February 2021

Suppression of Unintentional Touch Inputs Caused by Moving Content on a Display

Zoltan Egyed

Suppression of Unintentional Touch Inputs Caused by Moving Content on a Display

Abstract:

This publication describes techniques for suppression of unintentional touch inputs caused by moving content on a display of a computing device. Due to a time delay between the moment a user decides to provide a touch input (e.g., a tap, click, swipe, pinch) to the display and the moment the touch input is detected by the device, content may move, changing the local meaning of the touch input. For example, a user may accidentally click “delete” instead of “save” if the content moves during the time it takes the user to provide the touch input. To suppress unintentional touch inputs, an Input Manager may determine the local meaning at the location of the touch input when the input has been detected and determine approximately when the user intended to touch the display. If the local meaning has changed, the Input Manager instructs the device to ignore actions associated with the touch input. If the local meaning has not changed, then actions are performed as usual.

Keywords:

Unintentional touch, unintentional input, accidental, mis-click, false click, mistake, touch screen, haptic input, control touch, control input, user intent, unintentional haptic input, display change, local meaning change, clickjacking, redress attack

Background:

There is a time delay between the moment a user decides to provide a touch input to a display of a computing device (e.g., a touchscreen of a mobile phone) and the moment the touch input is actually detected by the device. Even though this time delay may only include, for example, a delay of a fraction of a second, content may move on the display during that second, resulting in unintentional touch inputs. For example, a user may decide to press "send" on a message and move their finger towards the display to provide a click corresponding with "send." If the content moves before the user clicks the display, the user may accidentally press "cancel" and have to start all over again. This problem may escalate as the user performs more sensitive actions on their device such as responding to emails, purchasing merchandise, and managing bank accounts. Furthermore, this problem may be exploited by advertisement services to lure users (e.g., via a user interface redress attack) into accidentally clicking on links that may allow these services to collect data or send unsolicited messages to the user. These unintentional touch inputs due to moving content on the display may diminish a user experience. As a result, there are barriers preventing a user from reliably providing touch inputs that correctly correspond with their intentions.

Description:

This publication describes techniques for preventing unintentional touch inputs caused by moving content on a display of a computing device. While the example computing device described in this publication is a mobile phone, other types of computing devices may also support the techniques described herein.

A computing device may include one or more processors, transceivers for transmitting data to and receiving data from a base station (e.g., wireless access point, another computing device), sensors (e.g., a touch sensor), a computer-readable medium (CRM), and/or an input/output device (e.g., a display, a speaker, a microphone). The CRM may include any suitable memory or storage device like random-access memory (RAM), static RAM (SRAM), dynamic RAM (DRAM), non-volatile RAM (NVRAM), read-only memory (ROM), or flash memory. The CRM includes device data (e.g., user data, multimedia data, applications, and/or an operating system of the device), which are executable by the processor(s) to enable the techniques described herein.

The touch sensor (e.g., a touchscreen) may be integrated into the display to detect touch input from a user to control, monitor, or perform operations on the device. The touch input may include, for example, a single touch, a multi-touch, or a motion of a finger, multiple fingers, or hand across a surface of the display.

The device data may also include an Input Manager. The computing device performs operations under the direction of the Input Manager to determine if a touch input detected on the display is intentional. A user may decide to provide a touch input to change or manage information displayed on the device at a first time (e.g., time 1). In the example illustrated in Figure 1, a user may decide to save information via a touch (e.g., a tap with an index finger) of a “save” button at time 1 before contact is made between the user’s index finger and the display. Since it takes time for the user to make contact with the display (e.g., 0.7–1.5 seconds), the content may move during that time (e.g., upwards on the display), resulting in an unintentional touch of a “delete” button at time 2 during the contact.

