



(51) International Patent Classification:
G06F 3/048 (2006.01)

(21) International Application Number:
PCT/US2011/022525

(22) International Filing Date:
26 January 2011 (26.01.2011)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/298,519 26 January 2010 (26.01.2010) US
12/790,516 28 May 2010 (28.05.2010) US

(71) Applicant (for all designated States except US): **APPLE INC.** [US/US]; 1 Infinite Loop, Cupertino, CA 95014 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **CAPELA, Jay, Christopher** [US/US]; 322 Palm Street, Santa Cruz, CA 95060 (US). **MIGOS, Charles, J.** [CA/US]; 1900 Claremont Drive, San Bruno, CA 94066 (US). **THIMBLEBY, William, John** [GB/US]; 395 E. Evelyn Avenue, Sunnyvale, CA 94086 (US). **WEELDREYER, Christopher, Douglas** [US/US]; 228 Beverly Drive, San Carlos, CA 94070 (US).

(74) Agents: **WILLIAMS, Gary, S.** et al.; Morgan Lewis & Bockius LLP, 2 Palo Alto Square, 3000 El Camino Real, Suite 700, Palo Alto, CA 94306 (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, 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, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, 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, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, 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, ML, MR, NE, SN, TD, TG).

Published:

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

(54) Title: DEVICE, METHOD, AND GRAPHICAL USER INTERFACE FOR PRECISE POSITIONING OF OBJECTS

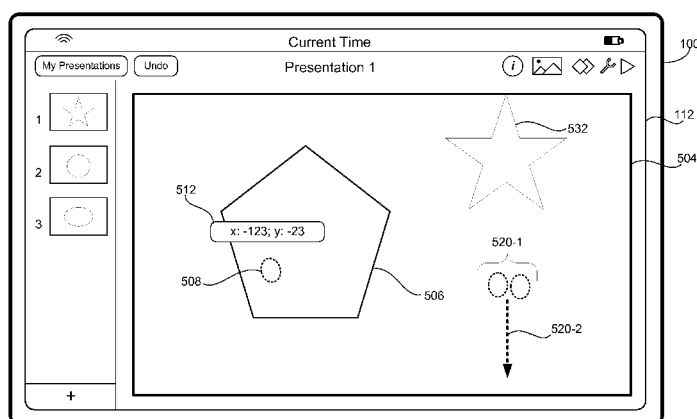


Figure 5E 500E

(57) Abstract: A method includes, at a computing device with a touch-sensitive display: displaying a user interface object on the touch-sensitive display; detecting a contact on the user interface object; while continuing to detect the contact on the user interface object: detecting an M-finger gesture, distinct from the contact, in a first direction on the touch-sensitive display, where M is an integer; and, in response to detecting the M-finger gesture, translating the user interface object a predefined number of pixels in a direction in accordance with the first direction.

Device, Method, and Graphical User Interface for Precise Positioning of Objects

TECHNICAL FIELD

[0001] The disclosed embodiments relate generally to electronic devices with touch-sensitive surfaces, including but not limited to electronic devices with touch-sensitive surfaces configured to precisely move and place user interface objects.

BACKGROUND

[0002] The use of touch-sensitive surfaces as input devices for computers and other electronic computing devices has increased significantly in recent years. Exemplary touch-sensitive surfaces include touch pads and touch screen displays. Such surfaces are widely used to manipulate user interface objects on a display.

[0003] Exemplary manipulations include adjusting the position and/or size of one or more user interface objects. Exemplary user interface objects include digital images, video, text, icons, and other graphics. A user may need to perform such manipulations on user interface objects in an image management application (e.g., Aperture or iPhoto from Apple Inc. of Cupertino, California), a drawing application, a presentation application (e.g., Keynote from Apple Inc. of Cupertino, California), a word processing application (e.g., Pages from Apple Inc. of Cupertino, California), a website creation application (e.g., iWeb from Apple Inc. of Cupertino, California), a disk authoring application (e.g., iDVD from Apple Inc. of Cupertino, California), or a spreadsheet application (e.g., Numbers from Apple Inc. of Cupertino, California).

[0004] But existing methods for performing these manipulations are cumbersome and inefficient. For example, existing keyboard-based methods require memorizing particular keys sequences for moving a user interface object by fine amounts (e.g., in one-pixel increments), which is tedious and creates a significant cognitive burden on a user. In addition, existing keyboard-based methods take longer than necessary, thereby wasting energy. This latter consideration is particularly important in battery-operated devices.

SUMMARY

[0005] Accordingly, there is a need for computing devices with faster, more efficient methods and interfaces for precisely positioning user interface objects that do not require the use of a keyboard. Such methods and interfaces may complement or replace conventional methods for positioning user interface objects. Such methods and interfaces reduce the cognitive burden on a user and produce a more efficient human-machine interface. For battery-operated computing devices, such methods and interfaces conserve power and increase the time between battery charges.

[0006] The above deficiencies and other problems associated with user interfaces for computing devices with touch-sensitive surfaces are reduced or eliminated by the disclosed devices. In some embodiments, the device is a desktop computer. In some embodiments, the device is portable (e.g., a notebook computer, tablet computer, or handheld device). In some embodiments, the device has a touchpad. In some embodiments, the device has a touch-sensitive display (also known as a "touch screen" or "touch screen display"). In some embodiments, the device has a graphical user interface (GUI), one or more processors, memory and one or more modules, programs or sets of instructions stored in the memory for performing multiple functions. In some embodiments, the user interacts with the GUI primarily through finger contacts and gestures on the touch-sensitive surface. In some embodiments, the functions may include image editing, drawing, presenting, word processing, website creating, disk authoring, spreadsheet making, game playing, telephoning, video conferencing, e-mailing, instant messaging, workout support, digital photographing, digital videoing, web browsing, digital music playing, and/or digital video playing. Executable instructions for performing these functions may be included in a computer readable storage medium or other computer program product configured for execution by one or more processors.

[0007] In accordance with some embodiments, a method is performed at a computing device with a touch-sensitive display. The method includes: displaying a user interface object on the touch-sensitive display; detecting a contact on the user interface object; while continuing to detect the contact on the user interface object: detecting an M-finger gesture, distinct from the contact, in a first direction on the touch-sensitive display, where M is an integer; and, in response to detecting the M-finger gesture, translating the user interface object a predefined number of pixels in a direction in accordance with the first direction.

[0008] In accordance with some embodiments, a method is performed at a computing device with a touch-sensitive display. The method includes: displaying a user interface object on the touch-sensitive display; detecting a contact on the user interface object; while continuing to detect the contact on the user interface object: detecting a gesture, distinct from the contact, in a first direction with a first velocity on the touch-sensitive display; and, in response to detecting the gesture: when the first velocity is below a predefined threshold, translating the user interface object a first predefined number of pixels in a direction in accordance with the first direction; and, when the first velocity is above the predefined threshold, translating the user interface object a second predefined number of pixels, distinct from the first predefined number of pixels, in a direction in accordance with the first direction.

[0009] In accordance with some embodiments, a computing device includes a touch-sensitive display, one or more processors, memory, and one or more programs; the one or more programs are stored in the memory and configured to be executed by the one or more processors and the one or more programs include instructions for performing the operations of any of the methods described above. In accordance with some embodiments, a graphical user interface on a computing device with a touch-sensitive display, a memory, and one or more processors to execute one or more programs stored in the memory includes one or more of the elements displayed in any of the methods described above, which are updated in response to inputs, as described in any of the methods above. In accordance with some embodiments, a computer readable storage medium has stored therein instructions which when executed by a computing device with a touch-sensitive display, cause the device to perform the operations of any of the methods described above. In accordance with some embodiments, a computing device includes: a touch-sensitive display; and means for performing the operations of any of the methods described above. In accordance with some embodiments, an information processing apparatus, for use in a computing device with a touch-sensitive display, includes means for performing the operations of any of the methods described above.

[0010] In accordance with some embodiments, a computing device includes a touch-sensitive display unit configured to display user interface objects and receive user contacts; and a processing unit coupled to the touch-sensitive display unit. The processing unit is configured to: enable display of a user interface object on the touch-sensitive display unit; detect a contact on the user interface object; while continuing to detect the contact on the user interface object: detect an M-finger gesture, distinct from the contact, in a first direction on

the touch-sensitive display unit, where M is an integer; and, in response to detecting the M-finger gesture, translate the user interface object a predefined number of pixels in a direction in accordance with the first direction.

[0011] In accordance with some embodiments, a computing device includes a touch-sensitive display unit configured to display user interface objects and receive user contacts; and a processing unit coupled to the display unit and the touch-sensitive display unit. The processing unit is configured to: enable display of a user interface object on the touch-sensitive display unit; detect a contact on the user interface object; while continuing to detect the contact on the user interface object: detect a gesture, distinct from the contact, in a first direction with a first velocity on the touch-sensitive display unit; and, in response to detecting the gesture: when the first velocity is below a predefined threshold, translate the user interface object a first predefined number of pixels in a direction in accordance with the first direction; and, when the first velocity is above the predefined threshold, translate the user interface object a second predefined number of pixels, distinct from the first predefined number of pixels, in a direction in accordance with the first direction.

[0012] Thus, computing devices with touch-sensitive displays are provided with faster, more efficient methods and interfaces for precisely positioning objects, thereby increasing the effectiveness, efficiency, and user satisfaction with such devices. Such methods and interfaces may complement or replace conventional methods for positioning objects.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] For a better understanding of the aforementioned embodiments of the invention as well as additional embodiments thereof, reference should be made to the Description of Embodiments below, in conjunction with the following drawings in which like reference numerals refer to corresponding parts throughout the figures.

[0014] Figures 1A and 1B are block diagrams illustrating portable multifunction devices with touch-sensitive displays in accordance with some embodiments.

[0015] Figure 1C is a block diagram illustrating exemplary components for event handling in accordance with some embodiments.

[0016] Figure 2 illustrates a portable multifunction device having a touch screen in accordance with some embodiments.

[0017] Figure 3 is a block diagram of an exemplary multifunction device with a display and a touch-sensitive surface in accordance with some embodiments.

[0018] Figures 4A and 4B illustrate exemplary user interfaces for a menu of applications on a portable multifunction device in accordance with some embodiments.

[0019] Figure 4C illustrates an exemplary user interface for a multifunction device with a touch-sensitive surface that is separate from the display in accordance with some embodiments.

[0020] Figures 5A-5J illustrate exemplary user interfaces for precisely positioning an object in accordance with some embodiments.

[0021] Figures 6A-6B are flow diagrams illustrating a method of precisely positioning an object in accordance with some embodiments.

[0022] Figures 7A-7B are flow diagrams illustrating a method of precisely positioning an object in accordance with some embodiments.

[0023] Figure 8 is a functional block diagram of a computing device in accordance with some embodiments.

[0024] Figure 9 is a functional block diagram of a computing device in accordance with some embodiments.

DESCRIPTION OF EMBODIMENTS

[0025] Reference will now be made in detail to embodiments, examples of which are illustrated in the accompanying drawings. In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to one of ordinary skill in the art that the present invention may be practiced without these specific details. In other instances, well-known methods, procedures, components, circuits, and networks have not been described in detail so as not to unnecessarily obscure aspects of the embodiments.

[0026] It will also be understood that, although the terms first, second, etc. may be used herein to describe various elements, these elements should not be limited by these terms. These terms are only used to distinguish one element from another. For example, a first contact could be termed a second contact, and, similarly, a second contact could be termed a

first contact, without departing from the scope of the present invention. The first contact and the second contact are both contacts, but they are not the same contact.

[0027] The terminology used in the description of the invention herein is for the purpose of describing particular embodiments only and is not intended to be limiting of the invention. As used in the description of the invention and the appended claims, the singular forms “a”, “an” and “the” are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will also be understood that the term “and/or” as used herein refers to and encompasses any and all possible combinations of one or more of the associated listed items. It will be further understood that the terms “includes,” “including,” “comprises,” and/or “comprising,” when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

[0028] As used herein, the term “if” may be construed to mean “when” or “upon” or “in response to determining” or “in response to detecting,” depending on the context. Similarly, the phrase “if it is determined” or “if [a stated condition or event] is detected” may be construed to mean “upon determining” or “in response to determining” or “upon detecting [the stated condition or event]” or “in response to detecting [the stated condition or event],” depending on the context.

[0029] As used herein, the term “resolution” of a display refers to the number of pixels (also called “pixel counts” or “pixel resolution”) along each axis or in each dimension of the display. For example, a display may have a resolution of 320x480 pixels. Furthermore, as used herein, the term “resolution” of a multifunction device refers to the resolution of a display in the multifunction device. The term “resolution” does not imply any limitations on the size of each pixel or the spacing of pixels. For example, compared to a first display with a 1024x768-pixel resolution, a second display with a 320x480-pixel resolution has a lower resolution. However, it should be noted that the physical size of a display depends not only on the pixel resolution, but also on many other factors, including the pixel size and the spacing of pixels. Therefore, the first display may have the same, smaller, or larger physical size, compared to the second display.

[0030] As used herein, the term “video resolution” of a display refers to the density of pixels along each axis or in each dimension of the display. The video resolution is often

measured in a dots-per-inch (DPI) unit, which counts the number of pixels that can be placed in a line within the span of one inch along a respective dimension of the display.

[0031] Embodiments of computing devices, user interfaces for such devices, and associated processes for using such devices are described. In some embodiments, the computing device is a portable communications device, such as a mobile telephone, that also contains other functions, such as PDA and/or music player functions. Exemplary embodiments of portable multifunction devices include, without limitation, the iPhone® and iPod Touch® devices from Apple Inc. of Cupertino, California. Other portable devices, such as laptops or tablet computers with touch-sensitive surfaces (e.g., touch screen displays and/or touch pads), may also be used. It should also be understood that, in some embodiments, the device is not a portable communications device, but is a desktop computer with a touch-sensitive surface (e.g., a touch screen display and/or a touch pad).

[0032] In the discussion that follows, a computing device that includes a display and a touch-sensitive surface is described. It should be understood, however, that the computing device may include one or more other physical user-interface devices, such as a physical keyboard, a mouse and/or a joystick.

[0033] The device supports a variety of applications, such as one or more of the following: a drawing application, a presentation application, a word processing application, a website creation application, a disk authoring application, a spreadsheet application, a gaming application, a telephone application, a video conferencing application, an e-mail application, an instant messaging application, a workout support application, a photo management application, a digital camera application, a digital video camera application, a web browsing application, a digital music player application, and/or a digital video player application.

