Wavelet Menus: A Stacking Metaphor for Adapting Marking Menus to Mobile Devices

Jeremie Francone[1] Gilles Bailly[1,2] Laurence Nigay[1] Eric Lecolinet[2]

[1]LIG, University of Grenoble, 385, Av. de la Bibliothèque, Grenoble, France
[2]TELECOM ParisTech – CNRS LTCI, 46 rue Barrault, 75013 Paris, France
{jeremie.francone; laurence.nigay}@imag.fr {gilles.bailly; eric.lecolinet}@telecom-paristech.fr

ABSTRACT

Exploration and navigation in multimedia data hierarchies (e.g., photos, music) are frequent tasks on mobile devices. However, visualization and interaction are impoverished due to the limited size of the screen and the lack of precise input devices. As a result, menus on mobile devices do not provide efficient navigation as compared to many innovative menu techniques proposed for Desktop platforms. In this paper, we present Wavelet, the adaptation of the Wave menu for the navigation in multimedia data on iPhone. Its layout, based on an inverted representation of the hierarchy, is particularly well adapted to mobile devices. Indeed, it guarantees that submenus are always displayed on the screen and it supports efficient navigation by providing previsualization of the submenus.

Categories and Subject Descriptors

H.5.2 **[Information Interfaces and Presentation]**: User Interfaces– Evaluation / methodology, Graphical user interfaces (GUI), Interaction styles (e.g., commands, menus, forms, direct manipulation); I.3.6 **[Computer Graphics]**: Methodology and Techniques– Interaction techniques

General Terms

Design, Human Factors.

Keywords

Mobile devices, Menu Techniques, Wave menus, Marking menus.

1. INTRODUCTION

Exploration and navigation in multimedia data hierarchies (e.g., phone contacts, music albums, videos, radio programs, podcasts) are common tasks on mobile devices. As shown in [1], *previsualization* significantly improves navigation in hierarchical menu systems. In these systems, previsualization [1][8] allows the user to see the content of a submenu when she drags the mouse over the corresponding menu item. As this technique does not require a mouse click, the user can browse the content of the whole menu system in a very fluid manner, just by dragging the mouse. By providing a quick overview of the available commands, this proactive feedback makes it possible to avoid many unnecessary up and down transitions in the hierarchy. Unfortunately, this important feature is often missing on mobile device menu systems. The probable reason is that previsualization requires several menus to be displayed simultaneously on the screen (typically, the current menu and the submenu being previsualized). The limited amount of screen real-estate on small devices thus imposes a trade-off between space occupation and navigation efficiency. As a consequence, navigation in menu systems is inherently more difficult on mobile devices than on the standard PC, not taking into account the fact that item selection is easier when using a mouse rather than a stylus (or a finger) on a handheld touchscreen.

Moreover, mobile interaction imposes various constraints on the inputs: 1) hotkeys are obviously not available on devices that do not have a keyboard (as the iPhone and most recent smartphones); 2) passive touchscreens do no make it possible to “right click” for opening menus (temporal delays must be used instead); 3) users generally prefer to interact directly with fingers (rather than using a stylus) and by using only one hand [7]. This implies several problems such as occlusion or limited accuracy, especially when interacting with the thumb [9]. Finally, attention is generally divided when using a mobile, and the user may even be unable to look at the device (typically when he is walking, running or driving).

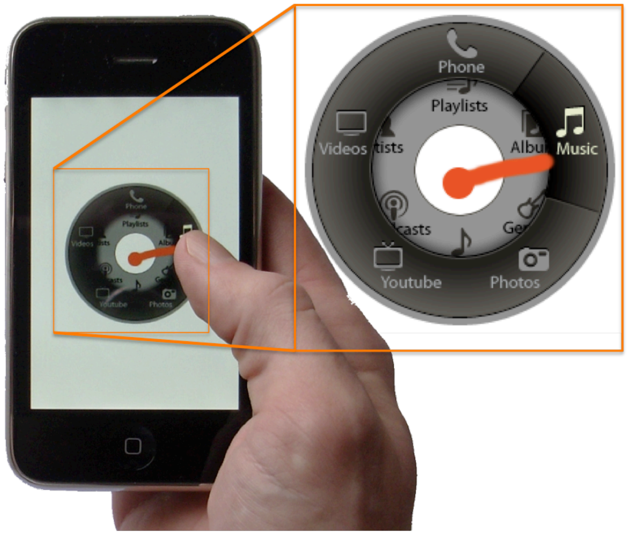


Figure 1: Using the Wavelet menu technique for navigating on the iPhone.

