



(51) International Patent Classification:

G06F 3/0488 (2013.01) G06F 3/01 (2006.01)
G06F 3/048 (2006.01)

(21) International Application Number:

PCT/US2016/019779

(22) International Filing Date:

26 February 2016 (26.02.2016)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

14/671,726 27 March 2015 (27.03.2015) US

(71) Applicant: INTEL CORPORATION [US/US]; 2200 Mission College Boulevard, Santa Clara, California 95054 (US).

(72) Inventors: MALOO, Raghvendra; C-1061, SRRI Campus, Bangalore, KA 560103 (IN). VALAVI, John J.; 7178 NE Rockridge Place, Hillsboro, Oregon 97124 (US). HSIN, Chih-Fan; 3566 NW Bronson Crest Loop, Portland, Oregon 97229 (US). SUBRAMANIAM, Gokul V.;

#23-56P, Outer Ring Road, Devarabeesanahalli, Varther Hobli, Bellandur Post, Bangalore, KA 560103 (IN).

(74) Agent: KELLETT, Glen M.; Barnes & Thornburg LLP, c/o CPA Global, P.O. Box 52050, Minneapolis, Minnesota 55402 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE,

[Continued on next page]

(54) Title: TECHNOLOGIES FOR GRAPHICAL USER INTERFACE MANIPULATIONS USING MULTI-FINGER TOUCH INTERACTIONS

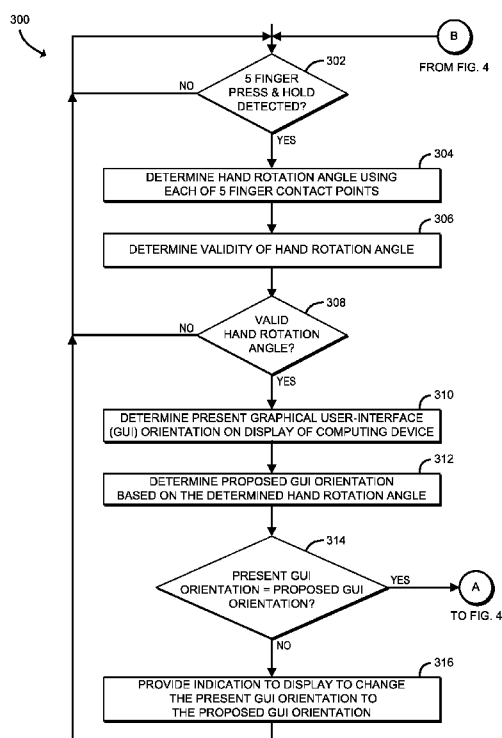


FIG. 3

(57) Abstract: Technologies for manipulating a graphical user interface (GUI) of a display of a computing device include a touch screen of the display capable of receiving user input via physical contact of the user's fingers. The computing device determines whether the user has initiated multi-finger input gesture from which and a hand rotation angle may be determined based on touch points corresponding to the physical contact between the user's fingers and the touch screen. Based on the hand rotation angle, the computing device may take an action on the GUI, such as rotate an orientation of the GUI and/or display a virtual keyboard on the GUI relative to the hand rotation angle and/or the touch points. Other embodiments are described herein and claimed.



DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT,
LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE,
SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA,
GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

— *as to the applicant's entitlement to claim the priority of
the earlier application (Rule 4.17(iii))*

Published:

— *with international search report (Art. 21(3))*

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted
a patent (Rule 4.17(ii))*

TECHNOLOGIES FOR GRAPHICAL USER INTERFACE MANIPULATIONS USING MULTI-FINGER TOUCH INTERACTIONS

CROSS-REFERENCE TO RELATED APPLICATION

[0001] The present application claims priority to U.S. Utility Patent Application Serial No. 14/671,726, entitled “TECHNOLOGIES FOR GRAPHICAL USER INTERFACE MANIPULATIONS USING MULTI-FINGER TOUCH INTERACTIONS,” which was filed on March 27, 2015.

BACKGROUND

[0002] Computing devices typically include various methods for supporting user interaction (i.e., receiving input from a user). For example, a computing device may support receiving user input via an input device, such as a mouse, touchpad, and/or a keyboard. Modern computing devices, such as smartphones, tablets, and all-in-one computing devices, are additionally or alternatively, capable of receiving user input via an electronic display (i.e., a touchscreen display) using various technologies, such as resistive, surface acoustic wave, and capacitive to support input via touch. As a result, a user can control the computing device by touching the touchscreen display using one or more fingers (e.g., a single finger touch, multi-finger touch, etc.). Touchscreen displays allow users to interact directly with a graphical user interface (GUI) displayed on the touchscreen displays through various touch-based input gestures (e.g., flicks, pinches, press-and-holds, press and taps, swipes, etc.). To facilitate the user interaction, certain GUI elements (e.g., graphical icons, visual indicators, virtual soft keys, etc.) may be displayed (i.e., rendered) on the touchscreen display. Such GUI elements allow the user to directly interact with applications running on the computing device.

[0003] Further, as modern computing devices have become smaller, mobile, and capable of being handheld during operation, the orientation of the GUI (e.g., soft keys, application frames, etc.) on the computing device displays can become skewed as the user rotates the computing device (i.e., the display). Traditional computing devices, or displays thereof, typically only allowed the user to change the orientation of the GUI display between landscape and portrait modes through a manual process, such as via an orientation setting of the display. However, enhancements in microelectromechanical devices have led to a number of computing devices being outfitted with various components, such as accelerometers, which may allow certain computing devices to automatically change the GUI display orientation when the user rotates the computing device.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] The concepts described herein are illustrated by way of example and not by way of limitation in the accompanying figures. For simplicity and clarity of illustration, elements illustrated in the figures are not necessarily drawn to scale. Where considered appropriate, reference labels have been repeated among the figures to indicate corresponding or analogous elements.

[0005] FIG. 1 is a simplified block diagram of at least one embodiment of a system for manipulating a graphical user interface (GUI) of a computing device using multi-finger touch interactions on a display of the computing device;

[0006] FIG. 2 is a simplified block diagram of at least one embodiment of an environment that may be established by the computing device of FIG. 1;

[0007] FIG. 3 is a simplified flow diagram of at least one embodiment of a method for changing a GUI orientation of a display of the computing device of FIG. 2 during a multi-finger touch interaction that may be executed by the computing device of FIG. 2;

[0008] FIG. 4 is a simplified flow diagram of at least one embodiment of a method for displaying a virtual keyboard on a display of the computing device of FIG. 2 during a multi-finger touch interaction that may be executed by the computing device of FIG. 2;

[0009] FIG. 5 is a simplified flow diagram of at least one embodiment of a method for determining a hand rotation angle during a multi-finger touch interaction that may be executed by the computing device of FIG. 2;

[0010] FIG. 6 is a simplified flow diagram of at least one embodiment of a method for validating a hand rotation angle during a multi-finger touch interaction that may be executed by the computing device of FIG. 2;

[0011] FIG. 6 is a simplified flow diagram of at least one embodiment of a method for validating a hand rotation angle during a multi-finger touch interaction that may be executed by the computing device of FIG. 2;

[0012] FIG. 7 is a simplified flow diagram of at least one embodiment of a method for determining a proposed GUI orientation of a display of the computing device of FIG. 2 based on a hand rotation angle during a multi-finger touch interaction that may be executed by the computing device of FIG. 2;

[0013] FIG. 8 is a simplified illustration of a view of at least one embodiment of touch point vectors based on fingertips in contact with a display of the computing device of FIG. 2 used to determine a rotation vector that may be used by the computing device of FIG. 2 to determine a hand rotation angle during a multi-finger touch interaction;

[0014] FIGS. 9A-9C are simplified illustrations of an orientation of a display of the computing device of FIG. 2 being adjusted during a multi-finger touch interaction;

[0015] FIGS. 10A-10C are simplified illustrations of a virtual keyboard on a display of the computing device of FIG. 2 being displayed during a multi-finger touch interaction;

[0016] FIG. 11 is a simplified illustration of a context-specific virtual keyboard on a display of the computing device of FIG. 2 having active finger contacts being displayed during a multi-finger touch interaction;

[0017] FIG. 12 is a simplified illustration of another context-specific virtual keyboard on a display of the computing device of FIG. 2 having active finger contacts being displayed during a multi-finger touch interaction; and

[0018] FIG. 13 is a simplified mapping table of illustrative applications that may be associated with default soft keys of the context-specific virtual keyboard of FIG. 12.

DETAILED DESCRIPTION OF THE DRAWINGS

[0019] While the concepts of the present disclosure are susceptible to various modifications and alternative forms, specific embodiments thereof have been shown by way of example in the drawings and will be described herein in detail. It should be understood, however, that there is no intent to limit the concepts of the present disclosure to the particular forms disclosed, but on the contrary, the intention is to cover all modifications, equivalents, and alternatives consistent with the present disclosure and the appended claims.

[0020] References in the specification to “one embodiment,” “an embodiment,” “an illustrative embodiment,” etc., indicate that the embodiment described may include a particular feature, structure, or characteristic, but every embodiment may or may not necessarily include that particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the same embodiment. Further, when a particular feature, structure, or characteristic is described in connection with an embodiment, it is submitted that it is within the knowledge of one skilled in the art to effect such feature, structure, or characteristic in connection with other embodiments whether or not explicitly described. Additionally, it should be appreciated that items included in a list in the form of “at least one of A, B, and C” can mean (A); (B); (C); (A and B); (A and C); (B and C); or (A, B, and C). Similarly, items listed in the form of “at least one of A, B, or C” can mean (A); (B); (C); (A and B); (A and C); (B and C); or (A, B, and C).

[0021] The disclosed embodiments may be implemented, in some cases, in hardware, firmware, software, or any combination thereof. The disclosed embodiments may also be

implemented as instructions carried by or stored on one or more transitory or non-transitory machine-readable (e.g., computer-readable) storage media, which may be read and executed by one or more processors. A machine-readable storage medium may be embodied as any storage device, mechanism, or other physical structure for storing or transmitting information in a form readable by a machine (e.g., a volatile or non-volatile memory, a media disc, or other media device).

[0022] In the drawings, some structural or method features may be shown in specific arrangements and/or orderings. However, it should be appreciated that such specific arrangements and/or orderings may not be required. Rather, in some embodiments, such features may be arranged in a different manner and/or order than shown in the illustrative figures. Additionally, the inclusion of a structural or method feature in a particular figure is not meant to imply that such feature is required in all embodiments and, in some embodiments, may not be included or may be combined with other features.

[0023] Referring now to FIG. 1, in an illustrative embodiment, a system 100 for manipulating a graphical user interface (GUI) of a display using multi-finger touch interactions includes a computing device 102 with a display 120 that supports receiving input from a user 104 via a touch screen 122. As will be described in further detail, the display 120 includes the touch screen 122 for receiving contact from the user 104, such as via a fingertip of the user 104. Accordingly, touch points corresponding to those contacts may be used to control the computing device 102 and software applications running thereon based on user interactions with the touch screen 122. To do so, the display 120 is capable of displaying GUI elements (e.g., graphical icons, visual indicators, virtual soft keys, etc.) corresponding to software applications running on the computing device 102, which allow the user 104 to interact with the software applications. The user interactions are typically performed via direct manipulation of the GUI elements.

[0024] Additionally, certain software applications may interpret certain user interactions, such as motions made with one or more fingers on the display 120, commonly referred to as input gestures. Typically, the GUI elements are rendered on the display 120, whereby the user can interact with the GUI elements via the input gestures. However, certain input gestures may not be specific to a particular GUI element (e.g., where no actionable GUI element is displayed to the user), but rather to a particular software application. The input gestures may include various flicks, pinches, press-and-holds, press and taps, swipes, and any other number of input gestures. Generally, supported input gestures are mapped to predefined functionality of the particular software application that is responsible for taking an action based

on the detected input gestures. In other words, input gestures may cause a software application running on the computing device 102 to execute a particular predefined action based on the detection of the input gesture. For example, a pinch gesture may zoom in on an image or a swipe gesture in a particular direction may navigate backward/forward through a gallery of images.

[0025] Some input gestures, such as the pinch gesture, require multi-finger touch interactions. Other multi-finger touch interactions may include a five-finger contact gesture, in which each of the five fingers (i.e., four fingers and a thumb) of one of the user's 104 hands makes and maintains contact with the touch screen 122 for a duration that exceeds a predetermined duration (i.e., a five-finger press-and-hold). Upon detecting the five-finger press-and-hold contact gesture, the computing device 102 may determine an angle at which the hand of the user 104 is interfacing with the computing device 102, based on the contact points, or touch points, of the fingers of the five-finger press-and-hold contact gesture.

[0026] As discussed in more detail below, in some embodiments, the computing device 102 may perform certain actions upon determining the hand angle of the user 104. For example, the GUI is typically displayed in either a landscape orientation or a portrait orientation relative to the display 120. In some embodiments, when the five-finger press-and-hold contact gesture is detected, the angle at which the hand of the user is interfacing with the computing device 102 may be used to adjust the GUI orientation of the display 120 based on the angle at which the user's 104 hand is interfacing with the computing device 102. Additionally or alternatively, in some embodiments, the computing device 102 may display a virtual keyboard that may be comprised of a number of soft keys for receiving user input and performing a particular task based on the touched soft key or input gesture received. Further, the virtual keyboard may be displayed at a size, location, and/or angle on the display 120 relative to the contact points of the five-finger press-and-hold contact gesture, such that the virtual keyboard is ergonomically shaped and/or located based on the angle at which the user's 104 hand is interfacing with the computing device 102.

[0027] The computing device 102 may be embodied as any type of computing device capable of performing the functions described herein. For example, the computing device 102 may be embodied as, without limitation, a laptop computer, a notebook computer, a tablet computer, a smartphone, a wearable computing device, a computer, a workstation, a server, a multiprocessor system, a consumer electronic device, a digital television device, a smart appliance, an all-in-one computing device, and/or any other computing device configured to support user input via touch. As shown in FIG. 1, the illustrative computing device 102

includes a processor 110, an input/output (I/O) subsystem 112, a memory 114, a data storage device 116, communication circuitry 118, and the display 120. Of course, the computing device 102 may include other or additional components, such as those commonly found in a computing device (e.g., a graphics processing unit), in other embodiments. Additionally, in some embodiments, one or more of the illustrative components may be incorporated in, or otherwise form a portion of, another component. For example, the memory 114, or portions thereof, may be incorporated in one or more processors 110 in some embodiments.

[0028] The processor 110 may be embodied as any type of processor capable of performing the functions described herein. The processor 110 may be embodied as a single or multi-core processor(s), digital signal processor, microcontroller, or other processor or processing/controlling circuit. The memory 114 may be embodied as any type of volatile or non-volatile memory or data storage capable of performing the functions described herein. In operation, the memory 114 may store various data and software used during operation of the computing device 102 such as operating systems, applications, programs, libraries, and drivers. The memory 114 is communicatively coupled to the processor 110 via the I/O subsystem 112, which may be embodied as circuitry and/or components to facilitate input/output operations with the processor 110, the memory 114, and other components of the computing device 102. For example, the I/O subsystem 112 may be embodied as, or otherwise include, memory controller hubs, input/output control hubs, integrated sensor hubs, firmware devices, communication links (i.e., point-to-point links, bus links, wires, cables, light guides, printed circuit board traces, etc.) and/or other components and subsystems to facilitate the input/output operations. In some embodiments, the I/O subsystem 112 may form a portion of a system-on-a-chip (SoC) and be incorporated, along with the processors 110, the memory 114, and other components of the computing device 102, on a single integrated circuit chip.

[0029] The data storage device 116 may be embodied as any type of device or devices configured for short-term or long-term storage of data such as, for example, memory devices and circuits, memory cards, hard disk drives, solid-state drives, or other data storage devices. The communication circuitry 118 of the computing device 102 may be embodied as any communication circuit, device, or collection thereof, capable of enabling network communications to and from the computing device 102 via a network. The communication circuitry 118 may be configured to use one or more communication technologies to support wired and/or wireless communications, and associated protocols (e.g., Ethernet, Bluetooth®, Wi-Fi®, WiMAX, etc.) to effect such communication.

[0030] The display 120 may be embodied as any type of display device configured to display digital information, such as a liquid crystal display (LCD), a light emitting diode (LED), a plasma display, a cathode ray tube (CRT), or other type of display device. The touch screen 122 may be embodied as any type of touch screen capable of generating input data in response to being touched by the user 104 of the computing device 102. The touch screen 122 may be embodied as a resistive touch screen, a capacitive touch screen, a camera-based touch screen, or the like. In some embodiments, the touch screen 122 may be comprised of a touch screen sensor responsive to multiple simultaneous touch points and a controller-based firmware to interpret the input gestures resulting from the touch points and interface with various applications capable of being executed on the computing device 102, such as via an application program interface (API).

[0031] Referring now to FIG. 2, in an embodiment, the computing device 102 establishes an environment 200 during operation. The illustrative environment 200 includes a finger touch detection module 210, an input gesture determination module 220, a hand orientation determination module 230, a GUI orientation management module 240, and a virtual keyboard display management module 250. Each of the modules, logic, and other components of the environment 200 may be embodied as hardware, software, firmware, or a combination thereof. For example, each of the modules, logic, and other components of the environment 200 may form a portion of, or otherwise be established by, a processor or other hardware components of the computing device 102. As such, in some embodiments, one or more of the modules of the environment 200 may be embodied as a circuit or collection of electrical devices (e.g., finger touch detection circuit, an input gesture determination circuit, etc.). In the illustrative environment 200, the computing device 102 includes virtual key mapped data 202 and hand rotation vector data 204, each of which may be accessed by the various modules of the computing device 102. It should be appreciated that the computing device 102 may include other components, sub-components, modules, and devices commonly found in a computing device, which are not illustrated in FIG. 2 for clarity of the description.

