Wavelet Menus on Handheld Devices: Stacking Metaphor for Novice Mode and Eyes-Free Selection for Expert Mode

Jeremie Francone, Gilles Bailly, Éric Lecolinet, Nadine Mandran, Laurence Nigay

Wavelet Menus on Handheld Devices: Stacking Metaphor for Novice Mode and Eyes-Free Selection for Expert Mode

[1] Jérémie Francone [2] Gilles Bailly [2] Eric Lecolinet [1] Nadine Mandran [1] Laurence Nigay

[1] LIG, CNRS University of Grenoble 1, Grenoble, France

[2] TELECOM ParisTech, CNRS LTCI, Paris, France

{jeremie.francone; nadine.mandran; laurence.nigay}@imag.fr

{gilles.bailly; eric.lecolinet}@telecom-paristech.fr

ABSTRACT

This paper presents the design and evaluation of the Wavelet menu and its implementation on the iPhone. The Wavelet menu consists of a concentric hierarchical Marking menu using simple gestures. The novice mode, i.e. when the menu is displayed, is well adapted to the limited screen space of handheld devices because the representation of the menu hierarchy is inverted, the deeper submenu being always displayed at the center of the screen. The visual design is based on a stacking metaphor to reinforce the perception of the hierarchy and to help users to quickly understand how the technique works. The menu also supports submenu previsualization, a key property to navigate efficiently in a hierarchy of commands. The quantitative evaluation shows that the Wavelet menu provides an intuitive way for supporting efficient gesture-based navigation. The expert mode, i.e. gesture without waiting for the menu to pop-up, is another key property of the Wavelet menu: By providing stroke shortcuts, the Wavelet favors the selection of frequent commands in expert mode and makes eyes-free selection possible. A user experiment shows that participants are able to select commands, eyes-free, while walking.

Categories and Subject Descriptors

H.5.2 **[Information Interfaces and Presentation]**: User Interfaces – *interaction styles*. I.3.6. [**Computer Graphics**]: Methodology and Techniques – *interaction techniques*.

General Terms

Design, Human Factors.

Keywords

Menu techniques, Handheld devices, Wave menus.

1. INTRODUCTION

Handheld touch-screen devices are raising unprecedented challenges for interaction design [8][13] while providing the opportunity for the users' acceptance of innovative interaction techniques that lie beyond the desktop paradigm [15]. In this context, we focus on command selection, a common task in interactive applications. Linear menus are currently the most common menu technique used on handheld touch-screen devices. However, linear menus are not easy to use on small devices, especially in mobile situations: (i) The limited screen space makes it difficult to display a large hierarchical linear menu while keeping items large enough to be easily selected. Moreover, previsualization, which allows users to explore the content of submenus is hard to implement because of limited screen space. This is a significant drawback because previsualization is a key factor for easing navigation in hierarchical menu systems [2]. (ii) Keyboard shortcuts, a common feature of linear menus, are inconvenient to use in mobile situations [1] since they require full visual attention. Besides they are obviously missing on keyboard-free devices. (iii) In mobile situations users very often prefer to interact with only one hand and with fingers rather than with a stylus (that requires both hands) [8] [13]. Using one hand and direct selection on touch-screen, users use their thumb to interact as with Leaf menus [14]. (iv) Finally, attention is generally divided when using a mobile device and the user may even be unable to look at the screen. Hence, eyes-free selection is a key property for mobile devices as shown by the earPod [18].

In this paper we address the above challenges of handheld touch-screen devices, by studying both the novice mode (visual and interaction design) and the expert mode (stroke gestures) of the Wavelet menu. The Wavelet menu consists of an inverted concentric hierarchical Marking menu based on simple gestures. The key properties of the Wavelet menu for handheld devices are the following ones:

- In novice mode, Wavelet menus provide: (1) Efficient screen space management, the user being able to interact with large hierarchies (using a circular and linear layout) even when there is not enough space to show all parent menus; (2) Submenu previsualization, a feature that enables users to efficiently explore the content of the menus; (3) A stacking metaphor that helps users to quickly understand how the technique works by naturally suggesting gestural interaction.
- In expert mode, the Wavelet menu provides eyes-free selection of commands by performing simple inflection-free and scale invariant gestures by using the thumb.

The Wavelet menu has been developed on the iPhone as an advanced product and demonstrated during two conferences [5] [6] allowing us to collect informal feedback of its usage. In this paper, we rationalize its design, describe its implementation and experimentally measure the benefits of the Wavelet menu for handheld devices, both for novice and expert modes. To do so, we first review related work on menu techniques for handheld devices. We then present the design of the Wavelet menu and its implementation. We finally describe two experiments and their results, the first one focusing on the novice mode while the second one is dedicated to the expert mode.

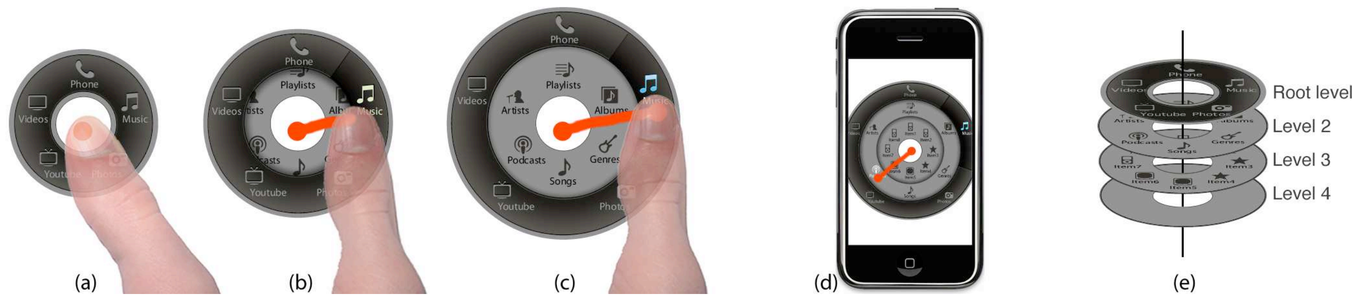

Figure 1. Wavelet menu: (a) The Wavelet menu appears at the center of the screen. (b) Drawing a mark in the direction of the desired item progressively enlarges the root ring, (c) until the submenu is completely displayed. (d) Another mark can be drawn to penetrate deeper into the menu hierarchy. (e) Stacking metaphor.

