Olwal, A., and Feiner, S. Spatially Aware Handhelds for High-Precision Tangible Interaction with Large Displays. *Proceedings of TEI 2009 (International Conference on Tangible and Embedded Interaction)*, Cambridge, UK, February 16-18, 2009, pp. 181-188.

# Spatially Aware Handhelds for High-Precision Tangible Interaction with Large Displays

**Alex Olwal**
School of Computer Science and Communication
KTH (Royal Institute of Technology), Sweden
alx@csc.kth.se

**Steven Feiner**
Department of Computer Science
Columbia University, USA
feiner@cs.columbia.edu

## ABSTRACT

While touch-screen displays are becoming increasingly popular, many factors affect user experience and performance. Surface quality, parallax, input resolution, and robustness, for instance, can vary with sensing technology, hardware configurations, and environmental conditions.

We have developed a framework for exploring how we could overcome some of these dependencies, by leveraging the higher visual and input resolution of small, coarsely tracked mobile devices for direct, precise, and rapid interaction on large digital displays.

The results from a formal user study show no significant differences in performance when comparing four techniques we developed for a tracked mobile device, where two existing touch-screen techniques served as baselines. The mobile techniques, however, had more consistent performance and smaller variations among participants, and an overall higher user preference in our setup. Our results show the potential of spatially aware handhelds as an interesting complement or substitute for direct touch-interaction on large displays.

## Keywords

spatially aware, interaction technique, MobileGesture, MobileRub, MobileDrag, MobileButtons, LightSense, mobile, touch-screen, touch, tangible

## INTRODUCTION

There has been an increasing interest over the past years in interactive surfaces that allow the user to interact with media on large digital displays. At the same time, improvements in computational power and connectivity have made mobile devices attractive platforms for mobile augmented reality [9, 18, 19, 22, 23] and for use as spatially aware handhelds [6, 15, 16, 17]. If the device is aware of its spatial position, context-specific information may be presented on its display, allowing it to act as a portable viewport into a larger space. Many research projects have been investigating the potential of such spatially aware devices, addressing manipulation and information exchange between a public display and a mobile device.

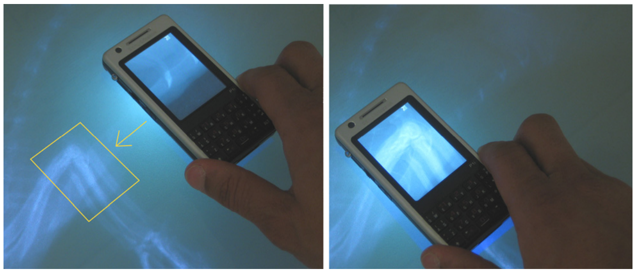


**Figure 1. A spatially tracked handheld display (Sony Ericsson P1i) provides a viewport with over ten times the visual and input resolution of the rear-projected system, on a smaller, but higher-quality, screen.**

In this paper, we investigate how tracked mobile devices can be used as an alternative or complementary means of input to direct touch-interaction for digital displays. Mobile touch-screen devices typically have considerably higher visual and input resolution (measured in pixels per unit area) than large displays. Mobile devices are also equipped with a rich set of additional sensors and input controls. We are especially interested in how we might leverage these features to use handheld devices for precise and fast interaction in context of a larger display, as shown in Figure 1. To explore this scenario, we conducted a user study that compares four techniques for interacting with a tracked mobile device in the space of a larger display, and two existing touch-screen techniques for the larger display alone.

## RELATED WORK

Interaction with spatially aware handhelds was pioneered by Fitzmaurice [6] and Rekimoto [18], who developed early prototypes with exocentric and egocentric tracking. Later, Yee [26] used tethered sensing to track a handheld prototype, with an emphasis on touch-screen interaction in a locally defined virtual space, while Benko et al. demonstrated a movable focus display [5] that used a tracked tablet PC on a front-projected surface.

Recent developments in mobile sensing have now made it possible for unmodified, commercially available, mobile devices to support novel interaction with digital displays. Ballagas et al. [2] review several techniques for direct and indirect control of content on a digital display, such as using the device's camera for optical flow or detection of visual codes, relying on an external camera to track a mobile display with flashing coded patterns, and employing different physical input mechanisms of the device. Rukzio et al. [20] evaluate and compare technologies for physical interaction with a real environment, categorizing them as "touching" (RFID), "pointing" (optical), and "scanning" (Bluetooth and GPS) interactions.

Specialized sensing technologies or visual alteration are problematic for mobile devices, and this has led to a preference for inside-out, egocentric tracking. Such camera-based approaches [2, 9, 18, 19, 22, 23], in which the mobile device's camera is used to recover its pose from tracked features in the environment, require that the device is held at a distance from the surface to ensure the camera's unblocked view of the scene. There is a small inconsistency here with tangible interfaces, in which physical objects are typically manipulated on the surface, rather than above it.

Reilly et al. [16] describe how a modified PDA can detect its position on paper maps with embedded RFID tags and display relevant information. Hardy and Rukzio [7] discuss a mobile device that infers its spatial position from a grid of embedded Near Field Communication tags behind a projection surface. The user can touch locations in the projected image and trigger commands with the mobile device's controls. Despite limited interactivity due to low tag read rate and coarse tracking resolution, several advantages of direct manipulation with mobile devices are demonstrated. They report comparable performance to, and better user preference than, direct finger-interaction on a large touch-screen.

