Figure 1: *X-O Arch Menu* used in MobiKat [3], an application for disaster management

X-O Arch Menu: Combining Precise Positioning with Efficient Menu Selection on Touch Devices

Felix Thalmann
TU Dresden,
Interactive Media Lab
Nöthnitzer Straße 46
Dresden, 01062 Germany
felix.thalmann@tu-dresden.de

Marcel Heckel
Fraunhofer Institute for
Transportation and
Infrastructure Systems
IVI
Zeunerstraße 38
Dresden, 01069 Germany
marcel.heckel@ivi.fraunhofer.de

Ulrich von Zadow
TU Dresden,
Interactive Media Lab
Nöthnitzer Straße 46
Dresden, 01062 Germany
uzadow@acm.org

Raimund Dachselt
TU Dresden,
Interactive Media Lab
Nöthnitzer Straße 46
Dresden, 01062 Germany
dachselt@acm.org

ITS 2014, November 16–19, 2014, Dresden, Germany.
ACM 978-1-4503-2587-5/14/11.
http://dx.doi.org/10.1145/2669485.2669539

Abstract

Support for precise positioning is cruical for many touch applications, and an efficient way to select the action at that point is very desirable in many cases as well. We draw upon existing work in the area of touch accuracy and touch menus to contribute the *X-O Arch Menu*. Our menu seamlessly combines precise positioning and fast, hierarcial menu selection. Furthermore, we introduce a novel optimization to pie menus that allows usage in limited screen space. The menu is fully implemented; we have created a touch-enabled version of a commercially available application using it.

Author Keywords

mobile devices, touchscreens, interaction techniques, occlusion, precise target acquisition, menu techniques

ACM Classification Keywords

H.5.2 [User Interfaces]: Input Devices and Strategies, Interaction Styles.

Introduction

Pointing at a position and then giving additional information about the object of interest is a common form of person-to-person communication. We present an innovative touch menu, the *X-O Arch Menu* (see Figure 1), that captures this principle, seamlessly integrating positioning

and the selection of pertinent information. For instance, the technique can be used wherever precise positioning of hierarcially organized content is needed (e.g., the placement of symbols on maps).

Accurate touch positioning can always be achieved by zooming an area of interest (e.g., Bederson and Hollan's PAD++ [1]). Integrating fingerprint recognition [6] and displaying the area below the touch in an unoccluded area (Shift by Vogel and Baudisch [10]) are two further possibilities. Finally, Take-Off by Sears and Shneiderman [9] shows a crosshair at an offset to the touch position. The crosshair is shown when a touch occurs and can be dragged by moving the finger; the position is fixed when the finger is removed.

Research in the area of efficient touch menus includes Francone et al.'s Wavelet Menus [4], which expand upon the initial Marking Menus by Kurtenbach et al. [8]. Hesselmann et al. [5] provides an occclusion-aware extension for tabletops, and Kin et al. [7] adapt marking menus for two-handed use.

Drawing upon this research, we combined the Take-Off technique for precise selection with a circular menu that handles occlusion and supports arbitrary hierarchies in restricted screen space. We support the undo of intermediate steps and deliver constant feedback as well as supporting novice and expert modes. The basic steps can be seen in figure 1: The *X-O Control* in the top image can be dragged. Upon release, a circular menu opens (middle image). Menu selection is performed either by tapping on the items (novice mode) or by performing flick gestures towards the desired items (expert mode, bottom image). In the case of hierarchial menus, all levels of the hierarchy are displayed at the same position, with feedback for previous selections provided at the crosshair position.

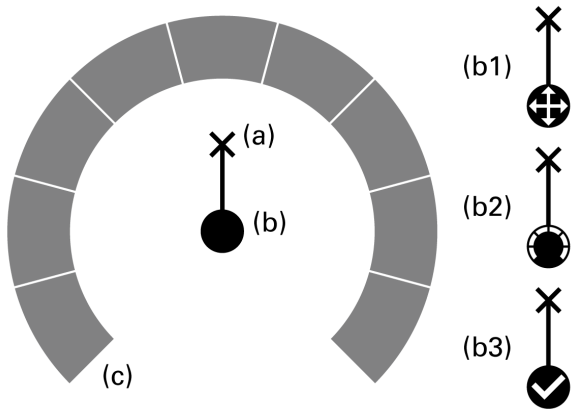

Figure 2: The elements of the *X-O Arch Menu*: The *X-O Control* consists of the (a) *X Pointer* and the (b) *O Handle*, the surrounding pie menu is the (c) *Arch*, (b1) *Move Cross* (b2) *Choice Mark* (b3) *Check Mark*.

