Hindawi Publishing Corporation
Advances in Human-Computer Interaction
Volume 2008, Article ID 450385, 11 pages
doi:10.1155/2008/450385

Research Article

Sensing-Based Interaction for Information Navigation on Handheld Displays

Michael Rohs and Georg Essl

Deutsche Telekom Laboratories, TU Berlin, Ernst-Reuter-Platz 7, 10587 Berlin, Germany

Correspondence should be addressed to Michael Rohs, michael.rohs@telekom.de

Received 1 December 2007; Accepted 14 July 2008

Recommended by Luca Chittaro

Information navigation on handheld displays is characterized by the small display dimensions and limited input capabilities of today's mobile devices. Special strategies are required to help users navigate to off-screen content and develop awareness of spatial layouts despite the small display. Yet, handheld devices offer interaction possibilities that desktop computers do not. Handheld devices can easily be moved in space and used as a movable window into a large virtual workspace. We investigate different information navigation methods for small-scale handheld displays using a range of sensor technologies for spatial tracking. We compare user performance in an abstract map navigation task and discuss the tradeoffs of the different sensor and visualization techniques.

1. Introduction

This work is concerned with finding sensor technologies and visualization techniques for one-handed interaction with spatially aware handheld displays. For this purpose, we focus on an abstract map navigation application as a template for general large area scanning and searching tasks. Map navigation is a typical task for mobile small-screen devices. For city maps, for example, this includes locating predefined points of interest, such as ATMs or restaurants; it also includes locating embedded and spatially extended features, like parks or street junctions, which are only recognizable in their local context. Furthermore, the techniques studied here have applications in mobile interactions with physical maps [1]. They can also be viewed as a way to allow spatial editing, as was demonstrated in the CaMus project for musical scoring [2]. Certainly, these techniques are suitable for augmented reality applications and gaming. In all these applications, suitable interaction techniques are necessary for effective navigation in the respective information spaces.

With the integration of sensors, like accelerometers, or by using the built-in cameras of many mobile phones, the user can perform a wider range of gestures. Sensing explicit user actions is part of the more general concept of sensing-based interaction [3]. Sensing capabilities have been used for some time in mobile devices to enable new kinds of user interfaces [4–6]. Early work in applying sensor technology to handheld displays are Fitzmaurice and Buxton [7] spatially aware displays. Hinckley et al. [6] describe sensing techniques for mobile interaction by exploring a range of sensors in mobile devices.

We present two studies using various navigational options and sensor technologies. These are dynamic peephole [8] navigation alone, with halo [9], with zoom, and with both halo and zoom combined. As sensor technologies, we use camera-based tracking on a marker grid, optical motion estimation, and accelerometer and magnetometer readings for tilt and rotation detection. The studies show the performance tradeoffs of the combinations of sensor technology and navigational aid for a range of task complexities.

2. Sensor Technologies

We consider three types of sensor technologies. The first uses accelerometers and magnetometers in an integrated unit (henceforth *shake*). The second uses the built-in camera of

the mobile phone to track a marker grid (*grid*) and the last tracks optical motion in the camera image (*flow*).

2.1. Accelerometer and Magnetometer

A range of sensors for detecting aspects of motion are available. In particular accelerometers are becoming more widespread in commodity mobile devices. Their main advantage is that they are very cheap, come as small IC units, and are already showing up in commodity hardware. For example, the Nokia 5500 mobile camera phone (http://www.forum.nokia.com/devices/5500) contains a 3-axis accelerometer. The Wii game console (http://www.wii.com/, http://www.en.wikipedia.org/wiki/Wii) uses the same technology. Apple's iPhone (http://www.apple.com/iphone/features/accelerometer.html) contains accelerometers to automatically align the screen depending on the direction the device is held. The main disadvantage is the lack of a stable reference frame. Continued integration of acceleration to get velocity and then displacement also integrates all noise of the sensor and leads to inevitable drift. Hence accelerometers cannot easily be used for absolute motion in all directions. The earth's gravitational field does provide an external reference, and rotation relative to the gravitational field can be measured reliably. We used this sensing of tilt with respect to the earth's magnetic field to give the user of a mobile device the ability to navigate up and down by tilting the device forward and backward. The inclination of a 3-axis accelerometer can be computed as the angle between gravity and the accelerometer's z-axis.

A somewhat less widespread sensing technology is magnetic field sensors, called magnetometers. These also come as small integrated units and are fairly easily accessible. These also start to become available, for example, in Nokia 5140 mobile phones for use as a compass. They are reasonably cheap and once calibrated, rather accurate, as long as the immediate environment does not have very strong electromagnetic interference (EMI).

To obtain the compass heading independent of inclination, the readings from a 3-axis magnetometer and a 3-axis accelerometer can be combined, if they are mounted in the same frame of reference. The accelerometer provides a vector A in the direction of gravity, the magnetometer provides a vector M towards north with an inclination depending on the geographic location. The cross products $V = A \times M$ and $W = A \times V$ produce vectors in the horizontal plane, oriented towards east and north, respectively. The system A, V, W is orthogonal. The projection of the x-axis in the horizontal plane is $P = (ka_x + 1, ka_y, ka_z)$ with $k = a_z/({a_x}^2 + {a_y}^2 + {a_z}^2)$. The angle in the horizontal plane between P and V is $\alpha = \arccos((p_x v_x + p_y v_y + p_z v_z)/(\|P\| \|V\|))$, analogously for the angle between P and W.

For the purpose of our study, we initially looked at two types of devices providing such sensor capabilities: Nokia 5500 camera phones and the SHAKE device. (http://www.samh-engineering.com/.) The first is a commodity phone with a built-in 3-axis accelerometer. The latter is a small device designed to incorporate a range of high-fidelity sensors for rapid prototyping of mobile interactions. We noted a factor of over 70 in error comparing the SHAKE accelerometers and the data from the Nokia 5500.

2.2. Grid Tracking with Extended Range

In the first optical tracking technique, we considered (henceforth *grid*) the camera phone is tracked above a grid of visual markers. The grid provides a fixed frame of reference for the virtual workspace in which the user interacts. The absolute position of the device within the physical space above the grid is tracked with low latency and high precision. Grid tracking can precisely sense very subtle movements. However, the grid has to be present in the camera view, which limits user mobility.

The approach discussed here is an extension to the one described in [10]. The markers have been extended to a capacity of 16 bits: 2×7 bits for index positions and 2 parity bits. The maximum grid size is thus 128×128 markers or 1024×1024 code coordinate units (ccu). Suitable printing sizes are 1.5–2.0 mm per black-and-white cell, which yields a maximum grid area of 1.54–2.05 m.

In the original implementation, the tracking range (the distance of the camera lens to the grid surface) was limited to between 2 and 10 cm. This proved insufficient for effective interactions along the z-dimension. In particular, the range was too small for mapping to the zoom scale, because slight distance changes resulted in very rapid zoom scale changes. In the current extension, we use the digital zoom feature that is present in many camera phones to substantially extend the vertical tracking range. Digital zoom increases the apparent focal length at which an image was taken by cropping an area at the image center with the same aspect ratio as the original image. The cropped area is rescaled to the original dimensions by interpolation. Digital zoom is done by the camera hardware and hence does not put load on the main processor of the device.