[0032] The finger touch detection module 210 is configured to receive contact data from the touch screen 122 and determine a touch point for each contact with the touch screen 122 based on the received contact data. For example, the finger touch detection module 210 may detect that one or more fingers (i.e., touch points) are in contact with the touch screen 122 and determine a central point of contact (i.e., a centroid of the touch point) for each finger. In some embodiments, one of more functions performed by the finger touch detection module 210 may be performed by sub-modules, for example, by a finger press detection module 212, a touch

point determination module 214, and a finger release detection module 216. In such embodiments, the finger press detection module 212 may be configured to detect a presence of the user's 104 finger(s) on the touch screen 122. Additionally or alternatively, in such embodiments, the touch point determination module 214 may be configured to determine touch points and the centroid of the touch points based on the detected presence of the user's 104 fingers on the touch screen. The finger release detection module 216, in such embodiments, may be configured to detect when the presence of the user's 104 finger(s) on the touch screen 122 are released.

[0033] The input gesture determination module 220 is configured to determine whether an input gesture is being performed by the user 104 based on the one or more touch points. In other words, the input gesture determination module 220 may monitor the touch point(s) to determine whether the user 104 has input an input gesture. To do so, the input gesture determination module 220 may track the touch point(s) over time to determine whether the user 104 has performed an input gesture. For example, in determining a press-and-hold gesture, the input gesture determination module 220 may determine whether the touch point(s) have remained in contact with the touch screen 122 for at least a predetermined period of time sufficient to indicate an intention of the user 104 to perform the press-and-hold input gesture.

[0034] The input gesture determination module 220 may contain various sub-modules for determining the input gestures. One such input gesture determining submodule includes a five-finger press-and-hold determination module 222. The five-finger press-and-hold determination module 222 is configured to determine whether five touch points, corresponding to five finger contacts, have remained in contact with the touch screen 122 for a predetermined period of time (i.e., a five-finger press-and-hold gesture). In some embodiments, the five-finger press-and-hold determination module 222 may be further configured to determine whether the five finger touch points correspond to five fingers of a single hand of the user 104. To do so, in some embodiments, the five-finger press-and-hold determination module 222 may determine whether four of the five touch points closest to each other can be fitted into a concave curve as discussed in more detail below. It should be appreciated that, in some embodiments, the input gesture determination module 220 may receive touch point data from the finger touch detection module 210.

[0035] The hand orientation determination module 230 is configured to determine a hand orientation based on the one or more touch points. To do so, the hand orientation determination module 230 includes a hand rotation angle determination module 232 and a confidence level determination module 234. The hand rotation angle determination module 232

is configured to determine a hand rotation angle based on the one or more touch points. To determine the hand rotation angle, the hand rotation angle determination module 232 determines a centroid and a centroidal axes (i.e., an x-axis and a y-axis) through the centroid based on each of the one or more touch points and a touch point vector for each touch point to the centroid position, and performs a summation of the determined touch point vectors to determine a rotation vector. The hand rotation angle determination module 232 may be further configured to determine the hand rotation angle based on an angle of the rotation vector from a positive x-axis through the centroid.

[0036] In some embodiments, the hand rotation angle determination module 232 may be further configured to determine a touch point that corresponds to a thumb position and apply a weight to the thumb position. To do so, in some embodiments, the hand rotation angle determination module 232 may determine the touch point vector having the furthest distance from the centroid position. In such embodiments, the hand rotation angle determination module 232 may apply the weight may be applied to the touch point position that is determined by the hand rotation angle determination module 232 to correspond to a thumb and add the weight to the rotation vector. It should be appreciated that, in some embodiments, the hand orientation determination module 230 may determine the hand orientation in response to an input gesture, such as the five-finger press-and-hold gesture, being determined. In such embodiments, the hand orientation determination module 230 may receive the determined input gesture from the input gesture determination module 220.

[0037] The confidence level determination module 234 is configured to determine a confidence level of the determined hand rotation angle. To do so, the confidence level determination module 234 may determine which three touch point vectors, of the five touch point vectors, have the shortest distance to the centroid. The confidence level determination module 234 may additionally calculate an angle to the determined rotation vector for each of the three shortest touch point vectors. As will be further described below, the confidence level determination module 234 may determine the confidence level of the hand orientation angle based on whether each of the angles is determined to be less than a maximum angle (e.g., 90°).

[0038] In some embodiments, historical hand orientation angles and corresponding confidence levels may be stored, such as in the hand rotation vector data 204, for the hand rotation angle determination module 232 and/or the confidence level determination module 234 to perform heuristics on the determined hand rotation angle and/or the confidence level of the determined hand rotation angle to adjust the weight, the maximum angle, and/or the confidence level threshold over time. It should be appreciated that while the five-finger press-and-hold

input gesture may be used to determine the hand orientation, it is contemplated that any multi-finger input gesture in which the hand orientation may be determined may be used by the hand orientation determination module 230 to determine the hand orientation based on the multiple touch points.

[0039] The GUI orientation management module 240 is configured to manage the orientation of the GUI on the display 120. To do so, the GUI orientation management module may determine a present orientation of the GUI on the display 120 and a difference between the present orientation of the GUI and the hand orientation angle. The GUI orientation management module 240 may be further configured to determine a GUI orientation rotation angle based on the difference between the present orientation of the GUI and the hand orientation angle. In other words, the GUI orientation management module 240 may be configured to rotate the present orientation of the GUI on the display 120 to a desired GUI orientation based on the hand orientation angle. Accordingly, the GUI orientation management module 240 may be further configured to provide an indication to the display 120, or a component (e.g., the processor 110, a graphics processing unit, a display driver, etc.) of the computing device 102 that controls the GUI orientation on the display 120, to rotate the GUI orientation based on the desired GUI orientation. In some embodiments, one of more functions performed by the GUI orientation management module 240 may be performed by sub-modules, such as a GUI orientation determination module 242 and a GUI orientation adjustment module 244, for example.

[0040] The virtual keyboard display management module 250 is configured to manage the display of a virtual keyboard on the GUI of the display 120. To do so, the virtual keyboard display management module 250 is configured to display the virtual keyboard at a position on the GUI based on the hand rotation angle and/or the touch points. For example, a virtual keyboard may be displayed on the GUI at an angle corresponding to the hand orientation angle and at a position, size, and/or shape corresponding to the touch points. The virtual keyboard display management module 250 is additionally configured to provide an interface between virtual keys of virtual keyboard and the application interfacing commands mapped to the virtual keys. In some embodiments, a default set of keys may be mapped to corresponding default actions. In such embodiments, a mapping table may be stored in the virtual key mapped data 202.

[0041] In some embodiments, the virtual keyboard display management module 250 may be configured to display a virtual keyboard that is context sensitive to an application or an active process running on the computing device 102. In other words, the contents (e.g., soft

keys, visual indicators, and/or the like) of the virtual keyboard may correspond to an application or process presently being executed on the computing device 102 at the time of the touch point detection (e.g., the five-finger input gesture). The application may be identified by any means to identify the application for intended use of the virtual keyboard, such as a z-order of the presently running applications (i.e., a top-most application), an application directly underlying the touch points, a type of media underlying the touch points, etc.

[0042] In embodiments wherein the virtual keyboard is context sensitive, the virtual keyboard display management module 250 may communicate with the application, such as via an API, to determine the soft keys, visual indicators, graphical icons, etc. to display. An example of which is shown in FIGS. 12 and 13, which will be described in further detail. In some embodiments, one or more functions performed by the virtual keyboard display management module 250 may be performed by one or more sub-modules, such as a virtual keyboard display toggle module 252, an application determination module 254, and an application interfacing module 256, for example.

[0043] Referring now to FIG. 3, in use, the computing device 102 may execute a method 300 for changing a GUI orientation of the display 120. The illustrative method 300 begins at block 302, in which the computing device 102 determines whether a five-finger-press-and-hold input gesture was detected. In other words, the computing device 102 determines whether five finger contact points, or touch points, have been detected by the display 120 (i.e., by the touch screen 122 of the display 120) for a predetermined period of time, thereby indicating an intentional press and hold input gesture. If the five-finger-press-and-hold input gesture was not detected, the method 300 loops back to block 302 to continue determining whether the five-finger-press-and-hold input gesture was detected.

[0044] If the five-finger-press-and-hold input gesture was detected, the method 300 advances to block 304. At block 304, the computing device 102 determines a hand rotation angle using each of the five finger contact points of the five-finger-press-and-hold input gesture. An example of which is shown in FIG. 5, which is discussed further below. At block 306, the computing device 102 determines a validity of the hand rotation angle (i.e., a hand rotation angle validity check) determined at block 304. In other words, the computing device 102 determines whether it has confidence in the determined hand rotation angle (see, e.g., method 600 of FIG. 6). At block 308, the computing device 102 determines whether the hand rotation angle is valid based on the validity of the hand rotation angle determined at block 306. If the hand rotation angle is determined to not be valid (i.e., the computing device 102 has determined it does not have enough confidence in the determined hand rotation angle to proceed), the

method 300 loops back to block 302, wherein the computing device 102 continues to monitor whether the five-finger-press-and-hold input gesture was detected.

[0045] If the hand rotation angle is determined to be valid, the method 300 advances to block 310. At block 310, the computing device 102 determines a present GUI orientation of the display 120 of the computing device 102. In other words, the computing device 102 determines the orientation at which the GUI is presently displayed on the display 120. At block 312, the computing device 102 determines a proposed orientation of the GUI based on the determined hand rotation angle. At block 314, the computing device 102 determines whether the present GUI orientation is equal to the proposed GUI orientation. If not, the method 300 advances to block 316, wherein the computing device 102 provides an indication to the display 120 to change the presently displayed GUI orientation to the proposed GUI orientation before looping back to block 302 to determine whether a five-finger-press-and-hold input gesture was detected.

[0046] If the computing device 102 determines the present GUI orientation is equal to the proposed GUI orientation at block 314, the computing device 102 advances to block 318, which is shown in FIG. 4. At block 318, the computing device 102 determines a user interfacing GUI application presently being executed on the computing device 102 to associate with a virtual keyboard. In some embodiments, at block 320, the computing device 102 may determine the user interfacing GUI application based on a z-order of applications presently running on the computing device 102. For example, the computing device 102 may determine a top-most GUI of an application whose z-order along a z-axis indicates that that application is in the foreground (i.e., on top, above, in front of, etc.) all the other GUIs of the other presently running applications. Additionally or alternatively, in some embodiments, at block 322, the computing device 102 may determine the user interfacing GUI application based on the finger contact points. For example, the computing device 102 may use the finger contact points to determine a proximate relationship between the finger contact points and each GUI of the presently running applications resides. For example, the computing device may determine which GUI of the presently running applications resides generally underneath the finger contact points, such as by determining which GUI a majority of the finger contact points are in contact with, or hovering over, depending on the z-order of the underlying GUI.

[0047] At block 324, the computing device 102 displays the virtual keyboard on the display 120. In some embodiments, at block 326, the computing device 102 displays the virtual keyboard on the display 120 at a location based on the finger contact points. For example, in some embodiments, the virtual keyboard may be located on the display directly under, or slightly above, the finger contact points. Additionally, in some embodiments, other

configuration attributes of the virtual keyboard, such as size, display orientation angle, and/or shape, may also be based on the finger contact points. In some embodiments, at block 328, the computing device 102 displays the virtual keyboard on the display based on the user interfacing GUI application determined at block 318.

[0048] At block 330, the computing device 102 determines whether one or more previously contacting finger contact points were release. In other words, the computing device 102 determines whether any of the five-finger-press-and-hold input gesture initiating contact points were released. If the computing device 102 determines at block 330 that a previously contacting finger contact point was not released, the method 300 branches to block 336, which is described further below. If the computing device 102 determines that a previously contacting finger contact point was released, the method advances to block 332. At block 332, the computing device 102 determines whether to hide the virtual keyboard. For example, in some embodiments, the computing device 102 may determine to hide the virtual keyboard if the virtual keyboard has not received a user input for a predetermined period of time and/or if all five finger contact points have been released (i.e., all five finger contact points are no longer in contact with touch screen 122 of the display 120).

[0049] If the computing device 102 determines the virtual keyboard is to be hidden, the computing device 102 hides the virtual keyboard. The method 300 subsequently loops back to block 302 in which the computing device 102 determines whether a five-finger-press-and-hold input gesture was detected, as discussed above. If the virtual keyboard is not hidden, the method 300 advances to block 334, wherein the previously contacting finger contact point is marked as an input contact before the method 300 advances to block 336. In other words, the user 104 can now use the finger associated with that finger contact point as an input to select soft keys of the virtual keyboard. At block 336, the computing device 102 determines whether an active input finger contact touched the touch screen 122 of the display 120 at a location corresponding to a soft key of the virtual keyboard. In other words, the computing device 102 determines whether an input tap on a soft key of the virtual keyboard from a finger whose contact has been activated as an input to select soft keys of the virtual keyboard.

[0050] If the computing device 102 determines a soft key of the virtual keyboard was not touched by an active input finger contact, the method 300 loops back to block 330 to determine whether a finger contact was released. If the computing device 102 determines a soft key of the virtual keyboard was touched by an active input finger contact, the method 300 advances to block 338. At block 338, the computing device 102 processes the touch of the active input finger contact based on the corresponding soft key of the virtual keyboard touched

by the active input finger contact before looping back to block 330 to determine whether a finger contact was released.

[0051] Referring now to FIG. 5, in use, the computing device 102 may execute a method 500 for determining a hand rotation angle during a multi-finger touch interaction with the touch screen 122 of the display 120. In some embodiments, the multi-finger touch interaction may comprise the five-finger touch interaction; however, it should be appreciated that any multi-finger touch interaction from which a hand rotation angle can be determined may be used by the computing device 102. The illustrative method 500 begins at block 502, in which the computing device 102 determines a centroid position based on touch points for each of the five finger contacts. For example, as shown in FIG. 8, a hand 802 includes five fingers may have five touch points (i.e., a touch point for each finger) including a pinky finger touch point 808, a ring finger touch point 810, a middle finger touch point 812, an index finger touch point 814, and a thumb touch point 816. Accordingly, the computing device 102 can determine a centroid 828 based on the five touch points. Further, an x-axis 804 and a y-axis 806 (i.e., a centroidal axes) may be established through the centroid 828 which is oriented based on the five touch points.

[0052] Referring again to FIG. 5, at block 504, the computing device 102 calculates a touch point vector for each of the five finger contacts. As shown in FIG. 8, a pinky finger touch point vector 818, a ring finger touch point vector 820, a middle finger touch point vector 822, an index finger touch point vector 824, and a thumb touch point vector 826 are each established from the centroid 828 to each of the respective finger touch points 808, 810, 812, 814, 816. Referring again to FIG. 5, in some embodiments, at block 506, the computing device 102 determines which finger contact is a thumb based on which touch point vector of the touch point vectors 818, 820, 822, 824, 826 is the furthest distance from the centroid. As shown in FIG. 8, the thumb vector 826 is the furthest from the centroid 828 (i.e., the longest), indicating that the thumb touch point 816 corresponds to the thumb.

[0053] Referring again to FIG. 5, at block 508, the computing device 102 calculates a rotation vector based on the touch point vectors 818, 820, 822, 824, 826. To do so, in block 510, the computing device 102 calculates a sum of the unit touch point vectors (i.e., normalized touch point vectors) using the equation:

$$\sum_{i=0..4} \frac{V_i}{\|V_i\|}$$

Equation 1

wherein V corresponds to a touch point vector and i corresponds to a touch point vector instance.

[0054] In some embodiments, in block 512, the computing device 102 may additionally apply a weight to the rotation vector. To do so, the computing device 102 may calculate a sum of the unit touch point vectors and multiply the thumb vector (e.g., the thumb vector 826 of FIG. 8) by a weight, which results in the following equation:

$$\sum_{i=0..4} \frac{V_i}{\|V_i\|} + \frac{V_j}{\|V_j\|} * w$$

Equation 2

wherein V_j corresponds to the thumb vector 826, and w corresponds to a weight.

[0055] For example, in some embodiments, the weight may be a value between zero and one. In such an embodiment, a weight of zero may push the rotation vector away from the thumb vector, while a weight of one may pull the rotation vector toward the thumb vector. As shown in FIG. 8, a rotation vector 830 has been determined based on the touch point vectors 818, 820, 822, 824, 826 and a weight of 0.3 applied to the thumb vector 826.

[0056] At block 514, the computing device 102 determines a hand rotation angle. In some embodiments, the computing device 102 determines the hand rotation angle based on the rotation vector calculated at block 508 relative to the positive x-axis (e.g., x-axis 804 of FIG. 8) established by the touch point vectors (e.g., the touch point vectors 818, 820, 822, 824, 826 of FIG. 8) and intersecting at the centroid (e.g., the centroid 828 of FIG. 8).

[0057] Referring now to FIG. 6, in use, the computing device 102 may execute a method 600 for validating a hand rotation angle during a multi-finger touch interaction, such as the five-finger-press-and-hold input gesture. The illustrative method 600 begins at block 602, in which the computing device 102 determines the three shortest touch point vectors from the centroid. As shown in FIG. 8, the ring finger touch point 810, the middle finger touch point 812, and the index finger touch point 814 are the three shortest touch point vectors from the centroid 828. Referring again to FIG. 6, at block 604, the computing device 102 determines an angle between the rotation vector and each of the three shortest touch point vectors identified at block 602. At block 606, the computing device 102 determines whether each of the angles between the rotation vector and the three shortest touch point vectors are less than or equal to a maximum angle. For example, in embodiments wherein the display 120 is distinguished by four equally sized quadrants, the maximum angle may be 90°. If the computing device determines that each of the angles between the rotation vector and the three shortest touch point vectors are less than or equal to a maximum angle, the method 600 advances to block 608,

wherein the computing device 102 provides an indication of a valid hand rotation angle. If not, the method 600 advances to block 610, wherein the computing device 102 provides an indication of an invalid hand rotation angle.

