



HAL
open science

A Multitouch Drawing Application with Occlusion-Free Interaction Strategies

Yi Ren Tan, Hyowon Lee, Insuk Ko, Dongliang Zhang, Chunxiao Li

► **To cite this version:**

Yi Ren Tan, Hyowon Lee, Insuk Ko, Dongliang Zhang, Chunxiao Li. A Multitouch Drawing Application with Occlusion-Free Interaction Strategies. 17th IFIP Conference on Human-Computer Interaction (INTERACT), Sep 2019, Paphos, Cyprus. pp.535-539, 10.1007/978-3-030-29390-1_32 . hal-02877646

HAL Id: hal-02877646

<https://inria.hal.science/hal-02877646v1>

Submitted on 22 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

A Multitouch Drawing Application with Occlusion-free Interaction Strategies

Yi Ren Tan¹, Hyowon Lee¹, Insuk Ko¹, Dongliang Zhang² and Chunxiao Li²

¹ Singapore University of Technology and Design, 8 Somapah Road, Singapore 487372

² Zhejiang University, Hangzhou, Zhejiang, China 300058
hlee@sutd.edu.sg

Abstract. Increasing number of desktop applications are becoming available on smartphones and tablets today with multitouch capabilities, allowing the users' fingers to perform sophisticated or fine-grained interactivities. However, finger occlusion and imprecision continue to limit the performance of multitouch interactions. Quite a few studies proposed the ways to address this issue, and some of them are now used in commonly encountered situations such as text editing. Many occlusion-avoiding techniques used today focus on initial target acquisition step of touch interaction (e.g. accurately selecting an item or touching a desired starting point in drawing a line), having possible consequences to any further intended action (e.g. dragging the selected item to a different location or drawing a line on the canvas). In order to better understand the influence of finger occlusion-free techniques on other parts of the overall interactions, in this paper we report a full-fledged sketch app that incorporates combinations of basic target acquisition features. As the app is a full-featured, end-to-end tablet prototype, such usability issues can be more readily revealed and discussed in the context of realistic drawing situations.

Keywords: Finger Occlusion, Pointing Precision, Multitouch interaction.

1 Introduction

Multitouch platforms are becoming increasingly popular today due to their intuitiveness and convenience. However, such platforms continue to suffer from the problem of finger occlusion, also known as “fat-finger problem”, which refers to the difficulty of selecting or manipulating small targets on a screen due to the relatively larger size of the user's finger. Small targets tend to be hidden under the finger during interaction, resulting in error and imprecision.

While stylus is a straightforward solution to imprecision and finger occlusion, it is not without its own set of disadvantages. Stylus requires a stable surface to perform well, and most styluses are inconvenient to carry along. Styluses can also introduce other usability problems such as parallax error and pen/hand occlusion as addressed in previous studies (e.g. [1, 3]). In the context of applications that allow the user to draw lines and other visual sketches, the impact of finger occlusion becomes greater as the

action of drawing often requires the finger to contact the display surface for longer durations and with much finer control.

A drawing action can be broken down into two distinct input steps: (1) accurately positioning of the initial point of selection (i.e. target acquisition), and (2) accurately dragging along the desired path of drawing. While past studies have attempted at solving occlusion and imprecision using various indirect pointing techniques (e.g. [2, 3, 4, 5]), their main focus of attention and evaluation was on positioning the initial and/or end point of selection. Little is known about the performances of these techniques in accurately supporting any remaining interaction such as dragging along a path, having once accurately targeted a point.

A quick review of existing sketching-related apps today shows that most of them do not include any occlusion-free features. Top surface-based drawing apps such as Procreate¹ and Autodesk Sketchbook² are feature-rich and used by hobbyists as well as professionals. Yet, none of them have explicitly addressed the issue of finger occlusion on their user-interfaces. Most users either depend on a stylus for better accuracy, or frequently “undo” to redraw their inaccurate lines repeatedly.

