SPad: A Bimanual Interaction Technique for Productivity Applications on Multi-touch Tablets

Cédric Foucault, Manfred Micaux, David Bonnet, Michel Beaudouin-Lafon

SPad: A Bimanual Interaction Technique for Productivity Applications on Multi-Touch Tablets

Cédric Foucault
Univ. Paris-Sud &
ENS Cachan
F-91405 Orsay, France
cedric.foucault@gmail.com

David Bonnet
Univ. Paris-Sud &
CNRS & Inria
F-91405 Orsay, France
bonnet@lri.fr

Manfred Micaux
EIT ICT Labs &
Univ. Paris-Sud
F-91405 Orsay, France
manfred@kth.se

Michel Beaudouin-Lafon
Univ. Paris-Sud &
CNRS & Inria
F-91405 Orsay, France
mbl@lri.fr

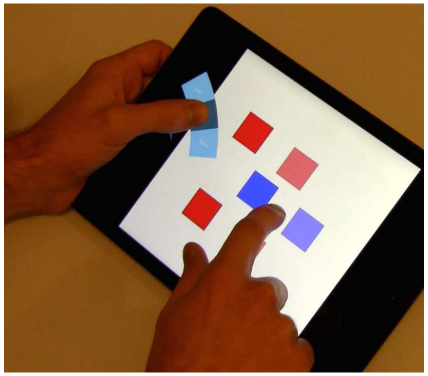


Figure 1: Copying multiple objects with SPad.

CHI 2014, April 26–May 1, 2014, Toronto, ON, Canada.
ACM 978-1-4503-2474-8/14/04. http://dx.doi.org/10.1145/2559206.2581277

Abstract
SPad is a new bimanual interaction technique designed to improve productivity on multi-touch tablets: the user activates quasimodes with the thumb of the non-dominant hand while holding the device with that hand and interacts with the content with the dominant hand. The paper describes the design of SPad and a tablet application that demonstrates how it enables faster, more direct and more powerful interaction without increasing complexity.

Author Keywords
SPad; bimanual interaction; tablet; multi-touch; thumb-based interaction; quasimode

ACM Classification Keywords
H.5.2 [Information interfaces and presentation (e.g., HCI)]: User Interfaces.

Introduction
Multi-touch tablets are now widely used for a growing number of tasks. However most users still do not see them as productivity tools: Recent surveys report that only 26% of tablet users find them effective for productivity[1], that they spend only 15% of their time on

[1]APQC survey: Should We Just Play Chopsticks? Personal Productivity in the Mobile Age http://bit.ly/1hc0L4l

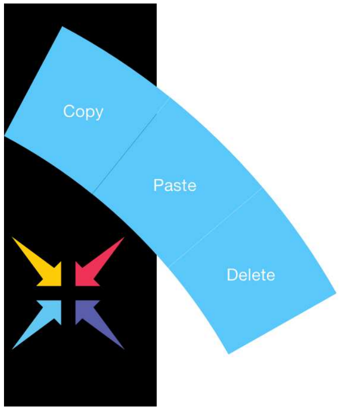


Figure 2: Final design of SPad.

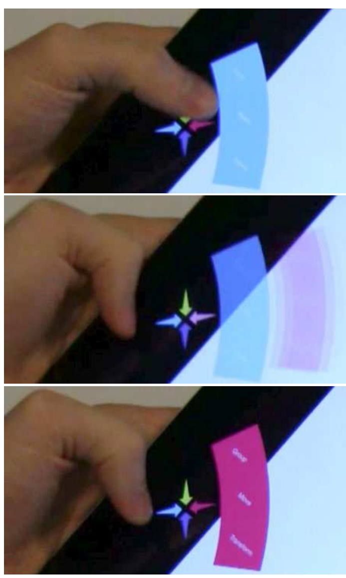


Figure 3: The user swipes to switch among button arches.

production activities[2] and that only 12% of students who use tablets create presentations or documents with them[3].

While simple gestures such as taps, drags and two-finger pinches make tablets suitable for browsing media-rich content, the more complex sequences of interactions required by productivity tasks make tablets less efficient than desktop computers. Indeed, content editing on tablets often use on-screen widgets such as contextual menus and sidebar buttons that are invoked by long taps or swipes. Compared to gestures, these widgets introduce a significant temporal offset that make them less direct according to Instrumental Interaction [2]. They also force users to input a sequence of disjointed editing commands rather than letting them seamlessly manipulate objects of interest as with gestural interaction. Extending the gesture vocabulary, however, has its own challenges: Users must be able to discover, learn and perform the new gestures, and the system must be able to recognize and disambiguate them.

