TapTap and MagStick: Improving One-Handed Target Acquisition on Small Touch-screens

Anne Roudaut[1]   Stéphane Huot[2]   Eric Lecolinet[1]

Anne.Roudaut@enst.fr, Stephane.Huot@lri.fr, Eric.Lecolinet@enst.fr

[1]TELECOM ParisTech, CNRS LTCI
46 rue Barrault
75013, Paris, France

[2]LRI – Univ. Paris-Sud & CNRS, INRIA
F-91405 Orsay
France

ABSTRACT

We present the design and evaluation of TapTap and MagStick, two thumb interaction techniques for target acquisition on mobile devices with small touch-screens. These two techniques address all the issues raised by the selection of targets with the thumb on small tactile screens: screen accessibility, visual occlusion and accuracy. A controlled experiment shows that TapTap and MagStick allow the selection of targets in all areas of the screen in a fast and accurate way. They were found to be faster than four previous techniques except *Direct Touch* which, although faster, is too error prone. They also provided the best error rate of all tested techniques. Finally the paper also provides a comprehensive study of various techniques for thumb based touch-screen target selection.

Categories and Subject Descriptors

H.5.2. User Interfaces: Input Devices and Strategies, Interaction Styles, Screen Design; D.2.2 User Interfaces

General Terms

Design, Human Factors

Keywords

Mobile devices, one-handed interaction, thumb interaction, touch-screens, interaction techniques.

1. INTRODUCTION

Many mobile devices are now fitted with touch-screens that enable us to interact directly with our fingers. However, most graphical interfaces still require users to click on small widgets by using a stylus. As highlighted in [7, 10], this interaction style is not the best way to interact with small devices in a mobile context: it requires too much attention (especially if the user is moving) and forces users to use both hands (one hand holding the device while the other manipulates the stylus). Ideally, mobile interaction should just require one hand, with the thumb being used for selecting objects. In fact, direct selection on the screen is intuitive and fast, and using only one hand is central as users may perform several simultaneous tasks.

However, direct thumb interaction on small touch-screens raises several issues: a) hand and thumb morphology makes it difficult to reach the corners of the screen; b) the thumb may occlude large parts of the screen that can contain the desired target; c) the relatively large contact zone between the fingertip and the tactile screen makes selection ambiguous, especially in applications that require users to click on tiny widgets for triggering actions. Despite these issues, this *Direct Touch* technique is still the most widely used.

Figure 1. TapTap and MagStick

Alternate techniques have been proposed to improve accuracy and eliminate visual occlusion, but they make the interaction slower and more complex. In this paper, we first present a thorough analysis of the properties of the techniques published so far. We then introduce two novel interaction techniques, **TapTap** and **MagStick** (Fig. 1) that solve the problems raised by our analysis (screen accessibility, visual occlusion and accuracy). Finally, we performed a controlled experiment which proved that our two techniques outperform the ones proposed previously (*Direct Touch*, *Offset Cursor* [11], *Shift* [14] and *Thumbspace* [6]).

2. RELATED WORK

Research on thumb interaction with mobile devices is a relatively recent field. The state of the art thus still largely relies upon research on interaction with regular touch-screens. In spite of recent innovations, the issues of reaching far targets, visual occlusion and accuracy are not yet completely solved.

Target accessibility. The borders of the screen are more difficult to reach [6], especially with the thumb because the morphology of the hand constrains thumb movements. This will degrade interaction in the screen areas that are farthest from the natural thumb extent (*i.e.* the top and left border for a right-handed user). Besides, thumb movements may also be hampered near borders because of the thickness of the device's edges around the screen.

Visual occlusion. When interacting, the finger hides a part of the screen and can even totally occlude small targets. This problem is more pronounced when interacting with one hand because the thumb pivots around the thumb joint and can hide half the screen.

Accuracy. A study [9] showed that 9.2 mm is the minimum size for targets to be easily accessible with the thumb. Some mobile devices, such as the iPhone or the HTC Touch, rely on a limited set of large buttons. But, this approach reduces the number of targets because of the lack of screen estate and is thus inappropriate for many applications. Besides, the exact location of the pointer tends to be imprecise because of the large contact surface between the thumb and the screen. As current touch-screen hardware technology computes the barycenter of the multiple contact points, small variations in the way of pressing the thumb can provoke jerky movements of the pointer.

In the following, we group the existing attempts to solve those issues in three categories depending on how they handle input: "tapping","dragging"and "hybrid" techniques.

2.1 Tapping Techniques

Tapping techniques capture the position of the pointer when the thumb touches the screen. The most widely used technique on regular or small touch-screens, Direct Touch, relies on this intuitive principle. The user must tap the screen precisely at the location where the target is displayed. This technique is fast but it is also very error prone for selecting small targets because, as mentioned previously, the location of the contact point is hard to anticipate. Finally, Direct Touch does not tackle the problem of targets located at the borders of the screen.

2.2 Dragging Techniques

Dragging techniques come from the *take-off* paradigm [11] which consists in pressing the screen, dragging a cursor, and lifting the finger to validate the selection. The former technique, Offset Cursor [11,13], was designed to avoid finger occlusion on large touch-screens and to solve the accuracy problem of Direct Touch. A cursor is always displayed at a fixed distance above the contact point to help the user reaching the topmost locations of the screen. Offset Cursor was shown to induce far fewer errors [11, 13] than Direct Touch, but it is also significantly slower. In [14], Vogel et al. noticed that users often overshoot or undershoot targets. They assumed that it is difficult for the users to estimate the offset distance and that a lengthy adjustment of the cursor, called *net correction distance*, is thus necessary to acquire targets.