Continuous tracking at interactive frame rates has also been demonstrated with optical, outside-in tracking methods. Bluetable [24] detects phones on an interactive front-projected surface and connects to them over Bluetooth. It disambiguates the device's position through optical communication using the device's IR port or LCD. LightSense [15] tracks the photo light of a mobile device on a static map and wirelessly informs the device of its position and estimated distance from the surface.

## INTERACTION TECHNIQUE TESTBED

We implemented a testbed to experiment with interaction techniques for spatially tracked mobile devices. We track the mobile device on a touch-sensitive rear-projected surface, allowing coordinated graphics to be displayed on the surface and the device, as shown in Figures 1–3. Interaction with content is supported through direct touch on the surface, or by using the input capabilities of the mobile device.

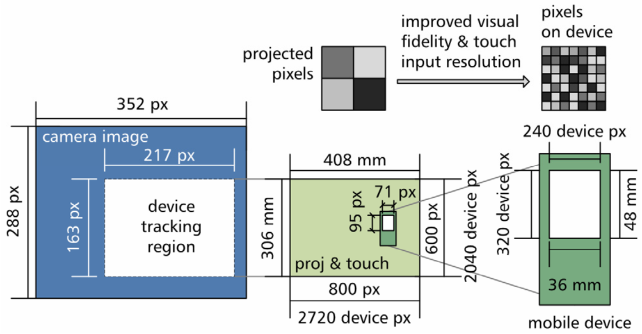


**Figure 2. Resolution levels in the system. The device is tracked in a region in the camera view that corresponds to the size of the projection and touch overlay. The SVGA projection on the 408×306mm touch screen area results in 0.51 mm sensing resolution and pixel width, compared to 0.15 mm achieved on the mobile device's 36×48mm QVGA display. The device thus achieves an order of magnitude (11.56 times) higher pixel density and touch resolution per area.**

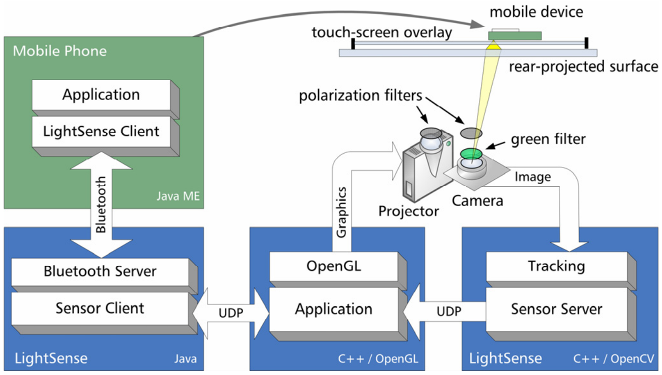


**Figure 3. System architecture. Optical filters ensure that only the device's photo light reaches the camera for tracking, while a projector provides visual feedback. Bidirectional Bluetooth communication allows the mobile device to receive application data and its position, and send input events to the PC.**

Our Sony Ericsson G900i mobile device has a 1.8" resistive touch-screen, with 240×320 resolution. The software on the mobile device was implemented in Java ME and communicates with a PC using Bluetooth to receive position updates and to transmit user input.

The rear-projected surface is supported by a dual 3.0 GHz Pentium Xeon PC with 2 GB RAM and an NVIDIA Quadro FX 500 graphics card, running Windows XP. The tracking software was implemented using C++ and OpenCV, the application software was implemented using C++ and OpenGL, and the software that communicates with the mobile device was written in Java SE. The components communicate through local UDP network packets.

Graphics are rear-projected at 800×600 resolution with a small DLP projector (Mitsubishi PK-20), and a USB camera (Microsoft LifeCam VX-6000) is installed under the surface. We use the LightSense framework [15] to track the mobile device's activated photo light in a 217×163 pixel region of the camera image.

While many different techniques exist to enable touch-sensitivity on surfaces, in this study we used a commercially available, cost-effective approach, based on a transparent (required for rear-projection and optical tracking) resistive single-touch overlay (Magic Touch KTMT-1921), with the same 800×600 resolution as the projected graphics. The 20" touch-sensitive surface is attached to a 10 mm diffuse glass projection surface, which is mounted horizontally on a 840 mm high metal frame. Unfortunately, this particular experimental setup offsets the touch-sensitive overlay from the projected graphics by 17 mm. This parallax is caused by the thickness of the projection surface, the height of the touch overlay frame, and the thickness of the glass to which the touch-sensitive film is applied.

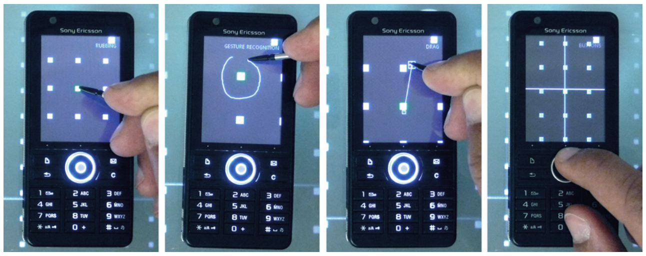
### Handheld Focus+Context Application

Using our framework, we developed an application that leverages the significantly higher pixel density of the spatially tracked handheld, similar to the motivation of Focus + Context displays [3] in general and the movable focus display of Benko et al. [5]. Figure 1 shows how the handheld display creates a viewport with a high-resolution portion of the larger, rear-projected image. As the handheld display is moved continuously on the surface, its images are dynamically updated. In addition to the higher visual and input resolution, this approach also benefits from the handheld display's overall superior quality, given its higher contrast, better brightness and larger viewing angles.