The Symbian camera API allows to set the digital zoom level between 0 and some device-dependent maximum value. In an experiment, we kept the distance to an object in the camera view constant, continuously changed the digital zoom level, and measured the size at which the object appeared in the camera view ($size_{zoomed}$). We found a good fit of the measured data to $size_{zoomed} = size_{unzoomed} \exp(k \text{ level})$, or equivalently $distance_{zoomed} = distance_{unzoomed} \exp(-k \text{ level})$. For a few devices, the constant k was determined. For Nokia the 6630 ($6 \times$ digital zoom) $k = 0.0347$ ($R^2 = 0.998$), for the Nokia N70 ($20 \times$ digital zoom) $k = 0.0386$ ($R^2 = 0.999$), and for the Nokia N80 ($20 \times$ digital zoom) $k = 0.0345$ ($R^2 = 0.997$). This can be done in a one-time setup procedure. With this constant and the above formula, the unzoomed distance can be computed given the current zoom level.

During grid tracking, digital zoom is continuously adjusted, such that the markers appear at a size that is best suited for detection. If no markers are detected in a camera frame, a different zoom level is tried. The algorithm is complicated by the fact that changes to the zoom level do not come into effect immediately. Instead, the new digital zoom setting becomes valid with a delay of 2 to 5 frames after the

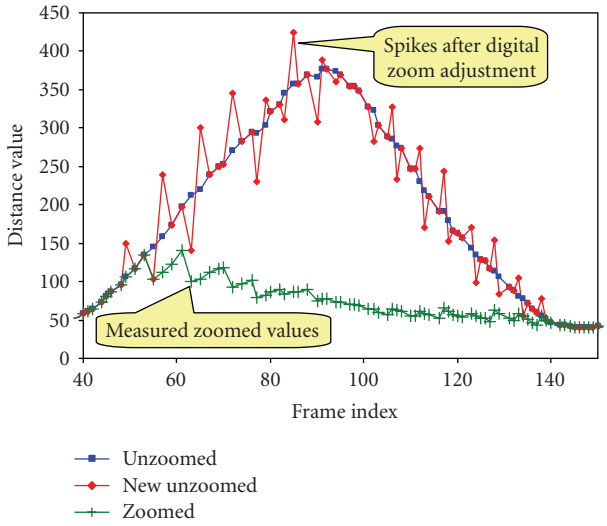

Figure 1: Digital zoom is continuously adjusted. New settings become valid after a delay of a few frames. The unzoomed value is chosen such that the resulting distance curve is smooth.

adjustment is made. Therefore, the algorithm computes the unzoomed distance at the old and the new digital zoom level and chooses the setting that yields the smoothest distance curve (Figure 1).

With this method, the vertical recognition range for a grid with a cell size of 1.5 mm is increased from 10 cm to 30–50 cm, depending on the device. In view finder mode with a frame size of 176 × 144 pixels, markers are recognized at a rate of 13–15 frames per second.

2.3. Optical Movement Detection

The optical movement detection algorithm determines device movement relative to the static background (henceforth *flow*). It provides linear movement in the x- and y-directions in the display plane. The algorithm subdivides video frames into block images, computes cross-correlations between successive pairs of block images for a range of different shift offsets, and looks for the maximum correlation. The correlation is based on blocks instead of pixels in order to make the method computationally feasible.

Each block has a size of 8 × 8 pixels. With a video resolution of 176 × 144 pixels in our case, the block image has 22 × 18 blocks. From each block, four pixel samples are taken at pixel positions (x, y), $x, y \in \{1, 5\}$. Just $4 \times 22 \times 18 = 1584$ pixels are sampled in each frame, which is only 6.25% of the pixel data. There is no need for grayscaling if the video stream of the camera has a planar YUV 4:2:0 format with 8 bits per pixel. The Y components already represent the luminance (grayscale) pixel data on which the sampling operates. The average gray value for each block is computed and centered at 0, that is, the resulting average gray values are in the range $\{-128, \ldots, 127\}$.

Current camera phones typically have a video stream frame rate of 15 fps. Since at 15 fps two temporally adjacent block images are only $\Delta t = 67$ milliseconds apart, we assume that there is considerable overlap between them—if movement is not too fast. To compute the spatial displacement between two block images, we use cross-correlation as a matching function. It determines which displacement to shift one block image against the other results in the best match. The cross-correlation function (between block images b_1 and b_2) is defined as

$$r(dx, dy) = \frac{\sum_{y=0}^{h-1} \sum_{x=0}^{w-1} b_1(x, y) b_2(x + dx, y + dy)}{(w - |dx|)(h - |dy|)}. \quad (1)$$

The denominator normalizes the cross-correlation to the size of the overlapping area. r is evaluated at 81 points for $dx, dy \in \{-4, \ldots, 4\}$. The most likely relative linear movement $(\Delta x, \Delta y)$ is a point at which r has a maximum.

If the displacement was scaled by the magnification of the camera view, the real movement velocity could be computed. However, since the scaling factor depends on unknown camera parameters and the variable distance of the camera to the background, no scaling is performed. As a result, the computed relative movement depends on the distance of the camera to the background, which is not a problem for the envisaged interactions.

In order to suppress spurious movements and shakes, movement updates are only reported if the signal-to-noise ratio—the maximum correlation relative to the mean correlation—is above a predetermined threshold. With this measure, the algorithm works quite reliably on a wide range of everyday backgrounds and detects relative motion even if the sampled backgrounds are not richly textured. However, on uniform gray or white backgrounds, the algorithm performs poorly. Because only a few pixels are sampled, the algorithm performs quickly. On current devices, it runs at the full frame rate with 15 updates per second.

Since the movement detection scheme is relative, drift is unavoidable. Particularly, when the user makes fast movements, the overlap between successive images is not sufficient and relative movement cannot be computed. The center of the phone's 5-way direction key is used as a clutch, which fixes the workspace on the screen and allows the user to reposit her arm. This mechanism is similar to lifting the mouse from the table.

2.4. Sensors in the Task Context

Multiple grids were printed on a DIN A4 sheet and attached to the wall at different heights. The height best suited to individual body height could be chosen by each user during initial test trials. In the experiment, we used a grid of size 27 × 18 cm and a tracking range of 50 to 285 mm. The height ranges of the zoom levels were set to 50–117 mm, 117–184 mm, and 184–285 mm, respectively.

For the optical movement detection, we attached a collage of colorized posters to the wall with a dimension about 1.2 × 0.8 m. This was done to create well-defined and predictable conditions for the optical movement detection technique in the usability test and to achieve a performance comparable to the other techniques. Outside the prepared

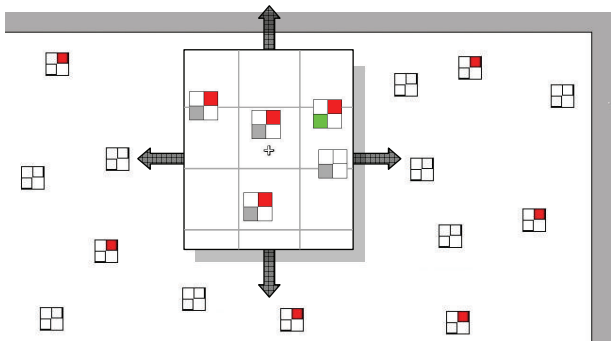

Figure 2: Panning a dynamic peephole window over a large fixed workspace. Only the part in the peephole window is visible at a time.

