Vincent, T., Nigay, L., Kurata, T. Precise pointing techniques for handheld Augmented Reality. In *Proc INTERACT 2013*, LNCS 8117, IFIP-Springer (2013). pp. 122-139.
http://dx.doi.org/10.1007/978-3-642-40483-2_9
Author Version. The original publication is available at www.springerlink.com.

Precise pointing techniques for handheld Augmented Reality

Thomas Vincent[1], Laurence Nigay[1], Takeshi Kurata[2]

[1] Joseph Fourier University, UJF-Grenoble 1,
Grenoble Informatics Laboratory (LIG), UMR 5217, Grenoble, F-38041, France
[2] Center for Service Research, AIST, Tsukuba Central 2, 1-1-1 Umezeno,
Tsukuba, Ibaraki, 305-8568 Japan

{thomas.vincent, laurence.nigay}@imag.fr, t.kurata@aist.go.jp

Abstract. We propose two techniques that improve accuracy of pointing at physical objects for handheld Augmented Reality (AR). In handheld AR, pointing accuracy is limited by both touch input and camera viewpoint instability due to hand jitter. The design of our techniques is based on the relationship between the touch input space and two visual reference frames for on-screen content, namely the screen and the physical object that one is pointing at. The first technique is based on Shift, a touch-based pointing technique, and video freeze, in order to combine the two reference frames for precise pointing. Contrastingly -without freezing the video-, the second technique offers a precise mode with a cursor that is stabilized on the physical object and controlled with relative touch inputs on the screen. Our experimental results show that our techniques are more accurate than the baseline techniques, namely direct touch on the video and screen-centered crosshair pointing.
Keywords: Handheld Augmented Reality, Interaction Techniques, Pointing.

1 Introduction

While still an open research area, Augmented Reality (AR) in terms of superimposition of graphics is now possible on camera-equipped handheld devices. However, issues related to interaction still need to be studied. In particular, pointing at physical objects through the live video playback of a handheld device with either direct touch or a screen-centered crosshair has limited accuracy [4, 11]. Nevertheless accurate pointing at physical objects would benefit several handheld AR applications including selection or in-situ positioning of digital annotations in dense physical environments such as paper maps.

Pointing accuracy in handheld AR is limited by various factors. First, interaction with handheld devices brings specific constraints [17]: the screen real estate is limited

and direct touch on the screen, the de-facto standard input modality on such devices, is impaired by finger occlusion and an ambiguous selection point (i.e. the ”fat-finger”

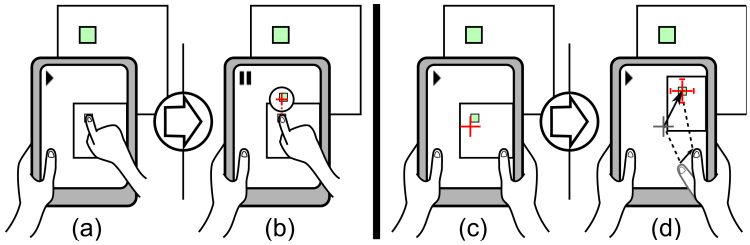

Fig. 1. Four techniques for pointing at physical objects through video on handheld devices: (a) *Direct Touch* on the live video; (b) *Shift&Freeze:* Shift [19] combined with freeze-frame; (c) Screen-centered *Crosshair;* (d) *Relative Pointing* with cursor stabilized on the physical object.

problem). Moreover, when considering handheld tablets which are larger but also heavier than phones, the trade-off between device handling (i.e., one or two handed hold) and touch interaction (i.e., available fingers for touch interaction and screen accessibility) need to be taken into account [20]. To address these issues, pointing accuracy on handheld devices has been studied and different techniques have been proposed [17, 19]. Yet, those techniques do not take into account the specificities of AR.

Indeed, in AR applications, when interacting through the video on handheld devices, the physical object one wants to point at is not stable on screen. As a consequence, it is also not stable within the touch input space. As the viewpoint is controlled by the device’s pose, its stability is impaired by hand tremor and motion induced by the user’s touch inputs. Furthermore, when using handheld tablets for AR applications, the trade-off is between viewpoint stability (i.e. a steady hold) and touch-screen accessibility. On the one hand, a steady hold with both hands (figure 1-c/d) only allows touch interaction with the thumbs in a limited region of the screen. On the other hand, when holding the tablet with one hand (figure 1-a/b), the other hand can interact with a larger area of the screen at the expense of more instability.

For handheld AR systems, pointing with a screen-centered crosshair has been studied [14, 15]. This technique is impaired by viewpoint instability. Freeze-frame techniques have been used to improve direct touch interaction by pausing the video [4, 10]. Nevertheless, one drawback of this approach is to prevent an up-to-date view of the physical scene.

To address the limitations of pointing for touch-based handheld AR systems, we propose two techniques:

- *Shift&Freeze* (figure 1-b) that addresses both direct touch accuracy limitations and viewpoint instability by combining Shift [19] with freeze-frame. Shift is a technique that extends *Direct Touch* pointing (figure 1-a) with a precise quasi-mode. We complemented this precise quasi-mode with freeze-frame to adapt Shift to

handheld AR. So *Shift&Freeze* improves accuracy while still allowing direct touch for coarse but fast pointing.

- *Relative Pointing* (figure 1-d), which improves pointing accuracy without pausing the live video playback. To improve accuracy, it uses a cursor that is stabilized in the physical object's frame of reference. The cursor is controlled by indirect relative touch inputs. As such, relative pointing in handheld AR does not share direct touch limitations and the cursor position is not impaired by viewpoint instability. Moreover this technique extends the screen-centered *Crosshair* pointing technique (figure 1-c) with a precise mode. This allows both coarse but fast pointing and accurate pointing when needed.

In this paper, we first review related work and then present the design rationale of our two techniques, *Shift&Freeze* and *Relative Pointing* in handheld AR. We then report two experiments comparing our techniques and the baseline techniques, namely *Direct Touch* on the video and screen-centered *Crosshair*. We conclude with a discussion of our results and directions for future work.

2 Related Work

We build on previous work on pointing techniques for touch-based handheld devices, as well as on pointing techniques for handheld AR and spatially aware interfaces.