We implemented the full *X-O Arch Menu* including novice and expert mode and integrated it into a preexisting application for disaster management, MobiKat [3]. Development proceeded in an iterative fashion, integrating feedback from HCI and domain experts as well as potential users.

In the remainder of the paper, we present the full menu design including static structure and possible interactions. A walkthrough of the integration into MobiKat follows. We then discuss the design decisions and alternative options we considered before closing with an outlook to future work.

Menu Design

In this section, we describe the menu design in more detail and present the structure and graphical elements used

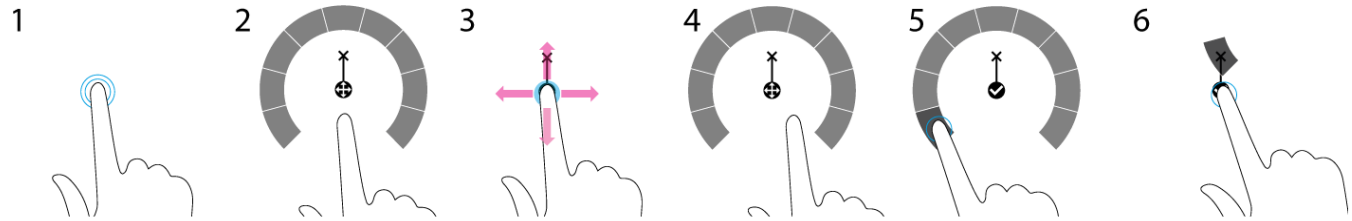

Figure 3: Novice workflow of the *X-O Arch Menu*: (1) doubletap to open, (2) menu is displayed, (3) precise positioning, (4) release, (5) tap on menu item to select, (6) preview on *X* is shown, tap on *O* to confirm.

as well as the dynamics of interaction and the supported workflow.

Elements
The menu consists of two main elements (see Figure 2): The central *X-O Control* that is used primarily for positioning and the *Arch* (c) that contains the menu items. In turn, the X-O control consists of the *O Handle* (b) for dragging and the *X Pointer* (a) that visualizes the selected position.

The O Handle The O Handle can be dragged around to specify the location. In addition, it functions as confirm button and displays the state of the menu. In the standard state (b1), it shows a drag icon. During stroke gestures (expert mode), it shows a *Choice Mark* (b2). Additionally, in novice mode, a *Check Mark* icon signifies the handle's function as a confirm button.

The X Pointer The X Pointer consists of a small semi-transparent dot with an "X" symbol in its center and is used for accurate positioning. After the position has been set, the area is used to show the already selected items in the menu hierarchy and appears like a history stack (See Figure 4). Tapping on the X Pointer reverts the last menu selection, while tapping on it when nothing is selected closes the menu.

The Arch This main element contains the actual menu items in an arch-like arrangement. It is visible when the position has been set; if the X-O Control is dragged, the Arch becomes semitransparent to signify that it is inactive.

Interaction and Workflow
The X-O Arch Menu supports two usage modes: novice and expert. Switching between the modes is seamless and possible at any point during the interaction.

Novice Mode In novice mode, interaction with the menu consists of simple taps and drags. Figure 3 shows a prototypical interaction sequence. The menu is opened using a doubletap (1). At this point, the X-O Arch menu is displayed. If the position is not accurate, the user can drag the O Handle to apply modifications. Following this, the user simply taps on the menu items. Each tap moves the selected item to a stack at the X Pointer position and opens a new level of hierarchy (in place of the last one) if available. At any point, a tap on the X Pointer reverses the last selection; tapping on the O Handle confirms the choice and closes the menu.

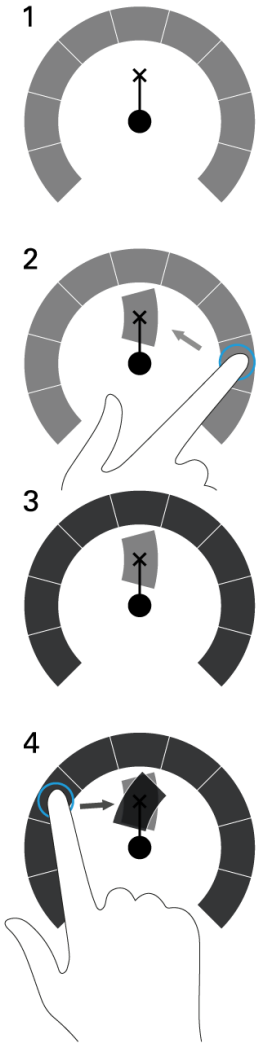

Figure 4: The history stack: As menu items are selected, the chosen items are placed at the X Pointer position.