The SPad technique introduced in this paper addresses these issues by combining two features: (i) a menu widget controlled by the thumb of the non-dominant hand holding the device, and (ii) bimanual use of this menu with touch-based manipulation gestures of the dominant hand for content editing. The menu gives access to quasimodes that control the effects of the finger touches of the dominant hand to allow faster, richer and more fluid interactions. After describing SPad and its design rationale, we present an example of its use in a graphical application along with a discussion on the benefits of SPad and its implications for gestural interaction.

[2]Gartner Survey Says Entertainment Accounts for Half of Device Screen Time: http://gtnr.it/KbWu2f
[3]Nielsen survey: A Computer in Every Classroom and a Tablet in Every Backpack? http://bit.ly/JomVka

Description of SPad

SPad is designed to be operated by the thumb of the non-dominant hand on the side of the tablet so that the user can support the device with the palm of this hand at the same time, in either portrait or landscape orientation. It consists of a *button arch* with three buttons and a *swipe zone* with four arrows (Fig. 2). Users have access to four different button arches by swiping along the direction of the arrows in the swipe zone (Fig. 3). Swiping towards the bezel hides SPad, swiping from the bezel shows it if it was hidden or repositions it vertically. Swiping from the right bezel moves SPad there, for left-handed users.

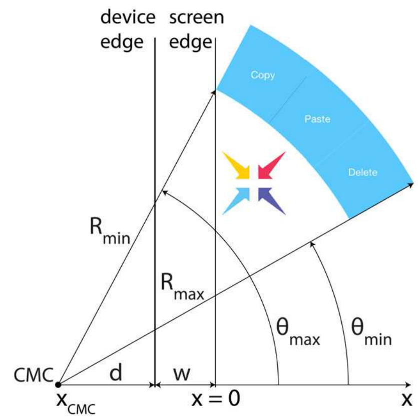
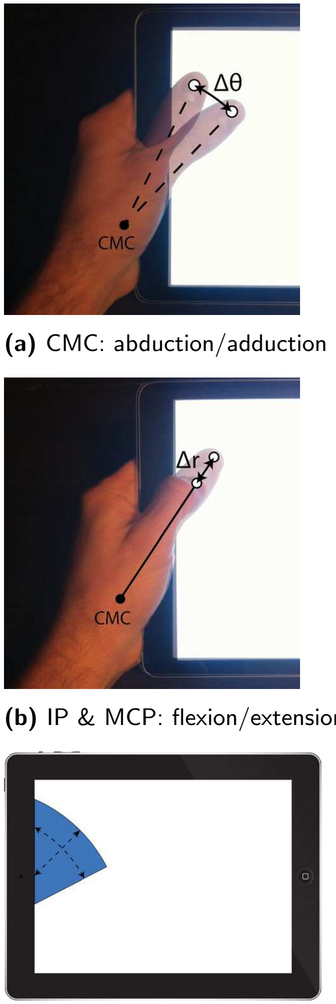


Figure 4: Design parameters of SPad.

The size and location of the arch and swipe zone are derived from the estimated location of the carpometacarpal (CMC) thumb joint (Fig. 4). We set $d = 21.1\,\mathrm{mm}$ and $\theta_{min} = 29.5°$ based on empirical measurements, $R_{min} = 90\,\mathrm{mm}$ and $R_{max} = 110\,\mathrm{mm}$ based on the average thumb length ($115\,\mathrm{mm}$ according to [12]). The bezel width w is set at runtime based on the device's type and orientation. The CMC's horizontal



(a) CMC: abduction/adduction

(b) IP & MCP: flexion/extension

(c) Resulting interaction area

Figure 5: Using the degrees of freedom of the thumb

coordinate is then computed as $x_{CMC} = -d - w$, and θ_{max} is deduced from the intersection of the inner circle (with radius R_{min}) and the edge of the screen. The vertical coordinate is set when the user performs a swipe from the bezel so that the topmost button of the arch is horizontally centered at the swipe location.

The Design of SPad

SPad is the result of an iterative design process: First, we created a paper prototype based on an analysis of the interaction possibilities of the non-dominant thumb. Then, a user study with the paper prototype led us to refine the design and to implement SPad in an iPad application. Finally, based on additional feedback, we further refined our design and created the final prototype.

