May 2022

INPUT LIFT-OFF DETECTION AND DRIFT COMPENSATION FOR PRESENCE-SENSITIVE DEVICES

Mingyuan Zhong

Wenxin Feng

Shumin Zhai

INPUT LIFT-OFF DETECTION AND DRIFT COMPENSATION FOR PRESENCE-SENSITIVE DEVICES

ABSTRACT

A mobile computing device (e.g., smartwatch, smartphone, tablet computer, laptop computer with touchscreen input, etc.) may determine when a user has lifted their finger, pen, pencil, stylus, or other input element from a presence-sensitive input device (e.g., resistive touchscreens, surface acoustic wave touchscreens, capacitive touchscreens, projective capacitance touchscreens, pressure sensitive screens, acoustic pulse recognition screens, etc.) and may compensate for drift in the touch centroid resulting from the lifting off of the input element. The mobile computing device may use a sliding window analysis, a recurrent neural network, or other algorithms to determine when a user has begun moving the input element away from the input device such that the input device ceases detecting the presence of the input element (e.g., a lift-off event). Based on the lift-off event and characteristics of the gesture prior to the lift-off event, the mobile computing device may adjust the gesture termination location to correct for drift caused by lifting off the input element. In this way, the techniques described herein may enable the mobile computing device to more accurately determine the location at which the gesture terminated.

DESCRIPTION

FIG. 1 below shows a computing device 100 that performs input lift-off detection and drift compensation in accordance with techniques of this disclosure. As shown in FIG. 1, computing device 100 includes a presence-sensitive display 102, one or more processors 104, and one or more storage devices 106. As further shown in FIG. 1, storage devices 106 includes an input lift-off detection module 108, a drift compensation module 110, and a heatmap feature repository 116.

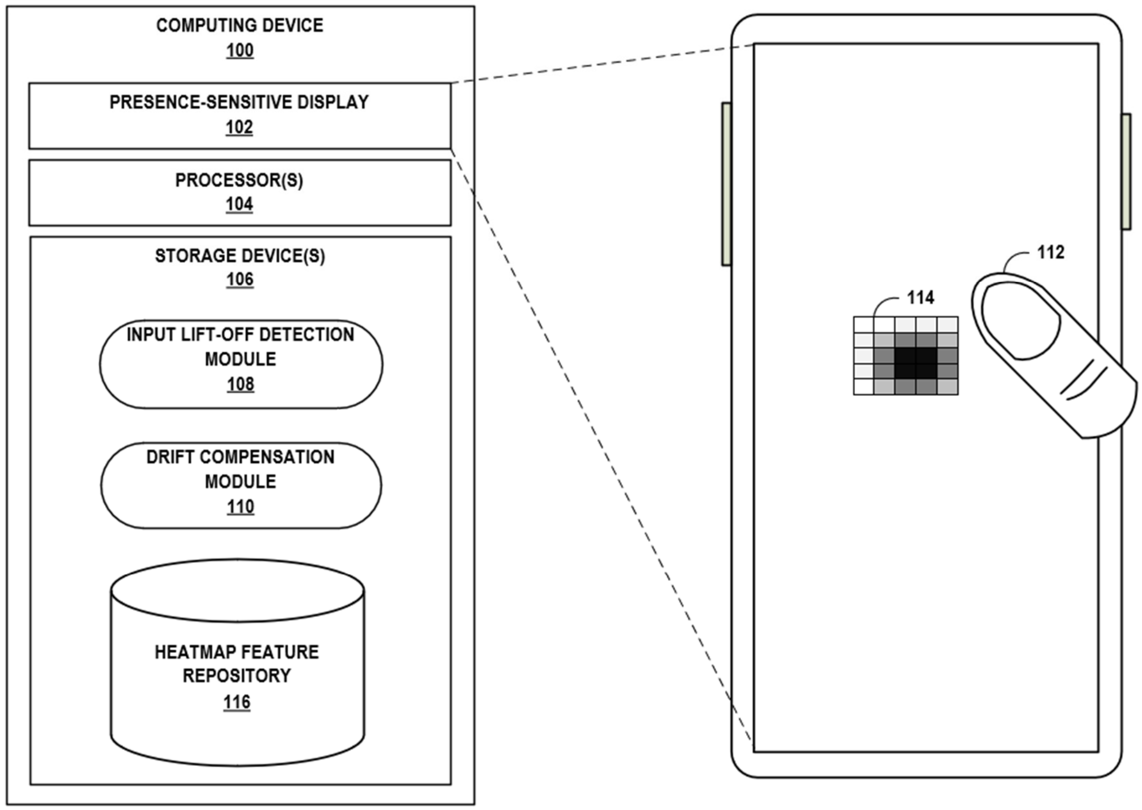

FIG. 1

Presence-sensitive display 102 may function as an input device for computing device 100 using a touchscreen, pressure sensitive screen, an acoustic pulse recognition touchscreen or another presence-sensitive screen technology. Presence-sensitive display 102 may function as an output device for computing device 100 using any one or more of display technologies (e.g., liquid crystal display, dot matrix display, light emitting diode (LED) display, organic light emitting diode (OLED) display, electronic ink display, or similar display technology.