[0034] The various applications that may be executed on the device may use at least one common physical user-interface device, such as the touch-sensitive surface. One or more functions of the touch-sensitive surface as well as corresponding information displayed on the device may be adjusted and/or varied from one application to the next and/or within a respective application. In this way, a common physical architecture (such as the touch-sensitive surface) of the device may support the variety of applications with user interfaces that are intuitive and transparent to the user.

[0035] The user interfaces may include one or more soft keyboard embodiments. The soft keyboard embodiments may include standard (QWERTY) and/or non-standard

configurations of symbols on the displayed icons of the keyboard, such as those described in U.S. Patent Applications 11/459,606, "Keyboards For Portable Electronic Devices," filed July 24, 2006, and 11/459,615, "Touch Screen Keyboards For Portable Electronic Devices," filed July 24, 2006, the contents of which are hereby incorporated by reference in their entireties. The keyboard embodiments may include a reduced number of icons (or soft keys) relative to the number of keys in existing physical keyboards, such as that for a typewriter. This may make it easier for users to select one or more icons in the keyboard, and thus, one or more corresponding symbols. The keyboard embodiments may be adaptive. For example, displayed icons may be modified in accordance with user actions, such as selecting one or more icons and/or one or more corresponding symbols. One or more applications on the device may utilize common and/or different keyboard embodiments. Thus, the keyboard embodiment used may be tailored to at least some of the applications. In some embodiments, one or more keyboard embodiments may be tailored to a respective user. For example, one or more keyboard embodiments may be tailored to a respective user based on a word usage history (lexicography, slang, individual usage) of the respective user. Some of the keyboard embodiments may be adjusted to reduce a probability of a user error when selecting one or more icons, and thus one or more symbols, when using the soft keyboard embodiments.

[0036] Attention is now directed toward embodiments of portable devices with touch-sensitive displays. Figures 1A and 1B are block diagrams illustrating portable multifunction devices 100 with touch-sensitive displays 112 in accordance with some embodiments. Touch-sensitive display 112 is sometimes called a "touch screen" for convenience, and may also be known as or called a touch-sensitive display system. Device 100 may include memory 102 (which may include one or more computer readable storage mediums), memory controller 122, one or more processing units (CPU's) 120, peripherals interface 118, RF circuitry 108, audio circuitry 110, speaker 111, microphone 113, input/output (I/O) subsystem 106, other input or control devices 116, and external port 124. Device 100 may include one or more optical sensors 164. These components may communicate over one or more communication buses or signal lines 103.

[0037] It should be appreciated that device 100 is only one example of a portable multifunction device, and that device 100 may have more or fewer components than shown, may combine two or more components, or may have a different configuration or arrangement of the components. The various components shown in Figures 1A and 1B may be

implemented in hardware, software, or a combination of both hardware and software, including one or more signal processing and/or application specific integrated circuits.

[0038] Memory 102 may include high-speed random access memory and may also include non-volatile memory, such as one or more magnetic disk storage devices, flash memory devices, or other non-volatile solid-state memory devices. Access to memory 102 by other components of device 100, such as CPU 120 and the peripherals interface 118, may be controlled by memory controller 122.

[0039] Peripherals interface 118 can be used to couple input and output peripherals of the device to CPU 120 and memory 102. The one or more processors 120 run or execute various software programs and/or sets of instructions stored in memory 102 to perform various functions for device 100 and to process data.

[0040] In some embodiments, peripherals interface 118, CPU 120, and memory controller 122 may be implemented on a single chip, such as chip 104. In some other embodiments, they may be implemented on separate chips.

[0041] RF (radio frequency) circuitry 108 receives and sends RF signals, also called electromagnetic signals. RF circuitry 108 converts electrical signals to/from electromagnetic signals and communicates with communications networks and other communications devices via the electromagnetic signals. RF circuitry 108 may include well-known circuitry for performing these functions, including but not limited to an antenna system, an RF transceiver, one or more amplifiers, a tuner, one or more oscillators, a digital signal processor, a CODEC chipset, a subscriber identity module (SIM) card, memory, and so forth. RF circuitry 108 may communicate with networks, such as the Internet, also referred to as the World Wide Web (WWW), an intranet and/or a wireless network, such as a cellular telephone network, a wireless local area network (LAN) and/or a metropolitan area network (MAN), and other devices by wireless communication. The wireless communication may use any of a plurality of communications standards, protocols and technologies, including but not limited to Global System for Mobile Communications (GSM), Enhanced Data GSM Environment (EDGE), high-speed downlink packet access (HSDPA), wideband code division multiple access (W-CDMA), code division multiple access (CDMA), time division multiple access (TDMA), Bluetooth, Wireless Fidelity (Wi-Fi) (e.g., IEEE 802.11a, IEEE 802.11b, IEEE 802.11g and/or IEEE 802.11n), voice over Internet Protocol (VoIP), Wi-MAX, a protocol for e-mail (e.g., Internet message access protocol (IMAP) and/or post office protocol (POP)), instant

messaging (e.g., extensible messaging and presence protocol (XMPP), Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIMPLE), Instant Messaging and Presence Service (IMPS)), and/or Short Message Service (SMS), or any other suitable communication protocol, including communication protocols not yet developed as of the filing date of this document.

[0042] Audio circuitry 110, speaker 111, and microphone 113 provide an audio interface between a user and device 100. Audio circuitry 110 receives audio data from peripherals interface 118, converts the audio data to an electrical signal, and transmits the electrical signal to speaker 111. Speaker 111 converts the electrical signal to human-audible sound waves. Audio circuitry 110 also receives electrical signals converted by microphone 113 from sound waves. Audio circuitry 110 converts the electrical signal to audio data and transmits the audio data to peripherals interface 118 for processing. Audio data may be retrieved from and/or transmitted to memory 102 and/or RF circuitry 108 by peripherals interface 118. In some embodiments, audio circuitry 110 also includes a headset jack (e.g., 212, Figure 2). The headset jack provides an interface between audio circuitry 110 and removable audio input/output peripherals, such as output-only headphones or a headset with both output (e.g., a headphone for one or both ears) and input (e.g., a microphone).

[0043] I/O subsystem 106 couples input/output peripherals on device 100, such as touch screen 112 and other input control devices 116, to peripherals interface 118. I/O subsystem 106 may include display controller 156 and one or more input controllers 160 for other input or control devices. The one or more input controllers 160 receive/send electrical signals from/to other input or control devices 116. The other input control devices 116 may include physical buttons (e.g., push buttons, rocker buttons, etc.), dials, slider switches, joysticks, click wheels, and so forth. In some alternate embodiments, input controller(s) 160 may be coupled to any (or none) of the following: a keyboard, infrared port, USB port, and a pointer device such as a mouse. The one or more buttons (e.g., 208, Figure 2) may include an up/down button for volume control of speaker 111 and/or microphone 113. The one or more buttons may include a push button (e.g., 206, Figure 2). A quick press of the push button may disengage a lock of touch screen 112 or begin a process that uses gestures on the touch screen to unlock the device, as described in U.S. Patent Application 11/322,549, "Unlocking a Device by Performing Gestures on an Unlock Image," filed December 23, 2005, which is hereby incorporated by reference in its entirety. A longer press of the push button (e.g., 206) may turn power to device 100 on or off. The user may be able to customize a

functionality of one or more of the buttons. Touch screen 112 is used to implement virtual or soft buttons and one or more soft keyboards.

[0044] Touch-sensitive display 112 provides an input interface and an output interface between the device and a user. Display controller 156 receives and/or sends electrical signals from/to touch screen 112. Touch screen 112 displays visual output to the user. The visual output may include graphics, text, icons, video, and any combination thereof (collectively termed "graphics"). In some embodiments, some or all of the visual output may correspond to user-interface objects.

[0045] Touch screen 112 has a touch-sensitive surface, sensor or set of sensors that accepts input from the user based on haptic and/or tactile contact. Touch screen 112 and display controller 156 (along with any associated modules and/or sets of instructions in memory 102) detect contact (and any movement or breaking of the contact) on touch screen 112 and converts the detected contact into interaction with user-interface objects (e.g., one or more soft keys, icons, web pages or images) that are displayed on touch screen 112. In an exemplary embodiment, a point of contact between touch screen 112 and the user corresponds to a finger of the user.

[0046] Touch screen 112 may use LCD (liquid crystal display) technology, LPD (light emitting polymer display) technology, or LED (light emitting diode) technology, although other display technologies may be used in other embodiments. Touch screen 112 and display controller 156 may detect contact and any movement or breaking thereof using any of a plurality of touch sensing technologies now known or later developed, including but not limited to capacitive, resistive, infrared, and surface acoustic wave technologies, as well as other proximity sensor arrays or other elements for determining one or more points of contact with touch screen 112. In an exemplary embodiment, projected mutual capacitance sensing technology is used, such as that found in the iPhone® and iPod Touch® from Apple Inc. of Cupertino, California.

[0047] A touch-sensitive display in some embodiments of touch screen 112 may be analogous to the multi-touch sensitive touchpads described in the following U.S. Patents: 6,323,846 (Westerman et al.), 6,570,557 (Westerman et al.), and/or 6,677,932 (Westerman), and/or U.S. Patent Publication 2002/0015024A1, each of which is hereby incorporated by reference in its entirety. However, touch screen 112 displays visual output from portable device 100, whereas touch sensitive touchpads do not provide visual output.

[0048] A touch-sensitive display in some embodiments of touch screen 112 may be as described in the following applications: (1) U.S. Patent Application No. 11/381,313, "Multipoint Touch Surface Controller," filed May 2, 2006; (2) U.S. Patent Application No. 10/840,862, "Multipoint Touchscreen," filed May 6, 2004; (3) U.S. Patent Application No. 10/903,964, "Gestures For Touch Sensitive Input Devices," filed July 30, 2004; (4) U.S. Patent Application No. 11/048,264, "Gestures For Touch Sensitive Input Devices," filed January 31, 2005; (5) U.S. Patent Application No. 11/038,590, "Mode-Based Graphical User Interfaces For Touch Sensitive Input Devices," filed January 18, 2005; (6) U.S. Patent Application No. 11/228,758, "Virtual Input Device Placement On A Touch Screen User Interface," filed September 16, 2005; (7) U.S. Patent Application No. 11/228,700, "Operation Of A Computer With A Touch Screen Interface," filed September 16, 2005; (8) U.S. Patent Application No. 11/228,737, "Activating Virtual Keys Of A Touch-Screen Virtual Keyboard," filed September 16, 2005; and (9) U.S. Patent Application No. 11/367,749, "Multi-Functional Hand-Held Device," filed March 3, 2006. All of these applications are incorporated by reference herein in their entirety.

[0049] Touch screen 112 may have a video resolution in excess of 100 dpi. In some embodiments, the touch screen has a video resolution of approximately 160 dpi. The user may make contact with touch screen 112 using any suitable object or appendage, such as a stylus, a finger, and so forth. In some embodiments, the user interface is designed to work primarily with finger-based contacts and gestures, which can be less precise than stylus-based input due to the larger area of contact of a finger on the touch screen. In some embodiments, the device translates the rough finger-based input into a precise pointer/cursor position or command for performing the actions desired by the user.

[0050] In some embodiments, in addition to the touch screen, device 100 may include a touchpad (not shown) for activating or deactivating particular functions. In some embodiments, the touchpad is a touch-sensitive area of the device that, unlike the touch screen, does not display visual output. The touchpad may be a touch-sensitive surface that is separate from touch screen 112 or an extension of the touch-sensitive surface formed by the touch screen.

[0051] In some embodiments, device 100 may include a physical or virtual wheel (e.g., a click wheel) as input control device 116. A user may navigate among and interact with one or more graphical objects (e.g., icons) displayed in touch screen 112 by rotating the click wheel or by moving a point of contact with the click wheel (e.g., where the amount of

movement of the point of contact is measured by its angular displacement with respect to a center point of the click wheel). The click wheel may also be used to select one or more of the displayed icons. For example, the user may press down on at least a portion of the click wheel or an associated button. User commands and navigation commands provided by the user via the click wheel may be processed by input controller 160 as well as one or more of the modules and/or sets of instructions in memory 102. For a virtual click wheel, the click wheel and click wheel controller may be part of touch screen 112 and display controller 156, respectively. For a virtual click wheel, the click wheel may be either an opaque or semitransparent object that appears and disappears on the touch screen display in response to user interaction with the device. In some embodiments, a virtual click wheel is displayed on the touch screen of a portable multifunction device and operated by user contact with the touch screen.

[0052] Device 100 also includes power system 162 for powering the various components. Power system 162 may include a power management system, one or more power sources (e.g., battery, alternating current (AC)), a recharging system, a power failure detection circuit, a power converter or inverter, a power status indicator (e.g., a light-emitting diode (LED)) and any other components associated with the generation, management and distribution of power in portable devices.

[0053] Device 100 may also include one or more optical sensors 164. Figures 1A and 1B show an optical sensor coupled to optical sensor controller 158 in I/O subsystem 106. Optical sensor 164 may include charge-coupled device (CCD) or complementary metal-oxide semiconductor (CMOS) phototransistors. Optical sensor 164 receives light from the environment, projected through one or more lens, and converts the light to data representing an image. In conjunction with imaging module 143 (also called a camera module), optical sensor 164 may capture still images or video. In some embodiments, an optical sensor is located on the back of device 100, opposite touch screen display 112 on the front of the device, so that the touch screen display may be used as a viewfinder for still and/or video image acquisition. In some embodiments, an optical sensor is located on the front of the device so that the user's image may be obtained for videoconferencing while the user views the other video conference participants on the touch screen display. In some embodiments, the position of optical sensor 164 can be changed by the user (e.g., by rotating the lens and the sensor in the device housing) so that a single optical sensor 164 may be used along with the touch screen display for both video conferencing and still and/or video image acquisition.

[0054] Device 100 may also include one or more proximity sensors 166. Figures 1A and 1B show proximity sensor 166 coupled to peripherals interface 118. Alternately, proximity sensor 166 may be coupled to input controller 160 in I/O subsystem 106. Proximity sensor 166 may perform as described in U.S. Patent Application Nos. 11/241,839, "Proximity Detector In Handheld Device"; 11/240,788, "Proximity Detector In Handheld Device"; 11/620,702, "Using Ambient Light Sensor To Augment Proximity Sensor Output"; 11/586,862, "Automated Response To And Sensing Of User Activity In Portable Devices"; and 11/638,251, "Methods And Systems For Automatic Configuration Of Peripherals," which are hereby incorporated by reference in their entirety. In some embodiments, the proximity sensor turns off and disables touch screen 112 when the multifunction device is placed near the user's ear (e.g., when the user is making a phone call).