Mobile touch-screen techniques can overcome the limitations of the equipment in the environment, by supporting precise interaction through their locally defined user interface. While the mobile device is tracked quite coarsely relative to the projection surface in our experimental setup, its display provides much higher fidelity, as illustrated in Figure 2. The visual and input resolution per unit area is over 11 times higher on the mobile device than on our rear-projected touch screen surface.

### Interaction techniques for tracked mobile devices

Based on experiments and results from pilot studies, we chose four designs for handheld interaction techniques on the mobile device (See Figure 4). While all of the mobile touch-screen techniques support finger interaction, we had all participants use the device's stylus in this study.



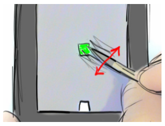
**a) Rub** **b) Gesture** **c) Drag** **d) Buttons**

**Figure 4. The four interaction techniques for a tracked mobile device. a) MobileRub zooms in on the target through small rubbing gestures. b) MobileGesture zooms up to the drawn circle. c) MobileDrag zooms in when the user drags upwards from the target. d) MobileButtons uses a dedicated hardware button for zooming into the center of the screen.**

#### *Touch-screen selections*

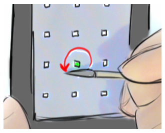
To minimize accidental selections for the touch-screen techniques, we use a quick touch-and-release action (referred to in previous work as "lift-and-tap" [12] or "click" [14]) that must be completed within 250 ms. This allows the user to release the finger from the surface in-between successive dragging gestures, before the final touch-and-release that triggers a selection.

#### *MobileRub*



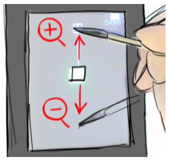
Rub-Pointing [14], enables zooming that is controlled through small diagonal rubbing gestures over the target, while maintaining contact with the screen. The actions are distinguished by the slope of the diagonal; rubbing along the diagonal with a positive slope zooms in incrementally for each stroke, while rubbing along the diagonal with negative slope resets the zoom level. Two strokes are necessary to detect the rubbing, after which each subsequent stroke magnifies the screen around the rubbing location. The rub strokes had to be 3–100 pixels in length and each stroke must happen within 500 ms of the previous stroke. A 1.5 magnification factor was experimentally determined to be suitable for this system. The gestures are flipped for left-handed users, as in the original version [14].

#### *MobileGesture (handheld)*



We implemented Wobbrock et al.'s $1 gesture toolkit [25] in Java ME, and inspired by previous work [13, 21], we use it to detect circular gestures. Counterclockwise circles zoom up to the extent of the circle, while clockwise circles reset the zoom level. The gestures are reversed for left-handed users.

#### *MobileDrag (handheld)*

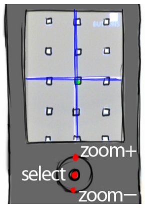


Simple dragging gestures control zooming for MobileDrag. The user zooms in on the desired location by touching the location and dragging upwards or downwards, to zoom in or out, respectively. The zoom factor is based on the distance the user has dragged with the stylus.

We calculate the zoom factor as $z_f = 1 + s \times \Delta x^2 \times \Delta y^2$, where $\Delta x$ is the horizontal dragging distance, $\Delta y$ is the vertical dragging distance, and $s$ is an experimentally determined constant that we set to 0.0015.

MobileDrag achieves its simplicity by trading ease of use for flexibility, since its zooming mode cannot simultaneously coexist with other touch-screen actions (such as panning).

#### *MobileButtons (handheld)*



MobileButtons is the only technique that is not based on touch-screen interaction, and the target acquisition is thus dependent on the coarse tracking from the LightSense camera. It uses three tactile buttons to trigger distinct actions. The device's *up*-key zooms in on the location indicated by a crosshair in the center of the screen, the *down*-key resets the zoom level, and the *center*-button selects the pixel under the crosshair.

### Baseline interaction techniques on large touch-screen

As baselines for comparison, we implemented two existing techniques, in which direct finger-interaction was used on the large touch screen.

#### *Baseline 1: Zoom-Pointing*

Variations of Zoom-Pointing, a tool-based technique, can be found in many applications that support zooming.

After entering zoom mode by pressing the zoom button, the user drags out a rectangle around the target, to which the system then zooms. The procedure can be repeated until the target is sufficiently large to be unambiguously selected through direct touch. The view can be reset by pressing a dedicated reset button.

We slightly modified the version used by Albinsson and Zhai [1], by placing the two buttons in the lower-right and lower-left corners of the screen (for right- and left-handed users, respectively), rather than in the upper-right and upper-left corners. This made it easier to reach the buttons and minimized occlusion problems.

### *Baseline 2: Rub-Pointing*

Rub-Pointing is similar in its behavior to MobileRub, but uses the finger directly on the large touch screen. The same constants as in the original description of the technique [14] were applied (3 pixels < rub stroke length < 50 pixels, within 500 ms). The magnification factor was set to 1.5 for each stroke.