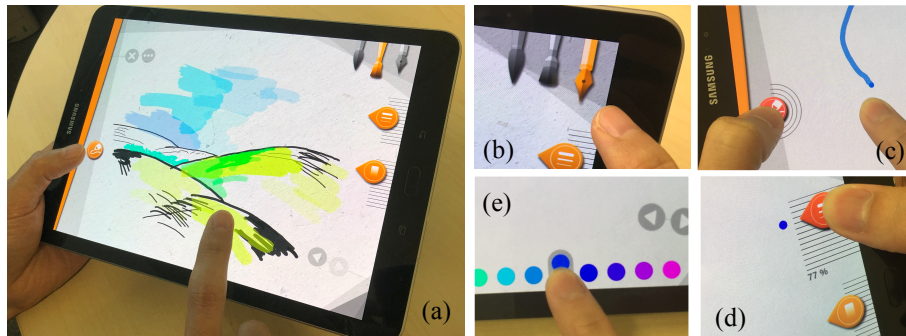


Fig. 1. (a) Drawing performed in the occlusion-free app; (b) Main drawing tools; (c) Activation button tapped with offset pointer; (d) Selecting brush thickness; (e) Selecting a colour.

In this demonstration, we present a full-fledged tablet sketch app we developed (Fig. 1), incorporating some combinations of basic occlusion-free interaction strategies as integral features of the app. By implementing these occlusion-free features in a realistic sketch app, we aim to explore, reveal and better understand how some of the techniques for multitouch interaction can impact the usability when adopted in a real tablet app in the context of drawing tasks. We believe that the app, though in itself does not feature any novel occlusion-free techniques, very effectively raises a number of useful discussion points that help re-visit the concept, scope and value of finger occlusion-avoiding techniques thus far proposed and adopted, by placing some of the representative techniques in the context of realistic application situations.

¹ Procreate: <https://procreate.art/>

² Sketchbook: <https://sketchbook.com/mobile>

2 Sketch Application Overview

The design of our sketching app aims to fulfill three key objectives. Firstly, in order to capture the full spectrum of use cases in actual drawing applications, we attempt to replicate essential drawing features found in commercial apps such as variety of brush types (Fig. 1(b)), controls for brush size and opacity (Fig. 1(e)), color palette (Fig. 1(d)), canvas zoom/rotation/pan, as well as the undo/redo functions. It is not feasible to replicate every single feature we see in commercial apps such as layers, filters and image manipulation, so we decided to prioritize features that would directly impact the action of drawing on the canvas.

Secondly, all elements of the user interface were designed to be occlusion-free when interacting directly with a finger. We approached this using large finger-sized buttons that are easy to target and clear animated visual feedback that pops out around the point of touch to prevent information from being hidden under the finger (Fig. 1).

Thirdly, some of the occlusion-free target acquisition techniques are incorporated to support the actual drawing actions, further elaborated in the next section³.

The app was developed in Unity game engine, and works on any tablet platforms. It was installed and tested on Samsung Galaxy Tab S2 tablet with 9.7-inch multitouch display with the resolution of 2048px by 1536px, a mid-range tablet device widely used as of today.

3 Occlusion-free Strategies in the Application

When formulating our occlusion-free input strategies, we adopted a design space consisting of two factors: (1) Input Mapping and (2) Activation Method.

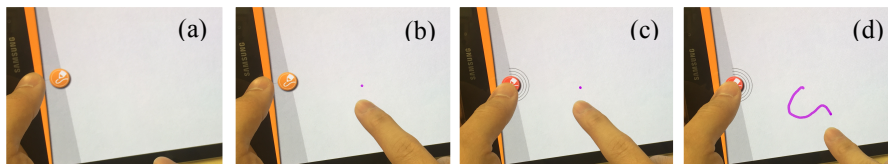


Fig. 2. Offset-UMM technique: (a) Activation button is featured to be pressed with non-dominant hand; (b) dominant hand now touches the surface, showing the offset pointer above the finger; (c) once targeting is satisfactory, the activation button is pressed; (d) draws while the button is kept pressed.