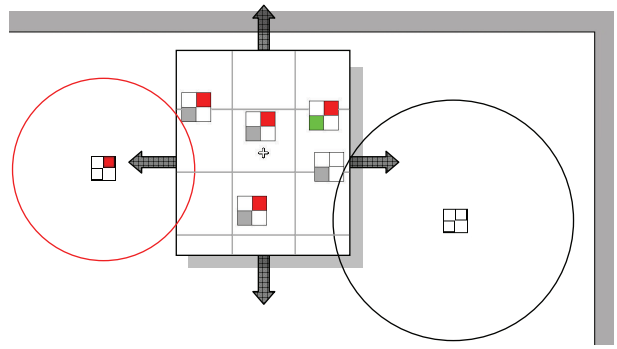

Figure 3: Halos indicate off-screen objects. Red halos indicate target candidates.

area, there was white wall and gray carpet on which the movement detection performed poorly.

The weight of the used Nokia N80 and the attached SHAKE unit was 170 g. The sensor was attached at the lower end of the phone, which was found to be the ergonomically best position. After attaching it to the phone, the magnetometer had to be recalibrated, because of EMI from the phone. It was then recalibrated again in the room in which the usability test was carried out in order to compensate for the influence of the local electromagnetic field.

The accelerometer was used to control the y-component of the cursor position. A range from 20° to 80°, from the horizontal plane, was used and linearly mapped to between 800 (bottom of workspace) and 0 (top of workspace). The magnetometer was mapped to the x-component of the cursor, ranging from −45° at the left end of the workspace and +45° at its right end. This corresponds to 13.3 pixels per degree. The raw data were filtered with an efficient least squares fit to a polynomial of degree 2.

3. Navigation on Small Displays

Small handheld displays require specific strategies for information navigation and visualization as well was appropriate interaction techniques. For example, the display can be used as a movable window ("peephole") into a virtual space [11]. The movement of the display is compensated by a corresponding movement of the virtual display in the opposite direction. Whereas [11] implements two-handed interaction techniques for touch screen devices, we focus on one-handed techniques, assuming that the user might only have one hand free to operate the device. Mehra et al. [8] show advantages of dynamic over static peephole navigation. Dynamic peephole navigation means moving the peephole across a static spatial layout. Static peephole navigation means moving the spatial layout behind a static peephole, that is, traditional scrolling. The former means temporal integration to construct an internal representation of the spatial layout. The latter requires spatiotemporal integration due to changing position of the elements in the layout. The *pan* condition shown in Figure 2 corresponds to the peephole model. In addition to a previous study, we included a vertical and horizontal lines in the background [10], which is beneficial to perceive motion of the workspace when there are only few objects.

Halo [9] visualizes off-screen objects by surrounding them with rings that reach into the border region of the display window (Figure 3). 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. This technique uses little screen space and has been shown to significantly reduce task completion times compared to arrow-based visualization techniques [9]. Whereas in [9], 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 camera phone.

With *zoomable* or *multiscale* interfaces [12–14] users can continuously adjust the scale at which virtual objects are rendered. In addition to standard cursor pointing for object selection, users have to perform *view pointing* [13] to navigate in scale and space to the target view that includes the object at the proper scale. For *grid*, we implemented a zoomable interface with extended tracking range compared to [10]. As the user moves away from the grid, the interface zooms out. As she moves closer to the grid, the interface zooms in.

In [14], Perlin and Fox introduce the concept of *semantic zooming*. Beyond simply scaling objects to different magnifications, the representation of an object and its level of detail change with scale. A particular representation is associated with a particular scale range. The abstract navigation task we used has three levels of detail. As shown in Figure 4, at the smallest scale, all objects look the same, at medium scale target candidates show a red mark, and 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 (gray mark).

For grid tracking, we implemented *pan*, *halo*, *zoom*, and the combination of halo and zoom (henceforth *halo&zoom*). For *flow* and *shake* we only implemented *pan* and *halo*, since these sensing methods lack a clear dimension for zoom.

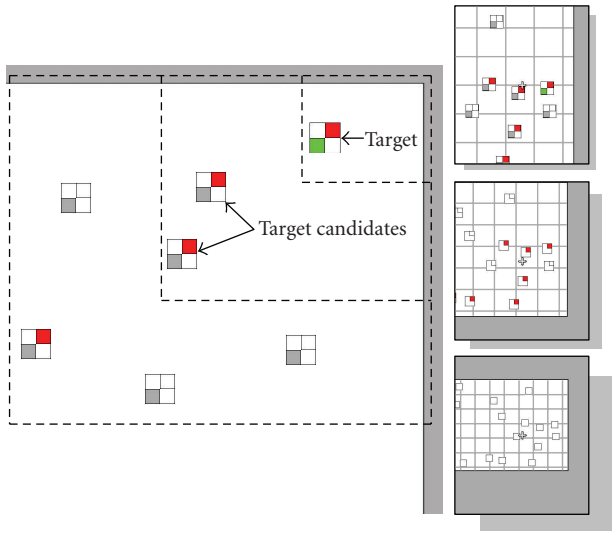

FIGURE 4: Semantic zooming with three levels of detail. Continuous scaling within each level. The dashed lines indicate the sizes of the respective areas at different zoom levels.

Initial results for grid tracking were presented in [10]. Here we include *flow* and *shake* as additional sensor technologies and the *grid* technique with an extended tracking range.

4. User Study

In the following, we present a user study with two separate task groups to compare the discussed small-display navigation techniques in the context of spatially aware displays with various sensor technologies. Our main hypothesis is that the different characteristics of the sensor technologies will lead to differences in both performance and subjective rating by the test users. Our aim is to learn more about the individual strengths and weaknesses of the sensing methods. A second hypothesis is that *halo* is most effective for small numbers of items in the workspace and that it does not scale well to a large number of off-screen items, whereas *zoom* is helpful irrespective of the number of items.

4.1. Participants and Apparatus

The study was conducted on 18 participants, 9 female, 9 male, age 22–32. These were split into two groups for separate tasks. 12 participants (6 female, 6 male) performed the first task set, the remaining 6 participants (3 female, 3 male) performed the second. Subjects were undergraduates, doctoral students, or postdoctoral researchers with varied degrees of technical background.

Subjects performed the test on a Nokia N80 Symbian phone which had a SHAKE unit attached to its lower back. The display (35×41 mm) was used in full-screen mode (352×416 pixels). Two identical devices with this setup were used. During each trial, the (x, y, z) ((x, y) for *flow* and *shake*) coordinates of the motion trajectory were sampled at an average rate of 13.2 updates per second. The time to target for each trial and the click patterns during the trials were recorded as well. The size of the virtual workspace were 1200×800 pixels.

