

Technical Disclosure Commons

Defensive Publications Series

September 2021

Implicit Gesture Confirmation for Touchscreen Input

Thomas Buckley

Robert Mickle

Abraham Murray

Tayeb Al Karim

Hiu Ying Chan

See next page for additional authors

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation

Buckley, Thomas; Mickle, Robert; Murray, Abraham; Al Karim, Tayeb; Chan, Hiu Ying; and Cirimele, Maria, "Implicit Gesture Confirmation for Touchscreen Input", Technical Disclosure Commons, (September 13, 2021)

https://www.tdcommons.org/dpubs_series/4578



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

Inventor(s)

Thomas Buckley, Robert Mickle, Abraham Murray, Tayeb Al Karim, Hiu Ying Chan, and Maria Cirimele

Implicit Gesture Confirmation for Touchscreen Input

ABSTRACT

Stylus apps support handwritten gestures for common actions by providing a mode to distinguish strokes from gestures. However, mode switching can create friction. Moreover, attempting to reduce friction by eliminating the mode switch can lead to gestures being triggered unintentionally, unreliably, or unpredictably. This disclosure describes techniques that reliably disambiguate strokes from gestures in stylus apps without mode switching and without the use of complex patterns to represent gestures. Per the techniques, a trace drawn on the touchscreen is initialized to a state in between a stroke and a gesture. The next action of the user determines the final state of whether a trace is classified as stroke or gesture. The classification of trace into stroke or gesture is quick such that the user can alternate strokes and gestures while the app seamlessly and accurately disambiguates between them.

KEYWORDS

- Sketching
- Handwriting input
- Mode switching
- Gesture disambiguation
- Stylus
- Gesture
- Stroke
- Note-taking
- Touchscreen interface

BACKGROUND

Applications that make use of a stylus or finger-based input, e.g., a sketching app, a note-taking app that accepts handwritten input, etc. recognize multiple types of input such as stroke, gesture, etc. performed using the stylus. Other input devices such as a mouse or drawing tablet can also enable users to provide such input. A stroke is a trajectory traced by the stylus (or finger) on a touchscreen or drawing tablet, or by the mouse pointer. A stroke can leave behind an electronic mark, such as a squiggle, on the touchscreen. A gesture is an action, e.g., select, move, erase, rotate, rescale, etc., on a stroke.

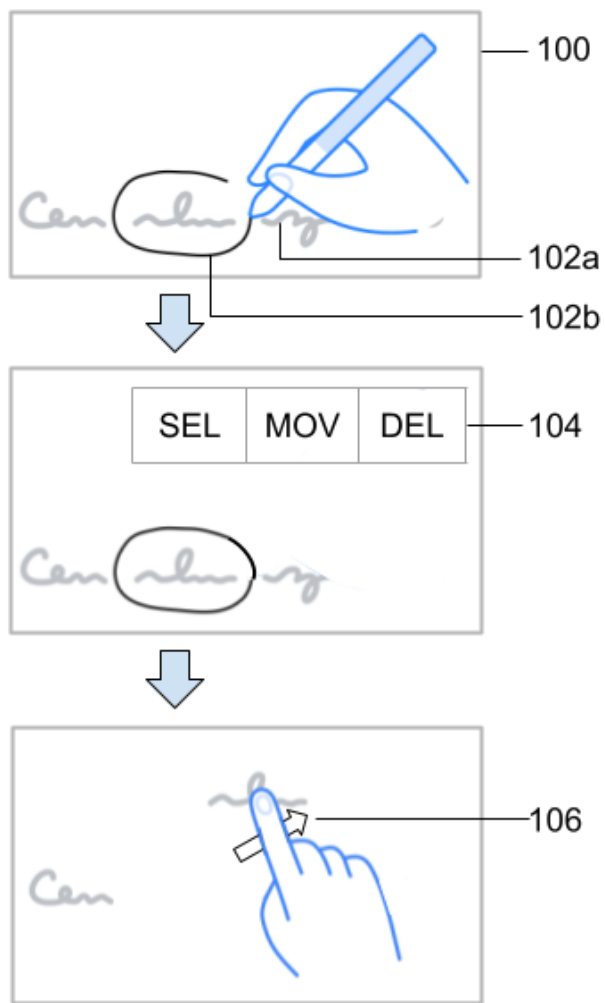


Fig. 1: Stroke and gesture

Fig. 1 illustrates an example sequence of input operations performed by the user on a touchscreen. The user enters a series of handwritten strokes (102a) on a canvas (100) to compose a handwritten note. To move a section of the note, the user marks the section by circling it (102b). At this point, it is unclear if the circle is intended as a stroke or a gesture. Note-taking or drawing apps typically pop up a menu (104) that offers gesture options, e.g., select, move, delete, etc. Sometimes the menu is not a popup menu but provided as a continuous presence at an edge of the canvas. If the circle is intended as a gesture (as it is, in this example), the user selects the appropriate option (MOV) from the menu to move the selection (106). Upon completion of the operation, the circle disappears.

Such switching of context between the canvas and the menu to disambiguate stroke from gesture is strenuous, especially when performed multiple times, as is likely the case for any note or drawing of ordinary complexity. While some stylus apps attempt to support hand-drawn gestures for common actions by providing a mode to distinguish strokes from gestures, the switching of mode can create friction. Moreover, attempting to reduce friction by eliminating the mode switch can lead to gestures triggering when not intended. Gestures sufficiently unusual to be easily disambiguated from strokes are difficult to learn and awkward to trigger. Stylus apps that use heuristics to disambiguate stroke from gesture, e.g., by using timing information, can be unreliable and act unpredictably.