2. RELATED WORK

Several studies on menu techniques [3] have inspired our work. We review them considering the two interaction modes of a menu: the novice and the expert modes. On the one hand, if the user waits for about 300 ms, the menu enters into novice mode and is displayed on screen. In this mode, the user interacts with the displayed items of the menu. On the other hand, the expert mode is activated when experienced users do not wait for the menu to be displayed and use shortcuts to quickly activate known commands.

On desktop platforms, linear menus facilitate navigation by providing previsualization, a proactive feedback making it possible to avoid many unnecessary up and down transitions in the menu hierarchy [2][12], as well as allowing keyboard shortcuts. But linear menus are not adequate mobile interaction techniques on handheld touch-screen devices as explained in the introduction. Due to the small screen space and the lack of physical buttons, linear menus on handheld devices are limited. In addition, pointing at small items with a finger may be challenging.

Facing the identified problems of linear menus, menu techniques with non-linear layout have been specifically designed for handheld devices. ThumbMenu and ArchMenu [7] consist of semicircular menus and an offset cursor to avoid occlusion of the thumb. But they can display only a limited number of items and they have no expert mode. The RollMark menu [13] uses a circular layout combined with roll movements of the thumb, a new kind of gesture that can be unambiguously recognized on a touch-screen. This menu contains only 6 items and is not hierarchical. Earpod [18] is a circular menu that features eyes-free selection with audio feedback. The item names are played when they are selected, and the user can shift its finger over every item to quickly hear the beginning of every name. However, this technique does not provide previsualization.

Interestingly, only one menu technique for handheld devices extends linear menus, namely the Leaf menu [14]. This technique consists of a hierarchical linear menu augmented with curved gestures: each item is assigned to a stroke that can be performed in expert mode when the menu is not displayed and therefore when the user does not look at the screen. The Leaf menu does not completely solve the problem of limited screen space but defines a promising solution for the expert mode, which is based on gestures that can be performed eyes-free in mobile situations.

Although the expert mode is crucial on handheld devices to support mobile situations, only few menu techniques (Earpod [18] and Leaf menu [14]) have been designed to address this issue. On desktop platforms, a large body of work has been developed during the last 15 years to improve the expert mode of menu techniques. This work could be transposed to the case of handheld devices. The most noticeable work for the case of handheld devices is probably the Marking menus [9], which are circular menus based on gestural interaction. Their key property is that they provide a seamless transition from novice to expert modes because the user performs the same gesture in both modes [10]. By frequently using the menu in novice mode, the user implicitly learns the expert mode. For the case of hierarchical Marking menus [10], the marks are spatially composed, but they require much space in novice mode [17]. In this way a three-level Marking menu requires more horizontal space than 10 linear menus to display its leftmost and rightmost branches. This may not be an important problem on desktop platforms with large screens but this is a major restriction on handheld devices. An alternate design of the Marking menu is the Multi-Stroke menu [17]. It is based on temporal composition of simple inflexion-free marks. A Multi-Stroke menu requires less screen space than a classical hierarchical Marking menu since a submenu is displayed on top of its parent menu. Unfortunately superposition makes previsualization impossible. To solve this problem and allow previsualization while minimizing the required screen space, the Wave menu [2] extends the Multi-stroke menu and improves its novice mode with an inverted representation of the hierarchy and a concentric layout that takes less physical space and focuses the users' attention at the center of the menu. Wave menus have been experimentally evaluated on desktop platforms: They offer the best performance for both novice and expert modes in comparison with existing multi-level Marking menus, while requiring less screen space than they do. This makes the Wave menu a very good candidate for handheld devices.

3. WAVELET MENU DESIGN

The Wavelet menu is an extension of the Wave menu [2] for handheld devices (Table 1). In novice mode, the root menu is displayed at the center of the screen as a ring (Figure 1-(a)). To select an item, the user makes a stroke from the central area towards the desired item. During the interaction, the ring is enlarged, directly controlled by the finger and the submenu is revealed as if it was hidden "below" its parent (Figure 1-(b)). If the user releases its finger while interacting, the ring reverts back to its initial size. Otherwise, once the ring is enlarged enough to completely display the submenu, the user can then release the finger from the screen, and the submenu remains open (Figure 1-(c)). The same effect occurs again if the user makes another stroke from the central area in the direction of an item of the submenu: both the root menu and the submenu move outwards from the center and a third level menu is displayed as if it was hidden below its parent (Figure 1-(d)). This **stacking metaphor** (Figure

1-(e)) is a major improvement over Wave menus because it implicitly suggests gestural interaction in a natural way. Since the submenu progressively appears, users fully control the enlargement effect. From a user's perspective, s/he does not "draw a stroke" (as with the Wave menu), but "handles a stack". Moreover, this metaphor reinforces the visual perception of the hierarchy. Hierarchy was noticed as difficult to perceive in former experiments with the Wave menu. The stacking metaphor may thus help users to learn and understand the working principle of this new technique, unconventionally based on an **inverted concentric layout**. This inverted concentric layout fits very well with the limited screen real estate of handheld devices. Indeed, submenus are displayed on the innermost rings while parent menus are displayed on the outmost rings. So even if the parent menu cannot be displayed because of screen limitation, the user can still interact with the submenus as shown in Figure 1-(d) because the last opened submenu is likely to be the main focus of interest of the user.

