MarkPad: Augmenting Touchpads for Command Selection

Bruno Fruchard[1,2] **Eric Lecolinet**[1] **Olivier Chapuis**[2]

[1]LTCI, CNRS, Telecom ParisTech,
Université Paris-Saclay
F-75013 Paris, France

[2]LRI, Univ. Paris-Sud, CNRS,
Inria, Université Paris-Saclay
F-91405 Orsay, France

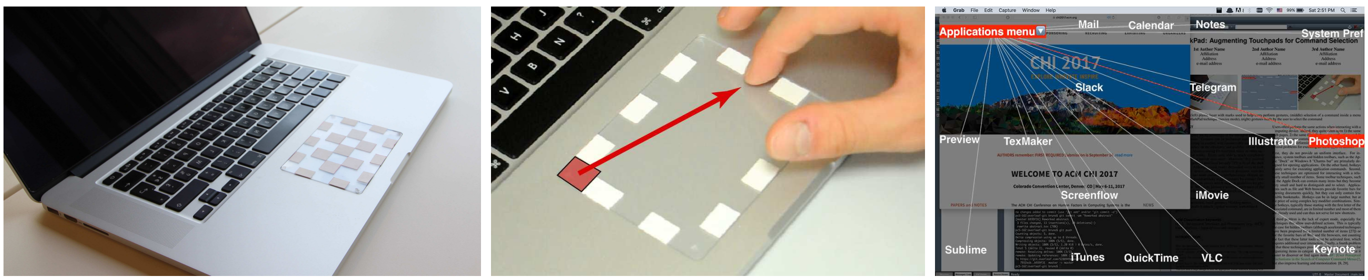

Figure 1: Tactile or visual marks on the touchpad help performing gestures: (Left) Dense configuration of tactile marks, (Middle) Light configuration with marks only on the borders, (Right) Example of a menu in novice mode. This menu and the selected shortcut on its right side correspond to the red area and the red gesture line in the middle picture.

ABSTRACT

We present MarkPad, a novel interaction technique taking advantage of the touchpad. MarkPad allows creating a large number of size-dependent gestural shortcuts that can be spatially organized as desired by the user. It relies on the idea of using visual or tactile marks on the touchpad or a combination of them. Gestures start from a mark on the border and end on another mark anywhere. MarkPad does not conflict with standard interactions and provides a novice mode that acts as a rehearsal of the expert mode. A first study showed that an accuracy of 95% could be achieved for a dense configuration of tactile and/or visual marks allowing many gestures. Performance was 5% lower in a second study where the marks were only on the borders. A last study showed that borders are rarely used, even when the users are unaware of the technique. Finally, we present a working prototype and briefly report on how it was used by two users for a few months.

Author Keywords

Gestural interaction; bezel gestures; tactile feedback; spatial memory; touchpad; user-defined gestures; Marking menus.

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation (e.g. HCI): User Interfaces – Input devices and strategies

http://dx.doi.org/10.1145/3025453.3025486

INTRODUCTION

Users often perform the same actions when interacting with a computing device, such as executing favorite application commands and opening favorite applications, files or Web pages. These actions typically require using different interaction techniques such as menus (for commands), permanent or hidden toolbars (for opening applications) or favorite bars (for opening Web pages). Beside menus, most of these techniques only rely on recognition and do not provide an expert mode. Menus provide hotkeys but "simple" hotkeys, typically those using the first letter of the associated command, are in limited number and most of them are already used. Creating new hotkeys thus tend to require complex key modifier combinations. Marking and Bezel menus [21, 27] are another solution but they only support a limited number of shortcuts at a menu level.

In this paper, we propose a new technique called *MarkPad* (Figure 1) which takes advantage of the touchpad of laptop computers to perform any type of frequent action. MarkPad supports a large number of commands (680 in our experiment) and offers much flexibility for organizing them. It provides an homogeneous way of performing all these different actions and allows grouping them in a way that is meaningful for the user, whatever their type. For instance, one could gather all actions (applications, commands, scripts, Web pages, etc.) related to certain task, activity or project in a single group acting as a kind of conceptual area.

MarkPad both provides a novice and an expert mode and relies on simple gestural shortcuts that are identical in both modes. Its design extends Marking and Bezel menus [21, 27] by taking the size of the gestures into account, which considerably increases the number of possible gestures. In order to achieve sufficient performance, MarkPad relies on augmenting the touchpad by adding *visual* or *tactile* marks on it (Figure 1). This simple solution, which only requires cheap materials (e.g. paper stickers or adhesive tape) makes it possible to use the technique at virtually no cost on current devices.

MarkPad gestures start from the border of the touchpad to avoid conflict with standard interactions. More precisely, they start from user-defined areas (materialized by tactile or visual marks) that correspond to menus. Menu items are activated by performing a gesture starting from one of these areas and ending in another area located where the user wants on the touchpad (Figure 1). Each menu can contain a large number of items (34 in our experiment), which can be laid out in a way that is meaningful to the user by using spatial arrangements that highlight semantic relationships. As this design leverages spatial memory, it should also favor memorization [15, 35, 41]. Moreover, it avoids forcing users to arbitrarily dispatch related items in different menus because of their limited capacity, as with Marking and Bezel menus. Groupings relying on user-defined semantics, not only make it easier to discover or find again items [32] but should also improve learning and memorization [4, 10, 47].

After presenting the related work and describing the technique, we report on two lab studies showing that 95% accuracy can be obtained for a dense configuration of 680 gestural shortcuts and that marks can be removed at the center of the touchpad at the price of slightly lower performance (about 90% accuracy for the same challenging configuration). In a last study, we show that users rarely use the borders of the touchpad, even when they are unaware of the proposed interaction technique, and that the gestures we propose are likely less prone to involuntary activations than directional gestures. Finally, we present a working prototype and briefly report on how it was used by two users (authors of this article).

RELATED WORK

Command selection is one of the elemental actions when interacting with a GUI. Standard techniques include menus, toolbars, hotkeys and gestures. A large body of work in HCI proposes improvements of these techniques as well as adaptations to various contexts, such as mobile interaction. Considering the large number of studies, we focus on a few of them related to gestural interaction, spatial layout and tactile cues.

Gestural interaction and gestural menus. Using gestures for command selection has been proposed as an alternative to menus or hotkeys [3]. Most modern operating systems now allow performing gestures for activating commands. But they are usually specified by manufacturers and bound to specific actions that cannot be changed. Some applications, such as [29, 20] on the Mac, solve this problem by allowing users to define custom actions and gestures. However, few commands elicit gestural agreement, especially if they are of an abstract nature [48]. Providing a mechanism for *discovering* and *learning* gestures is thus of primary importance.

Marking menus [27] solve both problems by offering a novice mode that is similar to the expert mode. Users *discover* gestures by browsing menus and *learn* them through the force of repetition. But Marking menu only support a limited number of items at a given menu level, usually up to eight [27], which may force users or UI designers to employ awkward groupings of menu items [49]. Various derived techniques have been proposed, such as for instance Flower menus [5], which rely on curved gestures to increase capacity, Octopocus [7], which leverages discoverability using feedforward, Augmented letters [38], which combines handwriting and menus to improve memorization. Yet, these techniques do not provide the same capacity and the same flexibility for organizing commands as MarkPad and they require an appropriate activation mechanism because they conflict with other interactions such as drag and drop, drawing or text selection [26].

Spatial layout. Another approach is to use specific areas of the visual or motor space for interaction purpose. Bezel menus [21] avoid conflicting with standard gestures by using only gestures that start from the border of a touch screen, but they only support small menus and might suffer from involuntary activations. Similarly, bezels were used for performing gestures on smartwatches [25] but for a limited set of 16 possible gestures. Using several modalities as in BezelTap [43] (a bump on the bezel and a slide on the screen) prevents involuntary activation but requires slightly more complex gestures and the use of an accelerometer.