4.2. Tasks

The general scenario for both tasks was the same. Users had to find a target among a number of distractors in a virtual workspace (Figures 2–4). There was always exactly one target present in the workspace at a time, which was indicated by a green square in the lower-left corner. Candidate distractors were marked with a red square in the upper right corner. False distractors had no color in the upper-right corner and gray in the lower-left corner of the target. Once found and clicked, the next target appeared at a different place in the workspace. 10 trials were performed for each arrangement of distractors, combination of navigation techniques, and types of sensor technology. In the first task, group 0, 32, and 48 targets were generated. We chose this wide span in the number of targets to make any effects of this variable more clearly visible in the results. In the second task, group 4, 8, and 16 targets were generated. This task group was chosen to focus on the more detailed effects of a smaller numbers of targets. If the navigation technique allowed revealing of proximity information, 50% of the distractors appeared as target candidates.

As sensor technology, each of the following was used during the experiment:

(i) marker grid tracking via mobile phone camera;

(ii) optical motion detection via mobile phone camera; and

(iii) tilt and rotation detection via SHAKE accelerometer and magnetometer readings.

For grid tracking the offered navigational options were as follows:

(i) panning only,

(ii) halo only,

(iii) zoom only, and

(iv) halo and zoom combined.

For *flow* and *shake* the navigational options were as follows:

(i) panning only, and

(ii) halo only.

Zoom is an interaction feature that was only available for marker grid interaction, hence the types of interactions were reduced for other sensor technology types in the trial.

Panning is a standard flat navigation method where one navigates close up to zoom moving in the plane until one finds the target in one's display area. As a basic mechanism, panning was available in all conditions. Halo, as discussed earlier, provides circular arcs, which are a guidance for existing objects. Halos reached 40 pixels into the display.

Zoom allows the user to continuously move in and out of level of detail by using distance to the plane. Our

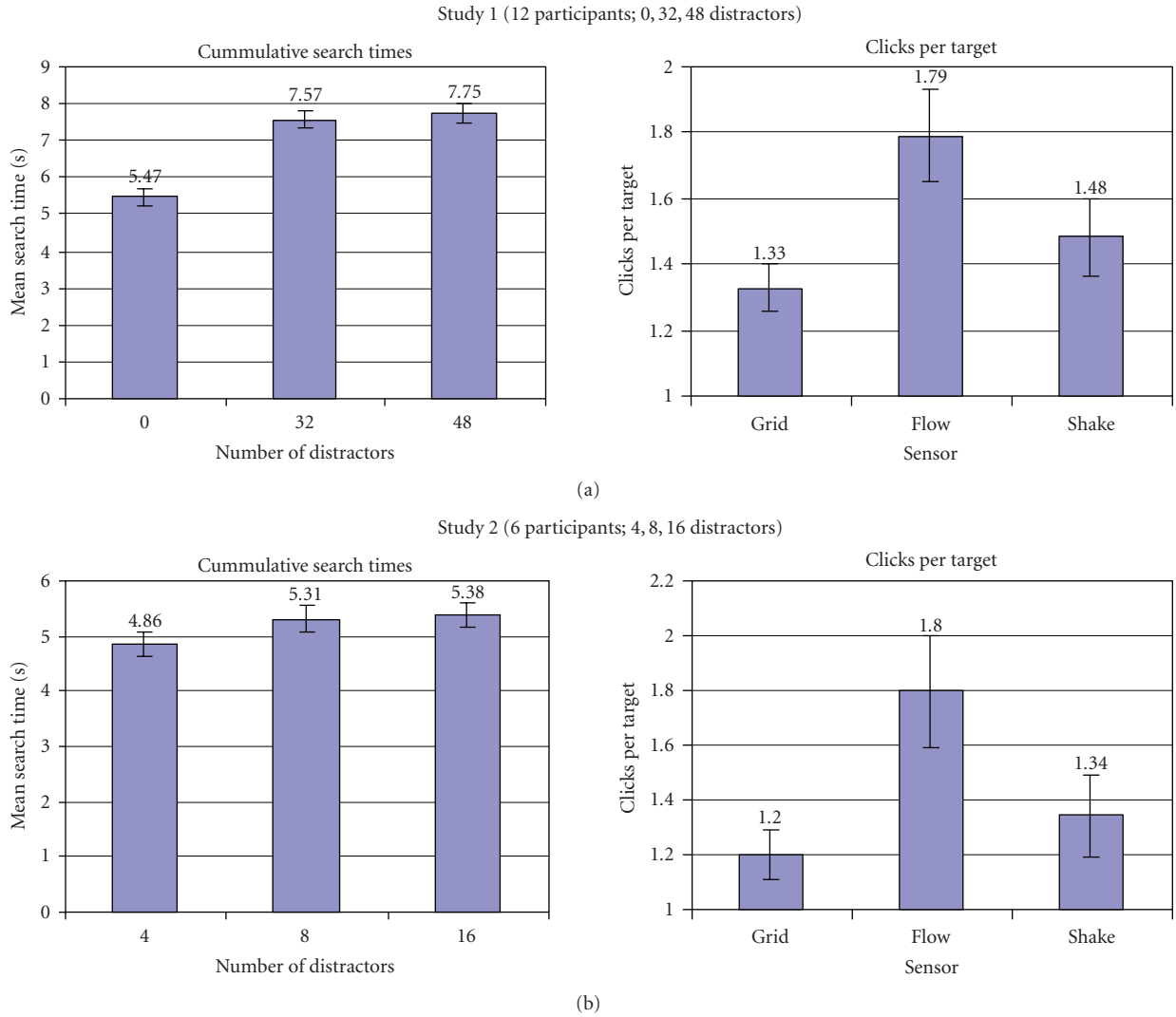

Figure 5: Overall task time by number of distractors (left) and mean number of clicks per target (right).

implementation used three zoom regions for three levels of detail. The minimum height for sensible registration was set to 50 mm. If a lower value was found, the display background became gray to indicate that the detection became problematic. Same holds for the maximum height value of 285 mm. The level changes occur at values of 117 mm and 184 mm. At the highest levels, all objects—whether valid or distractors—look alike. Entering the second zoom level reveals 50% of objects as possible candidates. Only at the closest zoom level can distractors and the target be completely differentiated. The dynamic range of the zoom is much larger than in an earlier experiment [10] and was included to provide clues whether the range of this interaction critically limited its performance in this earlier study.

A similar behavior was implemented for halo to allow comparable navigational information between the two approaches. When one was within the range of a target that corresponds to the viewable area of a given zoom level, the color of the halo would differentiate between target candidates and distractors in the same proportions as zoom levels do. Hence if one gets closer to targets, halos would change colors to first reject more distractors.

4.3. Design

The two studies were designed as within-participants factorial designs with three factors. Study 1 (12 participants) used the following factors:

(i) sensor technology: *grid*, optical *flow*, *shake*;

(ii) navigation technique: *pan*, *halo*, *zoom*, *halo&zoom*;

(iii) distractor count: 0, 32, 48.