	Wave menu	Wavelet menu
		
Circular layout	✓	✓
Previsualization	✓	✓
Stroke shortcuts	✓	✓
Stacking metaphor	✗	✓
Long list management	✗	✓
Platform	PC	**iPhone**
Finalization	Prototype	**Advanced product**

Table 1: Analytical comparison of Wave and Wavelet menus

A limitation of Marking menus (Hierarchical Marking menu, Multi-Stroke menu, Wave menu) is that the breath of a menu is equal to 8 in order to guarantee good performance (i.e., large enough sectors to be easily selected especially with a finger) [10] [16]. The breath limitation may imply an increased menu depth for the case of a large number of items. Navigating in a larger number of submenus may increase decision time and possible disorientation. To solve this problem since long lists of multimedia data are commonly available on handheld devices, the Wavelet menu is hybrid and combines two possible representations: according to the number of items that must be displayed, items are either laid out in a circular or linear way. For preserving the staking metaphor, the linear representation appears from "behind" the Wavelet menu, as shown in Figure 2.

Figure 2. Wavelet menu: Hybrid menu combining circular and linear menus. Once the user releases her/his finger, the circular menu progressively disappears thanks to a short animation.

If the user taps once in the central area, the deepest submenu is closed and its parents are moved inwards in a short collapsing animation. A double tap in this area closes all the submenus. To provide a rich user experience, the Wavelet menu emphasizes **direct manipulation**: The user can drag an item inward or outward to close or open a submenu. Moreover the Wavelet menu also provides **previsualization**. The user can then make a continuous circular gesture to browse all submenus as shown in Figure 3. Submenus are automatically displayed when the finger hovers over the corresponding item in the parent menu. This makes it possible to browse the menu system in an efficient and natural way. This is a major advantage as compared with Multi-Stroke menus [17] that force users to perform "blind" up and down transitions in the menu tree. Previsualization is a key property to allow the user to efficiently explore and learn the content of a menu [2]. Inherited from the Wave menu (Table 1), this property is generally not supported by existing menu techniques on handheld devices.

Figure 3. Previsualization using the Wavelet menu. (a) The user can browse all the submenus by performing a continuous circular gesture. (b) For consistency, the user can also browse linear lists with the same mechanism.

The expert mode of the Wavelet menu is the same one as that in the Multi-Stroke menu [17] that has been shown to outperform classical hierarchical Marking menus. The elementary strokes are drawn in quick succession and each stroke (which is inflexion free) is completed when the user releases its finger from the screen. One particular property of this expert mode is the scale invariance of the marks. This makes it possible to anticipate the usage of the menu without looking at the screen by performing eyes-free selection of commands.

4. WAVELET MENU IMPLEMENTATION

The Wavelet menu has been implemented in Objective-C using the iPhone Cocoa Touch API. We fully followed the Cocoa programming guidelines. We therefore applied the Cocoa Model-View-Controller (MVC) design pattern. The Model includes the menu hierarchy that is described in XML as an external resource file. The View is composed of two independent parts that ensure modifiability: the first one manages the display of the Wavelet menu and the second one is a set of hierarchical state machines that manage the input interactions (touch-screen). For example, it is easy to add multi-touch or accelerometer-based interactions by defining new state machines and without modifying the part that displays the menu. The Controller acts as a cement that allows us to connect the menu with other Cocoa interactors (linear lists, contact lists, custom views, etc.). Moreover, the management of users' actions makes it very simple to use the Wavelet in any iPhone application. Indeed any object can be the target responder of a Wavelet menu item selection. Finally the code of the Wavelet menu is ready to be integrated in Interface Builder as a widget in the palette. We also plan to provide the Wavelet menu as a standalone iPhone application (e.g. a multimedia manager).

The menu has been demonstrated during two conferences allowing us to test its robustness and to collect informal feedback. For instance, several participants deplored the lack of tactile feedback during the first demonstration. We thus added this feature, which was tested during the second demonstration. The tactile feedback consists of a short vibration that occurs once the user hovers over a new item in expert mode. In this way the user can feel the boundaries of each item so that the selection accuracy is improved. Moreover the controlled experiment of the Wavelet menu described in Section 6 provides guidelines for designers to organize the menu items by identifying where frequently used items should be preferentially placed in the menu for better performance. Hence, the most frequent items (e.g. for starting a phone call) should be positioned along the on-axes.

To sum up, the Wavelet menu is fully developed on the iPhone as an advanced product. By combining the circular layout of the menu with a more classical linear one, the Wavelet menu also provides a realistic solution to the breath limitation of Marking menus (which may be one reason why they are rarely used in commercial applications). Because of these reasons, the Wavelet menu is ready to be integrated and used in any iPhone application.

5. EXPERIMENT 1: LEARNING OF THE TECHNIQUE

Users' ability to learn a novel interaction technique is a prerequisite for its acceptance. Users must be able to quickly understand how the technique works in novice mode else the technique will be rejected even if it is an efficient one. However, learning ability is uneasy to evaluate, and this may be the reason why this aspect has been mainly neglected in former studies on menu techniques. The goal of the experiment described below was to do so by comparing the learning performances of Wave and Wavelet menus. In particular, we wanted to check whether the stacking metaphor helps users in understanding the working principle of Wavelet menus.

