# Which One is Better? – Information Navigation Techniques for Spatially Aware Handheld Displays


Michael Rohs
Deutsche Telekom Laboratories, TU Berlin
Ernst-Reuter-Platz 7
10587 Berlin, Germany
michael.rohs@telekom.de

Georg Essl
Deutsche Telekom Laboratories, TU Berlin
Ernst-Reuter-Platz 7
10587 Berlin, Germany
georg.essl@telekom.de



## ABSTRACT

Information navigation techniques for handheld devices support interacting with large virtual spaces on small displays, for example finding targets on a large-scale map. Since only a small part of the virtual space can be shown on the screen at once, typical interfaces allow for scrolling and panning to reach off-screen content. Spatially aware handheld displays sense their position and orientation in physical space in order to provide a corresponding view in virtual space. We implemented various one-handed navigation techniques for camera-tracked spatially aware displays. The techniques are compared in a series of abstract selection tasks that require the investigation of different levels of detail. The tasks are relevant for interfaces that enable navigating large scale maps and finding contextual information on them. The results show that halo is significantly faster than other techniques. In complex situations zoom and halo show comparable performance. Surprisingly, the combination of halo and zooming is detrimental to user performance.



## Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*input devices and strategies, interaction styles*

## General Terms

Design, Experimentation, Human Factors

## Keywords

information navigation, navigation aids, small displays, spatially aware displays, spatial cognition, spatial interaction, handheld devices, camera phones


## 1. INTRODUCTION

Typical tasks with mobile devices include personal information management operations, like creating calendar entries and looking up phone numbers. Increasingly more complex operations are performed, like browsing the Web, booking theater tickets, and finding movies that are currently played at local cinemas. As a consequence of the scarce screen real-estate, current interfaces are often structured as deeply nested menu hierarchies.

Available interaction possibilities of mobile devices are not well suited to support complex tasks. Keypad-based input is difficult and requires more keystrokes per operation than PC-based interfaces, since multiple characters are mapped to a single key. Touch-screen input allows for more sophisticated input, but is still restricted because of the limited screen space.

We argue for externalizing the user interface into 3-D space in order to allow users to treat their handheld device as a window into a larger information space, which has a fixed position and orientation in physical space. This idea was originally proposed by Fitzmaurice as *spatially aware displays* [4, 5]. More recent work includes *peephole displays* [18], *mixed interaction spaces* [10, 11], as well as camera-based interfaces for two-handed input [8, 9].

An important class of applications for mobile devices concerns data that has an inherently spatial layout, such as map-based applications, or data that is conveniently viewed and interacted with in a 2-D (or 3-D) arrangement, such as calendar applications. Spatial data can best be viewed on larger areas and is difficult to deal with on the tiny screens of handheld devices. Current interfaces for mobile devices do not support these kinds of data very well. With the approach of spatial interaction for handheld devices this state can be improved.

A central problem of small screen interfaces is that a small physical display space represents a much larger virtual space. The user can only visualize a small area of the workspace at any instant in time. Several researchers have developed methods for navigating large-scale virtual workspaces for desktop-sized as well as for handheld devices. These include *halo* [1], zoomable and multiscale interfaces [2, 6, 7, 15], and hierarchically sub-segmented views [16].

We believe that these small-screen visualization techniques on the output side are complementary to and work well together with spatial input [12] on the input side. If objects in the virtual workspace have a fixed position with respect to a real-world reference frame, then users can use their spatial cognition skills for remembering the location of objects. A handheld device that behaves like a symbolic magnifying glass – moving closer to an object shows it at larger magnification and provides more detail – and acts as a window into a virtual space seems like an intuitive metaphor.



*ICMI'06,* November 2–4, 2006, Banff, Alberta, Canada.

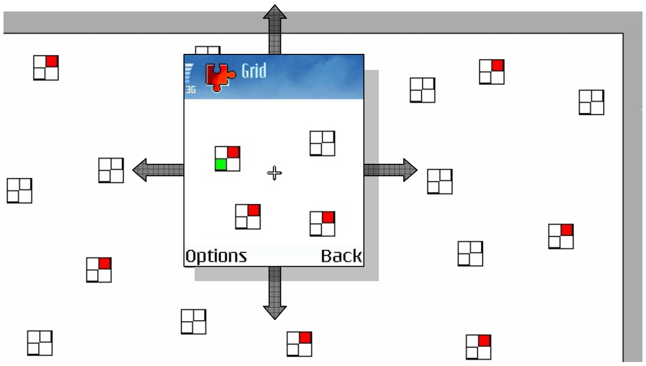



**Figure 1: Panning a movable window over a large fixed workspace. Only the portion in the movable window is visible to the user at a time.**

