Side-Crossing Menus: Enabling Large Sets of Gestures for Small Surfaces

BRUNO FRUCHARD, Saarland University, Saarland Informatics Campus, Germany
ERIC LECOLINET, LTCI, Télécom Paris, Institut Polytechnique de Paris, France
OLIVIER CHAPUIS, Université Paris-Saclay, CNRS, Inria, LRI, France

Supporting many gestures on small surfaces allows users to interact remotely with complex environments such as smart homes, large remote displays, or virtual reality environments, and switching between them (e.g., AR setup in a smart home). Providing eyes-free gestures in these contexts is important as this avoids disrupting the user's visual attention. However, very few techniques enable large sets of commands on small wearable devices supporting the user's mobility and even less provide eyes-free interaction. We present Side-Crossing Menus (SCM), a gestural technique enabling large sets of gestures on a smartwatch. Contrary to most gestural techniques, SCM relies on broad and shallow menus that favor small and rapid gestures. We demonstrate with a first experiment that users can efficiently perform these gestures eyes-free aided with tactile cues; 95% accuracy after training 20 minutes on a representative set of 30 gestures among 172. In a second experiment, we focus on the learning of SCM gestures and do not observe significant differences with conventional Multi-stroke Marking Menus in gesture accuracy and recall rate. As both techniques utilize contrasting menu structures, our results indicate that SCM is a compelling alternative for enhancing the input capabilities of small surfaces.

CCS Concepts: • **Human-centered computing → Gestural input**.

Additional Key Words and Phrases: Gestural interaction; Marking menus; Small surface; Eyes-free interactions

ACM Reference Format:
Bruno Fruchard, Eric Lecolinet, and Olivier Chapuis. 2020. Side-Crossing Menus: Enabling Large Sets of Gestures for Small Surfaces. *Proc. ACM Hum.-Comput. Interact.* 4, ISS, Article 189 (November 2020), 19 pages. https://doi.org/10.1145/3427317

1 INTRODUCTION

Eyes-free interaction is especially useful when users interact remotely with their surrounding environment. For instance, users can control Internet-of-Things devices rapidly in a smart home, or when interacting with a remote large display, keep their visual focus on the primary task without interruptions. It is also beneficial when users cannot see the input device, for instance, when using opaque Head-Mounted Displays (HMD) in Virtual Reality (VR) setups. These environments are becoming more and more complex, as they require users to interact with more smart objects, with multiple displays at the same time (e.g., visualisation [26] and control rooms [41]), or to perform complex visual analytics tasks [14, 35].

Authors' addresses: Bruno Fruchard, Saarland University, Saarland Informatics Campus, Campus E 1.7, 66123, Saarbrücken, Germany; Eric Lecolinet, LTCI, Télécom Paris, Institut Polytechnique de Paris, Palaiseau, 91120, France; Olivier Chapuis, Université Paris-Saclay, CNRS, Inria, LRI, Bât. 660, Orsay, 91405, France.

2573-0142/2020/11-ART189 $15.00
https://doi.org/10.1145/3427317

Supporting user mobility is essential in these environments [8]. Small wearable devices like smartwatches are adequate as they do not hinder mobility while leveraging proprioceptive skills to interact eyes-free. However, they only provide small interactive surfaces that limit touch interactions [53]. Gestural interactions, and particularly stroke movements, adapt to small surfaces [10, 22, 29, 55, 58], but only a few support eyes-free interactions [62].

We present *Side-Crossing Menus* (SCM), a gestural technique enabling large sets of gestures on small surfaces (172 gestures on a smartwatch) to issue multiple commands consistently in various environments. The design of SCM aims at optimizing the input capabilities of small interactive surfaces. It relies on a grid interface [24, 33] and defines a gesture as an arbitrary stroke between two cells that enters the ending cell through a specific side. An SCM gesture thus involves two original characteristics: a) it depends on the side crossed to enter the final cell, which substantially increases the number of possible gestures, and b) it does not force users to perform a specific shape nor to follow a specific path between the two cells. As observed in our experiments, users can leverage this feature to favor speed (small quick gestures) versus reliability (long safe gestures), or draw shapes that are easier for them to remember. Furthermore, in contrast with conventional gestural techniques [59], this design relies on broad and shallow menus that reveal their set of commands at once and do not require users to navigate deep hierarchies.

Many gestural techniques provide a smooth transition from novice to expert usages [31, 59]. To preserve this advantage, we focus on promoting eyes-free interactions with SCM by building on bezel gestures [43]. Bezel gestures enable users to place their finger correctly on the bezels of the interactive surface before performing a gesture on the sensitive surface. Moreover, they provide an explicit modifier for avoiding conflicts with common interactions, like scrolling gestures on a smartwatch. To support accurate eyes-free interactions, we leverage passive tactile feedback using tactile aids on the bezels [10, 22, 55], and active tactile feedback using short vibrations when the user enters a cell.

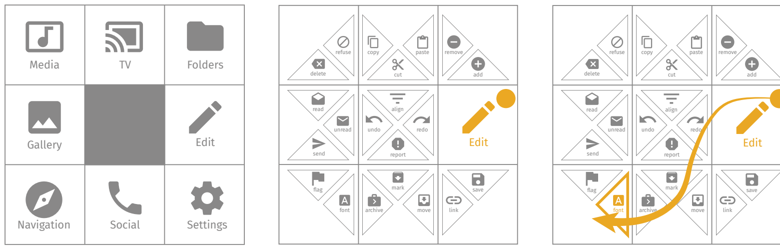
To validate the design of SCM, we report results from two experiments. The first focuses on expert (eyes-free) selections and suggests that users require little training to master SCM gestures (95% accuracy after training 20 minutes on a representative set of 30 gestures among 172). In the second experiment, we sought to understand how efficiently users learn SCM gestures as the SCM structure contrasts with conventional techniques. Hence, we compared our approach to Multi-stroke Marking Menus (MMM) [62] and did not find significant differences in the gesture accuracy and recall rate. This result is promising as it hints that SCM is likely a good alternative to this conventional baseline, while providing a complementary hierarchical structure. Furthermore, our results provide empirical data on gesture learning using stroke gestures, which remain rarely studied in the literature [21, 38].

2 SIDE-CROSSING MENUS

In this section, we present the design of SCM and its various features. We also explain how we support expert users performing eyes-free gestures with simple passive and active tactile aids.

2.1 Interaction Design

The design of SCM relies on a grid interface [24, 33]. On a smartwatch, this grid typically consists of $3 \times 3 = 9$ cells [33] (Figure 1). SCM builds on bezel gestures to facilitate eyes-free interactions and avoid conflicting with common interactions such as pointing and swipe gestures [29, 43]. To distinguish these gestures without actuating the bezels, we make SCM gestures start from the border cells (only touches less than 3mm away from the bezel are considered as the start of a gesture). As several gestures can originate from a unique starting cell, a starting cell corresponds to an *SCM menu* (e.g., the middle-right starting cell corresponds to the menu "edit" on Figure 1).



(a) (b) (c)

Fig. 1. SCM novice mode rendered using an auxiliary display, e.g., a headset, a large display, a smart glass. (a) An SCM menu is bound to a starting cell (eight menus are depicted). (b) Commands are located in arrowheads indicating how to enter ending cells. The orange circle depicts the user's initial touch. (c) The user performs a SCM gesture to select a command. The orange arrow depicts the user's trace.

An *SCM Gesture* consists of a stroke from one cell to another by crossing a specific side of the latter (Figure 1c). Three components merely characterize it: the *starting* cell, the *ending* cell (different from the starting cell and not necessarily on the border), and the *side* of the ending cell crossed to enter it. Because they depend on two areas (a starting and an ending cell), these gestures are *position-dependent*.

2.1.1 Number of Commands and Layout. Using a grid of 9 cells, an SCM menu can contain between 21 and 22 commands (depending on its location, an ending cell can support 2 to 4 commands, Figure 1b and 1c). The technique can thus support a total of $21\times4+22\times4=172$ commands.