[0058] Referring now to FIG. 7, in use, the computing device 102 may execute a method 700 for determining a proposed orientation of the GUI on a display (e.g., the display 120) of the computing device 102 based on the hand rotation angle. The illustrative method 700 begins at block 702, in which the computing device 102 determines whether the hand rotation angle is greater than or equal to 45° and less than 135° . If so, the method 700 advances to block 704, wherein the computing device 102 sets the proposed GUI orientation to a normal orientation (i.e., a portrait orientation, as shown in FIG. 9A). If the hand rotation angle is not greater than or equal to 45° and not less than 135° , the method 700 advances to block 706, wherein the computing device 102 determines whether the hand rotation angle is greater than or equal to 135° and less than 225° . If so, the method 700 advances to block 708, wherein the computing device 102 sets the proposed GUI orientation to rotate 90° counterclockwise from normal orientation (i.e., a landscape orientation, as shown in FIG. 9C).

[0059] If the hand rotation angle is not greater than or equal to 135° and not less than 225° , the method 700 advances to block 710, wherein the computing device 102 determines whether the hand rotation angle is greater than or equal to 225° and less than 315° . If so, the method 700 advances to block 712, wherein the computing device 102 sets the proposed GUI orientation to rotate 180° counterclockwise from normal orientation (i.e., a flipped portrait orientation). If the hand rotation angle is not greater than or equal to 225° and not less than 315° , the method 700 advances to block 714, the computing device 102 sets the proposed GUI orientation to rotate 270° counterclockwise from normal orientation (i.e., a flipped landscape orientation).

[0060] Referring now to FIGS. 9A-9C, a sequence for changing the GUI orientation on the display 120 of the computing device 102 using the five-finger-press-and-hold input gesture is shown. As shown in FIG. 9A, the computing device 102 includes a GUI display 902 at a normal portrait orientation. In other words, the GUI display 902 is displayed in a vertical orientation (i.e., taller along a y-axis 906 than wide along an x-axis 904) and is normal relative to the default viewing orientation of the display 120, typically as compared to physical components or statically positioned soft keys of the computing device 102. It should be appreciated that in some embodiments, the default viewing orientation (i.e., the normal orientation) may be a portrait orientation or a landscape orientation; however, for clarity of the description, the normal orientation is described herein as a portrait orientation. As shown in

FIG. 9B, a hand 802 is positioned at a generally 90° angle relative to the touch screen 122 of the display 120. Additionally, each of the five fingers has respective touch points 808, 810, 812, 814, 816 in contact with the touch screen 122. As shown in FIG. 9C, the GUI display 902 is rotated 90° counterclockwise along path 908 based on the position of the hand 802 relative to the touch screen 122. In other words, the GUI display 902 is rotated to a viewing angle relative to the touch points 808, 810, 812, 814, 816 of the hand 802, such that the user 104 does not need to perform any other action (i.e., open display settings, modify orientation settings, etc.) other than the five-finger-press-and-hold input gesture to initiate the rotation of the GUI display 902.

[0061] Referring now to FIGS. 10A-10C, a sequence for displaying a virtual keyboard with application context awareness on the display 120 of the computing device 102 using the five-finger-press-and-hold input gesture is shown. As shown in FIG. 10A, an application 1002 running on the computing device 102 includes a text entry GUI. As shown in FIG. 10B, a hand 802 is positioned at a generally 45° angle relative to the touch screen 122 of the display 120. Additionally, each of the five fingers has respective touch points 808, 810, 812, 814, 816 in contact with the touch screen 122. As shown in FIG. 10C, a context-specific virtual keyboard 1004 is displayed with a number of text entry soft keys 1006. As described previously, in use, the computing device 102 detects the five-finger-press-and-hold input gesture, determines a corresponding application (i.e., the application 1002), and determines that the application 1002 requires text entry. Accordingly, the computing device may display the context-specific virtual keyboard 1004 with a number of text entry soft keys 1006 at a size, shape, and location relative to the hand rotation angle determined from the touch points 808, 810, 812, 814, 816 of the hand 802.

[0062] While the virtual keyboard 1004 is shown underlying each of the touch points 808, 810, 812, 814, 816 of the hand 802, it should be appreciated that, in other embodiments, the virtual keyboard may have a different size and/or shape relative to the touch points 808, 810, 812, 814, 816. For example, in some embodiments, the virtual keyboard may be displayed directly below and/or along an arc of the four non-thumb touch points 808, 810, 812, 814 (i.e., no portion of the virtual keyboard underlying the thumb touch point 816). Accordingly, in such embodiments, the user 104 may maintain contact between the touch screen 122 and the thumb touch point 816, while the four non-thumb touch points 808, 810, 812, 814 may be used as active inputs (i.e., can be released from the touch screen 122 and used as inputs). As such, the user 104 can maintain at least one touch point with the touch screen 122 during input in such embodiments that require at least one touch point be maintained with the touch screen to keep

the virtual keyboard displayed. As shown in FIG. 11, the context-specific virtual keyboard 1004 is displayed relative to (e.g., below) the touch points 808, 810, 812, 814, 816 and each of the four non-thumb touch points 808, 810, 812, 814 are active finger contacts. In other words, each of the four non-thumb touch points 808, 810, 812, 814 have been released since the five-finger-press-and-hold input gesture was detected and are not active inputs for the context-specific virtual keyboard 1004.

[0063] As shown in FIG. 12, an illustrative alternative embodiment of a context-specific virtual keyboard 1202 is also displayed relative to (e.g., below) the touch points 808, 810, 812, 814, 816 and only the index finger touch point 814 is presently an active finger contact. In other words, only the index finger touch point 814 has been released since the five-finger-press-and-hold input gesture was detected and are not active inputs for the context-specific virtual keyboard 1202. Unlike the context-specific virtual keyboard 1004 of FIGS. 10C and 11, which provided a text entry layout of soft keys 1006, an alternative embodiment of the context-specific virtual keyboard 1202 may include a number of predefined spaces allocated for soft keys 1204 that may be assigned to various functions upon selection by the user 104 interfacing with the context-specific virtual keyboard 1202. In some embodiments, the various functions mapped to the default keys may include default, user 104 defined, and/or application defined functions, which may be based on the type of application or specific interfacing with the context-specific virtual keyboard 1202. For example, in some embodiments, one or more of the functions assigned, or mapped, to the soft keys of the context-specific virtual keyboard 1202 may be user-specified via settings of the context-specific virtual keyboard 1202.

[0064] An illustrative a mapping table 1300 is shown in FIG. 13 that includes various applications 1302 that may be executed on the computing device 102 and a number of default soft keys 1304 that may be associated with functions (i.e., mapped to the default soft keys 1304) of the various applications 1302. In the illustrative mapping table 1300, a video chat application may be mapped to default soft keys 1304 to be displayed by default on the context-specific virtual keyboard 1202. The default soft keys 1304 for the video chat application may include soft keys mapped to functions supported by the video chat application, such as a full screen toggle soft key, a volume control soft key, a background segmentation soft key, a drop call soft key, an initiate call soft key, and/or an add friends soft key, for example. Of course, it should be appreciated that additional and/or alternative default soft keys 1304 may be mapped to additional and/or alternative applications 1302.

EXAMPLES

[0065] Illustrative examples of the technologies disclosed herein are provided below. An embodiment of the technologies may include any one or more, and any combination of, the examples described below.

[0066] Example 1 includes a computing device to manipulate a graphical user interface (GUI) of a display of the computing device, the computing device comprising a display that includes a touch screen to receive input from a user based on physical contact of one or more fingers of the user with the touch screen, wherein each physical contact between a finger of the user and the touch screen establishes a corresponding touch point; an input gesture determination module to detect a presence of five touch points on the touch screen and determine a five-finger-press-and-hold input gesture received from the user in response to a determination that each of the five touch points is established for a duration that exceeds a predetermined duration; a hand orientation determination module to determine a hand rotation angle based on the detected five touch points; and a GUI orientation management module to (i) determine a present orientation of the GUI, (ii) determine a proposed orientation of the GUI based on the hand rotation angle, and (iii) cause a rotation of an orientation of the GUI on the display in response to a determination that that the proposed orientation and the present orientation of the GUI do not match.

[0067] Example 2 includes the subject matter of Example 1, and further including a virtual keyboard display management module to (i) determine, from a number of software applications presently running on the computing device, a software application that corresponds to the five-finger-press-and-hold input gesture and (ii) display a context-specific virtual keyboard at a location based on the five touch points, wherein the context-specific virtual keyboard includes a plurality of soft keys mapped to functions related to the determined software application.

[0068] Example 3 includes the subject matter of any of Examples 1 and 2, and wherein to determine the software application that corresponds to the five-finger-press-and-hold input gesture comprises to determine the software application based on a z-order of a GUI of each of the presently running software applications.

[0069] Example 4 includes the subject matter of any of Examples 1-3, and wherein to determine the application that corresponds to the five-finger-press-and-hold input gesture comprises to determine the application based on a proximate relationship of a GUI of each of the presently running software applications to the five touch points.

[0070] Example 5 includes the subject matter of any of Examples 1-4, and wherein the GUI orientation management module is further to (i) determine at least one of a size, a shape, and a location of the context-specific virtual keyboard based on at least a portion of the five touch points and (ii) determine one or more soft keys of the context-specific virtual keyboard mapped to one or more functions of the determined application.

[0071] Example 6 includes the subject matter of any of Examples 1-5, and wherein to determine the hand rotation angle comprises to (i) determine a centroid based on the five touch points, (ii) determine a touch point vector for each of the five touch points based on the determined centroid and each corresponding touch point, (iii) calculate a rotation vector based on the five touch point vectors, and (iv) determine the hand rotation angle based on the rotation vector.

[0072] Example 7 includes the subject matter of any of Examples 1-6, and wherein to determine the touch point vector for each of the five touch points comprises to determine a vector from the centroid to a touch point for each of the five touch points.

[0073] Example 8 includes the subject matter of any of Examples 1-7, and wherein to calculate the rotation vector comprises to (i) normalize each of the five touch point vectors and (ii) calculate a sum of the five normalized touch point vectors.

[0074] Example 9 includes the subject matter of any of Examples 1-8, and wherein to calculate the rotation vector further comprises to (i) determine a furthest touch point based on which touch point of the five touch point vectors is furthest from the centroid and (ii) determine the rotation vector further based on a weight applied to furthest touch point.

[0075] Example 10 includes the subject matter of any of Examples 1-9, and wherein to determine the hand rotation angle based on the rotation vector comprises to determine the hand rotation angle based on the rotation vector relative to a positive x-axis through the centroid.

[0076] Example 11 includes the subject matter of any of Examples 1-10, and wherein the hand orientation determination module is further to validate the hand rotation angle.

[0077] Example 12 includes the subject matter of any of Examples 1-11, and wherein to validate the hand rotation angle comprises to (i) determine a centroid based on the five touch points, (ii) determine a touch point vector for each of the five touch points based on the determined centroid and each corresponding touch point, (iii) determine three touch point vectors of the five touch point vectors that are closest to the centroid, (iv) determine an angle between each of the three touch point vectors and an axis of a centroidal axes through the centroid, and (iv) determine a confidence level of the hand rotation angle based on each angle being less than a maximum angle.

[0078] Example 13 includes the subject matter of any of Examples 1-12, and wherein the maximum angle is equal to 90°.

[0079] Example 14 includes the subject matter of any of Examples 1-13, and wherein to determine the proposed orientation of the GUI based on the hand rotation angle comprises to determine a proposed GUI rotation angle relative to a normal orientation of the display.

[0080] Example 15 includes the subject matter of any of Examples 1-14, and wherein to cause the rotation of the orientation of the GUI on the display comprises to cause the rotation of the orientation of the GUI to rotate the GUI to the normal orientation of the display in response to a determination that the hand rotation angle is greater than or equal to 45° and less than 135°.

[0081] Example 16 includes the subject matter of any of Examples 1-15, and wherein to cause the rotation of the orientation of the GUI on the display comprises to cause the rotation of the orientation of the GUI to rotate the GUI to the normal orientation of the display in response to a determination that the hand rotation angle is greater than or equal to 135° and less than 225°.

[0082] Example 17 includes the subject matter of any of Examples 1-16, and wherein to cause the rotation of the orientation of the GUI on the display comprises to cause the rotation of the orientation of the GUI to rotate the GUI to the normal orientation of the display in response to a determination that the hand rotation angle is greater than or equal to 225° and less than 315°.

[0083] Example 18 includes the subject matter of any of Examples 1-17, and wherein to cause the rotation of the orientation of the GUI on the display comprises to cause the rotation of the orientation of the GUI to rotate the GUI to the normal orientation of the display in response to a determination that the hand rotation angle is greater than or equal to 315°.

[0084] Example 19 includes a method for manipulating a graphical user interface (GUI) of a display of the computing device, the method comprising detecting a presence of five touch points on a touch screen of a display of the computing device, wherein each of the five touch points is indicative of a physical contact of a finger of a user of the computing with the touch screen; determining a five-finger-press-and-hold input gesture received from the user in response to a determination that the presence of each of the five touch points is established for a duration that exceeds a predetermined duration; determining a hand rotation angle based on the detected five touch points; determining a present orientation of the GUI; determining a proposed orientation of the GUI based on the hand rotation angle; and causing a rotation of an orientation of the GUI on the display in response to a determination that that the proposed orientation and the present orientation of the GUI do not match.

[0085] Example 20 includes the subject matter of Example 19, and further including determining, from a number of software applications presently running on the computing device, a software application that corresponds to the five-finger-press-and-hold input gesture; and displaying a context-specific virtual keyboard at a location based on the five touch points, wherein the context-specific virtual keyboard includes a plurality of soft keys mapped to functions related to the determined software application.

[0086] Example 21 includes the subject matter of any of Examples 19 and 20, and wherein determining the software application that corresponds to the five-finger-press-and-hold input gesture comprises determining the software application based on a z-order of a GUI of each of the presently running software applications.

[0087] Example 22 includes the subject matter of any of Examples 19-21, and wherein determining the application that corresponds to the five-finger-press-and-hold input gesture comprises determining the application based on a proximate relationship of a GUI of each of the presently running software applications to the five touch points.

[0088] Example 23 includes the subject matter of any of Examples 19-22, and further including determining at least one of a size, a shape, and a location of the context-specific virtual keyboard based on at least a portion of the five touch points; and determining one or more soft keys of the context-specific virtual keyboard mapped to one or more functions of the determined application.

[0089] Example 24 includes the subject matter of any of Examples 19-23, and wherein determining the hand rotation angle comprises (i) determining a centroid based on the five touch points, (ii) determining a touch point vector for each of the five touch points based on the determined centroid and each corresponding touch point, (iii) calculating a rotation vector based on the five touch point vectors, and (iv) determining the hand rotation angle based on the rotation vector.

[0090] Example 25 includes the subject matter of any of Examples 19-24, and wherein determining the touch point vector for each of the five touch points comprises determining a vector from the centroid to a touch point for each of the five touch points.

[0091] Example 26 includes the subject matter of any of Examples 19-25, and wherein calculating the rotation vector comprises (i) normalizing each of the five touch point vectors and (ii) calculating a sum of the five normalized touch point vectors.

[0092] Example 27 includes the subject matter of any of Examples 19-26, and wherein calculating the rotation vector further comprises (i) determining a furthest touch point based on

which touch point of the five touch point vectors is furthest from the centroid and (ii) determining the rotation vector further based on a weight applied to the furthest touch point.

[0093] Example 28 includes the subject matter of any of Examples 19-27, and wherein determining the hand rotation angle based on the rotation vector comprises determining the hand rotation angle based on the rotation vector relative to a positive x-axis through the centroid.

[0094] Example 29 includes the subject matter of any of Examples 19-28, and further including validating the hand rotation angle.

[0095] Example 30 includes the subject matter of any of Examples 19-29, and wherein validating the hand rotation angle comprises (i) determining a centroid based on the five touch points, (ii) determining a touch point vector for each of the five touch points based on the determined centroid and each corresponding touch point, (iii) determining three touch point vectors of the five touch point vectors that are closest to the centroid, (iv) determining an angle between each of the three touch point vectors and an axis of a centroidal axes through the centroid, and (v) determining a confidence level of the hand rotation angle based on each angle being less than a maximum angle.

[0096] Example 31 includes the subject matter of any of Examples 19-30, and wherein determining the confidence level of the hand rotation angle based on each angle being less than the maximum angle comprises determining the confidence level of the hand rotation angle based on each angle being less than 90°.

[0097] Example 32 includes the subject matter of any of Examples 19-31, and, wherein determining the proposed orientation of the GUI based on the hand rotation angle comprises determining a proposed GUI rotation angle relative to a normal orientation of the display.

[0098] Example 33 includes the subject matter of any of Examples 19-32, and wherein causing the rotation of the orientation of the GUI on the display comprises causing the rotation of the orientation of the GUI to rotate the GUI to the normal orientation of the display in response to a determination that the hand rotation angle is greater than or equal to 45° and less than 135°.

[0099] Example 34 includes the subject matter of any of Examples 19-33, and wherein causing the rotation of the orientation of the GUI on the display comprises causing the rotation of the orientation of the GUI to rotate the GUI to the normal orientation of the display in response to a determination that the hand rotation angle is greater than or equal to 135° and less than 225°.