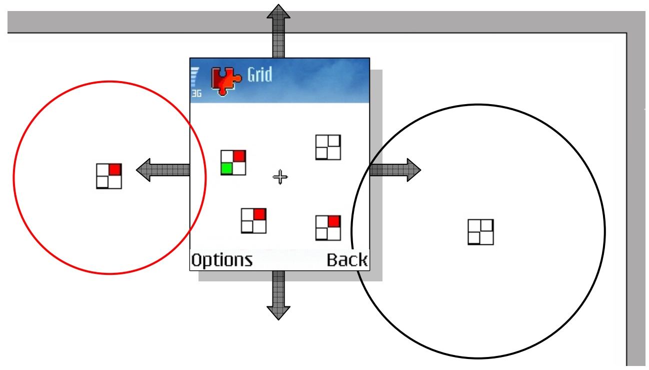



**Figure 2: Halos indicate off-screen objects. Red halos indicate target candidates.**

In order to evaluate these assumptions we implemented a spatially-aware handheld display on a Symbian phone. The integrated camera tracks a printed pattern of 26.7×18.5 cm to determine device position and orientation in 3-D. We compared user performance of an abstract map navigation task for different tasks complexities and different one-handed navigation and visualization techniques, including object halos, 2-D panning by horizontal movement, zooming by height change, and the combination of zooming and halos.

We found that halo-only is significantly faster than the other techniques. Halo is particularly useful if the number of target candidates is small. In more complex situations, zoom and halo show comparable performance. Surprisingly, for moderately complex situations the combination of halo and zooming leads to lower performance than panning alone.

In the following sections we describe the small-screen navigation techniques we compared in a user study and present the prototype implementation of our spatially aware display. We then report on the methods and results of the study. We conclude with general design recommendations for designers of small-screen spatially aware displays.

## 2. NAVIGATION TECHNIQUES FOR SMALL DISPLAYS

We implemented *pan*, *halo*, *zoom*, and the combination of halo and zoom (in the following called *halo&zoom*) for one-handed interaction with spatially aware displays in the context of an abstract map navigation application. Map navigation is a typical task for mobile small-screen devices. For city maps, this includes locating predefined points of interest, such as ATMs or restaurants. It also includes identifying features that are spatially extended over a certain area, like a park or a street junction. Extended features are typically not visually highlighted as a predefined point of interest. They are only recognizable in their local context and require a broader view of the area.

Fitzmaurice's spatially aware displays show 3-D scenes [4]. In [18] a 2-D version of a spatially aware display is presented, in which information is spread out on a flat surface that is larger than the display. The display is used as a movable window ("peephole") on the virtual space. The movement of the display is compensated by a corresponding movement of the virtual display in the opposite direction. If the user moves the display 5 cm to the right, the virtual view moves by the same amount. This creates the illusion of a virtual workspace that is fixed relative to the environment. The user can simultaneously navigate and interact with objects in the workspace. Whereas [18] implements two-handed interaction techniques for devices with stylus input, we focus on one-handed techniques. These are in line with current phone usage models, which assume that the user has only one hand free to operate the phone. The *pan* condition shown in Figure 1 corresponds to the peephole model.

*Halo* [1] visualizes off-screen objects by surrounding them with rings that reach into the border region of the display window (see Figure 2). From the curvature and position of the ring fragment that is visible on-screen, users can infer the position of the target at the ring center. Even if the visible arc is only a tiny fraction of the ring, it contains all the information needed to intuitively judge the approximate direction and distance of the target. The technique uses very little screen space and has been shown to have significantly shorter task completion times than arrow-based visualization techniques [1]. Whereas in [1] the halo technique was only evaluated in an emulation on a desktop computer, we implemented and evaluated it in the context of spatially aware displays on a Symbian phone.

The *hop (halo+proxy)* technique [14] improves halo-based navigation by displaying proxies of distant objects. The user is teleported to the object by clicking on the proxy. We did not include hop in this comparison for two reasons. First, it is unclear whether displaying multiple proxies scales down to very small display sizes. The display size of our test application is 176×144 pixels, at a physical size of 35×28 mm (target size 40×40 pixels). Hop was tested on a 17" monitor with a virtual size of 1280×1024 pixels (target size 32×32 pixels). Secondly, automatic teleporting to the target violates our assumption of a virtual workspace that has a fixed location in physical space. However, there might be alternative ways of teleporting to the target that maintain the user's physical awareness of the virtual space.

With *zoomable* or *multiscale* interfaces [2, 6, 7, 15] users can continuously adjust the scale at which they view virtual objects. In addition to standard cursor pointing to select objects, users have to perform *view pointing* [7] to navigate in scale and space to the target view that includes the object at the proper scale. Guiard and Beaudouin-Lafon [7] show that view pointing – and target acquisition in zoomable interfaces in general – obey Fitts' law [3].

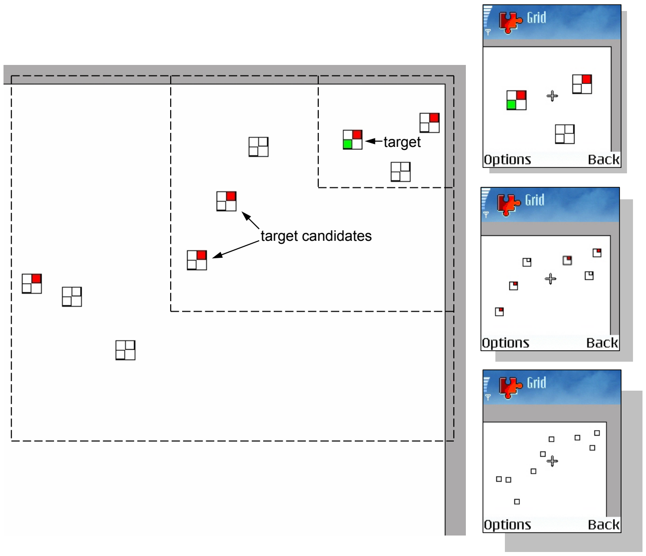



**Figure 3: Semantic zooming with three levels of detail. Continuous scaling within each level. The dashed lines indicate the respective areas that are visible in the screenshots.**