Interacting with the Non-Dominant Thumb

Moving the thumb involves the Carpometacarpal (CMC), Metacarpophalangeal (MP) and Interphalangeal (IP) joints. When holding a tablet, they allow the contact point on the touchscreen to move along two degrees of freedom (Fig. 5): the angle θ from the CMC can change with CMC adduction/abduction, and its distance r from the CMC can change with MP & IP flexion/extension. We estimated the ranges of motion using average thumb sizes [12], resulting in the interaction area depicted in Fig. 5c, which SPad fully exploits.

User Study

Based on the above analysis, we created a paper prototype with a sheet attached to an Apple iPad 3 and UI objects with an adhesive on the back so we could easily move them around. We then conducted a user test with five right-handed participants (age 20 to 30), one of whom had previous tablet experience. The test involved a "computer" who performed the changes in the UI based on the user's actions and a "facilitator" who guided the user through the test. We used the think-aloud protocol and video-recorded the tests. This test led to the following findings and changes to the prototype:

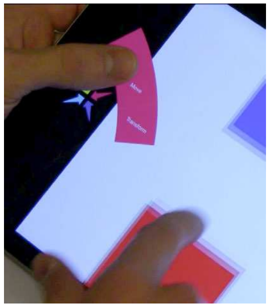
Teach the user to use both hands Without prior instruction, most participants tried unimanual interaction and needed a hint from the facilitator that they should use both hands. A real application should therefore include a tooltip the first time it is used, or if the user tries to repeatedly tap buttons without using the non-dominant hand.

Let the user move his non-dominant hand
Participants did not naturally position their non-dominant hand at the same location and also wanted to be able to move their hand. Thus, we allow users to reposition SPad with a bezel swipe [13].

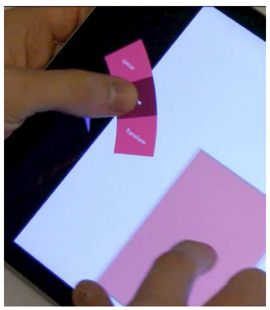
Use direct mapping instead of scrolling In the first prototype, the user swiped clockwise or counterclockwise in the swipe zone to scroll to a given button arch. While intuitive, this solution caused the participants to feel disoriented and it took a long time before reaching the desired arch. We replaced scrolling by mapping each of the four arches to a diagonal swipe. This way, each arch is accessible in one swipe and the mapping is easy to learn.

Final Design

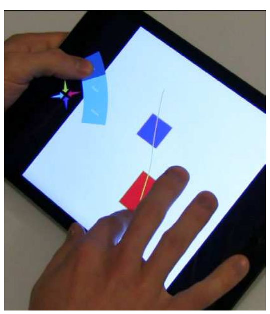
Additional feedback based on a prototype implementation of SPad on an iPad led us to color-code the arches and their corresponding arrow to reinforce learning and to add animated transitions in the direction of the swipe when switching among button arches to provide visual feedback. We also used bezel swipes [13] to show, hide and move SPad, saving screen real estate when not using it.



(a) Panning the canvas



(b) Moving a shape



(c) Drawing a selection line

Figure 6: Quasimodes with a one-finger drag

Sample Application of SPad

To demonstrate the potential of SPad in productivity applications, we developed a graphical editor for creating, copying, deleting, moving and resizing simple shapes on a canvas that can also be panned and zoomed[4]. We now describe how this application makes interaction faster and more direct than the state of the art and gives the user more power without increasing complexity.

Quasimodes and Polymorphic Gestures

By pressing a button on SPad, the non-dominant hand activates a *quasimode* that modifies the operations applied by the dominant hand. For example, when the user taps on shapes while the "Delete" quasimode is active, these shapes are deleted, but if "Copy" is active, they are copied to the pasteboard once the user releases the "Copy" button. Quasimodes enable the user to perform different types of operations, e.g. copying and deleting, with the same gesture, e.g. a tap. Note that we follow the principles of Guiard's kinematic chain model for bimanual interaction [5]: The non-dominant acts first and does not need high precision, while the dominant hand acts within the interaction context set by the non-dominant hand, and can perform precise gestures.