Although this tends to be replaced by two-finger gestures, some touchpads still have a dedicated scroll zone on one or several of their sides. Specific areas of the touchpad were used also for performing commands in [9]. This study proposed a set of one-finger and multitouch static gestures (with the `fn` key pressed and at least one finger touching the touchpad border) and dynamic gestures joining one border to another. While the present study also takes advantage of the touchpad borders, it does not rely on the same kind of gestures and proposes an unified technique both providing a novice and an expert mode. Finally, combining locations on a sensitive area and directional gestures was explored in Zone menus [49] for increasing the number of available commands in Marking menus. As other previous Marking menu-derived approaches, this technique conflicts with standard interactions and does not offer the capacity and the flexibility of MarkPad.

Computer systems often provide "hot" corners or hidden toolbars that either trigger commands or display menus when the mouse cursor reaches the corners or the borders of the screen (e.g., the Mac Dock or the Windows Charm Bar). Schramm et al. [42] proposed four techniques for improving hidden toolbars. Using several devices, they found a trade-off between completion time and error detection while using vision-free interfaces. Contrary to MarkPad, these techniques require moving the cursor as they do not rely on the motor space. Moreover they only support a limited number of commands.

Finally, the spatial layout can leverage spatial memory as in CommandMaps [40] that flattens the hierarchy of commands to improve performance, or in FastTap [18] and its adaptation to smartwatches [28] which rely on a static grid that allows fast command selection on tablets and small devices.

Tactile cues. Tactile sense plays an important role for eyes-free interaction or in multi-focus situations. For example, physical buttons or the border of a smartphone can help users to interact with a smartphone [11]. Helping users to alleviate the lack of vision through tactile cues was proposed in several studies. For example, Jansen et al. [22] used tangible widgets fixed on a tablet to interact with a wall display, Corsten et al. [14] used marks on the back of a mobile device to improve pointing on the device, and Harrison et al. [19] compared different types of tangible buttons. Haptic feedback can also be used for different purposes like guiding [17] or providing more information to users [23].

THE MARKPAD TECHNIQUE

Attributes and modalities. MarkPad gestures rely on the combination of several geometric attributes and modalities. While these gestures only depend on their starting and ending areas, users may also perceive them as the combination of a *position*, a *direction* and a *distance*. Touchpads also have a small bezel and four corners that act as *visual* and *tactile* landmarks, which is likely to help interaction [43] as well as the *visual* or *tactile* marks that augment the touchpad.

In a famous article, Miller pointed out that multidimensional stimuli improve absolute judgments [30]. Other studies in psychology and HCI suggest that combined stimuli, or the presence of contextual cues improve learning and memorization [4, 24, 35, 39], which should make our gestures easier to perform and to memorize. While the idea of combining locations and directions has already been proposed for improving Marking menus [49] and Bezel gestures [21, 43], using distance without a visual representation has rarely been studied and either the number of items was relatively small [18] or accuracy was limited [31]. This is because it may be difficult for users to draw strokes of a given length without any visual reference to provide a sense of scale [27, 49]. Our first study will show that *visual* or *tactile* marks (or, more precisely, *visuo-tactile* marks as tactile marks can be seen by the user) solve this problem, even for a dense configuration of a maximum of 680 gestural shortcuts that was used to test the limits of the technique.

Visual/tactile marks. Such marks only involve small modifications of the touchpad and can be unobtrusive. Visual marks can consist of small landmarks. Tactile marks can be almost invisible, for example by using transparent adhesive tape or by slightly changing the surface roughness. Their presence is not annoying on the borders of the touchpad because borders are seldom used, as our last study will show. However, they might be unpleasant in the center of the touchpad, especially during pointing tasks. One solution consists of having marks only in the border areas (Figure 1, middle), a case we will consider in our second study. Another solution would be to use dynamic tactile marks that are activated after the user initiates a gesture from the borders, using technologies providing tactile feedback in real time [1, 8, 12]. Such technology has for instance been used in Métamorphe [6] to promote hotkey usage.

Usage without marks. The technique can also work without marks provided that areas are large enough, thus not too numerous (e.g., 5 horizontally and 3 vertically on the borders [43]). In particular, for border areas, the user can take advantage of the bezel and the corners of the touchpad to position his finger.

Absolute vs. relative pointing. Importantly, MarkPad uses the touchpad as an *absolute* pointing device. This means that it does not conflict with interaction techniques such as active borders, hidden toolbars, charm bars, etc. because these techniques rely on the position of the *mouse cursor*, not on where the user touches the touchpad. In other words, we take advantage of the fact that the touchpad is normally used as an *relative* pointing device. Because we start gestures from the borders, both ways of using the touchpad (i.e. absolute or relative pointing device) do not conflict, except in the rare cases where the user touches the border involuntary. Moreover, the software can prevent cursor movement once a MarkPad gesture is initiated on the border.

Interaction. MarkPad provides three interaction modes (full-novice, expert and semi-novice) depending on user expertise. In *full-novice mode*, the user first holds the Function key (she could also double tap the border, touch it in a special area or use one of the methods described in [42]). The names of the available menus are then displayed on the borders of the screen. Their location correspond to those of the starting areas (or "menu areas") of the MarkPad gestures on the touchpad, which can be materialized by visual or tactile marks.

Once the desired menu is selected by touching the appropriate menu area, the names of the items contained in this menu are displayed on the screen, as well as lines indicating the gestures that must be done for activating these items (Figure 1, right). The menu is transparent, displayed in full-screen mode and superimposed on top of the applications running on the desktop. The user can then perform the desired gesture, which consists in moving his finger from the menu area to the item area. The associated action is triggered when the user releases his finger. The gesture can be canceled by moving the finger to an empty area, or back again to the menu area. All these operations can be performed without looking at the touchpad because the location of the finger touch is continuously displayed on the screen at the same relative location.

Expert mode occurs when the user already knows the gesture. He can then directly perform this gesture without holding the Function key. *Semi-novice mode* is an intermediate case when the user knows the menu area, but not the appropriate gesture. In this case he can double touch the desired menu area, or touch it for at least 500ms, to make this menu appear. This design relies on *rehearsal* [27, 42]. It allows fluid transition from novice to expert use as the same gestures are performed in all cases.

Areas and marks. Menu and menu item areas can either correspond to the tactile/visual marks on the touchpad or to the *absence* of such a mark between two marks (especially along the borders), a "no-mark" being in fact a kind of a mark. Menu areas must be located on the borders, but the

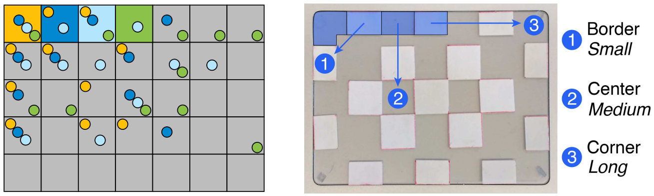

Figure 2: (Left) Evaluated gestures: the circles correspond to the ending areas and their color indicates their starting area (no colors were shown on the actual interfaces). (Right) Three examples of gestures shown on top of the actual *Tactile* interface.

user can place menu item areas wherever she wants. Areas can have various sizes (Figure 9) and can be larger than actual tactile/visual marks. For instance, corner areas do not need to be large, but large areas are beneficial in the center of the touchpad, especially if the technique is used without marks in the center of the touchpad.

As locations do not change, after some practice, users should be able to perform some of the gestures without looking at the touchpad in expert mode, especially when using tactile marks. This would let them focus on their main task by preventing to-and-fro head and eye movements between the screen and the touchpad.

COMMAND SELECTION IN EXPERT MODE

This first study aims at evaluating the performance of the technique in expert mode when visual or tactile marks are present on the touchpad. For this purpose, we built three versions of a plastic layer that we fixed on top of the touchpad (Figure 2-right). The first version had thin *Tactile* marks that the user could see and feel, and the second version *Visual* marks that could only be seen. The last version, called *Mixed*, combined the two previous cases: tactile marks were provided on the borders and visual marks in the center. We built this last version because we thought that while tactile marks might improve performance, they might be inconvenient in the center of the touchpad.

As shown on Figure 2, in this experiment, we used a grid layout containing a large number of rectangular areas ($7 \times 5 = 35$ areas). A valid gesture can start from one of the border areas and end in any area, except the area it was started from. Tactile and visual marks appear in light gray while other areas, which look darker, were left empty. Both types of areas (with or without marks) were used in the experiment. The prototypes were built using cheap materials (plastic sheets, paper stickers, adhesive tape, marker paint...) that anyone could use to customize his own touchpad.