Study 2 (6 participants) used the following factors:

(i) sensor technology: *grid*, optical *flow*, *shake*;

(ii) navigation technique: *pan*, *halo*, *zoom*, *halo&zoom*;

(iii) distractor count: 4, 8, 16.

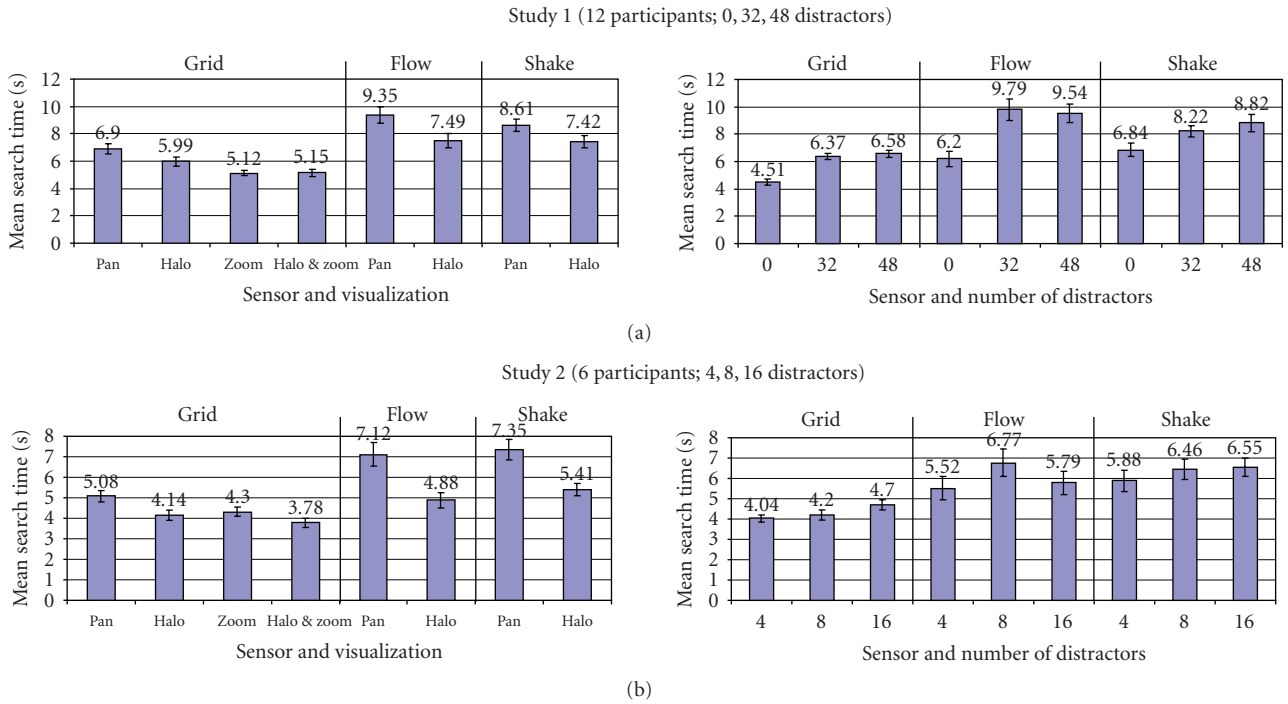

Figure 6: Mean search times for each sensing method grouped by navigation technique (left) and number of distractors (right). Note that the scaling of the y-axis is different for the upper and the lower panels.

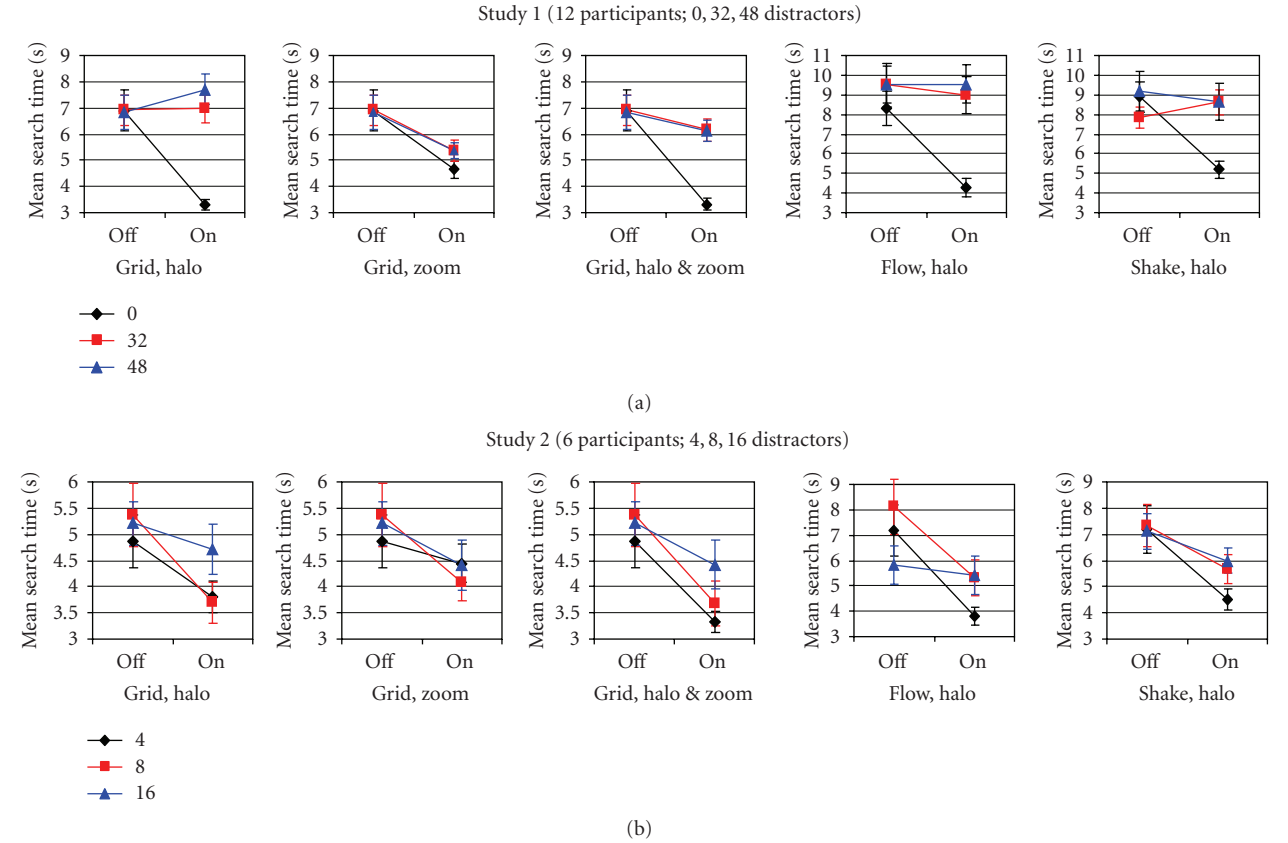

Figure 7: Change in performance as different visualization techniques are enabled for different numbers of distractors. For study 1, an interaction between the number of distractors and halo, but not zoom, is apparent.