[00100] Example 35 includes the subject matter of any of Examples 19-34, and wherein causing the rotation of the orientation of the GUI on the display comprises causing the rotation of the orientation of the GUI to rotate the GUI to the normal orientation of the display in response to a determination that the hand rotation angle is greater than or equal to 225° and less than 315°.

[00101] Example 36 includes the subject matter of any of Examples 19-35, and wherein causing the rotation of the orientation of the GUI on the display comprises causing the rotation of the orientation of the GUI to rotate the GUI to the normal orientation of the display in response to a determination that the hand rotation angle is greater than or equal to 315°.

[00102] Example 37 includes a computing device comprising a processor; and a memory having stored therein a plurality of instructions that when executed by the processor cause the computing device to perform the method of any of Examples 19-36.

[00103] Example 38 includes one or more machine readable storage media comprising a plurality of instructions stored thereon that in response to being executed result in a computing device performing the method of any of Examples 19-36.

[00104] Example 39 includes a computing device for manipulating a graphical user interface (GUI) of a display of the computing device, the computing device comprising means for detecting a presence of five touch points on a touch screen of a display of the computing device, wherein each of the five touch points is indicative of a physical contact of a finger of a user of the computing with the touch screen; means for determining a five-finger-press-and-hold input gesture received from the user in response to a determination that the presence of each of the five touch points is established for a duration that exceeds a predetermined duration; means for determining a hand rotation angle based on the detected five touch points; means for determining a present orientation of the GUI; means for determining a proposed orientation of the GUI based on the hand rotation angle; and means for causing a rotation of an orientation of the GUI on the display in response to a determination that that the proposed orientation and the present orientation of the GUI do not match.

[00105] Example 40 includes the subject matter of Example 39, and further including means for determining, from a number of software applications presently running on the computing device, a software application that corresponds to the five-finger-press-and-hold input gesture; and means for displaying a context-specific virtual keyboard at a location based on the five touch points, wherein the context-specific virtual keyboard includes a plurality of soft keys mapped to functions related to the determined software application.

[00106] Example 41 includes the subject matter of any of Examples 39 and 40, and wherein the means for determining the software application that corresponds to the five-finger-press-and-hold input gesture comprises means for determining the software application based on a z-order of a GUI of each of the presently running software applications.

[00107] Example 42 includes the subject matter of any of Examples 39-41, and wherein the means for determining the application that corresponds to the five-finger-press-and-hold input gesture comprises means for determining the application based on a proximate relationship of a GUI of each of the presently running software applications to the five touch points.

[00108] Example 43 includes the subject matter of any of Examples 39-42, and further including means for determining at least one of a size, a shape, and a location of the context-specific virtual keyboard based on at least a portion of the five touch points; and means for determining one or more soft keys of the context-specific virtual keyboard mapped to one or more functions of the determined application.

[00109] Example 44 includes the subject matter of any of Examples 39-43, and wherein the means for determining the hand rotation angle comprises means for (i) determining a centroid based on the five touch points, (ii) determining a touch point vector for each of the five touch points based on the determined centroid and each corresponding touch point, (iii) calculating a rotation vector based on the five touch point vectors, and (iv) determining the hand rotation angle based on the rotation vector.

[00110] Example 45 includes the subject matter of any of Examples 39-44, and wherein the means for determining the touch point vector for each of the five touch points comprises means for determining a vector from the centroid to a touch point for each of the five touch points.

[00111] Example 46 includes the subject matter of any of Examples 39-45, and wherein the means for calculating the rotation vector comprises means for (i) normalizing each of the five touch point vectors and (ii) calculating a sum of the five normalized touch point vectors.

[00112] Example 47 includes the subject matter of any of Examples 39-46, and wherein the means for calculating the rotation vector further comprises means for (i) determining a furthest touch point based on which touch point of the five touch point vectors is furthest from the centroid and (ii) determining the rotation vector further based on a weight applied to the furthest touch point.

[00113] Example 48 includes the subject matter of any of Examples 39-47, and, wherein the means for determining the hand rotation angle based on the rotation vector comprises means

for determining the hand rotation angle based on the rotation vector relative to a positive x-axis through the centroid.

[00114] Example 49 includes the subject matter of any of Examples 39-48, and further including means for validating the hand rotation angle.

[00115] Example 50 includes the subject matter of any of Examples 39-49, and wherein the means for validating the hand rotation angle comprises means for (i) determining a centroid based on the five touch points, (ii) determining a touch point vector for each of the five touch points based on the determined centroid and each corresponding touch point, (iii) determining three touch point vectors of the five touch point vectors that are the closest to the centroid, (iv) determining an angle between each of the three touch point vectors and an axis of a centroidal axes through the centroid, and (v) determining a confidence level of the hand rotation angle based on each angle being less than a maximum angle.

[00116] Example 51 includes the subject matter of any of Examples 39-50, and wherein means for determining the confidence level of the hand rotation angle based on each angle being less than the maximum angle comprises means for determining the confidence level of the hand rotation angle based on each angle being less than 90°.

[00117] Example 52 includes the subject matter of any of Examples 39-51, and wherein the means for determining the proposed orientation of the GUI based on the hand rotation angle comprises means for determining a proposed GUI rotation angle relative to a normal orientation of the display.

[00118] Example 53 includes the subject matter of any of Examples 39-52, and wherein the means for causing the rotation of orientation of the GUI to rotate the GUI to the normal orientation of the display in response to a determination that the hand rotation angle is greater than or equal to 45° and less than 135°.

[00119] Example 54 includes the subject matter of any of Examples 39-53, and wherein the means for causing the rotation of the orientation of the GUI on the display comprises means for causing the rotation of the orientation of the GUI to rotate the GUI to the normal orientation of the display in response to a determination that the hand rotation angle is greater than or equal to 135° and less than 225°.

[00120] Example 55 includes the subject matter of any of Examples 39-54, and wherein the means for causing the rotation of the orientation of the GUI on the display comprises means for causing the rotation of the orientation of the GUI to rotate the GUI to the normal orientation of the display in response to a determination that the hand rotation angle is greater than or equal to 225° and less than 315°.

[00121] Example 56 includes the subject matter of any of Examples 39-55, and wherein the means for causing the rotation of the orientation of the GUI on the display comprises means for causing the rotation of the orientation of the GUI to rotate the GUI to the normal orientation of the display in response to a determination that the hand rotation angle is greater than or equal to 315°.

WHAT IS CLAIMED IS:

1. A computing device to manipulate a graphical user interface (GUI) of a display of the computing device, the computing device comprising:

a display that includes a touch screen to receive input from a user based on physical contact of one or more fingers of the user with the touch screen, wherein each physical contact between a finger of the user and the touch screen establishes a corresponding touch point;

an input gesture determination module to detect a presence of five touch points on the touch screen and determine a five-finger-press-and-hold input gesture received from the user in response to a determination that each of the five touch points is established for a duration that exceeds a predetermined duration;

a hand orientation determination module to determine a hand rotation angle based on the detected five touch points; and

a GUI orientation management module to (i) determine a present orientation of the GUI, (ii) determine a proposed orientation of the GUI based on the hand rotation angle, and (iii) cause a rotation of an orientation of the GUI on the display in response to a determination that that the proposed orientation and the present orientation of the GUI do not match.

2. The computing device of claim 1, further comprising:

a virtual keyboard display management module to (i) determine, from a number of software applications presently running on the computing device, a software application that corresponds to the five-finger-press-and-hold input gesture and (ii) display a context-specific virtual keyboard at a location based on the five touch points,

wherein the context-specific virtual keyboard includes a plurality of soft keys mapped to functions related to the determined software application.

3. The computing device of claim 2, wherein to determine the software application that corresponds to the five-finger-press-and-hold input gesture comprises to determine the software application based on at least one of a z-order of a GUI of each of the presently running software applications and on a proximate relationship of a GUI of each of the presently running software applications to the five touch points.

4. The computing device of claim 2, wherein the GUI orientation management module is further to (i) determine at least one of a size, a shape, and a location of the context-

specific virtual keyboard based on at least a portion of the five touch points and (ii) determine one or more soft keys of the context-specific virtual keyboard mapped to one or more functions of the determined application.

5. The computing device of claim 1, wherein to determine the hand rotation angle comprises to (i) determine a centroid based on the five touch points, (ii) determine a touch point vector for each of the five touch points based on the determined centroid and each corresponding touch point, (iii) calculate a rotation vector based on the five touch point vectors, and (iv) determine the hand rotation angle based on the rotation vector relative to a positive x-axis through the centroid.

6. The computing device of claim 5, wherein to determine the touch point vector for each of the five touch points comprises to determine a vector from the centroid to a touch point for each of the five touch points.

7. The computing device of claim 6, wherein to calculate the rotation vector comprises to (i) normalize each of the five touch point vectors and (ii) calculate a sum of the five normalized touch point vectors.

8. The computing device of claim 6, wherein to calculate the rotation vector further comprises to (i) determine a furthest touch point based on which touch point of the five touch point vectors is furthest from the centroid and (ii) determine the rotation vector further based on a weight applied to furthest touch point.

9. The computing device of claim 1, wherein the hand orientation determination module is further to validate the hand rotation angle, wherein to validate the hand rotation angle comprises to (i) determine a centroid based on the five touch points, (ii) determine a touch point vector for each of the five touch points based on the determined centroid and each corresponding touch point, (iii) determine three touch point vectors of the five touch point vectors that are closest to the centroid, (iv) determine an angle between each of the three touch point vectors and an axis of a centroidal axes through the centroid, and (iv) determine a confidence level of the hand rotation angle based on each angle being less than a maximum angle.

10. The computing device of claim 9, wherein the maximum angle is equal to 90°.

11. The computing device of claim 1, wherein to determine the proposed orientation of the GUI based on the hand rotation angle comprises to determine a proposed GUI rotation angle relative to a normal orientation of the display.

12. The computing device of claim 11, wherein to cause the rotation of the orientation of the GUI on the display comprises to cause the rotation of the orientation of the GUI to rotate the GUI to the normal orientation of the display in response to a determination that the hand rotation angle is greater than or equal to 45° and less than 135° ,

wherein to cause the rotation of the orientation of the GUI on the display comprises to cause the rotation of the orientation of the GUI to rotate the GUI to the normal orientation of the display in response to a determination that the hand rotation angle is greater than or equal to 135° and less than 225° ,

wherein to cause the rotation of the orientation of the GUI on the display comprises to cause the rotation of the orientation of the GUI to rotate the GUI to the normal orientation of the display in response to a determination that the hand rotation angle is greater than or equal to 225° and less than 315° , and

wherein to cause the rotation of the orientation of the GUI on the display comprises to cause the rotation of the orientation of the GUI to rotate the GUI to the normal orientation of the display in response to a determination that the hand rotation angle is greater than or equal to 315° .

13. A method for manipulating a graphical user interface (GUI) of a display of the computing device, the method comprising:

detecting a presence of five touch points on a touch screen of a display of the computing device, wherein each of the five touch points is indicative of a physical contact of a finger of a user of the computing with the touch screen;

determining a five-finger-press-and-hold input gesture received from the user in response to a determination that the presence of each of the five touch points is established for a duration that exceeds a predetermined duration;

determining a hand rotation angle based on the detected five touch points;

determining a present orientation of the GUI;

determining a proposed orientation of the GUI based on the hand rotation angle;

and

causing a rotation of an orientation of the GUI on the display in response to a determination that that the proposed orientation and the present orientation of the GUI do not match.

14. The method of claim 13, further comprising:

determining, from a number of software applications presently running on the computing device, a software application that corresponds to the five-finger-press-and-hold input gesture; and

displaying a context-specific virtual keyboard at a location based on the five touch points, wherein the context-specific virtual keyboard includes a plurality of soft keys mapped to functions related to the determined software application.

15. The method of claim 14, wherein determining the software application that corresponds to the five-finger-press-and-hold input gesture comprises determining the software application based on at least one of a z-order of a GUI of each of the presently running software applications and a proximate relationship of a GUI of each of the presently running software applications to the five touch points.

16. The method of claim 14, further comprising:

determining at least one of a size, a shape, and a location of the context-specific virtual keyboard based on at least a portion of the five touch points; and

determining one or more soft keys of the context-specific virtual keyboard mapped to one or more functions of the determined application.

17. The method of claim 13, wherein determining the hand rotation angle comprises (i) determining a centroid based on the five touch points, (ii) determining a touch point vector for each of the five touch points based on the determined centroid and each corresponding touch point, (iii) calculating a rotation vector based on the five touch point vectors, and (iv) determining the hand rotation angle based on the rotation vector relative to a positive x-axis through the centroid.

18. The method of claim 17, wherein determining the touch point vector for each of the five touch points comprises determining a vector from the centroid to a touch point for each of the five touch points.

19. The method of claim 17, wherein calculating the rotation vector comprises (i) normalizing each of the five touch point vectors and (ii) calculating a sum of the five normalized touch point vectors.

20. The method of claim 17, wherein calculating the rotation vector further comprises (i) determining a furthest touch point based on which touch point of the five touch point vectors is furthest from the centroid and (ii) determining the rotation vector further based on a weight applied to the furthest touch point.

21. The method of claim 13, further comprising validating the hand rotation angle, wherein validating the hand rotation angle comprises (i) determining a centroid based on the five touch points, (ii) determining a touch point vector for each of the five touch points based on the determined centroid and each corresponding touch point, (iii) determining three touch point vectors of the five touch point vectors that are closest to the centroid, (iv) determining an angle between each of the three touch point vectors and an axis of a centroidal axes through the centroid, and (v) determining a confidence level of the hand rotation angle based on each angle being less than a maximum angle.

22. The method of claim 21, wherein determining the confidence level of the hand rotation angle based on each angle being less than the maximum angle comprises determining the confidence level of the hand rotation angle based on each angle being less than 90°.

23. The method of claim 13, wherein causing the rotation of the orientation of the GUI on the display comprises causing the rotation of the orientation of the GUI to rotate the GUI to the normal orientation of the display in response to a determination that the hand rotation angle is greater than or equal to 45° and less than 135°,

wherein causing the rotation of the orientation of the GUI on the display comprises causing the rotation of the orientation of the GUI to rotate the GUI to the normal orientation of the display in response to a determination that the hand rotation angle is greater than or equal to 135° and less than 225°,

wherein causing the rotation of the orientation of the GUI on the display comprises causing the rotation of the orientation of the GUI to rotate the GUI to the normal orientation of the display in response to a determination that the hand rotation angle is greater than or equal to 225° and less than 315°, and

wherein causing the rotation of the orientation of the GUI on the display comprises causing the rotation of the orientation of the GUI to rotate the GUI to the normal orientation of the display in response to a determination that the hand rotation angle is greater than or equal to 315° .

24. A computing device comprising:
a processor; and
a memory having stored therein a plurality of instructions that when executed by the processor cause the computing device to perform the method of any of claims 19-23.

25. One or more machine readable storage media comprising a plurality of instructions stored thereon that in response to being executed result in a computing device performing the method of any of claims 19-23.