In order not to bias the experiment in favor of the Wavelet menu and to only focus on the stacking metaphor, both menus were implemented on the iPhone by using the same graphical layout (with the aesthetic black ring). Six participants ranging in age from 20 to 25 were divided into two groups in order to counterbalance the results. The first group was asked to select six 1-level or 2-level items on the Wave menu, without any explanation. Then participants were asked to select six 1-level or 2-level items on the Wavelet menu without explanation. Participants from the second group did the same experiment in reverse order (Wavelet first). At the end of each part, they were asked to fill in a questionnaire. At the end of the experiment, participants were also asked to fill in a questionnaire that focused on the comparison of the two menus.

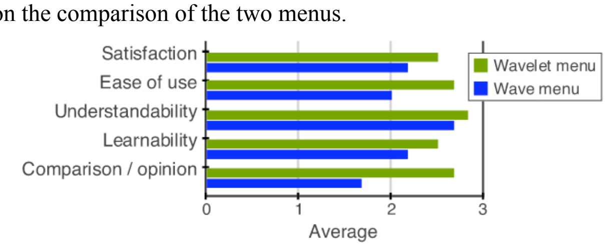

Figure 4. Comparative evaluation of the Wave and the Wavelet menu on the iPhone: average answers.

Figure 4 shows the results for five questions. For each question, participants had four choices, ranging from "I completely disagree" (0) to "I completely agree" (3). Figure 4 shows the average answers for each menu. As the groups were symmetrical, the results have been combined. They show that the Wavelet technique is easier and faster to understand. Five participants said that they understood the Wavelet technique thanks to the graphical ring representation and the stacking metaphor. Only two participants said that the manipulation of the Wavelet menu helped them to understand the technique. In contrast, five participants needed to manipulate the Wave menu to understand it. The Wavelet menu was also easier to use and more pleasant than the Wave menu. All participants said that the ease of use is the criterion that counts the most in the choice of a menu technique. This experiment showed that the design of the Wavelet menu makes it easier to learn. Because of its stacking metaphor, the perceived affordance [11] is better than for the Wave menu: The Wavelet design suggests how the menu works and offers visual feedback that helps user interaction.

6. EXPERIMENT 2: EXPERT MODE

For the case of handheld touch-screen devices, two design parameters of the stroke-based expert mode are important to study, especially in mobile situations: the starting point of the strokes and the visual feedback. On the one hand the starting point of each stroke can be fixed to a central area of the screen, making the strokes to be performed in a manner very close to the ones performed in novice mode. On the other hand, the stroke could be initiated from any point of the screen, this offering more flexibility to the users while moving. This design alternative makes gestures completely independent from the graphical design of the menu. In expert mode, the menu is not displayed and therefore the screen does not require full visual attentional resources. Nevertheless visual lexical feedback that may require less visual attention can be provided or not. One design solution is to display the ink trail of the gesture as well as the central area of where to start the gesture. An alternative solution is to consider complete eyes-free selection of commands so that nothing is displayed on the screen in expert mode. For studying these design alternatives of the expert mode of the Wavelet menu, we performed an experiment by comparing them in two different configurations: either while sitting or walking.

6.1 Goals

We performed the experiment to answer the following questions:

Is there a performance difference for certain mark directions?

Previous studies on hierarchical Marking menus [10] and Multi-Stroke menus [17] have shown that performance is lower when the user has to deal with off-axis marks instead of on-axis marks. On-axis marks are the ones that are horizontal and vertical (N, S, E, W on a compass) and off-axis marks are the 45° marks (NE, NW, SE, SW). These studies have been conducted with a mouse and a pen. Due to anatomy differences, the same study must be conducted to know if there are some directions that are easier to hit than others with the thumb on a small touch-screen device.

What is the impact on speed and accuracy to bind start points of the marks in the central area of the screen?

We have seen that the novice mode of the Wavelet menu emphasizes direct yet intuitive handling. For example, we believe that allowing touch and movement of an item to expand or reduce its submenu could be more intuitive for some users than just drawing a stroke from the center or clicking the central area. Thus making a stroke from the central area or from the "item area" might trigger different behaviors. Knowing if the user is able to target the central area, even while in motion, is then a key question in validating our design.

How does the lack of visual feedback affect speed and accuracy?

The expert mode is used without the menu being displayed. However it is interesting to compare the respective performances of menu selections once they come with a visual feedback (display of the current mark and of the central area) or not in the studied cases (mobility, constraints on the starting point). Answering this question could determine whether simple mark combinations can be performed fully eyes-free while in motion.

How does being mobile affect speed and accuracy?

One advantage of the Wavelet menu is to provide an expert mode that can be used in mobile situations. Enabling the user to select a command with a stroke gesture on the touch-screen is a promising avenue for mobile users. There could be a significant value of the Wavelet menu to support efficient selection of commands while the user is in motion. We therefore consider two configurations i.e., the user sitting and the user walking, to explore the possible differences of performance (speed and accuracy).

6.2 Participants and Apparatus

Six right-handed participants ranging in age from 21 to 27 years (mean 24) were recruited from within the university community. The subjects were skilled in using mobile devices, but were mainly not familiar with using a touch-screen based phone. None of them had previous experience with Marking menus. The experiment was conducted with two iPhones and a MacBook Pro (Figure 5). The experimental software was a distributed application made up of three parts: a server running on the MacBook Pro and implemented in Objective-C/Cocoa; and two different clients running on the iPhones and implemented in Objective-C/Cocoa Touch. One iPhone was handled by the left hand and was used to display the stimuli to be reproduced and the other one was handled by the right hand and was used to recognize the strokes and to possibly provide a visual feedback. The MacBook Pro was used to generate new stimuli, to log user actions and to manage the communication by an ad-hoc Wi-Fi network used to connect the devices with a reliable protocol.