In both cases, *zoom* was only factored for the grid sensor technology. This results in $2 \times 2 \times 3 + 4 \times 3 = 24$ conditions for each study. The order of technologies was counterbalanced and presented in blocks, for example, all shake interactions happened in one block without switching to another sensor technology. The order of conditions within blocks was randomized beforehand and was distinct for each participant. The test application recorded the movement trajectories and trial completion times of 12 users × 24 conditions/user × 10 trials/condition = 2800 trials for study 1 and of 6 users × 24 conditions/user × 10 trials/condition = 1400 trials for study 2.

The distance from one target to the next was always 500 pixels. Participants were not informed about this fact. The target size was 60 pixels (6 mm). Thus each trial had a Fitts' Law index of difficulty [15] of ID = $\log_2(500/60 + 1) = 3.2$.

4.4. Procedure

Prior to recording actual test data, users performed a number of practice trials, until they felt familiar with each navigation method and sensor technology. This typically took about 10 to 15 minutes. Before each new condition, the user was informed about the sensor technology and navigational methods provided. When the participants were ready, they clicked the right selection button on the device to start the next task, consisting of finding 10 individual targets in sequence. 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 in the center of 5-way direction key, a new target appeared and the task continued until 10 targets were found.

The orders 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 of Study 1

Trial time is taken as the main performance measure, which is the time from starting a trial to finding the target. The cumulative average results can be seen in Figure 5 (top left). (Error bars in all figures show 95% confidence intervals.) Zero distractors provides better performance (5.47 seconds) than 32 distractors (7.57 seconds) and 48 distractors (7.75 seconds). A three-way repeated measures ANOVA shows that the number of distractors has a significant effect on trial time ($F_{2,195} = 12.79$, $p < 0.01$). A Tukey HSD multiple comparison test shows that the trial time for 0 distractors is significantly different from 32 and 48 distractors, but there is no significant difference in trial time between 32 and 48 distractors. In terms of sensing methods, the cumulative trial time is 5.80 seconds for *grid*, 8.02 seconds for *shake*, and 8.49 seconds for *flow* ($F_{2,195} = 6.95$, $p < 0.01$). A multiple comparison test shows that trial time for *grid* is significantly different from *flow* and *shake*, but trial time for *flow* is not significantly different from *shake*. Switching the *halo* visualization on/off also has a significant effect on trial times ($F_{1,195} = 8.77$, $p < 0.01$). Since the *zoom* visualization is only available in the *grid* sensor condition, we did separate analyses. The above ANOVA was performed by omitting the conditions in which *zoom* was active. The results for *zoom* presented below were analyzed within the *grid* sensor condition.

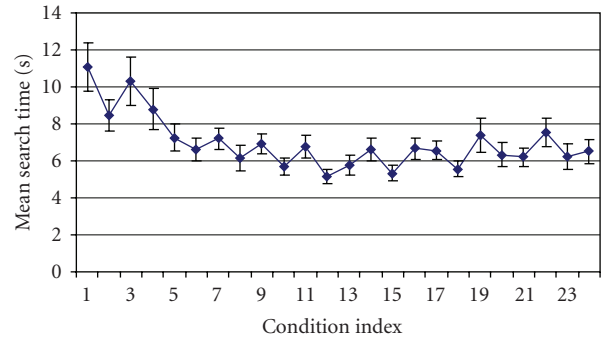

Figure 8: Task time over progressive trials averaged over all sensor types and visualization techniques.

In Figure 5 (top right), one can see how many clicks the participants performed. This serves as a measure of the difficulty to select a target, and hence gives an indication of the fine motor aspect of precise targeting with a given sensor technology. Here *grid* performs best (1.33 clicks), followed by *shake* (1.48 clicks), then *flow* (1.79 clicks). A three-way repeated measures ANOVA shows that the sensing method has a significant effect on the number of clicks ($F_{2,195} = 12.47$, $p < 0.01$). A multiple comparison test shows that the number of clicks for *flow* is significantly different from the other two sensing methods. From our observations, we conclude that this is due to reclutching in the *flow* condition.

The relative performance of the visualization techniques can be seen in Figure 6 (top left). For the *grid* condition, *pan* takes longest to finish (6.90 seconds), followed by *halo* (5.99 seconds). *Zoom* (5.12 seconds) and *halo&zoom* (5.15 seconds) show comparable performance. A three-way repeated measures ANOVA for the *grid* condition shows that there is a significant effect on trial time for the number of distractors ($F_{2,126} = 15.11$, $p < 0.01$) and for the activation of *zoom* ($F_{1,126} = 20.79$, $p < 0.01$). There is no significant effect for *halo* ($F_{1,126}, = 2.05$, $p = 0.16$), which is due to the large number of distractors in study 1 and the lower effectiveness of *halo* in this case (further discussed below).

The effect of visualization becomes most apparent when looking at the impact of the number of distractors on the various techniques. Figure 7 shows the trial time for each combination of sensing method and visualization technique. The upper leftmost diagram, for example, shows the change in trial time for *grid* as *halo* is switched off (left) and on (right). The endpoints of each line indicate the trial time for a particular number of distractors (see legend at the top). One sees that *halo* receives a drastic performance boost in the absence of distractors but for a large number of distractors either gains no performance or even loses some. The second diagram in the top row shows that *zoom*

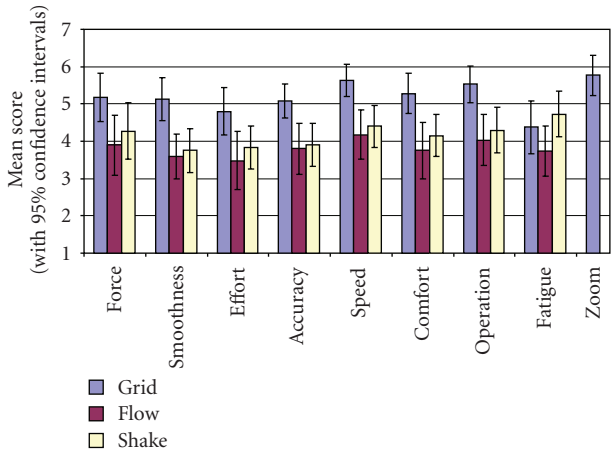

Figure 9: Results of the user interface evaluation questionnaire (studies 1 and 2 combined).

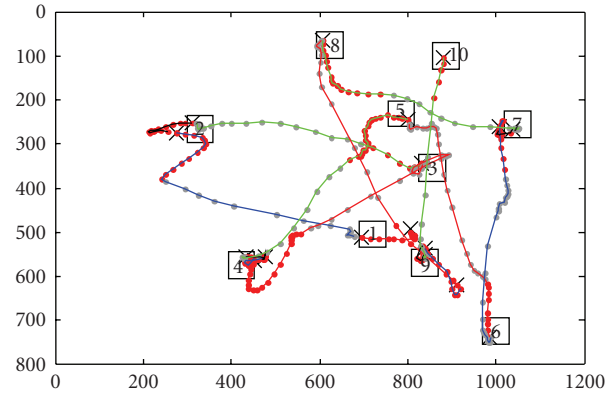

Figure 10: A search path in the absence of distractors using halos.