We choose a 7×5 area configuration because 1) it allows for a large number of commands (20 starting areas $\times 35 - 1$ ending areas = 680 shortcuts); 2) according to [43] an odd number of border areas is preferable to an even number; 3) the size of all areas was in line with recommendations for interacting with touchscreens [33, 46]. Areas in the center were about 15×15 mm. Border areas were a bit smaller in order not to occupy too much space. Their smallest side (i.e. their width for left/right border or their height for top/bottom borders) was 10mm. As can be noticed on Figure 2, there was a small dead zone of 5 mm between the border and center areas. As will be explained in the last study, this dead zone is intended to reduce involuntary activations. It was included for the sake of realism but has no or marginal impact on this study.

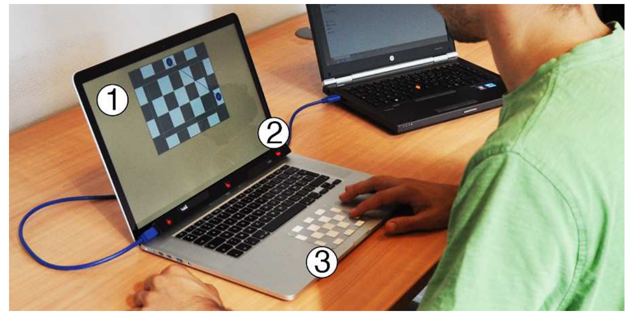

Figure 3: Experiment setup: 1) Stimulus shown during command selection tasks (two blue circles indicate the starting/ending areas), 2) Tobii EyeX device tracking users' gaze, 3) *Tactile* interface on the touchpad.

Experimental design

Hypotheses. We hypothesized that: (**H1**) all interfaces should provide sufficiently high accuracy, but that, because they provide more information, *Tactile* marks should (**H2**) offer better performance and (**H3**) require participants to take less time looking at the touchpad (and/or look at it less often).

As we could not test the performance of all 680 possible gestures, we chose a representative subset considering three characteristics: the type of the starting and ENDING areas and the LENGTH of the gesture. The starting areas were the top left corner, the top middle area (both easier to point at according to [16]) and the two areas between them, as shown on Figure 2. We chose neighboring areas because we thought they might lead to more confusions than areas that would have been far from each other. Moreover, these areas were representative of all the possible kinds of starting areas (corner, middle, between). The ENDING area was either a *Corner*, *Border* or *Center* area and LENGTH was the number of areas between the starting and ending areas (either 0 = *Small*, 1 = *Medium*, 2 = *Long*), as shown on Figure 2. We balanced these characteristics to obtain 14 *Small*, 14 *Medium* and 14 *Long* gestures (thus a total of 42 gestures), with 4 of them ending in a *Corner*, 19 in a *Border* and 19 in a *Center* area.

We used an AUXILIARY task to make the command selection task more realistic. The user had to do this task and then to perform a selection using the technique. The AUXILIARY was either a *Pointing* or *Key input* task. The participants had to place the mouse cursor inside a circle of 35 pixels, 250 pixels away in the first case, and to press the '5' and '7' keys on the keyboard in the second case. We expected that these tasks would impact performance and users' gaze.

Stimulus. After the participant performed the AUXILIARY task, the 7×5 grid was displayed on the laptop screen with a blue circle showing the starting area (Figure 3). When the participant touched the touchpad, the ending area appeared on the grid (also as a blue circle). She could then complete the gesture. We used this design to simulate an expert behavior where the user keeps looking at the screen while performing the gesture on the touchpad. We did not show both areas at

	Visual	*Mixed*	*Tactile*
Success rate (%)	95.64 ± 2.11	94.83 ± 1.84	95.63 ± 1.69
Task duration (in ms)	2013 ± 152	2061 ± 156	2100 ± 125
Execution time (in ms)	975 ± 89	1019 ± 78	1041 ± 67
Number of glances	1.84 ± 0.13	1.72 ± 0.16	1.74 ± 0.15
Glance duration (in %)	36.05 ± 2.4	32.38 ± 3.5	31.85 ± 3.1

Table 1: Summary of the main results for each interface. Mean with 95% Confidence Interval computed relatively to the participants mean.

once because we noticed in pretests that participants then systematically only looked at the touchpad to perform the gesture. This two steps procedure thus encourages participants to keep their focus on the screen.

No feedback was displayed on the screen *while* performing the gesture, but feedback was provided *after* this step. Once the participant released her finger, a line between the starting and ending areas she selected was displayed on the grid. This line was green and shown during 1 second in case of success, but red and shown during 2.5 s (as a penalty) in case of an error. The participant could then start the next trial by pressing the space bar.

Procedure. We asked 12 right-handed participants to perform our study (9 men, 3 women, aged from 22 to 35 years). The whole experiment lasted 30 to 40 minutes. We blocked by INTERFACE (*Tactile*, *Visual* and *Mixed*) and the presentation order was counter-balanced over participants. For each INTERFACE, participants had to perform a training block of 6 trials followed by an evaluation block of 84 trials. Participants performed each gesture two times, each time with a different AUXILIARY task (randomized). Finally, participants were asked to fill a questionnaire at the end of the experiment.

Apparatus. The experiment was performed on a 15” Macbook pro with a 2.2 Ghz Intel Core i7 processor. The size of the touchpad was 105×76 *mm*. We used a Tobii EyeX eye-tracking system [44] running on a Windows 7 computer to record users’ gaze. Both computers were interacting through the UDP protocol. The INTERFACES used for the experiment were made of plastic sheets on which paper stickers were stuck to create tactile marks (Figure 3). Visual marks were drawn using a white Posca pen.

Results

We based our analysis on the following measures: the *success rate*, the *task duration* (total completion time of command selection), the *execution time* (time elapsed between touching and releasing the touchpad for performing the gesture), the *number of glances* at the touchpad and the *glance duration* (in percentage of the *task duration*). Table 1 provides a summary of the results. The main result is that all interfaces reached high accuracy.

ANOVA tests show no significant effect of the INTERFACE on *success rate* ($F_{2,22} = 0.43$, $p = 0.655$), *task duration* ($F_{2,22} = 0.57$, $p = 0.576$), *execution time* ($F_{2,22} = 1.54$, $p = 0.236$) or the *number of glances* ($F_{2,22} = 3.14$, $p = 0.063$). However, another ANOVA reveals a significant effect on the *glance duration* ($F_{2,22} = 9.83$, $p = 0.01$). A post-hoc t-test shows that the *Visual* interface required more time looking at the touchpad than the *Mixed*

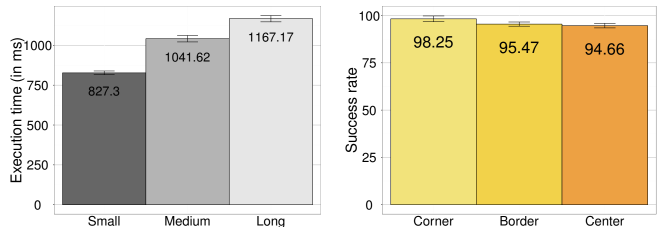

Figure 4: (Left) ***Execution time*** **relative to LENGTH, (Right)** ***Success rate*** **relative to ENDING AREA.**

($p = 0.014$) and *Tactile* ($p < 0.01$) interfaces[1]. While we expected this result, the *execution time* was not significantly different (its mean value was in fact slightly shorter for the *Visual* interface, Table 1). We also ran ANOVAs on the same measures to evaluate the effect of the AUXILIARY task preceding the command selection tasks, but we found no significant differences.

