Finger-Count & Radial-Stroke Shortcuts: Two Techniques for Augmenting Linear Menus on Multi-Touch Surfaces

Gilles Bailly **Eric Lecolinet** **Yves Guiard**
Telecom ParisTech – CNRS LTCI
46 rue Barrault, 75013, Paris, France
{gilles.bailly; eric.lecolinet; yves.guiard}@telecom-paristech.fr

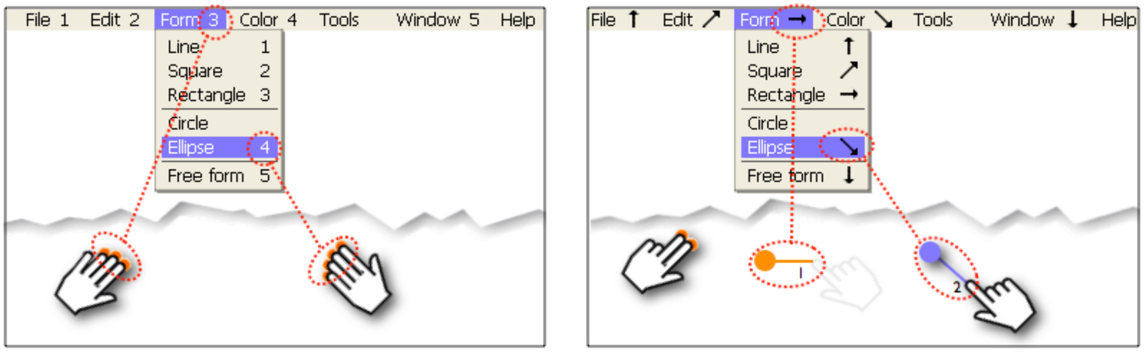

Figure 1: Left: Finger-Count Shortcuts: the non-dominant hand (NDH) selects the pulldown menu in the menubar while the dominant hand (DH) selects the item. Favorite Menus (resp. items) are selected according to the number of fingers of the NDH (resp. DH) that are pressed on the surface. Right: Radial-Stroke Shortcuts: at least two fingers of the NDH are pressed to activate the menubar mode while one finger of the DH performs a multi-stroke radial gesture.

ABSTRACT
We propose Radial-Stroke and Finger-Count Shortcuts, two techniques aimed at augmenting the menubar on multi-touch surfaces. We designed these multi-finger two-handed interaction techniques in an attempt to overcome the limitations of direct pointing on interactive surfaces, while maintaining compatibility with traditional interaction techniques. While Radial-Stroke Shortcuts exploit the well-known advantages of Radial Strokes, Finger-Count Shortcuts exploit multi-touch by simply counting the number of fingers of each hand in contact with the surface. We report the results of an experimental evaluation of our technique, focusing on expert-mode performance. Finger-Count Shortcuts outperformed Radial-Stroke Shortcuts in terms of both easiness of learning and performance speed.

Author Keywords
Menu techniques, multi-touch, multi-finger interaction, two-handed interaction.

ACM Classification Keywords
H.5.2 **[Information Interfaces and Presentation]**: User Interfaces– Evaluation/methodology, Interaction styles.

General Terms
Human factors

INTRODUCTION
Menus are widespread in current applications and most of them are hierarchical, since the number of commands is continuously increasing [2]. Most menu techniques have been originally designed for personal computers. But beside our laptops, we also have today at our disposal new sorts of computing devices based on interactive surfaces such as smartphones, tablet PCs, tabletops or wall displays. Thus improving the usability of classic menus in this new technological context is a real issue. In this paper we focus on the augmentation of traditional menubars and pull-down menus, which are still massively present in graphical interfaces, in the case of interactive multi-touch surfaces.