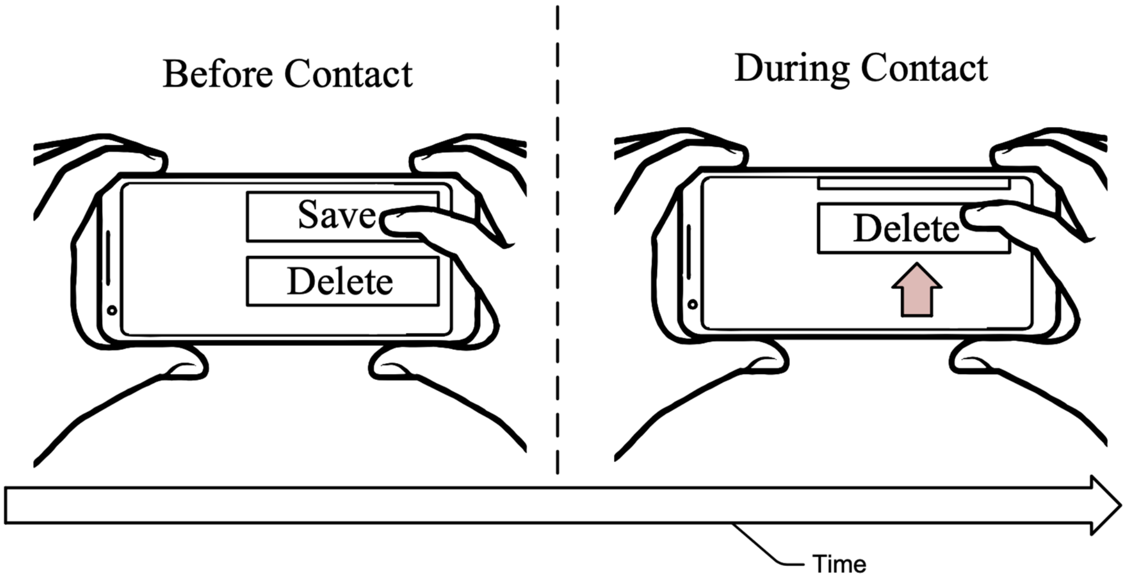

Figure 1

To suppress unintentional touch inputs, the Input Manager may detect a touch input received by the device at time 2 as illustrated in Figure 2. The Input Manager does not know exactly when the user decided to provide the touch input at time 1 but may assume an appropriate time based on a reaction rate of the user (e.g., corresponding to a lockout time period). The lockout time period may include an amount of time the user needs to perform the touch input and may be adjusted, for example, between 0.7–1.5 seconds to improve a user experience. In general, the lockout time period may be longer than 1.5 seconds and shorter than 0.7 seconds depending on the needs of the user.

In aspects, the Input Manager may utilize a machine-learned (ML) model to adjust the lockout time period based on a speed at which the user performs touch inputs. For example, if the user consistently provides touch inputs with a slow reaction rate, then the ML model may instruct a processor to increase a user-specific lockout time period to provide the user with enough time to perform the touch input.