We also looked at the impact of the LENGTH and the ENDING AREA. ANOVAs reveal an effect for both of them on the *success rate* (respectively $F_{2,22} = 6.23$, $p < 0.01$ and $F_{2,22} = 11.32$, $p < 0.01$) and on the *execution time* (respectively $F_{2,22} = 7.04$, $p < 0.01$ and $F_{2,22} = 135.62$, $p < 0.01$). Post-hoc t-tests show that *Small* gestures are completed successfully more often than *Long* gestures (96.73% vs. 94.72%, $p < 0.01$). Unsurprisingly, smaller gestures were also performed significantly faster than longer ones (respectively 827ms, 1042ms, 1167ms, all p's < 0.01, Figure 4). Other t-tests reveal that gestures finishing on a *Corner* were completed successfully more often than others (98.25% vs. *Border* 95.47% and *Center* 94.66%, Figure 4) and that gestures finishing on a *Corner* or a *Border* area were quicker than the ones finishing in the *Center* (respectively 997ms, 978ms and 1049ms).

Finally, we wanted to evaluate when users looked at the touchpad (whether on *touch* or *release* of the gesture). An ANOVA for the *percentage of time* users looked at the touchpad when *releasing* their finger reveals an effect of the INTERFACE ($F_{2,22} = 3.83$, $p = 0.037$). A post-hoc t-test does not reveal any further difference, although *Visual* required users to look while releasing their finger 74% of all trials while *Mixed* and *Tactile* required respectively 65% and 64% of them.

Questionnaire. We asked participants to rank all interfaces from the least to most appreciated with possible ties (on a scale from 1 to 3, with 3 being the best). Small non significant differences were found between them (means: *Visual* 2.00, *Mixed* 2.17 and *Tactile* 2.33). Then we asked them to fill Likert-scales with levels from 1 to 7 about their perception of their *success rate*, their *confidence* level, how often they *looked* at the touchpad and on the *mental demand* required for each INTERFACE. The *Tactile* interface seemed to be better perceived in all cases but no significant difference was found. We also asked participants if they found that tactile marks were disturbing when performing the pointing task. A median value of 5 (slightly agree) was obtained on a scale from 1 to 7 (7 for "completely agree").

[1]All our t-tests are without Bonferroni adjustment for the reasons explained in [34]. However, they stay significant with Bonferroni-Holm adjustment.

Discussion

The main result of this experiment is that all interfaces reached a completion rate of about 95%, which satisfies (**H1**) and validates the feasibility of the technique. A limitation of the experiment is that, because of their large number (680), we only considered a subpart (42 gestures) of the overall possible gestures. However, we tested the technique under somewhat extreme conditions. Not only we used a very dense grid but we chose neighboring starting and ending areas. These two characteristics are likely to increase confusions because gestures were more similar to each other than if they were starting from large and distant areas. In real usage, most users will use less gestures, larger areas (especially in the center of the touchpad) and will favor the most convenient areas (as for instance, *Corner* and *Center* areas).

The *execution time* was about one second and the total *selection time* of about two seconds for all interfaces, which seems reasonably fast as all participants were using the technique for the first time. While comparing studies is uneasy because tasks and devices differ, these results seem in phase with previous research on shortcut techniques. For instance, *selection times* about 2200 ms were obtained for 3×8 hierarchical Marking menus and about 2000 ms for both 16×16 Zones and Polygons menus [27, 49]. Shorter times were obtained for smaller menus (e.g. about 1300 ms for 2×8 Marking menus [27] and 1517 ms for the Swipe technique in [42]), but these results are not comparable as we used a dense configuration with small areas close to each other. Times would probably be shorter for simpler configurations with large areas. As a possible indication, Bezel menus, which can approximately be seen as a kind of MarkPad menu with very large (infinite) areas, were found to have an *execution time* of 382 ms for trained users [21].

As expected, the 14 short gestures were faster (about 830 ms for the *execution time*) and should thus be favored for frequent commands. Gestures along the borders were also slightly faster and offered slightly better success rates. Again, we did not test all possible gestures but, in case some other gestures would be slower, this should not be a problem considering the large number of available gestures. Finally, it is worth noticing that *Key input* tasks were not disadvantaged comparing to *Pointing* tasks, although the fingers were not already on the touchpad in the first case.

Contrary to our expectations, the *Tactile* and *Mixed* interfaces did not provide better *success rate* or *execution time* than the *Visual* interface, which refutes (**H2**). We thought that participants would take less time looking at the touchpad (**H3**) and would thus perform gestures faster (**H2**). While they actually spent less time looking at the touchpad and seemed to make less glances ($p = 0.063$), the execution time was not significantly different. This might be because this setup was new for the participants, or because tactile marks slightly slowed their movements. Interestingly, in all conditions, participants were partly able to interact without looking at the touchpad. Given how the stimulus was displayed, they needed to look two times at the touchpad to see where were their fingers, but the mean *number of glances* was between 1.72 and 1.84 depending on the condition. A further analysis showed this was because they ended the gestures eyes-free in these cases.

Participants seemed to slightly prefer the *Tactile* interface but were also (slightly) concerned about the fact that tactile marks might be annoying, especially in the center of the touchpad. This suggests that the *Mixed* design might be the best alternative. Another option, that we studied in the next experiment, consists in using marks only on the border of the touchpad.

SIMPLIFIED INTERFACES

The goal of this second experiment was to evaluate whether sufficient performance could be achieved without *Visual* or *Tactile* cues in the center of the touchpad. For the sake of comparison, we compared 4 different conditions: *Full Tactile*, *Light Tactile*, *Light Visual* and *None*. *Full Tactile* uses the same *Tactile* interface as in the previous experiment, and serves as a baseline. We chose the *Tactile* rather than the *Visual* interface to gather more results about users feelings with tactile cues. *Light Tactile* and *Light Visual* are similar to the *Tactile* and *Visual* interfaces except they do not have marks in the center. Finally, *None* corresponds to the extreme case where the touchpad does not have any mark.

Experimental design

Procedure. The procedure was the same than in the previous experiment except that we ran the experiment with 16 right-handed participants (11 men, 5 women, aged from 23 to 38 years old, none participated to the previous experiment) and that had to perform a training block of 12 trials using the *Full Tactile* interface, so that they could get used to the technique.

Hypotheses. We hypothesized that (**H1**) the *Full Tactile* interface would lead to the best success rate and the *None* interface to the worst and (**H2**) that participants would take more time looking at the touchpad (and look at it more often) with *Light Tactile* and *Light Visual* than with *Full Tactile* as this interface should help them performing part of the task eyes-free.

Results

Table 2 shows the results for our main measures. We use similar ANOVAs as in the previous experiment. They reveal an effect of the INTERFACE on the *success rate* ($F_{3,45} = 45.30$, $p < 0.01$), the *number of glances* ($F_{3,45} = 10.48$, $p < 0.01$) and the *glance duration* ($F_{3,45} = 2.97$, $p = 0.042$). Post-hoc t-tests show (1) that the *None* interface leads to the worst results ($p < 0.01$ compared to all others) and *Full Tactile* to the best results ($p = 0.011$ vs. both *Light Visual* and *Light Tactile*); (2) that participants looked less often to the touchpad with the *None* interface ($p < 0.01$ vs. *Light Visual* and *Light Tactile*; $p = 0.023$ vs. *Full Tactile*), and less often with the *Full Tactile* than with the *Light Visual* interface ($p = 0.047$); (3) that *glances duration* is shorter for *None* than for *Light Visual* ($p < 0.01$, we found no other significant differences between the other interfaces).

The ANOVA also reveals a significant interaction between INTERFACE and ENDING AREA on *success rate* ($F_{6,90} = 3.32$, $p < 0.01$), see Figure 5. Indeed, post-hoc t-tests show that the *None* interface leads to the worst success rates for all the ENDING AREA (all p's < 0.02), but we find a significant difference between *Full Tactile* and both *Light Visual* and *Light Tactile* only for

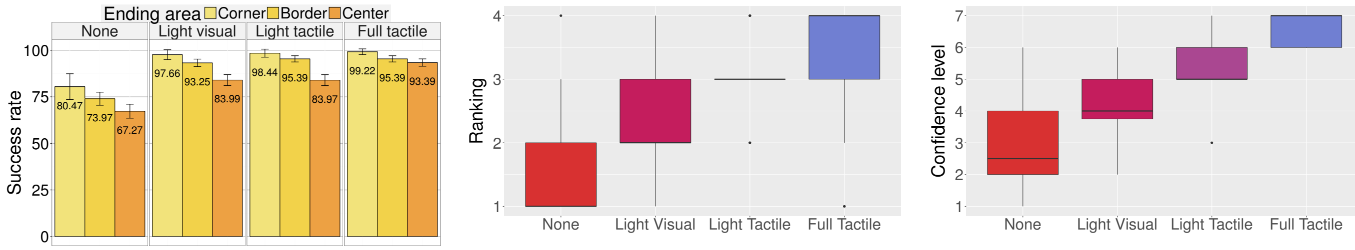

Figure 5: (Left) The *success rate* by ENDING AREA for each INTERFACE; (Middle) ranking of the INTERFACE on a scale from 1 to 4 (4 - best); (Right) participants confidence on the INTERFACE on a scale from 1 to 4 (7 - complete confidence).

