleznik, R., Miller, T., 2006. Fluid inking: augmenting the medium of freeform inking with gestures. In: GI ’06: Proceedings of the 2006 conference on Graphics interface. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, pp. 155–162.

Zhai, S., Smith, B. A., 1999. Multistream input: an experimental study of document scrolling methods. IBM Syst. J. 38 (4), 642–651.

Zhai, S., Smith, B. A., Selker, T., 1997. Improving browsing performance: A study of four input devices for scrolling and pointing tasks. In: INTERACT ’97: Proceedings of the IFIP TC13 International Conference on Human-Computer Interaction. Chapman & Hall, London, UK, UK, pp. 286–293.

Tables

Table 1
Influence of the two view sizes for semantic tasks.

	600x600 pixels	300x300 pixels	std. error	sig
task type 2	25.49 sec	33.05 sec	2.09 sec	0.008
task type 3	40.71 sec	49.26 sec	2.96 sec	0.025
task type 4	36.54 sec	44.73 sec	3.73 sec	0.0655

Table 2
The mean task-completion times for the tilt, pressure, and standard interfaces with respect to view size.

	standard	pressure	tilt
600x600 pixels	43.16 sec	41.67 sec	42.93 sec
300x300 pixels	43.15 sec	50.99 sec	48.87 sec

Table 3
The mean task-completion times for halo-navigation tasks.

	mean time	std. error
SDAZ	76.34 sec	3.12 sec
standard	83.9 sec	2.45 sec
pressure	94.8 sec	3.31 sec
tilt	98.66 sec	4.18 sec

Table 4
The mean task-completion times for the task types 2 to 4.

	mean time	std. error
SDAZ	41.47 sec	2.84 sec
standard	36.36 sec	1.84 sec
tilt	37.11 sec	1.78 sec
pressure	38.26 sec	2.53 sec

Table 5
The differences from the mean of the two view sizes for semantic tasks (higher values indicate that the interface was more influenced by the different view sizes).

	diff. from the mean	std. error
SDAZ	5.24 sec	2.79 sec
standard	0.98 sec	1.85 sec
pressure	6.13 sec	2.49 sec
tilt	3.85 sec	1.76 sec

Table 6
First-place votes out of 16 votes for each view regarding interface preference.

	standard	pressure	tilt	SDAZ
600x600 pixels	4 votes	5 votes	6 votes	1 votes
300x300 pixels	9 votes	4 votes	3 votes	0 vote

Table 7
Mean ranks for interface preference votes.

	mean rank	std. error
SDAZ	3.47	0.16
standard	1.94	0.16
pressure	2.28	0.18
tilt	2.34	0.19

Table 8
Preference votes across view sizes.

	standard	pressure	tilt	SDAZ
First place	13 votes	9 votes	9 votes	1 votes
Second place	9 votes	9 votes	8 votes	6 vote
Third place	9 votes	10 votes	10 votes	2 vote
Fourth place	1 votes	4 votes	5 votes	23 vote

Table 9
Subjective workload scores.

	workload score	std. error
SDAZ	52.93	3.79
standard	35.79	2.20
pressure	46.53	3.67
tilt	45.18	2.93