Another point is that Offset Cursor does not cover the entire extent of the screen. As the cursor is always located at the same distance from the top of the finger, targets at the bottom of the screen remain unreachable. Moreover, the thickness of screen edges makes it difficult to select targets located near the corners. An adaptative horizontal offset has been proposed in [5] to improve Offset Cursor: this offset is null at the center of the screen and grows smoothly towards the left and right borders. This technique makes it easier to reach items that are close to the left and right borders, but requires slightly more training.

Thumbspace [6] has been designed to improve access to the borders and corners of the screen. It uses an on-demand "radar view" that the user can trigger at the center of the screen. Interacting directly on this radar view allows the user to reach all locations on the screen. Thumbspace thus works as an absolute positioning touchpad superimposed on the standard touch-screen. A drawback of this approach is that the thumb is above the cursor in some areas of the screen, thus causing an occlusion. To get around this issue, the authors proposed to use Thumbspace for targets that are difficult to reach and Direct Touch for near targets.

Thumbspace also relies on *Object Pointing* [3]. The original feature of this interaction technique is that the cursor never visits empty regions and jumps from one target to another, according to the direction of the pointer. Thumbspace uses this strategy with a triggering threshold of 10 pixels to avoid jerky cursor movements. The screen is subdivided into "proxy" areas which are associated to a unique target. This way of "tiling" makes unused background areas active and thus provides more motor space for selecting each target. However, this approach may lose in efficiency when many targets are present on the screen or if they are close to each other.

2.3 Hybrid Techniques

Shift [13] attempts to decrease the selection time of Offset Cursor by a hybrid approach: a coarse Direct Touch on the target can be followed by a precise cursor adjustment if needed. Touching the screen triggers a callout that shows a copy of the occluded area in a non-occluded area. The actual selection point (under the finger) is represented by a cursor in the callout, and the user adjusts its position to fine tune selection before releasing his finger. This technique reduces the *net correction distance* and selection time as the user touches the screen directly on the target. Besides, the callout only appears when needed, after a delay that depends on the target size (the larger the target, the longer the delay). This strategy should improve selection time as the callout is only used for fine-tuning small target selections. However, Shift does not completely solve the screen coverage problem as it requires users to put their fingers close to the target location. Finally, the experiment that was presented in [13] was performed by using both hands to manipulate the device.

2.4 Summary

Direct Touch is the fastest technique proposed so far. However, it remains unusable in most real-life applications because of its high error rate. Some alternatives, inspired by the *take-off* paradigm [11], have been proposed. However, even if they solve the accuracy problem of Direct Touch, the other issues of thumb interaction remain unaddressed. Offset Cursor avoids occlusion and increases accuracy but it limits access to targets at the bottom of the screen and it is not very well suited for reaching targets in the right and left corners (this problem can however be solved by using an adaptive horizontal offset). Thumbspace was specifically designed to address this accessibility problem in the corners, but it does not prevent occlusions in the center of the screen. Finally, Shift which was evaluated by using both hands, does not fully address the corner accessibility issue of the thumb as users must tap close to the desired targets. To sum up, as illustrated in Table 1, efficient solutions have been proposed to solve the problems involved with thumb interaction individually, but none of the existing techniques address them all together. This is the challenge we met by designing TapTap and MagStick, two new interaction techniques that we introduce in the next sections.
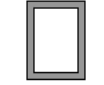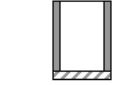
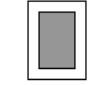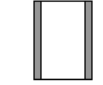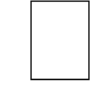|	Direct Touch	Offset Cursor	Adaptive Offset	Thumbspace	Shift	TapTap	MagStick
Overview							
Target Accessibility							
	Grayed areas are difficult to reach – Hatched areas are impossible to reach						
Thumb Occlusion	Everywhere	None	None	Center (if same relative and absolute positions)	On top left	None	None
Pointing Accuracy	Coarse	Medium (net correction distance time)	Medium (net correction distance time)	Fine (facilitated by Object Pointing)	Medium (small targets) and coarse (large targets)	One coarse and one fine (increase target size)	Fine (facilitated by Semantic Pointing)

Table1. Comparison of the features of one-handed interaction techniques

3. TAPTAP AND MAGSTICK

TapTap and MagStick are specifically designed for interacting with the thumb on small touch-screens. Both techniques address the issues of thumb interaction that we previously pointed up. Their respective designs result from a twofold strategy: TapTap was conceived as an improvement of Direct Touch and solves its accuracy and accessibility problems and MagStick is an improvement of Offset Cursor and other techniques based on the take-off principle. A video demonstration can be viewed at http://www.anneroudaut.fr

3.1 TapTap

TapTap comes from a simple idea: if a single tap is not efficient for selecting a small target accurately, a second tap should suffice to disambiguate the selection. More precisely, the first tap defines an area of interest on the screen (Fig. 2a); this area is then magnified and displayed as a popup on the center of the screen (Fig. 2b); the second tap selects the desired target in the popup (Fig. 2c) (or cancels the selection if an empty space is selected).