	None	*Light Visual*	*Light Tactile*	*Full Tactile*
Success rate (%)	71.56 ± 5.9	89.49 ± 4.3	90.53 ± 3.5	94.85 ± 2.3
Task duration (ms)	2347 ± 284	2388 ± 337	2353 ± 220	2223 ± 161
Execution time (ms)	1161 ± 153	1215 ± 187	1232 ± 132	1144 ± 95
Number of glances	1.58 ± 0.22	1.91 ± 0.22	1.85 ± 0.19	1.76 ± 0.15
Glance duration (%)	30.88 ± 4.2	36.08 ± 4.0	34.28 ± 4.3	34.15 ± 3.6

Table 2: Summary of the results for each interface.

the *Center* area (93.39% for *Full Tactile* vs. 83.99% for *Light Visual* and 83.97% for *Light Tactile*, Figure 5).

We also evaluated *when* users were looking at the touchpad. An ANOVA on the percentage of trials for which users looked at the touchpad while *releasing* their finger shows an effect of the INTERFACE ($F_{3,45} = 9.51$, $p < 0.01$). A post-hoc t-test reveals that *None* requires less glances than the other interfaces when releasing the finger (48.13% of the trials vs. *Light Visual* 69.79%, *Light Tactile* 67.28%, and *Full Tactile* 68.33%, all p's < 0.01).

Questionnaire. A non-parametric Wilcox t-test reveals that the *None* interface was perceived as the worst (all p's < 0.01 except for *Light Visual* $p = 0.016$), as shown on Figure 5. More t-tests show differences for the perceived *success rate* and *confidence level*. *Full Tactile* achieves the best perceived *success rate* (vs. *None* and *Light Visual* $p < 0.01$, vs. *Light Tactile* $p = 0.024$) and *Light Tactile* is perceived as more successful than *Light Visual* ($p = 0.024$) and *None* ($p < 0.01$). A t-test on the *confidence level* shows that participants felt more confident with *Full Tactile* than with the other interfaces, and with *Light Tactile* than with *Light Visual* and *None* (all p's < 0.01), as shown in Figure 5.

Discussion

This experiment shows that marks are needed to obtain a good *success rate*, at least with a dense configuration, which validates (**H1**). While poor results are obtained with *None* (71.6%), they are lower but still acceptable with *Light Tactile* and *Light Visual* (about 90%). As the errors for the light interfaces mainly come from the gestures that end in the center, this suggests that these latter configurations would provide satisfactory success rates with less numerous and larger areas in the center of the touchpad.

Interestingly, the success rate is higher for the *Center* areas with *Light Tactile* and *Light Visual* than with *None* although these interfaces are identical (no marks) in the center. A likely reason is that the participants used the marks on the border in the first cases to guess the location of the areas in the center of the touchpad. Moreover, we noticed that the way they started the gestures was quite important. Marks on the border not only helped participants to start from the proper area but also to locate their finger in the center of this area. This reduces errors when moving the finger to the ending area because the finger tends to "drift". Hence, errors are less likely if the finger is well positioned at the beginning of the gesture.

Participants looked less often at the touchpad with *Full Tactile* than with *Light Visual* but there is no significant difference for the *glance duration* and the *execution time* although mean values show the same trend. Also, mean values slightly differ for *Full Tactile* vs. *Light Tactile* but no significant difference was found. Moreover, the interfaces with *tactile cues* were more comfortable for users since they felt more confident using them. As with the previous experiment, tactile marks seem to help users, which partly validates (**H2**), but a longitudinal experiment with experienced users would be needed to measure this effect more accurately.

Finally, participants looked at the touchpad significantly less often in the *None* condition, which is consistent with the fact that looking at it provided much less information than in the other conditions.

BORDER USAGE

Sensitive surfaces are primary designed for pointing. Bezel gestures [25, 37] and MarkPad are based on the hypothesis that users do not start pointing gestures from the border of the surface. However, they may accidentally touch the border from time to time. In particular, because of its location on the computer, they may inadvertently swipe at the touchpad when moving their hands to use the keyboard.

For this reason we conducted a study for estimating how frequently users perform gestures starting from the borders in real-life situations. Importantly, participants were not aware that an interaction technique might use borders for a specific purpose. As users would probably adapt their behavior in such a case, the results we obtained can be seen as an upper bound of the number of possible involuntary activations.

In addition to MarkPad gestures joining two different areas on the touchpad (here called *"Strokes"*), we also considered 8-directional Marking menu gestures (called *"Slides"*), all of them starting from the touchpad border. After collecting the data, we computed how many gestures would have been detected, depending on two varying factors: the *border width* (W on Figure 6) and the *minimal length* of the gestures.

Gesture minimal length (in mm)	Border width (in mm): 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1.86	3.01	3.58	3.98	4.27	4.57	4.91	5.39	5.95	6.57	7.33	8.19	9.18	10.36	11.73
	0.71	0.94	0.98	1.07	1.15	1.25	1.42	1.71	2.06	2.48	3.03	3.66	4.38	5.22	6.22
3	0.65	1.14	1.44	1.69	1.88	2.11	2.37	2.79	3.27	3.82	4.50	5.26	6.15	7.22	8.47
	0.44	0.65	0.75	0.87	0.96	1.08	1.25	1.53	1.88	2.31	2.85	3.47	4.17	5.00	6.00
5	0.41	0.70	0.91	1.12	1.27	1.46	1.70	2.09	2.55	3.07	3.72	4.44	5.29	6.32	7.52
	0.34	0.54	0.65	0.77	0.86	0.97	1.14	1.43	1.78	2.20	2.73	3.33	4.03	4.86	5.84
7	0.34	0.56	0.71	0.87	1.00	1.18	1.41	1.79	2.23	2.72	3.34	4.03	4.85	5.84	6.99
	0.29	0.47	0.57	0.69	0.78	0.89	1.06	1.35	1.70	2.11	2.62	3.21	3.89	4.70	5.66
9	0.30	0.47	0.60	0.73	0.84	1.01	1.23	1.59	2.01	2.50	3.10	3.77	4.57	5.52	6.63
	0.27	0.42	0.52	0.64	0.72	0.84	1.01	1.29	1.63	2.03	2.54	3.11	3.79	4.58	5.53

Table 3: Detection of Slide and Stroke gestures depending on the *border width* and their *minimal length*.

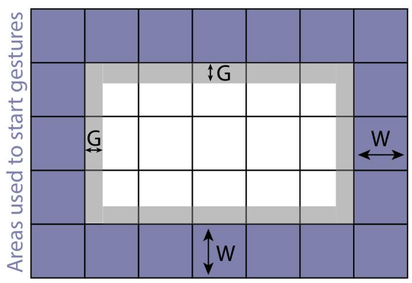

Figure 6: Touchpad configuration used for our study (W is the *border width*, G is the *gap*).

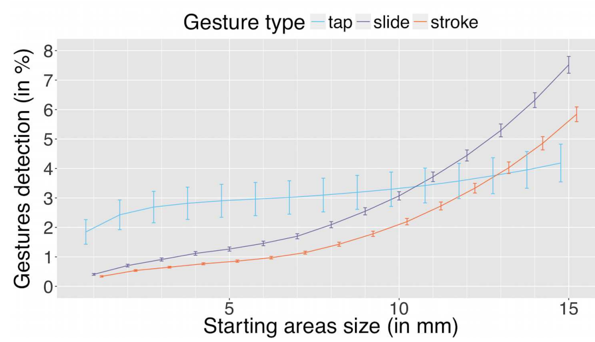

Figure 7: Percentage of Tap (blue), Slide (purple) and Stroke (orange) gestures depending on *border width* for a *minimum length* of 5mm.