2.1 Pointing techniques for touch-based handheld devices

Much prior work has addressed how to improve pointing accuracy on touch-screen. Within the scope of our work, we examine pointing techniques on touch-based handheld devices that do not require prior knowledge of the targets, excluding for instance target expansion techniques as in *Starburst* [3]. Indeed for the case of in-situ positioning of annotations in handheld AR (e.g., annotations at any position on paper maps), there is no available knowledge of possible targets.

A first approach is zooming to enlarge the information space to a scale appropriate for accurate pointing [2]. When using zooming, the user is facing the classical trade-off between the level of zoom (for accurate pointing) and the visible context (for finding the area of interest). The interaction process can be quite tedious on handheld devices with limited screen real estate: zoom in to focus and zoom out for context. Based on zooming, *TapTap* [17] increases pointing accuracy. Two taps on screen are performed for pointing. The first tap selects a coarse area on the screen that is displayed zoomed in a pop-up view centered on the screen. The second tap performs the precise selection in the zoomed area and closes the pop-up view.

A second approach is to display a cursor to address both finger occlusion and selection point ambiguity. Potter et al. [12] proposed *Take-Off* that enables pointing adjustment and avoids finger occlusion by showing a cursor slightly above the finger position. One drawback of this technique is that the user does not know the position of the cursor until her/his finger touches the screen. Building on this technique, *Shift*

[19] extends direct touch pointing with a precise quasi-mode. While in this mode, *Shift* displays a circular callout showing a copy of the screen area occluded by the finger and places it in a non-occluded location. The callout also shows a cursor representing the selection point of the finger, whose position can be adjusted by moving the finger. Validation is performed on finger lift. *MagStick* [17] also extends direct touch pointing by using a telescopic stick metaphor to enable further adjustment. When the finger touches the screen, it defines a reference point; then, by dragging the finger away from the reference point, the user can extend a telescopic stick centered on the reference point with the finger at one end and the cursor at the other end.

2.2 Pointing techniques for handheld AR and spatially-aware interfaces

Seminal works on handheld AR like *NaviCam* [13] have paved the way for an active research area. For example, *TouchProjector* [4] allows users to move pictures on remote screens through direct touch on the live video of a handheld device. Handheld AR systems augmenting different kind of objects such as sights [1], printed conference proceedings [11], photo books [8] or paper maps [16] have been developed.

In handheld AR settings, the viewpoint in the augmented scene is usually controlled by the absolute device's pose in space (controlling the back-face camera viewpoint). As a handheld device is not self-stabilized (as opposed to the mouse for example), its pose is subject to hand jitter as for other freehand interaction techniques like laser pointers or handheld projectors [7]. As a consequence, the augmented scene the user interacts with is not stable on the screen. This is different from typical GUI situation on either desktop or handheld devices (see previous section) where the objects the user wants to interact with usually remain still on the screen during the interaction.

With such settings, pointing is usually performed with either a screen-center crosshair [8, 11, 14-16] or by direct input on the screen, using a pen or bare fingers [4, 8-11, 18]. Rohs et al. [14, 15] studied pointing with a screen-centered crosshair on a phone. They showed that the performance of this technique could be modeled with a two parts Fitt's law: physical pointing (i.e. moving the device in space) and virtual pointing (i.e. when the target is visible on screen). Hand jitter impairs accuracy of those interactions and different strategies have been proposed to improve interaction with handheld AR settings.

A first strategy is to increase target size on the screen by coming closer to the physical object or by zooming the live video. Zooming is compatible with both screen-centered cursor and direct input as well as with other strategies for improving interaction. *TouchProjector* user study shows that automatic zooming was overall the best performing technique: While zooming improves interaction, it does not render the image steady. The study also highlights that for precise manipulation a freeze-frame mode (which also performs automatic zooming) outperforms automatic zooming alone.

The freeze-frame technique belongs to the second strategy that overcomes viewpoint instability due to small hand motions. Indeed when pausing the live video, the viewpoint becomes steady. Freezing the frame also allows moving to a comfortable position for interaction. This approach is not compatible with a screen-centered cur-

sor, but it has been proven useful to improve pen and touch interaction. In a user study, Lee et al. [10] showed that a video freeze mode improves accuracy for a drawing task with a pen through the handheld device video frame. They also noted that some users become lost when the live video is resumed as the viewpoint has changed. Another issue of freezing the frame is that the scene is no longer updated. *TouchProjector* overcomes this issue by updating the video snapshot with a digital copy of the remote screen one is interacting with. Unfortunately, a digital copy of the object of interest is not available for all AR scenarios. Freeze and zoom can be combined as previously explained in the case of *TouchProjector* and as in *TapTap* combined with video freeze for handheld AR [18]. In the latter, the combination of video freeze and zoom is a 'once' mode rather than a truly persistent mode. Nevertheless, any selection then requires two taps. Another way to stabilize the viewpoint is to use 'loose registration' as in *PACER* [11]. To interact with paper documents, they propose to display a digital copy of the document on the handheld device screen instead of the live video. This relaxes tracking requirements and allows for a coarse and filtered viewpoint to be used. Again, this requires a digital copy of the object of interest.

Finally, a third strategy consists of stabilizing inputs in the frame of reference of the physical object (or of its projection on the screen) rather than stabilizing the object of interest on the screen. *Snap-to-feature* [9] proposes to snap touch input on features of physical objects detected in the live video. This allows for better drawing of contours of physical objects on the screen without relying on freeze-frame or a digital copy of the scene. Our *Relative Pointing* technique is based on a similar strategy but does not rely on detecting features of the physical objects in the live video.

As opposed to on-screen content stabilization techniques that sever the live relation with the surrounding or use a digital copy of the scene, input stabilization offers the opportunity to improve accuracy without loosing the live relation with the surrounding. Both strategies can be complemented with zooming. Those approaches address limitations specific to the handheld AR context but not necessarily limitations of touch inputs. This is the challenge we addressed by designing *Shift&Freeze* and *Relative Pointing*, two pointing techniques that we introduce in the next sections.

3 Handheld AR Pointing

3.1 Design rationale

To systematically analyze the issue of accurate pointing for handheld AR, we base our study on the relationships between the touch input space and two frames of reference for on-screen content: that of the screen and that of the physical object of interest.