6.3 Task and Stimuli

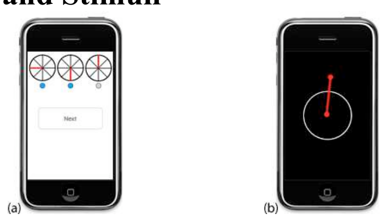

Figure 5. Screenshot of experiment setup: (a) iPhone handled by the left hand to display the stimuli. The participant is instructed to perform three marks. (b) iPhone handled by the right hand displaying visual feedback (central area of where to start the gesture and ink trail of performed gesture).

We intended to evaluate the ability of the users to perform strokes on the small touch-screen in various conditions in expert mode. In this mode, the menu is not displayed and the user has to draw the marks from memory. This requires the user to be completely familiar with the menu layout. Asking participants to learn the entire set of marks of a menu was not realizable in the context of this controlled experiment. Moreover, we did not seek to study user learning, as [10] has already shown that Marking menus are easy to learn because of the fluid transition between novice and expert modes. Thus we modeled our experimental task after [4] and eliminated the need to learn the menu layout by directly displaying the strokes to be performed. We therefore only focused on the ability of the user to draw the strokes without seeing the menu and we set aside the memorization phase. We have chosen to evaluate the expert mode for the worst case of the Wavelet menu, i.e. with a breadth equal to 8 on every level. We have also decided to use a 3-level menu. In this way we had enough material to study the marks and the combinations of marks. Moreover by considering the first stroke only, we obtained results for a 1-level menu and by considering the two first strokes we obtained results for a 2-level menu. The stimulus was a picture of the 3 successive marks that the participant had to draw (Figure 5). Each time a new stroke was performed, a light turned blue on the left device to inform the user. Once the participant had drawn 3 consecutive marks (correct or not) using the right device, s/he was invited to touch the “next” button on the left device to display a new stimulus. If the three drawn marks were correct, a short song was played, otherwise a short vibration occurred.

6.4 Design and Procedure

The experiment was divided into two parts, each composed of four blocks. During the first part, participants were sitting in front of a table in a quiet room. During the second part they were walking in a long and quiet corridor to simulate a mobile situation in which the user has to share its attention between the device and an external context. All the participants performed the experiment in the same order. This division into two parts has been made for practical reasons, in order to avoid participants alternating between sitting and walking phases.

	Starting point anywhere	Visual feedback
Block 1	Yes	Yes
Block 2	No	Yes
Block 3	Yes	No
Block 4	No	No

Part 1: Sitting

	Starting point anywhere	Visual feedback
Block 5	Yes	Yes
Block 6	Yes	No
Block 7	No	Yes
Block 8	No	No

Part 2: Walking

Figure 6. Experimental design: First part: The participant is sitting; Second part: The participant is walking.

Each part is composed of four blocks of item selections, ordered by increasing estimated difficulty. Each block involved two parameters: (a) the starting point of the mark (whether it must start in the central area of the screen or not) and (b) the presence or not of a visual lexical feedback on the device. According to the Wavelet menu design, the central area of the screen contains a centered disk with a 92 pixels radius. The visual feedback consists of a red segment that links the start point with the current point, and of a circle delimiting the central area (if needed). Figure 6 details the 8 blocks of the experiment. For each block, participants made selections in a 3-level, 8-item Wavelet menu in expert mode. Asking participants to perform the entire set of possible combinations (512) for each block would have resulted in a too long experiment. Thus, as in [10] [17], we decided to choose a subset of menu items, equally distributed between on-axis marks (N, S, E, W) and off-axis marks (NE, SE, SW, NW). Therefore, there are 8 possible combinations of on/off-axis marks: on-on-on, on-on-off, on-off-on, on-off-off, off-on-on, off-on-off, off-off-on, off-off-off. We randomly chose 8 items from each of these combinations, which resulted in 64 item selections.

After a brief explanation, the participants performed a set of warm-up strokes for every configuration. The entire experiment was filmed. Participants could take short breaks between blocks and were encouraged to express their feelings/preferences during these breaks. A 4-minute break was done between the two parts of the experiment in order to fill in a questionnaire. Each participant performed the experiment in approximately 45 minutes.

In summary, the design was as follows:

6 participants X 2 parts X 4 blocks X 64 menu selections
= 3072 3-level menu selections in total.

6.5 Results

The results of the experiment include data regarding accuracy, selection time and subjective preference of the participants.

6.5.1 Accuracy

Accuracy refers to the percentage of menu selections that matched with the given stimulus. As the data are discrete (the menu selection is correct or not), we performed a Pearson's Chi-square test to check the independence of the variables. For each case, we decided to reject the null hypothesis (i.e. the variables are independent) if the p-value was less than a value of 5%.

Figure 7 shows the error rates for the 8 blocks. For each block, it displays the error rate for the first mark (which corresponds to a 1-level menu selection), for the 2 first marks (which correspond to a 2-level menu selection), and for the 3 marks (which correspond to a 3-level menu selection). Error rates were ranging from: 4% (block 1) to 15% (blocks 4, 8) for 1-level menu selections; from 7% (block 1) to 24% (blocks 4, 8) for 2-level menu selections; and from 10% (block 1) to roughly 30% (blocks 4, 8) for 3-level menu selections. This shows that regardless of the depth of the menu, the most difficult blocks were always the 4th and the 8th blocks (i.e. marks starting from the fixed central area of the screen and no visual feedback), which is consistent with our expectations.