[0055] Device 100 may also include one or more accelerometers 168. Figures 1A and 1B show accelerometer 168 coupled to peripherals interface 118. Alternately, accelerometer 168 may be coupled to an input controller 160 in I/O subsystem 106. Accelerometer 168 may perform as described in U.S. Patent Publication No. 20050190059, "Acceleration-based Theft Detection System for Portable Electronic Devices," and U.S. Patent Publication No. 20060017692, "Methods And Apparatuses For Operating A Portable Device Based On An Accelerometer," both of which are which are incorporated by reference herein in their entirety. In some embodiments, information is displayed on the touch screen display in a portrait view or a landscape view based on an analysis of data received from the one or more accelerometers. Device 100 optionally includes, in addition to accelerometer(s) 168, a magnetometer (not shown) and a GPS (or GLONASS or other global navigation system) receiver (not shown) for obtaining information concerning the location and orientation (e.g., portrait or landscape) of device 100.

[0056] In some embodiments, the software components stored in memory 102 include operating system 126, communication module (or set of instructions) 128, contact/motion module (or set of instructions) 130, graphics module (or set of instructions) 132, text input module (or set of instructions) 134, Global Positioning System (GPS) module (or set of instructions) 135, and applications (or sets of instructions) 136. Furthermore, in some embodiments memory 102 stores device/global internal state 157, as shown in Figures 1A, 1B and 3. Device/global internal state 157 includes one or more of: active application state, indicating which applications, if any, are currently active; display state, indicating what applications, views or other information occupy various regions of touch screen display 112;

sensor state, including information obtained from the device's various sensors and input control devices 116; and location information concerning the device's location and/or attitude.

[0057] Operating system 126 (e.g., Darwin, RTXC, LINUX, UNIX, OS X, WINDOWS, or an embedded operating system such as VxWorks) includes various software components and/or drivers for controlling and managing general system tasks (e.g., memory management, storage device control, power management, etc.) and facilitates communication between various hardware and software components.

[0058] Communication module 128 facilitates communication with other devices over one or more external ports 124 and also includes various software components for handling data received by RF circuitry 108 and/or external port 124. External port 124 (e.g., Universal Serial Bus (USB), FIREWIRE, etc.) is adapted for coupling directly to other devices or indirectly over a network (e.g., the Internet, wireless LAN, etc.). In some embodiments, the external port is a multi-pin (e.g., 30-pin) connector that is the same as, or similar to and/or compatible with the 30-pin connector used on iPod (trademark of Apple Inc.) devices.

[0059] Contact/motion module 130 may detect contact with touch screen 112 (in conjunction with display controller 156) and other touch sensitive devices (e.g., a touchpad or physical click wheel). Contact/motion module 130 includes various software components for performing various operations related to detection of contact, such as determining if contact has occurred (e.g., detecting a finger-down event), determining if there is movement of the contact and tracking the movement across the touch-sensitive surface (e.g., detecting one or more finger-dragging events), and determining if the contact has ceased (e.g., detecting a finger-up event or a break in contact). Contact/motion module 130 receives contact data from the touch-sensitive surface. Determining movement of the point of contact, which is represented by a series of contact data, may include determining speed (magnitude), velocity (magnitude and direction), and/or an acceleration (a change in magnitude and/or direction) of the point of contact. These operations may be applied to single contacts (e.g., one finger contacts) or to multiple simultaneous contacts (e.g., "multitouch"/multiple finger contacts). In some embodiments, contact/motion module 130 and display controller 156 detects contact on a touchpad. In some embodiments, contact/motion module 130 and controller 160 detects contact on a click wheel.

[0060] Contact/motion module 130 may detect a gesture input by a user. Different gestures on the touch-sensitive surface have different contact patterns. Thus, a gesture may be detected by detecting a particular contact pattern. For example, detecting a finger tap gesture includes detecting a finger-down event followed by detecting a finger-up (lift off) event at the same position (or substantially the same position) as the finger-down event (e.g., at the position of an icon). As another example, detecting a finger swipe gesture on the touch-sensitive surface includes detecting a finger-down event followed by detecting one or more finger-dragging events, and subsequently followed by detecting a finger-up (lift off) event.

[0061] Graphics module 132 includes various known software components for rendering and displaying graphics on touch screen 112 or other display, including components for changing the intensity of graphics that are displayed. As used herein, the term “graphics” includes any object that can be displayed to a user, including without limitation text, web pages, icons (such as user-interface objects including soft keys), digital images, videos, animations and the like.

[0062] In some embodiments, graphics module 132 stores data representing graphics to be used. Each graphic may be assigned a corresponding code. Graphics module 132 receives, from applications etc., one or more codes specifying graphics to be displayed along with, if necessary, coordinate data and other graphic property data, and then generates screen image data to output to display controller 156.

[0063] Text input module 134, which may be a component of graphics module 132, provides soft keyboards for entering text in various applications (e.g., contacts 137, e-mail 140, IM 141, browser 147, and any other application that needs text input).

[0064] GPS module 135 determines the location of the device and provides this information for use in various applications (e.g., to telephone 138 for use in location-based dialing, to camera 143 as picture/video metadata, and to applications that provide location-based services such as weather widgets, local yellow page widgets, and map/navigation widgets).

[0065] Applications 136 may include the following modules (or sets of instructions), or a subset or superset thereof:

- contacts module 137 (sometimes called an address book or contact list);
- telephone module 138;

- video conferencing module 139;
- e-mail client module 140;
- instant messaging (IM) module 141;
- workout support module 142;
- camera module 143 for still and/or video images;
- image management module 144;
- video player module 145;
- music player module 146;
- browser module 147;
- calendar module 148;
- widget modules 149, which may include one or more of: weather widget 149-1, stocks widget 149-2, calculator widget 149-3, alarm clock widget 149-4, dictionary widget 149-5, and other widgets obtained by the user, as well as user-created widgets 149-6;
- widget creator module 150 for making user-created widgets 149-6;
- search module 151;
- video and music player module 152, which merges video player module 145 and music player module 146;
- notes module 153;
- map module 154; and/or
- online video module 155.

[0066] Examples of other applications 136 that may be stored in memory 102 include other word processing applications, other image editing applications, drawing applications, presentation applications, JAVA-enabled applications, encryption, digital rights management, voice recognition, and voice replication.

[0067] In conjunction with touch screen 112, display controller 156, contact module 130, graphics module 132, and text input module 134, contacts module 137 may be used to manage an address book or contact list (e.g., stored in application internal state 192 of

contacts module 137 in memory 102 or memory 370), including: adding name(s) to the address book; deleting name(s) from the address book; associating telephone number(s), e-mail address(es), physical address(es) or other information with a name; associating an image with a name; categorizing and sorting names; providing telephone numbers or e-mail addresses to initiate and/or facilitate communications by telephone 138, video conference 139, e-mail 140, or IM 141; and so forth.

[0068] In conjunction with RF circuitry 108, audio circuitry 110, speaker 111, microphone 113, touch screen 112, display controller 156, contact module 130, graphics module 132, and text input module 134, telephone module 138 may be used to enter a sequence of characters corresponding to a telephone number, access one or more telephone numbers in address book 137, modify a telephone number that has been entered, dial a respective telephone number, conduct a conversation and disconnect or hang up when the conversation is completed. As noted above, the wireless communication may use any of a plurality of communications standards, protocols and technologies.

[0069] In conjunction with RF circuitry 108, audio circuitry 110, speaker 111, microphone 113, touch screen 112, display controller 156, optical sensor 164, optical sensor controller 158, contact module 130, graphics module 132, text input module 134, contact list 137, and telephone module 138, videoconferencing module 139 includes executable instructions to initiate, conduct, and terminate a video conference between a user and one or more other participants in accordance with user instructions.

[0070] In conjunction with RF circuitry 108, touch screen 112, display controller 156, contact module 130, graphics module 132, and text input module 134, e-mail client module 140 includes executable instructions to create, send, receive, and manage e-mail in response to user instructions. In conjunction with image management module 144, e-mail client module 140 makes it very easy to create and send e-mails with still or video images taken with camera module 143.

[0071] In conjunction with RF circuitry 108, touch screen 112, display controller 156, contact module 130, graphics module 132, and text input module 134, the instant messaging module 141 includes executable instructions to enter a sequence of characters corresponding to an instant message, to modify previously entered characters, to transmit a respective instant message (for example, using a Short Message Service (SMS) or Multimedia Message Service (MMS) protocol for telephony-based instant messages or using XMPP, SIMPLE, or

IMPS for Internet-based instant messages), to receive instant messages and to view received instant messages. In some embodiments, transmitted and/or received instant messages may include graphics, photos, audio files, video files and/or other attachments as are supported in a MMS and/or an Enhanced Messaging Service (EMS). As used herein, “instant messaging” refers to both telephony-based messages (e.g., messages sent using SMS or MMS) and Internet-based messages (e.g., messages sent using XMPP, SIMPLE, or IMPS).

[0072] In conjunction with RF circuitry 108, touch screen 112, display controller 156, contact module 130, graphics module 132, text input module 134, GPS module 135, map module 154, and music player module 146, workout support module 142 includes executable instructions to create workouts (e.g., with time, distance, and/or calorie burning goals); communicate with workout sensors (sports devices); receive workout sensor data; calibrate sensors used to monitor a workout; select and play music for a workout; and display, store and transmit workout data.

[0073] In conjunction with touch screen 112, display controller 156, optical sensor(s) 164, optical sensor controller 158, contact module 130, graphics module 132, and image management module 144, camera module 143 includes executable instructions to capture still images or video (including a video stream) and store them into memory 102, modify characteristics of a still image or video, or delete a still image or video from memory 102.

[0074] In conjunction with touch screen 112, display controller 156, contact module 130, graphics module 132, text input module 134, and camera module 143, image management module 144 includes executable instructions to arrange, modify (e.g., edit), or otherwise manipulate, label, delete, present (e.g., in a digital slide show or album), and store still and/or video images.

[0075] In conjunction with touch screen 112, display controller 156, contact module 130, graphics module 132, audio circuitry 110, and speaker 111, video player module 145 includes executable instructions to display, present or otherwise play back videos (e.g., on touch screen 112 or on an external, connected display via external port 124).

[0076] In conjunction with touch screen 112, display system controller 156, contact module 130, graphics module 132, audio circuitry 110, speaker 111, RF circuitry 108, and browser module 147, music player module 146 includes executable instructions that allow the user to download and play back recorded music and other sound files stored in one or more

file formats, such as MP3 or AAC files. In some embodiments, device 100 may include the functionality of an MP3 player, such as an iPod (trademark of Apple Inc.).

[0077] In conjunction with RF circuitry 108, touch screen 112, display system controller 156, contact module 130, graphics module 132, and text input module 134, browser module 147 includes executable instructions to browse the Internet in accordance with user instructions, including searching, linking to, receiving, and displaying web pages or portions thereof, as well as attachments and other files linked to web pages.

[0078] In conjunction with RF circuitry 108, touch screen 112, display system controller 156, contact module 130, graphics module 132, text input module 134, e-mail client module 140, and browser module 147, calendar module 148 includes executable instructions to create, display, modify, and store calendars and data associated with calendars (e.g., calendar entries, to do lists, etc.) in accordance with user instructions.

[0079] In conjunction with RF circuitry 108, touch screen 112, display system controller 156, contact module 130, graphics module 132, text input module 134, and browser module 147, widget modules 149 are mini-applications that may be downloaded and used by a user (e.g., weather widget 149-1, stocks widget 149-2, calculator widget 149-3, alarm clock widget 149-4, and dictionary widget 149-5) or created by the user (e.g., user-created widget 149-6). In some embodiments, a widget includes an HTML (Hypertext Markup Language) file, a CSS (Cascading Style Sheets) file, and a JavaScript file. In some embodiments, a widget includes an XML (Extensible Markup Language) file and a JavaScript file (e.g., Yahoo! Widgets).

[0080] In conjunction with RF circuitry 108, touch screen 112, display system controller 156, contact module 130, graphics module 132, text input module 134, and browser module 147, the widget creator module 150 may be used by a user to create widgets (e.g., turning a user-specified portion of a web page into a widget).

[0081] In conjunction with touch screen 112, display system controller 156, contact module 130, graphics module 132, and text input module 134, search module 151 includes executable instructions to search for text, music, sound, image, video, and/or other files in memory 102 that match one or more search criteria (e.g., one or more user-specified search terms) in accordance with user instructions.

[0082] In conjunction with touch screen 112, display controller 156, contact module 130, graphics module 132, and text input module 134, notes module 153 includes executable

instructions to create and manage notes, to do lists, and the like in accordance with user instructions.

[0083] In conjunction with RF circuitry 108, touch screen 112, display system controller 156, contact module 130, graphics module 132, text input module 134, GPS module 135, and browser module 147, map module 154 may be used to receive, display, modify, and store maps and data associated with maps (e.g., driving directions; data on stores and other points of interest at or near a particular location; and other location-based data) in accordance with user instructions.

[0084] In conjunction with touch screen 112, display system controller 156, contact module 130, graphics module 132, audio circuitry 110, speaker 111, RF circuitry 108, text input module 134, e-mail client module 140, and browser module 147, online video module 155 includes instructions that allow the user to access, browse, receive (e.g., by streaming and/or download), play back (e.g., on the touch screen or on an external, connected display via external port 124), send an e-mail with a link to a particular online video, and otherwise manage online videos in one or more file formats, such as H.264. In some embodiments, instant messaging module 141, rather than e-mail client module 140, is used to send a link to a particular online video. Additional description of the online video application can be found in U.S. Provisional Patent Application No. 60/936,562, "Portable Multifunction Device, Method, and Graphical User Interface for Playing Online Videos," filed June 20, 2007, and U.S. Patent Application No. 11/968,067, "Portable Multifunction Device, Method, and Graphical User Interface for Playing Online Videos," filed December 31, 2007, the content of which is hereby incorporated by reference in its entirety.