We used the same $7 \times 5 = 35$ area configuration as in the previous experiments (Figure 6). The width of the 20 border areas (in blue in the figure), or their height for those on top and bottom borders, depends on *border width*. Directional *Slides* can start from any of these areas and end anywhere, provided they are longer than *minimal length*. Smaller gestures are ignored and would thus have no effect in real-life applications. The same rules apply for MarkPad *Strokes*, except that, by construction, they cannot start and end in the same area.

Procedure. We performed a study on 12 right-handed users (8 men, 4 women) that lasted one week. They all used their own Macbook with an integrated touchpad of 105×76 mm. We implemented a logging software to detect all finger contacts when the computers were being used. The collected data included the number of finger touches, their locations and size. We collected a total of 1 547 680 gestures, with an average number of 128 973 gestures per user (mean gesture duration was about 600ms).

Analysis. After collecting the data, we computed which gestures would have been detected as *Slides* or *Strokes* for a *border width* between 0 and 15 mm and a *minimum length* between 1 and 9 mm. The results are given as percentages of detected *Slides* or *Strokes* relatively to the total number of *"gesture"* on the touchpad. A *"gesture"* consisted of touch followed by a release. We only considered one-finger gestures in this study. For the sake of comparison, we also detected *Taps* occurring in the border area (with a tolerance of 1 mm and a maximum duration of 250 ms).

Results

As can be seen in Figure 7 and Table 3, few gestures were detected (at least for reasonable values of the *minimal length* and the *border width*) although the participants were not aware of the techniques. This means that users seldom use the touchpad borders, at least when the touchpad is relatively large (105×76 mm), a common trend on laptops.

MarkPad *Strokes* were less often detected that directional *Slides*. For instance, for a *minimal length* of 5 mm, the difference was about 20% for a *border width* of 1 mm, then reached a maximum of 50% for 6 mm and, finally decreased to reach 29% for a width of 15 mm. This difference (strongly) increases for smaller values of the *minimal length* and decreases for larger values. The reason for this difference is that *Strokes* cannot end in the area they started from, meaning they must *cross* the edges of their starting area. This reduces the probability of doing the gesture by chance (i.e. only a part of the gestures starting from a given area crosses its edges).

As can be expected, both the *minimal length* and the *border width* strongly impact results. Increasing the *minimal length* to reduce the number of detected gestures may be problematic because users may think the technique is unreactive or buggy if no action occurs. In this regard, a length of about 5 mm may be a reasonable compromise. Conversely, reducing too much the *border width* may lead to misdetections. Experienced users, who will perform the gestures quickly, without paying attention because they are focused on their main task, may not start the gestures from the very border of the touchpad and miss the active area. Hence, this value should probably depend on user habits and preferences. As will be explained in the next section, two users have been using the technique for a few months. After some testing, they chose values between 5.5 and 8.5 mm.

Further analyzing the data, we noticed that many detected gestures ended in the center, close to their starting areas. We thus added a *gap* between the border areas and the center areas (Figure 6) to see whether this would improve results. Fig-

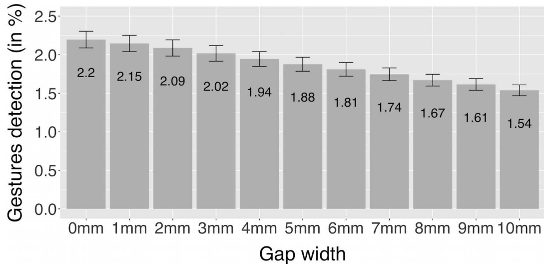

Figure 8: Gap impact on gestures detection (*border width* = 10mm).

ure 8 shows the results for a *gap* varying from 0 to 10 mm with a *border width* of 10 mm and a *minimal length* of 5 mm. As can be seen, this solution has a valuable effect.

Discussion

Our analysis of touchpad usage data shows that users seldom use the touchpad borders, even when unaware of the technique. For this reason, the detection rates obtained can be seen as an upper bound of the number of possible involuntary activations. These rates strongly depend on the *minimal length* and the *border width*. The *minimal length* should be small, to avoid making the interaction unpleasant, but not too small, to avoid involuntary activations. A length of about 5 mm seems a good compromise. Considering this, the detection rate for MarkPad *Strokes* was respectively of 0.34%, 0.86%, 2.20% for a *border width* of 1, 5, 10 mm. Using a *gap* of 10 mm, these values decreased to 0.22%, 0.61%, 1.54%. A gap requires longer Strokes, but only toward the center.

A very small *border width* may make the technique hard to use. Considering the values chosen by the two users who tested the technique, a value between 5 and 10 mm seems a reasonable choice. This is also in phase with recommendations for interacting with touchscreens (about 8 or 9 mm for optimizing accuracy) [33, 46]. Moreover, while we performed our two first experiments width a *border widths* of 10 mm, we noticed that, using the same data, the success rates where almost the same (about 1% lower) for a *border width* of 8 mm. This means that, although the visual/tactile marks were 10 mm large, participants almost only used 80% of their surface. Hence, 8 mm may be a good default value (the detection rate is then of 1.03% for a gap of 10 mm). However, the user should be allowed to customize this value.

Finally, MarkPad *Strokes* were noticeably less often detected that directional *Slides* although the ending areas were not taken into account (except that the ending area could not be the starting area). The difference in results strongly depends on the *minimal length* and the *border width* and can be rather large, especially for small *minimal lengths* (up to 3.7 times). This technique, or, more precisely, the fact it requires crossing the edges of the starting area [2], may thus provide a simple way for reducing involuntary activations when using Bezel menus [21].

PROTOTYPE AND ACTUAL USE

MarkPad has been implemented on MacOS and relies on various mechanisms (system calls, AppleScript, hotkey/event generation, private MacOS multitouch API) for performing all the different actions that have been mentioned so far. It is application agnostic in that sense that it can launch any application, execute almost any command and open any document (provided that an application allows displaying it). It can also send messages to other applications through a TCP/IP connection. The prototype works as a background task and consumes very little CPU and only when executing shortcut actions or when touching the touchpad (about 0.5-1% CPU charge on a MacBook Pro 3 GHz Core i7).

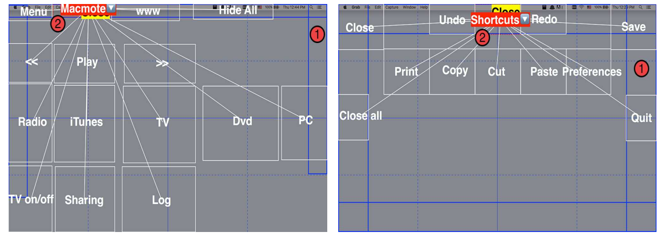

Figure 9: Menu examples for two different users. The (2) red cirles indicate the name of the currently selected menu. The (1) circles show where starting areas can be located (inside the blue rectangles), according to the preferences of each user. Note that the edges of the shortcut areas are only displayed in edition mode to avoid cluttering the display.

The prototype has been continuously used for 3 or 6 months by the two first authors while it was being developed. Interestingly, these two users used the prototype differently in many aspects. While they both created shortcuts and menus for accessing their favorite applications and documents (mainly Web pages) they did not organize them in the same way and did not put the focus on the same actions. For instance, as shown in Figure 9, one user created gestural shortcuts for replacing common hotkeys (copy, paste, etc.) which allowed fast keyboard-free copy-paste [13], while the other user continued using standard hotkeys. Conversely, the second user created a menu for controlling iTunes and his multimedia home environment.

Both users created a menu for opening the most frequently used applications. This menu was considered especially helpful because it allowed fast switching from an application to another. Moreover, activating an application twice in sequence makes it disappear, which allows "glancing" at an application (e.g. quickly looking at the calendar or the mail). Menus contained shortcuts either performing the same kind of system function (e.g. launching applications) or mixing them. For instance, the Multimedia menu shown in Figure 9 contains shortcuts for opening Web pages (a Web interface) and applications (iTunes), activating commands (play/pause, next/previous), or sending commands to a remote application (for switching on the TV, etc.). This highlights the fact that users should not be arbitrarily forced to use different interaction techniques for performing related actions just because they trigger different mechanisms.