With video freeze, the physical object's frame of reference is fixed within the image plane and thus it is stable on the screen (figure 2-b). This case is similar to GUI interfaces: Existing pointing techniques for handheld devices can be combined with video freeze. *Shift&Freeze* combines the existing pointing technique Shift [19] with video freeze.

When interacting through live video, the physical object is jittery on the screen (figure 2-a). In this case, we consider (see figure 3): (1) whether pointing is performed with or without an instrument (i.e. a cursor), and (2) in which frame of reference pointing is performed - either the frame of the screen or the frame of the physical object.

Direct Touch is the case of pointing in the screen frame without a cursor. Screen-centered *crosshair* makes use of a cursor and points in the screen frame. Those two techniques (with and without a cursor) are impaired by hand jitter as pointing occurs in the screen frame where the physical object of interest is not stable (figure 2-a).

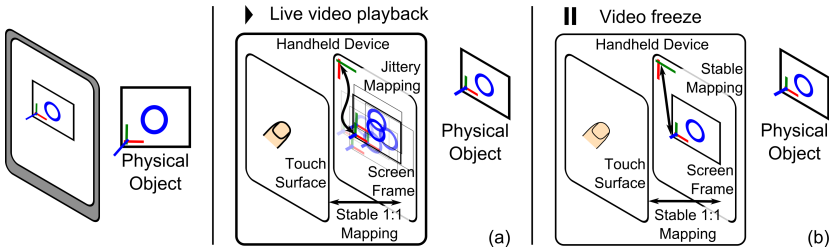

Fig. 2. Relationships between the frames of reference of touch input, of the screen and of the physical object on the screen: (a) with live video playback; (b) while the video is frozen.

If pointing was performed in the frame of reference of the physical object rather than in the screen frame, pointing accuracy would not be impaired by hand jitter. Pointing in the physical object's frame of reference without a cursor instrument implies interaction directly on the physical object. We did not consider this case and focused on interaction with the touch-screen of the handheld device. Moreover such physical interaction seems cumbersome while holding a handheld tablet. Our *Relative Pointing* technique is the case where pointing is performed with a cursor in the physical object's frame of reference. In this case, we use an indirect relative mapping of touch inputs to cursor motions in the frame of reference of the physical object. As such, the cursor position is not impaired by hand jitter.

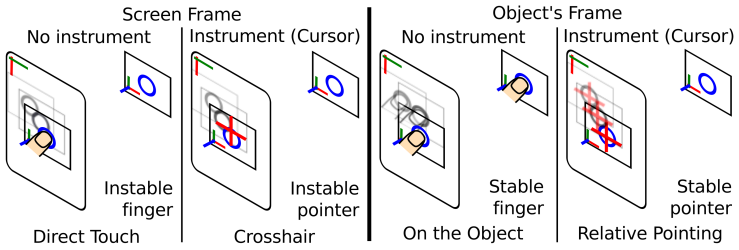

Fig. 3. Pointing through live video: four cases.

This analytical framework based on the spatial relationships between the input space and two visual output frames of reference guided the design of our two tech-

niques *Shift&Freeze* and *Relative Pointing*. Their respective designs result from a twofold strategy: *Shift&Freeze* is conceived as an improvement of *Direct Touch* and solves its accuracy problem by freezing the video and *Relative Pointing* is an improvement of *Crosshair* by adding a relative cursor stabilized on the remote physical object.

3.2 Shift&Freeze

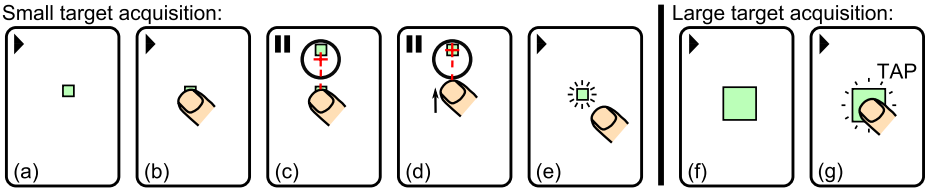

Fig. 4. *Shift&Freeze* walkthrough. (a-e) Small target acquisition with Shift and frozen video; (f-g) Large target acquisition with the *Direct Touch* technique (one tap on the screen).

Figure 4 shows a walkthrough of the two modes of our *Shift&Freeze* technique.

Scenario 1: (a) The user points the handheld device camera at the target so that the target appears on the screen. (b-c) After a short dwell time after finger contact, *Shift&Freeze* enters a precise quasi-mode and the video is frozen. A callout is placed above the finger and shows the area under the finger and a cursor at the current selection point position. (d) While in this mode, the video remains frozen and the user can adjust the position by moving its finger. (e) On finger lift, the target is selected and the live video playback is resumed.

Scenario 2: (f-g) For large enough targets where hand tremor and finger occlusion are not a problem, selection can be performed with a tap on the screen at the position of the target.

To cope with touch input limited accuracy, we chose to use *Shift* [19] for the following reasons. First, *Shift* does not require knowledge of existing targets to improve accuracy. Also, *Shift* extends *Direct Touch*, thus fast but imprecise pointing is still possible. Finally, similar techniques have been used to precisely place the cursor in text entry in commercial products and to improve accuracy when using 'loose registration' [11].

To cope with viewpoint instability in handheld AR settings, we combined *Shift* with freeze-frame. Touch-based pointing techniques in general and *Shift* in particular are designed for pointing at static targets on the screen. Instead of implementing freeze-frame as a mode, we complemented *Shift's* precise quasi-mode with video freeze. Compared to the original *Shift* technique, no extra user action is necessary to control video freeze/unfreeze. Nevertheless, as noted in [10], resuming the live video leads to a discontinuity of viewpoint that might disorient the user.

As such, the *Shift&Freeze* technique has the following properties: (1) By extending *Direct Touch,* this technique requires interaction overhead only when accuracy is required; (2) It allows precise pointing using *Shift's* callout on a frozen frame.