[0085] Each of the above identified modules and applications correspond to a set of executable instructions for performing one or more functions described above and the methods described in this application (e.g., the computer-implemented methods and other information processing methods described herein). These modules (i.e., sets of instructions) need not be implemented as separate software programs, procedures or modules, and thus various subsets of these modules may be combined or otherwise re-arranged in various embodiments. For example, video player module 145 may be combined with music player module 146 into a single module (e.g., video and music player module 152, Figure 1B). In some embodiments, memory 102 may store a subset of the modules and data structures identified above. Furthermore, memory 102 may store additional modules and data structures not described above.

[0086] In some embodiments, device 100 is a device where operation of a predefined set of functions on the device is performed exclusively through a touch screen and/or a touchpad. By using a touch screen and/or a touchpad as the primary input control device for operation of device 100, the number of physical input control devices (such as push buttons, dials, and the like) on device 100 may be reduced.

[0087] The predefined set of functions that may be performed exclusively through a touch screen and/or a touchpad include navigation between user interfaces. In some embodiments, the touchpad, when touched by the user, navigates device 100 to a main, home, or root menu from any user interface that may be displayed on device 100. In such embodiments, the touchpad may be referred to as a “menu button.” In some other embodiments, the menu button may be a physical push button or other physical input control device instead of a touchpad.

[0088] Figure 1C is a block diagram illustrating exemplary components for event handling in accordance with some embodiments. In some embodiments, memory 102 (in Figures 1A and 1B) or 370 (Figure 3) includes event sorter 170 (e.g., in operating system 126) and a respective application 136-1 (e.g., any of the aforementioned applications 137-151, 155, 380-390).

[0089] Event sorter 170 receives event information and determines the application 136-1 and application view 191 of application 136-1 to which to deliver the event information. Event sorter 170 includes event monitor 171 and event dispatcher module 174. In some embodiments, application 136-1 includes application internal state 192, which indicates the current application view(s) displayed on touch sensitive display 112 when the application is active or executing. In some embodiments, device/global internal state 157 is used by event sorter 170 to determine which application(s) is(are) currently active, and application internal state 192 is used by event sorter 170 to determine application views 191 to which to deliver event information.

[0090] In some embodiments, application internal state 192 includes additional information, such as one or more of: resume information to be used when application 136-1 resumes execution, user interface state information that indicates information being displayed or that is ready for display by application 136-1, a state queue for enabling the user to go back to a prior state or view of application 136-1, and a redo/undo queue of previous actions taken by the user.

[0091] Event monitor 171 receives event information from peripherals interface 118. Event information includes information about a sub-event (e.g., a user touch on touch-sensitive display 112, as part of a multi-touch gesture). Peripherals interface 118 transmits information it receives from I/O subsystem 106 or a sensor, such as proximity sensor 166, accelerometer(s) 168, and/or microphone 113 (through audio circuitry 110). Information that peripherals interface 118 receives from I/O subsystem 106 includes information from touch-sensitive display 112 or a touch-sensitive surface.

[0092] In some embodiments, event monitor 171 sends requests to the peripherals interface 118 at predetermined intervals. In response, peripherals interface 118 transmits event information. In other embodiments, peripheral interface 118 transmits event information only when there is a significant event (e.g., receiving an input above a predetermined noise threshold and/or for more than a predetermined duration).

[0093] In some embodiments, event sorter 170 also includes a hit view determination module 172 and/or an active event recognizer determination module 173.

[0094] Hit view determination module 172 provides software procedures for determining where a sub-event has taken place within one or more views, when touch sensitive display 112 displays more than one view. Views are made up of controls and other elements that a user can see on the display.

[0095] Another aspect of the user interface associated with an application is a set of views, sometimes herein called application views or user interface windows, in which information is displayed and touch-based gestures occur. The application views (of a respective application) in which a touch is detected may correspond to programmatic levels within a programmatic or view hierarchy of the application. For example, the lowest level view in which a touch is detected may be called the hit view, and the set of events that are recognized as proper inputs may be determined based, at least in part, on the hit view of the initial touch that begins a touch-based gesture.

[0096] Hit view determination module 172 receives information related to sub-events of a touch-based gesture. When an application has multiple views organized in a hierarchy, hit view determination module 172 identifies a hit view as the lowest view in the hierarchy which should handle the sub-event. In most circumstances, the hit view is the lowest level view in which an initiating sub-event occurs (i.e., the first sub-event in the sequence of sub-events that form an event or potential event). Once the hit view is identified by the hit view

determination module, the hit view typically receives all sub-events related to the same touch or input source for which it was identified as the hit view.

[0097] Active event recognizer determination module 173 determines which view or views within a view hierarchy should receive a particular sequence of sub-events. In some embodiments, active event recognizer determination module 173 determines that only the hit view should receive a particular sequence of sub-events. In other embodiments, active event recognizer determination module 173 determines that all views that include the physical location of a sub-event are actively involved views, and therefore determines that all actively involved views should receive a particular sequence of sub-events. In other embodiments, even if touch sub-events were entirely confined to the area associated with one particular view, views higher in the hierarchy would still remain as actively involved views.

[0098] Event dispatcher module 174 dispatches the event information to an event recognizer (e.g., event recognizer 180). In embodiments including active event recognizer determination module 173, event dispatcher module 174 delivers the event information to an event recognizer determined by active event recognizer determination module 173. In some embodiments, event dispatcher module 174 stores in an event queue the event information, which is retrieved by a respective event receiver module 182.

[0099] In some embodiments, operating system 126 includes event sorter 170. Alternatively, application 136-1 includes event sorter 170. In yet other embodiments, event sorter 170 is a stand-alone module, or a part of another module stored in memory 102, such as contact/motion module 130.

[00100] In some embodiments, application 136-1 includes a plurality of event handlers 190 and one or more application views 191, each of which includes instructions for handling touch events that occur within a respective view of the application's user interface. Each application view 191 of the application 136-1 includes one or more event recognizers 180. Typically, a respective application view 191 includes a plurality of event recognizers 180. In other embodiments, one or more of event recognizers 180 are part of a separate module, such as a user interface kit (not shown) or a higher level object from which application 136-1 inherits methods and other properties. In some embodiments, a respective event handler 190 includes one or more of: data updater 176, object updater 177, GUI updater 178, and/or event data 179 received from event sorter 170. Event handler 190 may utilize or call data updater 176, object updater 177 or GUI updater 178 to update the application internal state 192.

Alternatively, one or more of the application views 191 includes one or more respective event handlers 190. Also, in some embodiments, one or more of data updater 176, object updater 177, and GUI updater 178 are included in a respective application view 191.

[00101] A respective event recognizer 180 receives event information (e.g., event data 179) from event sorter 170, and identifies an event from the event information. Event recognizer 180 includes event receiver 182 and event comparator 184. In some embodiments, event recognizer 180 also includes at least a subset of: metadata 183, and event delivery instructions 188 (which may include sub-event delivery instructions).

[00102] Event receiver 182 receives event information from event sorter 170. The event information includes information about a sub-event, for example, a touch or a touch movement. Depending on the sub-event, the event information also includes additional information, such as location of the sub-event. When the sub-event concerns motion of a touch the event information may also include speed and direction of the sub-event. In some embodiments, events include rotation of the device from one orientation to another (e.g., from a portrait orientation to a landscape orientation, or vice versa), and the event information includes corresponding information about the current orientation (also called device attitude) of the device.

[00103] Event comparator 184 compares the event information to predefined event or sub-event definitions and, based on the comparison, determines an event or sub-event, or determines or updates the state of an event or sub-event. In some embodiments, event comparator 184 includes event definitions 186. Event definitions 186 contain definitions of events (e.g., predefined sequences of sub-events), for example, event 1 (187-1), event 2 (187-2), and others. In some embodiments, sub-events in an event 187 include, for example, touch begin, touch end, touch movement, touch cancellation, and multiple touching. In one example, the definition for event 1 (187-1) is a double tap on a displayed object. The double tap, for example, comprises a first touch (touch begin) on the displayed object for a predetermined phase, a first lift-off (touch end) for a predetermined phase, a second touch (touch begin) on the displayed object for a predetermined phase, and a second lift-off (touch end) for a predetermined phase. In another example, the definition for event 2 (187-2) is a dragging on a displayed object. The dragging, for example, comprises a touch (or contact) on the displayed object for a predetermined phase, a movement of the touch across touch-sensitive display 112, and lift-off of the touch (touch end). In some embodiments, the event also includes information for one or more associated event handlers 190.

[00104] In some embodiments, event definition 187 includes a definition of an event for a respective user-interface object. In some embodiments, event comparator 184 performs a hit test to determine which user-interface object is associated with a sub-event. For example, in an application view in which three user-interface objects are displayed on touch-sensitive display 112, when a touch is detected on touch-sensitive display 112, event comparator 184 performs a hit test to determine which of the three user-interface objects is associated with the touch (sub-event). If each displayed object is associated with a respective event handler 190, the event comparator uses the result of the hit test to determine which event handler 190 should be activated. For example, event comparator 184 selects an event handler associated with the sub-event and the object triggering the hit test.

[00105] In some embodiments, the definition for a respective event 187 also includes delayed actions that delay delivery of the event information until after it has been determined whether the sequence of sub-events does or does not correspond to the event recognizer's event type.

[00106] When a respective event recognizer 180 determines that the series of sub-events do not match any of the events in event definitions 186, the respective event recognizer 180 enters an event impossible, event failed, or event ended state, after which it disregards subsequent sub-events of the touch-based gesture. In this situation, other event recognizers, if any, that remain active for the hit view continue to track and process sub-events of an ongoing touch-based gesture.

[00107] In some embodiments, a respective event recognizer 180 includes metadata 183 with configurable properties, flags, and/or lists that indicate how the event delivery system should perform sub-event delivery to actively involved event recognizers. In some embodiments, metadata 183 includes configurable properties, flags, and/or lists that indicate how event recognizers may interact with one another. In some embodiments, metadata 183 includes configurable properties, flags, and/or lists that indicate whether sub-events are delivered to varying levels in the view or programmatic hierarchy.

[00108] In some embodiments, a respective event recognizer 180 activates event handler 190 associated with an event when one or more particular sub-events of an event are recognized. In some embodiments, a respective event recognizer 180 delivers event information associated with the event to event handler 190. Activating an event handler 190 is distinct from sending (and deferred sending) sub-events to a respective hit view. In some

embodiments, event recognizer 180 throws a flag associated with the recognized event, and event handler 190 associated with the flag catches the flag and performs a predefined process.

[00109] In some embodiments, event delivery instructions 188 include sub-event delivery instructions that deliver event information about a sub-event without activating an event handler. Instead, the sub-event delivery instructions deliver event information to event handlers associated with the series of sub-events or to actively involved views. Event handlers associated with the series of sub-events or with actively involved views receive the event information and perform a predetermined process.

[00110] In some embodiments, data updater 176 creates and updates data used in application 136-1. For example, data updater 176 updates the telephone number used in contacts module 137, or stores a video file used in video player module 145. In some embodiments, object updater 177 creates and updates objects used in application 136-1. For example, object updater 176 creates a new user-interface object or updates the position of a user-interface object. GUI updater 178 updates the GUI. For example, GUI updater 178 prepares display information and sends it to graphics module 132 for display on a touch-sensitive display.

[00111] In some embodiments, event handler(s) 190 includes or has access to data updater 176, object updater 177, and GUI updater 178. In some embodiments, data updater 176, object updater 177, and GUI updater 178 are included in a single module of a respective application 136-1 or application view 191. In other embodiments, they are included in two or more software modules.

[00112] It shall be understood that the foregoing discussion regarding event handling of user touches on touch-sensitive displays also applies to other forms of user inputs to operate multifunction devices 100 with input-devices, not all of which are initiated on touch screens, *e.g.*, coordinating mouse movement and mouse button presses with or without single or multiple keyboard presses or holds, user movements taps, drags, scrolls, etc., on touch-pads, pen stylus inputs, movement of the device, oral instructions, detected eye movements, biometric inputs, and/or any combination thereof, which may be utilized as inputs corresponding to sub-events which define an event to be recognized.

[00113] Figure 2 illustrates a portable multifunction device 100 having a touch screen 112 in accordance with some embodiments. The touch screen may display one or more graphics within user interface (UI) 200. In this embodiment, as well as others described

below, a user may select one or more of the graphics by making contact or touching the graphics, for example, with one or more fingers 202 (not drawn to scale in the figure) or one or more styluses 203 (not drawn to scale in the figure). In some embodiments, selection of one or more graphics occurs when the user breaks contact with the one or more graphics. In some embodiments, the contact may include a gesture, such as one or more taps, one or more swipes (from left to right, right to left, upward and/or downward) and/or a rolling of a finger (from right to left, left to right, upward and/or downward) that has made contact with device 100. In some embodiments, inadvertent contact with a graphic may not select the graphic. For example, a swipe gesture that sweeps over an application icon may not select the corresponding application when the gesture corresponding to selection is a tap.

[00114] Device 100 may also include one or more physical buttons, such as "home" or menu button 204. As described previously, menu button 204 may be used to navigate to any application 136 in a set of applications that may be executed on device 100. Alternatively, in some embodiments, the menu button is implemented as a soft key in a GUI displayed on touch screen 112.

[00115] In one embodiment, device 100 includes touch screen 112, menu button 204, push button 206 for powering the device on/off and locking the device, volume adjustment button(s) 208, Subscriber Identity Module (SIM) card slot 210, head set jack 212, and docking/charging external port 124. Push button 206 may be used to turn the power on/off on the device by depressing the button and holding the button in the depressed state for a predefined time interval; to lock the device by depressing the button and releasing the button before the predefined time interval has elapsed; and/or to unlock the device or initiate an unlock process. In an alternative embodiment, device 100 also may accept verbal input for activation or deactivation of some functions through microphone 113.