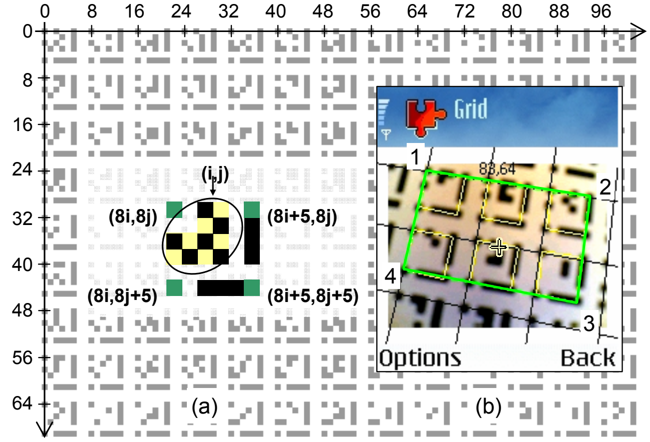



**Figure 4: The grid defines an absolute coordinate system. The camera phone computes the coordinates of the cross hair (83,64 in the screenshot), the distance from the grid surface, and the amount of rotation and tilting. (a) Computation of grid coordinates from index position stored in the marker's data area. (b) For robustness, the perspective mapping is computed from the maximum area of recognized markers.**

Zooming interfaces often provide overviews, also called radar views, showing a thumbnail outline of the items in the workspace and the location of the current detail view within the overall area. However, the results concerning the benefits of overviews in zoomable user interfaces are mixed [13]. Overviews help users in keeping track of their current position in the workspace. However, the spatial indirection between overview and detail view might strain memory and increase the time required for visual search. Moreover, the display size of our prototype implementation would only have allowed for a rudimentary overview. The focus was on providing a virtual workspace that has a fixed position with respect to the external physical space. Therefore, we decided not to include overviews in this comparison.

In [15] Perlin and Fox introduce the concept of *semantic zooming*. Beyond simply scaling objects to different magnifications, the representation of an object and the level of detail it shows changes with scale. A particular representation is associated with a scale range, within which that representation is linearly scaled. In a map application, a city might be represented by a dot and a name at small scale and by a street map at a higher scale. In the abstract navigation task we use three levels of detail (see Figure 3). At the smallest scale, all objects are drawn as squares. At medium scale, target candidates show a red mark. At high scale, all details are fully visible. The user has to zoom in to decide whether a target candidate is actually a target (red and green mark) or whether it is a false target (no green mark). Scale is continuously updated as a function of vertical distance. Moving down vertically (closer to the grid, see Section 3) increases scale, moving up decreases it.

## 3. CAMERA-BASED GRID TRACKING

To implement spatial awareness with low latency and high precision we use camera-equipped handheld devices that track their position and orientation relative to a grid pattern. The pattern consists of closely arranged visual markers, which are derived from visual codes [17]. The grid represents a large workspace, of which different parts can be accessed by simply placing the camera phone over the relevant area.

The grid defines a coordinate system that provides an absolute frame of reference for spatial interaction (see Figure 4). The size of the printed frame that we used in the comparison is 26.7×18.5 cm, which is roughly the size of a DIN A 4 sheet. The grid could of course also be printed on larger areas. The top left corner of the grid is the origin, the upper edge is the $x$-axis, and the left edge is the $y$-axis of the grid coordinate system. One coordinate unit corresponds to a single black-and-white cell. Each marker has a width and height of 6 cells. Markers are placed two coordinate units apart, which results in one marker for each 8×8 unit area of the grid. The left upper corner stones of each marker are placed at grid coordinates $(8x, 8y), x, y \in \{0, 1, 2, ..., 31\}$ (see Figure 4a).

The markers have a layout similar to visual codes [17], but consist of only two corner stones and two guide bars and a smaller data area. The small size of the markers and their dense layout on the surface ensure that there is always a marker contained in the camera image, even at close distances from the grid. The data area of a single marker has a raw capacity of 12 bits. It is used to store the $x$ index (5 bits) and the $y$ index (5 bits) of the marker within the grid, as well as two parity bits. A grid can thus have a maximum size of 32×32 markers, which is equivalent to 256×256 coordinate units.

In our current implementation on Symbian camera phones with a resolution in view finder mode of 160×120 pixels and with a printed size of 6.67 grid units per cm (i.e. the size of a single marker is about 9×9 mm), the grid is detectable at distances between 20 mm and 100 mm from grid surface to camera lens. This results in a 3D interaction space of

$length \times width \times height = 37.6 \times 37.6 \times 8$ cm, if the maximum of 32×32 markers are used. In this space, we can precisely determine the position and orientation of the phone at a rate of up to 10 frames per second with our prototype device (a Nokia 6630 Symbian phone), depending on the complexity of the rendered virtual workspace. In particular, the focus point on the grid surface can be tracked with high precision.