3.3 Relative Pointing

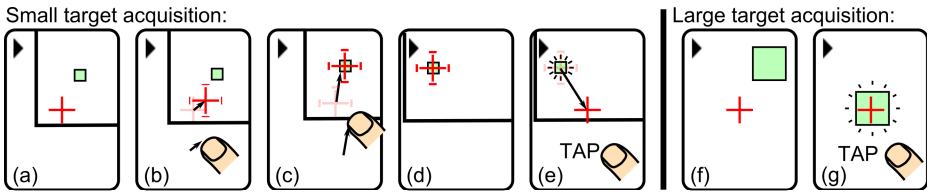

Fig. 5. *Relative Pointing* walkthrough. (a-e) Small target acquisition with a relative cursor stabilized on the physical object; (f-g) Large target acquisition with the *crosshair* technique.

Figure 5 shows a walkthrough of the two modes of our *Relative Pointing* technique.

Scenario 1: (a) The user points the handheld device camera at the target so that the target appears on the screen. (b) In order to mitigate the instability due to hand tremor, when the user touches the screen and starts moving its finger on the screen, a relative pointing mode is triggered. (c) While in this mode, the cursor is no longer bound to the screen center. Instead, it is stabilized on the remote physical object at its current position. The user fine-tunes the cursor position by controlling the cursor indirectly with finger gestures on the screen. (d) On finger lift, no special action is performed. (e) The validation of a position is performed with a tap on the screen. Upon validation, a short animation moves the cursor back to the screen center, thus leaving the relative pointing mode.

Scenario 2: (f-g) When acquiring large enough targets, hand tremor is not a problem. In this case, the user does not need to use the relative pointing mode and can trigger a target acquisition at the position of the screen-centered cursor with a tap on the screen. This is similar to the screen-centered *Crosshair* technique.

To make relative pointing effective for handheld AR context, the following issues have been addressed.

Combining Absolute Physical Pointing and Touch-Based Relative Pointing. As the device's pose controls the camera viewpoint, the target in the physical world needs to be placed in the camera field of view before interaction with it can start. So, cursor-based relative pointing needs to be combined with this absolute direct pointing in space. That is why we chose to extend the screen-centered *crosshair* pointing technique as it already uses a cursor and only relies on the device's pose for both viewpoint control and pointing. We extended this technique with a relative pointing 'once' mode where the cursor is no longer fixed at the center of the screen. Instead, the finger indirectly controls the cursor's position. This mode is triggered by finger motion on the screen. Lifting the finger from the screen does not deactivate this mode. This

allows both finger clutching and checking the current cursor position before validation. A tap on the screen triggers the pointing validation. It is possible to cancel this relative pointing mode by pressing a button. Also, when tracking is lost, the relative pointing mode is cancelled. Finally, the cursor is bound to the screen. In case a change in the camera's viewpoint or a finger motion would otherwise make the cursor invisible on the screen (i.e., outside the screen), the cursor is automatically moved so that it remains visible on the screen.

Transfer Function. When dealing with indirect relative input, the transfer function (figure 6) that maps input motions to cursor displacements in the visual output space is of particular interest.

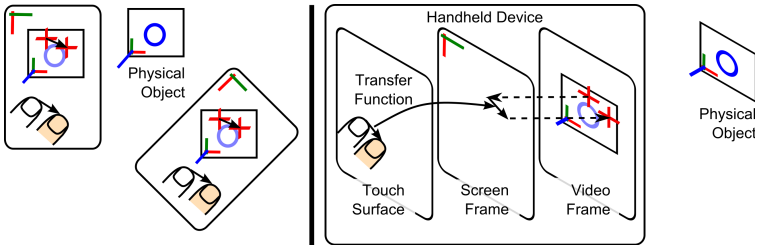

Fig. 6. With *Relative Pointing*: (left) Effect of screen rotation on the cursor's position; (right) The cursor is stabilized in the physical object's frame of reference and relative touch motions are applied on the screen.

First, a transfer function that maps touch motions in the screen frame directly to cursor displacements in the physical object's frame is not appropriate. In this case, the relative rotation between the device and the physical object's frame, the distance between the device and the physical object, and the zoom factor would affect the displacements of the cursor on the screen. Yet the user is looking at the physical object through the live video on the screen. Therefore the control loop is between the finger motions and the cursor displacements on the screen (and not in the physical object's frame).

Instead, for *Relative Pointing*, the transfer function is applied in the screen frame. When a finger motion input is received, the cursor position is projected from the physical object's frame onto the screen; the cursor displacement is applied on the screen; and the new cursor position is projected back onto the physical object (figure 6-right). This guarantees that the behavior of the cursor on the screen is consistent when the device is rotated (figure 6-left) and when the viewpoint or the zoom factor changes. In short, we use the physical object's frame of reference to stabilize the cursor and the screen frame of reference to apply cursor displacements.

A second question is which transfer function should be used. Transfer function is the place for interaction improvements such as adjusting the control-to-display ratio dynamically according to the input device speed. Dynamic transfer function is a default feature for the mouse and touchpad inputs in common desktop environments.

While dynamic transfer function has been studied for desktop environment, we are not informed of thorough evaluation of indirect mapping on handheld device's touch-screens [6]. A dynamic transfer function can be used with the *Relative Pointing* technique. For our developed technique and experiments, we used the transfer function of touchpad inputs on Mac OS X: `osx:touchpad?setting=0.875` [6]. With this configuration, the transfer function allows both reaching of most of the screen with fast movement and accurate positioning at lower speed.

To sum up, the *Relative Pointing* technique has the following properties: (1) By extending *Crosshair,* this technique requires interaction overhead only when accuracy is required. (2) By stabilizing the cursor on the remote physical object, it offers accuracy assistance without relying on video freeze.

4 Experiments

We ran two experiments to evaluate four handheld AR pointing techniques (the two baseline techniques and the two proposed techniques):

- *Direct Touch:* Pointing with selection at the finger press position;
- *Crosshair:* Screen-centered crosshair pointing where validation is triggered on finger press with a tap anywhere on the screen;
- *Shift&Freeze* with a 300 ms delay for escalation, a 44mm wide callout initially placed 22 mm above the initial touch position, and no zoom in the callout; and
- *Relative Pointing* as described above but without a cancel button.

All cursor-based techniques (i.e. *Crosshair, Shift&Freeze*'s callout and *Relative Pointing)* are using the same red square cross cursor, which is 7.7mm in size with a stroke width of 0.2mm.