Shortcuts areas were positioned in very different ways by the two users. For instance, one user liked using areas on the bottom border of the touchpad, but the other user found this inconvenient and never used such areas, probably because of his specific way of placing hands. Another difference, visible at Figure 9, is that one user liked using areas in the touchpad center, while the other user mainly used areas along or close to the borders. This figure also shows that users customized the sizes of the areas to improve interaction. Initially, the two users also used directional Slides (a possibility offered by the software) but eventually only used MarkPad gestures because they kept creating shortcuts and space was missing.

Both users used tactile marks on the border, but none in the center of the touchpad because they found them inconvenient. One user used almost invisible transparent tape while the other preferred white marks as those shown on Figure 1. Both users used the marks on the border to locate the other areas (i.e. areas without marks towards the center of the touchpad). They used different border widths (about 5.5 and 8.5 mm) and mainly used the expert mode, which is consistent with the fact they created their own shortcuts and menus. These different strategies highlight the numerous possibilities offered by the technique and the importance of customization.

We developed a shortcut editor which allows associating MarkPad Strokes (and directional Slides) with all the above-described actions (Figure 9). This editor can be invoked by clicking on the MarkPad icon on the MacOS menu bar or by pressing the mouse on a menu name while the Function key is held down. As said before, pressing this key shows all menu names. The user can then press the mouse on the desired menu name, then drag the mouse to create a new shortcut that is positioned where it is released. The editor is then automatically opened so that the user can specify the desired action. Various actions (such as generating common hotkeys) are predefined and only require selecting them in a menu. Opening applications, scripts or documents is almost as simple: the user just needs to select the corresponding action in a menu and paste or type the proper URL or command name. Help is provided for more complex commands.

As said above, this system allows for creating one's own customized vocabulary of gestures, but this may be too burdensome for novice users. A solution is to provide predefined sets of gestures activating common actions that the user would only customize if needed. More specialized sets of actions corresponding to functional themes could also be proposed. As the technique is especially useful for bookmarking Web pages and accessing them quickly, a functionality which is rarely offered by current software (or only for few items via a small favorite bar), it could be specially adapted for this purpose. As another direction, this system could also serve to implement some sort of a multiple-object clipboard that would allow users to easily store and access temporary content as in [36].

Finally, another important issue is the cost of mistakes (e.g. closing the wrong application), which may happen from time to time. This problem is already taken into account by quickly displaying the name of the activated command on the screen after performing a gesture, so that the user can know which command she triggered. However, a more sophisticated history mechanism may be useful to retrieve what gesture was fired, and, ideally, to undo undesired actions. Morever, the editing interface could guide the user to help him chose gestures that are very unlikely to be triggered unvoluntary for activating dangerous commands.

CONCLUSION AND FUTURE WORK

We presented MarkPad, a technique taking advantage of the touchpad to perform size-dependant gestural shortcuts. By taking the size of the gestures into account, which considerably increases the number of possible gestures, MarkPad supports a large number of commands and offers much flexibility for organizing them. MarkPad relies on augmenting the touchpad by adding visual or tactile marks on it, a simple solution that only requires cheap materials and allows using the technique at virtually no cost on current devices.

We reported on the results of two experimental studies evaluating command selection in expert mode with a dense configuration of 680 possible gestures, more than all existing techniques with a comparable input channel. Using a full tactile or visual interface to help selection, we obtained an accuracy of 95% and a completion time similar to well known techniques. Subjective results suggested that a mixed interface (borders with tactile marks and center with visual marks) was a promising solution. The accuracy reached 90% without marks at the center (without speed lost), but the results suggest that this simpler interface is viable when used with larger areas in the center of the touchpad. These studies also revealed that users need to look more at the touchpad using visual cues, which, interestingly, did not impede selection time.

We evaluated the usability of our design in everyday use with a study on 12 participants that lasted one week. It confirmed our assumption that borders are seldom used for pointing interaction, but still enough to trigger involuntary gestures. Our analysis highlighted 1) the impact of the width of the border areas, 2) that a gap between border and center areas reduces gesture detection, 3) that the gestures we propose are less prone to involuntary activation than directional slides. A MarkPad running prototype was finally presented along with two users feedback during a few months.

In future work, we plan to adapt the technique to touchscreens using technologies providing dynamic tactile feedback and to the mouse, using, for instance, a multitouch mouse [45]. Other ideas include (1) replacing the touchpad with a small touchscreen to dynamically display possible actions, (2) mapping the top border marks with the current application menus to allow accessing them directly without moving the cursor, (3) mixing MarkPad gestures with text entry (by pressing key(s) in sequence while touching a mark), thus creating a new type of hotkeys, (4) using new gestures, as for instance Augmented letters [38] starting from the borders. Finally, as the MarkPad design leverages spatial memory, it should also favor memorization, an aspect we also plan to study.

ACKNOWLEDGMENTS

This research was partially funded by Labex DigiCosme (ANR-11-LABEX-0045-DIGICOSME), operated by the French Agence Nationale de la Recherche (ANR) as part of the program "Investissement d'Avenir" Idex Paris-Saclay (ANR-11-IDEX-0003-02). The authors would like to thank Gilles Bailly and Sylvain Malacria for fruitful discussions, and the reviewers for their comments that helped improve the paper.

REFERENCES

1. Actronika. Tactronik. Retrieved January 2017 from http://www.actronika.com/
2. Georg Apitz and François Guimbretière. 2004. Crossy: A crossing-based drawing application. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST '04)*. ACM, 3–12. DOI: 10.1145/1029632.1029635
3. Caroline Appert and Shumin Zhai. 2009. Using strokes as command shortcuts: Cognitive benefits and toolkit support. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, 2289–2298. DOI: 10.1145/1518701.1519052
4. A.D. Baddeley. 2013. *Essentials of Human Memory (Classic Edition)*. Psychology Press, East Sussex, UK.
5. Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2008. Flower menus: A new type of marking menu with large menu breadth, within groups and efficient expert mode memorization. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '08)*. ACM, 15–22. DOI: 10.1145/1385569.1385575
6. Gilles Bailly, Thomas Pietrzak, Jonathan Deber, and Daniel J. Wigdor. 2013. Métamorphe: Augmenting hotkey usage with actuated keys. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, 563–572. DOI: 10.1145/2470654.2470734
7. Olivier Bau and Wendy E. Mackay. 2008. Octopocus: A dynamic guide for learning gesture-based command sets. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST '08)*. ACM, 37–46. DOI: 10.1145/1449715.1449724
8. Olivier Bau, Ivan Poupyrev, Ali Israr, and Chris Harrison. 2010. Teslatouch: Electrovibration for touch surfaces. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, 283–292. DOI: 10.1145/1866029.1866074
9. Mathieu Berthellemy, Elodie Cayez, Marwan Ajem, Gilles Bailly, Sylvain Malacria, and Eric Lecolinet. 2015. SpotPad, LociPad, ChordPad and InOutPad: Investigating gesture-based input on touchpad. In *Proceedings of the 27th Conference on L'Interaction Homme-Machine (IHM '15)*. ACM, Article 4, 8 pages. DOI: 10.1145/2820619.2820623
10. Gordon H. Bower, Michal C. Clark, Alan M. Lesgold, and David Winzenz. 1969. Hierarchical retrieval schemes in recall of categorized word lists. *Journal of Verbal Learning and Verbal Behavior* 8, 3, 323–343. DOI: 10.1016/S0022-5371(69)80124-6
11. Andrew Bragdon, Eugene Nelson, Yang Li, and Ken Hinckley. 2011. Experimental analysis of touch-screen gesture designs in mobile environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, 403–412. DOI: 10.1145/1978942.1979000
12. Géry Casiez, Daniel Vogel, Qing Pan, and Christophe Chaillou. 2007. RubberEdge: Reducing clutching by combining position and rate control with elastic feedback. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST '07)*. ACM, 129–138. DOI: 10.1145/1294211.1294234
13. Olivier Chapuis and Nicolas Roussel. 2007. Copy-and-paste between overlapping windows. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, 201–210. DOI: 10.1145/1240624.1240657
14. Christian Corsten, Christian Cherek, Thorsten Karrer, and Jan Borchers. 2015. Hapticase: Back-of-device tactile landmarks for eyes-free absolute indirect touch. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, 2171–2180. DOI: 10.1145/2702123.2702277
15. Mary Czerwinski, Maarten Van Dantzich, George Robertson, and Hunter Hoffman. 1999. The contribution of thumbnail image, mouse-over text and spatial location memory to web page retrieval in 3D. In *Proceedings of INTERACT*, Vol. 99. ISO Press, 163–170. http://www.msr-waypoint.net/en-us/um/people/marycz/interact99.pdf
16. Jérémie Gilliot, Géry Casiez, and Nicolas Roussel. 2014. Impact of form factors and input conditions on absolute indirect-touch pointing tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, 723–732. DOI: 10.1145/2556288.2556997
17. Tovi Grossman, Xiang Anthony Chen, and George Fitzmaurice. 2015. Typing on glasses: Adapting text entry to smart eyewear. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '15)*. ACM, 144–152. DOI: 10.1145/2785830.2785867
18. Carl Gutwin, Andy Cockburn, Joey Scarr, Sylvain Malacria, and Scott C. Olson. 2014. Faster command selection on tablets with FastTap. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, 2617–2626. DOI: 10.1145/2556288.2557136
19. Chris Harrison and Scott E. Hudson. 2009. Providing dynamically changeable physical buttons on a visual display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, 299–308. DOI: 10.1145/1518701.1518749
20. Andreas Hegenberg. BetterTouchTool application. Retrieved January 2017 from https://www.boastr.net
21. Mohit Jain and Ravin Balakrishnan. 2012. User learning and performance with bezel menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, 2221–2230. DOI: 10.1145/2207676.2208376