For robustness, multiple markers are combined to compute the perspective mapping between corner points of the code and corresponding grid coordinates (see Figure 4b). We need four corresponding pairs of (triple-wise non-collinear) points to establish a perspective mapping between the image coordinate system and the grid coordinate system. Once the markers are recognized, the image pixel coordinates of their corner stones and guide bars are known. This is indicated by the yellow frames around each recognized marker in Figure 4b. (The camera image is not shown during normal use.) The closer these points lie together, the less accurate the perspective mapping will be. However, at medium distances, multiple markers are present in the camera image. Hence, we use the largest possible area to get the most accurate mapping. In the frame shown in Figure 4b, 6 markers have been detected. They are highlighted by yellow frames. Instead of basing the perspective mapping on any of the markers' corner points, the points (1) to (4) are used, which represent elements of different markers. For these corner points, of which we know the image coordinates from the marker recognition step, we establish correspondences as follows (see Figure 4a). For an upper left corner, its grid coordinates are $(8i_1, 8j_1)$, where $i_1$ and $j_1$ are the horizontal and vertical indices that are stored in the marker. The grid coordinates of an upper right corner are $(8i_2 + 5, 8j_2)$, of a lower right corner $(8i_3 + 5, 8j_3 + 5)$, and of a lower left corner $(8i_4, 8j_4 + 5)$. Again, $i_k$ and $j_k$, $k \in \{1, 2, 3, 4\}$, are the vertical and horizontal index values that are stored in the respective code. From these correspondences, we can compute a perspective mapping (a planar homography) between image coordinate system and grid coordinate system.

In order to provide a virtual view that is fixed in space, the physical size of a pixel needs to be known. Our prototype device has a physical screen width of 35 mm and a virtual screen width of 176 pixels, i.e. about 5 pixels/mm. With a grid size of 1.5 mm/ccu (code coordinate unit), this results in 7.543 pixels/ccu. The mapping between grid coordinates and workspace pixel coordinates can simply be expressed as

$$(x, y)_{pixel} = 7.543(x, y)_{ccu}.$$

In the zooming interface, we use the vertical distance measure of the marker recognition algorithm as the z-coordinate. The size of displayed objects is continuously adjusted to height. Moreover, the level of detail changes in three ranges as follows:

| | | |
|---|---|---|
| too close | <30 mm | maximum scale, gray background |
| level 1 | 30-41 mm | full detail, target distinguishable |
| level 2 | 41-62 mm | medium detail, candidates visible |
| level 3 | 62-80 mm | objects visible, no details |
| too far | >80 mm | minimum scale, gray background |

If the user moves out of the height range, the background color changes from white to gray. This helps the user to adapt to the boundaries of the interaction space. The same is true for panning. If the user moves out of the range in horizontal direction, a gray frame is drawn that indicates the border of the workspace (see the screenshots in Figure 3 that focus on the upper right corner of the workspace).

Zooming out offers a bird's eye view of the workspace: more context is visible on the screen, but at a lower level of detail. In contrast to other zooming interfaces, the movement distances to a target remain constant in physical space. This is compatible with physical reality. In other zooming interfaces, panning in zoomed-out view means moving at much higher speed in virtual space. We chose a different strategy since we wanted to maintain the fixed-workspace metaphor.

# 4. USER STUDY

We conducted a user study to compare the discussed small display navigation techniques in the context of spatially aware displays.

## 4.1 Participants and Apparatus

The study was conducted on 9 participants, 5 female, 4 male, age 23-32. Subjects were doctoral students or post-doctoral researchers with technical background at the Technische Universität Berlin as well as 2 design students.

Subjects performed the test on a Nokia 6630 Symbian phone. During each trial, the $(x, y, z)$ coordinates of the motion trajectory were sampled at a rate of 10 updates per second. The time to target for each trial was also recorded.

## 4.2 Tasks

Users successively had to find a target among a number of distractors in the workspace (see Figures 1-3). There was always one target present in the workspace at a time, which was indicated by a green square. Once found and clicked, the next target appeared at a different place in the workspace. 10 trials were performed for each arrangement of distractors and combination of navigation techniques. In addition to the target 0, 16, or 32 distractors were generated. If the navigation technique allowed revealing of proximity information, either 25% or 50% of the distractors appeared as target candidates.

For each of these settings a number of navigational options were offered:

- Panning only
- Halo only
- Zoom only
- Halo&zoom combined

Panning is a standard flat navigation method where one navigates close up zoom moving in the plane until one finds the target in one's display area. The physical area available for panning was 22.9×18.7 cm in which a virtual workspace of 1152×942 pixels was embedded. As a basic mechanism, panning was available in all conditions.

Halo, as discussed earlier, provides circular arcs, which are guidance for existing objects.