The goal of the first experiment was to collect both user feedbacks and quantitative data on those techniques while performing a rather realistic task: placing marks on a wall map. Building on this experiment, we ran a second experiment to further evaluate those four techniques in a controlled experiment while acquiring small targets.

4.1 Experiment 1: User experience

For this experiment, we formulated the following hypothesis:

- H1: *Relative Pointing* and *Shift&Freeze* are preferred over *Crosshair* and *Direct Touch.* This is motivated by the extra precise mode offered by our two techniques. Moreover, this precise mode does not prevent the use of *Crosshair* or *Direct Touch* as a basic mode.
- H2: On tablet form factor, indirect cursor-based techniques are preferred over direct pointing techniques. So *Relative Pointing* is preferred over *Shift&Freeze* and *Crosshair* is preferred over *Direct Touch*. This is based on both the finger occlusion issue for direct touch input and the trade-off between tablet hold and screen accessibility.

Procedure, apparatus and participants. For each of the four techniques, we first explained the technique and participants had a chance to freely try it. Then, participants performed 5 different pointing tasks to place AR marks on a physical wall map with a handheld tablet. Each task was repeated three times. For each trial, participants started at 2.5m from the wall map. They were instructed to move freely in the room and to hold the tablet in portrait mode. Finally, a debriefing questionnaire and interview concluded each technique's experiment.

Before starting the experiment with the four techniques, participants started to perform each of the 5 tasks once with no interaction by only finding the required targets through the video on the tablet screen. This was introduced to help participants to become acquainted with the tasks and the experimental setting (in particular form factor and video quality). After this initial training, all participants started with the *Direct Touch* technique. The presentation ordering of the other three techniques was then counter-balanced across participants using a Latin-square. We used this design so that all participants share *Direct Touch*, the de facto standard interaction, as a common baseline. Experiments lasted about one hour including a debriefing discussion.

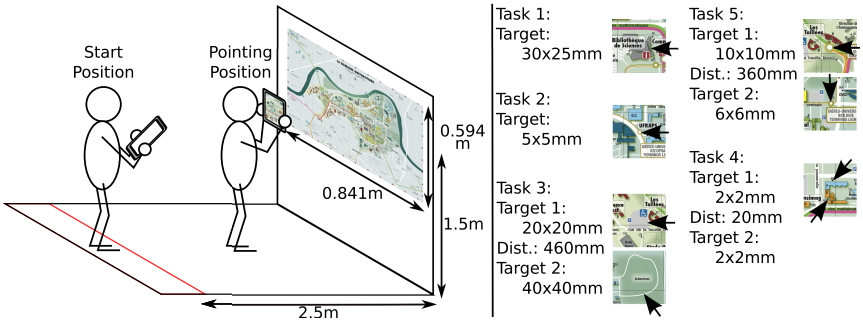

Fig. 7. Experimental set-up: (left) Participants started 2.5m away from the physical wall map and then could move freely to perform the pointing tasks; (right) the targets used for each task. Tasks 3, 4 and 5 consist of placing AR marks on 2 targets.

We use a map of our campus site in A1 format in landscape orientation (841mm x 594mm). It was placed vertically on a wall with the middle of the map 1.5m above the floor (figure 7-left). The targets of the 5 tasks are shown on the right of figure 7. Tasks 1 and 2 consist of placing a mark on a single target. Tasks 3, 4 and 5 consist of placing marks on two different targets. The target sizes range from 2mm to 4cm.

The experiment was conducted on an iPad2 (weight: 601g, screen resolution 1024x768 pixels (132 dpi)). The system provides touch input with the same resolution as the screen. We developed an ad hoc application for the experiment using OpenGL|ES 1.1[1], Vuforia SDK 1.5.9[2] and libpointing[3] [6]. This application runs at

[1] http://www.khronos.org/opengles/1_X/
[2] https://www.vuforia.com/

about 30 frames per second. The size of the camera images is 480x640 pixels and images are displayed full-screen. Statistical analysis was performed with R[4].

Twelve unpaid volunteers (4 female, 8 male; 1 left-handed and 1 ambidextrous), ranging in age from 22 to 45 years (mean 27 years), were recruited from our institution. All participants had previous experience with touch-based handheld devices (seven on a daily basis) amongst whom nine had used a handheld tablet before.

Results.

User preference. The questionnaire was composed of 7 questions taken from the System Usability Scale questionnaire [5] (except questions 4, 5 and 6 that are applied to more complex systems). Responses were on four point Likert scale and gathered as a global usability score ranging from 0 (poor) to 21 (high). The overall median score is 17/21, *Crosshair* has the lowest median score (14/21), followed by *Relative Pointing* (16.5/21), then *Shift&Freeze* (17.5/21) and finally *Direct Touch* (18/21) (figure 8-left). Score differences are not statistically significant (Kruskal-Wallis rank sum test of score by technique: X^2=6.651, p>0.05).

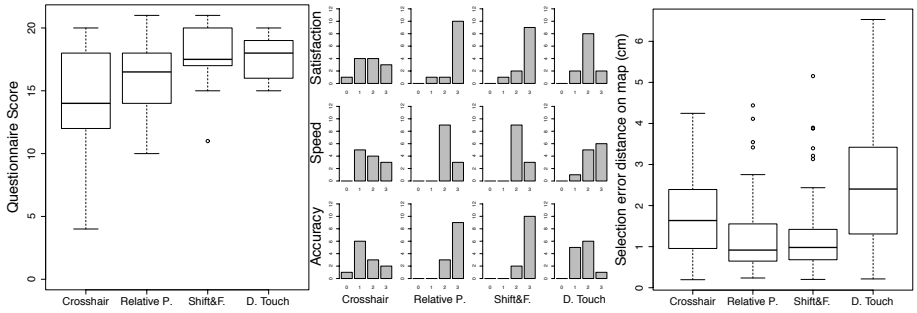

Fig. 8. For each technique: (left) Questionnaire scores; (center) Histograms of overall satisfaction, speed and accuracy rankings; (right) Boxplot of distances between the selection point and the target's center for Task 4 (2mm targets).