MobileHCI09, September 15 - 18, 2009, Bonn, Germany.
ACM 978-1-60558-281-8/09/09.

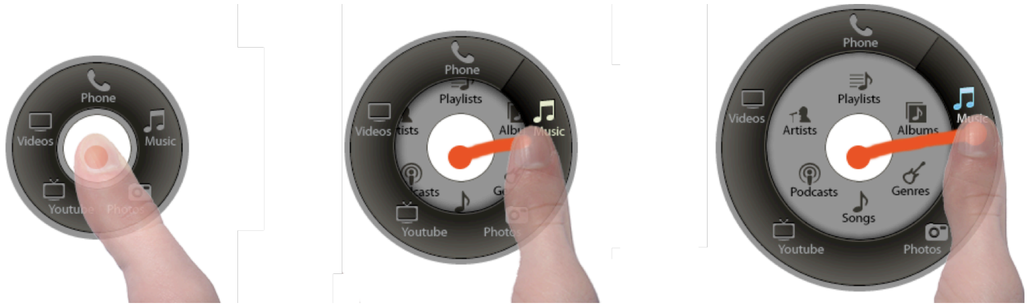


Figure 2: The Wavelet menu appears centered around the contact point. By drawing a stroke towards the desired item, the first level is enlarged permitting progressive appearance of the submenu. A second stroke selects an item in the submenu.

This paper presents the *Wavelet Menu* (Figure 1)*,* a new menu technique that was developed as an attempt to answer some of the problems described above. The Wavelet technique is an improvement of hierarchical Marking menus [6] that is specifically designed for mobile devices. Its design is inspired by Multi-Stroke [13] and Wave menus [1], as explained below.

This paper is organized as follows: we first discuss related work. We then present the design of the Wavelet menu and explain why they are well suited to the constraints of mobile devices.

2. RELATED WORK

We first present menu techniques designed for desktop platforms and their limitations on mobile devices. We then describe some menu techniques that were specifically conceived for handheld devices.

Linear menus, which are so commonly used on PCs, offer previsualization to facilitate navigation. However, most often, their mobile versions do not provide this feature nor hotkeys. Besides, selecting small items on a touchscreen may be a challenge, especially when the thumb is used [10]

Marking menus [6] combine circular menus and gestural interaction by proposing two functional modes. When the user waits for about 300 ms, the menu enters in *novice mode* and appears on the screen. The user can then select an item in the usual way. Conversely, *expert mode* is triggered if the user does not wait for this delay: the menu does not appear and the user selects a command by making a stroke in the direction of the desired item. The key property of Marking menus is that they support a seamless transition from novice to expert mode because the user executes the same gesture in both modes. In other words, the user learns the expert mode implicitly just by using the menu repeatedly in novice mode. Marking menus can be hierarchical, in which case strokes are spatially composed. Their original version supports previsualization, although this property has been lost in some variants of this technique. A major drawback of hierarchical Marking menus is that they require much space in novice mode, a characteristic that is especially unwelcome on mobile devices. This is not only because of their circular shape but also because their submenus are always displayed in the direction of the item that opens them (in contrast, linear submenus can either appear on the left or right side of the item depending on the available space). A three-level Marking menu hence requires more horizontal space than 10 linear menus to display its leftmost and rightmost branches [1]. Besides, performance tends to decrease when the depth increases, especially for gestures along the diagonal axes.

Multi-Stroke menus [13] solve this last problem by adopting a “temporal” strategy: users must draw a series of simple inflexion-free marks instead of a single compound mark. This strategy makes recognition performance mostly independent of the menu depth. Submenus are superimposed in novice mode and thus require less physical input space than for "compound" marking menus. Unfortunately, superposition makes previsualization impossible.