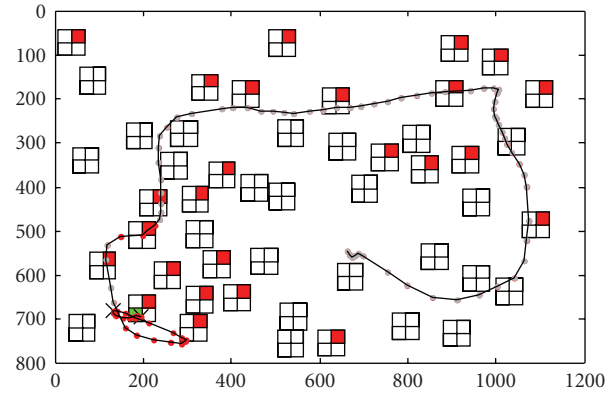

Figure 11: A single search trajectory in the presence of 48 distractors using halos.

gains performance independent of the number of distractors, but the overall performance gain is less than with *halo* for no distractors. Interestingly, these two behaviors combine well, as is depicted in the third diagram. To analyze this phenomenon, we again treated *grid* independently from the other sensing methods, because it is the only one that includes *zoom*. For *grid*, ANOVA reveals an interaction between *halo* and the number of distractors ($F_{2,77} = 12.89$, $p < 0.01$), whereas there is no interaction between *zoom* and the number of distractors ($F_{2,77} = 0.18$, $p = 0.83$). When separately analyzing the conditions without zoom (over all sensing methods), there is again an interaction effect for *halo* and the number of distractors ($F_{2,132} = 19.61$, $p < 0.01$). In summary, this means that the effectiveness of *halo* depends on the number of distractors, which is not the case for *zoom*. This supports our second hypothesis.

The averaged trial time over all subjects does show a learning trend (Figure 8). The performance remains flat after the first 4 trials. This trend exists for all methods and its impact on the study is mitigated by the counterbalanced presentation of the techniques.

At the end of each task, the participants were asked to fill out a modified version of the "user interface evaluation questionnaire" of ISO 9241-9 [16]. The modification consists of having only a single Fatigue category as question 8 and adding a 9th question on the helpfulness of zoom. The ISO questionnaire is a 7-point rating evaluation (higher means better). The results are shown in Figure 9. We see that for all categories of the questionnaire, *grid* was evaluated most favorably, followed by *shake*, then *flow*. *Shake* was rated as slightly less tiring than *flow*.

4.6. Results of Study 2

In study 1, we observed a saturation of trial time for 32 and 48 distractors. In order to investigate the range of low numbers of distractors, we used 4, 8, and 16 distractors in a second study. All other factors were the same as in the first study. The overall results can be seen in Figure 5 (bottom left). From the same figure (bottom right), we see that clicks per target are again considerably higher for *flow* due to reclutching. Overall trends are comparable to the results of study 1. The detailed results of the impact of number of distractors, visualization method, and sensor technique are given in Figure 6 (bottom). There is less benefit from *halo* with 16 distractors than with 4 and 8.

We again treat the effect of *zoom* only within the *grid* condition, because it is not available in the other conditions. For the conditions without *zoom*, a three-way repeated measures ANOVA shows that the sensing method has a significant effect on trial time ($F_{2,96} = 17.98$, $p < 0.01$): 4.33 seconds for *grid*, 5.97 seconds for *flow*, and 6.34 seconds for *shake*. A Tukey HSD multiple comparison test shows that there is no significant difference between *flow* and *shake*. There is no significant effect of the number of distractors ($F_{2,96} = 2.26$, $p = 0.11$): 4.86 seconds for 4, 5.31 seconds for 8, and 5.38 seconds for 16 distractors. This is apparently due to the fact that the range of distractor counts is smaller in this second study. However, there is a significant effect of *halo* ($F_{1,96} = 48.84$, $p < 0.01$) and an interaction between *halo* and the number of distractors ($F_{2,63} = 4.03$, $p = 0.02$).

When focusing on the *grid* method (Figure 6, lower part), it turns out that there is a weakly significant effect of the number of distractors ($F_{2,62} = 3.24$, $p = 0.046$), and

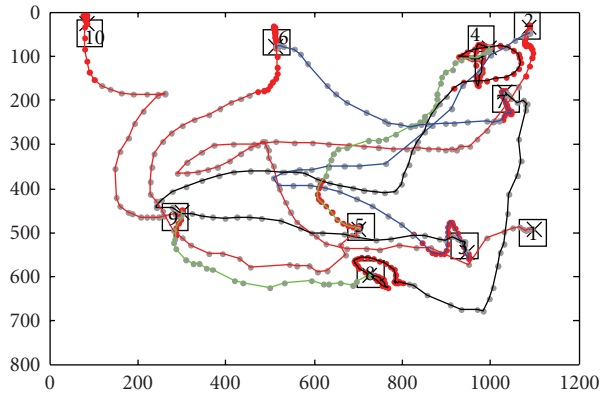

Figure 12: A search path using halo&zoom in the plane.

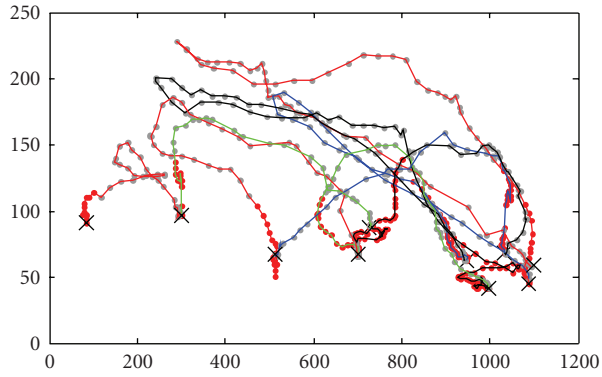

Figure 13: A search path using halo&zoom showing the height.

significant effects of *zoom* ($F_{1,62} = 7.02$, $p < 0.01$), and *halo* ($F_{1,62} = 15.33$, $p < 0.01$). Within the *grid* method, no interaction effects were found.

4.7. Motion and Search Strategies

An array of motion strategies was used. In the absence of any visualization support through either halo or zoom, participants either used left-right sweeping gestures, or circular motions to find a target. This is similar to results reported in an earlier study on these visualization techniques [10]. The motion patterns also reveal in detail how the visualization techniques help the interaction. Figure 10 shows all search paths during 10 trials of a distractor-free configuration when halos were presented. As there is only one circular arc, participants could immediately identify the center of the target from the halo and follow direct paths towards those targets. In the presence of distractors, this advantage disappears. Figure 11 shows a single trail to find a target among 48 distractors. One can see that while halos help identify target candidates and the participant uses a path towards red-marked candidates to ultimately find the target, the navigation technique no longer guides straight to the target. In this figure, one also sees that once the target is found, the participant over-shoots and corrects the motion for the select. This is a typical feature that can be seen in most cases.