## USER STUDY

We investigated the potential of a tracked mobile touch-screen device as a means for improved interaction through a user study where the six different interaction techniques are used for zooming and selection of small targets on a rear-projected surface.

### Participants

Ten male and two female participants were recruited from students at our institution, none of whom had previous experience with our interaction techniques. They were 21–31 years old (mean = 25.2, $\sigma = 2.8$) and all but one were right-handed. Eight had used large touch-screens a few times, while three used them often. Seven had used small touch-screens a few times, while four used them often. One had never used touch-screens. Four owned touch-screen PDAs or mobile phones. All participants were positive about touch-screens before the study, and most commented that touch-screens were more direct, easier and faster to use than conventional interfaces.

### Design

We used a repeated measures within-subject factorial design (6×5), where each participant individually tested each of the six interaction techniques ten times for each of five target sizes (1, 2, 4, 8, and 16 pixels = 0.51, 1.02, 2.04, 4.08, and 8.16 mm).

### Procedure

The participants were asked to select alternating square green targets of varying sizes in a reciprocal 1D pointing task, similarly to the procedure used in recent touch-screen studies [1, 4, 14]. The targets appeared 250 pixels apart on a black background, with only the active target visible at a given time. Since it could be hard to see the target sometimes, a grey square outline (200×200 pixels) indicated the target location. The target was also surrounded by a rectangular grid of light grey squares of the same size as the target, which were spaced five target widths apart.

The participants were instructed to select the target as quickly and as accurately as possible, with an emphasis on avoiding errors. They were informed of the select, zoom in, and zoom out actions. Auditory feedback was provided through low- and high-frequency beeps for errors and successful selections, respectively.

After first filling out a background questionnaire, the participants went through the following procedure for each of the six techniques:

1. A demonstration, where the experimenter explained and showed the technique in five trials.
2. Ten practice trials, where successful selections were enforced.
3. Ten practice trials, which mimicked the behavior of the real test. Regardless of a correct or incorrect selection, the system would move on to the next target.
4. Fifty test trials (ten trials for each of five target sizes), where error rates and completion times were recorded.
5. The participant provided qualitative feedback in a questionnaire. They were instructed to provide individual feedback for the technique and not make comparisons with other techniques at this point. Although there was a risk that previous techniques would influence the rating, we found it necessary to present the questionnaire after each technique, due to the number of different techniques, their similarity, and the length of the experiment.

The order in which the techniques were presented was randomized for the participants, and the order in which the five target sizes appeared was randomized in each block:

| | | | | |
|---|---|---|---|---|
| | 2 trials | × 5 widths = | | 10 practice trials (must succeed) |
| + | 2 trials | × 5 widths = | | 10 practice trials (similar to test) |
| + | 10 trials | × 5 widths = | | 50 test trials |
| | | | | 70 trials |
| | | | × | 6 techniques (2 baselines + 4 handheld) |
| | | | | 420 selections / participant |

At the end of the study, the participant filled out a final qualitative questionnaire, in which they were asked to compare the different techniques. The entire experiment lasted from 75 to 120 minutes for each participant.

### Hypotheses

Prior to the experiment, we formulated the following hypotheses:

$H_0$: There will be no differences in error rate.

$H_1$: There will be no differences in error speed.

$H_2$: Rub-Pointing and Zoom-Pointing will be faster than the handheld techniques, all of which have an additional step, in which the target must be framed by the mobile device.

$H_3$: The handheld techniques will have fewer errors, due to more precise stylus interaction, higher input resolution, and higher visual fidelity.

$H_4$: MobileButtons will be the slowest handheld technique, since it is dependent on the coarse LightSense tracking for aligning the crosshair in the center of the screen with the target.

## Quantitative results

We first analyzed our data to identify outliers based on excessively short completion times or long distances between the place of selection and the target location. We chose to not include trials that were shorter than 0.986 s (45 unsuccessful trials = 1.25% of all trials) or selections that were triggered more than 0.5 screen widths (400 pixels) from the target (32 unsuccessful trial = 0.89 % of all trials). We analyzed the error rate and median completion times of the remaining 3523 trials using $\alpha = 0.05$ for significance.

### *Error rate*

We reject $H_0$ since an analysis of variance (ANOVA) indicated that technique ($F_{5, 55} = 6.44$, $p < 0.001$) and target size ($F_{4, 44} = 7.6$, $p < 0.001$) had a significant effect on mean error rates. There was also a significant interaction between technique and target size ($F_{20, 220} = 1.72$, $p < 0.05$). The Bonferroni-adjusted paired samples t-test did not, however, identify significance for any pairs, despite the variations in mean error rates (See Figure 5).

All of the mobile techniques, except MobileDrag (Mean 9.0%, SEM 2.4%), had mean error rates below 4% (MobileRub, Mean 3.8%, SEM 1.2%; MobileGesture, Mean 2.2%, SEM 0.9%; MobileButtons, Mean 1.7%, SEM 0.5%). Zoom-Pointing (Mean 5.8, SEM 1.9%) was better than MobileDrag and Rub-Pointing (Mean 15.5%, SEM 3.9%). While $H_3$ seems to hold for three of the mobile technique, the significance of the differences could not be verified and we thus do not accept $H_3$.