[00116] Figure 3 is a block diagram of an exemplary multifunction device with a display and a touch-sensitive surface in accordance with some embodiments. Device 300 need not be portable. In some embodiments, device 300 is a laptop computer, a desktop computer, a tablet computer, a multimedia player device, a navigation device, an educational device (such as a child's learning toy), a gaming system, or a control device (e.g., a home or industrial controller). Device 300 typically includes one or more processing units (CPU's) 310, one or more network or other communications interfaces 360, memory 370, and one or more communication buses 320 for interconnecting these components. Communication buses 320 may include circuitry (sometimes called a chipset) that interconnects and controls

communications between system components. Device 300 includes input/output (I/O) interface 330 comprising display 340, which is typically a touch screen display. I/O interface 330 also may include a keyboard and/or mouse (or other pointing device) 350 and touchpad 355. Memory 370 includes high-speed random access memory, such as DRAM, SRAM, DDR RAM or other random access solid state memory devices; and may include non-volatile memory, such as one or more magnetic disk storage devices, optical disk storage devices, flash memory devices, or other non-volatile solid state storage devices. Memory 370 may optionally include one or more storage devices remotely located from CPU(s) 310. In some embodiments, memory 370 stores programs, modules, and data structures analogous to the programs, modules, and data structures stored in memory 102 of portable multifunction device 100 (Figure 1), or a subset thereof. Furthermore, memory 370 may store additional programs, modules, and data structures not present in memory 102 of portable multifunction device 100. For example, memory 370 of device 300 may store drawing module 380, presentation module 382, word processing module 384, website creation module 386, disk authoring module 388, and/or spreadsheet module 390, while memory 102 of portable multifunction device 100 (Figure 1) may not store these modules.

[00117] Each of the above identified elements in Figure 3 may be stored in one or more of the previously mentioned memory devices. Each of the above identified modules corresponds to a set of instructions for performing a function described above. The above identified modules or programs (i.e., sets of instructions) need not be implemented as separate software programs, procedures or modules, and thus various subsets of these modules may be combined or otherwise re-arranged in various embodiments. In some embodiments, memory 370 may store a subset of the modules and data structures identified above. Furthermore, memory 370 may store additional modules and data structures not described above.

[00118] Attention is now directed towards embodiments of user interfaces ("UI") that may be implemented on portable multifunction device 100.

[00119] Figures 4A and 4B illustrate exemplary user interfaces for a menu of applications on portable multifunction device 100 in accordance with some embodiments. Similar user interfaces may be implemented on device 300. In some embodiments, user interface 400A includes the following elements, or a subset or superset thereof:

- Signal strength indicator(s) 402 for wireless communication(s), such as cellular and Wi-Fi signals;
- Time 404;
- Bluetooth indicator 405;
- Battery status indicator 406;
- Tray 408 with icons for frequently used applications, such as:
 - Phone 138, which may include an indicator 414 of the number of missed calls or voicemail messages;
 - E-mail client 140, which may include an indicator 410 of the number of unread e-mails;
 - Browser 147; and
 - Music player 146; and
- Icons for other applications, such as:
 - IM 141;
 - Image management 144;
 - Camera 143;
 - Video player 145;
 - Weather 149-1;
 - Stocks 149-2;
 - Workout support 142;
 - Calendar 148;
 - Calculator 149-3;
 - Alarm clock 149-4;
 - Dictionary 149-5; and
 - User-created widget 149-6.

[00120] In some embodiments, user interface 400B includes the following elements, or a subset or superset thereof:

- 402, 404, 405, 406, 141, 148, 144, 143, 149-3, 149-2, 149-1, 149-4, 410, 414, 138, 140, and 147, as described above;
- Map 154;
- Notes 153;
- Settings 412, which provides access to settings for device 100 and its various applications 136, as described further below;
- Video and music player module 152, also referred to as iPod (trademark of Apple Inc.) module 152; and
- Online video module 155, also referred to as YouTube (trademark of Google Inc.) module 155.

[00121] Figure 4C illustrates an exemplary user interface on a device (e.g., device 300, Figure 3) with a touch-sensitive surface 451 (e.g., a tablet or touchpad 355, Figure 3) that is separate from the display 450 (e.g., touch screen display 112). Although many of the examples which follow will be given with reference to inputs on touch screen display 112 (where the touch sensitive surface and the display are combined), in some embodiments, the device detects inputs on a touch-sensitive surface that is separate from the display, as shown in Figure 4C. In some embodiments the touch sensitive surface (e.g., 451 in Figure 4C) has a primary axis (e.g., 452 in Figure 4C) that corresponds to a primary axis (e.g., 453 in Figure 4C) on the display (e.g., 450). In accordance with these embodiments, the device detects contacts (e.g., 460 and 462 in Figure 4C) with the touch-sensitive surface 451 at locations that correspond to respective locations on the display (e.g., in Figure 4C 460 corresponds to 468 and 462 corresponds to 470). In this way, user inputs (e.g., contacts 460 and 462, and movements thereof) detected by the device on the touch-sensitive surface (e.g., 451 in Figure 4C) are used by the device to manipulate the user interface on the display (e.g., 450 in Figure 4C) of the multifunction device when the touch-sensitive surface is separate from the display. It should be understood that similar methods may be used for other user interfaces described herein.

[00122] Additionally, while the following examples are given primarily with reference to finger inputs (e.g., finger contacts, finger swipe gestures), it should be understood that, in some embodiments, one or more of the finger inputs are replaced with input from another input device (e.g., a mouse based input, a pen input, or a stylus input). For example, a continuous contact may be replaced with a mouse click and a holding down of the mouse

click while the cursor is located over the location of the continuous contact. Similarly, when multiple user inputs are simultaneously detected, it should be understood that multiple computer mice may be used simultaneously, or a mouse and finger (or pen or stylus) contacts may be used simultaneously.

[00123] Attention is now directed towards embodiments of user interfaces ("UI") and associated processes that may be implemented on a multifunction device with a display and a touch-sensitive surface, such as device 300 or portable multifunction device 100.

[00124] Figures 5A-5J illustrate exemplary user interfaces for precisely positioning an object in accordance with some embodiments. The user interfaces in these figures are used to illustrate the processes described below, including the processes in Figures 6A-6B and 7A-7B.

[00125] In Figures 5A-5J, some finger contact sizes and contact movement sizes, as well as some object translation sizes, may be exaggerated for illustrative purposes. No depiction in the figures bearing on finger contact movements should be taken as a requirement or limitation for the purpose of understanding sizes and scale associated with the methods and devices disclosed herein.

[00126] Although Figures 5A-5J illustrate exemplary user interfaces containing graphical objects (e.g., shapes), the illustrative discussions below are not intended to be exhaustive or to limit the invention to the precise forms disclosed. For example, similar user interfaces can be used to precisely move and place other objects (e.g., text boxes, tables, charts, diagrams, figures, pictures, photographs, pages, documents, etc.).

[00127] UI 500A (Figure 5A) depicts an example user interface for manipulating user interface objects on touch screen 112 of device 100. For example, UI 500A can be a user interface for a drawing application, a presentation application, an image editing or manipulation application, or any other application where user interface objects can be manipulated. UI 500A includes a content area 502 and a canvas 504 within content area 502. Content area 502 is an area of the user interface where the content of interest is displayed. Canvas 504 within content area 502 represents an area to which content can be added and will be visible, as in a canvas for painting. For example, in a presentation or drawing application, canvas 504 represents a page where the presentation or drawing, respectively, will appear and be visible. Canvas 504 and the contents on canvas 504 (e.g., user interface objects) may be zoomed in and out within content area 502.

[00128] One or more user interface objects can be placed in canvas 504. For example, in UI 500A, user interface object 506 is on canvas 504. Various manipulations can be performed on user interface object 506, including but not limited to resizing the object, entering text within the object, rotating the object, changing the characteristics and properties of the object, and moving the object within canvas 504. At least some of these manipulations can be activated by performing, for example, one or more gestures on touch screen 112. Figure 5A depicts a contact 508 that is detected at a location within the boundary of user interface object 506.

[00129] UI 500B (Figure 5B) depicts contact 508 that continues to be detected within the boundary of user interface object 506. In some embodiments, a bounding box 510 is displayed in response to detection of contact 508, indicating that user interface object 506 has been selected for manipulation.

[00130] In some embodiments, in response to detection of contact 508, a callout 512 showing the position of user interface object 506 on canvas 504 is displayed. In some embodiments, callout 512 is displayed while contact 508 is continuously detected.

[00131] In some embodiments, the position of user interface object 506 is displayed in callout 512 as coordinates of the center or centroid of user interface object 506 on canvas 504. Thus, in Figure 5B, the centroid of user interface object 506 is located at position (-123, -22) on canvas 504. In some embodiments, the point of origin (0, 0) on canvas 504 is located at the top left corner of canvas 504, and the positive directions along the x and y axes are rightward and downward, respectively, from the point of origin. In some other embodiments, the point of origin on canvas 504 is located at the center of canvas 504, and the positive directions along the x and y axes are rightward and downward, respectively, from the point of origin.

[00132] UI 500B also depicts the dragging of user interface object 506 by moving contact 508. Contact 508 moves 513 in a direction on touch screen 112. In response to movement 513, user interface object 506 moves 515 in the same direction as movement 513. In some embodiments, user interface object 506 does not move in response to movement 513 if movement 513 does not exceed a predefined length threshold (e.g., if movement 513 is minute movement caused by jittering of contact 508). Examples of user interface objects not moving in response to minute movement of the contact are described in U.S. Patent Application No. 12/567,642, titled "Device and Method for Jitter Reduction on Touch-

Sensitive Surfaces and Display,” filed September 25, 2009, which is incorporated by reference herein in its entirety. Further, in some embodiments, as the amount of elapsed time during which movement of contact 508 does not exceed the length threshold increases, the length threshold itself increases.

[00133] UI 500C (Figure 5C) depicts a finger gesture (e.g., a flick or swipe gesture) detected on touch screen 112 while contact 508 continues to be detected. The gesture can be initiated from within the boundary (or in some embodiments, from within the bounding box 510) of user interface object 506 or in an empty area (i.e., area not occupied by another user interface object) on canvas 504. For example, gesture 514, initiated from an empty area of canvas 504, includes a contact 514-1 on touch screen 112 that moves 514-2 downward. As another example, gesture 516, initiated from within the boundary of user interface object 506, includes a contact 516-1 on touch screen 112 that moves 516-2 downward. In some embodiments, the gesture may also be initiated from within the boundary or bounding box of another user interface object besides object 506 on canvas 504. For example, gesture 534, initiated from within the boundary of user interface object 532, includes a contact 534-1 on touch screen 112 that moves 534-2 downward.

[00134] UI 500D (Figure 5D) depicts translation of user interface object 506 in response to detection of gesture 514, 516, or 534. In response to detection of any one of gestures 514, 516 or 534, user interface object 506 translates 518 downward by a predefined number of pixels. In some embodiments, the predefined number of pixels is one pixel. The position of user interface object 506 as shown in callout 512 changes to reflect the translation. For example, in Figure 5D, the y-coordinate shown in callout 512 is changed from -23 to -22, reflecting the change in the vertical position of user interface object 506 due to the translation. Depending on the zoom level of canvas 504, the translation of user interface object 506 may not be apparent to the user because of the fineness of the amount of translation relative to the zoom level of canvas 504. While contact 508 continues to be detected, additional finger gestures (e.g., a flick or swipe) can be performed to move user interface object 506 by another increment of the predefined number of pixels in the same direction as the gesture or any other direction.

[00135] In some embodiments, translation of user interface object 506 in response to detection of a finger gesture is constrained to be along the horizontal or the vertical in accordance with the gesture. For example, if the finger gesture is substantially horizontal, user interface object 506 translates horizontally. If the finger gesture is substantially vertical,

user interface object 506 translates vertically. In some embodiments, a gesture is substantially in a direction (up or down vertically, right or left horizontally) if the gesture is exactly in that direction or within a predefined range of angles toward that direction. For example, if the finger gesture is rightward but is not exactly horizontal, the user interface object still translates rightward horizontally if the gesture is within a predefined range of angles (e.g., 20°, 30°, or 40°) above or below the rightward horizontal.

[00136] In some embodiments, if the gesture is not exactly horizontal or vertical, the direction of translation for user interface object 506 can be determined by comparing the magnitude of the horizontal component of gesture with the magnitude of the vertical component of the gesture; the translation of user interface object 506 is horizontal if the horizontal magnitude is larger, and the translation is vertical if the vertical magnitude is larger.

[00137] In some embodiments, user interface object 506 translates by different predefined numbers of pixels in response to a finger gesture based on the velocity of the finger gesture. For example, returning to Figures 5C-5D, if the velocity of gesture 514/516 is below a predefined threshold, user interface object 506 translates 518 by a first predefined number of pixels (e.g., one pixel) in response to gesture 514/516. If the velocity of gesture 514/ 516 is above the predefined threshold, user interface object 506 translates 518 by a second predefined number of pixels (e.g., ten pixels) in response to gesture 514/516.

[00138] In some embodiments, user interface object 506 translates by different predefined numbers of pixels in response to a finger gesture based on the number of fingers used in the finger gesture. UI 500E (Figure 5E) depicts a two-finger gesture (e.g., a two-finger flick or swipe gesture) detected on touch screen 112 while contact 508 continues to be detected. The gesture can be initiated from within the boundary of user interface object 506, in an empty area on canvas 504, or within the boundary or bounding box of another user interface object 532 on canvas 504. For example, gesture 520, includes a two-finger contact 520-1 on touch screen 112 that moves 520-2 downward.

[00139] UI 500F (Figure 5F) depicts translation of user interface object 506 in response to detection of gesture 520 (Figure 5E). In response to detection of gesture 520, user interface object 506 translates 522 downward by a different predefined number of pixels than if the gesture used one finger. In some embodiments, the different predefined number of pixels is ten pixels. The position of user interface object 506 as shown in callout 512

changes to reflect the translation. For example, in Figure 5F, the y-coordinate in callout 512 is changed from -23 to -13, reflecting the change in the vertical position of user interface object 506 due to the translation.

[00140] UI 500G (Figure 5G) depicts a three-finger gesture (e.g., a flick or swipe gesture) detected on touch screen 112 while contact 508 continues to be detected. The gesture can be initiated from within the boundary of user interface object 506 or in an empty area (i.e., area not occupied by another user interface object) on canvas 504. For example, gesture 524 includes a three-finger contact 524-1 on touch screen 112 that moves 524-2 downward.

[00141] UI 500H (Figure 5H) depicts translation of user interface object 506 in response to detection of gesture 524 (Figure 5G). In response to detection of gesture 524, user interface object 506 translates 526 downward by another predefined number of pixels (e.g., 20 pixels, 50 pixels (shown in Figure H), or 100 pixels). The position of user interface object 506 as shown in callout 512 changes to reflect the translation. For example, in Figure 5H, the y-coordinate in callout 512 is changed from -23 to 27, reflecting the change in the vertical position of user interface object 506 due to the translation.