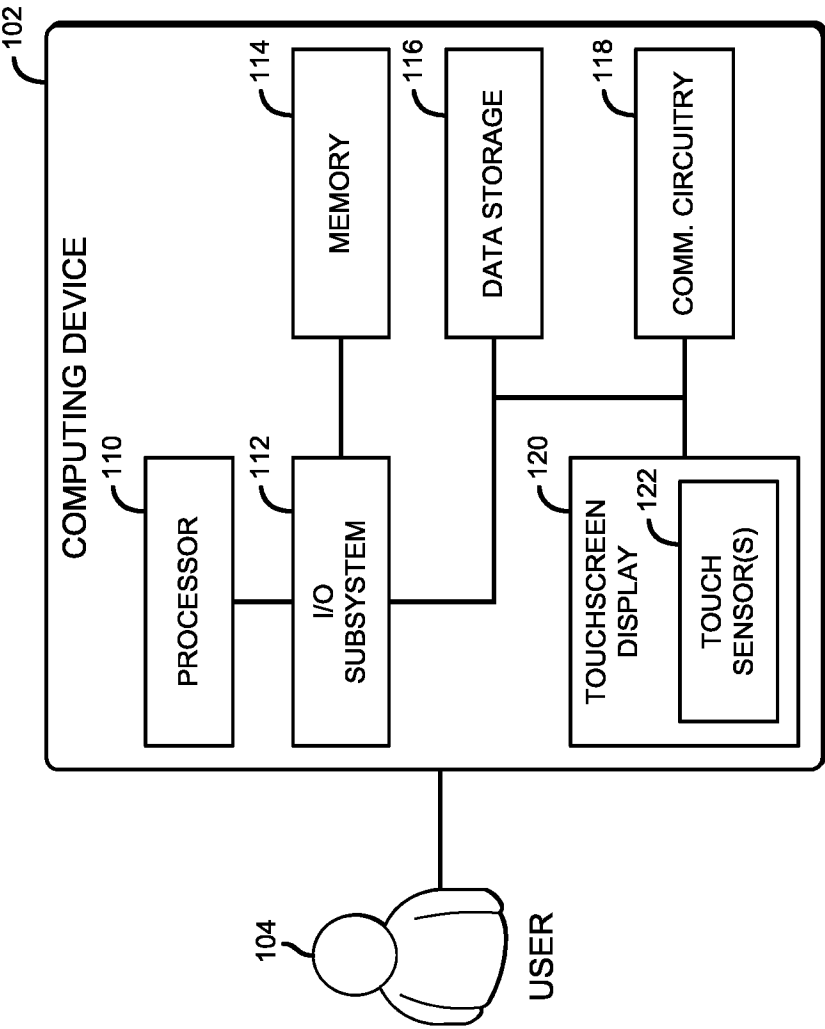


FIG. 1

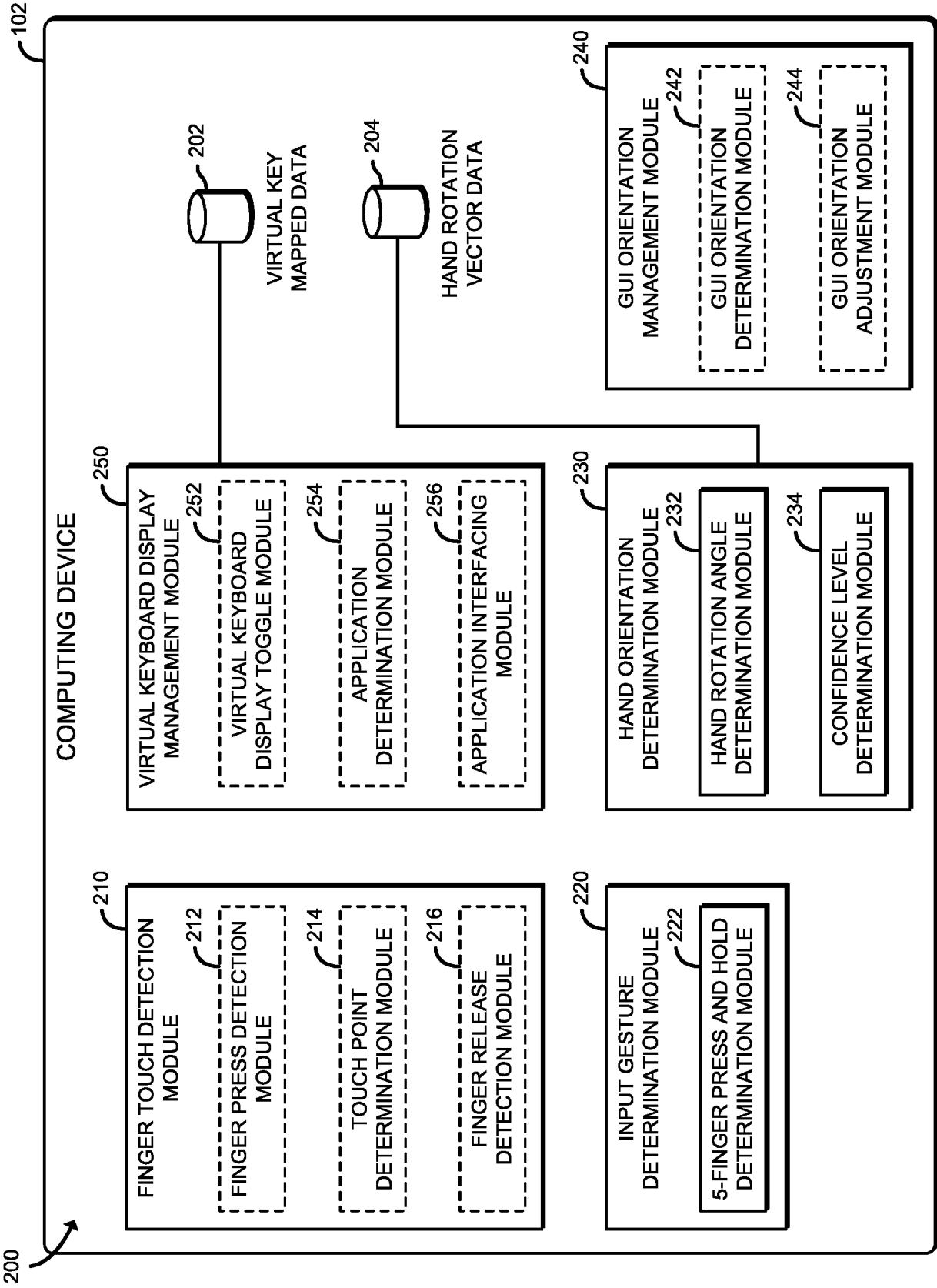


FIG. 2

3/17

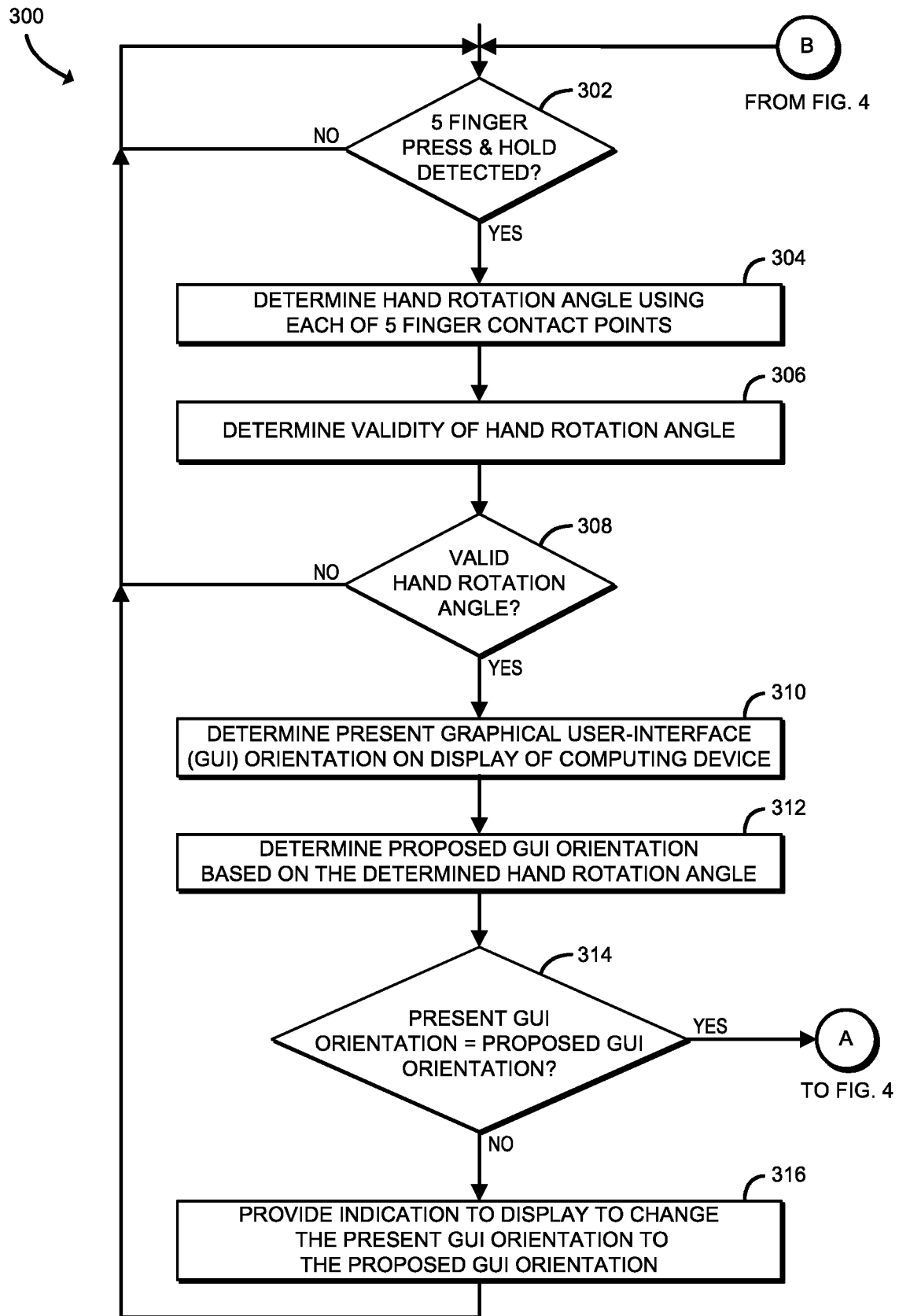


FIG. 3

4/17

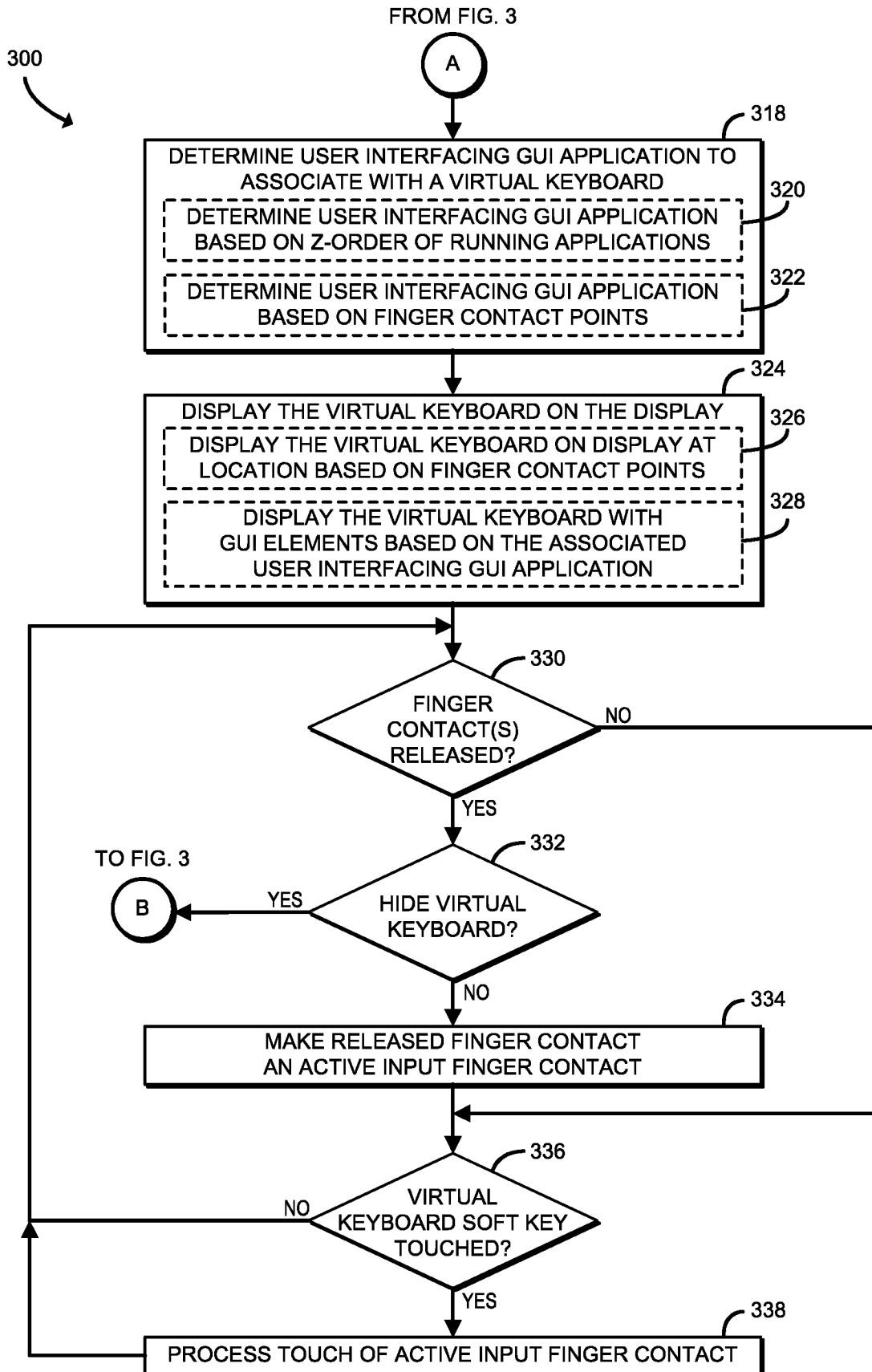


FIG. 4

5/17

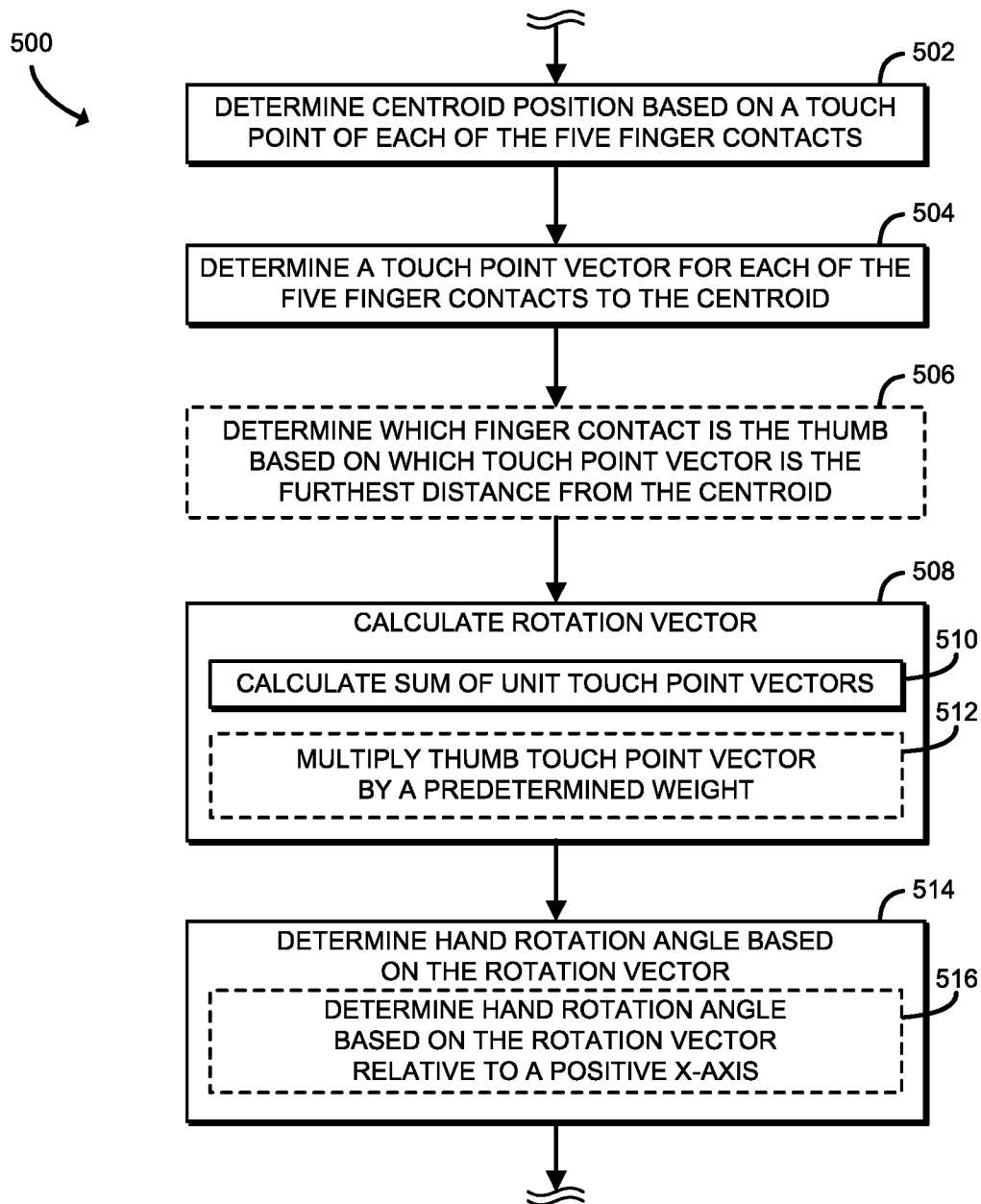


FIG. 5

6/17

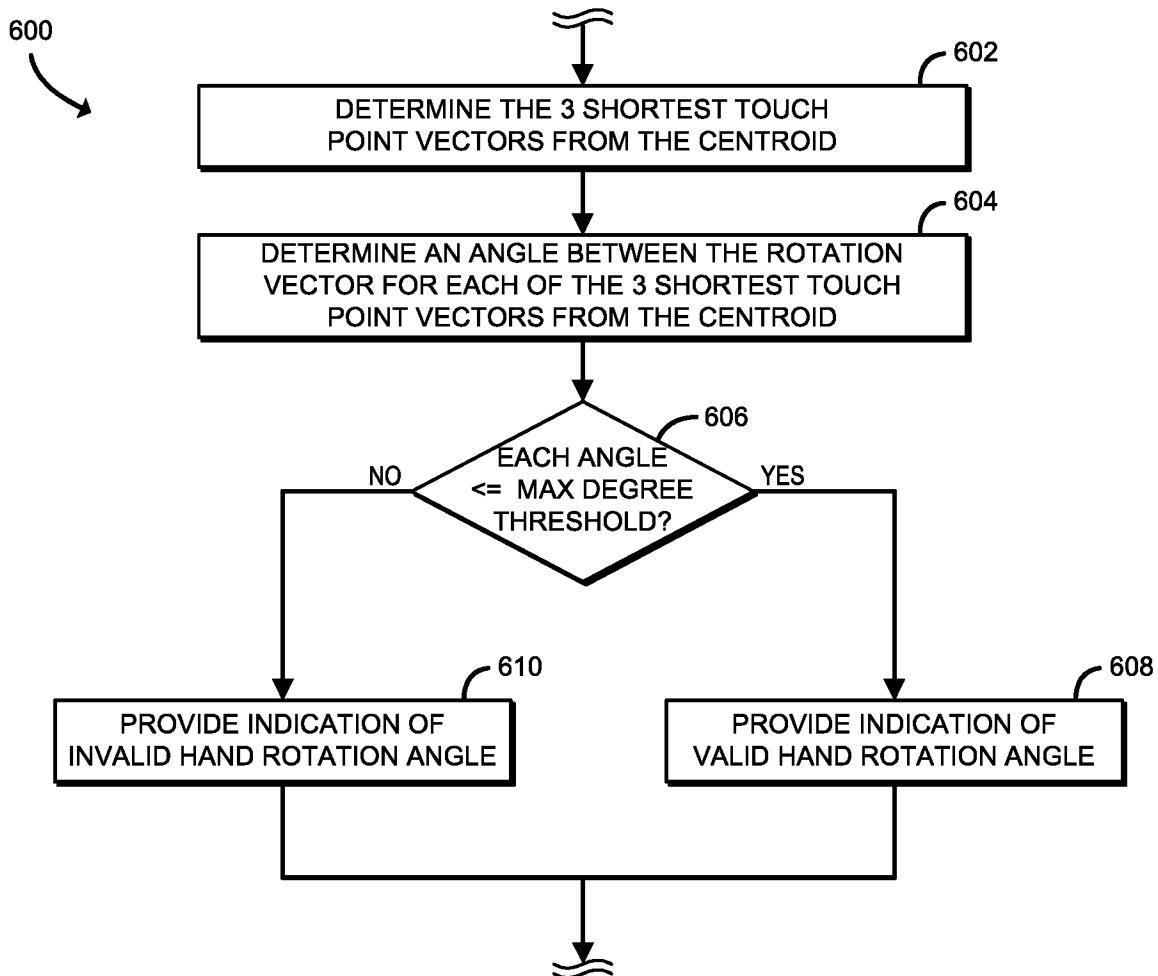


FIG. 6

7/17

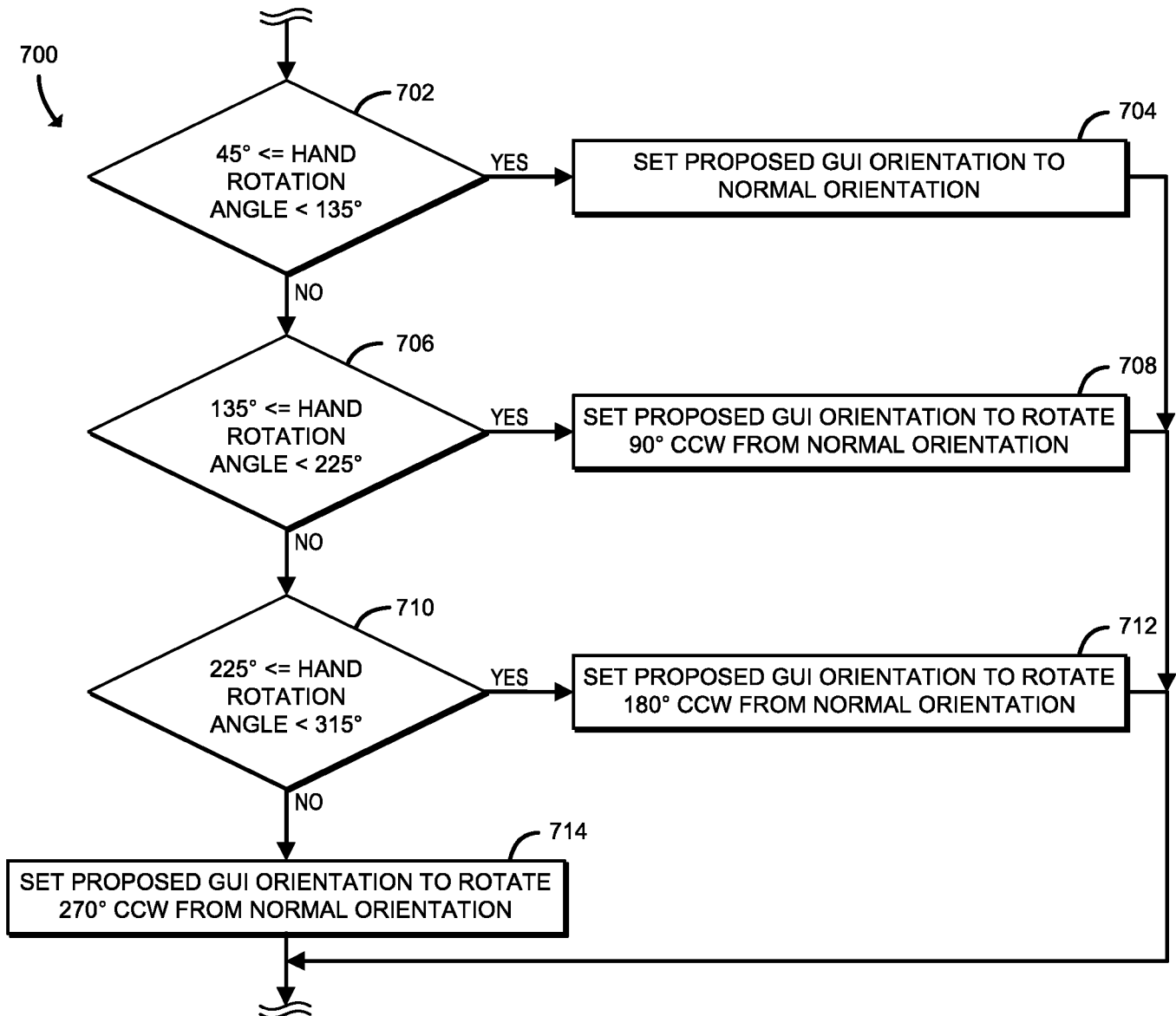


FIG. 7

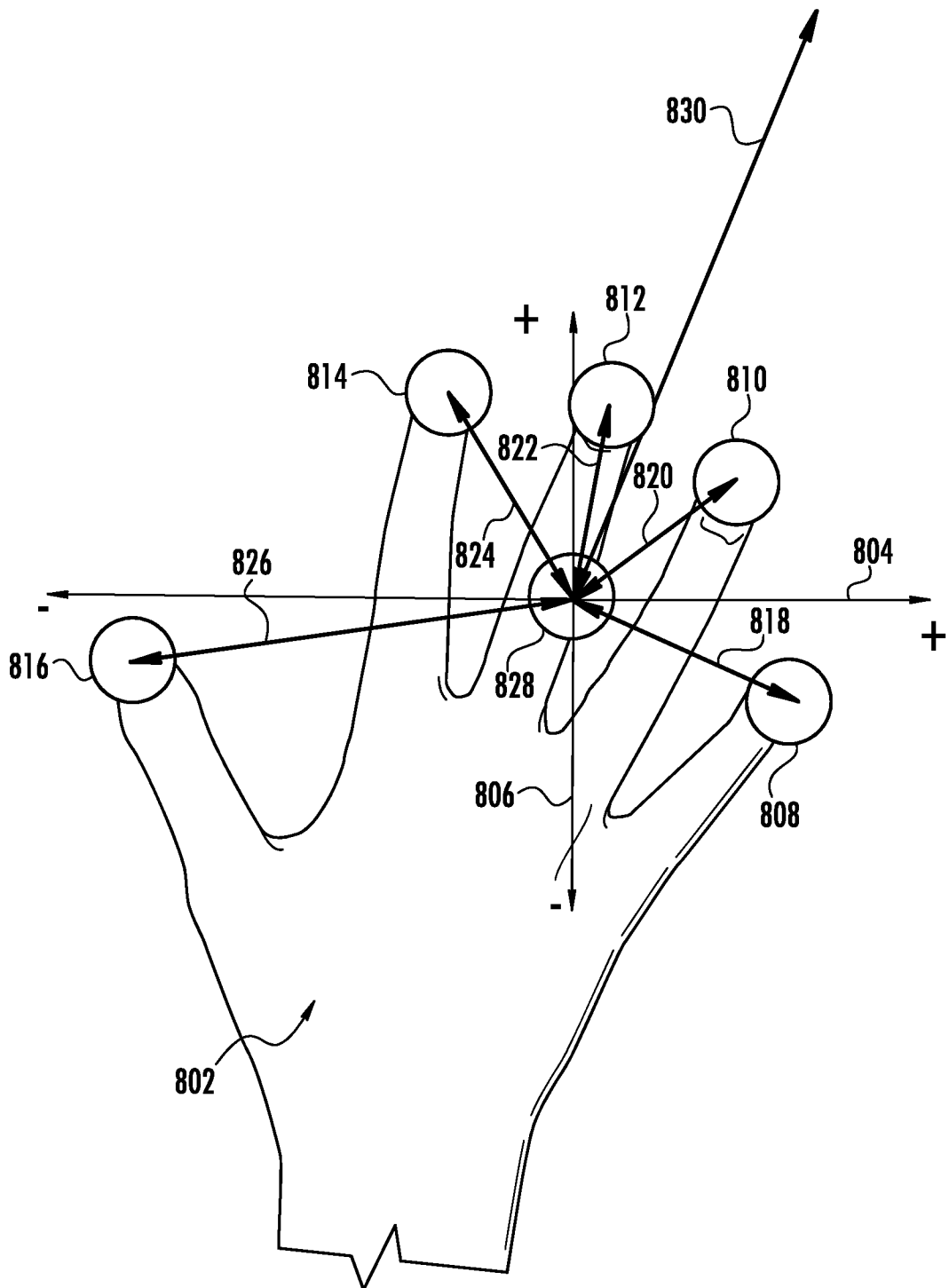


FIG. 8

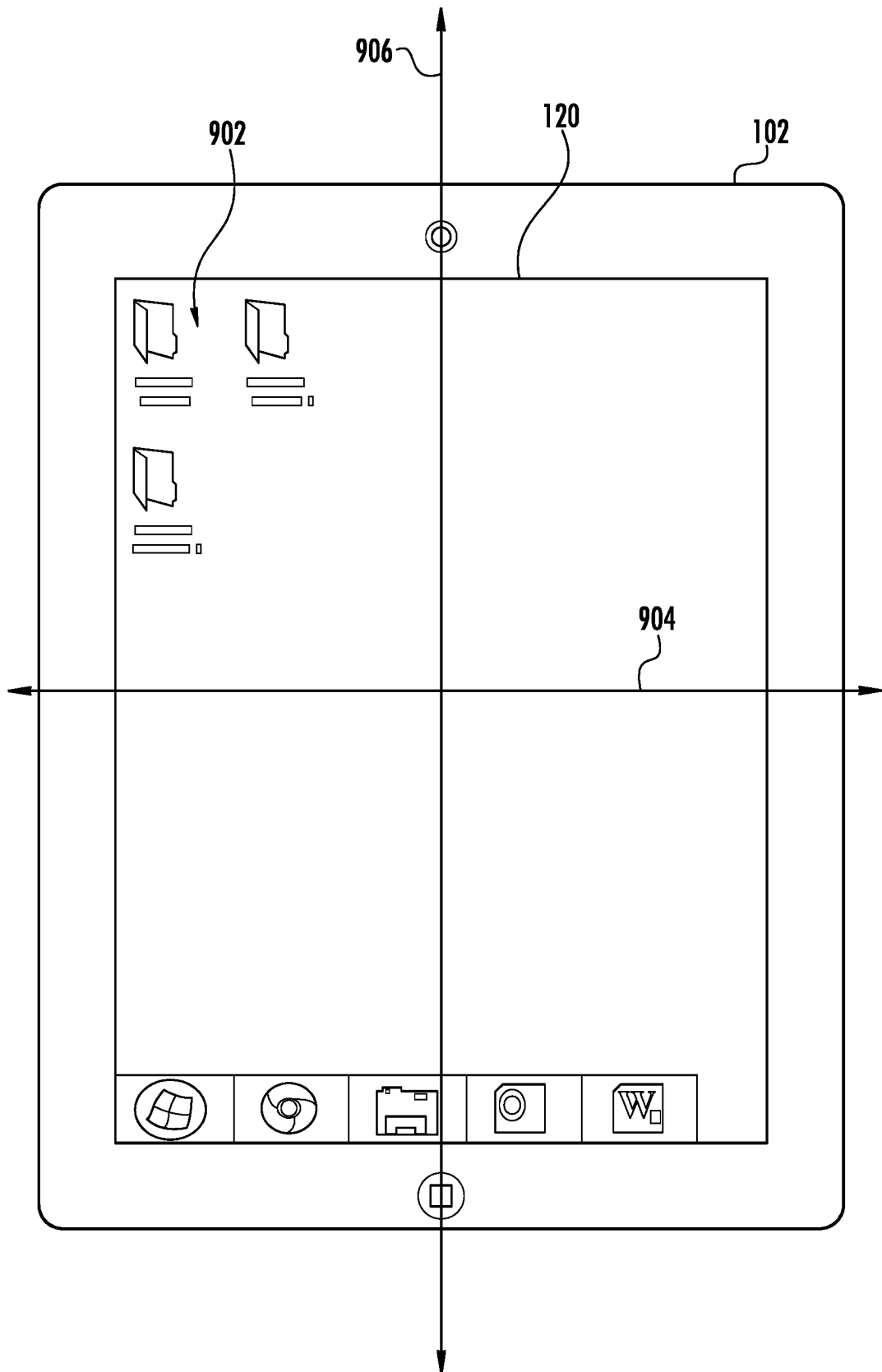


FIG. 9A

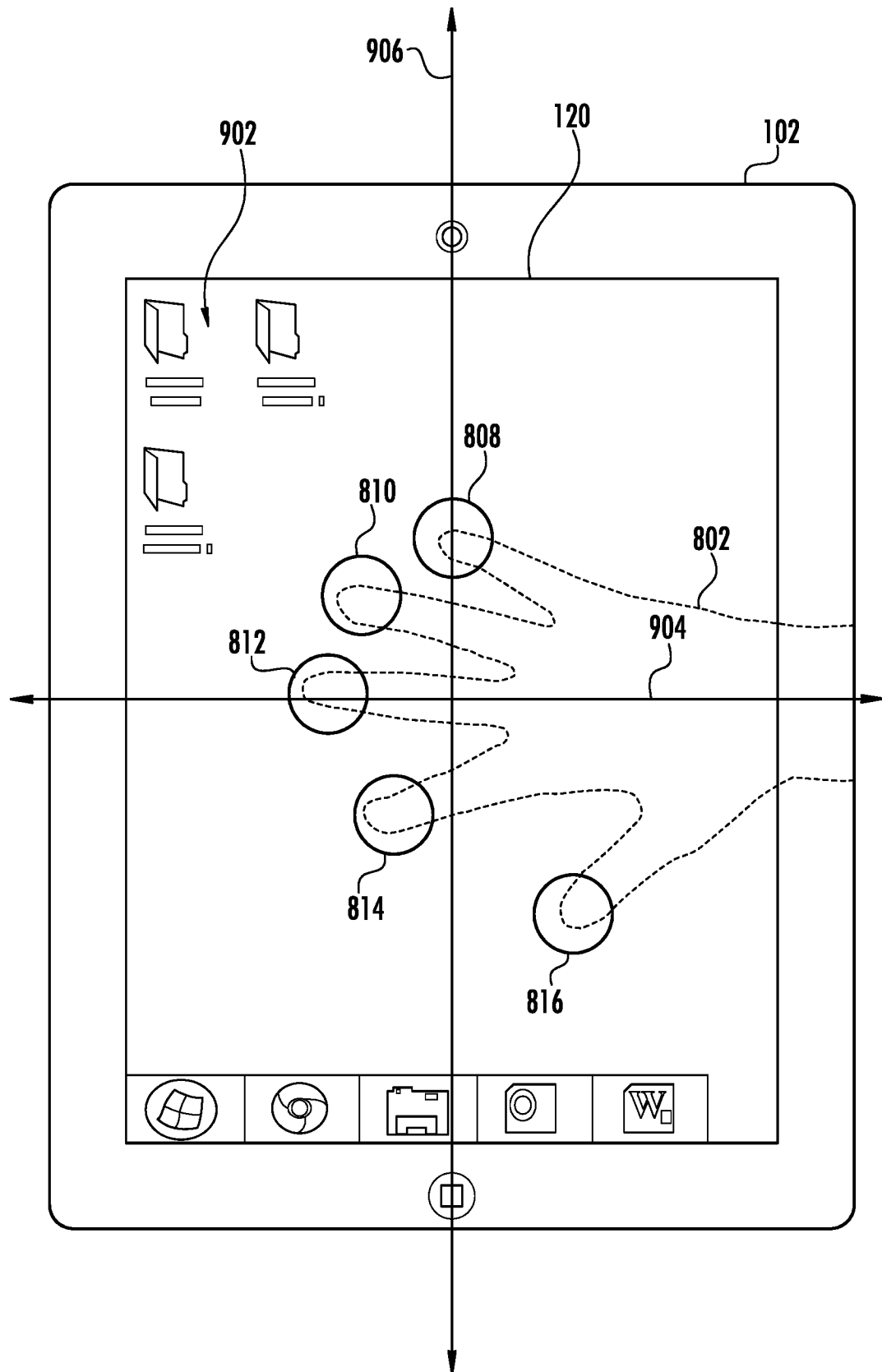


FIG. 9B

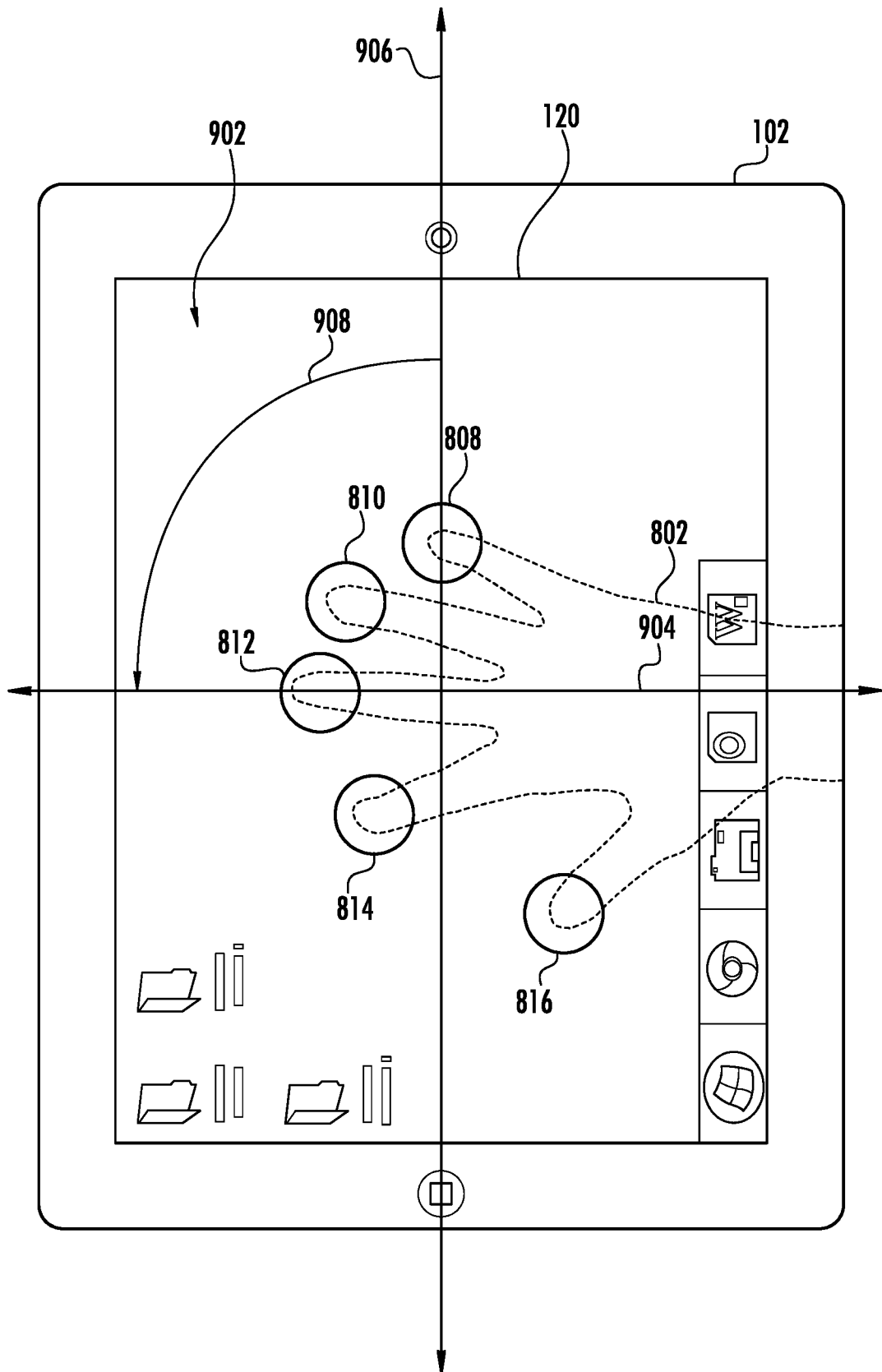


FIG. 9C

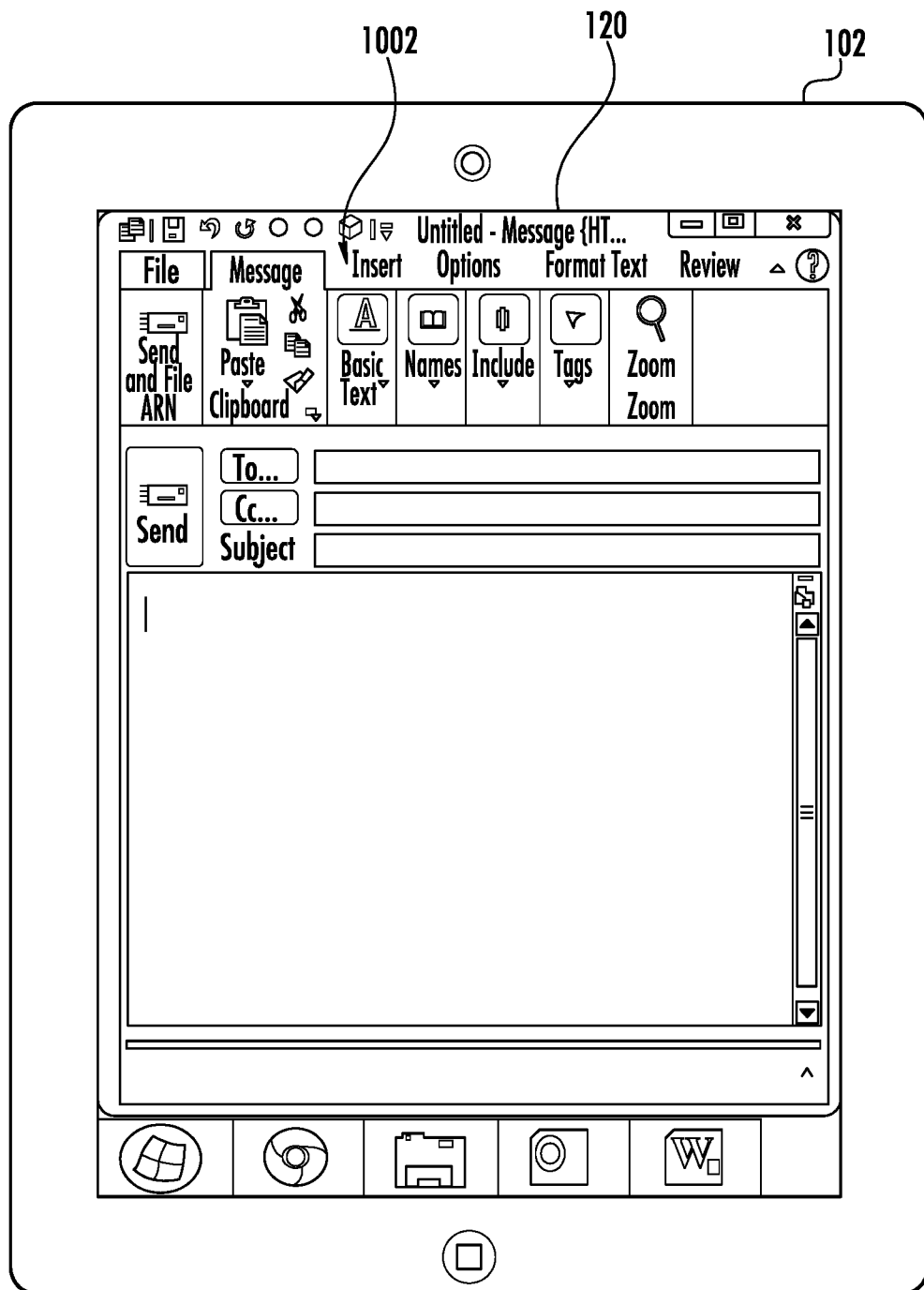


FIG. 10A

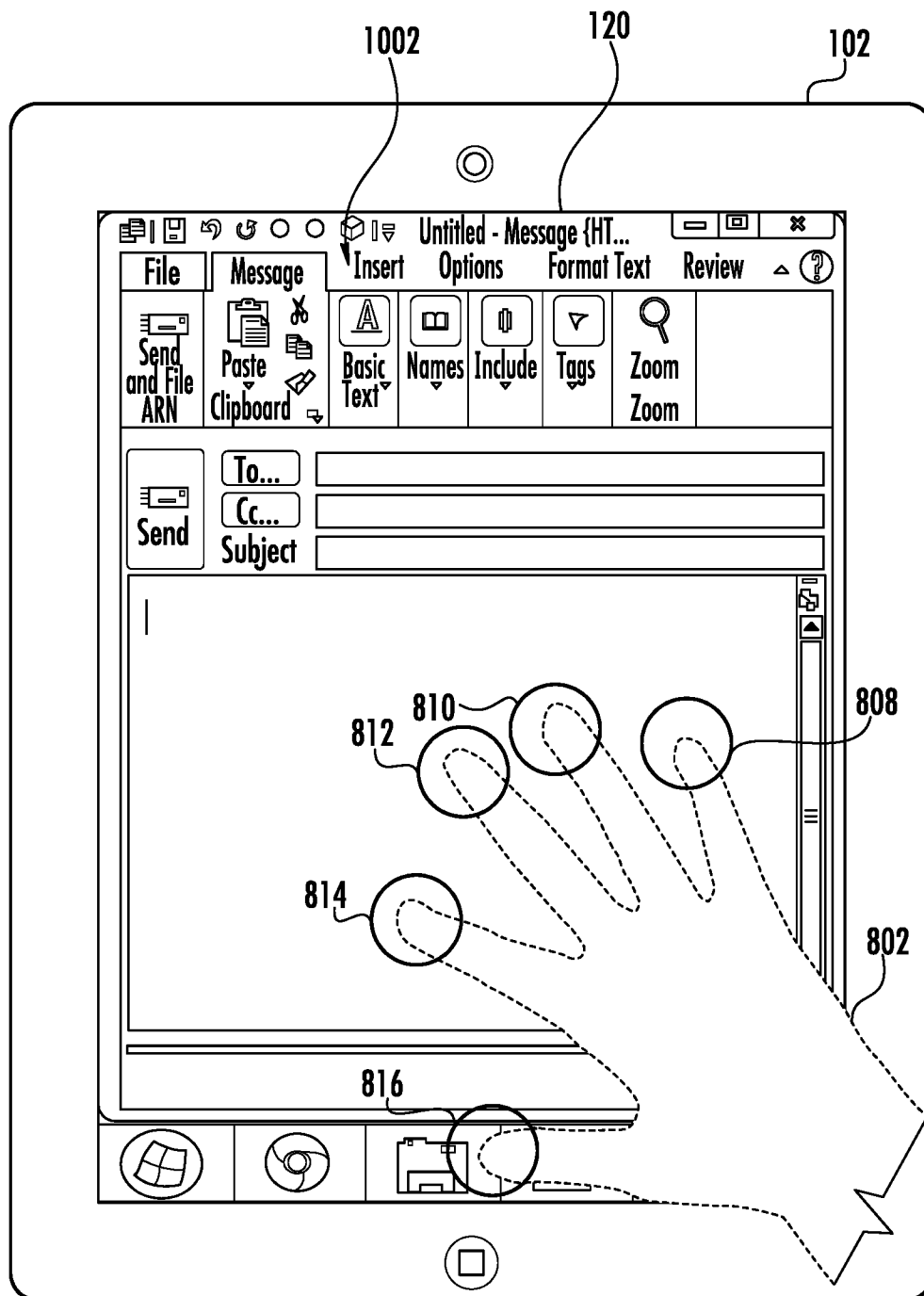


FIG. 10B

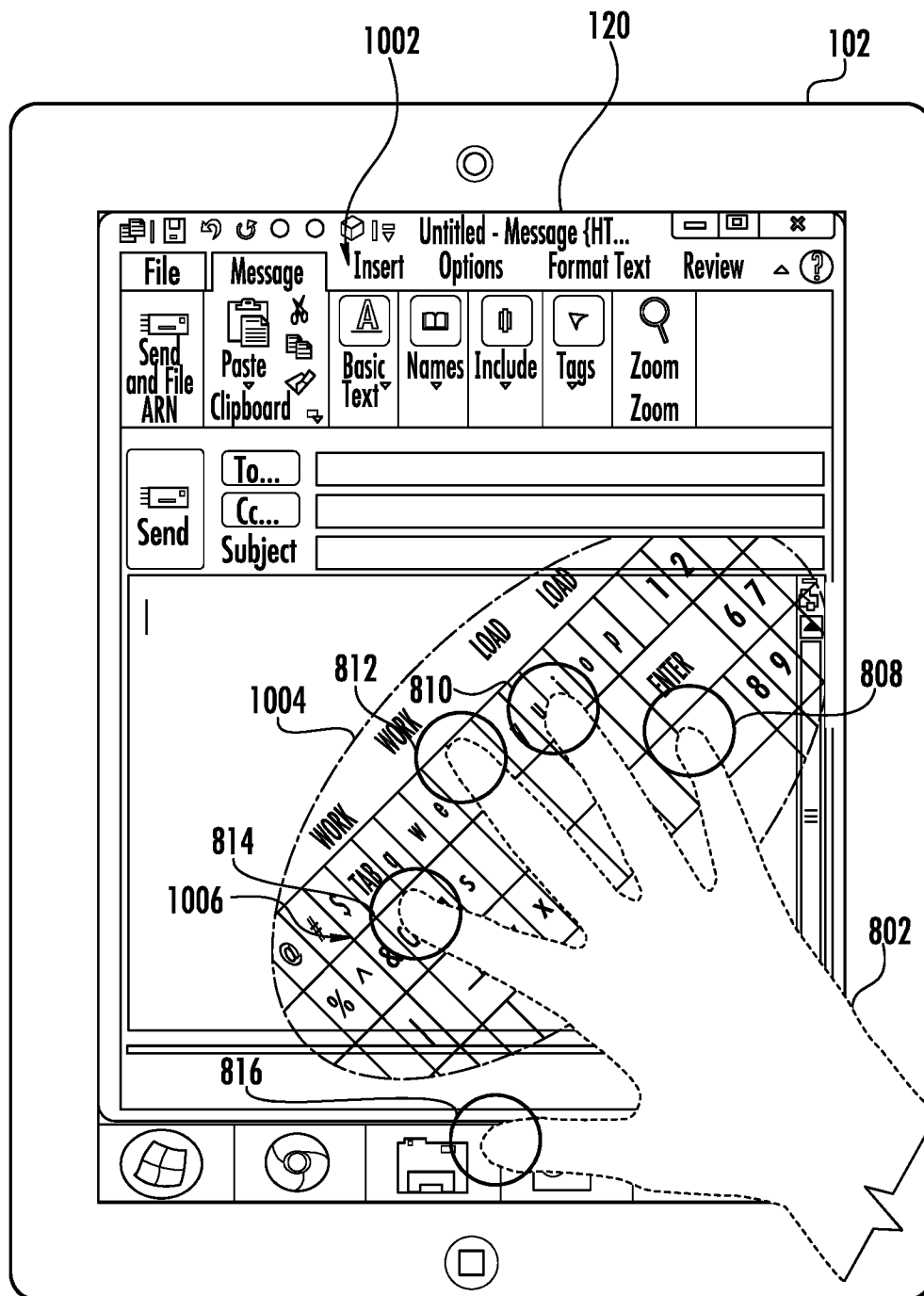


FIG. 10C

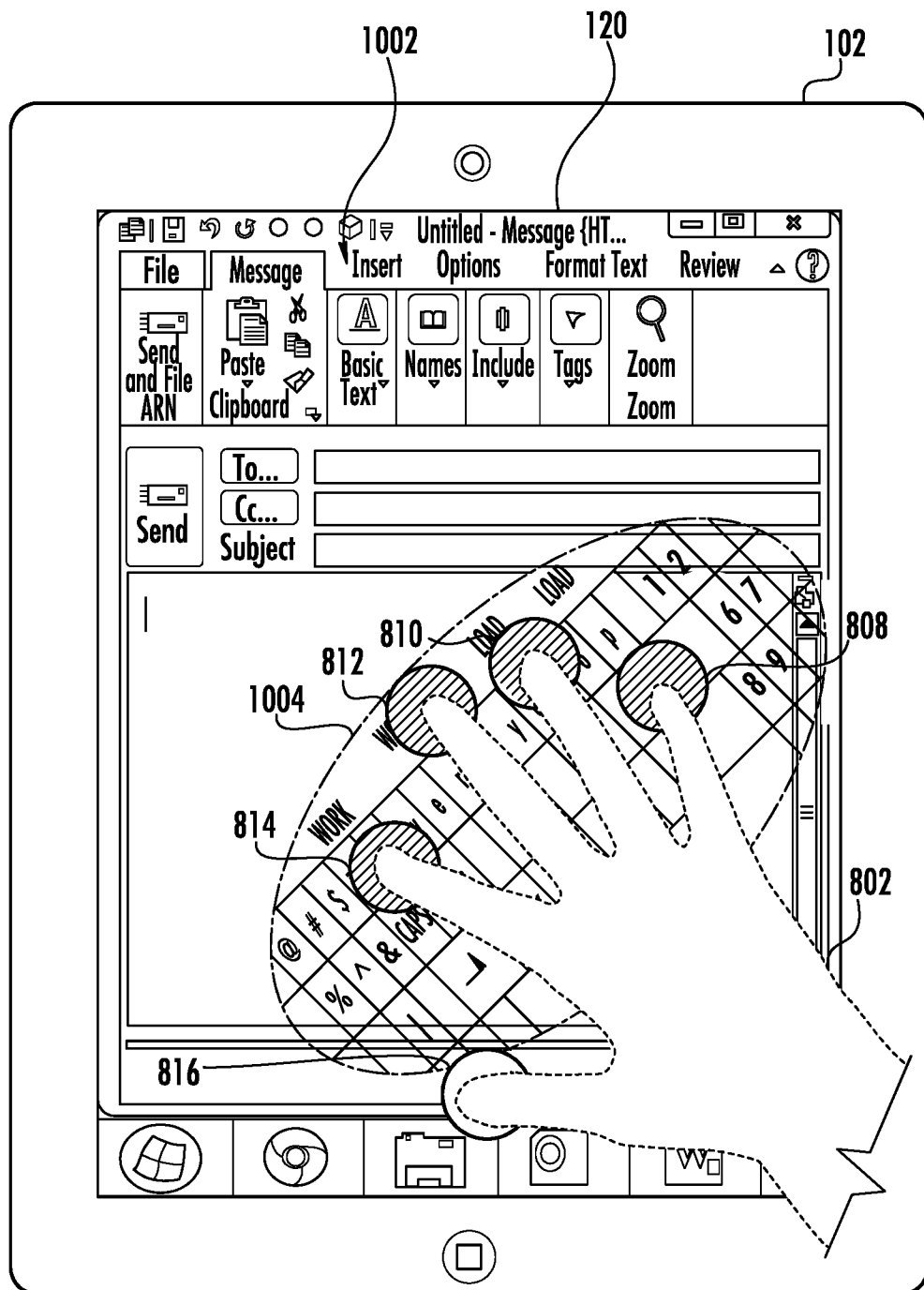


FIG. 11

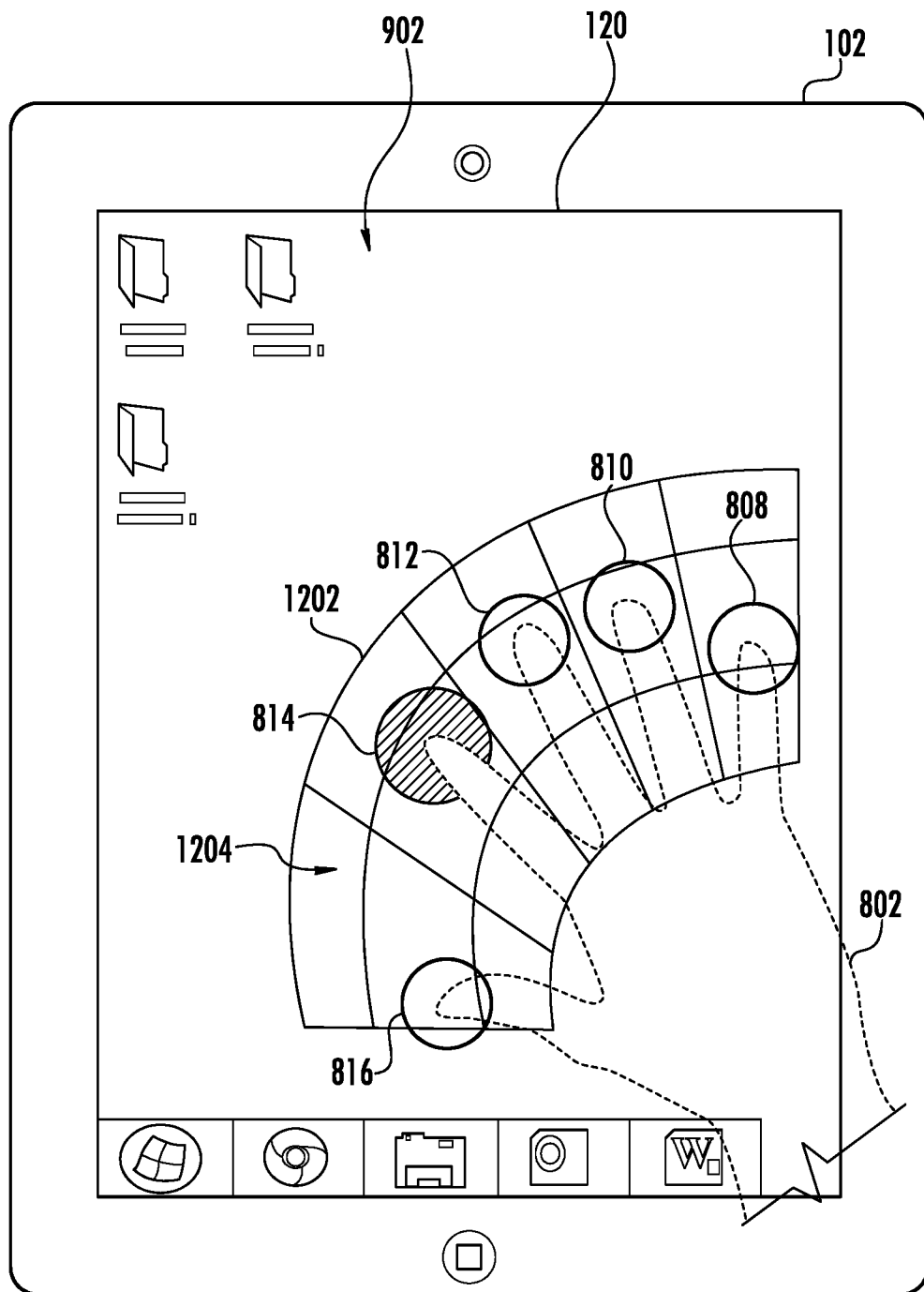


FIG. 12

17/17

1300



APPLICATION <u>1302</u>	DEFAULT SOFT KEYS <u>1304</u>
VIDEO CHAT	FULL SCREEN TOGGLE VOLUME CONTROL BACKGROUND SEGMENTATION INITIATE CALL DROP CALL ADD FRIENDS
INTERNET BROWSER	FAVORITES BOOKMARK NAVIGATION CONTROL SCROLL UP/DOWN SEARCH NEW TAB
VIDEO PLAYBACK	FULL SCREEN TOGGLE VOLUME CONTROL BRIGHTNESS CONTROL PLAY/PAUSE TOGGLE ADJUST VIDEO QUALITY
DOCUMENT READER	FULL SCREEN TOGGLE NEXT SLIDE PREVIOUS SLIDE GO TO SLIDE SEARCH PROJECT
PRESENTATION VIEWER	FULL SCREEN TOGGLE NEXT SLIDE PREVIOUS SLIDE GO TO SLIDE SEARCH PROJECT
IMAGE VIEWER	NEXT PHOTO PREVIOUS PHOTO CHANGE ORIENTATION SHARE SLIDESHOW

FIG. 13

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2016/019779**A. CLASSIFICATION OF SUBJECT MATTER****G06F 3/0488(2013.01)i, G06F 3/048(2006.01)i, G06F 3/01(2006.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 3/0488; G06T 3/60; G06F 3/041; G09G 5/00; G06F 3/01

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: display, GUI, fingers, orientation, rotation, touch, angle

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 8269729 B2 (JEFFERSON Y. HAN et al.) 18 September 2012 See column 4, lines 9-61; column 8, lines 1-3; claim 5; and figures 2A-2D.	1-25
Y	US 2013-0207920 A1 (ERIC MCCANN et al.) 15 August 2013 See paragraphs [0047], [0120]-[0121], [0128]; and figures 10a-10d.	1-25
Y	US 7978182 B2 (BAS ORDING et al.) 12 July 2011 See column 19, lines 32-34; claims 15, 19; and figures 5A-6.	11-12, 23
A	US 2013-0127825 A1 (PUSHKAR P. JOSHI) 23 May 2013 See paragraph [0026]; and figure 8.	1-25
A	US 2010-0079493 A1 (EDWARD TSE et al.) 01 April 2010 See paragraph [0094]; and figure 9d.	1-25



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

08 July 2016 (08.07.2016)