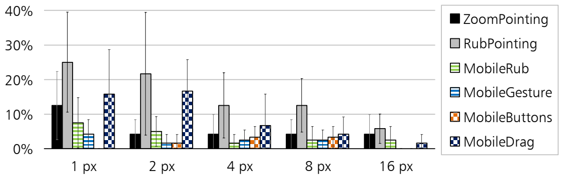


**Figure 5. Mean error rates with 95% confidence error bars for the five target sizes.**

### *Completion time*

As in previous studies [4, 14], we analyzed the variance of median completion times for successful selections, rather than the mean, to minimize the influence of human response times. Since one user failed all trials for Rub-Pointing at the 2-pixel target size, we separated the analysis of the completion time data into two parts.

First, we analyzed the completion time for all techniques over all target sizes except 2-pixel targets. The ANOVA showed no significant effect of technique ($F_{5, 55} = 1.57$, $p = 0.184$) on completion time. There was however a significant effect of target size ($F_{3, 33} = 85.06$, $p < 0.001$), which is likely due to the increased number of zooming actions and more precise targeting required for small targets. Figure 6 shows the small, but insignificant, differences among the techniques in performance, and the improved completion time as the target size increases. A significant interaction between technique and target size was also present ($F_{15, 165} = 1.92$, $p < 0.05$).

Second, we analyzed the completion time for all techniques except Rub-Pointing for 2-pixel targets, but found no significant effect of technique ($F_{4, 44} = 0.79$, $p = 0.541$).

We thus cannot reject $H_1$, and although the results indicate that MobileButtons was the slowest, and Rub-Pointing the fastest technique, these differences were not significant, and we thus cannot accept $H_2$ and $H_4$.

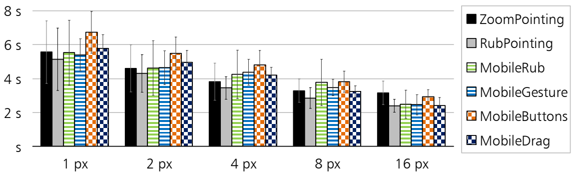


**Figure 6. Mean median completion times with 95% confidence error bars for the five target sizes.**

## Qualitative Feedback

After each set of test trials, participants provided individual feedback about the technique. They ranked the technique on a seven-point Likert scale (from −3 to 3) according to ten criteria (intuitiveness, ease of use, comfort, preference, speed, efficiency, fatigue, accuracy, level of frustration, and appropriateness for zooming), which were adapted from the NASA task load index [8] and the IBM Computer Usability Questionnaire [11]. They were asked to state what they liked or disliked about a particular technique, and were encouraged to leave additional comments. The results are summarized in Figure 7 in a median and mode plot. A Friedman test ($\alpha = 0.05$) was performed on the responses for each of the ten categories, but statistical significance was only found for accuracy ($\chi^2(5) = 18.706$, $p < 0.0001$). We analyzed this result with a Wilcoxon signed rank test, which was Bonferroni corrected, so the effects are reported at a 0.0031 significance level. The Wilcoxon test shows a borderline significance for Rub-Pointing's lower score compared to Zoom-Pointing ($z = -2.687$, $p = 0.04$), and a significantly lower score for Rub-Pointing compared to MobileButtons ($z = -3.089$, $p < 0.001$) for accuracy.

Most of the techniques received neutral to positive ratings

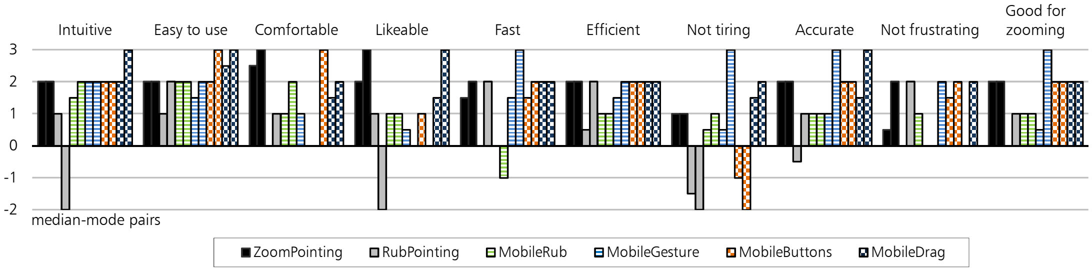


**Figure 7. Median and mode values of qualitative feedback provided for each technique by the participants. The techniques were ranked in ten categories on a seven-point Likert scale (-3 to 3).**

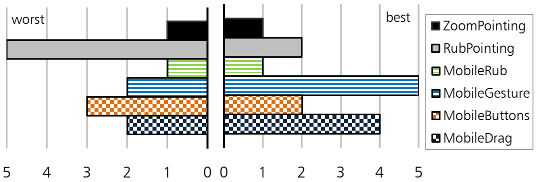


**Figure 8. The left and right graphs summarize the number of times a technique was selected as the least and most preferred, respectively.**

for the different criteria.

*Zoom-Pointing* was mostly appreciated (0.5 ≤ median ≤ 2.5, 1 ≤ mode ≤ 3) due to its ease of use and simplicity (eight participants), with the main complaint being the technique's fundamental requirement for pressing the zoom button each time the user wished to zoom (eight participants). Zoom-Pointing was ranked the highest for comfort (median 2.5, mode 3).