As shown in Figure 8, a fine-grain analysis considered the three parameters that were tested during the experiment: mobility, starting point of the marks and visual feedback.

Mobility had a small effect on accuracy for 1-level menu selections ($\chi^2 = 5.1$, $df = 1$, $p < .02$), with an error rate of 7.4% (sitting) and 9.7% (walking). No effect on accuracy was observed for 2-level menu selections but an effect was observed for 3-level menu selections ($\chi^2 = 5.8$, $df = 1$, $p < .01$).

Constraining the marks to start from the central area had a significant effect on accuracy. For 1-level menu selections ($\chi^2 = 34.4$, $df = 1$, $p < .0001$), error rates were 5.6% (not constrained) and 11.5% (constrained to the central area). An effect on accuracy was also observed for 2-level ($\chi^2 = 53.4$, $df = 1$, $p < .0001$) and for 3-level ($\chi^2 = 45.8$, $df = 1$, $p < .0001$) menu selections.

The presence or not of a visual feedback also had a significant effect on accuracy. For 1-level menu selections ($\chi^2 = 16.5$, $df = 1$, $p < .0001$), error rates were 6.5% (feedback) and 10.6% (no feedback). A similar effect on accuracy was observed for 2-levels ($\chi^2 = 32.8$, $df = 1$, $p < .0001$) and 3-levels ($\chi^2 = 43.4$, $df = 1$, $p < .0001$). The effect thus clearly increases with the menu depth.

The above analysis focused on error rates without distinguishing between mark orientations. The next step was to compare the 8 possible orientations of the marks to know if some marks are harder to draw than others. To do so, we considered if the marks were constrained to start from the central area or not, as the accuracy of drawing a mark is likely to be different if the user can start anywhere on the screen or not. Our analysis showed that there was a significant effect for mark orientation on error rates when the marks have to start from the central area. For example, the two most difficult orientations for the first marks ($\chi^2 = 26.7$, $df = 7$, $p < .001$) were North-West (20.0%) and South-East (16.7%) while all the other orientations corresponded to an error rate of under 11%. Figure 9-(a) details these results. A smaller effect appeared when the marks were not constrained ($\chi^2 = 18.2$, $df = 7$, $p < .01$): South-East (10.1%) and North-West (8.7%) were the less accurate performed marks. We also studied the effect for mark direction (on-axis, off-axis) on accuracy. The analysis for the first marks showed that there is no effect for mark direction for both of the two cases of unconstrained marks and of marks starting from the central area. Additionally there was no effect for combination of the three mark directions on accuracy for the 3-level menu selections for both cases (unconstrained and constrained marks).

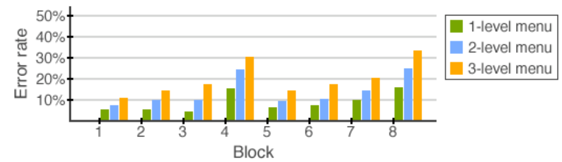

Figure 7. Error rate by block (8 blocks) and by menu level (1-level, 2-level and 3-level menu selections).

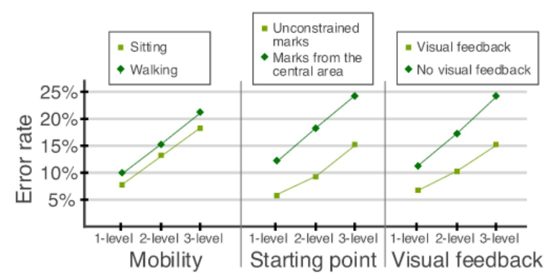

Figure 8. Error rate by parameter (mobility, starting point of the marks and visual feedback) and by menu level (1-level, 2-level and 3-level menu selections).

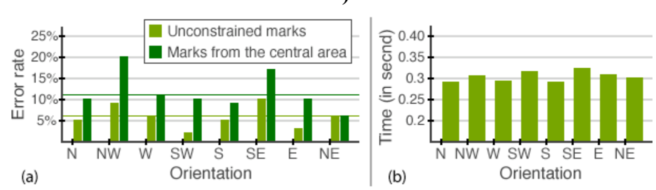

Figure 9. First marks: (a) Error rate by orientation (8 orientations) for the two cases of unconstrained marks and of marks starting from the central area. (b) Average selection time by orientation (8 orientations).

6.5.2 Selection time

The time needed to select an item in a 3-level Multi-Stroke menu comprises two components: the *reaction time* (interval between the appearance of the stimulus and the first touch of the screen) and the *selection time* (interval between the first touch and the end of the last stroke). As we are interested in the ability of users to draw the marks in expert mode, we focused on the selection time. Figure 10 shows the average times for each block.

The ANOVA for 3-level menu selections indicated an effect for mobility on selection time ($F_{1,11} = 5.2$, $p < .02$), with an average time of 1.48 second for the case where participants were sitting and 1.52 second for the case where they were walking. A similar effect was observed for 2-level menu selections but there was no effect for 1-level menu selections, as shown on Figure 11.

Constraining the marks to start from the central area also had a significant main effect on selection time ($F_{1,11}$ = 45.2, p < .0001). For 3-level menu selections, the average selection times reached 1.55 seconds for the constrained marks and 1.46 seconds for the non-constrained ones. A similar effect was observed for 2-level menu selections. However, there was no significant effect for 1-level menu selections (Figure 11).

Finally the presence or not of a visual feedback had no effect on selection time (Figure 11).