Selection is by design more precise because the selecting tap takes place on a magnified view of the area of interest where the targets are large enough to be easily selected with the thumb. TapTap also improves accessibility in screen border areas. Not only does the first tap not need to be performed on the desired target (it must only be performed reasonably close to this target), but also the magnified view pops up in the center of the screen. Targets that are close to the borders in the original view thus appear in a location that is much easier to reach in the magnified view.
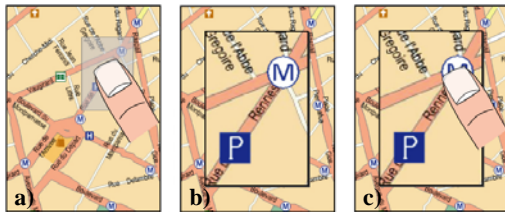

Figure 2. TapTap Design

TapTap is thus based on a temporal multiplexing strategy where the first tap serves to specify the focus area in the original view so that this focus will be displayed at a scale that makes it possible to select the target precisely. Although based on zooming, this strategy has some interesting characteristics that make it different from usual multi-scale approaches. First, there is no interactive control of the zooming factor nor of the amount of XY panning as they are automatically adjusted. Interaction is very fast and works

practically like a quasi-mode: the first tap enters the selection mode and makes the zoomed view appear, while the second tap closes this view and leaves the selection mode.

The zooming factor was chosen in order to take into account the size constraints of small touch-screens on mobile devices. Besides, in an attempt to satisfy contradictory constraints, the view and the targets are not zoomed in with the same factor.

On the one hand the focus zone that is selected by the initial tap must be relatively large so that it contains the desired target even if the tap location is (reasonably) far from the target. A size of 80 x 120 pixels was empirically chosen (for a QVGA screen of 240 x 320 pixels). This makes it possible to tap as far as 40 pixels horizontally and 60 pixels vertically from the desired target, a distance that is sufficient to prevent almost all errors in the first tap.

On the other hand, the relatively large size of the focus zone constrains the zooming factor that can be applied in the magnified view because of the small size of the QVGA screen. Moreover, the whole screen real estate can not be used because of the accessibility problem (the areas close to the borders are difficult to reach with the thumb). As a consequence, the focus area is only magnified by a factor of 2 in the pop up (its size is thus 160 x 240 pixels) and placed in the center of the screen (Fig. 2). It is hence located in the most favorable area of the screen for interacting [6].

However, this zooming factor may be insufficient for making common targets large enough to be selected precisely. According to [9] targets should be at least 9,2mm large for making thumb selection easy. But many mobile applications have targets as small as 3 mm [14,12]. In order to ensure sufficient size, targets are zoomed in by a factor of 3 instead of a factor of 2 for the rest of the focus view. Ours observations showed this choice to be effective: users had no difficulties in selecting 9mm targets (i.e. 3mm targets magnified 3 times) and they were not disoriented by this dual zooming factor (in fact none of them noticed this feature).

3.2 MagStick

Dragging techniques are more accurate than Direct Touch but they are significantly slower and do not solve all screen accessibility issue. MagStick solves these problems by providing a telescopic stick that controls a "magnetized" cursor. The telescopic stick can reach any target on the screen while the magnetization of the cursor (which can be seen as form of semantic pointing [8]) speeds up the adjustment of the cursor to the target location. Finally, the offset distance of the cursor is not constant, but dynamically adjusted by the user in a highly predictable way.

MagStick works as follows: 1) when the user presses the screen, he defines a reference point (Fig. 3a); 2) by dragging his thumb he makes a two-part stick appear (Fig. 3b): the two parts emanate from the reference point and end at the current position of the thumb and the location of the cursor; 3) as both parts always have the same length and (initially) the same direction, the user can control the location of the cursor by dragging his thumb continuously on the screen (changing the size of one part of he stick automatically changes the size of its other part); 4) targets *attract* the cursor as if it was "magnetized", with the effect of bending the stick as shown in Fig. 3c); 5) finally the user releases the thumb to select the target that is currently below the cursor (or to cancel if an empty space is selected)
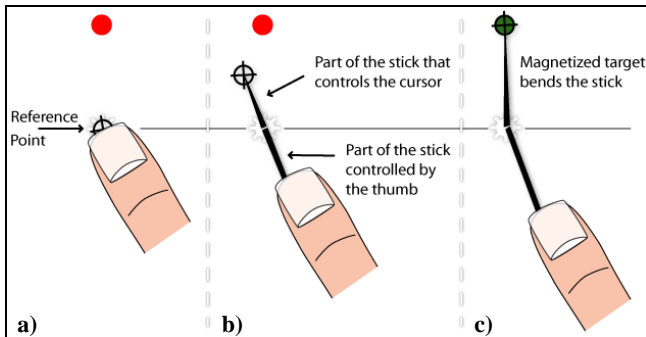

Figure 3. MagStick Design

A key feature of this technique, which was inspired by games such as electronic billiards, is that the cursor moves in the opposite direction of the fingertip. This strategy is especially efficient for avoiding visual occlusions as the thumb must be moved *away* from the desired target: not only the thumb will not hide the target but a large part of its visual context will be made visible.

Another important feature of MagStick is that its symmetrical design allows the user to easily predict the movement to perform. An important drawback of Offset Cursor is that most users, with the exception of very well-trained ones, can not know the exact location of the cursor until they touch the screen. They must thus wait for the cursor to appear before starting to adjust its position finely. Conversely, as the two parts of the stick are of equal length, this problem does not exist with MagStick. The user can predict how far he will have to move his finger *before* touching the screen as this distance is equal to the distance between the target and the reference point.