While providing many commands on a small surface is an advantageous feature of SCM, one is unlikely to use such a large set of gestures on a small device. The real advantage is to give freedom to designers or users in the way they design the menu system. Indeed, because of its large size, an SCM menu can usually contain all related commands to avoid using submenus, contrary to techniques that only support a small number of gestures at each level of the menu as Marking menus (8 items). This feature avoids forcing designers to use deep and narrow hierarchies when they think this is undesirable [31, 62]. Moreover, using only a subset of all possible gestures enables creating spatial groupings to group related commands can together. This feature is common on the Desktop, through "inner groups" of commands separated by horizontal lines in linear menus. Nevertheless, linear menus do not support eyes-free interaction and techniques that do usually lack efficient means for grouping related commands.

2.1.2 Interaction Modes. SCM provides a *novice mode*, rendered using an auxiliary display, for depicting SCM gestures allowing to learn the associations between the commands and the gestures (Figure 1). Dwelling on a menu cell for more than 300ms triggers the novice mode and reveals all the commands this menu comprises. Commands are depicted inside arrowheads that indicate which side of the cell the users must cross to trigger them (Figure 1b and 1c). When entering a cell, the users highlight the command they will trigger (Figure 1c). They can then trigger this command by releasing their finger from the surface. They can cancel a gesture by moving their finger back to the starting area. While we considered canceling by sliding out of the interactive surface, this would override bezel-to-bezel gestures [29] that SCM supports.

Considering the large number of possible gestures and the small size of a smartwatch, the visual rendering of the novice mode cannot reasonably be rendered on the watch, and is, thus, rendered on an auxiliary display depending on the context: a VR/AR headset, a TV set (smart home), or a large screen display. Similarly, one could use smart glasses [48] or on-arm projected graphics [58] in a mobile or smart home context. Once users can *recall* a set of gestures (expert users), they can perform them eyes-free.

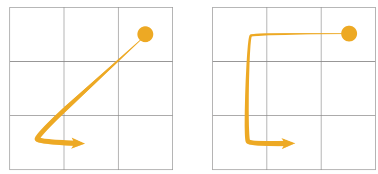


Fig. 2. Example of two strategies for selecting the same command. The user maximizes speed without relying on vibrations (left) or maximizes accuracy by relying on them and the edges of the interactive surface (right).

2.1.3 Path Flexible Gestures. SCM gestures are not necessarily straight lines and offer much path flexibility to users for performing gestures according to their preferences. Despite Marking menus share this feature [31, 61, 62], SCM emphasizes this flexibility by providing loose constraints. Users can thus adopt various strategies for performing gestures. For instance, they might prefer moving their finger slowly, cell by cell, to ensure accurate movement, or sliding it rapidly to the ending area for faster interaction (Figure 2).

Moreover, enabling users to perform multiple paths provides several advantages: (1) users can correct gestures that are partially erroneous by changing their trajectory, (2) they can use gestures that they find easier to memorize (e.g., symbolic shapes [4, 34], *metaphorical* gestures [28, 54], or shapes like phone unlock patterns), and (3) can adapt their movements according to the constraints of the interaction context (e.g., thumb or index finger used for input).

2.1.4 Simple Recognition Algorithm. Although users can draw complex paths instead of straight lines, SCM gestures can be easily and reliably recognized because the program only needs to detect the starting and ending cells and the direction from which the ending cell is entered (which can be done trivially by checking the position of the user's finger just before entering this cell). Therefore, there is no need for a complex gesture recognition algorithm [56] and this design ensures an almost perfect recognition rate.

2.2 Passive and Active Tactile Aids

SCM relies on small movements performed on a surface. This characteristic is especially advantageous when one uses SCM in AR or VR environments [17] as it avoids performing large tiring 3D gestures. However, performing small gestures without visual assistance can be challenging [11, 22]. We thus propose two kinds of tactile aids for supporting eyes-free interactions.

(a) (b)

Fig. 3. The bezels of a watch can provide tactile aids to help the users interacting eyes-free. (a) The smartwatch used in our experiments augmented with tactile aids (paper stickers highlighted with dotted lines) on the bezels. (b) An example of a commercial watch with bumps integrated to its design that one could use as tactile aids (highlighted with dotted lines).

Passive Tactile Aids provide help for initiating gestures. They consist of stickers placed on the bezels that users can feel under their fingertip (depicted inside dotted white lines in Figure 3a). The *locations* and *shapes* of the spaces *between* the stickers respectively indicate the ideal *locations* and *directions* to start the gestures. This simple aid noticeably improves performance because it prevents users from drifting their finger to the wrong cell. These aids are of little cost as the stickers do not cover the touchscreen nor drastically impact the smartwatch's aesthetic. In fact, they are hardly visible and could be integrated into the bezels of a commercial product (Figure 3b).

Active Tactile Aids provide help for completing gestures. They consist of short vibrations emitted when the user's finger crosses the side of a cell. This mechanism is a simple way to notify users their finger just entered a cell and one can count the total number of vibrations corresponding to a given path (e.g., the path on Figure 1c would involve four vibrations). An advantage of active aids over passive aids is that they do no modify the surface, which might be uncomfortable when performing drag or swipe operations in the center of the screen.

A drawback of using vibrations is that they slow down the interaction because of their latency. However, active tactile aids are mostly intended for helping the transition from novice to expert use. We assume that users would no longer use them after some training, i.e., when they know the paths sufficiently well.

3 RELATED WORK

In this section, we contextualize our work according to relevant literature. Primarily, we present stroke gestural techniques [59] supporting eyes-free interaction on small touch surfaces and demonstrate how our approach differs and complements them.

3.1 Marking Menus and Variants

Marking Menus [30] and their variants use *size-independent* directional gestures [7] that enable eyes-free interactions [11, 27, 49] but strongly constrain each menu level to eight items [31]. They rely on deep and narrow hierarchies that one can navigate using compound marks [31]. While these compound marks require large surfaces to interact, Multi-stroke Marking Menus [62] only require small surfaces by relying on successive independent strokes. However, this may conflict with the idea that chunking [12] should help perform and memorize commands.

Previous work also considered additional dimensions such as taps [49], stroke curvature [6], or the starting position [61] of gestures to increase the number of commands while avoiding deep hierarchies of menus. While most of these techniques have been designed for the mouse or the trackpad, some of them have been adapted to smartphones [44], smartwatches [15], and smart glasses [22] but only provide a limited set of commands (28 in [44]) or text entry. These two last studies ([15, 22]) focused on speed rather than accuracy and revealed error rates of approximately 20%.

While the deep and narrow structure of Marking menus constrains the number of items per level and lacks the conventional pre-visualization of linear menus [5], it facilitates visual search by displaying a small set of items at once. Conversely, SCM's broad and shallow structure imposes a demanding visual search but supports a large number of items in a menu and shows them all at once.

3.2 Position-Dependent Gestures

Position-dependent gestures [19, 25, 63] require the user to start and end gestures at specific locations. Compared to techniques using *size-independent* gestures, such as Marking menus, this feature considerably increases the gesture set's size. However, they are more difficult to perform eyes-free [36]. This probably explains why recent techniques such as M3 [63] or PageFlip [25] did not focus on eyes-free interaction. By guiding users with cheap and simple tactile aids, SCM provides a compelling means for interacting eyes-free.

3.3 Leveraging Spatial Memory

Spatially consistent interfaces leverage user spatial memory skills to facilitate command learning [32, 45, 46]. For example, displaying commands in a grid showed benefits for learning [24, 33]. This approach relies mostly on spatial landmarks on which users can rely [20, 52]. We build on previous work with SCM by disposing commands inside nine cells on a smartwatch [33] to facilitate user learning on the long-term.

3.4 Crossing Interfaces

Crossing allows selecting a command by entering its area [1, 3, 43, 47]. This enables selecting several items in a single gesture, or selecting an item and controlling its parameters as demonstrated by FlowMenus [23] and Control menus [40]. In the context of eyes-free interaction, these continuous movements can be guided by providing active feedback, e.g., sounds [18, 37] or vibrations [2, 9, 42].