*Rub-Pointing* had, almost always, the lowest scores. Although there were comments that the technique was "easy" (two participants), "fast" (three participants) and "not tiring" (two participants), the main issue seemed to be discomfort, which eight participants complained about. Four participants found that it was hard to control and two complained about its inaccuracy. The median and mode plots illustrate its low scores regarding intuitiveness (median 1, mode −2), likeability (median 1, mode −2), fatigue (median −1.5, mode −2), and accuracy (median −0.5, mode 1).

*MobileRub*, on the other hand, made a better impression than its large touch-screen counterpart. It got high scores for intuitiveness (median 1.5, mode 2), ease-of-use (median 2, median 2), and vaguely positive scores for likeability (median 1, mode 1) and fatigue (median 0.5, mode 1). It was however ranked as one of the slowest techniques (median 0, mode −1).

*MobileGesture* got complaints regarding unrecognized circles, which clearly was frustrating as the gesture had to be repeated (seven participants). Nine participants, however, commented positively on its ease-of-use (median 1.5, mode 2) and intuitiveness (median 2, mode 2). Two participants found it problematic that the smaller the target was, the smaller and more accurately they had to circle it. One participant mentioned that it was difficult to remember the directions for zooming in and zooming out (counterclockwise/clockwise).

*MobileButtons* was regarded as accurate by nine participants, but as tedious by seven participants, since the buttons had to be pressed many times for small targets. Three participants commented on wrist pain, due to the way the device was horizontally placed on the surface while pressing the buttons. This explains the low score for fatigue (median −1, mode −2), while its simplicity was reflected in its high ranking for ease of use (median 2, mode 3).

*MobileDrag* received the highest score for ease of use (median 2.5, mode 3) and intuitiveness (median 2, mode 3), on which nine participants also made positive comments. Five participants mentioned that they would have wanted a higher zoom factor to avoid multiple strokes. Three participants appreciated that they could zoom in and out in the same mode. One participant found that the bottom part of the screen was the most effective location for the target to maximize the possible zooming in a stroke, but felt that this was counterintuitive compared to a central position.

### *Most and least preferred techniques*

At the end of the study, the participants filled out an exit questionnaire, selecting the techniques they liked the most and the least (they were allowed to select more than one for each category). The results are summarized in Figure 8. Rub-Pointing was the lowest ranked technique, with five participants selecting it as their least preferred and two selecting it as their most preferred. Zoom-Pointing and MobileRub received the least attention with one positive and one negative vote each. MobileButtons had three users who disliked it the most, and two who liked it the most. MobileGesture and MobileDrag were the highest ranked techniques, with five and four participants, respectively selecting them as their most preferred techniques. Two participants each selected them as their least preferred technique.

### *Ranking of mobile techniques*

The participants were also instructed to compare the mobile techniques, and pick the best and worse techniques according to five criteria (ease of use, speed, comfort, enjoyment, and performance). Here, MobileGesture was also ranked highest, followed by MobileDrag and MobileRub, with MobileButtons ranked lowest, as shown in Figure 9.

## Discussion

### *The large touch-screen techniques*

The results indicate that several factors affected the performance of the large touch-screen techniques, including the sensing hardware, its physical configuration and individual characteristics of the subjects.

Our previous study [14] used a 15" XGA (1024×768) touch-screen, which was vertically mounted, and tilted 15° backwards for better comfort. The lower visual and input

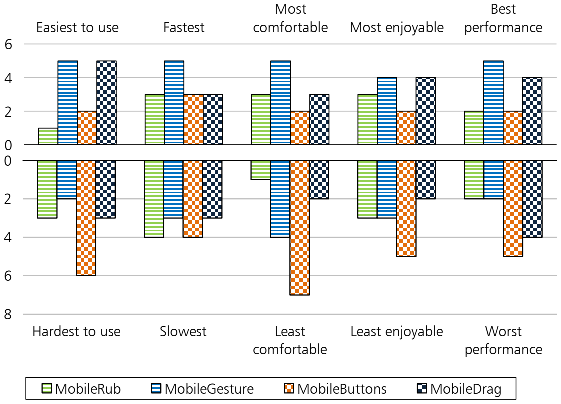


**Figure 9. The mobile techniques were also compared in five categories. The top and bottom graphs summarize the number of times a technique was selected as the best and worst for a category, respectively.**

Olwal, A., and Feiner, S. Spatially Aware Handhelds for High-Precision Tangible Interaction with Large Displays. *Proceedings of TEI 2009 (International Conference on Tangible and Embedded Interaction)*, Cambridge, UK, February 16-18, 2009, pp. 181-188.

resolution of the setup in this study (800×600 on a 20" screen) both results in pixels that are approximately 70% larger, and a coarser sampling grid for touch input. This means that participants had to make somewhat longer rubbing gestures in this setup, since the rubbing algorithm uses a minimum stroke length of three input pixels. A horizontal touch-screen surface might also vary in how well it ergonomically matches different users when rapid gestures are to be performed. For example, some participants explicitly mentioned wrist pain, as discussed previously. The manufacturer of the touch-screen overlay reports a "3H pencil hardness scratch resistance" and an activation force of 50-120 g/cm$^2$, which could indicate a deliberate design choice, where prioritizing robustness and durability results in less responsiveness. These factors make it more difficult to stay in contact with the screen while performing repetitive dragging gestures, which made Rub-Pointing suffer more than Zoom-Pointing from these surface qualities. The considerably better performance of MobileRub, which has virtually no friction between the stylus and the screen, and has a higher visual and input resolution, seems to confirm these arguments.