Magnetization, which derives from Semantic Pointing [2], also contributes to speed up the selection task. Each target has a proximity area that attracts the cursor and "bends" the stick. When the cursor enters a proximity area, it is attracted to the center of the corresponding target. This feature makes fine positioning unnecessary but also avoids "empty selection" errors that would otherwise occur when the user overshoots or undershoots the desired target. Conversely, when the user moves the stick (and the cursor) away from a target area, the magnification effect vanishes and the two parts of the stick become aligned again until the cursor is attracted by another target. A possible refinement would be to assign different attraction powers to targets, as proposed in the original Semantic Pointing technique. It could facilitate the selection of targets that are very frequently used, or, conversely, to prevent the accidental activation of dangerous commands. However, this feature should be carefully tested in the context of thumb interaction where cursor movements are necessarily more imprecise than when using a mouse on a desktop.

4. PROPERTIES OF THE TECHNIQUES

This section compares the properties and the respective advantages of our techniques. In particular, it shows that they provide efficient solutions to the three problems presented in the 'related work' section: target accessibility, visual occlusion and accuracy. We also investigate the compatibility of our techniques with dragging gestures and other target sizes and layouts.

4.1 Target accessibility

TapTap and MagStick can select targets anywhere on the screen although they use different principles. TapTap uses a two-step zooming strategy where the user specifies a focus of interest that is then displayed at a larger scale in the center of the screen. The first tap does not need to be very close to the target and the second tap is always performed in the most favorable area of the screen.

Conversely, MagStick relies on a space-shifting strategy by providing a "telescopic arm" that reaches targets close to the borders. As with TapTap, MagStick makes it possible to perform the dragging gesture in the most favorable area of the screen, but it leaves freedom to the user to interact by following two different strategies. The first one consists in touching the screen very close to the target in order to minimize the length of the dragging gesture. Another strategy is to systematically start the dragging gesture from the center of the screen. Any target can then be selected, either by placing the thumb below the target if it is in the upper part of the screen, or above the target if it is in its lower part. This strategy was in fact used by most of our participants during the evaluations. Another of its advantages is that it allows the user to hold the mobile device firmly with the hand that performs the interaction. The thumb joint is then located in the middle of the right border of the screen (for a right hand user) and the center of gravity of the handheld device is roughly above the center of the hand. This position is safe and convenient because it prevents the risk of dropping the device accidentally. The user then moves his thumb upward or downwards when the target is exactly located beneath the natural position of the thumb joint, but this case seldom occurs and does not require cumbersome hand movements.

4.2 Visual occlusion

The zooming strategy of TapTap prevents visual occlusion by design: as targets are magnified by a x3 ratio, they are large enough not to be completely hidden by the thumb.

The design of MagStick also ensures that visual occlusion can not occur as the thumb moves away from the desired target. Both the target and the focus of attention are clearly visible. It also prevents occlusion in the thumb joint area as shown in Fig. 4 for the same reason as explained in the previous section: the thumb is naturally located in the middle of the screen and can easily be slightly shifted up or down when needed.
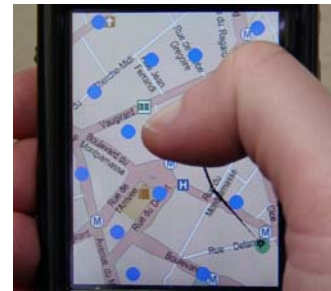

Figure 4. No occlusion on the thumb joint with MagStick

4.3 Accuracy without reducing speed

Both techniques attempt to "circumvent" the constraints of the Fitts' Law for a homogeneous 2D space in different manners. TapTap relies on a multi-scale space (that can be seen as a generalization of magnification tools as those proposed in [8,15]). As shown by Guiard et al. [4] multi-scale spaces significantly increase the range of indexes of difficulty that users can handle and Fitts' Law applies uniformly over this range. TapTap makes it possible to decrease the index of difficulty through zooming (that is to say a translation on the scale axis of the space-scale diagram). The two taps required by TapTap are thus performed faster than two "standard" successive taps (this assumption was confirmed by experimental data): The size of the "target" is increased, and the distance between the thumb and the target decreased in both steps of the interaction (the "target" being a zone of interest in the first case, and an actual but magnified and centered target in the second case). This property also increases accuracy and allows the user to view TapTap as a double click with a fast spatial readjustment between the two taps. As detailed in the experiment section, this effect was striking when conducting the evaluation: users did not give the impression that they were performing two successive taps but rather a compound gesture.

Similarly, MagStick relies on Semantic Pointing, a technique that distorts the motor space and thus artificially reduces the pointing distance. This technique also avoids the cursor leaving the target when the thumb is slightly, and involuntarily, moved. As stated above, the input signal provided by current touch-screen technology is somewhat imprecise and instable when interacting with the thumb. Although filtered by a low pass filter to remove outliers and smooth the input curve [13,14] this signal is still far from being perfectly reliable. Besides, the user may also involuntarily move his thumb when he releases it and thus miss the target. Magnetization solves both problems.

Finally, the ability to predict the movement before starting the gesture is probably another key feature for making the selection faster. The property relies on the fact that both parts of the stick always have the same length. Using a variable gain, as in [1], sounds appealing but could decrease performance in our case because this important property would be lost. This was confirmed by preliminary experiments we made when designing MagStick.

4.4 Other properties

Real mobile computer operations are combination of different interaction techniques, such as pointing or dragging. In our experiments, we focus on pointing with small and randomly laid out targets. In this section, we present some other interesting properties of TapTap and MagStick. More precisely, we investigate how our techniques work with different kinds of targets (size and layout) and their compatibility with other interaction styles.