Zoom allows the user to continuously move in and out of level of detail by using distance to the plane. Our implementation used three zoom regions for three levels of detail. The minimum height for sensible registration was set to 30 mm. If a lower value was found, the display background became gray to indicate to the user that the detection became problematic. Same holds for the maximum

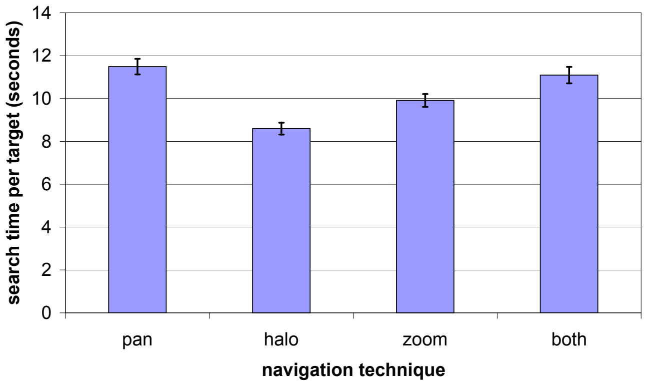



**Figure 5: Comparison results of the various navigational techniques averaged over all participants. Error bars show one standard error.**

height value of 80 mm. The level changes occur at values of 41 mm and 62 mm. At the highest levels, all objects, whether valid or distractors look alike. Entering a zoom level reveals a certain percentage of objects as possible candidates. Only at the closest zoom level can distractors and the target be completely differentiated.

Halos and zooming were both present in the combined halo& zoom setting.

## 4.3 Design

The study was designed as a within-participants factorial design with three factors:

- navigation technique: pan, halo, zoom, halo&zoom
- distractor count: 0, 16, 32
- candidate ratio: 25%, 50%

This results in $4 \times 2 \times 2 + 4 = 20$ conditions. The orders of conditions were all randomized beforehand and were distinct for each participant. The test application recorded the movement trajectories and trial completion times of 9 users $\times$ 20 conditions/user $\times$ 10 trials/condition = 1800 trials.

The distance from one target to the next was always 600 pixels. Participants were not informed about this fact. The target size was 40 pixels. Thus each trial had an index of difficulty [3] of $ID = \log_2(600/40 + 1) = 4$.

## 4.4 Procedure

Prior to recording actual test data users performed a number of practice trials, until they felt familiar with each navigation method.

Before each new condition, the user was informed about the navigational methods provided. Hence the user would see one of the following texts on the screen: "Next: <method> <x>/20, <n> objects", where <method> could be any of "pan", "pan & zoom", "pan & halo", or "pan & zoom & halo". <x> would be the current task starting at 1 and incrementally increasing to 20. <n> is the number of objects (distractors and the target) for this task. When the participants were ready, they clicked the joystick button to start the next condition, consisting of finding 10 individual targets in sequence.

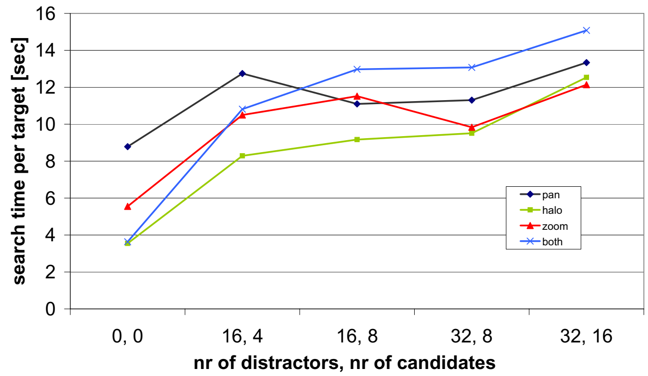



**Figure 6: Comparison results of the various navigational techniques averaged over all participants, grouped by distractor configurations (increasing numbers of false candidates).**

Then the navigational display would appear and the user could start searching for targets given the navigational features available. After a target was successfully selected by clicking the mobile phone joystick button, a new target appeared and the task continued until 10 targets were found.

The order of the configurations (number of distractors, level of distractors revealed with navigational techniques, type of navigational technique) were all randomized beforehand and distinct for each participant.

## 4.5 Results

Figure 5 shows the overall result of the information navigation techniques. The figure is the combined average across all subjects and across all distractor configurations. One can see that in general halo alone performs best followed by zoom. Joint presentation of halo and zoom performs just barely better than panning. The numerical values can be seen in Table 1. Table 2 shows the performance ratios between the individual techniques. On average, halo is 25% faster than pan.

| | pan | halo | zoom | halo&zoom |
|---|---|---|---|---|
| mean | 11.49 | 8.60 | 9.91 | 11.09 |
| stdev | 7.18 | 5.51 | 5.93 | 7.59 |

**Table 1: Numerical values of the accumulated task times for the different navigation techniques.**

| | pan | halo | zoom | both |
|---|---|---|---|---|
| pan | | 25% | 14% | *3%* |
| halo | | | 13% | 22% |
| zoom | | | | 11% |
| both | | | | |