Zoom allows participants to get an overview of the configuration quickly and then zoom in onto the target for a select. Figure 12 shows the motion in the plane and Figure 13 shows the motion in the zoom direction for a set of trials without distractors but in presence of both halo and zoom. The selection points are marked with a cross.

4.8. Discussion

The impact of distractor densities on different visualization techniques supports the hypothesis that halo is helpful in scenarios with low numbers of distractors and successively loses its helpfulness as the number of distractors increases. At very large numbers, the halos seem to serve as distractors themselves, and hence performance decreases slightly over the case without visual guidance. Zoom seems to help independently of the number of markers but generally less than the best case for halo. We observe that these two effects can be combined to achieve both benefits. In a practical implementation, one might consider disabling halos if the number of ambiguous targets becomes too large and solely provide a zoom interface. In a previous study [10], the benefits of zoom were not as obvious as in this study. We suppose that the improved performance of zoom is due to the increased tracking range of the dynamic digital zoom technique.

Of the studied sensor technologies, *grid* is still the most desirable. Of the studied techniques, it is however also the only one that specifically needs a fixed reference. Of the sensor methods that do not explicitly require a reference, *shake* performs better than *flow*. Optical flow tracking was hindered by the limitations of the need of contrast flow in the optical field of view. Because of the difficulty of this, a repositioning mechanism was used which in turn required additional effort of the participants, which was not required using other methods.

In the user study, *shake* showed least fatigue, though not with clear significance. It could support the hypothesis that the free choice of arm and hand-position helped counter fatigue during the trials. Further evidence is necessary to support this hypothesis.

5. Conclusions

We conducted two studies to evaluate the performance of three sensing methods of one-hand motions for a mobile device for the purpose of 2-D scanning tasks as they occur, for example, in map navigation. All combinations of sensors and visualizations provided a viable solution for the small display navigation problem, but with different tradeoffs.

The study shows that optical marker grid tracking is the most desirable technology of the three, both in terms of measured performance and in terms of subjective evaluation by the participants in a post-study questionnaire. Accelerometer and magnetometer sensing was found to be more desirable than an optical flow tracking method.

The absence or presence of visual guidance was studied as well, showing that halos—circular arcs in the visual periphery of the display—are most helpful for low numbers of distracting targets. Zooming helps independently of the number of distractors available. The effect of these two methods can be combined and the joint performance keeps the desirable features of the individual performance. This is a refinement of findings of a previous study which indicated that combining these two methods does not show increased benefit [10].

Possible directions for future work include expanding the range of sensor technologies to be considered. One possibility is to explore enhanced methods for optical tracking, such as using active markers, or using nonoptical short-distance pointing technologies such as RFID. There are also various less frequently used motion-based sensors such as gyroscopes, which may be worth considering as additions or alternatives to accelerometers and magnetometers. Another aspect is the comparison with traditional keypad navigation. In [17], we have shown that absolute camera-based tracking is faster than keypad navigation. This could be investigated with a wider range of sensors.

Finally, it is of interest to consider extending the scope of the interaction further. For example, one can imagine combining the range of motion interactions studied in this paper with device-bound interaction types such as using the direction key or allowing additional interactions on the touch screen of the device. It is as yet open which of the techniques studied here are best suited for this kind of hybrid input paradigm.

Acknowledgment

An earlier version of this paper has been presented at MobileHCI 2007.

References

[1] B. Hecht, M. Rohs, J. Schöning, and A. Krüger, "Wikeye—using magic lenses to explore georeferenced Wikipedia content," in *Proceedings of the 3rd International Workshop on Pervasive Mobile Interaction Devices (PERMID '07)*, Toronto, Canada, May 2007.

[2] M. Rohs, G. Essl, and M. Roth, "CaMus: live music performance using camera phones and visual grid tracking," in *Proceedings of the 6th International Conference on New Instruments for Musical Expression (NIME '06)*, pp. 31–36, Paris, France, June 2006.

[3] S. Zhai and V. Bellotti, "Introduction to sensing-based interaction," *ACM Transactions on Computer-Human Interaction*, vol. 12, no. 1, pp. 1–2, 2005.

[4] S. Benford, H. Schnädelbach, B. Koleva, et al., "Expected, sensed, and desired: a framework for designing sensing-based interaction," *ACM Transactions on Computer-Human Interaction*, vol. 12, no. 1, pp. 3–30, 2005.

[5] K. Hinckley, J. Pierce, E. Horvitz, and M. Sinclair, "Foreground and background interaction with sensor-enhanced mobile devices," *ACM Transactions on Computer-Human Interaction*, vol. 12, no. 1, pp. 31–52, 2005.

[6] K. Hinckley, J. Pierce, M. Sinclair, and E. Horvitz, "Sensing techniques for mobile interaction," in *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology (UIST '00)*, pp. 91–100, San Diego, Calif, USA, November 2000.

[7] G. Fitzmaurice and W. Buxton, "The Chameleon: spatially aware palmtop computers," in *Proceedings of the Conference Companion on Human Factors in Computing Systems (CHI '94)*, pp. 451–452, Boston, Mass, USA, April 1994.

[8] S. Mehra, P. Werkhoven, and M. Worring, "Navigating on handheld displays: dynamic versus static peephole navigation," *ACM Transactions on Computer-Human Interaction*, vol. 13, no. 4, pp. 448–457, 2006.

[9] P. Baudisch and R. Rosenholtz, "Halo: a technique for visualizing off-screen objects," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*, pp. 481–488, Fort Lauderdale, Fla, USA, April 2003.

[10] M. Rohs and G. Essl, "Which one is better?—information navigation techniques for spatially aware handheld displays," in *Proceedings of the 8th International Conference on Multimodal Interfaces (ICMI '06)*, pp. 100–107, Banff, Canada, November 2006.

[11] K.-P. Yee, "Peephole displays: pen interaction on spatially aware handheld computers," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*, pp. 1–8, Fort Lauderdale, Fla, USA, April 2003.

[12] G. W. Furnas and B. B. Bederson, "Space-scale diagrams: understanding multiscale interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95)*, vol. 1, pp. 234–241, Denver, Colo, USA, May 1995.

[13] Y. Guiard and M. Beaudouin-Lafon, "Target acquisition in multiscale electronic worlds," *International Journal of Human Computer Studies*, vol. 61, no. 6, pp. 875–905, 2004.

[14] K. Perlin and D. Fox, "Pad: an alternative approach to the computer interface," in *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '93)*, pp. 57–64, Anaheim, Calif, USA, August 1993.

[15] P. M. Fitts, "The information capacity of the human motor system in controlling the amplitude of movement," *Journal of Experimental Psychology: General*, vol. 121, no. 3, pp. 262–269, 1992.

[16] International Organization for Standardization, "Ergonomic requirements for office work with visual display terminals (VDTs)—part 9: requirements for non-keyboard input devices," ISO/IEC 9241-9, 2000.

[17] M. Rohs, J. Schöning, M. Raubal, G. Essl, and A. Krüger, "Map navigation with mobile devices: virtual versus physical movement with and without visual context," in *Proceedings of the 9th International Conference on Multimodal Interfaces (ICMI '07)*, pp. 146–153, Nagoya, Japan, November 2007.