Participants also ranked each technique in terms of overall satisfaction, speed and accuracy on four point Likert scale (figure 8-center). Kruskal-Wallis rank sum tests found significant differences between techniques for overall satisfaction (X^2=14.3897, p<.01) and accuracy (X^2=24.2827, p<.0001) rankings by technique. A post-hoc pairwise comparison of overall satisfaction and accuracy rankings shows significant difference (with p<0.05 for overall satisfaction and p<0.01 for accuracy) for all pairwise comparisons except for the comparisons between *Shift&Freeze* and *Relative Pointing* and between *Crosshair* and *Direct Touch.* Table 1 gives satisfaction and accuracy ranking means.

---

[3] http://www.libpointing.org/

[4] http://www.R- project.org

Table 1. Means of satisfaction and accuracy ranking by techniques.

Rank Means	Crosshair	Relative Pointing	Shift&Freeze	Direct Touch
Satisfaction	1.75	2.75	2.67	2
Accuracy	1.5	2.75	2.83	1.67

Finally, during the experiment debriefing, we asked participants for the techniques they found to be the fastest and the more precise and for the techniques they preferred overall (multiple answers were allowed). Six participants said *Relative Pointing* seems to be the fastest one, four said *Direct Touch,* three said *Crosshair* and one said *Shift&Freeze*. All but one participant said *Shift&Freeze* seems to be the most precise one and five said it was *Relative Pointing*. Eight participants preferred *Relative Pointing* and six preferred *Shift&Freeze*. Two more participants would have also preferred *Shift&Freeze* given that it provided zoom and a cancel option.

Selection accuracy. We looked at the spread of selection points around the small targets of Task 4 (2mm wide). From 288 target selections, we removed 7 outliers noted during the experiment. The overall median of distances to the targets on the map is 1.7cm. The *Direct Touch* median (2.4cm) is more than twice that of *Relative Pointing* (0.9cm) and *Shift&Freeze* (1.0cm). The *Crosshair* median (1.6cm) lies in between (figure 8-right).

Discussion. These results support hypothesis H1. Indeed, *Shift&Freeze* and *Relative Pointing* are preferred over *Direct Touch* and *Crosshair*. Participants gave the best ranks in terms of accuracy and overall satisfaction to *Shift&Freeze* and *Relative Pointing*. Moreover participants have never mentioned either *Direct Touch* or *Crosshair* when asked for their preferred technique or for the most precise technique. However, these results do not support hypothesis H2. *Crosshair* received the lowest SUS scores, and *Shift&Freeze* was almost unanimously said to be the most precise technique. Moreover *Shift&Freeze* and *Relative Pointing* were almost equally preferred. So, indirect pointing techniques were not preferred over direct touch-based ones (i.e. *Direct Touch* and *Shift&Freeze)* even if the tablet form factor was presumably less convenient for direct touch-based techniques. Actually, some of our participants were used to direct touch input up to the point that they were tempted to tap on the cursor for the two indirect pointing techniques (*Crosshair* and *Relative Pointing).* The trend given by the measurable results is consistent with feedback gathered during the interviews.

Some participants complained about the handheld tablet form factor. A first reason is that due to its size and weight it is best held with both hands, but, as already explained, this impairs access to the screen with the *Direct Touch* and *Shift&Freeze* techniques. A second reason is that holding the tablet for AR application is different from other applications. The user needs to maintain the camera focus while interacting with the screen. Some participants felt that they held the tablet unsafely as they found it to be slippery and proposed to add some grips to the device. Also, the screen borders are not broad enough to allow all users to hold the tablet with their thumb on

the side of the screen. This results in accidental touch inputs and an uncomfortable tablet hold when trying to hold the tablet with one hand in order to interact with the other one.

As for the distance from the wall map, most of the participants walked about the same distance for all tasks and all techniques. This is a surprising result since we expected the participants to adapt the distance to the map according to the difficulty of the task. Only one participant clearly adapted his distance from the map according to both target size and ease of manipulation of the technique. He did so up to the point that he did not walk at all for large targets with *Relative Pointing* (as he felt more comfortable with this technique).

The spread of selection points around small targets suggests that *Relative Pointing* and *Shift&Freeze* have higher accuracy than the two baseline techniques. It also suggests that *Direct Touch* is the least precise technique and that *Crosshair* has an intermediate accuracy. In the next controlled experiment, we further study small target acquisition.

4.2 Experiment 2: Performance

For the second experiment, we made the following hypothesis:

- H1: *Relative Pointing* and *Shift&Freeze* are more accurate than *Direct Touch* and *Crosshair* but they take longer to operate for small targets.
- H2: *Crosshair* is more accurate than *Direct Touch.* While both techniques are impaired by hand jitter, *Crosshair* does not suffer from finger occlusion.
- H3: *Relative Pointing* and *Shift&Freeze* offers similar accuracy. Both techniques overcome limitations inherent to touch input and hand jitter.

Procedure, apparatus and participants. This experiment was carried out utilizing the cyclical multi-direction pointing task paradigm. We used thirteen targets arranged in a circle on a remote screen. As the handheld tablet application uses computer vision to track the device's pose, the targets were overlaid on a background image. One target at a time was highlighted in black on the remote screen: this target must be selected by pointing at it on the tablet through the live video. In order to ensure a good visibility of the highlighted item regardless of its width, it was surrounded by a 3 cm wide white square with a green cross. Targets always appear in the same order: starting from the top item, the next item is always opposite and slightly clockwise from the selected one. One block thus consists of thirteen target selections plus the selection of the first target. The subjects were instructed to hold the tablet in portrait mode, to select the highlighted target as quickly and accurately as possible and to rest between blocks.

We used a single movement amplitude of 30 cm and 3 target widths (0.5 cm, 1 cm and 2 cm). We wished to have a consistent distance between the remote screen and the handheld tablet across participants and blocks. To do so, before each block, participants had to place the handheld tablet 1 meter (+/- 5cm) away from the remote screen by following indications displayed on the tablet screen. Those indications were hid-

den as soon as subjects started the block to avoid disturbing them during the experiment.

Presentation ordering of the four techniques and the three widths were counterbalanced using Latin squares. Each condition was presented three times including one time for training. The experimental design is:

4 Techniques x 3 Widths x 2 Blocks x 13 Selections = 312 acquisitions per subject,

and *4 Techniques x 3 Widths x 13 Training Selections* =156 training acquisitions per subject.