Date of mailing of the international search report

11 July 2016 (11.07.2016)

Name and mailing address of the ISA/KR

International Application Division

Korean Intellectual Property Office

189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea

Facsimile No. +82-42-481-8578

Authorized officer

LEE, EUN KYU

Telephone No. +82-42-481-3580



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2016/019779

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 8269729 B2	18/09/2012	US 2008-0180404 A1	31/07/2008
		US 2008-0180405 A1	31/07/2008
		US 2008-0180406 A1	31/07/2008
		US 8368653 B2	05/02/2013
		US 8674948 B2	18/03/2014
		WO 2008-095132 A2	07/08/2008
		WO 2008-095132 A3	23/10/2008
		WO 2008-095137 A2	07/08/2008
		WO 2008-095137 A3	09/10/2008
		WO 2008-095139 A2	07/08/2008
		WO 2008-095139 A3	16/10/2008
US 2013-0207920 A1	15/08/2013	US 9013430 B2	21/04/2015
		WO 2012-024022 A2	23/02/2012
		WO 2012-024022 A3	12/04/2012
US 7978182 B2	12/07/2011	AU 1999-24673 A1	09/08/1999
		AU 1999-24673 B2	17/04/2003
		AU 2006-291338 A1	22/03/2007
		AU 2006-291338 B2	20/01/2011
		AU 2006-295353 A1	05/04/2007
		AU 2006-295354 A1	05/04/2007
		AU 2006-295354 B2	21/10/2010
		AU 2007-283771 A1	03/04/2008
		AU 2007-283771 B2	01/10/2009
		AU 2007-283771 B8	01/10/2009
		AU 2007-283771 C1	03/06/2010
		AU 2007-286532 A1	03/04/2008
		AU 2007-286532 A8	03/04/2008
		AU 2007-286532 B2	06/08/2009
		AU 2007-286532 B8	06/08/2009
		AU 2007-286532 C1	27/05/2010
		AU 2007-289019 A1	01/05/2008
		AU 2007-289019 B2	04/02/2010
		AU 2007-292383 A1	13/03/2008
		AU 2007-292384 A1	13/03/2008
		AU 2007-292384 B2	01/12/2011
		AU 2007-292473 A1	13/03/2008
		AU 2007-292473 B2	20/05/2010
		AU 2007-342102 A1	17/07/2008
		AU 2007-342102 B2	21/04/2011
		AU 2008-100004 A4	14/02/2008
		AU 2008-100004 B4	05/06/2008
		AU 2008-100004 B4	14/02/2008
		AU 2008-100010 A4	14/02/2008
		AU 2008-100010 B4	05/06/2008
		AU 2008-100010 B4	14/02/2008
		AU 2008-100011 A4	14/02/2008
		AU 2008-100011 B4	05/06/2008

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2016/019779

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		AU 2008-100011 B4	14/02/2008
		AU 2008-201540 A1	05/06/2008
		AU 2008-201540 A8	05/06/2008
		AU 2008-201540 B2	03/09/2009
		AU 2008-203349 A1	20/11/2008
		AU 2008-203349 B2	28/07/2011
		AU 2008-204988 A1	17/07/2008
		AU 2008-204988 B2	21/04/2011
		AU 2008-271011 A1	08/01/2009
		CA 2318815 A1	29/07/1999
		CA 2318815 C	10/08/2004
		CA 2627118 A1	13/03/2008
		CA 2627118 C	06/03/2012
		CA 2658177 A1	17/07/2008
		CA 2658177 C	07/08/2012
		CA 2658413 A1	13/03/2008
		CA 2658413 C	01/11/2011