ThumbMenus and ArchMenus [5] are dedicated to mobile devices. They use a semi-circular layout to avoid the occlusion of the thumb while interacting. However, they have no expert mode and can only contain a limited number of items. The RollMark menu [9] is a circular menu that uses roll movements of the thumb, a new type of gesture whose specific signature makes it possible to be unambiguously recognized on a touchscreen. But these menus are not hierarchical and limited to 6 items. Earpod [14] is a circular menu, which is augmented with audio feedback. In order to provide eyes-free selection, item names are played when an item is selected. This technique does not provide previsualization as it mainly focuses on eyes-free selection.

3. WAVELET MENU

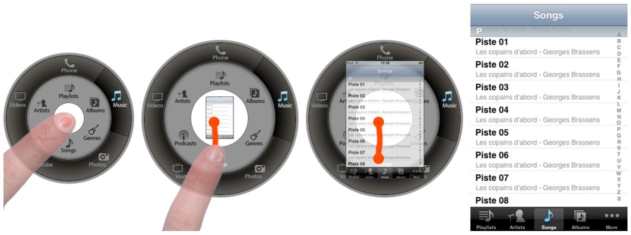
3.1 Principle

The Wavelet technique (Figure 1) is inspired from the Wave menu [1], which was recently introduced for desktop platforms as a way to enhance Multi-Stroke menus [13]. Wave and Multi-Stroke menus have the same expert mode: the user draws a series of simple inflexion-free strokes that can be overlapped. This design was chosen because it has been shown to allow better performance (i.e. less erroneous selections) than hierarchical Marking menus [13].

In novice mode (i.e. after a 300ms delay), the root menu appears as a ring centered around the cursor (Figure 2). To select an item, the user draws a mark in the direction of the desired item. Besides its different graphical representation, a major difference with the Multi-Stroke menu is that the menu (i.e. the ring that represents it) is enlarged during the interaction as if the cursor "pushed" it from the inside. When the ring is sufficiently enlarged, the submenu appears in the center. The same effect occurs again if another stroke is then performed in the submenu: the submenu and the root menu move outwards from the center and the third-level menu appears in the center of the representation. This effect recalls the propagation of waves, and gave its name to the technique. Hence, whatever the depth of a Wave menu system, the root menu is always the outmost menu while the deepest submenu currently shown is always displayed at the center. Clicking in the center (i.e. on the white area in Figure 2) closes the deepest submenu and all its parent menus move inwards. An animation is then performed to produce a “collapsing effect”.

3.2 Application to mobile devices

The Wavelet menu is an adaptation of the Wave menu for mobile devices. Our prototype has been designed to explore and navigate multimedia data. It thus appears in the center of the device screen (Figure 1) and contains five items corresponding to frequently used multimedia categories: Music, Photos, YouTube, Videos and Phone. More categories could be added if needed as the root menu can contain up to eight items. Second-level submenus contain three to six subcategories depending on the selected parent item. For instance, the Music category (Figure 2) is divided into: Albums, Music Styles, Songs, Podcasts, Artists, Playlists and Albums.



Songs

P

Piste 01
Les copains d'abord - Georges Brassens

Piste 02
Les copains d'abord - Georges Brassens

Piste 03
Les copains d'abord - Georges Brassens

Piste 04
Les copains d'abord - Georges Brassens

Piste 05
Les copains d'abord - Georges Brassens

Piste 06
Les copains d'abord - Georges Brassens

Piste 07
Les copains d'abord - Georges Brassens

Piste 08

Figure 3: Long lists management: the linear list appears in the center of the Wavelet menu and is surrounded by its parent menus.

The Wavelet menu is not only intended to manage categories but also lists of multimedia data. These lists (e.g. Songs) can be quite long and may contain several thousand items. So much data could obviously not fit in a circular layout because sectors would then be ridiculously thin and thus impossible to interact with. As a solution, the Wavelet menu combines two different possible representations: items are either laid out in a circular or linear way according to the number of items that must be displayed. According to the idea of "wave propagation", the linear representation appears in the center of the Wavelet menu, while parent menus appear around it (Figure 3). The representation (and the way to interact with it) thus remain coherent with the original Wave menu design. Finally, once the linear list is displayed, the user can scroll it by performing drag and swipe gestures, as they would do when using standard built-in iPhone lists.