One important advantage of pointing-activated menus is that they are quite easy to understand and to use, notably because direct manipulation ensures an excellent level of stimulus-response compatibility [6]: the 'response' to the display does not require of the user any arbitrary encoding operation. However, when it comes to interactive surfaces offered to direct finger contact, these menus suffer a number of serious drawbacks:

- D1: *Occlusion*. The hand and the fingers may hide parts of the menu display.
- D2: *Accuracy*. The large surface area of finger-screen contact may induce item selection errors.

- D3: *Lack of shortcuts*. In the absence of a keyboard, the expert mode of Linear menus based on keyboard shortcuts are unavailable.

In addition, for large interactive surfaces such as tabletops:

- D4: *Reachability*. The length of the human arm being what it is, the menubar may be difficult to reach.
- D5: *Groupware*. Collaborative work often requires the user identity to be known. So long as a menu technique relies on pointing in the absence of specific technology, this information is missing [5].

Below we introduce two selection techniques aimed at alleviating the abovementioned shortcomings. Finger-Count shortcuts and Radial-Stroke shortcuts exploit the multi-touch capabilities to facilitate access to favorite items, thus augmenting linear menus (the user can still point to menu items as usual).

RELATED WORK

As stated in [11], multi-touch technologies open the way for fairly natural interactions where users can use both hands and several fingers. Multi-touch techniques have helped to improve menus [4, 9, 13, 16] but none of them overcomes all the above-listed limitations and some conflict with common multi-touch techniques like Pitch&Expand for zooming [13,16].

Several menu techniques have been designed in light of the kinematic chain theory [8] to improve two-handed interaction [3,7], but they have no expert mode and provide a limited number of commands.

Marking menus [9,16] provide an expert mode based on gestural interaction. However, using them in a commercial application would require to deeply change its GUI, and this is probably why they are seldom used. Besides, they essentially amount to context menus and do not fit well with the popular menubar model. A recent study proposed to augment the traditional menubar by using stroke shortcuts as a substitute of keyboard shortcuts [1]. The user must however still interact with the menubar in novice mode, with the above-mentioned drawbacks. Finally, a common problem with this technique and Marking menu on interactive surfaces is that they use drag events. They may thus conflict with other interaction techniques such as text selection, drag and drop, etc. Thus they can be performed only on specific zones of the GUI or otherwise require recourse to interaction states (such as modifier keys or mouse buttons).

FINGER-COUNT SHORTCUTS

This technique relies on the human ability to code numbers with fingers. Rather than exhibiting fingers, users will actually put *N* fingertips in contact with the interactive surface, the system being programmed to simply count the contacts to determine *N*. With two hands a user can specify 5x5=25 favorite items in a classical 2-level hierarchical menu bar.

Each *N*-finger touch with the non-dominant hand is associated with a menu of the menubar, users being reminded of the correspondence by digits displayed next to the corresponding items (Fig. 1). Likewise, the dominant hand is associated to an item in the currently selected menu. So, the user simply selects an item by putting *N* fingers with each hand in contact with the interactive surface. The corresponding command will be activated when the user lifts all his fingers. As fingers cannot be lifted up at exactly the same time, simultaneousness is defined with some time tolerance. The technique also makes it possible to quickly explore the different menus just by adding or removing the appropriate number of fingers of the NDH.

The Finger-Count shortcuts technique does not conflict with basic drag gestures (e.g., panning) simply because it ignores single-finger contacts. Moreover, the traditional two-finger Zoom/Rotate command, which requires one finger of each hand, is integrated in our technique simply by having this command correspond to the first item of the first menu. We have also implemented the merging of command selection with direct manipulation [12], meaning that users may proceed directly with the same gesture from command selection to direct manipulation. For example, not only is the item requiring one finger of each hand reserved for the Zoom/Rotate command, but once this command is selected the two fingers that have been pressed may start their zooming and rotation task straightaway. In fact the Zoom/Rotate command is generally not self-revealing in common applications, and so the Finger-Count shortcuts technique offers an opportunity to make this command explicit and to control it in a fluid manner.