Processors 104 may implement functionality and/or execute instructions associated with computing device 100. Examples of processors 104 include one or more of an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), an application processor, a display controller, an auxiliary processor, a central processing unit (CPU), a graphics processing unit (GPU), one or more sensor hubs, and any other hardware configure to function as a processor, a processing unit, or a processing device.

Storage devices 106 may include one or more computer-readable storage media. For example, storage devices 106 may be configured for long-term, as well as short-term storage of information, such as instructions, data, or other information used by computing device 100. In some examples, storage devices 106 may include non-volatile storage elements. Examples of such non-volatile storage elements include magnetic hard disks, optical discs, solid state discs, and/or the like. In other examples, in place of, or in addition to the non-volatile storage elements, storage devices 106 may include one or more so-called “temporary” memory devices, meaning that a primary purpose of these devices may not be long-term data storage. For example, the devices may comprise volatile memory devices, meaning that the devices may not maintain stored contents when the devices are not receiving power. Examples of volatile memory devices include random-access memories (RAM), dynamic random-access memories (DRAM), static random-access memories (SRAM), etc.

Presence-sensitive display 102 of computing device 100 may detect an input (e.g., provided by an input element 112, such as a finger) and may provide a plurality of locations at which the input is detected along with an intensity of the input detected at each of the plurality of locations. In various instances, the presence-sensitive display may continue to detect the input as input element 112 moves to different locations of presence-sensitive display 102. In such instances, the input is called a gesture input. When input element 112 is no longer detected by presence-sensitive display 102 (e.g., when input element 112 moves away from presence-sensitive display 102), computing device 100 determines that the gesture input has ceased (e.g., determines that a lift-off event has occurred). Computing device 100 may perform various actions based on the location at which the gesture input terminated.

As input element 112 moves relative to presence-sensitive display 102, computing device 100 may be unable to precisely determine the terminal location of the gesture input. For example, when using computing device 100, a user may experience issues with the accuracy of an input and may experience unintended drifting of a cursor or other selector when removing input element 112 from the range of presence-sensitive display 102. The accuracy and drifting issues may be due to a variety of factors including shifts in pressure or capacitance detected or changes in the contact area on the touchscreen among other factors.

In accordance with the techniques described herein, an input lift-off detection module 108 may analyze changes in a gesture input heatmap 114 (“heatmap 114”) during the gesture and during the lift-off event, and a drift compensation module 110 may adjust the terminal location of the gesture input based on the analysis by input lift-off detection module 108. In this way, the techniques described herein may enable computing device 100 to more accurately determine the location at which the gesture terminated, in turn increasing the probability that computing device 100 performs the action intended by a user of computing device 100.

FIG. 1 shows computing device 100 with heatmap 114 visually overlaid on presence-sensitive display 102 of computing device 100. Although FIG. 1 shows heatmap 114 for purposes of explanation, computing device 100 may not generate or display a visual representation of heatmap 114 via presence-sensitive display 102. Heatmap 114 is one example of the intensity of input detected at various locations of presence-sensitive display 102 at one particular time. In examples where presence-sensitive display 102 is a capacitive touchscreen, the intensity of the input may correspond to a capacitance value at each location at which the capacitive touchscreen detects the input. For instance, the darker colored portions of heatmap 114 may indicate a higher capacitance value while the lighter colored portions may indicate a lower capacitance value at those locations. In general, heatmap 114 may indicate higher intensities near the center of the locations at which input element 112 is detected and lower intensities near the periphery of the locations at which input element 112 is detected. Heatmap repository 116 may store information relating to features of heatmap 114, such as a two-dimensional (2D) raw heatmap corresponding to heatmap 114, an area of heatmap 114, a perimeter of heatmap 114, a histogram of the row and column values of heatmap 114, the sum of the heatmap values of heatmap 114, derivatives of the above features, etc.