The Wavelet menu design may seem somewhat unconventional at first glance as the children of the menu tree appears in the center of the graphical representation while parents are shifted on the outmost rings. This can make it difficult for users to understand how this new technique works and thus jeopardize its acceptance. Prior testing with the former Wave menu design showed that some users could actually be somewhat disoriented when using it for the first time (although they were quite efficient in using it after sufficient training [[1]).

In order to solve this problem, we adopted a *stacking metaphor* when designing the Wavelet menu. The idea is that all submenus are stacked one upon another, the root menu being on the top, and the deepest menu on the bottom of the stack. According to this metaphor, when a parent menu is enlarged and moves outwards (as explained above), it permits the submenu that was "hidden" appear beneath it.

Of course, this is only a mind's eye as several submenus actually correspond to a parent menu, depending on which item is selected. However, this metaphor seems to work quite well according to our informal observations. In the current Wavelet menu implementation this metaphor is enforced by letting the submenu appear progressively when its parents move outwards (Figure 2). A shading effect also emphasizes this effect.

3.3 Exploration and Navigation

As stated above, the deepest submenu that is currently displayed is always located at the center of the screen. This is a key property as this submenu is likely to be the main focus of interest of the user. This design hence avoids the user having to perform unnecessary pointing movements to reach submenus. It also fits very well with the expert mode where strokes are typically superimposed over one another.

However, an even more important feature of Wavelet menus is that they can still work when limited screen real estate is available. As the user only needs to interact with the inmost menu (and the center area, to close submenus), s/he can still use the menu system when parent menus are partially (or even completely) hidden because they lie outside the borders of the screen. This is a major difference with hierarchical Marking menus, which can hardly cope with such a problem. Wavelet menus are thus especially well suited for (very) small sized devices.

Finally, when sufficient space is available, Wavelet menus also provides the advantage of previsualization. The user can then make a continuous circular gesture to browse all submenus as shown in Figure 4. Submenus are automatically displayed when the stylus (or the finger) hovers over the corresponding item in the parent menu. This makes it possible to browse the menu system in

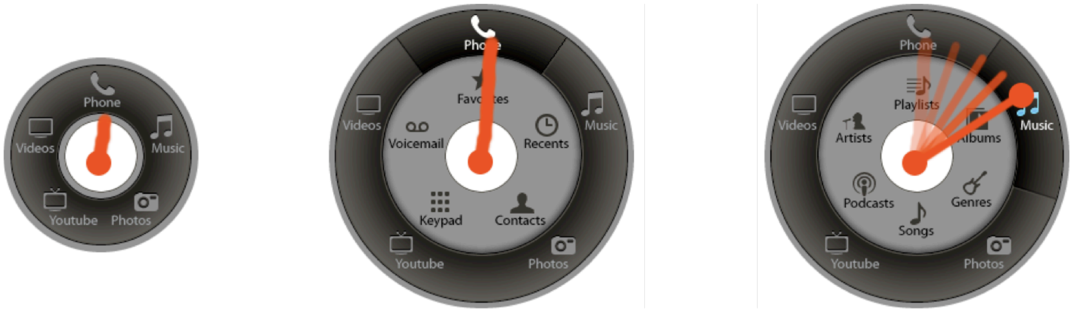


Figure 4: Exploration in Wavelet menus: the user can browse all submenus by performing a continuous circular gesture.

quite an efficient and natural way. This is a major advantage as compared with Multi-Stroke menus that force users to perform "blind" up and down transitions in the menu tree.

Wavelet menus thus provide a good trade-off between efficiency in novice mode (when enough space is available) and adaptability to hardware constraints (when limited space is available).

3.4 Other properties

Wavelet menus bring several other interesting properties (deriving from those of Wave and Marking menus [6]) that are well suited for mobile interaction.

Accuracy. Linear menu items tend to be too small to be accurately selected with fingers, especially when the thumb is used. In contrast, Wavelet menu gestures offer an efficient alternative for increasing accuracy on mobile devices [11].

Eyes-free selection. There are many cases where eyes-free selection is necessary when using mobile devices [14]. This can be because of: divided attention (e.g. when the user is moving); the absence of a dedicated display (e.g. the iPod shuffle); improper luminosity conditions that makes it hard to see the display; limited viewing accuracy because of age or disabilities. As selection is mainly based on orientation when using Wavelet menus, eyes-free selection is thus possible in most cases.