For the handheld tablet, we used a similar apparatus as for the previous experiment. In addition, the targets were displayed on a 27“ Apple Thunderbold display with 2560x1440 pixel resolution (109 dpi). The screen was placed vertically so that its center was 1.5m high from the ground. An ad-hoc application was developed to control target widths and to highlight the target on the remote screen.

Twelve unpaid volunteers (4 female, 8 male; 1 left-handed), ranging in age from 22 to 41 years (mean 30 years), were recruited from our institution. All participants had previous experience with touch-based handheld device (nine on a daily basis) amongst whom ten had used a touch-based tablet before.

Results. From 3744 selections, we removed 33 obvious outliers. Distance from the screen when selections were performed is on average 1.02m (1st quartile: 0.99m, 3rd quartile: 1.05m, range: 0.90m to 1.18m). This indicates that our experimental set-up that constrains participants to placing the handheld tablet at 1m (+/- 5cm) from the screen before starting each block was sufficient to confine the distance between the handheld tablet and the remote screen to a small range.

Errors. A Pearson’s Chi-squared independence test between success of target acquisition and the 4 *Techniques* shows a significant dependence ($X^2 = 616.0356$, $p < .0001$). The overall error rate is 44.6%. This high error rate is explained by the choice of rather small target widths. The lowest error rate over the 3 target *Widths* is for *Relative Pointing* (20.1%), then *Shift&Freeze* (34.6%), *Crosshair* (49.4%) and the highest error rate is observed for *Direct Touch* (75.0%) (Figure 9-left).

We performed a 4 x 3 (*Technique* x *Width*) within subject analysis of variance on error rate by user. The *Technique* ($F_{3,143}=50.835$; $p<.0001$) and *Width* ($F_{2,143}=57.286$; $p<.0001$) main effects as well as the *Technique:Width* interaction ($F_{6,143}=3.397$; $p<.01$) were found significant. A post-hoc Tukey multiple means comparison found significant difference for all comparisons. Differences between *Relative Pointing* and *Shift&Freeze* and between *Shift&Freeze* and *Crosshair* were found significant with $p<.012$. All other differences were found significant with $p<.0001$.

We also performed a 4 x 3 (*Technique* x *Width)* within subject analysis of variance on median by user of distance between target center and selection point. Significant main effects were found for *Technique* ($F_{3,143}= 42.605$; $p < .0001$) and *Width* with $p<.015$ ($F_{2,143}= 4.389$). *Technique:Width* interaction was not significant. A post-hoc Tukey multiple means comparison found significant difference for all comparisons (with $p<.01$ for *Relative Pointing*-*Crosshair* comparison and $p<.0001$ for the others)

except between *Relative Pointing* and *Shift&Freeze* and between *Shift&Freeze* and *Crosshair.*

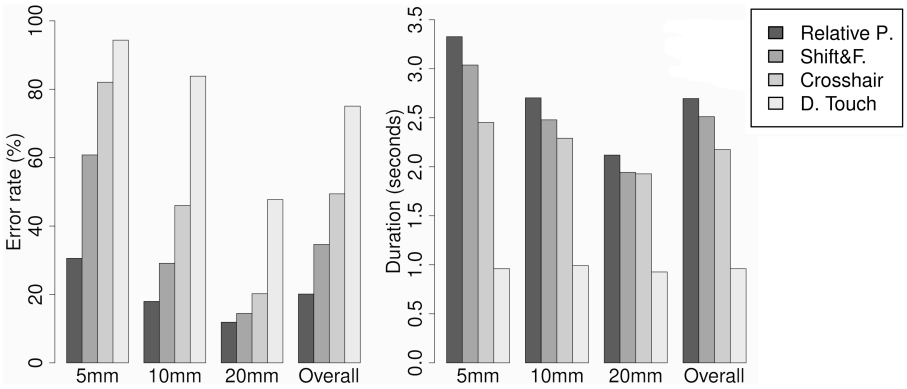

Fig. 9. (left) Error rates (%) and (right) selection durations (sec.) by *Technique* by *Width*.

Duration. The overall median of selection durations is 2.1 seconds, and medians of all selections for each technique are 2.7 seconds for *Relative Pointing,* 2.5 seconds for *Shift&Freeze,* 2.2 seconds for *Crosshair* and 1.0 second for *Direct Touch* (figure 8-right). We performed a 4 x 3 (*Technique* x *Width)* within subject analysis of variance on median of selection durations by user. Significant main effects were found for *Technique* ($F_{3,143}$= 67.781, p<.0001) and *Width* ($F_{2,143}$= 17.478, p<.0001). Effect of the *Technique:Width* interaction was also significant though with p=.0127 ($F_{6,143}$= 2.827). A post-hoc Tukey multiple means comparison found significant difference for all (with at least p<.01) except for two comparisons. Again, the differences between *Relative Pointing* and *Shift&Freeze* and between *Shift&Freeze* and *Crosshair* were not significant.

Discussion. The chosen tasks were quite hard to perform, which results in high error rates for all the techniques. Still we can observe that the different techniques offer different trade-offs between speed and accuracy or better said here, between duration and error rate (figure 9). The results partly support hypothesis H1. *Relative Pointing* is significantly more accurate and longer to operate than the two baseline techniques (i.e. *Direct Touch* and *Crosshair),* but this is not the case for *Shift&Freeze.* Indeed, *Shift&Freeze* does not show significant difference with *Crosshair. Crosshair* is significantly more accurate than *Direct Touch,* which supports hypothesis H2. Actually, *Direct Touch* is not adapted for those small target widths as indicated by both high error rates and no difference of duration across target widths. *Relative Pointing* and *Shift&Freeze* offers similar performance as indicated by non-significant difference of both duration and error distance. This supports hypothesis H3.

While participants held the tablet with both hands with *Relative Pointing* and *Crosshair,* they adopted different strategies with *Direct Touch* and *Shift&Freeze.* For

Direct Touch and *Shift&Freeze,* participants used two different strategies: (1) holding the tablet with one hand and interacting with the other hand's finger (9/12 for *Direct Touch* and 6/12 for *Shift&Freeze)* and (2) holding the tablet with both hands and interacting with their thumb (3/12 for *Touch* and 6/12 for *Shift&Freeze)*. This highlights the trade-off between holding the tablet and interacting on the screen for direct touch based techniques.

