International Journal of Human - Computer Studies 99 (2017) 21–36

Contents lists available at ScienceDirect

Int. J. Human–Computer Studies

journal homepage: www.elsevier.com/locate/ijhcs

Leveraging finger identification to integrate multi-touch command selection and parameter manipulation

Alix Goguey[a,*], Daniel Vogel[b], Fanny Chevalier[a], Thomas Pietrzak[c], Nicolas Roussel[a], Géry Casiez[c]

[a] Inria Lille – Nord Europe, 40 Avenue du Halley, 59650 Villeneuve-d'Ascq, France
[b] Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada
[c] Université de Lille, Cité Scientifique, 59650 Villeneuve-d'Ascq, France

ARTICLE INFO

Keywords:
Multi-touch
Finger identification
Shortcuts
Command selection
Parameter control
Direct manipulation

ABSTRACT

Identifying which fingers are touching a multi-touch surface provides a very large input space. We describe FingerCuts, an interaction technique inspired by desktop keyboard shortcuts to exploit this potential. FingerCuts enables integrated command selection and parameter manipulation, it uses feed-forward and feedback to increase discoverability, it is backward compatible with current touch input techniques, and it is adaptable for different touch device form factors. We implemented three variations of FingerCuts, each tailored to a different device form factor: tabletop, tablet, and smartphone. Qualitative and quantitative studies conducted on the tabletop suggests that with some practice, FingerCuts is expressive, easy-to-use, and increases a sense of continuous interaction flow and that interaction with FingerCuts is as fast, or faster than using a graphical user interface. A theoretical analysis of FingerCuts using the Fingerstroke-Level Model (FLM) matches our quantitative study results, justifying our use of FLM to analyse and validate the performance for the other device form factors.

1. Introduction

A common belief is that multi-touch input is an excellent match for direct manipulation interaction. Certainly panning and zooming with two-finger pinch-to-zoom, or simultaneously rotating, translating and scaling objects with multiple fingers are better semantic and articulatory translations (Hutchins et al., 1985) of real world actions than dragging scrollbars, sliders or transformation handles. But, a direct manipulation interface should also provide access to a variety of *commands* to trigger global actions (e.g. “undo”) and contextual actions (e.g. “delete this object”), activate modes (e.g. “start drawing rectangles”) and modify modes (e.g. “draw rectangles from the centre”). With a proportionately small number of possible multi-touch input actions, most commands are represented visually, away from the object of interest in toolbars, palettes or menus which occupy screen space and break interaction flow into a complex and time consuming “back-and-forth” graphical syntax (Beaudouin-Lafon, 2000). The restricted display area together with limited input vocabulary results in real-world touch interfaces that tend to be less rich in functionalities than their desktop counterparts.[1]

In desktop computing, keyboard shortcuts provide an effective alternative to clicking on graphical buttons when issuing commands (Grossman et al., 2007). Once mastered, they decrease command activation time, enable redundant palettes to be hidden, and allow for a more fluid flow between command selection and the continuous control of command parameters. For example, pressing the R key with one hand could activate a “draw rectangles” mode and then the rectangle location and size parameters can be controlled with the mouse in the other hand. Current multi-touch input primarily uses simultaneous touch points and long presses which provides a far more limited vocabulary than keyboard shortcuts.

No wonder researchers and designers are constantly searching for ways to increase the multi-touch input space using techniques like gestures (Kin et al., 2011), touch locations (Serrano et al., 2013), touch patterns (Ghomi et al., 2012), normal and tangential forces (Wang and Ren, 2009), discriminating finger parts (Harrison et al., 2011) and more. Yet, these still treat all fingers equally which confines the input vocabulary – identifying *which* fingers are touching the surface provides a much larger input space. Recent studies show that even for single-touch command selection, finger identification is a promising

* Corresponding author.
E-mail addresses: alix.goguey@inria.fr (A. Goguey), dvogel@uwaterloo.ca (D. Vogel), fanny.chevalier@inria.fr (F. Chevalier), thomas.pietrzak@univ-lille1.fr (T. Pietrzak), nicolas.roussel@inria.fr (N. Roussel), gery.casiez@univ-lille1.fr (G. Casiez).

[1] Wagner et al. (2014) note that the desktop version of Adobe Photoshop (CS6) has 648 menu commands while the tablet version (Express) has only 35 commands.

http://dx.doi.org/10.1016/j.ijhcs.2016.11.002
Received 29 January 2016; Received in revised form 31 August 2016; Accepted 10 November 2016
Available online 16 November 2016
1071-5819/ © 2016 Elsevier Ltd. All rights reserved.

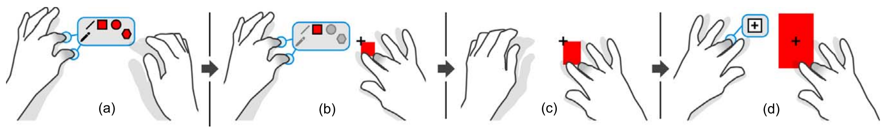

Fig. 1. A bimanual tabletop example of how FingerCuts merges command selection with direct manipulation of command parameters with feed-forward: (a) specific finger chords with non-dominant hand displays feed-forward showing possible commands, e.g. thumb and index finger chord shows different drawing modes; (b) dominant hand selects command using specific corresponding finger, e.g. middle finger triggers mode activation command to "start drawing rectangle"; (c) dominant hand movement provides continuous control of command parameters, e.g. specifies the size of the rectangle; (d) other specific fingers with non-dominant hand further tunes the command parameters, e.g. middle finger modifies draw rectangle mode to "draw rectangle from the centre". See Section 6 for a restricted bimanual example with a tablet and unimanual example with a smartphone.

solution to increase the touch input vocabulary (Roy et al., 2015). Robust finger identification sensing is on the horizon (Holz and Baudisch, 2013; Benko et al., 2009), but examples of the interaction it enables have only been isolated point designs without cohesive design. For example, Marquardt et al. (2011) built an exploratory toolkit for tabletops and explored a range of interaction technique ideas to emphasize the toolkit's expressiveness, but they did not provide a consistent way to apply finger identification to direct manipulation interfaces. Our research moves beyond point designs by contributing a cohesive finger identification-enabled interaction technique focused on the classic goal of merging command selection with direct manipulation of command parameters (Guimbretière et al., 2005; Bier et al., 1993).