The parallax caused by the 17 mm offset of the touch-sensitive overlay from the projected graphics (not present in the previous study [14] or for the mobile techniques) is another factor that probably affected the results for Zoom-Pointing and Rub-Pointing; however, participants did not mention this explicitly. For example, an incorrect selection that a participant perceived as being a hit because of parallax error would have been recorded as a miss, increasing the error rates for both large-screen techniques.

Rub-Pointing also had the largest variations in how well different participants were able to use the technique in this hardware setup. This is indicated by the lack of statistical significance for its, on average, lower performance, and its larger confidence intervals and standard errors, which, on average, were 2–9 times larger than those of the other techniques (See Figure 5).

### The mobile techniques

MobileDrag's high error rate for small targets (on average 2.4–5.3 times higher than the other mobile techniques) was apparently related to its dependence on vertical space above the target for zooming. If space was limited, the user could make repeated short upwards strokes, instead of repositioning the device to increase the space. However, as some participants made these strokes increasingly faster, their short length and fast execution time classified them as the quick touch-and-release action, resulting in accidental selections.

The performance of MobileButtons was encouraging (average error rate 1.7%), as the crosshair targeting was done by moving the device, and was therefore dependent on the coarse camera tracking. The somewhat slower completion time (on average 4.9s compared to 4.2s for the other mobile techniques) can probably be attributed to more need for repeated zooming, discrete button presses, and more careful placement of the device.

Even in this fairly simple test scenario, the respective advantages and disadvantages of the mobile techniques made enough difference to create individual variations in preference. While the mobile techniques seem to perform as well as, or better than the baselines, the participants' ratings did not identify a superior technique with statistical significance.

Figure 9 shows how each mobile technique was selected by at least one, and at most five, participants as the best in each category. Similarly, at least one, and at most seven, participants selected a given technique as the worst in each category. While MobileGesture and MobileDrag got picked as the top techniques in each category, there were still at least two participants who felt they were the worst. MobileButtons, on the other hand, was considered as the worst by most participants, but had at least two participants in each category who thought it was the best. MobileRub got fairly symmetrical scores, with some participants liking it the most, and some liking it the least.

An important advantage of the mobile device is its exclusive linkage to its owner. While it is not practical for a large public display, used by many different people, to be reconfigured for each new user, this is not the case for a mobile device. Just as users customize and personalize their devices (e.g., using alerts, menu shortcuts, bookmarks, and themes), it might also be reasonable to provide a set of interaction styles from which to choose. This can already be seen on several mobile devices, such as the Sony Ericsson P1i, where the user has simultaneous access to a physical QWERTY keyboard, a numerical keypad, handwriting recognition, and an on-screen keyboard. The user can switch the text input method of choice at any given time.

We can, in fact, support a similar scenario for touch-screen interaction, since the modes of our best-performing mobile interaction techniques (MobileGesture, MobileRub, and MobileButtons) do not conflict. This makes simultaneous and concurrent operation possible, in addition to other touch-screen interaction, such as panning.

## CONCLUSIONS

We have developed an experimental testbed for the design and evaluation of fast and precise interaction techniques for spatially aware devices. We compared interaction using a tracked mobile device, with baseline techniques employing direct finger-interaction, for zooming and selecting of small targets on a larger display.

Our results indicate that characteristics of the hardware configuration and the commercial touch overlay in our setup negatively affected the performance of the baseline techniques, apparently due to a combination of lower resolution, parallax, friction and ergonomics. Although technologies exist that are superior to the transparent touch overlay we used, it may not always be feasible to employ them. Environmental constraints may encourage the use of robust, break-proof, scratch-resistant surfaces for discrete presses, over ones that support fluid and continuous multi-touch manipulations. Commercial availability and cost-effectiveness also influence the choice.

Our user study indicates that a coarsely tracked mobile device can overcome such constraints by providing precise control through tactile buttons or precise touch-screen interaction. Three of our mobile techniques, MobileGesture, MobileRub and Mobile Buttons, show consistently low error-rates (on average 1.7–3.8%). We further note that an implementation of the very same interaction technique on the tracked mobile device was able to avoid problems caused by the hardware configuration, when comparing MobileRub and Rub-Pointing.

While the results from our qualitative analysis indicate a general preference for mobile techniques in this setup, individual preferences seem to strongly affect the technique of choice. The nature of the three best-performing techniques, however, allows them to coexist in a single mode, which makes it possible to let the user seamlessly alternate between preferred mobile techniques.

## FUTURE WORK

We find it interesting that the coarse tracking of the mobile device did not seem to affect the precision with which users were able to interact. This indicates that these techniques could also be applied to novel, low-resolution sensing technologies that are more compact than our current experimental setup (e.g., light sensors [15, 10]).