**Table 2: Performance differences between the individual techniques in percent, based on the times in Table 1.**

Prior to significance testing we looked at individual conditions and removed trials with completion times greater than 2 standard deviations from the mean. This led to a removal of about 3% of the trials as outliers. The histograms in Figure 7 show that trial completion times are not normally distributed, but rather appear log-normally distributed. ANOVA on log trial completion times revealed a significant effect of navigation technique ($F(3,1562) = 15.65$, $p = 4.98E\text{-}10$). In order not to rely on any assumption on the distribution of the data, we used the Kolmogorov-Smirnov test for pairwise difference significance testing under the hypothesis that each pair is the same. The results can be found in Table 3. We conclude that all differences between results of Figure 5 are significant except for the difference between panning and halo&zoom.

| | pan | halo | zoom | both |
|---|---|---|---|---|
| pan | | 0.000 | 0.020 | *0.203* |
| halo | | | 0.004 | 0.000 |
| zoom | | | | 0.041 |
| both | | | | |

**Table 3: K-S test significances for the differences between the times in Table 1. The difference between pan and halo&zoom is not significant.**

Figure 6 shows the performance per navigational technique, averaged over all users and grouped by distractor configuration. One sees that generally halo along performs best given our test conditions. Panning performs generally worst, especially with a low number of distractors. Somewhat surprisingly, a mixed presentation of halo and zooming does increase performance over both halo and zooming methods alone. Zooming performs worse than halo in our test configuration.

When no distractors are present, halo and combined techniques perform best. Halo remains the best navigation technique until the number of distractors and target candidates becomes high. With 32 distractors and 8 or 16 candidates, zooming and halo have comparable performance. This effect is worth examining in more detail with $> 32$ distractors. It might turn out that zooming is in fact superior to halo for large number of candidates and distractors.

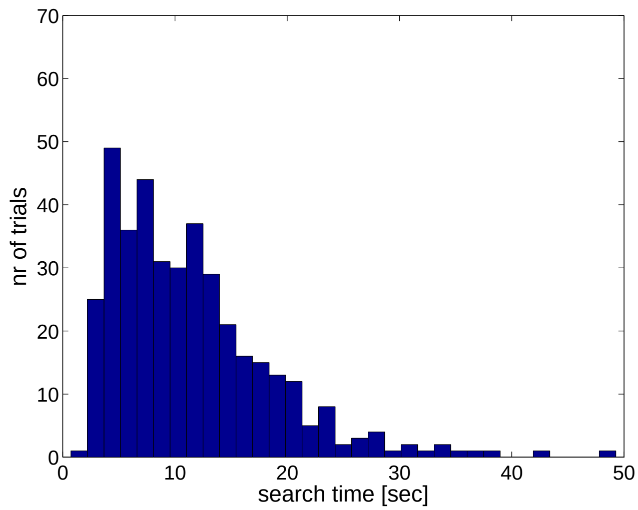
Surprisingly, the combined display does not show comparable performance to zoom or halo alone and rapidly deteriorates in performance with increased distractors. It is also interesting to note that the increase of distractors from 16 to 32 has little effect on performance when the number of target candidates is held constant at 8, except for zoom which shows a surprising performance increase with increasing numbers of distractors. We have no intuition for this effect. There should be further analysis of this point with additional experiments.

The poor performance of the combined display is somewhat puzzling. Maybe the joined display of different navigation strategies was more distracting than helpful.

## 4.6 Subjective Feedback

We asked the participants for their subjective preference for navigational techniques. We also asked them to indicate if they perceived short-comings in a method.

Halo was generally judged most favorably, especially in the case of few target candidates. Some users noted that halo was confusing with many objects and that they liked zoom better for the more complex distractor configurations. One participant found that zoom scales better to a large numbers of objects. Two participants remarked that the