The interface uses this *gesture polymorphism* extensively: A one-finger drag gesture pans the canvas when no quasimode is active, moves a shape when "Move" is active, or draws a line that selects the shapes it crosses when "Copy" is active (Fig. 6); A pinch gesture zooms out when no quasimode is active or scales a shape when "Transform" is active; A two-finger drag gesture with the two fingers horizontally aligned moves a shape vertically when "Move" is active or resizes it vertically when "Transform" is active (Fig. 7). Moreover, operations can be applied to multiple objects by first selecting them, then performing the operation on a selected object. The user can select or deselect objects by tapping them or by drawing a selection line while a quasimode is active. The selection disappears once the quasimode is released[5].

Faster and More Direct Interaction

Applications such as Apple Keynote for iOS[6] use contextual menus and resize handles to copy, delete and resize objects. These widgets add a level of indirection since the user must interact with them instead of manipulating the objects directly. They also add a temporal offset since the user must first select objects then wait for the widget to appear. By constrast, in our application the user manipulates objects directly with the dominant hand: To delete a shape, the user taps on it and it disappears immediately; For pasting, the user touches the screen, the system pastes the shapes and the user can precisely adjust their location by moving the finger before releasing it (Fig. 1). To resize proportionally, the user pinches the shape without having to precisely follow a diagonal guideline as with Keynote's resize handles, since free resizing is accessible by a separate gesture.

Informal tests show the benefits of this approach. For example, as seen in the video, copying three shapes and pasting them at two different locations takes about 15.5s with Keynote vs. 3.5s with our application. We also found that the combination of quasi-modes with bimanual interaction reduced the risk of accidental activation.

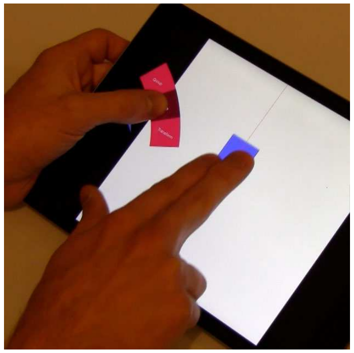
More Power without Increasing Complexity

Most previous work on gestural interfaces has assumed a one-to-one mapping between gestures and operations,

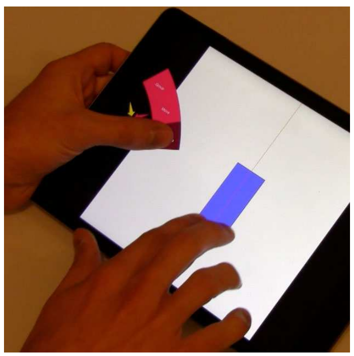
[4]See the video at http://vimeo.com/81943672

[5]Persistent selection should be handled by a separate grouping operation, but has not been implemented yet

[6]http://www.apple.com/ios/keynote



(a) Moving a shape vertically



(b) Resizing a shape vertically

Figure 7: Quasimodes when dragging two horizontally aligned fingers

requiring new gestures for additional operations. However, large gesture vocabularies cause usability problems: gestures are not discoverable, they have to be learned, and recognition errors are more likely [11]. Systems such as OctoPocus [1] and ShadowGuides [4] address these issues, but they still require learning.

Our approach is different: instead of introducing new gestures, we make gestures polymorphic. With SPad, one gesture can potentially trigger 13 different operations: one per quasimode plus one when no quasimode is active. With a typical set of five touch gestures (one- and two-finger taps and swipes, pinch), this gives access to 65 commands. We can therefore support a large set of operations with a small gesture vocabulary, making the interface more powerful without making it more complex.

Discussion and Related Work

SPad builds on Wagner et al.'s BiTouch and BiPad [14]. We use bimanual taps and gestures, but not bimanual chords because they give access to fewer items (eight three-finger combinations) and require holding the tablet on the forearm. We also found that the zones identified by Wagner et al. for one- and two-handed palm support needed to be refined to more accurately describe the capabilities of the thumb (Fig 5c). For users who prefer to work on a stand or a flat surface, SPad can still be used efficiently with the non-dominant index finger or thumb. Finally, Wagner et al. found that users often change position for comfort, which we address by supporting both orientations and by letting users reposition SPad.

SPad also builds on Hinckley et al.'s combination of pen and touch [6], where the object touched by the non-dominant hand can affect the action of the pen held in the dominant hand. The resulting interface does not use widgets, but requires learning, e.g. that crossing an object while holding it turns the pen into a cutter. Similarly, Matulic & Norrie's pen+touch interface for document editing [10] requires the user to learn the non-dominant hand gestures. By contrast, SPad uses visible buttons, which are more easily discoverable.