[00142] Thus, in some embodiments, by changing the number of contacts (e.g., fingers) used in the gesture, the number of pixels by which user interface object 506 translates per detected gesture changes. In the examples described above, user interface object 506 translates 1 pixel per one-finger gesture, 10 pixels per two-finger gesture, and 50 pixels per three-finger gesture. In some embodiments, the number of pixels to translate is predefined for the respective number of fingers used in the gesture. Similarly, in some embodiments, if the number of pixels for translation is based on velocity of the gesture, numbers of pixels for translation may be predefined for different velocity gradations (e.g., one pixel for a low velocity gesture, 10 pixels for a medium velocity gesture, 50 pixels for a high speed gesture).

[00143] It should be appreciated that a user may perform any number of finger gestures in sequence, where the number of fingers and/or direction of a respective gesture is the same as or different from the number of fingers and/or direction of the immediately preceding gesture, to move the user interface object by a desired number of pixels in any desired direction. Thus, for example, a user may perform a one-finger gesture, then a two-finger gesture, then 2 one-finger gestures in a row, and so forth. Similarly, in embodiments

where the number of pixels for translation is based on the gesture velocity, the gesture velocity may vary from gesture to gesture.

[00144] In some embodiments, user interface object 506 can be moved by a fine amount diagonally, as well as horizontally or vertically as described above. UI 500I (Figure 5I) depicts a diagonal one-finger gesture (e.g., a flick or swipe gesture) detected on touch screen 112 while contact 508 continues to be detected. The gesture can be initiated from within the boundary of user interface object 506 or in an empty area (i.e., area not occupied by another user interface object) in canvas 504. For example, gesture 528, includes a contact 528-1 on touch screen 112 that moves 528-2 upward and to the left.

[00145] UI 500J (Figure 5J) depicts translation of user interface object 506 in response to detecting a diagonal gesture. In response to detection of gesture 528 (Figure 5I), user interface object 506 moves 530 diagonally, at a 45-degree angle, by a predefined number of pixels horizontally and by the same predefined number of pixels vertically. In some embodiments, the predefined number of pixels is one pixel. The position of user interface object 506 as shown in callout 512 changes to reflect the translation. For example, in Figure 5H, the coordinates in callout 512 is changed from (-123, -23) to (-124, -24), reflecting the change in the horizontal and vertical position of user interface object 506 due to the translation.

[00146] Figures 6A-6B are flow diagrams illustrating a method 600 of precisely positioning an object in accordance with some embodiments. The method 600 is performed at a multifunction device (e.g., device 300, Figure 3, or portable multifunction device 100, Figure 1) with a display and a touch-sensitive surface. In some embodiments, the display is a touch screen display and the touch-sensitive surface is on the display. In some embodiments, the display is separate from the touch-sensitive surface. Some operations in method 600 may be combined and/or the order of some operations may be changed.

[00147] As described below, the method 600 provides an intuitive way to precisely move and position an object without using a keyboard. The method reduces the cognitive burden on a user when moving an object by fine amounts, thereby creating a more efficient human-machine interface. For battery-operated computing devices, enabling a user to move an object by fine amounts faster and more efficiently conserves power and increases the time between battery charges.

[00148] The device displays (602) a user interface object on the touch-sensitive display (e.g., a digital image, video, text, geometric object, or other object in a drawing application, presentation application, word processing application, spreadsheet application, website creation application, or other application). For example, in Figure 5A, a user interface object 506 (in this case, a geometric object) is displayed in UI 500A of a presentation application on touch screen 112.

[00149] The device detects (604) a contact on the user interface object. The detected contact can be, for example, a finger contact. In some embodiments, the contact is detected on the user interface object if it is detected on a location on touch screen 112 that corresponds to a location within the boundary of the user interface object. In some embodiments, the contact is detected on the user interface object if it is detected on a location on touch screen 112 that corresponds to a location within a bounding box surrounding the user interface object. For example, in Figure 5B, contact 508 is detected on user interface object 506. Contact 508 is within the boundary of user interface object 506 and within bounding box 510 surrounding user interface object 506.

[00150] While continuing to detect the contact on the user interface object (606), the device detects (608) an M-finger gesture, distinct from the contact, in a first direction on the touch-sensitive display, where M is an integer. The M-finger gesture can be, for example, a flick or swipe gesture up, down, right, or left on the touch-sensitive display. The gesture can be initiated from within the user interface object (or from within a bounding box surrounding the user interface object), in an empty area, or within a user interface object (or within its bounding box) other than the one for which movement is desired. For example, Figure 5C depicts a downward gesture 514, in the empty area of canvas 504 on touch screen 112, distinct from contact 508, while contact 508 continues to be detected. Gesture 514 includes one finger, which is represented by contact 514-1. As another example, Figure 5C also depicts a downward gesture 516, initiated from within user interface object 506 on touch screen 112, distinct from contact 508, while contact 508 continues to be detected. Gesture 516 includes one finger, which is represented by contact 516-1. As another example, Figure 5C also depicts a downward gesture 534, initiated from within user interface object 532 on touch screen 112, distinct from contact 508, while contact 508 continues to be detected. Gesture 534 includes one finger, which is represented by contact 534-1.

[00151] While continuing to detect the contact on the user interface object (606), in response to detecting the M-finger gesture, the device translates (610) the user interface

object a predefined number of pixels in a direction in accordance with the first direction. For example, in Figure 5D, in response to gesture 514 or 516, user interface object 506 translates 518 by a predefined number of pixels (which, in the case of Figure 5D, is one pixel for a one-finger gesture) in a direction (downward) in accordance with the direction (downward) of gesture 514/516.

[00152] In some embodiments, rather than translating the object, the M-finger gesture changes the size of the user interface object by a predefined number of pixels (e.g., when the contact (e.g., contact 508) is on a resizing handle for the user interface object). In some embodiments, rather than translating the object, the M-finger gesture rotates the user interface object by a predefined angle (e.g., when the contact (e.g., contact 508) is on a rotational handle for the user interface object).

[00153] In some embodiments, the predefined number of pixels is one pixel when M is one (612). When the gesture includes one finger, the user interface object translates by one pixel. For example, in Figure 5D, in response to one-finger gesture 514/516, user interface object 506 translates 518 by one pixel.

[00154] In some embodiments, the predefined number of pixels is ten pixels when M is two (614). When the gesture includes two fingers, the user interface object translates by ten pixels. For example, in Figure 5E-5F, in response to two-finger gesture 520, user interface object 506 translates 522 by ten pixels.

[00155] In some embodiments, the predefined number of pixels is 1, 10, and 20 pixels for M = 1, 2, and 3, respectively. In some other embodiments, the predefined number of pixels is 1, 10, and 50 pixels for M = 1, 2, and 3, respectively. Examples in which M=1 or M=2 are described above. In some further embodiments, the number of pixels for translation is also predefined for values of M greater than 3. Figures 5G-5H depicts an example in which the predefined number of pixels is 50 for M=3. In Figure 5G-5H, in response to three-finger gesture 524, user interface object 506 translates 526 by 50 pixels. Further, it should be appreciated that M may vary from gesture to gesture in a sequence of gestures; a user may perform a two-finger gesture, then a three-finger gesture, then several one-finger gestures in a row, and so on. In other words, user may perform a sequence of M-finger gestures, where M and/or the direction for one gesture in the sequence may be the same as or different from that of the immediately preceding gesture, to move the user interface object to a desired location on the display.

[00156] In some embodiments, the first direction is a rightward direction (e.g., to the right or within a predefined range of angles towards the right) and the user interface object moves to the right; the first direction is a leftward direction (e.g., to the left or within a predefined range of angles towards the left) and the user interface object moves to the left; the first direction is an upward direction (e.g., vertically up or within a predefined range of angles of being vertically up) and the user interface object moves up; or the first direction is a downward direction (e.g., vertically down or within a predefined range of angles of being vertically down) and the user interface object moves down (616). The direction (up, down, left, right) of the translation of the user interface object follows the direction (up, down, left, right, respectively) of the gesture, but may be constrained to the vertical or horizontal directions. For example, in Figures 5C-5D, user interface object 506, following the downward direction of gesture 514/516, translates 518 downward. In some embodiments, if the gesture is not exactly vertical or horizontal, the translation is nevertheless vertical or horizontal, respectively, if the gesture is within a predefined range of angles from the vertical or horizontal, respectively. For example, in Figures 5C-5D, if movement 514-2 of gesture 514 is not exactly vertically downward, translation 518 of user interface object 506 is still downward if movement 514-2 is within a predefined range of angles (e.g., 20°, 30°, or 40° to the left or to the right) of the downward vertical.

[00157] In some embodiments, the first direction is a diagonal direction and the user interface object moves in the diagonal direction (618). If the direction of the gesture is a diagonal direction (e.g., at a 45° from the vertical or horizontal axis or within a predefined range of angles (e.g., 15°) from the 45° line), then the user interface object translates in the diagonal direction. In some embodiments, translation in the diagonal direction includes translation by a predefined number of pixels (e.g., 1 pixel for M=1, 10 pixels for M=2, etc.) horizontally and by the same predefined number of pixels vertically. For example, in Figures 5I-5J, user interface object 506 translates 530 one pixel upward and one pixel leftward in response to diagonal gesture 528.

[00158] In some embodiments, prior to detecting the M-finger gesture, the device detects (620) movement of the contact across the touch sensitive display, and moves (622) the user interface object in accordance with the movement of the contact. In some embodiments, there may be coarse movement/positioning of the user interface object (e.g., by dragging the user interface object), which moves in concert with the movement of the contact, followed by precise positioning of the user interface object with the M-finger gesture. For

example, in Figure 5B, contact 508 moves 513 in a direction on the touch screen 112, and in response to detecting movement 513, user interface object 506 moves 515 in accordance with movement 513; user interface object 506 is dragged by moving contact 508. Movement of the user interface object by dragging may be coarse relative to movement using M-finger gestures as described above.

[00159] In some embodiments, the device displays (624) location indicia of the user interface object while the contact is detected on the touch-sensitive display. For example, in Figure 5B, a temporary text box (e.g., callout 512, a heads-up display) with the coordinates of user interface object 506 appears proximate to contact 508 while contact 508 is detected on the touch-sensitive display. The coordinates in callout 512 change as user interface object 506 moves or translates, as shown in Figures 5D, 5F, 5H, 5J. In some embodiments, when the user interface object translates in response to a gesture, the un-affected coordinate in the location indicia is grayed out, dimmed, or otherwise de-emphasized. For example, in Figure 5D, as user interface object 506 translates 518 downward, the x-coordinate in callout 512 may be dimmed.

[00160] Figures 7A-7B are flow diagrams illustrating a method 700 of precisely positioning an object in accordance with some embodiments. The method 700 is performed at a multifunction device (e.g., device 300, Figure 3, or portable multifunction device 100, Figure 1) with a display and a touch-sensitive surface. In some embodiments, the display is a touch screen display and the touch-sensitive surface is on the display. In some embodiments, the display is separate from the touch-sensitive surface. Some operations in method 700 may be combined and/or the order of some operations may be changed.

[00161] As described below, the method 700 provides an intuitive way to precisely move and position an object without using a keyboard. The method reduces the cognitive burden on a user when moving an object by fine amounts, thereby creating a more efficient human-machine interface. For battery-operated computing devices, enabling a user to move an object by fine amounts faster and more efficiently conserves power and increases the time between battery charges.

[00162] The device displays (702) a user interface object on the touch-sensitive display (e.g., a digital image, video, text, geometric object, or other object in a drawing application, presentation application, word processing application, spreadsheet application, website creation application, or other application). For example, in Figure 5A, a user interface object

506 (in this case, a geometric object) is displayed in UI 500A of a presentation application on touch screen 112.

[00163] The device detects (704) a contact on the user interface object. The detected contact can be, for example, a finger contact. In some embodiments, the contact is detected on the user interface object if it is detected on a location on touch screen 112 that corresponds to a location within the boundary of the user interface object. In some embodiments, the contact is detected on the user interface object if it is detected on a location on touch screen 112 that corresponds to a location within a bounding box surrounding the user interface object. For example, in Figure 5B, contact 508 is detected on user interface object 506. Contact 508 is within the boundary of user interface object 506 and within bounding box 510 surrounding user interface object 506.

[00164] While continuing to detect the contact on the user interface object (706), the device detects (708) a gesture, distinct from the contact, in a first direction with a first velocity on the touch-sensitive display. The gesture can be, for example, a flick or swipe gesture up, down, right, or left on the touch-sensitive display. The gesture can be initiated from within the user interface object (or from within a bounding box surrounding the user interface object) or in an empty area. For example, Figure 5C depicts a downward gesture 514, in the empty area of canvas 504 on touch screen 112, distinct from contact 508, while contact 508 continues to be detected. As another example, Figure 5C also depicts a downward gesture 516, initiated from within user interface object 506 on touch screen 112, distinct from contact 508, while contact 508 continues to be detected.

[00165] The gesture has a first velocity. For example, in Figure 5C, gesture 514 has a velocity; contact 514-1 moves 514-2 at some speed. Similarly, gesture 516 has a velocity; contact 516-1 moves 516-2 at some speed.

[00166] In response to detecting the gesture (710), when the first velocity is below a predefined threshold, the device translates (712) the user interface object a first predefined number of pixels in a direction in accordance with the first direction; and when the first velocity is above the predefined threshold, the device translates (714) the user interface object a second predefined number of pixels, distinct from the first predefined number of pixels, in a direction in accordance with the first direction. For example, in Figures 5C-5D, if the velocity of gesture 514 is below a first predefined threshold, user interface object 506 translates 518 by a first predefined number of pixels (e.g., one pixel). If the velocity of

gesture 514 is greater than the first predefined threshold, user interface object 506 translates 518 by a second predefined number of pixels (e.g., ten pixels) that is different from the first predefined number.

[00167] In some embodiments, rather than translating the object, the gesture changes the size of the user interface object by a predefined number of pixels (e.g., when the contact (e.g., contact 508) is on a resizing handle for the user interface object). In some embodiments, rather than translating the object, the gesture rotates the user interface object by a predefined angle (e.g., when the contact (e.g., contact 508) is on a rotational handle for the user interface object).

[00168] In some embodiments, the first predefined number of pixels is one pixel (716). When the gesture is below the first predefined threshold, the user interface object translates by one pixel. For example, in Figures 5C-5D, in response to gesture 514/516, if the velocity of gesture 514/516 is below a first predefined threshold, user interface object 506 translates 518 by one pixel.