Learnability. Learning and memorization are important criteria for interaction techniques acceptance. Wavelet menus use simple gestures that heavily rely on spatial and procedural memory. Users perform the same gestures in novice and expert modes, a feature that favors a fluid transition from novice to expert usage [2][6].

3.5 Implementation

The Wavelet menu has been implemented in Objective-C using the iPhone Cocoa API. We chose this platform for several reasons. First, mobile interaction provides a unique opportunity to propose new paradigms and interaction techniques, contrary to PC desktop platforms that have hardly evolved in 25 years (and where users' habits are not prone to change [3][12]). Second, the iPhone is currently the most advanced mobile platform and it is now leading industry trends in this domain.

4. CONCLUSION AND FURTHER WORK

Wavelet menus are a new interaction technique, inspired from Wave menus, that improve exploration and navigation tasks on mobile devices. Wavelet menus are especially well adapted to the constraints of mobile devices. Their inverted-concentric layout makes it possible to navigate in the menu hierarchy even when little screen space is available. They also provide previsualization in an efficient way to browse submenus. Wavelet menus also make eyes-free selection possible, a feature that is often useful in mobile situation. Finally Wavelet menus both provide circular and linear representations, the latter being used when many items need to be displayed.

Future work will first focus on performing a comparative experiment between the existing iPhone multimedia manager and our adaptation of the Wavelet menu. We also want to evaluate the ability of users to perform eyes-free selection when using the Wavelet menu (a feature that is not allowed by standard iPhone menu). Finally, we plan to explore how multi-touch capabilities could improve the efficiency of the Wavelet menu.

5. REFERENCES

[1] Bailly, G., Lecolinet, E., and Nigay, L. 2007. Wave Menus: Improving the Novice Mode of Hierarchical Marking menus, INTERACT'07. Springer, 475-488.

[2] Bailly, G., Lecolinet, E., and Nigay, L. 2008. Flower menus: a new type of marking menu with large menu breadth, within groups and efficient expert mode memorization. ACM AVI '08, 15-22.

[3] Beaudouin-Lafon, M. 2004. Designing interaction, not interfaces. ACM AVI '04, 15-22.

[4] Bederson, B. B. and Hollan, J. D. 1995. Pad++: a zoomable graphical interface system. ACM CHI'95, 23-24.

[5] Huot, S., Lecolinet, E. 2007. ArchMenu et ThumbMenu : Contrôler son dispositif mobile "sur le pouce". *ACM IHM'07*.

[6] Kurtenbach, G. and Buxton, W. 1994. User learning and performance with marking menus. ACM CHI '94, 258-264.

[7] Parhi, P., Karlson, A., Bederson, B. 2006. Target Size Study for One-Handed Thumb Use on Small Touchscreen Devices. *MobileHCI'06*, 203-210

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

[9] Roudaut, A., Huot, S., and Lecolinet, E. 2008. TapTap and MagStick: improving one-handed target acquisition on small touch-screens. ACM AVI '08, 146-153.

[10] Roudaut, A., Lecolinet, E., and Guiard, Y. 2009. MicroRolls: expanding touch-screen input vocabulary by distinguishing rolls vs. slides of the thumb. ACM CHI '09, 927-936.

[11] Yatani, K., Partridge, K., Bern, M., Newman, M. W. 2008. Escape: a target selection technique using visually-cued gestures. ACM CHI'08, pp. 285-294

[12] Zhai, S., Kristensson, P., Gong, P., Greiner, M., Peng, S. A., Liu, L. M., and Dunnigan, A. 2009. Shapewriter on the iPhone: from the laboratory to the real world. Extended Abstracts of CHI'09, 2667-2670.

[13] Zhao, S. and Balakrishnan, R. 2004. Simple vs. compound mark hierarchical marking menus. ACM UIST '04, 33-42.

[14] Zhao, S., Dragicevic, P., Chignell, M., Balakrishnan, R., and Baudisch, P. 2007. Earpod: eyes-free menu selection using touch input and reactive audio feedback. ACM CHI '07, 1395-1404.