		CA 2661856 A1	13/03/2008
		CA 2661856 C	21/10/2014
		CA 2661886 A1	13/03/2008
		CA 2661886 C	21/10/2014
		CA 2662134 A1	13/03/2008
		CA 2662134 C	03/11/2015
		CA 2662137 A1	13/03/2008
		CA 2662137 C	24/11/2015
		CA 2666438 A1	02/05/2008
		CA 2666438 C	18/06/2013
		CA 2735309 A1	13/03/2008
		CA 2735309 C	25/08/2015
		CA 2759066 A1	17/07/2008
		CA 2759066 C	04/08/2015
		CA 2759090 A1	17/07/2008
		CA 2759091 A1	17/07/2008
		CA 2762030 A1	13/03/2008
		CA 2762030 C	19/08/2014
		CA 2812011 A1	02/05/2008
		CA 2853273 A1	13/03/2008
		CN 101052939 A	10/10/2007
		CN 101052939 B	04/01/2012
		CN 101198925 A	11/06/2008
		CN 101198925 B	19/06/2013
		CN 101263443 A	10/09/2008
		CN 101263443 B	05/09/2012
		CN 101263448 A	10/09/2008
		CN 101356492 A	28/01/2009
		CN 101356492 B	27/06/2012
		CN 101356493 A	28/01/2009
		CN 101384977 A	11/03/2009
		CN 101384977 B	08/12/2010
		CN 101390039 A	18/03/2009

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2016/019779

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		CN 101482794 A	15/07/2009
		CN 101482794 B	13/08/2014
		CN 101482795 A	15/07/2009
		CN 101482795 B	29/01/2014
		CN 101529367 A	09/09/2009
		CN 101529368 A	09/09/2009
		CN 101529368 B	04/07/2012
		CN 101529874 A	09/09/2009
		CN 101529878 A	09/09/2009
		CN 101529878 B	02/04/2014
		CN 101535938 A	16/09/2009
		CN 101535938 B	08/05/2013
		CN 101535939 A	16/09/2009
		CN 101535939 B	14/12/2011
		CN 101535940 A	16/09/2009
		CN 101535940 B	12/06/2013
		CN 101542424 A	23/09/2009
		CN 101542424 B	12/03/2014
		CN 101558372 A	14/10/2009
		CN 101558372 B	05/03/2014
		CN 101563667 A	21/10/2009
		CN 101563667 B	19/09/2012
		CN 101617288 A	30/12/2009
		CN 101627359 A	13/01/2010
		CN 101627359 B	17/07/2013
		CN 101627361 A	13/01/2010
		CN 101627361 B	12/12/2012
		CN 101627617 A	13/01/2010
		CN 101627617 B	26/06/2013
		CN 101641946 A	03/02/2010
		CN 101641946 B	23/04/2014
		CN 101682585 A	24/03/2010
		CN 101682585 B	12/06/2013
		CN 101727275 A	09/06/2010
		CN 101802817 A	11/08/2010
		CN 101861562 A	13/10/2010
		CN 101893992 A	24/11/2010
		CN 101893992 B	10/10/2012
		CN 102541434 A	04/07/2012
		CN 102541434 B	08/10/2014
		CN 102681776 A	19/09/2012
		CN 102681776 B	06/05/2015
		CN 102722314 A	10/10/2012
		CN 102736848 A	17/10/2012
		CN 102769705 A	07/11/2012
		CN 102769705 B	20/08/2014
		CN 102981727 A	20/03/2013
		CN 102981727 B	13/01/2016
		CN 102981770 A	20/03/2013
		CN 103150104 A	12/06/2013

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2016/019779

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		CN 103150106 A	12/06/2013
		CN 103176701 A	26/06/2013
		CN 103226444 A	31/07/2013