3.5 Bezels and Tactile Aids

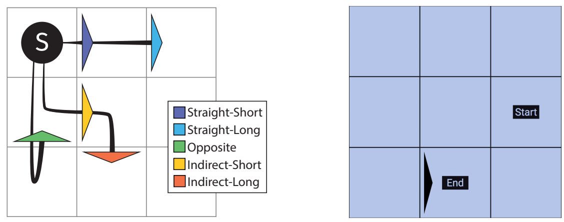
Bezels [27, 29, 43, 49, 57] and other tactile aids provide a way to improve accuracy [55] or to interact eyes-free [10, 60], e.g., by taking advantage of a watchband [39, 60]. Tactile aids can augment the back [16] or the front of the device [10, 13, 22] to alleviate the lack of visual information. However, either these techniques were designed for relatively large surfaces (e.g., a mobile phone [27], tablet [49], or trackpad [19]) or they only support a limited number of commands [27, 29]. As an exception, Blasko and Feiner's technique [10] allows performing a large number of commands on a smartwatch, but at the price of multi-segmented strokes and multi-level menus and its efficiency has not been evaluated. While EdgeWrite [55] leverages extruded bezels to support strokes from corners to corners of a small interactive area, its design does not support large gesture sets. Moreover, it does not offer as much flexibility for categorizing commands as SCM or MMM and it requires adding a permanent template on the interactive area to help the finger moving appropriately. As such a modification might conflict with common interactions, we opted to use MMM as a baseline in the user study reported below.

3.6 Menu Layout

Compared to most previous techniques, SCM provides much freedom for users and designers for laying out menus. For instance, conventional Marking menus leverage deep and narrow hierarchies, which highly impacts the organization of menus and makes it difficult to create groups of commands and highlight their relations. While SCM might be difficult to navigate at first because of visual search, it better highlights semantic relations between commands and should leverage spatial memory on the long-term. These different types of techniques are thus complementary, as one or the other may be more adapted depending on the context (e.g., the number of commands and their semantic relations).

4 EFFECTIVENESS OF SCM

To validate the design of SCM, we performed a first experiment evaluating whether users can perform SCM gestures accurately. We evaluated the usability of the technique in *expert mode*, i.e., eyes-free. This was a mandatory first evaluation as such a technique would be of little interest without providing an efficient expert mode.



(a) (b)

Fig. 4. (a) The five gesture Types evaluated in the experiment (for a given starting area "S"). (b) Visual stimulus indicating the gesture to perform to complete a trial.

Participants. We recruited 12 right-handed participants in our local universities (3 women, 9 men), aged from 21 to 38 (mean = 27, median = 26). Three of them already used a smartwatch, two on a daily basis.

Apparatus. We used an Android Polar M600 smartwatch with a touch-screen of 23.3×23.3 mm and a resolution of 240×240 pixels (Figure 3a). The size of the bezels was 6.5 mm, except for the top bezel (15.5 mm). We connected the smartwatch through WiFi to a node.js [51] server running on a 15" Macbook. We used an Android program running on the smartwatch to capture SCM gestures. This program was interacting remotely with the server for storing data and controlling the experiment.

Context. The participants wore the smartwatch on their left wrist and interacted with their right index finger. They sat at a desk in front of the laptop displaying the visual stimuli. We used an external display for showing the stimuli, because SCM is designed to mainly control "external" environments. To ensure the smartwatch was out of their sight, they had to lay their wrist on their knee below the desk. We made sure they felt comfortable before starting the experiment.

Experimental Design. We distinguish five representative gesture Types that one can perform with SCM (Figure 4a). *Straight* gestures that consist in straight strokes, *Opposite* gestures that consist in entering the ending cell from the side opposite of the starting cell, and *Indirect* gestures that includes all others. Since the distance between cells for *Straight* and *Indirect* gestures can vary, we differentiate *Short* and *Long* movements (Figure 4a).

We chose six starting cells (StartCell), i.e., menu cells, three starting from a corner (top-left, top-right, bottom-right) and three starting from the middle of a side (middle-left, bottom-middle, middle-right), and randomly picked five gestures for each of them (one for each Type) to compose a representative set of gestures as in [31, 61, 63]. We hence obtained a total of 30 (Type × StartCell) representative gestures of the 172 possible SCM gestures.

The participant started a trial by touching the touchscreen of the smartwatch. A "Start" and "End" labels were then displayed to indicate the starting and ending cells of the gesture to perform, and the ending cell comprised an arrowhead indicating how to enter it (Figure 4b). No other information was provided to the participants (e.g., the movement to perform). Moreover, no visual feedback was displayed on the screen, and the participants could not see the smartwatch screen, as explained above. A trial ended once the participant completed a gesture (i.e., finger released outside the starting cell of the movement).

Our purpose in this experiment was to evaluate the user's ability to perform SCM gestures. We thus instructed the participants to be *as accurate as possible* to avoid a confounding factor between accuracy and speed. To ensure that participants would follow this instruction, they had to wait after completing a trial for two seconds if they completed a trial correctly, and for six seconds otherwise. At the end of each trial, they were notified whether they performed the gesture successfully or not, and we depicted the actual trace of their movement for two seconds.

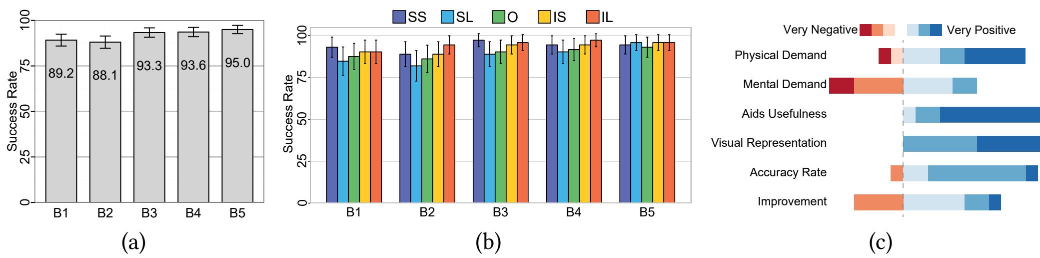


Fig. 5. Results of the first experiment with 95% CIs. Success rates (a) for all gestures, (b) depending on the type of gestures (*Straight-Short* (SS), *Straight-Long* (SL), *Opposite* (O), *Indirect-Short* (IS), *Indirect-Long* (IL)), over the blocks. (c) Results of the questionnaire's Likert-scales.

The experiment consisted of six blocks and for each block the participants had to perform the 30 gestures described above (random order). The first block (B0) was a *training* block. We recorded data over 5 blocks (BlockNbr, B1 to B5), for a total of $5 \times 30 \times 12 = 1800$ gestures. On completing all the blocks, the participants filled a questionnaire consisting of Likert-scales ranging from 1 (*very negative*) to 7 (*very positive*) about: the *physical* and *mental* demand required to perform gestures, whether *tactile aids* helped them, whether *visual representation* of the gestures was easy to understand, their evaluation of their *accuracy* in the last block, and of their *improvement* throughout the whole experiment.

In summary, we used a within-subject design with 3 factors: Type, StartCell, and BlockNbr. The experiment lasted approximately 30 minutes.

4.1 Results

The main result of the experiment is that participants performed SCM gestures accurately from the beginning of the experiment (B1: 89.2%), and that their performance improved until the end (B5: 95.0%), see Figure 5a. We analyzed our data using three-way ANOVA's for the model Type × StartCell × BlockNbr, and post-hoc paired Student t-tests (Holm-Bonferonni corrected) and report Cohen's d effect sizes. We further report empirical results on the participants' *success rate*, *trial completion time*, and *gesture execution time*, i.e., the time spent on the interactive surface to perform the gesture (Shapiro-Wilk tests of normality show no evidence of non-normality for the above measures). We also discuss the results from the questionnaires filled by the participants.