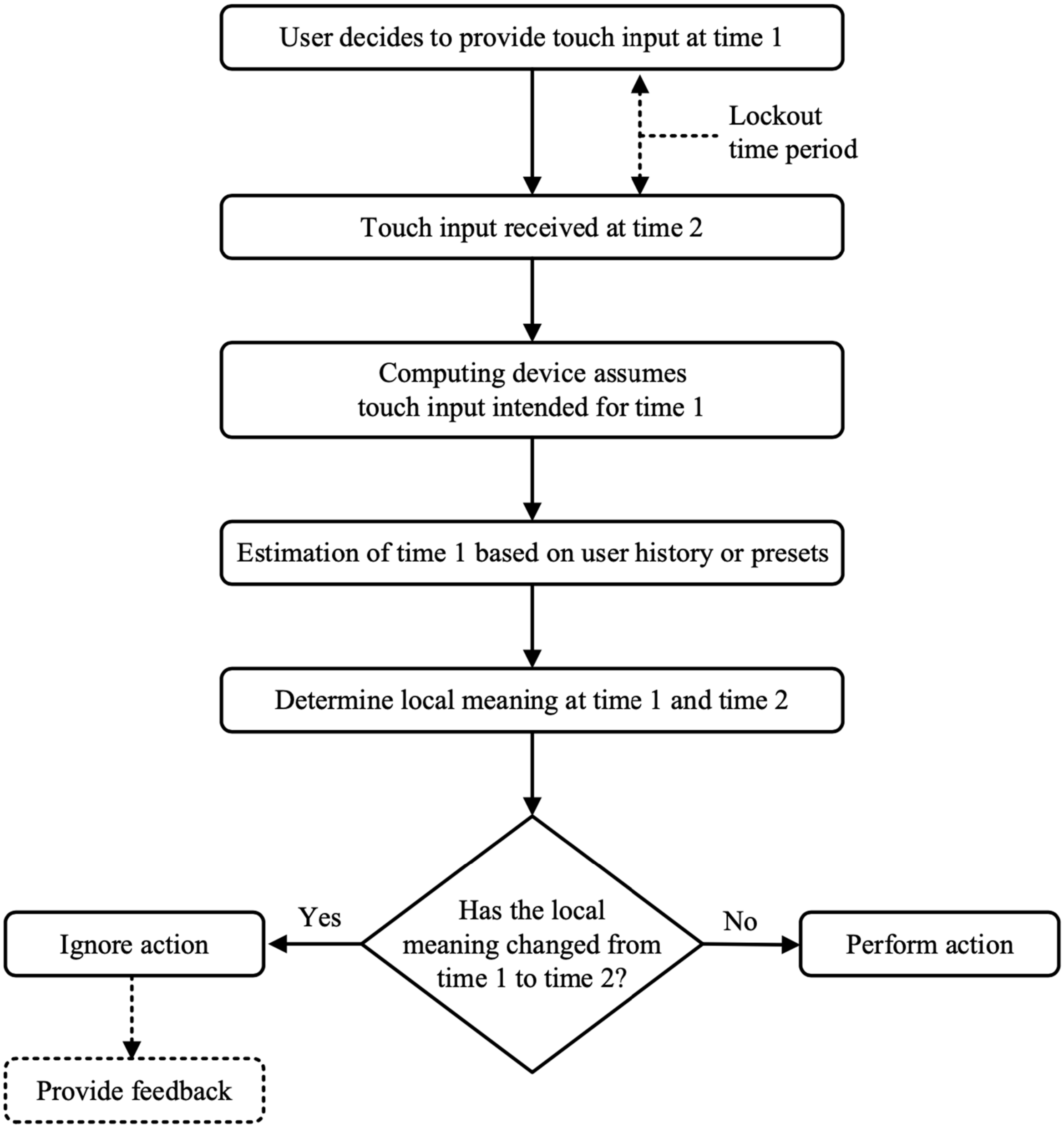

Figure 2

The ML model may run on touch-controlled firmware (e.g., as part of a firmware running on a touch controller) on the CRM, as an extension of a touch driver, on an application processor (e.g., a host or driver), or on a power-efficient system (e.g., using flash memory). The ML model may be a standard neural-network-based model with corresponding layers required for processing input features like fixed-side vectors, text embeddings, or variable-length sequences. The ML model may be implemented as one or more of a support-vector machine (SVM), a recurrent neural network (RNN), a convolutional neural network (CNN), a dense neural network (DNN), one or more heuristics, other machine-learning techniques, a combination thereof, and so forth. The ML model may be trained to detect the lockout time period associated with intentional touch inputs based on a user history and/or presets of the device. The user-specific lockout time period and associated preferences may be stored, for example, on the device or in a cloud, and used by a processor(s) of the device to execute the techniques of the Input Manager.

After the Input Manager determines time 1 and time 2 associated with the touch input, the local meanings of the display may be determined at time 1 and time 2. The local meaning may include operations executable by the touch input. In Figure 1, the local meaning at the location of the touch input before contact included a “save” button. Whereas, the local meaning at the location of the touch input during the contact by the user included a “delete” button. The Input Manager may detect these two local meanings and determine that there has been a change in the local meaning during the lockout time period.

In another example, a user may be browsing a website for a new backpack on their mobile phone and decide to click on a bag to view further details. However, the content on the display moves before the user clicks on the bag. Instead, an advertisement for a car company appears in that location of the display. Now, the user has unintentionally clicked on a link to the car website. To suppress this unintentional click, the Input Manager may detect that the local meaning has changed. In this example, the local meaning at time 1 included a link to the bag for further details and the local meaning at time 2 included a link to the car website. The Input Manager may compare the links to determine a change in the local meaning and assume that the user unintentionally clicked on the car website.

If the local meaning has changed between time 1 and time 2 at the location of the touch input, then the Input Manager may instruct a processor to ignore any actions associated with the touch input (e.g., loading the car website). The device may further provide audio and/or haptic feedback to notify the user that actions associated with the touch input have been ignored. If the local meaning has not changed between time 1 and time 2, then the Input Manager may instruct the processor to accept the touch input and perform associated actions. The Input Manager may be further configured to accept touch inputs and perform actions, even if the local meaning has changed, if the local meaning at time 2 does not include an action that, if performed, may diminish a user experience. For example, if the local meaning at time 1 includes a button to "save" and the local meaning at time 2 does not include an operation (e.g., a blank portion of the screen), the Input Manager may determine that the user intended to press "save" and perform the action.

References:

[1] Patent Publication: US20110181538A1. Input Apparatus. Priority Date: December 25, 2008.

[2] Patent Publication: US20120154294A1. Using Movement of a Computing Device to Enhance Interpretation of Input Events Produced when Interacting with the Computing Device. Priority Date: December 17, 2010.

[3] Patent Publication: US20150379558A1. Detecting Unintentional User Input to Mobile Advertisements. Priority Date: June 26, 2014.

[4] Patent Publication: US20200342494A1. Dynamically modifying activation behavior of a computerized graphical advertisement display. Priority Date: April 26, 2019.

[5] B. Vogel. August 8, 2017. "Reducing Unintentional Clicks." Facebook Audience Network.

https://www.facebook.com/audiencenetwork/news-and-insights/reducing-unintentional-clicks.