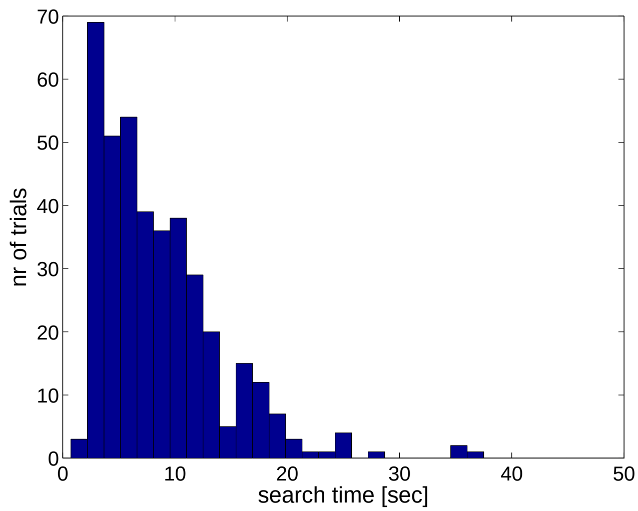
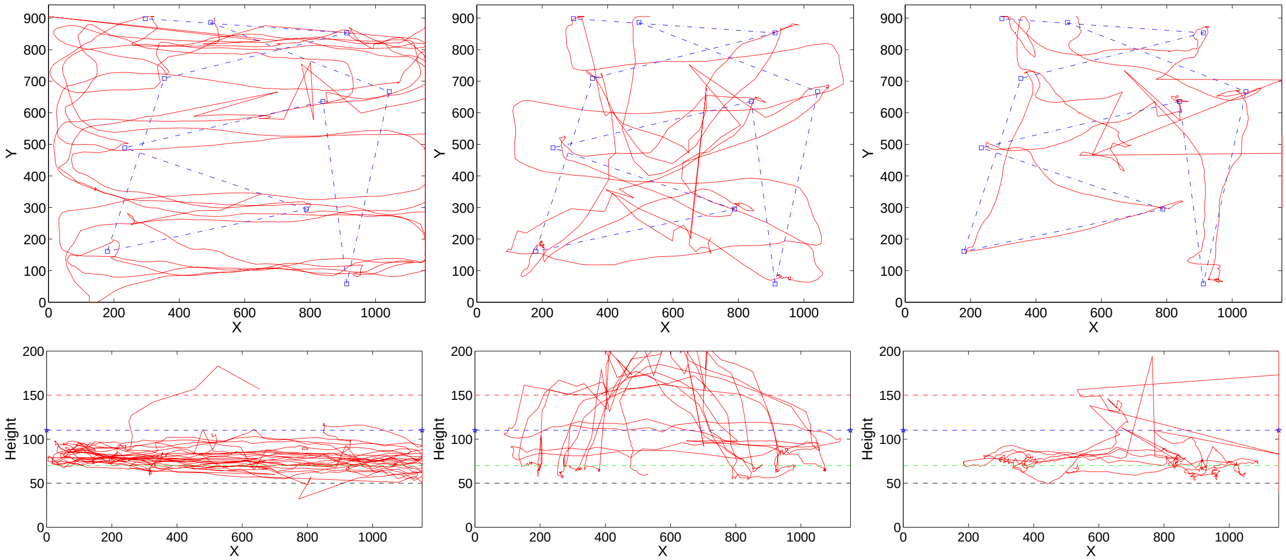




**Figure 7: Histograms of averaged task performance times for navigational technique (top) panning (bottom) halo. Other techniques have qualitatively similar distributions.**

range of vertical motion available for zooming was too limited. In the zooming condition, one user disliked the changing appearance of the objects when level-of-detail boundaries were crossed, since this triggered a sudden change in the appearance of the objects in the workspace. Pan was consistently rated last. It was particularly frustrating to locate a single target in the no distractor case, since there was few visual feedback. One user mentioned the low update rate of the display (10 updates per second) and the corresponding lag in the interface.

A comparison between the subject's ranking for preference and their performance in task time can be found in Table 4. One can see that while often the rough trend between preference and performance is followed, this is not always the case, and participants prefer methods, which actually perform worse. For example, participants 6, 8, and 9 prefer mixed techniques, while performing best with just zoom or just halo.




(a) (b) (c)

**Figure 8: Search patterns by one participant for (a) panning (b) zoom (c) halo. Top view shows the planar movement in the X-Y-plane. Bottom view shows the height profile in the X-Z-plane. The dashed line is the ideal path between targets. The solid line trajectories are the actual motion of the participant. These trials had no distractors.**

| User | | pan | halo | zoom | halo&zoom |
|---|---|---|---|---|---|
| 1 | Ranking | 4 | 1 | 2 | 3 |
| | Time | 12.25 | 8.52 | 9.65 | 10.05 |
| 2 | Ranking | 4 | 1 | 2 | 3 |
| | Time | 11.51 | 9.51 | 9.41 | 9.09 |
| 3 | Ranking | 4 | 3 | 1 | 2 |
| | Time | 13.55 | 9.89 | 9.53 | 11.22 |
| 4 | Ranking | 4 | 1 | 2 | 3 |
| | Time | 8.28 | 7.45 | 10.60 | 10.43 |
| 5 | Ranking | 4 | 1 | 2 | 3 |
| | Time | 12.94 | 8.62 | 12.71 | 14.50 |
| 6 | Ranking | 4 | 3 | 2 | 1 |
| | Time | 10.14 | 8.07 | 7.71 | 9.70 |
| 7 | Ranking | 1 | 3 | 2 | 4 |
| | Time | 13.29 | 10.35 | 12.69 | 13.41 |
| 8 | Ranking | 4 | 2 | 3 | 1 |
| | Time | 11.11 | 7.90 | 7.59 | 10.48 |
| 9 | Ranking | 4 | 2 | 3 | 1 |
| | Time | 9.60 | 6.82 | 9.11 | 11.08 |

**Table 4: Comparison of subjective preference ranking and task time of participants for different navigational techniques.**

## 4.7 Motion and Search Strategies

Subjects used an array of search strategies as is evident from their recorded motion patterns. Participants used a variety of motion patterns. Here we want to illustrate one example, which is depicted in Figure 8. These are all trials of a single participant with no distractors present, hence constitute the easiest search task. The four horizontal lines in the height profiles mark the level-of-detail borders: Below height 70 full details are visible, from 70 to 110 target candidates are marked with a red dot, and above 110 all objects look the same. The height values are provided by the marker recognition system.

One sees that for panning the participant chose a left-right sweep motion to find the target. Comparing Figure 8 (a) and (b) one sees that the path for zoom is less dense than panning. Zoom is accompanied by a repeated moving in and out, as is evident from the height profile. A hint of a coarse sweeping motion can also be found for zooming, but is less prevalent than in the panning case.