Success rate. The participants performed the gestures accurately from the first block (89.2%) and they improved their performance until the last block (95%) (Figure 5a). This is confirmed by an effect of BlockNbr ($F_{4,44} = 4.28$, $p = 0.005$).

We found an effect of Type ($F_{4,44} = 4.29$, $p = 0.005$) on the success rate, but no interaction with BlockNbr ($F_{16,176} = 0.53$, $p = 0.929$, see Figure 5b). Post-hoc t-tests show a worse overall success rate for *Straight-Long* against *Indirect-Long* ($p < 0.001$, $d = 1.16$, 88.3% vs. 94.7%), and for *Opposite* against *Straight-Short* ($p = 0.036$, $d = 0.50$, 89.7% vs. 93.6%). Note, however, that post-hoc t-tests do not show significant differences for the last block.

We also found an effect of StartCell on the success rate ($F_{5,55} = 8.42$, $p < 0.001$), showing that participants performed worse by starting from the middle-left cell than any other (p's < 0.05). We found no interaction of StartCell with BlockNbr ($F_{20,220} = 1.34$, $p = 0.157$), but an interaction with Type on success rate ($F_{20,220} = 3.20$, $p < 0.001$). Post-hoc t-tests suggest that the worse performance of the the middle-left cell is mainly caused by *Straight-Long* and *Opposite* gestures. Note also that participants have been, overall, significantly more accurate for gesture starting from a corner, than from the middle of a side ($p = 0.023$, $d = 0.72$, 94.1% vs. 89.5%). However, again, these effects disappear for the last block.

Trial Completion and Gesture Execution Times. We found an effect of the block number on the trial completion time ($F_{4,44} = 5.52$, $p < 0.001$). Indeed, the participants improved from 4.7s in the first block to 3.55s in the last block, showing an important effect of training. The average gesture execution time (time spent on the interactive surface) was 1.54s, and we found no effect of the block number ($F_{4,44} = 0.95$, $p = 0.45$). Note that the difference between the trial completion time and the gesture time comprises the time spent to react to the stimuli plus the time spent on the bezel of the device.

Without surprise we found an effect of the gesture Type on both times, shorter gestures being faster than long gestures, and *Opposite* gestures being the slowest gestures. As for the success rate, the participants performed better when the gestures started from a corner ($p = 0.010$, $d = 0.52$, 3.72s vs. 4.38s), but this was not the case for the gesture execution time.

Questionnaire. We depict the results of the questionnaire in Figure 5c. The participants found the *tactile aids* very useful to perform gestures accurately. They reported to use them to guide their movements by sliding on the edges, thus rely on them to initiate a gesture. Interestingly, while participants were positive about their perceived performance in the last block, they did not perceive a great *improvement* throughout the experiment albeit they did improve. This hints that one can easily master SCM gestures. Furthermore, the participants did not seem to have difficulties *understanding* the SCM concept. Finally, participants mentioned a small level of *physical demand* but a relatively high level of *mental demand*, which is not surprising as the technique relies on a different approach than conventional techniques.

Participant Movements. Figure 6 depicts the movement traces of all participants in the last block for the gesture depicted in Figure 4b. We can see that most of them chose to move across the sensitive surface horizontally (crossing its central cell), but three participants preferred to cross the ending cell before entering it from the proper side, and one participant preferred to perform a long movement, using the four edges as tactile aids. This demonstrates that participants use genuinely different strategies to perform the same gestures, even though no instruction was given in this regard.

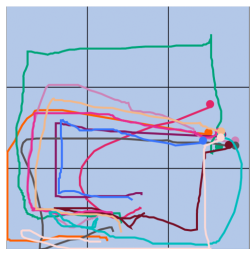


Fig. 6. Each colored line depicts the movement of one user during the last block of the experiment for the gesture depicted in Figure 4b.

5 LEARNABILITY OF SCM AND MARKING GESTURES

This section presents a second experiment aimed at assessing the learnability of SCM gestures. To better understand the capabilities of SCM, we chose to compare it to Multi-stroke Marking Menus (MMM) [62], one of the rare gestural techniques that also enable large gesture sets for eyes-free interaction on small surfaces. As mentioned above, these techniques rely on different hierarchical structures that likely leverage different memorization types (e.g., spatial memory, shape memory). By comparing them, we aim at understanding how easily can users learn both hierarchical structures and whether SCM provides an interesting alternative to the conventional MMM.

Moreover, this experiment investigates whether users can use the SCM novice mode, and it provides data on the transition from novice to expert users while using SCM. Besides, it provides empirical data on the learning of marking gestures, which remain mostly evaluated on their mechanical benefits, but rarely on their learning benefits [6, 20].

Participants. We recruited 12 new participants including two left-handed in our local universities (7 women, 5 men), aged from 23 to 32 (mean=27.8, median=28). Five of them already used a smartwatch, one on a daily basis.

Apparatus and Context. We used the same setup as in the first experiment, but this time recorded the participants' movements on the bezels using a camera. To do so while keeping the smartwatch out of the participants' sight, we set up a cardboard screen behind which participants could lay their hands to interact.

Task and Stimulus. The participants had to select commands in a menu hierarchy. Commands were textual items (e.g., "monkey"), and the visual stimuli were the names of the commands to select. We displayed the stimulus on the laptop screen as soon as a trial started (i.e., when the user touched the screen of the smartwatch). On completing a trial, we informed participants of their success or failure by providing the actual name of the command they selected in a respectively blue or red square.

Interaction design. We implemented MMM as described in [62]. On selecting a submenu, the items it comprises are revealed with an offset, and the items of the previous level fade out during one second. To return to a previous level in case of a wrong selection, if the user initiated a stroke, she can slide back to the starting position; otherwise, she can perform a quick tap. Lastly, if the user does not interact for more than 2s, all selections are canceled and the novice mode is deactivated.

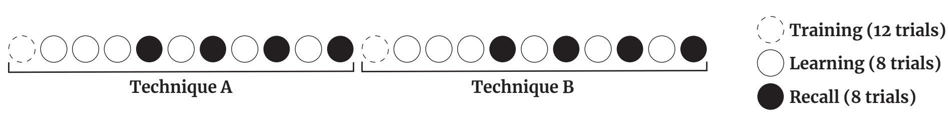


Fig. 7. Experimental design that consists of one *training*, six *learning*, and four *recall* blocks for each technique.

Experimental Design. We follow a within-subject design with primary factor TECH, half of the participants starting with MMM and the other half with SCM. For each technique, the experiment consisted of *learning* and *recall* blocks (Figure 7). The participants always started a trial in the *expert* mode (i.e., commands are hidden and no visual feedback). During *learning* blocks, however, they could trigger the novice mode with a long press of 300ms to reveal the commands and display their movement trace on the laptop screen (Figure 8). Before starting the series of *learning* and *recall* blocks, the participants had to complete a *training* block of twelve trials. We used abstract command names for this specific block (e.g., "c7_10" for the tenth command of menu 7) to avoid possible learning effects.

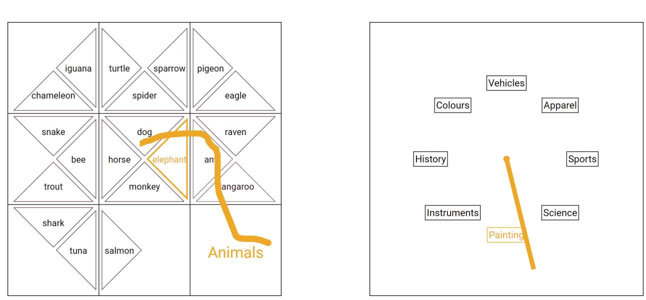


Fig. 8. Novice modes of the SCM (left) and MMM (right) techniques used in the experiment. The orange line depicts the user movement trace; a straight line joining the starting and ending point of the movement in the case of MMM.