[00169] In some embodiments, the second predefined number of pixels is ten pixels (718). When the velocity is greater than the first predefined threshold, the user interface object translates by ten pixels. For example, in Figures 5C-5D, in response to gesture 514/516, if the velocity of gesture 514/516 is above the first predefined threshold, user interface object 506 translates 518 by ten pixels.

[00170] In some embodiments, there can be multiple predefined thresholds and corresponding predefined numbers of pixels by which the user interface object can translate. For example, if gesture 514 does not exceed a first threshold, user interface object 506 translates 518 by a first predefined number of pixels (e.g., 1 pixel). If gesture 514 exceeds the first threshold but does not exceed a second threshold, user interface object 506 translates 518 by a second predefined number of pixels (e.g., 10 pixels). If gesture 514 exceeds the second threshold (and thus also exceeds the first threshold), user interface object 506 translates 518 by a third predefined number of pixels (e.g., 20 pixels).

[00171] In some embodiments, the first direction is a rightward direction (e.g., to the right or within a predefined range of angles towards the right) and the user interface object moves to the right; the first direction is a leftward direction (e.g., to the left or within a predefined range of angles towards the left) and the user interface object moves to the left; the first direction is an upward direction (e.g., vertically up or within a predefined range of

angles of being vertically up) and the user interface object moves up; or the first direction is a downward direction (e.g., vertically down or within a predefined range of angles of being vertically down) and the user interface object moves down (720). The direction (up, down, left, right) of the translation of the user interface object follows the direction (up, down, left, right, respectively) of the gesture, but may be constrained to the vertical or horizontal directions. For example, in Figures 5C-5D, user interface object 506, following the downward direction of gesture 514/516, translates 518 downward. In some embodiments, if the gesture is not exactly vertical or horizontal, the translation is nevertheless vertical or horizontal, respectively, if the gesture is within a predefined range of angles from the vertical or horizontal, respectively. For example, in Figures 5C-5D, if movement 514-2 of gesture 514 is not exactly vertically downward, translation 518 of user interface object 506 is still downward if movement 514-2 is within a predefined range of angles (e.g., 20°, 30°, or 40° to the left or to the right) of the downward vertical.

[00172] In some embodiments, the first direction is a diagonal direction and the user interface object moves in the diagonal direction (722). If the direction of the gesture is a diagonal direction (e.g., at a 45° from the vertical or horizontal axis or within a predefined range of angles (e.g., 15°) from the 45° line), then the user interface object translates in the diagonal direction. In some embodiments, translation in the diagonal direction includes translation by a predefined number of pixels (e.g., 1 pixel for a gesture with a velocity below the first predefined threshold, 10 pixels for a gesture with a velocity greater than the first predefined threshold) horizontally and by the same predefined number of pixels vertically. For example, in Figures 5I-5J, use interface object 506 translates 530 one pixel upward and one pixel leftward in response to diagonal gesture 528 if the velocity of gesture 528 is below the first predefined threshold.

[00173] In some embodiments, prior to detecting the gesture, the device detects (724) movement of the contact across the touch sensitive display, and moves (726) the user interface object in accordance with the movement of the contact. In some embodiments, there may be coarse movement/positioning of the user interface object (e.g., by dragging the user interface object), which moves in concert with the movement of the contact, followed by precise positioning of the user interface object with the gesture. For example, in Figure 5B, contact 508 moves 513 in a direction on the touch screen 112, and in response to detecting movement 513, user interface object 506 moves 515 in accordance with movement 513; user

interface object 506 is dragged by moving contact 508. Movement of the user interface object by dragging may be coarse relative to movement using gestures as described above.

[00174] In some embodiments, the device displays (728) location indicia of the user interface object while the contact is detected on the touch-sensitive display. For example, in Figure 5B, a temporary text box (e.g., callout 512, a heads-up display) with the coordinates of user interface object 506 appears proximate to contact 508 while contact 508 is detected on the touch sensitive display. The coordinates in callout 512 change as user interface object 506 moves or translates, as shown in Figures 5D, 5F, 5H, 5J. In some embodiments, when the user interface object translates in response to a gesture, the un-affected coordinate in the location indicia is grayed out, dimmed, or otherwise de-emphasized. For example, in Figure 5D, as user interface object 506 translates 518 downward, the x-coordinate in callout 512 may be dimmed.

[00175] The operations in the information processing methods described above may be implemented by running one or more functional modules in information processing apparatus such as general purpose processors or application specific chips. These modules, combinations of these modules, and/or their combination with general hardware (e.g., as described above with respect to Figures 1A, 1B and 3) are all included within the scope of protection of the invention.

[00176] The operations described above with reference to Figures 6A-6B and 7A-7B may be implemented by components depicted in Figures 1A-1C. For example, detection operations 604, 608, and translation operation 610 may be implemented by event sorter 170, event recognizer 180, and event handler 190. Event monitor 171 in event sorter 170 detects a contact on touch-sensitive display 112, and event dispatcher module 174 delivers the event information to application 136-1. A respective event recognizer 180 of application 136-1 compares the event information to respective event definitions 186, and determines whether a first contact at a first location on the touch-sensitive surface corresponds to a predefined event or sub-event, such as selection of an object on a user interface. When a respective predefined event or sub-event is detected, event recognizer 180 activates an event handler 180 associated with the detection of the event or sub-event. Event handler 180 may utilize or call data updater 176 or object updater 177 to update the application internal state 192. In some embodiments, event handler 180 accesses a respective GUI updater 178 to update what is displayed by the application. Similarly, it would be clear to a person having ordinary skill

in the art how other processes can be implemented based on the components depicted in Figures 1A-1C.

[00177] In accordance with some embodiments, Figure 8 shows a functional block diagram of an electronic device 800 configured in accordance with the principles of the invention as described above. The functional blocks of the device may be implemented by hardware, software, or a combination of hardware and software to carry out the principles of the invention. It is understood by persons of skill in the art that the functional blocks described in Figure 8 may be combined or separated into sub-blocks to implement the principles of the invention as described above. Therefore, the description herein may support any possible combination or separation or further definition of the functional blocks described herein.

[00178] As shown in Figure 8, a computing device 800 includes: a touch-sensitive display unit 802 configured to display user interface objects and receive user contacts; and a processing unit 806 coupled to the touch-sensitive display unit 802. In some embodiments, the processing unit 806 includes a detecting unit 808, a translating unit 810, a moving unit 812, and a display enabling unit 814.

[00179] The processing unit 806 is configured to: enable display of a user interface object (e.g., with the display enabling unit 814); detect a contact (e.g., on the touch-sensitive display unit 802) on the user interface object (e.g., with the detecting unit 808); while continuing to detect the contact on the user interface object: detect an M-finger gesture, distinct from the contact, in a first direction on the touch-sensitive display unit 802, where M is an integer (e.g., with the detecting unit 808); and, in response to detecting the M-finger gesture, translate the user interface object a predefined number of pixels in a direction in accordance with the first direction (e.g., with the translating unit 810).

[00180] In some embodiments, the predefined number of pixels is one pixel when M is one.

[00181] In some embodiments, the predefined number of pixels is ten pixels when M is two.

[00182] In some embodiments, the first direction is a rightward direction and the user interface object moves to the right; the first direction is a leftward direction and the user interface object moves to the left; the first direction is an upward direction and the user

interface object moves up; or the first direction is a downward direction and the user interface object moves down.

[00183] In some embodiments, the first direction is a diagonal direction and the user interface object moves in the diagonal direction.

[00184] In some embodiments, the processing unit 806 is configured to: prior to detecting the M-finger gesture, detect movement of the contact across the touch-sensitive display unit 802 (e.g., with the detecting unit 808); and move the user interface object in accordance with the movement of the contact (e.g., with the moving unit 812).

[00185] In some embodiments, the processing unit 806 is configured to enable display of location indicia of the user interface object while the contact is detected on the touch-sensitive display unit 802 (e.g., with the display enabling unit 814).

[00186] In accordance with some embodiments, Figure 9 shows a functional block diagram of an electronic device 900 configured in accordance with the principles of the invention as described above. The functional blocks of the device may be implemented by hardware, software, or a combination of hardware and software to carry out the principles of the invention. It is understood by persons of skill in the art that the functional blocks described in Figure 9 may be combined or separated into sub-blocks to implement the principles of the invention as described above. Therefore, the description herein may support any possible combination or separation or further definition of the functional blocks described herein.

[00187] As shown in Figure 9, a computing device 900 includes a touch-sensitive display unit 902 configured to display user interface objects and receive user contacts; and a processing unit 906 coupled to the touch-sensitive display unit 902. In some embodiments, the processing unit 906 includes a detecting unit 908, a translating unit 910, a moving unit 912, and a display enabling unit 914.

[00188] The processing unit 906 is configured to: enable display of a user interface object on the touch-sensitive display unit 902; detect a contact (e.g., on the touch-sensitive display unit 902) on the user interface object (e.g., with the detecting unit 908); while continuing to detect the contact on the user interface object: detect a gesture, distinct from the contact, in a first direction with a first velocity on the touch-sensitive display unit 902 (e.g., with the detecting unit 908); and, in response to detecting the gesture: when the first velocity is below a predefined threshold, translate the user interface object a first predefined number

of pixels in a direction in accordance with the first direction (e.g., with the translating unit 910); and, when the first velocity is above the predefined threshold, translate the user interface object a second predefined number of pixels, distinct from the first predefined number of pixels, in a direction in accordance with the first direction (e.g., with the translating unit 910).

[00189] In some embodiments, the first predefined number of pixels is one pixel.

[00190] In some embodiments, the second predefined number of pixels is ten pixels.

[00191] In some embodiments, the first direction is a rightward direction and the user interface object moves to the right; the first direction is a leftward direction and the user interface object moves to the left; the first direction is an upward direction and the user interface object moves up; or the first direction is a downward direction and the user interface object moves down.

[00192] In some embodiments, the first direction is a diagonal direction and the user interface object moves in the diagonal direction.

[00193] In some embodiments, the processing unit 906 is configured to: prior to detecting the gesture, detect movement of the contact across the touch-sensitive display unit 902 (e.g., with the detecting unit 908); and move the user interface object in accordance with the movement of the contact (e.g., with the moving unit 912).

[00194] In some embodiments, the processing unit 906 is configured to enable display of location indicia of the user interface object while the contact is detected on the touch-sensitive display unit 902 (e.g. with the display enabling unit 914).

[00195] The foregoing description, for purpose of explanation, has been described with reference to specific embodiments. However, the illustrative discussions above are not intended to be exhaustive or to limit the invention to the precise forms disclosed. Many modifications and variations are possible in view of the above teachings. The embodiments were chosen and described in order to best explain the principles of the invention and its practical applications, to thereby enable others skilled in the art to best utilize the invention and various embodiments with various modifications as are suited to the particular use contemplated.

What is claimed is:

1. A computing device, comprising:
a touch-sensitive display;
one or more processors;
memory; and
one or more programs, wherein the one or more programs are stored in the memory and configured to be executed by the one or more processors, the one or more programs including instructions for:
displaying a user interface object on the touch-sensitive display;
detecting a contact on the user interface object;
while continuing to detect the contact on the user interface object:
detecting an M-finger gesture, distinct from the contact, in a first direction on the touch-sensitive display, where M is an integer; and,
in response to detecting the M-finger gesture, translating the user interface object a predefined number of pixels in a direction in accordance with the first direction.
2. The device of claim 1, wherein the predefined number of pixels is one pixel when M is one.
3. The device of any of claims 1-2, wherein the predefined number of pixels is ten pixels when M is two.
4. The device of any of claims 1-3, wherein:
the first direction is a rightward direction and the user interface object moves to the right;
the first direction is a leftward direction and the user interface object moves to the left;
the first direction is an upward direction and the user interface object moves up; or
the first direction is a downward direction and the user interface object moves down.
5. The device of any of claims 1-3, wherein the first direction is a diagonal direction and the user interface object moves in the diagonal direction.
6. The device of any of claims 1-5, including instructions for:

prior to detecting the M-finger gesture, detecting movement of the contact across the touch sensitive display; and,
moving the user interface object in accordance with the movement of the contact.

7. The device of any of claims 1-6, including instructions for:
displaying location indicia of the user interface object while the contact is detected on the touch-sensitive display.

8. A computing device, comprising:
a touch-sensitive display;
one or more processors;
memory; and
one or more programs, wherein the one or more programs are stored in the memory and configured to be executed by the one or more processors, the one or more programs including instructions for:

displaying a user interface object on the touch-sensitive display;

detecting a contact on the user interface object;

while continuing to detect the contact on the user interface object:

detecting a gesture, distinct from the contact, in a first direction with a first velocity on the touch-sensitive display; and,

in response to detecting the gesture:

when the first velocity is below a predefined threshold, translating the user interface object a first predefined number of pixels in a direction in accordance with the first direction; and,

when the first velocity is above the predefined threshold, translating the user interface object a second predefined number of pixels, distinct from the first predefined number of pixels, in a direction in accordance with the first direction.

9. The device of claim 8, wherein the first predefined number of pixels is one pixel.

10. The device of any of claims 8-9, wherein the second predefined number of pixels is ten pixels.

11. The device of any of claims 8-10, wherein:
the first direction is a rightward direction and the user interface object moves to the right;

the first direction is a leftward direction and the user interface object moves to the left;

the first direction is an upward direction and the user interface object moves up; or

the first direction is a downward direction and the user interface object moves down.

12. The device of any of claims 8-10, wherein the first direction is a diagonal direction and the user interface object moves in the diagonal direction.

13. The device of any of claims 8-12, including:

prior to detecting the gesture, detecting movement of the contact across the touch sensitive display; and,

moving the user interface object in accordance with the movement of the contact.

14. The device of any of claims 8-13, including:

displaying location indicia of the user interface object while the contact is detected on the touch-sensitive display.

15. A method, comprising:

at a computing device with a touch-sensitive display:

displaying a user interface object on the touch-sensitive display;

detecting a contact on the user interface object;

while continuing to detect the contact on the user interface object:

detecting an M-finger gesture, distinct from the contact, in a first direction on the touch-sensitive display, where M is an integer; and,

in response to detecting the M-finger gesture, translating the user interface object a predefined number of pixels in a direction in accordance with the first direction.

16. The method of claim 15, wherein the predefined number of pixels is one pixel when M is one.

17. The method of any of claims 15-16, wherein the predefined number of pixels is ten pixels when M is two.

18. The method of any of claims 15-17, wherein:

the first direction is a rightward direction and the user interface object moves to the right;

the first direction is a leftward direction and the user interface object moves to the left;

the first direction is an upward direction and the user interface object moves up; or

the first direction is a downward direction and the user interface object moves down.