Expert Mode Expert users can use additional gestures, enabling significantly faster, more seamless interaction (Figure 7 shows advanced usage in the context of the developed application). When opening the menu, the last UP of the doubletap can be omitted. The resulting *tap and a half* gesture puts the menu in positioning mode immediately, in which it can be dragged to the precise position intended. This can be followed by a *reverse tap* (UP-DOWN event) that puts the menu in a marking menu-like mode. In this mode, items are selected by swiping to the element and back. As in novice mode, selecting an item opens up a new hierarchy level in place of the last one. In this way, a series of swipes – performed without lifting the finger – can select an item deep in the hierarchy. Lifting the finger on the O Handle accepts and closes the menu, while lifting the finger at other points enables novice mode controls.

Note that expert mode swipes are enabled at any time, even after selecting one or more menu items in novice mode.

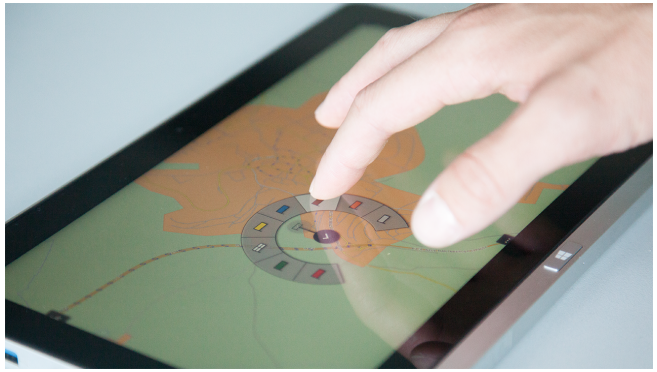

Figure 5: X-O Arch Menu in use in MobiKat.

Application Sample: Tactical Signs

We have integrated X-O Arch Menu into a mobile version of the disaster management software MobiKat [3] (see Figure 5). MobiKat is used in emergency situations to coordinate relief forces, where fast and effective decisions are important. The application includes a situation map. Users can place tactical signs that denote relevant actors on this map. As can be seen in Figure 6, these signs are composed of elements that can be assembled hierarchially. An X-O Arch Menu is used to place them on the map quickly and accurately. In this case, the menu hierarchy corresponds to the hierarchy of available tactical signs, and the history stack builds to gradually show the sign the user intends.

The original MobiKat is a traditional desktop interface implemented in Java using JavaFX. The mobile version was developed in cooperation with potential users and runs on a Samsung Series 7 Slate tablet (11.6" display, 1366x768 resolution).

Figure 6: Tactial sign example: (1) Basis, (2) organization, (3) size, (4) mission, (5) details.

Discussion

In integrating precise positioning and complex menus, we made a series of design decisions based on feedback by novice and expert users as well as HCI professionals. We describe them here.

As mentioned in the introduction, one design goal was occlusion avoidance. At no point are menu items occluded in X-O Arch Menu. When the finger is touching

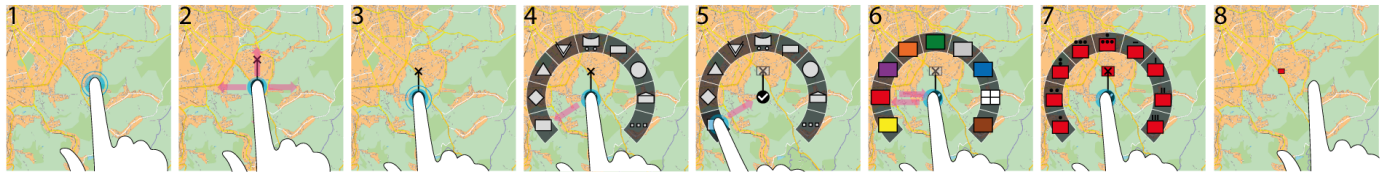

Figure 7: *X-O Arch Menu* when used by an expert in MobiKat: (1) *Tap and a half* to open X-O, (2) *drag* to define position, (3) *reverse tap* on O to open Arch, (4) *swipe* to item, (5) *swipe* from item back to O to confirm selection and go to next level (also exploring by dragging over entries is possible), (6) repeat *swipes* at second level, (7) display next level, (8) lift the finger on O to accept

the O, the Arch leaves room for the hand. Neither novice mode taps nor expert mode swipes cause significant occlusion. The design is similar to Brandl et al.'s occlusion-aware menus [2], with the difference that we do not distinguish between left- and right-handed users.