Large targets. Although targets can hardly be much smaller, and still easily visible, than those we considered (3mm, a size found in many mobile applications [14,12]), they can however be much larger. MagStick then operates as Direct Touch: as the cursor appears below the thumb when it is pressed on the screen, the user can just release it without performing any movement to select the target. TapTap can be replaced by Direct Touch for large targets. This can be made explicit by a visual cue. But choosing target sizes in a consistent way may suffice (for instance targets with only 2 or 3 different heights). Selecting targets in two different ways may not be a real problem after some training: a) people do

that all the time when using desktops (documents must be double-clicked, while other buttons are, generally, simple-clicked); b) a small inactivation delay could be used in such a way that a second click on a large button (or the view it generates) would have no effect. Hence, a useless second tap would never produce an unexpected result.

Lists and Groups. Aligned or grouped targets are often common in real applications: this case typically occurs in menus, lists, tool boxes, tabbed panes, etc. While TapTap performance is likely to be similarly high whatever the layout, the specific design of MagStick can provide interesting features in this case. It makes it for instance possible to access items organized as lists or trees by moving the thumb away and keeping it approximately at the same location of the screen. This could be very useful for browsing a menu system without having to perform multiple target selections. Besides, as the thumb can be placed rather far away from the target, this would noticeably reduce occlusion and would thus make it possible to display more contextual information.

Dragging gestures. TapTap does not interfere with interaction styles based on dragging gestures as it only requires users to tap the screen. A target can be moved by dragging the thumb on the screen instead of releasing it immediately after the second tap (the popup does not cause visual occlusion because it disappears when the user starts the second tap by pressing the screen). This way of dragging objects is in fact quite similar to the usual one except that the target is not beneath the cursor but remotely controlled by the movements of the thumb. The target moves in the global view according to the movement of the thumb from the position of the second tap. In order to move the target anywhere on the screen, this movement is multiplied by a constant gain of 2. In addition, TapTap also makes it possible to pan the entire view by dragging on its "background". An image, a map or a page could for instance be panned in this way.

MagStick also has interesting properties regarding this criterion. First, it allows an object to be dragged, although in a slightly less usual way than with TapTap. Instead of releasing the thumb immediately when the proper target is reached, the user must wait for a small temporal delay. The target is then implicitly selected and can be moved by dragging the finger.

To sum up, we have seen in this section that TapTap and MagStick address all the issues raise by one-handed interaction, and that they can be applied in different kinds of application without preventing the use of other interaction styles. The next section shows the effectiveness of TapTap and MagStick through a controlled experiment that compares them with the main techniques proposed so far.

5. EXPERIMENTAL EVALUATION

We conducted a controlled experiment to compare TapTap and MagStick with the main techniques published before: Direct Touch, Offset Cursor [11], Thumbspace [6] and Shift [14]. Since the previous techniques principally explored the pointing task, our experimentation focuses on this problem of pointing only. According to the design of our techniques and the properties that were previously described, our hypotheses are that:

H1: *TapTap and MagStick are the fastest techniques after Direct Touch.*

H2: *TapTap and MagStick are the techniques with the lowest error rate*

H3: *TapTap and MagStick are efficient for accessing targets anywhere on the screen.*

5.1 Task

The task consisted in performing series of target selections with the six techniques. Participants were asked to hold the device with their dominant hand and to use their thumb. Several targets were displayed on the screen and one of them was to be selected. The participants were instructed to perform the selection as quickly and accurately as possible. Before each trial, the user presses a "Next trial" button and a city map appears with a set of 16 targets. They are displayed in blue color, except for the one to be selected that is in red. The blue targets are distractors in order to improve the realism of the target acquisition task. The color of the target changes to green when the cursor flies over it (except for tapping techniques such as Direct Touch and TapTap). The trial ends when the user lifts his thumb from the screen, whether he succeeds or not the selection. A sound indicates the result of the acquisition.

5.2 Apparatus and participants

The techniques have been implemented in C# (with the .Net Compact Framework) and operate on the Windows Mobile 5.0 OS. Experiments have been performed on a HTC P3600 PDA-phone with a QVGA (320x240) touch-screen. Twelve volunteers (1 female), ranging in age from 23 to 47 years, were recruited from our institution and received a handful of candies for their participation. All of them were using a mobile device with a touch-screen for the first time. Two subjects were left-handed and we mirrored their results so that each user used their dominant hand to perform the experimentation.

5.3 Experimental conditions

The efficiency of the interaction techniques involved in this experimentation is likely to depend on the location of the target. Karlson et al. took this aspect into account in their experiment [6]. They subdivided the screen into 12 areas arranged as a regular matrix. We used a different subdivision pattern, with 8 zones of the same surface area (Fig. 5a represents the 12 areas for a right-handed person), which provides a clear separation between the areas located at the center of the screen and those close to the borders, which may degrade performance. This analysis of the screen areas is important because it can have strong implications on the design of interactive applications.

To reduce the task time for our participants, we only considered one target size of 3 mm, because this value was reported to be the actual minimal widget size in mobile applications [14,12]. Besides, Vogel also reported in [14] that Direct Touch and Shift outperforms other techniques for targets larger than 18 pixels. The study thus focuses on small targets, as they constitute a more difficult case and are commonly found in mobile applications. The proximity areas for the MagStick magnetize effect measure 10.8mm.