Command sets: We built *two* sets of commands consisting of *eight* menus each. Both sets consisted of four menus containing commands to select and four menus serving as distractors. We counterbalanced these sets between participants for each technique. We filled each of these eight menus with *twenty* commands; they comprised four submenus comprising four commands (16), and four *independent* commands. The submenus contained unrelated items, e.g., the menu *Animals* contained four submenus named *fishes*, *birds*, *reptiles*, and *mammals*.

Menu Hierarchies: The MMM hierarchy consisted of 8 menus at the first level; 4 submenus (placed on the off-axes) and 4 independent commands at the second level; and 4 commands per submenu at the third level. We chose this way of organizing commands because the two techniques can support almost the same amount of commands (172 for SCM and 160 for MMM), and this organization provides a neat hierarchy for MMM that would be disadvantaged otherwise. The SCM hierarchy consisted of submenus next to spatial landmarks like the corners or the center of the surface to help categorizing them spatially. We placed the independent commands on the remaining free slots.

Set of Targets: We asked the participants to learn *two commands* per menu (overall eight commands per technique). We chose the commands by considering their "distance" from the menu root, either *close* or *far* (factor DISTANCE). This "distance" is either related to the cells (for SCM, we use the same criterion as in the first experiment), or to the menu depth (for MMM). For SCM, *close* commands correspond to *straight-* or *indirect-close* gestures, and *far* commands to *indirect-far* gestures. For MMM, *close* commands correspond to *second-level* selections, and *far* commands to *third-level* selections. We placed the commands randomly and chose the positions of the target commands randomly.

In summary, we used a within-subject design with 3 factors: TECH, BLOCKNBR (recall or learning blocks), and DISTANCE. On completing the experiment, we asked the participants to fill the same questionnaire as in the first experiment.

5.1 Results

We report in the following on the *success rate*, *recall rate*, *learning curves*, *trial completion time*, *bezel interaction time* (time spent on the bezels for SCM), and *gesture execution time* (time spent on the interactive surface). We use three-way ANOVA's with the model TECH × BLOCKNBR × DISTANCE (the data does not exhibit a strong departure from normality). We also discuss the results from the questionnaires filled by the participants.

The primary outcome of this experiment is that we did not find significant differences between the two techniques (neither significant TECH × DISTANCE or TECH × BLOCKNBR interactions) for the success rate, recall rate, or task completion time.

Success and Recall Rate. The only difference between the *success* and *recall* rates is that the latter includes the gestures that the participant remembered properly but performed inaccurately. To assess the *recall* rate, we asked the participants, on trial failures, what movement they intended to perform.

Figure 9a shows the evolution of the success rate for each technique. As stated above we found no significant effect of TECH on the success rate ($F_{1,11} = 0.002, p = 0.934$) and the recall rate ($F_{1,11} = 0.04$, $p = 0.834$), and no significant interactions TECH × BLOCKNBR ($F_{3,33} = 0.69, p = 0.564$ and $F_{3,33} = 1.26$, $p = 0.304$) and TECH × DISTANCE ($F_{1,11} = 1.84, p = 0.203$ and $F_{1,11} = 0.71, p = 0.417$).

As expected, we have a significant effect of BLOCKNBR on both success rate ($F_{3,33} = 22.9, p < 0.001$, Figure 9a) and recall rate ($F_{3,33} = 41.2, p < 0.001$). Indeed, the success rate increased between recall blocks, from 52.1% for the first recall block to 79.2% for the last and the recall rate from 57.8% for the first block to 85.4% for the last.

We also have a significant effect of DISTANCE on both success rate ($F_{1,11} = 17.85, p < 0.001$) and recall rate ($F_{1,11} = 12.91, p = 0.004$). The participants selected *Close* commands more successfully

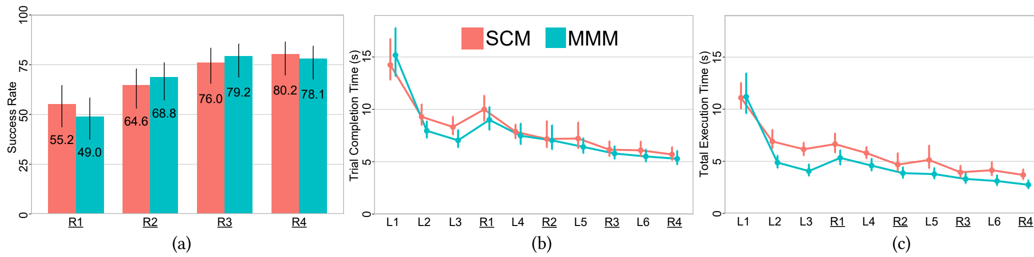


Fig. 9. Main results of the second experiment with 95% bootstrap CIs. (a) Success rate of trials during recall blocks. Evolution of the (b) trial completion time and (c) total execution time (time spent on the interactive surface, and the bezels using SCM) throughout the experiment (in seconds).

(76.7% vs. 61.1%) and recalled them more efficiently (81.8% vs. 67.7%) than *Far* commands. Thus, *deeper* or *farther* commands seem more difficult to learn using both techniques.

Learning Curves. To evaluate the evolution of the user expertise, we consider the rate of trials completed in *expert* mode during the learning blocks. With this approach, we can observe that the participants relied less and less on the novice mode throughout the experiment to reach 22.9% in the sixth learning block ($F_{5,55} = 4.9$, $p < 0.001$), and that the participants seemed to rely less on the novice mode using SCM than MMM (overall 5.3% vs. 9.2%, $F_{1,11} = 4.32$, $p = 0.06$).

Bezel Interaction Time. Using video analyses, we assessed how long participants touched the tactile cues on the bezel of the device while using SCM. The participants relied less on these passive tactile cues during the *learning* blocks than the *recall* blocks ($p < 0.001$, 0.67 vs. 1.35s), presumably because they could see the trace of the movement in the novice mode. For the *recall* phases, we found no significant effect of BlockNbr or distance on the bezel interaction time (p's > 0.183).

Gesture and Execution Time. We found no significant effect of the techniques on the gesture execution time (4.94 vs. 4.68s on average for all blocks, $F_{1,11} = 0.80$, $p = 0.39$). However, when taking into account both bezel interaction and gesture execution time, the *total execution time* is slightly longer for SCM ($F_{1,11} = 25.40$, $p < 0.001$, Figure 9c, no interaction with BlockNbr or distance).

Trial Total Completion Time. We found no significant effect of the techniques on the *trial total completion time*: an ANOVA revealed no effect of tech ($F_{1,11} = 1.31$, $p = 0.276$, Figure 9b), nor an interaction with BlockNbr and distance. We explain this result by the fact that the participants recalled gestures faster using SCM [45], although they needed some extra time to place their finger correctly on the bezel for initiating a gesture.

Unsurprisingly, the participants selected commands faster throughout the experiment, and faster with *Close* commands, so that there is an effect of BlockNbr ($F_{9,99} = 35.3$, $p < 0.001$, Figure 9b) and of distance ($F_{1,11} = 48.6$, $p < 0.001$).

Questionnaire. The analysis of the various seven-level Likert-scales did not yield significant differences between the techniques. We use Wilcoxon's paired t-tests in the following and report the means of the Likert-items (1 is worst, 7 is best): *physical demand* ($p = 0.26$, mean=5.13), *mental demand* ($p = 0.44$, 4.17), *improvement* ($p = 0.38$, 4.87). Again, the participants found that vibrations and tactile cues were very helpful when interacting with SCM (respective means 6.25 and 5.25).

6 DISCUSSION

Our study results suggest that one can efficiently select multiple commands using SCM, although it supports an unusually large set of gestures. One of our concerns was that SCM could be more difficult to master than techniques that rely on very simple gestures, such as Marking menus. Nevertheless, the second experiment did not yield significant differences when comparing both techniques. The experiments also showed that users can quickly master SCM gestures, perform these gestures eyes-free (first experiment), and learn them efficiently (second experiment). In the following, we first discuss the effectiveness of SCM and then its learnability and efficiency in compared to MMM. We finally present design guidelines for implementing SCM gestures.