We contribute FingerCuts, an interaction technique inspired by keyboard shortcuts that leverages finger identification for direct manipulation. In desktop computing, experts typically use keyboard shortcuts in three steps: (1) the non-dominant hand triggers a command with a key; (2) the dominant hand manipulates with a pointing device; and (3) the non-dominant hand optionally tunes manipulation with a modifier key. With finger identification-enabled touch, a similar three-step pattern can be adopted and made universal so that subsequent command selection and manipulation are perfectly merged: (1) one set of fingers selects a command mapping for another set of fingers (e.g. the thumb and index finger chord in Fig. 1(a) shows a set of possible drawing commands); (2) another finger triggers a command and immediately begins direct manipulation of command parameters (e.g. the middle finger draws and adjusts the size of a rectangle in Fig. 1(b, c))— at this point, the fingers used to select the command mapping can be released; and (3) the first set of fingers optionally tunes the manipulation (e.g. the middle finger of the non-dominant hand specifies the mode of drawing, namely from the centre, in Fig. 1(d)). For novices, the progression of actions provides an opportunity to display simple feed-forward (Wensveen et al., 2004) for discoverability. For experts, this progression of overlapping finger actions forms a tightly coupled phrase that can be chunked (Buxton, 1986).

We perform a systematic analysis of the finger identification interaction space to motivate practical design considerations for FingerCuts. These argue for maintaining backward compatibility with the existing multi-touch techniques like pinch-to-zoom, and guide designers when applying the technique to different device form factors. To demonstrate how FingerCuts works in different device form factors, we implemented it in a vector drawing application for a tabletop, a document annotation application for a tablet, and a messaging application for a smartphone.[2] We evaluate the tabletop version in two user studies. A qualitative study suggests that with some practice, FingerCuts is expressive, easy-to-use, and increases a sense of continuous interaction flow. A quantitative study shows that FingerCuts can be as fast, or faster than a traditional graphical user interface, primarily due to the lower number of operations required. A Fingerstroke-Level Model (FLM) analysis supports the results of the tabletop quantitative study and enables us to generalize these results to the tablet and smartphone form factors.

[2] Video demonstration at http://ns.inria.fr/mjolnir/fingercut-ijhcs

2. Background and related work

A direct manipulation interface must provide a way to issue *commands* to the system. There are commands for global actions (e.g. "undo") and contextual actions to be performed on a specified object (e.g. "delete this rectangle"). There are also commands to activate *modes*, where the system should start interpreting input in a particular way (e.g. "start drawing rectangles"). Modes are conceptually associated with real world *tools*. Commands can also modify the current mode (e.g. "keep drawing rectangles, but draw them from the centre"). These *modifiers* are often temporarily maintained by a kinesthetic action, such as holding a key down. At a command level, this means a key down sends a command to start modifying the current mode and a key up stops modifying the current mode. This same temporary kinesthetic action can be applied to modes, which are then called "quasimodes" (Raskin, 2000).

Providing an efficient way to trigger all these commands is challenging with multi-touch. Multiple points of contact provide some expression, but nowhere near the discrete input space of keyboard shortcuts. Techniques to increase the multi-touch input space have focused on spatial information, dwell timeouts, motion to enrich the interaction vocabulary, and even cross device interactions (Chen et al., 2014; Houben et al., 2015). However, these also increase time (e.g. dwell) or space offset (e.g. gestures) and they move the user's focus away from the primary object of interest (Beaudouin-Lafon, 2000).

2.1. *Finger identification*

Finger identification—associating specific fingers and hands with touch points—is a way to increase the multi-touch input space. Significant previous work has tackled the technical sensing problem with approaches like: inferring finger identity based on geometric relationships between touch contact points (Au and Tai, 2010; Ewerling et al., 2012; Vinayak Murugappan et al., 2012; Westerman, 1999; Lepinski et al., 2010; Wagner et al., 2014); using an overhead camera to track bare hands (Malik et al., 2005) or fingers with coloured rings (Wang and Canny, 2004); wearing gloves with fiducial markers (Marquardt et al., 2011); recognizing fingerprints (Sugiura and Koseki, 1998; Holz and Baudisch, 2013); forearm electromyography (Benko et al., 2009); and using RFID attached to fake plastic nails (Vega and Fuks, 2013). It seems inevitable that one day finger identification will be a standard feature of consumer multi-touch devices. Our interest is interaction, not sensing, so we focus our review on interaction techniques enabled by finger identification rather than the underlying technologies.

2.1.1. *Invoking commands with fingers and chords*

Independent of sensing approach, a common interaction technique to demonstrate the potential of finger identification is associating a different command with each finger. Sugiura and Koseki (1998) used the index, middle, ring, and little fingers to operate a music player and add web browser bookmarks. Marquardt et al. (2011) mapped the middle and little fingers to cut and copy commands. However, with only 10 fingers, a limited set of commands can be triggered.

By recognizing *chords*—the simultaneous contact of two or more

Download English Version:

https://daneshyari.com/en/article/4945857

Download Persian Version:

https://daneshyari.com/article/4945857

Daneshyari.com