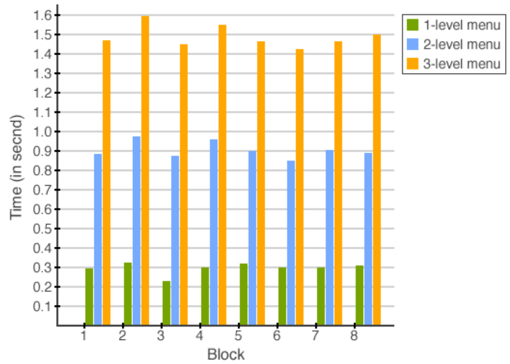

Figure 10. Average selection time by block (8 blocks) and by menu level (1-level, 2-level and 3-level menu selections).

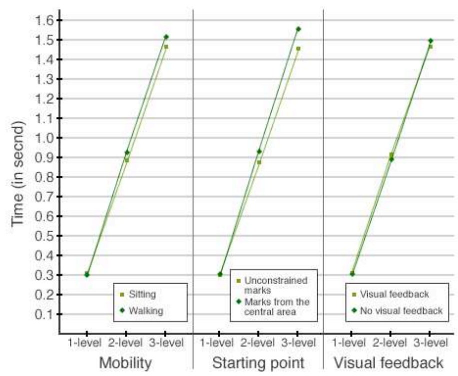

Figure 11. Average selection time by parameter (mobility, starting point of the marks and visual feedback) and by menu level (1-level, 2-level and 3-level menu selections).

Going into the details of mark orientations, we considered the first marks only (as for accuracy in the previous section). The analysis showed a significant effect for orientation on selection time ($F_{1,11}$ = 4.4, p < .0001). This means that some marks are performed faster than others: A post hoc Tukey test revealed that South-East and South-West correspond to the orientations of the slowest marks (0.32 second on average), while North, South and West are the fastest ones (0.29 second on average). Figure 9-(b) shows the average times for the first marks by orientation. More generally, the effect was different depending on the mark directions ($F_{1,11}$ = 15.1, p < .0001): On-axis marks corresponded to an average selection time of 0.29 second while off-axis marks to an average selection time of 0.31 second. Thus markings which consist of on-axis items out-perform off-axis markings as experimentally observed in [17].

6.5.3 Subjective preference

According to the post-study questionnaires, four participants over six said that the experiment was "very satisfying", arguing that drawing marks on the screen was pleasant. One participant said that drawing marks towards the right orientation was sometimes difficult due to the inclination of the device. Every participant expressed that on-axis marks were easier than off-axis marks. Two participants detailed this observation and said that South-East orientation was the most difficult mark to draw.

Every participant has noticed an increase of the difficulty in both parts of the experiment, and said that Part 1 was easier than Part 2. More precisely, blocks 1 and 3 were the easiest ones and block 4 was the hardest one of the first part; blocks 5 and 6 were the easiest ones and block 8 was the hardest one of the second part.

According to the participants, walking implies a different behavior than when sitting. Actually, of the six participants, four said that looking at the screen while walking was rather useless, except occasionally to find again the central area after a few of wrong item selections. One participant said that he consciously never looked at the screen when walking. Therefore five of the six participants said that the visual feedback was rather useless when walking. In contrast, three participants said that looking at the screen was rather useful when sitting. A majority of participants said that they only looked at the screen with their peripheral vision in order to target the central area. Based on the perceived difficulties of the blocks, finding the central area appeared to be a difficult task when walking. Contrastingly three participants said that it was rather easy to target the central area when walking, while the six participants said that is was rather easy when sitting.

6.6 Discussion

We can now attempt to answer the questions posed earlier:

Is there a performance difference for certain mark directions?

The results have clearly shown that North-West and South-East are the two orientations of the slowest and most inaccurate marks. Although, there was no difference in accuracy for on-axis and off-axis directions, our results showed that off-axis marks are 6% slower than on-axis marks. However, there was no performance difference for 3-level combinations of on-axis and off-axis marks. Such a result must be taken into account when designing a menu in order to position the less frequent items on the less efficient orientations. Another possibility consists of increasing the sector angle for the items difficult to select. It is important to note that the experiment evaluated the expert mode for the most difficult case of the Wavelet menu with 8 items at each level. With less items at a given level, the sector angle of some items can be increased in order to obtain better performance.

What is the impact on speed and accuracy to bind start points of the marks in the central area of the screen?

The results have shown that performance is highly influenced by this parameter. A 3-level Wavelet menu is 38% less accurate and 7% slower if the starting points of the marks are constrained to be in the central area of the menu. With an error rate of higher than 30% when there is no visual feedback (blocks 4 and 8), we conclude that users cannot use the technique in expert mode with enough performance for the case of a 3-level menu. Given these results, the best compromise is to avoid the central area constraint in expert mode in order to ensure better performance. In this way the novice mode of the Wavelet menu could support direct handling of items when gestures do not start from the central area, while the expert mode could only support stroke shortcuts regardless of the starting point location. Our result showed that such simplification of the expert mode ensures good performance even when there is no visual feedback (blocks 3 and 6).

How does the lack of visual feedback affect speed and accuracy?