Finally, SPad's use of the thumb builds on previous work on smartphones: Thumbmenu [7] is a hierarchical quarter-pie menu in the corner of the screen, two-handed marking menus [9] are multi-stroke marking menus in which users draw strokes with both hands simultaneously or alternatively, bezel menu [8] is a hierarchical marking menu invoked by a bezel swipe, ThumbRock [3] is a micro-gesture that provides an alternative to tapping. SPad differs from these techniques in several ways: it can be invoked anywhere along the screen edge, it is designed to follow the natural motion of the thumb and its buttons trigger quasimodes, not commands.

SPad however has some limitations. It takes screen real estate, hiding and showing it incurs extra interactions, and users may need to adjust to bimanual interaction. Finally, grouping commands into four groups of three buttons can be constraining. Nevertheless, we believe that SPad is easy to learn and that users will quickly discover the increase in productivity it gives them.

Conclusion and Future Work

SPad is a thumb-based interaction technique that improves the efficiency of multi-touch tablets in productivity applications by combining bimanual interaction, quasimodes and polymorphic gestures. We described the design rationale of SPad and showed how it can make interaction faster, more direct and more powerful than current techniques without increasing complexity. We discussed how SPad builds on and differs from previous work, as well as some of its limitations.

Future work includes refining the design, applying it to a full-fledged application and evaluating it in order to unleash the power of tablets for productivity applications.

References

[1] Bau, O., and Mackay, W. E. Octopocus: a dynamic guide for learning gesture-based command sets. In *Proc. User Interface Software and Technology*, UIST '08, ACM (2008), 37–46.

[2] Beaudouin-Lafon, M. Instrumental interaction: an interaction model for designing post-wimp user interfaces. In *Proc. Human Factors in Computing Systems*, CHI '00, ACM (2000), 446–453.

[3] Bonnet, D., Appert, C., and Beaudouin-Lafon, M. Extending the vocabulary of touch events with ThumbRock. In *Proc. Graphics Interface*, GI '13 (2013), 221–228.

[4] Freeman, D., Benko, H., Morris, M. R., and Wigdor, D. Shadowguides: visualizations for in-situ learning of multi-touch and whole-hand gestures. In *Proc. Interactive Tabletops and Surfaces*, ITS '09, ACM (2009), 165–172.

[5] Guiard, Y. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *J. Motor Behavior 19* (1987), 486–517.

[6] Hinckley, K., Yatani, K., Pahud, M., Coddington, N., Rodenhouse, J., Wilson, A., Benko, H., and Buxton, B. Pen + touch = new tools. In *Proc. User Interface Software and Technology*, UIST '10, ACM (2010), 27–36.

[7] Huot, S., and Lecolinet, E. Archmenu et thumbmenu: contrôler son dispositif mobile "sur le pouce". In *Proc. Interaction Homme-Machine*, IHM '07, ACM (2007), 107–110.

[8] Jain, M., and Balakrishnan, R. User learning and performance with bezel menus. In *Proc. Human Factors in Computing Systems*, CHI '12, ACM (2012), 2221–2230.

[9] Kin, K., Hartmann, B., and Agrawala, M. Two-handed marking menus for multitouch devices. *ACM Transactions on Computer-Human Interaction 18*, 3 (2011), 16.

[10] Matulic, F., and Norrie, M. C. Pen and touch gestural environment for document editing on interactive tabletops. In *Proc. Interactive Tabletops and Surfaces*, ITS '13, ACM (2013), 41–50.

[11] Norman, D. A., and Nielsen, J. Gestural interfaces: a step backward in usability. *ACM interactions 17*, 5 (2010), 46–49.

[12] Parhi, P., Karlson, A. K., and Bederson, B. B. Target size study for one-handed thumb use on small touchscreen devices. In *Proc. Human-Computer Interaction with Mobile Devices and Services*, ACM (2006), 203–210.

[13] Roth, V., and Turner, T. Bezel swipe: conflict-free scrolling and multiple selection on mobile touch screen devices. In *Proc. Human Factors in Computing Systems*, CHI '09, ACM (2009), 1523–1526.

[14] Wagner, J., Huot, S., and Mackay, W. BiTouch and BiPad: Designing bimanual interaction for hand-held tablets. In *Proc. Human Factors in Computing Systems*, CHI '12, ACM (2012), 2317–2326.