Based on the results from the study, we identified some possible redesigns of our techniques that could improve performance, accuracy, and usability. We plan to update the detection of "tap"/"click"-selections to address the accidental errors caused by quick strokes in MobileDrag. We are also experimenting with different zooming controls, such as continuous zooming for MobileButtons and elastic zooming for MobileDrag, to improve ergonomics.

We are investigating the combination of multiple techniques in a single mode, both as alternative mappings for the same action, and with different actions, to provide additional functionality. An extension of this work could include more complex interaction techniques, which combine finger-interaction on the surface, with interaction on the mobile device.

We are also interested in applying these insights to multi-touch devices, including interactive surfaces, as well as handhelds. This could expand the design space for highly interactive user interfaces, by exploiting the richer set of input available from both device types.

## ACKNOWLEDGMENTS

We thank Ericsson, Mitsubishi, and Microsoft Research for supporting our work with mobile phones, the projector, and the camera, respectively. Thanks also to Lars Winkler Pettersson for feedback and comments on the paper, and to Areti Goulati for the administration of the user study.

## REFERENCES

1. Albinsson, P-A. and Zhai, S. High precision touch screen interaction. *Proc. CHI '03* (2003), 105–112.
2. Ballagas, R., Borchers, J., Rohs, M., and Sheridan, J.G. The smart phone: A ubiquitous input device. *Pervasive Computing, IEEE*, vol.5, no.1 (2006). 70–77.
3. Baudisch, P., Good, N., Bellotti, V., and Schraedley, P. 2002. Keeping things in context: A comparative evaluation of focus plus context screens, overviews, and zooming. *Proc. CHI '02* (2002), 259–266.
4. Benko, H., Wilson, A., and Baudisch, P. Precise selection techniques for multi-touch screens. *Proc. CHI '06* (2006), 1263–1272.
5. Benko, H., Ishak, E. W., and Feiner, S. 2004. Collaborative mixed reality visualization of an archaeological excavation. *Proc. ISMAR '04* (2004), 132–140.
6. Fitzmaurice, G. W. Situated information spaces and spatially aware palmtop computers. *Commun. ACM* 36, 7 (Jul. 1993), 39–49.
7. Hardy, R. and Rukzio, E. Touch & interact: Touch-based interaction of mobile phones with displays. *Proc. MobileHCI '08* (2008), 245–254.
8. Hart, S. G., and Staveland, L. E. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In Human Mental Workload (1988), 239–250.
9. Henrysson, A. and Ollila, M. UMAR: Ubiquitous mobile augmented reality. *Proc. MUM '04* (2004), 41–45.
10. Hodges, S., Izadi, S., Butler, A., Rrustemi, A., and Buxton, B. ThinSight: Versatile multi-touch sensing for thin form-factor displays. *Proc UIST '07* (2007), 259-268.
11. Lewis, J. R. IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *Int. J. Hum.-Comput. Interact.*, 7(1). (1995), 57–78.
12. MacKenzie, I.S. and Oniszczak, A. A comparison of three selection techniques for touchpads. *Proc. CHI '98* (1998), 336–343.
13. Moscovich, T. and Hughes, J. F. Navigating documents with the virtual scroll ring. *Proc. UIST '04* (2004), 57–60.
14. Olwal, A., Feiner S., and Heyman, S. Rubbing and tapping for precise and rapid selection on touch-screen displays. *Proc. CHI '08* (2008), 295–304.
15. Olwal, A. LightSense: Enabling spatially aware handheld interaction devices. *Proc. ISMAR '06* (2006). 119-122.
16. Reilly, D., Rodgers, M., Argue, R., Nunes, M., and Inkpen, K. Marked-up maps: Combining paper maps and electronic information resources. *Personal Ubiquitous Comput.* 10, 4 (2006), 215–226.
17. Rekimoto, J. A multiple device approach for supporting whiteboard-based interactions. *Proc. CHI '98* (1998), 344–351.
18. Rekimoto, J. and Nagao, K. The world through the computer: Computer augmented interaction with real world environments. *Proc. UIST '95* (1995), 29–36.
19. Rohs, M., Schöning, J., Krüger, A., and Hecht, B. Towards real-time markerless tracking of magic lenses on paper maps. *Adjunct Proc. Pervasive '07 (2007)*, 69–72.
20. Rukzio, E., Broll, G., Leichtenstern, K., and Schmidt, A. Mobile interaction with the real world: An evaluation and comparison of physical mobile interaction techniques. *Proc. AmI-07* (2007), 1–18.
21. Smith, G. and schraefel, mc. The radial scroll tool: Scrolling support for stylus- or touch-based document navigation. *Proc. UIST '04* (2004), 53–56.
22. Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T., and Schmalstieg, D. Pose tracking from natural features on mobile phones. *Proc. ISMAR '08* (2008), 125–134.
23. Wagner, D. and Schmalstieg, D. First steps towards handheld augmented reality. *Proc. ISWC '03* (2003), 127–135.
24. Wilson, A. D. and Sarin, R. BlueTable: Connecting wireless mobile devices on interactive surfaces using vision-based handshaking. *Proc. GI 2007* (2007), 119-125.
25. Wobbrock, J. O., Wilson, A. D., and Li, Y. Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes. *Proc. UIST'07* (2007), 159–168.
26. Yee, K. Peephole displays: Pen interaction on spatially aware handheld computers. *Proc. CHI '03 (2003)*, 1–8.