Input Mapping refers to how the cursor is positioned on the screen in relation to the point of contact of the finger. When we tap on a surface, the action of the cursor usually happens directly under our finger, causing the cursor action to be hidden. Through introducing an indirect input mapping to our system, we can position the cursor away from under the finger so the action of the cursor becomes visible again.

³ A short video demonstrates the app in action: <https://youtu.be/YYTWjidi4OM>

Activation Method refers to the gesture used to commit an action with the cursor. For example, on a desktop mouse input, the activation method is to press the left mouse button to commit an action. The usual way of committing an action on an interactive surface is to simply tap the finger on the display. This method is error-prone, as it forces the user to commit the action before the user can visually identify exactly where the cursor is. By introducing alternative activation methods, along with indirect input mapping, the accuracy of touch input may be significantly improved.

Combination of the two factors allows us to arrive at our primary occlusion-free strategy, the Offset-UMM technique. This technique uses an input mapping known as Offset and an activation method called UMM (User-Maintained Mode) interaction [6]. The Offset input mapping positions the cursor a short distance (e.g. 1.5cm) vertically above the point of touch of the finger [2]. This prevents occlusion while keeping the cursor close to the finger so it is easy for the user to locate. However, the offset cursor only appears when the finger touches the screen (Figs. 2(a), 2(b)). In this strategy, the UMM interaction is used for activation where the line drawing is committed only when a dedicated, separate button is pressed and kept pressed (Figs. 2(c), 2(d)). Overall this strategy works well when both hands are available. Such bimanual use of UMM in interaction design has been shown to be a promising alternative interaction in a number of application scenarios [6]. In a similar fashion, three other occlusion-free strategies were formulated, using different input mappings and activation methods. While a usability testing was conducted and its detailed analysis is reported elsewhere [7], in this demonstration paper we mainly introduce the app that feature the 4 occlusion-free strategies as a demonstrator to encourage discussions on the usability of the techniques on the remaining interactivity when conducting a task.

References

1. Lee, D., Son, K., Lee, J. and Bae, S. 2012. PhantomPen: virtualization of pen head for digital drawing free from pen occlusion & visual parallax. In *Proceedings of the 25th Annual ACM symposium on User interface software and technology*. (UIST '12), 331-340.
2. Matejka, J., Grossman, T., Lo, J., and Fitzmaurice, G. 2009. The design and evaluation of multi-finger mouse emulation techniques. In *Proceeding of the SIGCHI Conference on Human Factors in Computing Systems*. (CHI '09), 1073-1082.
3. Vogel, D. and Balakrishnan, R. 2010. Occlusion-aware interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. (CHI '10), 263-272.
4. Roudaut, A., Huot, S. and Lecolinet, E. 2008. TapTap and MagStick: Improving One-Handed Target Acquisition on small touch-screens. In *Proceedings of the working conference on Advanced visual interfaces*. (AVI '08), 146-153.
5. Lai, J. and Zhang, D. 2014. ExtendedThumb: a motion-based virtual thumb for improving one-handed target acquisition on touch-screen mobile devices. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*. (CHI EA '14), 1825-1830.
6. Fennedy, K., Lee, H., Ko, I., Tan, Y., Zhang, D. and Li, C. 2018. Augmenting User-Maintained Interaction Through Mode Locking and Reversing. *32nd British Computer Society Human Computer Interaction Conference*.
7. Lee, H., Ko, I., Tan, Y. R., Zhang, D. and Li, C. 2019. Usability Impact of Occlusion-free Techniques on Commonly-used Multitouch Actions. *5th International ACM In Cooperation HCI and UX Conference (CHIuXiD 2019)*.