One drawback of this experiment is that the choice of the selection mode for *Shift&Freeze* and *Relative Pointing* was left to the participants. This results in different strategies as some users always used the precise mode while others adapted the mode according to the difficulty of the task. Nevertheless our goal was to evaluate our two techniques that include two modes.

5 Conclusion and Future Work

This paper provides a comprehensive study on precise pointing techniques for handheld Augmented Reality (AR). Our contributions are twofold. First we have presented an analytical framework for the design of interaction techniques for handheld AR that is based on the relationship between the touch input space and two visual output frames, namely the screen and the physical object. The usefulness of the framework is demonstrated by the classification of existing techniques and the design of two pointing techniques. Second we have presented two pointing techniques, *Shift&Freeze* and *Relative Pointing*. Their respective designs result from a twofold strategy: *Shift&Freeze* is conceived as an improvement of *Direct Touch* and solves its accuracy problem by freezing the video and using Shift's callout. *Relative Pointing* improves on the screen-centered *Crosshair* technique by stabilizing the cursor on the remote physical object. The two experiments revealed that those two techniques are preferred to the two commonly used techniques (*Direct Touch* and screen-centered *Crosshair*) and are more accurate than these baseline techniques. Further controlled studies must be performed to compare the two techniques, firstly, with less difficult tasks and then with a phone as the tablet form factor probably favors *Relative Pointing*. We also plan to run experiments with 3D physical objects (e.g., production machines). Several extensions to the two proposed techniques are envisioned including zooming for *Shift&Freeze* and testing different dynamic transfer functions (with or without known targets) for *Relating Pointing*. Finally, since our studies formulate the hypothesis of a perfect tracking of the device's pose, one further research avenue we must explore is the design of handheld AR pointing techniques that takes into account the imperfection of the underlying tracking system. We expect that precise pointing techniques in any context of use will become more and more crucial in the future, as a large range of richer and more complex handheld AR applications are designed.

Acknowledgements. This work has been supported by the French-Japanese ANR/JST AMIE project (http://amie.imag.fr). Special thanks to G. Serghiou for reviewing the article and to N. Mandran (LIG) for helping with the experiments.

References

1. M. Alessandro, A. Dünser, and D. Schmalstieg. Zooming interfaces for augmented reality browsers. In *Proc. MobileHCI '10*, ACM (2010), 161–170.
2. P-A. Albinsson, S. Zhai. High precision touch screen interaction. In *Proc. CHI '03*, ACM(2003),105-112.
3. P. Baudisch, A. Zotov, E. Cutrell and K. Hinckley. Starburst: a target expansion algorithm for non-uniform target distributions. In *Proc. AVI'08*, ACM(2008), 129-137.
4. S. Boring, D. Baur, A. Butz, S. Gustafson, and P. Baudisch. Touch projector: Mobile interaction through video. In *Proc. CHI '10*, ACM(2010), 2287–2296.
5. J. Brooke. SUS-a quick and dirty usability scale. In *Usability Evaluation in Industry,* chapter 21. London: Taylor and Francis (1996).
6. G. Casiez and N. Roussel. No more bricolage!: methods and tools to characterize, replicate and compare pointing transfer functions. In *Proc UIST '11*, ACM(2011), 603–614.
7. C. Forlines, R. Balakrishnan, P. Beardsley, J. van Baar, R. Raskar. Zoom-and-pick: facilitating visual zooming and precision pointing with interactive handheld projectors. In *Proc. UIST '05*, ACM(2005), 73-82.
8. N. Henze and S. Boll. Who's that girl? Handheld augmented reality for printed photo books. In *Proc. INTERACT'11*, Springer-Verlag(2011), 134–151.
9. G. A. Lee, U. Yang, Y. Kim, D. Jo, and K.-H. Kim. Snap-to-feature interface for annotation in mobile augmented reality. In *Workshop on Augmented Reality Super Models at ISMAR '10*, 2010.
10. G. A. Lee, U. Yang, Y. Kim, D. Jo, K.-H. Kim, J. H. Kim, and J. S. Choi. Freeze-set-go interaction method for handheld mobile augmented reality environments. In *Proc. VRST '09*, ACM(2009), 143–146.
11. C. Liao, Q. Liu, B. Liew, and L. Wilcox. PACER: fine-grained interactive paper via camera-touch hybrid gestures on a cell phone. In *Proc. CHI '10*, ACM(2010), 2441–2450.
12. R. L. Potter, L. J. Weldon, and B. Schneiderman. Improving the accuracy of touch screens: An experimental evaluation of three strategies. In *Proc. CHI '88*, ACM(1988), 27–32.
13. J. Rekimoto and K. Nagao. The world through the computer: Computer augmented interaction with real world environments. In *Proc. UIST '95*, ACM(1995), 29–36.
14. M. Rohs and A. Oulasvitra. Target acquisition with camera phones when used as magic lens. In *Proc. CHI '08*, ACM(2008), 1409–1418.
15. M. Rohs, A. Oulasvitra, and T. Suomalainen. Interaction with magic lens: Real-world validation of a Fitt's law model. In *Proc. CHI '11*, ACM(2011), 2725–2728.
16. M. Rohs, R. Schleicher, J. Schöning, G. Essl, A. Naumann, and A. Krüger. Impact of item density on the utility of visual context in magic lens interactions. *Journal Personal and Ubiquitous Computing,* 13(8):633–646, November 2009.
17. A. Roudaut, S. Huot, and E. Lecolinet. TapTap and Magstick: improving one- handed target acquisition on small touch-screens. In *Proc. AVI '08*, ACM(2008), 146–153.
18. T. Vincent, L. Nigay, and T. Kurata. Classifying handheld augmented reality: Three categories linked by spatial mappings. In *Workshop on Classifying the Augmented Reality Presentation Space at ISMAR '12,* 2012.
19. D. Vogel and P. Baudisch. Shift: A technique for operating pen-based interfaces using touch. In *Proc. CHI '07*, ACM(2007), 657–666.
20. J. Wagner, S. Huot, and W. E. Mackay. BiTouch and BiPad: Designing bimanual interaction for hand-held tablets. In *Proc. CHI '12*, ACM(2012), 2317-2326.