		CN 103294399 A	11/09/2013
		CN 103365595 A	23/10/2013
		CN 103795866 A	14/05/2014
		CN 103914261 A	09/07/2014
		EP 1058924 A1	13/12/2000
		EP 1621989 A2	01/02/2006
		EP 1621989 A3	17/05/2006
		EP 1717677 A2	02/11/2006
		EP 1717677 A3	02/01/2008
		EP 1717677 B1	17/06/2015
		EP 1717678 A2	02/11/2006
		EP 1717678 A3	21/01/2009
		EP 1717679 A2	02/11/2006
		EP 1717679 A3	04/02/2009
		EP 1717680 A2	02/11/2006
		EP 1717680 A3	24/03/2010
		EP 1717681 A2	02/11/2006
		EP 1717681 A3	14/01/2009
		EP 1717681 B1	29/04/2015
		EP 1717682 A2	02/11/2006
		EP 1717682 A3	16/04/2008
		EP 1717683 A2	02/11/2006
		EP 1717683 A3	17/03/2010
		EP 1717684 A2	02/11/2006
		EP 1717684 A3	23/01/2008
		EP 1774427 A2	18/04/2007
		EP 1774429 A2	18/04/2007
		EP 1934685 A1	25/06/2008
		EP 1934686 A1	25/06/2008
		EP 1934693 A2	25/06/2008
		EP 1969453 A2	17/09/2008
		EP 2000893 A2	10/12/2008
		EP 2000893 A3	14/01/2009
		EP 2000894 A2	10/12/2008
		EP 2000894 A3	14/01/2009
		EP 2059868 A2	20/05/2009
		EP 2059868 B1	29/09/2010
		EP 2060096 A1	20/05/2009
		EP 2067094 A1	10/06/2009
		EP 2067094 B1	20/04/2011
		EP 2069895 A1	17/06/2009
		EP 2069895 B1	30/11/2011
		EP 2069898 A1	17/06/2009
		EP 2069898 B1	28/08/2013
		EP 2069899 A1	17/06/2009
		EP 2069982 A2	17/06/2009
		EP 2074500 A2	01/07/2009

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2016/019779

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		EP 2078239 A2	15/07/2009
		EP 2080087 A2	22/07/2009
		EP 2082313 A2	29/07/2009
		EP 2082313 B1	23/07/2014
		EP 2095214 A2	02/09/2009
		EP 2095616 A1	02/09/2009
		EP 2095616 B1	27/08/2014
		EP 2106652 A1	07/10/2009
		EP 2118728 A1	18/11/2009
		EP 2118729 A2	18/11/2009
		EP 2118729 B1	14/11/2012
		EP 2118730 A2	18/11/2009
		EP 2118730 B1	23/01/2013
		EP 2119199 A1	18/11/2009
		EP 2126676 A1	02/12/2009
		EP 2126676 B1	22/05/2013
		EP 2126678 A2	02/12/2009
		EP 2126678 B1	02/11/2011
		EP 2160877 A1	10/03/2010
		EP 2160877 B1	22/02/2012
		EP 2166438 A1	24/03/2010
		EP 2256605 A2	01/12/2010
		EP 2256605 A3	04/01/2012
		EP 2256606 A2	01/12/2010
		EP 2256606 A3	14/12/2011
		EP 2256607 A2	01/12/2010
		EP 2256607 A3	14/12/2011
		EP 2282275 A1	09/02/2011
		EP 2282275 B1	11/12/2013
		EP 2296078 A2	16/03/2011
		EP 2296078 A3	10/12/2014
		EP 2336869 A1	22/06/2011
		EP 2336869 B1	12/02/2014
		EP 2387215 A1	16/11/2011
US 2013-0127825 A1	23/05/2013	US 8581901 B2	12/11/2013
US 2010-0079493 A1	01/04/2010	AU 2009-295320 A1	01/04/2010
		CA 2738185 A1	01/04/2010
		CN 102165405 A	24/08/2011
		EP 2332031 A1	15/06/2011
		EP 2332031 A4	01/05/2013
		US 8810522 B2	19/08/2014
		WO 2010-034122 A1	01/04/2010