As the user can perform Finger-Count shortcuts away from the menu display, the occlusion (D1), accuracy (D2) and reachability (D4) concerns vanish. Since users need not look at their hands nor at the GUI elements, Finger-Count shortcuts are eyes-free and can be performed ballistically.

Most importantly, Finger-Count shortcuts have an expert mode: an expert user may perform the desired selection without waiting for the menu display. The expert mode allows two-handed parallelism, with both hands executing their shortcuts concurrently, no matter the exact order of operations. In fact the system waits 300ms to evaluate the produced trace, and so the activation of the command can be chunked.

Some multi-touch surfaces like tabletops support multiple-user interaction. If Finger-Count shortcuts obviously cannot solve all the complex problems involved in groupware, they can help. For example, if the interactive surface is partitioned into different areas, one for each hand of each static participant (as in our implementation), multiple-user two-handed interaction is possible without the risk of having the system confuse users (D5).

While Finger-Count shortcuts extend linear menus, they do not constrain the total number of commands. Only the number of favorites that can have a Finger-Count shortcut is limited to 25. That looks like a sufficient number in practice, recalling that most users only use/know rather few keyboard shortcuts. However, this number can be raised to 64 with Radial-Stroke shortcuts (and even more if both techniques are used).

RADIAL-STROKE SHORTCUTS

Radial-Stroke shortcuts augment the traditional menubar by using strokes as proposed in [1], though in a different way. The users express their choice within the menubar and the selected menu by performing successively, at some convenient location on the interactive surface, two short linear strokes with certain orientations. The first stroke selects the menu, the second selects the item in the menu. For each stroke the choice is among eight directions (the four cardinal directions plus the four diagonals), the correspondence being recalled to users by arrows displayed next to the items (Fig.1). Hence, two short strokes executed successively suffice to specify one from a set of 8x8=64 favorite commands. This double gesture is similar to the multi-stroke radial gesture proposed in [17], the menu representation being different. In the case of a tabletop, orientations are interpreted relative to the table edge occupied by the user. Thus, from the user's viewpoint, a given gesture always corresponds to the same command. Another notable feature, of special interest for novice users, is that, thanks to the clockwise ordering of directions (see Fig. 1), a circular gesture makes it possible to explore the available menus.

In order to maintain compatibility with interaction techniques using one or two fingers (e.g., panning, zooming, or opening a context menu), the technique requires both hands. The NDH acts as a modifier that triggers the "menubar mode" so that the system will transfer events to the menubar instead of sending them to the GUI element located under the finger. More precisely, this mode is activated if the user presses at least two fingers with the NDH and one with the DH, the DH being used to execute the two stroke sequence described above. This scheme avoids collisions with common two-handed interaction techniques for zooming and rotating, while remaining fast and simple. We discarded the solution consisting of making two simultaneous strokes with the two hands, due both to ambiguity problems (e.g., possible confusion with the zoom/rotate command) and to the fact that it is difficult to independently control the directions of two simultaneous strokes.

Like Finger-Count shortcuts, Radial-Stroke shortcuts can be performed away from the display (D1, D2, D4) and by more than one user (D5). Radial-Stroke shortcuts provide an expert mode (D3) and eyes-free selection, like Marking menus. When the user performs rapid strokes, menus are not displayed (they appear after a delay of 300ms).

EXPERIMENT

Our goal was to compare the learning performance of the expert mode of our two novel techniques. We were more curious about learnability than absolute speed performance because two-handed finger counting is a pretty new principle and we wished to check how easily these shortcuts can be memorized and performed. We used the traditional technique of directly pointing at the menubar as a baseline.

Equipment and Menu configuration

The experiment was conducted on an Immersion Ilight multi-touch table [18] based on diffused-illumination technology, with a 72x96cm display. The menubar contained five 5-item menus. Item size (12-point font size, 1.6cm in height on the projected screen, i.e., that of the Windows menubar on that platform) was relatively large. The distance from the rest position of the user's hands was about 36cm, meaning a relatively easy pointing task for the baseline technique—on large tables, not to mention wall-screen displays, the menubar may be located a lot farther from the user. The shortcut area occupied one quarter of the screen, simulating the case of three or four users provided with equal private surface areas.