For halo, the motion in the plane depicted in Figure 8 (c) is rather close to the ideal path indicating that the presence of a single halo guided the participant well toward the target.

It is worthwhile noting that search strategies were not uniform across participants, and that some used circular or irregular motion patterns.

# 5. CONCLUSIONS

The general aim of the paper was to investigate and compare navigation techniques for small-screen interfaces with small-scale spatial awareness in order to identify their relative benefits and issues. Our results show a significant difference in performance between different navigation techniques. Halos perform very well for a range of configurations, most reliably with low numbers of distractors. Zoom becomes competitive as the number of distractors increases. Combining these two navigation techniques is detrimental to performance and should be avoided. Panning alone is generally worse than either of the other techniques.

The reason for the poor performance of the combined halo&zoom interface has to be investigated further. Presumably in this case, users concentrate either on one of the navigational aids and are not able to effectively control the other at the same time. Another reason might be that users need time to decide which navigational aid to use during the task.

However, in real-world map navigation tasks and other interfaces, both navigation techniques are useful; halo for showing predefined points of interest, zoom for inspecting context at freely chosen levels of detail. The results show that both techniques can be effectively implemented using spatially aware displays, even at very small display sizes.

The vertical movement range – and hence the parameter space for zooming – is limited by the low resolution camera. It remains to be seen whether the performance of zooming increases if the vertical movement range is larger than in our current prototype. Technical development is heading for better camera hardware in mobile devices, including higher resolution, higher frame rates, autofocus, and optical zoom.

Implementing spatial awareness with a printed grid pattern is well suited for stationary operation. It could widen the scope of plausible stationary applications for handheld devices, like more easily managing one's calendar, address book, or music collection, without requiring a PC. To support mobility, an approach followed in [9] is to print the pattern on a portable piece of cardboard. More advanced image processing techniques can operate without a grid pattern, which would be more appropriate to mobile operation. However, they require more processing power and are potentially less robust. Alternative technologies for implementing spatial awareness for handheld devices would require lightweight, fine-grained, low-delay, absolute location sensing.

## 6. REFERENCES


[1] P. Baudisch and R. Rosenholtz. Halo: A technique for visualizing off-screen objects. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 481–488, New York, NY, USA, 2003. ACM Press.

[2] B. B. Bederson and J. D. Hollan. 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*, pages 17–26, New York, NY, USA, 1994. ACM Press.

[3] P. M. Fitts. The information capacity of the human motor-system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381–391, 1954. (reprinted in J. Exp. Psychol.: General, September 1992, 121(3), 262-269).

[4] G. Fitzmaurice and W. Buxton. The Chameleon: Spatially aware palmtop computers. In *CHI '94: Conference companion on Human factors in computing systems*, pages 451–452. ACM Press, 1994.

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

[6] G. W. Furnas and B. B. Bederson. Space-scale diagrams: Understanding multiscale interfaces. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 234–241, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

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

[8] M. Hachet, J. Pouderoux, and P. Guitton. A camera-based interface for interaction with mobile handheld computers. In *SI3D '05: Proceedings of the 2005 symposium on Interactive 3D graphics and games*, pages 65–72, New York, NY, USA, 2005. ACM Press.

[9] M. Hachet, J. Pouderoux, P. Guitton, and J.-C. Gonzato. Tangimap: A tangible interface for visualization of large documents on handheld computers. In *GI '05: Proceedings of the 2005 Conference on Graphics Interface*, pages 9–15, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005. Canadian Human-Computer Communications Society.

[10] T. R. Hansen, E. Eriksson, and A. Lykke-Olesen. Mixed interaction space: Designing for camera based interaction with mobile devices. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1933–1936, New York, NY, USA, 2005. ACM Press.

[11] T. R. Hansen, E. Eriksson, and A. Lykke-Olesen. Mixed interaction spaces – a new interaction technique for mobile devices. Demonstration at UbiComp, 2005.

[12] K. Hinckley, R. Pausch, J. C. Goble, and N. F. Kassell. A survey of design issues in spatial input. In *UIST '94: Proceedings of the 7th annual ACM symposium on User interface software and technology*, pages 213–222, New York, NY, USA, 1994. ACM Press.

[13] K. Hornbaek, B. B. Bederson, and C. Plaisant. Navigation patterns and usability of zoomable user interfaces with and without an overview. *ACM Trans. Comput.-Hum. Interact.*, 9(4):362–389, 2002.

[14] P. Irani, C. Gutwin, and X. D. Yang. Improving selection of off-screen targets with hopping. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 299–308, New York, NY, USA, 2006. ACM Press.

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

[16] D. C. Robbins, E. Cutrell, R. Sarin, and E. Horvitz. ZoneZoom: Map navigation for smartphones with recursive view segmentation. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 231–234, New York, NY, USA, 2004. ACM Press.

[17] M. Rohs. Real-world interaction with camera phones. In H. Murakami, H. Nakashima, H. Tokuda, and M. Yasumura, editors, *Second International Symposium on Ubiquitous Computing Systems (UCS 2004), Revised Selected Papers*, pages 74–89, Tokyo, Japan, July 2005. LNCS 3598, Springer.

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