6.1 SCM effectiveness

The results of the first experiment show that users could perform all types of SCM gestures accurately eyes-free. We observed a success rate of 95% after 20 min of practice, and since the beginning, the level of accuracy was rather high for a gestural technique (89.2%). This suggests that the technique is not hard to master, at least in a lab context, considering that participants experienced the technique for the first time and that 75% of them never used a touch-sensitive smartwatch in the past. Moreover, we observed only small differences between the various types of SCM gestures. As these types represent SCM gestures, this also suggests that we would likely observe similar results for all the others.

We observed rather long SCM's trial completion times in the last blocks of both experiments (3.55s in the first, 5.69s in the second). These times include the reaction time (i.e., recall time too), bezel interaction time, and the time to perform the gesture on the touchscreen. Two main factors can explain these extended times: we focused our evaluation on the user accuracy hence did not constrain the trial time, and we tested the techniques in the most challenging case, i.e., by saturating the gesture set and their visual output. This exacerbated disadvantages of the two techniques: the visual search required by SCM, and the lack of a pre-visualization feature in the MMM design [5]. We hypothesize more experienced users would produce shorter completion times with smaller gesture sets.

In both experiments, the participants leveraged the path flexibility of the SCM design to perform SCM gestures using various paths (Figure 6). Some of them remarked they preferred to rely on the edges while others were more confident using vibrations. This ability to choose a specific path is likely to reduce selection errors because users can choose a "safer" path if they are not fully confident when interacting eyes-free [11].

6.2 Learnability and Efficiency

The results of our second experiment demonstrate that the participants learned the SCM and MMM gestures efficiently. Although SCM gestures may seem more complicated than marking gestures, our results do not provide evidence in this regard. This is promising as it indicates that SCM is a compelling alternative to MMM. Furthermore, these two techniques are complementary as SCM offers broad and shallow menus, which provides more freedom to designers for organizing menu items, whereas MMM are better adapted for creating numerous menus containing a small number of items (hardly more than 8 [31], and even less on the borders (5) and corners (3) of the screen). SCM provides an advantage in this regard, as many applications comprise menus with a relatively large number of items [5].

Moreover, SCM enables creating meaningful spatial layouts (e.g., placing related commands in the same corner), thus creating subgroups in the same menu. These subgroups are immediately visible and do not require opening submenus to visualize their elements, which contrasts with techniques that only support small menus, such as MMM. This was, for example, pointed out by a participant in the second experiment: "[SCM is] easier to visualize" (P10). Furthermore, this feature likely leverages spatial memory skills of the users on the long-term [33, 45, 46].

In addition, SCM provides the opportunity to drawing *symbolic* or *metaphorical* gestures [28] that might take longer to perform but are likely easier to memorize than conventional directional marks. Although we did not evaluate this opportunity, this feature of SCM is worth mentioning as users do not necessarily favor speed. For instance, when selecting infrequent commands that are difficult to undo (e.g., turning on a device), the ability to recall the correct gesture is more important than its duration. Similarly, performing a slower but more reliable gesture is likely more appropriate when selecting critical commands (e.g., quitting an application).

Enhancing user freedom also enables users to change their behavior based on their expertise. While novice users may favor accuracy, expert users may favor speed and thus change the way they perform the gestures. The user behavior may also depend on their understanding of the SCM design, their physical abilities, their level of experience, or the effect of the command (e.g., slower gestures for critical commands that should not be executed by mistake).

At last, we did not observe differences in the trial completion time between the techniques. While using passive tactile cues on the bezels helps improving accuracy, this does not seem to hinder the command selection process. As suggested by prior work (and observed in our pilot studies), using the technique eyes-free without assistance can be challenging. These results suggest that using cheap and simple tactile cues provides a simple means to ensure accurate eyes-free selection [19].

6.3 SCM Design guidelines

Based on the study results and our observations of the participants' behaviors, we propose several guidelines for designing SCM gestures. These guidelines mostly concern the types of SCM gestures that one should favor when mapping the most frequent actions.

(1) *Place important menus in the corner cells as they are easier to locate and, thus, provide faster and more accurate interaction.* We observed in both experiments that the participants selected commands significantly faster (first experiment 3.72s vs. 4.39s, second experiment 7.31 vs. 9.10s) and significantly more accurately (94.1% vs. 89.5%, 89.8% vs. 80.4%) when starting gestures from the corners.

(2) *Leverage straight gestures for frequent actions as they are faster to perform than other SCM gestures.* We observed in both experiments that *straight* gestures are significantly faster to perform (first experiment 3.31s vs 4.54s, second 5.64s vs. 8.91s) than other SCM gestures.

(3) *Provide to users examples of various paths for performing unique SCM gestures to stimulate their creativity.* We observed in our experiments that some users adapted their strategies based on their expertise, while others did not. It is unclear whether they did this on purpose, i.e., if they realized they could use other strategies. To ensure that users grasp the freedom provided by SCM, we recommend making this explicit as soon as possible, for instance, by demonstrating how one can perform the same gesture using different paths when users experience the technique for the first time.

6.4 Limitations

Interaction Contexts. Although SCM is a gestural interaction technique designed for small mobile devices, we did not study its efficiency in challenging contexts like walking outside or running. The experiment results, however, are still valuable as we envision users using SCM while being seated on a couch in front of a TV, standing in front of a wall display, or standing still to interact with an AR setup. While further research is needed to address the efficiency of SCM and MMM in challenging scenarios, previous work showed the advantage of using tactile aids and bezel gestures in such scenarios [11, 49] suggesting the SCM design is adequate for these scenarios.

Menu Hierarchies. The second experiment required a menu hierarchy that would not disadvantage MMM. For this purpose we chose a simple and coherent hierarchy, with similar features as SCM and conventional spring menus [7]: The first level consisted of menu roots, the second of commands or sub-menus, and the third only contained commands. Although we cannot say this hierarchy is optimal, it seemed to be the fairest in comparison with SCM.

(a) (b)

Fig. 10. SCM adapts to various device shapes. (a) Examples of small round touch-sensitive surfaces integrated in a smartwatch and a VR controller (highlighted in orange). (b) Examples of SCM layouts using hexagonal cells better suited for round surfaces.

7 EXTENSIONS OF SCM

Alternate Shapes and Devices. Not all devices have a rectangular shape. For examples, some smartwatches or devices for interacting with AR and VR systems have round surfaces (Figure 10a). In such cases, rectangular cells do not fit the form factor of the device [50]. SCM does not rely solely on rectangular cells and adapts to circular devices as illustrated in Figure 10b. One could for instance use polygonal cells to provide more sides, and therefore more commands per cell, which allows reducing the number of cells without impeding the input capabilities.

Scaling to larger surfaces. Despite we presented SCM as a solution to enable large gesture sets on *small surfaces*, it also adapts to *larger* surfaces (e.g., tablets and smartphones). For instance, extending M3 [63] and Bezel-Tap [49], one could dedicate a *small area* of the screen, typically close to the user's hand, to enable quick shortcuts using the thumb. In such a case, using a *sensitive bezel* would provide both passive feedback to guide the user and an activation mechanism to avoid conflicting with common interactions.

Another option consists in using a grid of 4×4 cells in order to provide more menus and more commands per menu. Larger surfaces also enable using cells of various sizes (Figure 11). Using this feature, one could emphasize the most important commands of a menu. Furthermore, the cell sizes can adapt dynamically to user actions (e.g., opening a menu) to change the layout of the grid while preserving spatial landmarks.

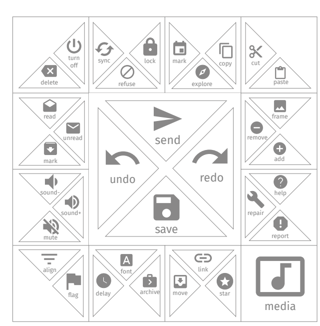


Fig. 11. Menu layout designed for *larger* surfaces. This layout uses a 4×4 grid and various cell sizes (the four middle cells are merge to produce one larger cell) to emphasize frequent commands.