A minor enhancement was made on Offset Cursor because its original design makes it impossible to reach targets on the bottom of the screen. So that this technique is not at disadvantage, the user can make the cursor appear below the thumb position (negative offset mode) by pressing a hardware button before touching the screen. The analysis of the experimental data confirmed that this improvement did not affect the results (the performance is not significantly different in the 'down' area than in "easy to reach" areas such as 'up' and 'Center'). Hence all targets can be selected by using any of the 6 tested techniques.

During the task, Time, errors and thumb movements were recorded. At the end, the subjects answered a questionnaire to give their opinion and satisfaction about all techniques (6 variables were measured on a 5 pt. Likert scale).
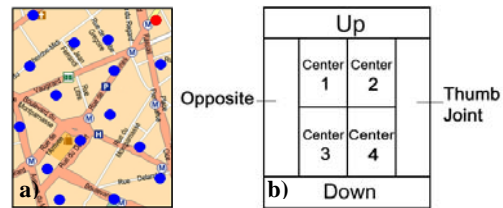

Figure 5. a) Targets layout b) Target Area subdivision

5.4 Experimental design

A repeated measures within-subject design was conducted. The independent variables are *Techniques* (Direct Touch, Offset Cursor, Thumbspace, Shift, TapTap and MagStick) and *Target Area* (8 areas shown in Fig. 4a). The presentation of *Technique* was circularly counterbalanced among participants. All of them performed 16 selections twice in all the 8 *Target Areas*. *Target Areas* were ordered in a sequence circularly counterbalanced for each technique. This sequence aims at balancing the regions that are easy or hard to reach. Finally, at the beginning of each technique, subjects performed 10 practicing trials. In summary, the design was: 6 *Techniques* x 8 *Target Areas* x 2 *blocks* = 96 selections (15-20 minutes) per participant.

5.5 Results

Repeated measures analysis of variance showed that the order of presentation of the techniques had no significant effect on selection time or error rate.

5.5.1 Selection time

Task time was measured from the moment the user released the "Next trial" button to the moment his thumb was lifted up from the screen. Trials with selection errors were excluded from the selection time analysis. We performed a 6 x 8 (*Technique*, *Target Area*) within subject analysis of variance. We found significant main effects for *Technique* ($F_{5,55}$=14.59, p<.001) and *Technique* x *Target Area* interaction ($F_{35,268}$=2.31, p<.001). Post hoc multiple means comparison tests allowed us to rank the techniques as follows: Direct Touch (1177.8 ms) and TapTap (1547.4 ms) (no significant results between them), MagStick (2037.6 ms), Shift (3046 ms) and Offset Cursor (3562.7 ms) (no significant results between them), and the slowest, Thumbspace (3897.3 ms). The results show that: TapTap is about to 2.3 times faster than Offset Cursor, 2 times faster than Shift, and almost 2.5 times faster than Thumbspace; MagStick is about 1.7 times faster than Offset Cursor, 1.5 faster than Shift and 1.9 faster than Thumbspace. These results, illustrated in Fig. 6, are all significant. We found that Direct Touch was the fastest, but (as described in the error result) the quantity of data collected is small compared to the other techniques. We can considerate Direct Touch "out of range" because a technique that produces so many errors is of course very frustrating for users, and can not be compared in this experiment with the other techniques that all provide better results.
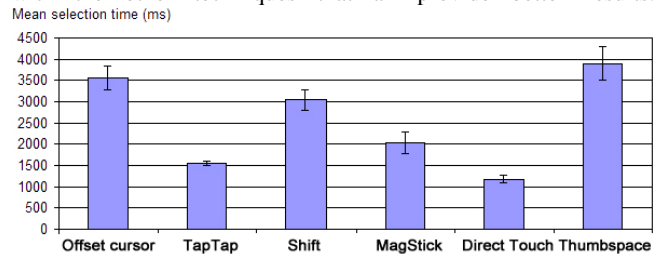

Figure 6. Mean time (ms) for Technique
Bars represent 95% confidence interval.

The analysis of the *Technique* x *Target Area* interaction showed that *Target Area* has no significant effect on selection with TapTap and MagStick. There is a significant effect for Offset Cursor, which is less efficient in the 'joint' area (see Fig.4) (2246.2 ms mean difference) and in the 'opposite' area (1250 ms mean difference) than in other zones. A similar effect was found for Shift in the 'up' (2445.6 ms mean difference) and 'opposite' (936.8 ms mean difference) areas. These results confirm our observations during the experimentation sessions where we noticed that users often hide the target with their thumb in these two areas. Some other significant effects were also found with Thumbspace, which performed better on the borders of the screen than in the center area (1381.9 ms of difference on average). This result corroborates the assumptions of the authors [6].

In summary, without considering Direct Touch, TapTap is the fastest and MagStick the second. The border areas are reached faster with MagStick than with the hybrid and the dragging techniques. Not only is MagStick quite efficient for reaching the edges, but it also does not impair interaction in the center of the screen (as Thumbspace does). TapTap is particularly fast and consistent across screen areas.