Task and Procedure

Twelve participants were asked to activate as quickly and accurately as possible one out of six equifrequent commands in response to a visual stimulus (the command's name) displayed at the top of the screen. When asked to use our two shortcut techniques, they were encouraged to do their best to learn the expert mode, that is, to select as many items as possible before the submenus could appear. When not sure enough, they could wait 300ms for the shortcut reminders to appear in the menubar. Whatever the participant's strategy, trial completion time was measured from the time of appearance of the visual stimulus to the command selection, whether correct or wrong.

Design

For each technique, the participants had to complete four blocks, each composed of four series in which the six different commands were presented in a randomized order. Three equivalent sets of item names were used, counterbalanced across techniques, and the ordering of the three techniques was counterbalanced across subjects. In total the experiment involved 12 participants*3 techniques*4 blocks*24 trials = 3,456 selections.

Results

A repeated-measures analysis of variance (ANOVA) run with the two shortcut techniques and practice as factors showed a significant effect of the block factor ($F_{3,33}$ =76.0, p<.0001) on the number of correct selections in expert mode (Fig. 2). The explanation of this effect is twofold: not only did the participants use the expert mode more and more but they also made fewer and fewer errors. There was a significant technique*block interaction ($F_{1,3}$=5.3, p<.005) reflecting the fact that Finger-Count shortcuts was learned faster. While it is already known that stroke shortcuts are quite efficient for learning [1] thanks to spatial memory, this result is welcome, confirming that people also easily learn to express numbers with their fingers.

An ANOVA on mean selection time with now all three techniques considered showed that performance improved with practice ($F_{3,33}$=80.9 p<.0001, see Fig. 3). The technique effect was also significant ($F_{2,22}$ =15.3 p<.0001). For the last block the Finger Count (2.0s) and the menubar (2.1s) outperformed the Radial-Stroke (2.6s) (Tukey tests). Unsurprisingly, performance was initially faster for traditional menu clicking: not only was direct menubar pointing quite familiar, but the shortcut techniques required learning. The key finding is that selection time dropped monotonically for our two shortcut techniques (final selection time in expert mode was 1.8s for Finger-Count and 2.4s for Radial-Stroke), no such improvement being observed with the baseline technique (Fig.3). With full practice, one may conjecture our techniques should eventually outperform the traditional pointing technique.

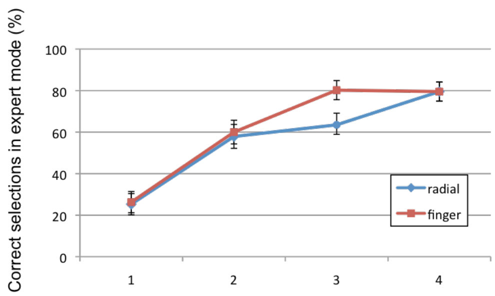

Figure 2: Mean correct selection in expert mode (%) by block number.

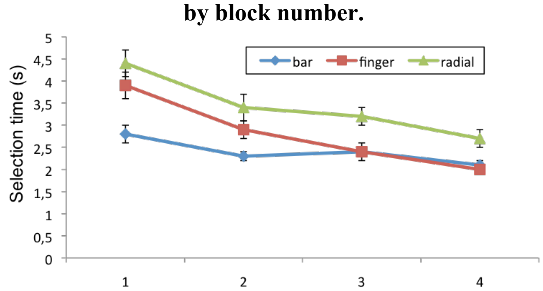

Figure 3: Mean selection time (s) by block number.