Continuous control. Using the same principle than FlowMenu [23] or Control Menus [40], SCM could enable continuous control by relying on a simple dwell mechanism. By dwelling on a cell (e.g., as proposed in M3 menus [63]), users can select a specific command and continue their movement to control the value of a variable without lifting the finger from the surface. This feature, which leverages the fact that SCM relies on position-dependent gestures, would not be compatible with size-independent marking gestures.

8 CONCLUSION

We presented Side-Crossing Menus, a new gestural interaction technique enhancing small surfaces' input capabilities with large gesture sets. One can use SCM eyes-free by using passive (stickers) and active (vibrations) tactile aids. SCM provides a solution for interacting with various types of devices in various contexts. It also enhances user freedom by allowing them to use various paths to perform a given gesture.

We presented a user study evaluating the efficiency of the SCM concept. A first experiment focused on the effectiveness of SCM showed that novice users found the gestures intuitive and quickly reached a high level of accuracy. A second experiment focusing on gesture learning showed no significant difference when comparing SCM to Multi-stroke Marking Menus (MMM), an established baseline. The results of these two experiments suggest that SCM is adequate for performing large sets of gestures on small surfaces and that it is complementary to Multi-stroke Marking menus and adapts to more diverse scenarios.

Future work should focus on evaluating SCM in the context of circular and large surfaces to assess whether our results can be generalized to these surfaces. Also, our study focused on interacting with the index finger, whereas VR controllers might rather use the thumb as a primary interacting finger (see Figure 10). Thus, further studies are required using the thumb. At last, to ensure SCM fits perfectly mobile contexts prone to jitters (e.g., when walking or running), further analyses are needed within these challenging contexts.

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).

We wish to thank the participants of our experiment. We also wish to thank the members of the Saarland HCI Lab, of the LTCI DIVA team, and of the Inria-LRI ILDA team for their feedback.

REFERENCES

[1] Johnny Accot and Shumin Zhai. 2002. More Than Dotting the I's — Foundations for Crossing-based Interfaces. In **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)**. ACM, 73–80. https://doi.org/10.1145/503376.503390

[2] Michel Amberg, Frédéric Giraud, Betty Semail, Paolo Olivo, Géry Casiez, and Nicolas Roussel. 2011. STIMTAC: A Tactile Input Device with Programmable Friction. In **Proceedings of the 24th Annual ACM Symposium Adjunct on User Interface Software and Technology (UIST '11 Adjunct)**. ACM, 7–8. https://doi.org/10.1145/2046396.2046401

[3] 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. https://doi.org/10.1145/1029632.1029635

[4] 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. https://doi.org/10.1145/1518701.1519052

[5] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2007. Wave Menus: Improving the Novice Mode of Hierarchical Marking Menus. In **Human-Computer Interaction – INTERACT 2007**. Springer, 475–488. https://doi.org/10.1007/978-3-540-74796-3_45

[6] 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. https://doi.org/10.1145/1385569.1385575

[7] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2016. Visual Menu Techniques. **ACM Comput. Surv.** 49, 4, Article 60 (Dec. 2016), 41 pages. https://doi.org/10.1145/3002171

[8] Robert Ball, Chris North, and Doug A. Bowman. 2007. Move to improve: promoting physical navigation to increase user performance with large displays. In **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)**. ACM, 191–200. https://doi.org/10.1145/1240624.1240656

[9] 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. https://doi.org/10.1145/1866029.1866074

[10] Gabor Blasko and Steven Feiner. 2004. An Interaction System for Watch Computers Using Tactile Guidance and Bidirectional Segmented Strokes. In **Proceedings of the Eighth International Symposium on Wearable Computers (ISWC '04)**. IEEE, 120–123. https://doi.org/10.1109/ISWC.2004.6

[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. https://doi.org/10.1145/1978942.1979000

[12] William A. S. Buxton. 1995. Chunking and Phrasing and the Design of Human-Computer Dialogues. In **Human-Computer Interaction: Toward the Year 2000**. Morgan Kaufmann, San Francisco, CA, USA, 494–499.

[13] Maria Claudia Buzzi, Marina Buzzi, Francesco Donini, Barbara Leporini, and Maria Teresa Paratore. 2013. Haptic Reference Cues to Support the Exploration of Touchscreen Mobile Devices by Blind Users. In **Proceedings of the Biannual Conference of the Italian Chapter of SIGCHI (CHItaly '13)**. ACM, Article 28, 8 pages. https://doi.org/10.1145/2499149.2499156

[14] Tom Chandler, Maxime Cordeil, Tobias Czauderna, Tim Dwyer, Jaroslaw Glowacki, Cagatay Goncu, Matthias, Klapperstueck, Karsten Klein, Kim Marriott, Falk Schreiber, and Elliot Wilson. 2015. Immersive Analytics. In **Big Data Visual Analytics (BDVA '15)**. IEEE, 1–8. https://doi.org/10.1109/BDVA.2015.7314296

[15] Xiang 'Anthony' Chen, Tovi Grossman, and George Fitzmaurice. 2014. Swipeboard: A Text Entry Technique for Ultra-small Interfaces That Supports Novice to Expert Transitions. In **Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)**. ACM, 615–620. https://doi.org/10.1145/2642918.2647354

[16] 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. https://doi.org/10.1145/2702123.2702277

[17] Raimund Dachselt and Anett Hübner. 2007. Three-dimensional Menus: A Survey and Taxonomy. **Comput. Graph.** 31, 1 (Jan. 2007), 53–65. https://doi.org/10.1016/j.cag.2006.09.006

[18] Jules Françoise, Olivier Chapuis, Sylvain Hanneton, and Frédéric Bevilacqua. 2016. SoundGuides: Adapting Continuous Auditory Feedback to Users. In **Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)**. ACM, 2829–2836. https://doi.org/10.1145/2851581.2892420

[19] Bruno Fruchard, Eric Lecolinet, and Olivier Chapuis. 2017. MarkPad: Augmenting Touchpads for Command Selection. In **Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)**. ACM, 5630–5642. https://doi.org/10.1145/3025453.3025486

[20] Bruno Fruchard, Eric Lecolinet, and Olivier Chapuis. 2018. How Memorizing Positions or Directions Affects Gesture Learning?. In **Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces (ISS '18)**. ACM, 107–114. https://doi.org/10.1145/3279778.3279787

[21] Bruno Fruchard, Eric Lecolinet, and Olivier Chapuis. 2018. Impact of Semantic Aids on Command Memorization for On-body Interaction and Directional Gestures. In **Proceedings of the 2018 International Conference on Advanced Visual Interfaces (AVI '18)**. ACM, Article 14, 9 pages. https://doi.org/10.1145/3206505.3206524

[22] 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. https://doi.org/10.1145/2785830.2785867

[23] François Guimbretiére and Terry Winograd. 2000. FlowMenu: Combining Command, Text, and Data Entry. In **Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology (UIST '00)**. ACM, 213–216. https://doi.org/10.1145/354401.354778

[24] Carl Gutwin, Andy Cockburn, Joey Scarr, Sylvain Malacria, and Scott C. Olson. 2014. Faster Command Selection on Tablets with FastTap. In **Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)**. ACM, 2617–2626. https://doi.org/10.1145/2556288.2557136

[25] Teng Han, Jiannan Li, Khalad Hasan, Keisuke Nakamura, Randy Gomez, Ravin Balakrishnan, and Pourang Irani. 2018. PageFlip: Leveraging Page-Flipping Gestures for Efficient Command and Value Selection on Smartwatches. In **Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)**. ACM, Article 529, 12 pages. https://doi.org/10.1145/3173574.3174103

[26] Tom Horak, Sriram Karthik Badam, Niklas Elmqvist, and Raimund Dachselt. 2018. When David Meets Goliath: Combining Smartwatches with a Large Vertical Display for Visual Data Exploration. In **Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)**. ACM, Article 19, 13 pages. https://doi.org/10.1145/3173574.3173593

[27] 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. https:

//doi.org/10.1145/2207676.2208376

[28] Maria Karam and m. c. schraefel. 2005. A Taxonomy of Gestures in Human Computer Interactions. (2005). https://eprints.soton.ac.uk/261149/ Project Report.