5.5.2 Error rate

The error rate measurement aggregates both empty and wrong target selections. We performed a 6 x 8 (*Technique*, *Target Area*) within subject analysis of variance on the aggregated number of errors. Error rate was significantly affected by *Technique* ($F_{5,55}$=45.91, p <.001) and *Technique* x *Target Area* interaction ($F_{35,268}$ = 1.74, p<.001). Post hoc multiple means comparison tests showed that TapTap (6.7%) has the lowest error rate and Direct Touch (59.9%) the highest in comparison to all other techniques (Fig. 6). No significant results were found in comparing the other techniques (i.e. Offset Cursor (16.1%), Shift (17.1), MagStick (10.4%) and Thumbspace (18.7%)). We can notice that the error rate of Direct Touch is considerably high. The error rate of TapTap is about 2.5 (and 1.6 for MagStick) times smaller than for Offset Cursor, Shift and Thumbspace.
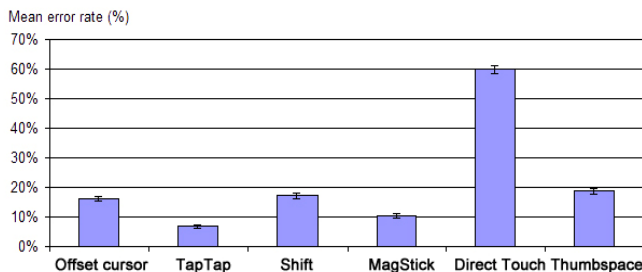

Figure 7. Mean Error rate for *Technique*
Bars represent 95% confidence interval.

The only significant result about *Technique* x *Target Area* interaction is mainly due to Direct Touch. Considering its high error rate and dissatisfaction of our participants with it, we will not discuss on these results. By considering empty and wrong selections separately (they were previously merged), we found that Thumbspace only produces wrong selections while the other techniques induce mostly empty selections. In fact these results are not surprising because by design Thumbspace "tiles" the space. This approach, which could be efficient because the target is then larger in the motor space, have also the disadvantage of causing more wrong errors that are much more costly than empty selections (canceling an action triggered by a wrong selection may be time-consuming and frustrating).

In summary, TapTap has the lowest error rate and Direct Touch the highest. All the "dragging" techniques and Shift have approximately the same error rates, except that Thumbspace errors are only wrong selections.

5.5.3 Subjective preferences

With the post-study questionnaire, participants ranked the six techniques as follows: TapTap, MagStick, Shift, Offset Cursor, Thumbspace and the most disliked Direct Touch. Their opinions about the *speed*, *accuracy*, *pleasantness*, *simplicity*, *learning* and *fun* are illustrated in Fig. 8. TapTap is the most liked technique for all criterions, except for 'fun' where it is placed second. Tapping approaches (TapTap and Direct Touch) are ranked first for the 'speed' assessment and users estimated that TapTap performs faster than Direct Touch, even if quantitative results showed the contrary. Direct Touch is disliked for the 'accuracy', 'pleasantness' and 'fun' criteria. Results for dragging approaches have a similar shape, with MagStick and Offset Cursor generally above Shift and Thumbspace. MagStick is judged slightly inferior for 'learnability' but ranked first for 'fun'.
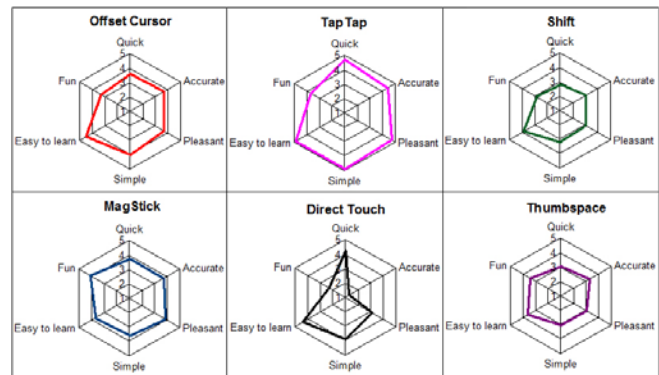

Figure 8. Questionnaire results (means).

5.6 Discussion

The results of our experimentation confirm our hypotheses. TapTap has the lowest error rate (H2) of all techniques and it is the fastest technique after Direct Touch (H1). In fact, it would be even faster than Direct Touch in the case of real usage. As Direct Touch is very error prone, many selections will have to be performed again. The average time needed to select a target is thus significantly higher than the time to correct selections given in the previous section. This average time can be estimated by considering that the selection task will take at least twice as much time in the case of wrong selections as the target must then be selected again (in fact it will take more time because a wrong selection may launch an undesired application that the user will have to close). According to this hypothesis, Direct Touch would require an estimated average time of 2002 ms while TapTap would only need 1676 ms as it produces much fewer errors.

Another interesting point is that the single tap of Direct Touch takes more time (1177.8 ms) than each tap of TapTap (803.3 ms for the first tap and 744.1 ms for the second tap). These results confirm the validity of the design hypotheses presented in section 3. Besides, users seem to perform the second tap slightly faster, an effect that may come from the fact that the magnified area is centred and the target thus pretty close to the natural position of the thumb.

Our results also validate the hypotheses that MagStick is faster (H1) than other techniques (TapTap and Direct Touch except, but Direct Touch is too error prone to be really usable, as stated before) and that it produces fewer errors (H2) than other

techniques (TapTap except again). Another interesting observation is that the time to press the screen is slightly faster for MagStick (844.2 ms) than for other dragging techniques (1140,6 ms for Offset Cursor, 958,7 ms for Shift and 935,8 ms for Thumbspace). This may be explained by the fact that users tend to place their thumb systematically in the centre of the screen without spending time to adjust the position of the thumb. Once they touch the screen (approximately) at its center they then move the thumb for the same distance as the distance between the center and the target. As the execution time of MagStick is also faster than for other dragging techniques, we hypothesize that the users make an estimation of this distance before touching the screen (only one participant among the twelve has made errors due to a wrong positioning of the thumb).