22. Yvonne Jansen, Pierre Dragicevic, and Jean-Daniel Fekete. 2012. Tangible remote controllers for wall-size displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, 2865–2874. DOI: 10.1145/2207676.2208691

23. Yvonne Jansen, Thorsten Karrer, and Jan Borchers. 2010. Mudpad: A tactile memory game. In *ACM International Conference on Interactive Tabletops and Surfaces (ITS '10)*. ACM, 306–306. DOI: 10.1145/1936652.1936734

24. William P. Jones and Susan T. Dumais. 1986. The spatial metaphor for user interfaces: Experimental tests of reference by location versus name. *ACM Trans. Inf. Syst.* 4, 1, 42–63. DOI: 10.1145/5401.5405

25. Yuki Kubo, Buntarou Shizuki, and Jiro Tanaka. 2016. B2B-Swipe: Swipe gesture for rectangular smartwatches from a bezel to a bezel. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, 3852–3856. DOI: 10.1145/2858036.2858216

26. Gordon Kurtenbach and William Buxton. 1991. Issues in combining marking and direct manipulation techniques. In *Proceedings of the 4th Annual ACM Symposium on User Interface Software and Technology (UIST '91)*. ACM, 137–144. DOI: 10.1145/120782.120797

27. Gordon Kurtenbach and William Buxton. 1993. The limits of expert performance using hierarchic marking menus. In *Proceedings of the INTERCHI '93 Conference on Human Factors in Computing Systems (INTERCHI '93)*. IOS Press, 482–487. http://dl.acm.org/citation.cfm?id=164632.164977

28. Benjamin Lafreniere, Carl Gutwin, Andy Cockburn, and Tovi Grossman. 2016. Faster command selection on touchscreen watches. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, 4663–4674. DOI: 10.1145/2858036.2858166

29. MagicPrefs. Magic Trackpad application. Retrieved January 2017 from http://magicprefs.com

30. George A Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review* 63, 2, 81. DOI: 10.1037/h0043158

31. Mathieu Nancel and Michel Beaudouin-Lafon. 2008. *Extending Marking Menus with Integral Dimensions: Application to the Dartboard Menu*. Technical Report Technical report 1503. LRI.

32. K. L. Norman. 1991. *The Psychology of Menu Selection: Designing Cognitive Control at the Human/Computer Interface*. Ablex Publishing Corp., Norwood, NJ, USA.

33. Pekka Parhi, Amy K. Karlson, and Benjamin B. Bederson. 2006. Target size study for one-handed thumb use on small touchscreen devices. In *Proceedings of the 8th Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '06)*. ACM, 203–210. DOI: 10.1145/1152215.1152260

34. Thomas V. Perneger. 1998. What's wrong with bonferroni adjustments. *British Medical Journal* 316, 7139, 1236–1238. DOI: 10.1136/bmj.316.7139.1236

35. Simon T. Perrault, Eric Lecolinet, Yoann Pascal Bourse, Shengdong Zhao, and Yves Guiard. 2015. Physical loci: Leveraging spatial, object and semantic memory for command selection. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, 299–308. DOI: 10.1145/2702123.2702126

36. Jeffrey Pierce, Matthew Conway, Maarten van Dantzich, and George Robertson. 1999. Toolspaces and glances: storing, accessing, and retrieving objects in 3d desktop applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. ACM, 163–168.

37. Volker Roth and Thea Turner. 2009. Bezel swipe: Conflict-free scrolling and multiple selection on mobile touch screen devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, 1523–1526. DOI: 10.1145/1518701.1518933

38. Quentin Roy, Sylvain Malacria, Yves Guiard, Eric Lecolinet, and James Eagan. 2013. Augmented letters: Mnemonic gesture-based shortcuts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, 2325–2328. DOI: 10.1145/2470654.2481321

39. Joey Scarr, Andy Cockburn, and Carl. Gutwin. 2012a. Supporting and exploiting spatial memory in user interfaces. *Foundations and Trends in Human-Computer Interaction* 6, 1, 1–84.

40. Joey Scarr, Andy Cockburn, Carl Gutwin, and Andrea Bunt. 2012b. Improving command selection with CommandMaps. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, 257–266. DOI: 10.1145/2207676.2207713

41. Joey Scarr, Andy Cockburn, Carl Gutwin, and Sylvain Malacria. 2013. Testing the robustness and performance of spatially consistent interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, 3139–3148. DOI: 10.1145/2470654.2466430

42. Katherine Schramm, Carl Gutwin, and Andy Cockburn. 2016. Supporting transitions to expertise in hidden toolbars. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, 4687–4698. DOI: 10.1145/2858036.2858412

43. Marcos Serrano, Eric Lecolinet, and Yves Guiard. 2013. Bezel-tap gestures: Quick activation of commands from sleep mode on tablets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, 3027–3036. DOI: 10.1145/2470654.2481421

44. Tobii Eyetracking. Tobii EyeX. Retrieved January 2017 from http://www.tobii.com/xperience/

45. Nicolas Villar, Shahram Izadi, Dan Rosenfeld, Hrvoje Benko, John Helmes, Jonathan Westhues, Steve Hodges, Eyal Ofek, Alex Butler, Xiang Cao, and Billy Chen. 2009. Mouse 2.0: Multi-touch meets the mouse. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology (UIST '09)*. ACM, 33–42. DOI: 10.1145/1622176.1622184
46. Daniel Vogel and Patrick Baudisch. 2007. Shift: A technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, 657–666. DOI: 10.1145/1240624.1240727
47. Julie Wagner, Eric Lecolinet, and Ted Selker. 2014. Multi-finger chords for hand-held tablets: Recognizable and memorable. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, 2883–2892. DOI: 10.1145/2556288.2556958
48. Jacob O. Wobbrock, Htet Htet Aung, Brandon Rothrock, and Brad A. Myers. 2005. Maximizing the guessability of symbolic input. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA '05)*. ACM, 1869–1872. DOI: 10.1145/1056808.1057043
49. Shengdong Zhao, Maneesh Agrawala, and Ken Hinckley. 2006. Zone and polygon menus: Using relative position to increase the breadth of multi-stroke marking menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, 1077–1086. DOI: 10.1145/1124772.1124933