[29] 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. https://doi.org/10.1145/2858036.2858216

[30] Gordon Kurtenbach. 1993. **The Design and Evaluation of Marking Menus**. Ph.D. Dissertation. University of Toronto, Canada. http://autodeskresearch.com/pdf/theses/kurtenbach-phd.pdf

[31] 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

[32] Benjamin Lafreniere, Carl Gutwin, and Andy Cockburn. 2017. Investigating the Post-Training Persistence of Expert Interaction Techniques. **ACM Trans. Comput.-Hum. Interact.** 24, 4, Article 29 (Aug. 2017), 46 pages. https://doi.org/10.1145/3119928

[33] 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. https://doi.org/10.1145/2858036.2858166

[34] Yang Li. 2010. Gesture Search: A Tool for Fast Mobile Data Access. In **Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology (UIST '10)**. ACM, 87–96. https://doi.org/10.1145/1866029.1866044

[35] K. Marriott, F. Schreiber, T. Dwyer, K. Klein, N. Henry Riche, T. Itoh, W. Stuerzlinger, and B.H Thomas (Eds.). 2018. **Immersive Analytics**. LNCS, Vol. 11190. Springer, Cham. https://doi.org/10.1007/978-3-030-01388-2

[36] Mathieu Nancel and Michel Beaudouin-Lafon. 2008. **Extending Marking Menus with Integral Dimensions**. Technical Report 1503. LRI, Univ. Paris-Sud, France.

[37] Uran Oh, Shaun K. Kane, and Leah Findlater. 2013. Follow That Sound: Using Sonification and Corrective Verbal Feedback to Teach Touchscreen Gestures. In **Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '13)**. ACM, Article 13, 8 pages. https://doi.org/10.1145/2513383.2513455

[38] 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. https://doi.org/10.1145/2702123.2702126

[39] Simon T. Perrault, Eric Lecolinet, James Eagan, and Yves Guiard. 2013. Watchit: Simple Gestures and Eyes-free Interaction for Wristwatches and Bracelets. In **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)**. ACM, 1451–1460. https://doi.org/10.1145/2470654.2466192

[40] Stuart Pook, Eric Lecolinet, Guy Vaysseix, and Emmanuel Barillot. 2000. Control Menus: Excecution and Control in a Single Interactor. In **CHI '00 Extended Abstracts on Human Factors in Computing Systems (CHI EA '00)**. ACM, 263–264. https://doi.org/10.1145/633292.633446

[41] Arnaud Prouzeau, Anastasia Bezerianos, and Olivier Chapuis. 2018. Awareness Techniques to Aid Transitions Between Personal and Shared Workspaces in Multi-Display Environments. In **Proceedings of the ACM International Conference on Interactive Surfaces and Spaces (ISS '18)**. ACM, 291–304. https://doi.org/10.1145/3279778.3279780

[42] Yosra Rekik, Radu-Daniel Vatavu, and Laurent Grisoni. 2014. Understanding Users' Perceived Difficulty of Multi-Touch Gesture Articulation. In **Proceedings of the 16th International Conference on Multimodal Interaction (ICMI '14)**. ACM, 232–239. https://doi.org/10.1145/2663204.2663273

[43] 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. https://doi.org/10.1145/1518701.1518933

[44] Anne Roudaut, Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2009. Leaf Menus: Linear Menus with Stroke Shortcuts for Small Handheld Devices. In **Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction (INTERACT '09)**. Springer-Verlag, 616–619. https://doi.org/10.1007/978-3-642-03655-2_69

[45] Joey Scarr, Andy Cockburn, Carl Gutwin, and Andrea Bunt. 2012. Improving Command Selection with CommandMaps. In **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)**. ACM, 257–266. https://doi.org/10.1145/2207676.2207713

[46] 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. https://doi.org/10.1145/2470654.2466430

[47] 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. https://doi.org/10.1145/2858036.2858412

[48] Marcos Serrano, Khalad Hasan, Barrett Ens, Xing-Dong Yang, and Pourang Irani. 2015. Smartwatches + Head-Worn Displays: the "New" Smartphone. In **ACM Workshop on Mobile Collocated Interaction (W28 - CHI 2015)**. ACM, 1–5. http://oatao.univ-toulouse.fr/15288/

[49] 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. https://doi.org/10.1145/2470654.2481421

[50] Marcos Serrano, Anne Roudaut, and Pourang Irani. 2017. Visual Composition of Graphical Elements on Non-Rectangular Displays. In **Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)**. ACM, 4405–4416. https://doi.org/10.1145/3025453.3025677

[51] Stefan Tilkov and Steve Vinoski. 2010. Node. js: Using JavaScript to build high-performance network programs. **IEEE Internet Computing** 14, 6 (2010), 80–83.

[52] Md. Sami Uddin, Carl Gutwin, and Benjamin Lafreniere. 2016. HandMark Menus: Rapid Command Selection and Large Command Sets on Multi-Touch Displays. In **Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)**. ACM, 5836–5848. https://doi.org/10.1145/2858036.2858211

[53] 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. https://doi.org/10.1145/1240624.1240727

[54] Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. 2009. User-defined Gestures for Surface Computing. In **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)**. ACM, 1083–1092. https://doi.org/10.1145/1518701.1518866

[55] Jacob O. Wobbrock, Brad A. Myers, and John A. Kembel. 2003. EdgeWrite: A Stylus-based Text Entry Method Designed for High Accuracy and Stability of Motion. In **Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST '03)**. ACM, 61–70. https://doi.org/10.1145/964696.964703

[56] Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. 2007. Gestures Without Libraries, Toolkits or Training: A $1 Recognizer for User Interface Prototypes. In **Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST '07)**. ACM, 159–168. https://doi.org/10.1145/1294211.1294238

[57] Pui Chung Wong, Kening Zhu, Xing-Dong Yang, and Hongbo Fu. 2020. Exploring Eyes-Free Bezel-Initiated Swipe on Round Smartwatches. In **Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)**. Association for Computing Machinery, 1–11. https://doi.org/10.1145/3313831.3376393

[58] Robert Xiao, Teng Cao, Ning Guo, Jun Zhuo, Yang Zhang, and Chris Harrison. 2018. LumiWatch: On-Arm Projected Graphics and Touch Input. In **Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)**. ACM, Article 95, 11 pages. https://doi.org/10.1145/3173574.3173669

[59] Shumin Zhai, Per Ola Kristensson, Caroline Appert, Tue Andersen, and Xiang Cao. 2012. Foundational Issues in Touch-Surface Stroke Gesture Design: An Integrative Review. **Found. Trends Hum.-Comput. Interact.** 5, 2 (Feb. 2012), 97–205. https://doi.org/10.1561/1100000012

[60] Cheng Zhang, Junrui Yang, Caleb Southern, Thad E. Starner, and Gregory D. Abowd. 2016. WatchOut: Extending Interactions on a Smartwatch with Inertial Sensing. In **Proceedings of the 2016 ACM International Symposium on Wearable Computers (ISWC '16)**. ACM, 136–143. https://doi.org/10.1145/2971763.2971775

[61] 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. https://doi.org/10.1145/1124772.1124933

[62] Shengdong Zhao and Ravin Balakrishnan. 2004. Simple vs. Compound Mark Hierarchical Marking Menus. In **Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST '04)**. ACM, 33–42. https://doi.org/10.1145/1029632.1029639

[63] Jingjie Zheng, Xiaojun Bi, Kun Li, Yang Li, and Shumin Zhai. 2018. M3 Gesture Menu: Design and Experimental Analyses of Marking Menus for Touchscreen Mobile Interaction. In **Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)**. ACM, Article 249, 14 pages. https://doi.org/10.1145/3173574.3173823

Received June 2020; revised August 2020; accepted September 2020