The rapidity of MagStick may also be explained by Semantic pointing. However this mechanism depends on target density, and should be carefully tested in this context. Our first experiments with a high target density (32 instead of 16 targets of 3mm randomly displayed), shows that MagStick performance is then equivalent to those of Shift and Offset Cursor (while TapTap efficiency is almost the same for both densities). To increase the performance of MagStick, we plan to implement a density-dependent approach that dynamically adapts the strength of the magnetizing effect according to the position of the cursor and its local context on the screen.

Our experimentation also shows that the selection time and the number of errors do not depend on screen areas when using TapTap and MagStick (H3). Conversely, Thumbspace is less efficient in central areas (as also demonstrated in [6]), Shift impairs interaction in the top and left corners because of visual occlusion, and Offset Cursor degrades performance in all screen corners. TapTap and MagStick both provide efficient solutions to these issues as they help users to reach any target in a short and constant time, whatever its location on the screen. MagStick performed well in border areas without decreasing efficiency in the center. Finally, MagStick tends to concentrate most thumb movements in the center as shown in Fig. 9. It also provides a comfortable grip for user interaction in mobility conditions and it is well-adapted to thumb morphologic capabilities.
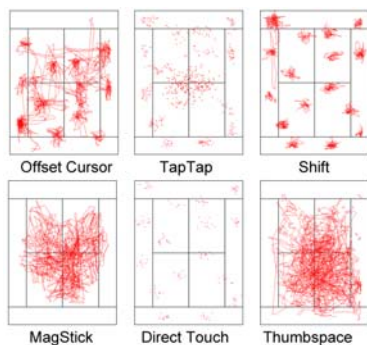

Figure 9. Thumb gesture traces.

6. CONCLUSION AND FUTURE WORK
We have presented TapTap and MagStick, two new interaction techniques that improve target acquisition on small touch-screens for mobile devices. TapTap is based on time-multiplexing through an automatic two-step zooming strategy. MagStick relies on magnetization, a variant of semantic zooming and also makes it possible to predict thumb movements and thus to reduce the net correction distance. Our experiments showed that both techniques are faster and produce fewer errors than the current state of the art.

They also cover the other issues raised by thumb interaction on small touch-screens such as visual occlusion and target accessibility in all parts of the screen. They are also both compatible with interaction techniques relying on dragging gestures. Finally, this paper also offers a significant benefit by presenting a thorough analysis of the techniques published so far.

In future work, we plan to adapt our techniques to constraints that depend on the application context (higher target densities, specific target layouts such as lists or trees…) and to perform further evaluations to evaluate their efficiency under these conditions.

7. ACKNOWLEDGMENTS

8. REFERENCES
1   Albinsson, P. and Zhai, S. 2003. High precision touch screen interaction. *Proc. CHI'03*. 105-112. 2003.
2   Blanch, R., Guiard, Y., Beaudouin-Lafon, M. Semantic pointing: improving target acquisition with control-display ratio adaptation. *Proc. CHI'04*. 519-526. 2004.
3   Guiard, Y., Blanch, R., Beaudouin-Lafon, M. Object pointing: a complement to bitmap pointing in GUIs. *Proc. Graphics interface 2004*. Vol. 62. 9-16. 2004.
4   Guiard, Y., Beaudouin-Lafon, M. (2004). Target Acquisition in Multi-Scale Electronic Worlds. International Journal of Human-Computer Studies, 61, 875-905.
5   Huot, S., Lecolinet, E. Focus+Context Visualization Techniques for Displaying Large Lists with Multiple Points of Interest on Small Tactile Screens. *Proc. Interact'07*.
6   Karlson, A., Bederson, B. ThumbSpace: Generalized One-Handed Input for Touchscreen-Based Mobile Devices. *Proc. Interact'07*. 324-338. 2007.
7   Karlson, A., Bederson, B., Contreras-Vidal, J. Understanding on User Interface Design and Evaluation for Mobile Technology, Idea Group, 2007.
8   Mankoff, J., Hudson, S. E., and Abowd, G. Interaction techniques for ambiguity resolution in recognition-based interfaces. *Proc. UIST'00*. 11-20. 2000.
9   Parhi, P., Karlson, A., Bederson, B. Target Size Study for One-Handed Thumb Use on Small Touchscreen Devices. *Proc. MobileHCI'06*. 203-210. 2006.
10 Pascoe, J., Ryan, N., Morse, D. Using while moving: HCI issues in fieldwork environments. *ACM Trans. Comput.-Hum. Interact.* 7(3):417-437. 2000.
11 Potter, R., Weldon, L., Shneiderman, B. Improving the Accuracy of Touchscreens: An Experimental Evaluation of Three Strategies. *Proc. CHI'88*. 27-32. 1988.
12 Ren, X. and Moriya, S. Improving selection performance on pen-based systems: a study of pen-based interaction for selection tasks. *ACM TOCHI*. 7(3).384-416. 2000.
13 Sears, A., Shneiderman, B. High precision touchscreens: design strategies and comparisons with a mouse. *Int. J. Man-Mach. Stud.* 34(4):593-613. 1991.
14 Vogel, D. and Baudisch, P. 2007. Shift: a technique for operating pen-based interfaces using touch. *Proc. CHI'07*. 657-666. 2007.
15 Grossman, T., Balakrishnan, R. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. *Proc.CHI'05*. 281-290. 2005.