The results have shown that displaying the trail of the mark while the user interacts highly influences performance. Disabling the visual feedback triggered 38% less accurate selections for a 3-level Wavelet menu. However, there was no effect of visual feedback on selection time. Thus providing a visual feedback may significantly improve accuracy. This assertion needs further consideration in the light of our adopted design solution of unconstrained marks for the expert mode (see previous question). Indeed on the one hand the visual feedback mainly affects blocks that impose the central area constraint (block 2 versus block 4 and block 7 versus block 8). On the other hand, blocks that do not impose this constraint obtain similar performance with and without the visual feedback (block 1 versus block 3 and block 5 versus block 6). Additionally Chi-square analyses showed that, when the participants were sitting, there was only a small effect for visual feedback on accuracy for blocks that do not constrain the starting area (e.g. ($\chi^2 = 6.8$, $df = 1$, $p < .02$) for 3-level menu selections). Furthermore, when the participants were walking, there was no effect of visual feedback for the blocks that do not constraint the starting area of the marks. The results showed that visual feedback is not required when users are walking. Thus using the expert mode of Wavelet menu in mobile situations is possible with acceptable performance if the starting point of the marks is not constrained, even if there is no visual feedback (eyes-free selection). As 4 of 6 participants said that they often looked at the marks with their peripheral vision when they were sitting, and 2 of 6 participants when they were walking, displaying the trail of the marks in expert mode could be an interesting design solution to help users especially when they are not mobile.

How does being mobile affect speed and accuracy?

The results have shown that performance is affected by mobility. The selection time is only affected for the case of hierarchical menus (2% slower for 2-level menus and 2.6% slower for 3-level menus). Menu selections are 24% less accurate while walking, independently of the depth of the menu. As for the previous question, we further studied these results according to the constraint on the starting point of the marks. Comparing block 3-sitting- with block 6-walking- (the two other parameters being set to <no constraint on the starting point> and to <no visual feedback>) showed that there was no effect for mobility on accuracy. As a consequence, the Wavelet menu can be used in expert mode in mobile situations if there is no constraint on the starting point of the marks.

7. CONCLUSIONS AND FUTURE WORK

Command selection in a menu is a fundamental task in interactive applications that requires new designs for addressing the challenges of handheld touch-screen devices. In our study of the Wavelet menu on handheld devices, we highlighted the dual need to address both the novice and the expert modes of the menu technique since the two modes always co-exist and the novice mode is an unavoidable step before the expert mode [2]. Our two user studies of the novice and expert modes of the Wavelet menu showed that the Wavelet menu is easy to learn and efficient to use even in mobile situations while the user is walking. This makes it an effective and promising eyes-free menu technique. Two interesting areas of future research would be to explore whether motion-based gestures could be used for the expert mode and how multi-touch capabilities could improve the efficiency of the Wavelet menu for both its novice and expert modes.

8. REFERENCES

[1] Appert, C., Zhai, S. 2009. Using Strokes as Command Shortcuts: Cognitive Benefits and Toolkit Support. Proc. of ACM CHI’09, 2289-2298

[2] Bailly, G., Lecolinet, E., Nigay, L. 2007. Wave Menus: Improving the Novice Mode of Hierarchical Marking Menus. Proc. of INTERACT’07, Springer LNCS, 475-488.

[3] Bailly, G., Lecolinet, E., Nigay, L. 2009. MenUA: a design space of menu techniques. www.gillesbailly.fr/menua/

[4] Balakrishnan, R., Patel, P. 1998. The PadMouse: Facilitating Selection and Spatial Positioning for the Non-Dominant Hand. Proc. of ACM CHI’98, 9-16.

[5] Francone, J., Bailly, G., Lecolinet, E., Nigay, L. 2009. Wavelet menus: a stacking metaphor for adapting marking menus to mobile devices. Ext. Abstracts of ACM MobileHCI’09.

[6] Francone, J., Bailly, G., Lecolinet, E., Nigay, L. 2009. Wavelet menu : une adaptation des marking menus pour les dispositifs mobiles. Proc. of ICPS IHM'09, 367-370.

[7] Huot, S., Lecolinet, E. 2007. ArchMenu et ThumbMenu : Contrôler son dispositif mobile "sur le pouce". Proc. of ICPS IHM'07,107-110.

[8] Karlson, A., Bederson, B., Contreras-Vidal, J. 2008. Understanding One Handed Use of Mobile Devices. In Lumsden, J. ed., Handbook of Research on User Interface Design and Evaluation for Mobile Technology, IGI, 86-101.

[9] Kurtenbach, G., Buxton, W. 1991. Issues in Combining Marking and Direct Manipulation Techniques. Proc. of ACM UIST’91, 137-144.

[10] Kurtenbach G., Buxton, W. 1993. The limits of expert performance using hierarchical marking menus. Proc. of ACM CHI'93, 35-42.

[11] Norman, D. 1999. Affordances, Conventions, and Design. Interactions, 6 (3), 38-41.

[12] Rekimoto, J., Ishizawa, T., Schwesig, C., and Oba, H. 2003. PreSense: interaction techniques for finger sensing input devices. Proc. of ACM UIST’03, 203-212.

[13] Roudaut, A., Lecolinet, E., Huot, S. 2008.Taptap and MagStick: Improving One-Handed Target Acquisition on Small Touch-screens. Proc. of ACM AVI’08, 146-153.

[14] Roudaut, A., Bailly, G., Lecolinet, E., Nigay, L. 2009. Leaf Menus: Linear Menus with Stroke Shortcuts for Small Handheld Devices. Proc. of INTERACT’09, Springer LNCS, 616-619.

[15] Zhai, S., et al.. 2009. Shapewriter on the iPhone: from the laboratory to the real world. Ext. Abstracts of ACM CHI’09, 2667-2670.

[16] Zhao, S., et al. 2006. Zone and polygon menus: using relative position to increase the breadth of multi-stroke marking menus. Proc. of ACM CHI’06, 1077-1086.

[17] Zhao, S., Balakrishnan, R. 2004. Simple vs. compound mark hierarchical marking menus. Proc. of ACM UIST'04, 33-44.

[18] Zhao, S., et al. 2007. Earpod: eyes-free menu selection using touch input and reactive audio feedback. Proc. of ACM CHI '07, 1395-1404.