The choice of the appropriate technology to use for accurate touch positioning was difficult. While using fingerprint recognition [6] would have been exciting, the technology is not available on standard tablets. Also, while it allows more precise positioning, the technology does not remove occlusion. Shift [10] and Take-Off [9] both provide effective methods for accurate touch positioning. We integrated Take-Off because it is very effective, easy to learn and easy to integrate into a menu. Still, a comparison with Shift would be very interesting.

In adapting Wavelet [4] menus to a smaller screen size, we decided to only show one menu level at a time and switch to back-and-forth swipes for selection. This is a trade-off, since it does result in longer touch paths. Still, the space savings are significant, making positions closer to the screen edge are available for selection. Contrary to most menu designs, the space needed stays constant for hierarchy depths. Furthermore, occlusion is eliminated.

To provide feedback about the selected elements of the hierarchy, we use the stack (see Figure 4).

We expect usage in expert mode to have similar characteristics to Wavelet menu usage. Since positions of menu items are fixed relative to the center, users should be able to leverage motor memory for fast and ultimately eyes-free operation of the menu. Compared to Wavelet menus, required paths are somewhat longer, but we expect no qualitative difference. Furthermore, the ability to switch between novice and expert mode at any time should make transition easier and provide a smooth learning curve for the menu.

Pointing close to the borders of the screen is not possible using X-O Arch Menus. This is not a restriction in the case of a pannable background such as a map, but it might be an issue in other use cases. Finally, we use a doubletap to open the menu. This is not a fixed characteristic and has no implication on menu operation. Integration into concrete applications might make other gestures (e.g., Tap, Long Tap) that are not used by the rest of the interface more appropriate.

Conclusion and Future Work

We have presented a novel touch menu technique that allows users to annotate precise positions in large information spaces. The X-O Arch Menu provides a fluid workflow for novices and advanced users. We combined existing approaches in the realms of precise touch interfaces and touch menus to adress the challenges of this area.

Furthermore, we have implemented the design and integrated the menu into an existing large application. At the current point in time, there has been no formal testing of the menu. Qualitative studies of usage in the wild as well as quantitative, comparative studies to verify efficiency are needed. Also, studies could be used to compare the different design alternatives – for instance, to test if integrating Take-Off or Shift is better in this context and to optimize menu parameters such as angles, sizes and distances.

Acknowledgements

This work was supported by the Fraunhofer Institute for Transportation and Infrastructure Systems IVI Dresden. We would like to thank them for supplying initial source code and map materials.

References

[1] Bederson, B. B., and Hollan, J. D. Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. In *Proc. UIST'94*, ACM (1994), 17–26.

[2] Brandl, P., Leitner, J., Seifried, T., Haller, M., Doray, B., and To, P. Occlusion-Aware Menu Design for Digital Tabletops. In *Proc. CHI'09*, ACM (2009), 3223–3228.

[3] Danowski, K. MobiKat-Integriertes System zur Unterstützung der Katastrophenbewältigung und der alltäglichen Gefahrenabwehr. *Fraunhofer-Institut fur Verkehrs-und Infrastruktursysteme IVI* (2007).

[4] Francone, J., Bailly, G., Nigay, L., and Lecolinet, E. Wavelet Menus: A Stacking Metaphor for Adapting Marking Menus to Mobile Devices. In *Proc. MobileHCI'09*, ACM (2009), 49.

[5] Hesselmann, T., Flöring, S., and Schmitt, M. Stacked Half-Pie Menus: Navigating Nested Menus on Interactive Tabletops. In *Proc. ITS'09*, ACM (2009), 173–180.

[6] Holz, C., and Baudisch, P. The Generalized Perceived Input Point Model and How to Double Touch Accuracy by Extracting Fingerprints. In *Proc. CHI'10*, ACM (2010), 581–590.

[7] Kin, K., Hartmann, B., and Agrawala, M. Two-Handed Marking Menus for Multitouch Devices. *TOCHI'11 18*, 3 (2011), 16.

[8] Kurtenbach, G., and Buxton, W. The Limits of Expert Performance Using Hierarchic Marking Menus. In *Proc. INTERACT'93 and CHI'93*, ACM (1993), 482–487.

[9] Sears, A., and Shneiderman, B. High Precision Touchscreens: Design Strategies and Comparisons with a Mouse. *Int. J. Man-Mach. Stud. 34*, 4 (Apr. 1991), 593–613.

[10] Vogel, D., and Baudisch, P. Shift: A Technique for Operating Pen-Based Interfaces using Touch. In *Proc. CHI'07*, ACM (2007), 657–666.