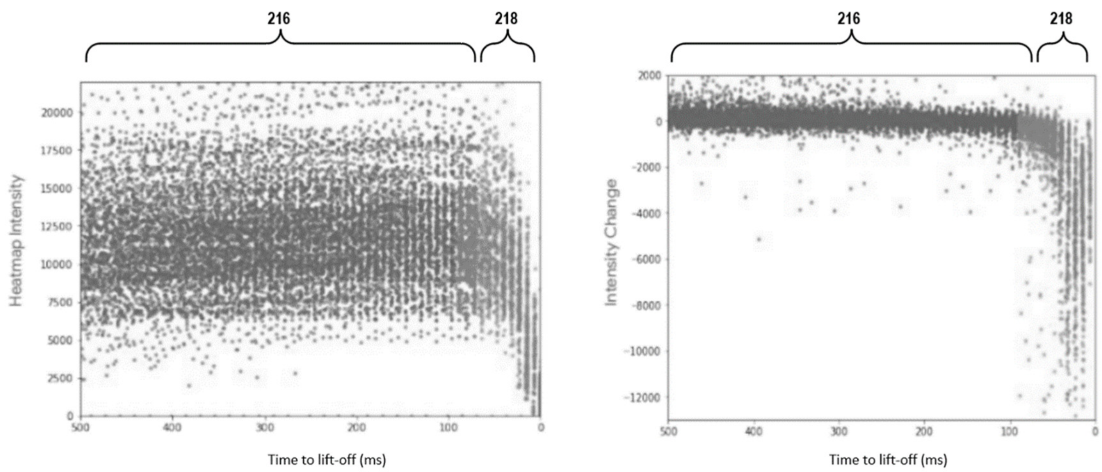

FIG. 2A FIG. 2B

FIG. 2A shows heatmap intensity as a function of time for a gesture input, and FIG. 2B shows intensity change as a function of time for the gesture input. The gesture input includes a gesture event 218 and a lift-off event 220. The end of gesture event 218 and/or the start of lift-off event 220 may indicate when lift-off starts to occur. As shown in FIGS. 2A-2B, the intensity of an input detected at various locations of presence-sensitive display 102 at any particular time during gesture event 218 is relatively high and mostly constant (e.g., because as the user performs the gesture, the user presses input element 112 into presence-sensitive display 102, resulting in a commensurate intensity of the input). As further shown in FIGS. 2A-2B, the intensity of an input detected at various locations of presence-sensitive display 102 at any particular time during lift-off event 220 is relatively low and progressively decreasing (e.g., because as the user begins to terminate the gesture, the user lifts input element 112 off of presence-sensitive display 102, reducing the intensity of the input until the intensity reaches zero).

Input lift-off detection module 108 may analyze changes in heatmap 114 over a temporal domain (e.g., as shown in FIGS. 2A-2B) to determine when a lift-off event has occurred. As an example, if, when a user is providing a gesture input to computing device 100, heatmap 114 indicates that a decrease in the heatmap intensities exceeds a threshold value (e.g., an absolute value, a percentage value, etc.), input lift-off detection module 108 may render a classification of lift-off event 220.

For instance, if at 500 ms to lift-off the average heatmap intensity is 12,500 units, at 480 ms to lift-off the average heatmap intensity is 12,600 units, and the threshold value is 20% (e.g., of the heatmap intensity at the earlier time, in this case 12,500 units), input lift-off detection module 108 may classify the temporal domain of 500-480 ms to lift-off as gesture event 218 instead of lift-off event 220. In the same example, if at 80 ms to lift-off the average heatmap intensity is 12,500 units and at 60 ms to lift-off the average heatmap intensity is 10,000 units, input lift-off detection module 108 may classify the temporal domain of 80-60 (as well as 60-0 ms to lift-off) to lift-off as lift-off event 220.

Although the above example describes using an average of heatmap intensities, the techniques of this disclosure may be applied using other features of heatmap 114. Examples of such features may include, but are not limited to, a mean, a median, a sum, etc. Also, time intervals other than 20 ms may be used, and the times being analyzed within those intervals may be selected in various ways. For instance, input lift-off detection module 108 may analyze heatmap intensities for two or more randomly selected times within an interval of 10 ms.

In some examples, input lift-off detection module 108 may use a sliding window analysis. The sliding window analysis may be substantially similar to the method described in the above example except that a sliding window is used. As used here, a sliding window refers to a number of the most recent contiguous frames (e.g., the 3 most recent frames, the 5 most recent frames, etc.). Additionally or alternatively, input lift-off detection module 108 may use a recurrent neural network, a support vector machine (SVM), logistic regression models, or other algorithms to determine the occurrence of the lift-off event.

For example, input lift-off detection module 108 may include a recurrent neural network trained to output a determination of when lift-off event 220 has occurred. The recurrent neural network may receive the time-series data of the heatmap features, including heatmap intensities, as sequential input data, which the recurrent neural network may analyze to output a determination. In some instances, the sequential input data may be pre-processed (e.g., filtered) to exclude data associated with changes in heatmap intensities below a threshold (which, in general, do not indicate lift-off event 220), such as a threshold of a 10% decrease in heatmap intensity in a time interval or frame window.

In any case, based on the analysis by input lift-off detection module 108, drift compensation module 110 may adjust the terminal location of the gesture input. For example, if, based on the analysis by input lift-off detection module 108, the start of lift-off event 220 occurs 80 ms before lift-off (i.e., 80 ms before input element 112 actually lifts off presence-sensitive display 102), drift compensation module 110 may report the touch centroid of heatmap 114 associated with the frame from 80 ms before lift-off as the terminal location of the gesture input, which computing device 100 (e.g., applications of computing device 100) may process accordingly. In another example, drift compensation module 110 may apply a Kalman filter to estimate the drift that occurred during lift-off event 218 and adjust the detected terminal location of the gesture input by the estimated draft. In yet another example, drift compensation module 110 may use a specific flag to label the frames associated with lift-off event 218, allowing computing device 100 to process those frames accordingly (e.g., by using an algorithm not described here).

It is noted that the techniques of this disclosure may be combined with any other suitable techniques or combination of techniques. As one example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Publication No. 2018/0329542A1. In an additional example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Publication No. 2020/0142582A1. In yet another example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Publication No. 2020/0387256A1.