DESCRIPTION

This disclosure describes techniques that reliably disambiguate strokes from gestures without mode or context-switching, and without the use of complex patterns to represent gestures. Per the techniques, a trace drawn on the touchscreen is initialized to an in-between state, where it is both a stroke and a gesture. The next action of the user determines the final

state, e.g., stroke or gesture, of the trace. If the user performs certain actions on the trace, e.g., tapping it, dragging it, etc., the trace is classified as a gesture. If the user continues writing normally, the trace is classified as a stroke. The classification of a trace into stroke or gesture is rapid (within low hundreds of milliseconds), such that the user can alternate strokes and gestures while the app seamlessly and accurately disambiguates between them.

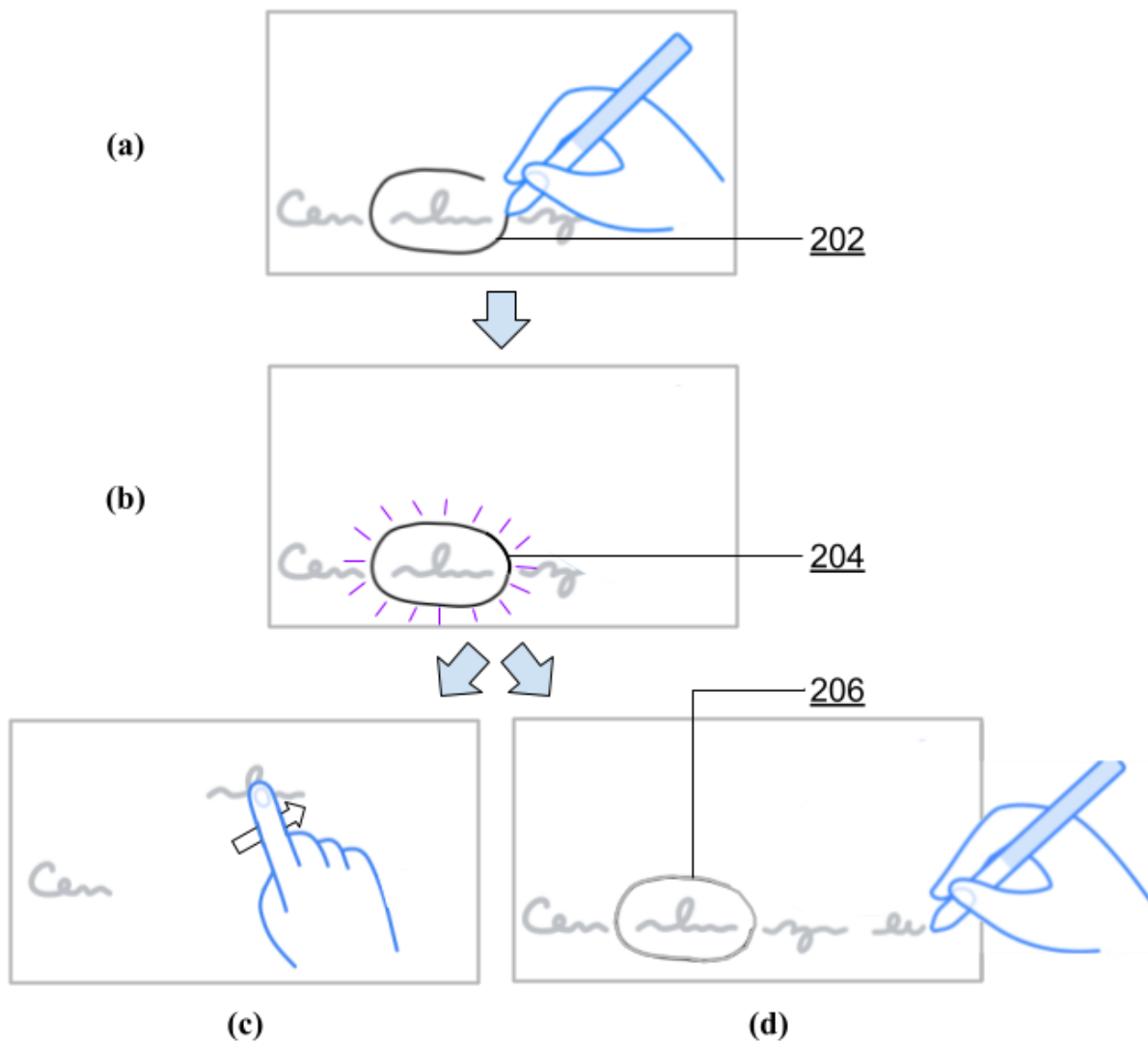


Fig. 2: A trace is initialized to a state in between stroke and gesture, and is classified as a trace or as a gesture depending on subsequent user action

Fig. 2 illustrates an example of the techniques. The user draws a trace (202, Fig. 2a) in the form of a circle. The trace is initialized to a state intermediate between stroke and gesture (204, Fig. 2b). The intermediate state of the trace is herein pictorially depicted by purple rays; an actual stylus app can use any suitable visual indication of intermediate state. In Fig. 2(c), the user touches the just-drawn circle and moves the finger (or stylus). Such user behavior causes the trace to be interpreted as a (select-and-move) gesture, and the text within the circle moves while the trace itself disappears. In an alternative illustrated Fig. 2(d), the user continues writing after drawing the circle. In this case, the trace is interpreted as a stroke, and the circle stays in position, becoming part of the handwritten text (or drawing).

CONCLUSION

This disclosure describes techniques that reliably disambiguate strokes from gestures in stylus apps without mode switching and without the use of complex patterns to represent gestures. Per the techniques, a trace drawn on the touchscreen is initialized to a state in between a stroke and a gesture. The next action of the user determines the final state of whether a trace is classified as stroke or gesture. The classification of trace into stroke or gesture is quick such that the user can alternate strokes and gestures while the app seamlessly and accurately disambiguates between them.