Turning to accuracy, an ANOVA indicated a significant effect for technique ($F_{2,22}$=3.76, p<.05): accuracy (Tukey tests) was higher overall for the menubar (97%) than for Radial-Strokes (90.2%) and Finger Count (91.8%). While in novice mode 93.8% of selections were correct, in expert mode similar accuracy obtained with Radial-Strokes (91.5%) and Finger-Counts (91.4%). Error rates reflecting both motor-control and memorization errors, accuracy should continue to decrease with more practice.

Finally, it is worth noticing that the experiment was favorable to direct menu selection because of the relatively low PPM (pixel per mm) display resolution of the tabletop, so that menu items were substantially larger than they would be on most recent interactive displays.

CONCLUSIONS

Shneiderman's [14] direct manipulation principle being less easily applicable when it comes to the menubar of large sensitive surfaces, alternative selection techniques are needed. One reason why those we introduced above seem worthy of consideration is that they rest on rather basic capabilities of humans: using one's fingers to code numbers and to indicate directions. Capitalizing on the present results, we plan to further explore this direction. Hybrid techniques that combine radial strokes and finger counting to increase the total number of shortcuts seem to have special promise.

REFERENCES

1. Appert, C. and Zhai, S. 2009. Using strokes as command shortcuts: cognitive benefits and toolkit support. Proc. CHI'09. ACM Press, 2289-2298.
2. Beaudouin-Lafon, M. 2004. Designing interaction, not interfaces. Proc. AVI '04, ACM Press, 15-22.
3. Bier, E. A., Stone, M. C., Pier, K., Buxton, W., and DeRose, T. D. 1993. Toolglass and magic lenses: the see-through interface. Proc. SIGGRAPH '93. ACM Press, 73-80.
4. Brandl, P., Leitner, J., Seifried, T., Haller, M., Doray, B., and To, P. 2009. Occlusion-aware menu design for digital tabletops. Proc. CHI'09, ACM Press, 3223-3228.
5. Dietz, P. and Leigh, D. 2001. DiamondTouch: a multi-user touch technology. Proc. UIST'01, ACM, 219-226.
6. Fitts, P. M., and Seeger, C. MS-R Compatibility: Spatial Characteristics of Stimulus and Response Codes. JEP, 1953, 46, 199-210.
7. Fitzmaurice, G., Baudel, T., Kurtenbach, G., and Buxton, B. 1997. A GUI paradigm using tablets, two-hands and transparency. Proc. CHI '97, ACM, 212-213.
8. Guiard, Y. 1987. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. Journal of Motor Behavior, 19(4), 486-517.
9. Koike, H., Xinlei, C., Nakanishi, Y., Oka, K., and Sato, Y. 2002. Two-handed drawing on augmented desk. Proc. CHI'02, ACM Press, 760-76.
10. Kurtenbach, G. and Buxton, W. 1991. Issues in combining marking and direct manipulation techniques. Proc. UIST '91, ACM Press, 137-144.
11. Nacenta, M. A., Baudisch, P., Benko, H., and Wilson, A. 2009. Separability of spatial manipulations in multi-touch interfaces. ACM GI'09, 175-182.
12. Pook, S., Lecolinet, E., Vaysseix, G., and Barillot, E. 2000. Control menus: excecution and control in a single interactor. Proc. CHI '00, ACM Press 263-264.
13. Shahzad Malik,S. 2007. An Exploration of Multi-finger Interaction on Multi-touch Surfaces. Ph.D. Thesis. University of Toronto.
14. Shneiderman, B. 1983. Direct Manipulation: A step beyond programming languages, IEEE Computer, 16(8), 57-69.
15. Vogel, D., Cudmore, M., Casiez, G., Balakrishnan, R., and Keliher, L. 2009. Hand occlusion with tablet-sized direct pen input. Proc. CHI'09. ACM Press, 557-566.
16. Wu, M. and Balakrishnan, R. 2003. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. Proc. UIST'03, ACM, 193-202.
17. Zhao, S. Balakrishnan, R. 2004. Simple vs. compound mark hierarchical marking menus. Proc UIST'04, ACM, 33-42.
18. Immersion 2009 http://www.immersion.fr/