19. The method of any of claims 15-17, wherein the first direction is a diagonal direction and the user interface object moves in the diagonal direction.

20. The method of any of claims 15-19, including:

prior to detecting the M-finger gesture, detecting movement of the contact across the touch sensitive display; and,

moving the user interface object in accordance with the movement of the contact.

21. The method of any of claims 15-20, including:

displaying location indicia of the user interface object while the contact is detected on the touch-sensitive display.

22. A method, comprising:

at a computing device with a touch-sensitive display:

displaying a user interface object on the touch-sensitive display;

detecting a contact on the user interface object;

while continuing to detect the contact on the user interface object:

detecting a gesture, distinct from the contact, in a first direction with a first velocity on the touch-sensitive display; and,

in response to detecting the gesture:

when the first velocity is below a predefined threshold, translating the user interface object a first predefined number of pixels in a direction in accordance with the first direction; and,

when the first velocity is above the predefined threshold, translating the user interface object a second predefined number of pixels, distinct from the first predefined number of pixels, in a direction in accordance with the first direction.

23. The method of claim 22, wherein the first predefined number of pixels is one pixel.

24. The method of any of claims 22-23, wherein the second predefined number of pixels is ten pixels.

25. The method of any of claims 22-24, wherein:
the first direction is a rightward direction and the user interface object moves to the right;
the first direction is a leftward direction and the user interface object moves to the left;
the first direction is an upward direction and the user interface object moves up; or
the first direction is a downward direction and the user interface object moves down.
26. The method of any of claims 22-24, wherein the first direction is a diagonal direction and the user interface object moves in the diagonal direction.
27. The method of any of claims 22-26, including:
prior to detecting the gesture, detecting movement of the contact across the touch sensitive display; and,
moving the user interface object in accordance with the movement of the contact.
28. The method of any of claims 22-27, including:
displaying location indicia of the user interface object while the contact is detected on the touch-sensitive display.
29. A graphical user interface on a computing device with a touch-sensitive display, a memory, and one or more processors to execute one or more programs stored in the memory, the graphical user interface comprising:
a user interface object on the touch-sensitive display;
wherein:
a contact on the user interface object is detected;
while the contact on the user interface object continues to be detected:
an M-finger gesture, distinct from the contact, in a first direction, is detected on the touch-sensitive display, where M is an integer; and,
in response to detection of the M-finger gesture, the user interface object is translated a predefined number of pixels in a direction in accordance with the first direction.
30. A graphical user interface on a computing device with a touch-sensitive display, a memory, and one or more processors to execute one or more programs stored in the memory, the graphical user interface comprising:
a user interface object on the touch-sensitive display;

wherein:

a contact on the user interface object is detected;

while the contact on the user interface object continues to be detected:

a gesture, distinct from the contact, in a first direction with a first velocity, is detected on the touch-sensitive display; and,

in response to detection of the gesture:

when the first velocity is below a predefined threshold, the user interface object is translated a first predefined number of pixels in a direction in accordance with the first direction; and,

when the first velocity is above the predefined threshold, the user interface object is translated a second predefined number of pixels, distinct from the first predefined number of pixels, in a direction in accordance with the first direction.

31. A computer readable storage medium storing one or more programs, the one or more programs comprising instructions, which when executed by a computing device with a touch-sensitive display, cause the device to:

display a user interface object on the touch-sensitive display;

detect a contact on the user interface object;

while continuing to detect the contact on the user interface object:

detect an M-finger gesture, distinct from the contact, in a first direction on the touch-sensitive display, where M is an integer; and,

in response to detecting the M-finger gesture, translate the user interface object a predefined number of pixels in a direction in accordance with the first direction.

32. A computer readable storage medium storing one or more programs, the one or more programs comprising instructions, which when executed by a computing device with a touch-sensitive display, cause the device to:

display a user interface object on the touch-sensitive display;

detect a contact on the user interface object;

while continuing to detect the contact on the user interface object:

detect a gesture, distinct from the contact, in a first direction with a first velocity on the touch-sensitive display; and,

in response to detecting the gesture:

when the first velocity is below a predefined threshold, translate the user interface object a first predefined number of pixels in a direction in accordance with the first direction; and,

when the first velocity is above the predefined threshold, translate the user interface object a second predefined number of pixels, distinct from the first predefined number of pixels, in a direction in accordance with the first direction.

33. A computing device, comprising:

a touch-sensitive display;

means for displaying a user interface object on the touch-sensitive display;

means for detecting a contact on the user interface object;

while continuing to detect the contact on the user interface object:

means for detecting an M-finger gesture, distinct from the contact, in a first direction on the touch-sensitive display, where M is an integer; and,

means, responsive to detecting the M-finger gesture, for translating the user interface object a predefined number of pixels in a direction in accordance with the first direction.

34. A computing device, comprising:

a touch-sensitive display;

means for displaying a user interface object on the touch-sensitive display;

means for detecting a contact on the user interface object;

while continuing to detect the contact on the user interface object:

means for detecting a gesture, distinct from the contact, in a first direction with a first velocity on the touch-sensitive display; and,

means, responsive to detecting the gesture, for:

when the first velocity is below a predefined threshold, translating the user interface object a first predefined number of pixels in a direction in accordance with the first direction; and,

when the first velocity is above the predefined threshold, translating the user interface object a second predefined number of pixels, distinct from the first predefined number of pixels, in a direction in accordance with the first direction.

35. An information processing apparatus for use in a computing device with a touch-sensitive display, comprising:

means for displaying a user interface object on the touch-sensitive display;

means for detecting a contact on the user interface object;

while continuing to detect the contact on the user interface object:

means for detecting an M-finger gesture, distinct from the contact, in a first direction on the touch-sensitive display, where M is an integer; and,

means, responsive to detecting the M-finger gesture, for translating the user interface object a predefined number of pixels in a direction in accordance with the first direction.

36. An information processing apparatus for use in a computing device with a touch-sensitive display, comprising:

means for displaying a user interface object on the touch-sensitive display;

means for detecting a contact on the user interface object;

while continuing to detect the contact on the user interface object:

means for detecting a gesture, distinct from the contact, in a first direction with a first velocity on the touch-sensitive display; and,

means, responsive to detecting the gesture, for:

when the first velocity is below a predefined threshold, translating the user interface object a first predefined number of pixels in a direction in accordance with the first direction; and,

when the first velocity is above the predefined threshold, translating the user interface object a second predefined number of pixels, distinct from the first predefined number of pixels, in a direction in accordance with the first direction.

37. A computing device, comprising:

a display;

a touch-sensitive surface;

one or more processors;

memory; and

one or more programs, wherein the one or more programs are stored in the memory and configured to be executed by the one or more processors, the one or more programs including instructions for performing any of the methods of claims 15-28.

38. A graphical user interface on a computing device with a display, a touch-sensitive surface, a memory, and one or more processors to execute one or more programs stored in the

memory, the graphical user interface comprising user interfaces displayed in accordance with any of the methods of claims 15-28.

39. A computer readable storage medium storing one or more programs, the one or more programs comprising instructions, which when executed by a computing device with a display and a touch-sensitive surface, cause the device to perform any of the methods of claims 15-28.

40. A computing device, comprising:
a display and a touch-sensitive surface; and
means for perform any of the methods of claims 15-28.

41. An information processing apparatus for use in a computing device with a display and a touch-sensitive surface, comprising:
means for performing any of the methods of claims 15-28.

42. A computing device, comprising:
a touch-sensitive display unit configured to display user interface objects and receive user contacts; and
a processing unit coupled to the touch-sensitive display unit, the processing unit configured to:

enable display of a user interface object on the touch-sensitive display unit;
detect a contact on the user interface object;
while continuing to detect the contact on the user interface object:
detect an M-finger gesture, distinct from the contact, in a first direction on the touch-sensitive display unit, where M is an integer; and,
in response to detecting the M-finger gesture, translate the user interface object a predefined number of pixels in a direction in accordance with the first direction.

43. The device of claim 42, wherein the predefined number of pixels is one pixel when M is one.

44. The device of any of claims 42-43, wherein the predefined number of pixels is ten pixels when M is two.

45. The device of any of claims 42-44, wherein:

the first direction is a rightward direction and the user interface object moves to the right;

the first direction is a leftward direction and the user interface object moves to the left;

the first direction is an upward direction and the user interface object moves up; or

the first direction is a downward direction and the user interface object moves down.

46. The device of any of claims 42-44, wherein the first direction is a diagonal direction and the user interface object moves in the diagonal direction.

47. The device of any of claims 42-46, wherein the processing unit is configured to:
prior to detecting the M-finger gesture, detect movement of the contact across the touch-sensitive display unit; and,
move the user interface object in accordance with the movement of the contact.

48. The device of any of claims 42-47, wherein the processing unit is configured to:
enable display of location indicia of the user interface object while the contact is detected on the touch-sensitive display unit.

49. A computing device, comprising:
a touch-sensitive display unit configured to display user interface objects and receive user contacts; and
a processing unit coupled to the touch-sensitive display unit, the processing unit configured to:

enable display of a user interface object on the touch-sensitive display unit;

detect a contact on the user interface object;

while continuing to detect the contact on the user interface object:

detect a gesture, distinct from the contact, in a first direction with a first velocity on the touch-sensitive display unit; and,

in response to detecting the gesture:

when the first velocity is below a predefined threshold, translate the user interface object a first predefined number of pixels in a direction in accordance with the first direction; and,

when the first velocity is above the predefined threshold, translate the user interface object a second predefined number of pixels, distinct from the first predefined number of pixels, in a direction in accordance with the first direction.

50. The device of claim 49, wherein the first predefined number of pixels is one pixel.

51. The device of any of claims 49-50, wherein the second predefined number of pixels is ten pixels.

52. The device of any of claims 49-51, wherein:

the first direction is a rightward direction and the user interface object moves to the right;

the first direction is a leftward direction and the user interface object moves to the left;

the first direction is an upward direction and the user interface object moves up; or

the first direction is a downward direction and the user interface object moves down.

53. The device of any of claims 49-51, wherein the first direction is a diagonal direction and the user interface object moves in the diagonal direction.

54. The device of any of claims 49-53, wherein the processing unit is configured to:

prior to detecting the gesture, detect movement of the contact across the touch-sensitive display unit; and,

move the user interface object in accordance with the movement of the contact.

55. The device of any of claims 49-54, wherein the processing unit is configured to:

enable display of location indicia of the user interface object while the contact is detected on the touch-sensitive display unit.

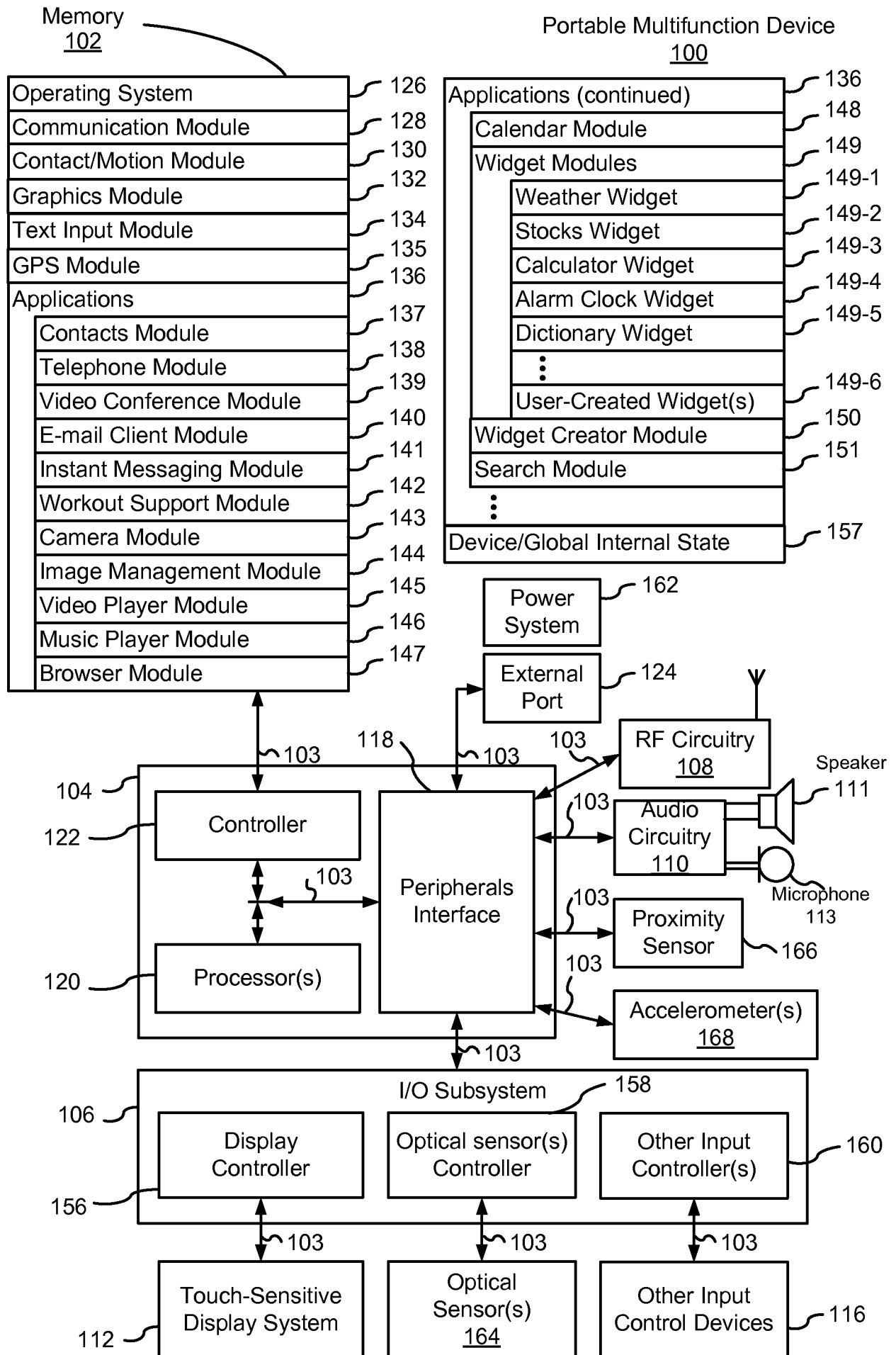


Figure 1A
1/24

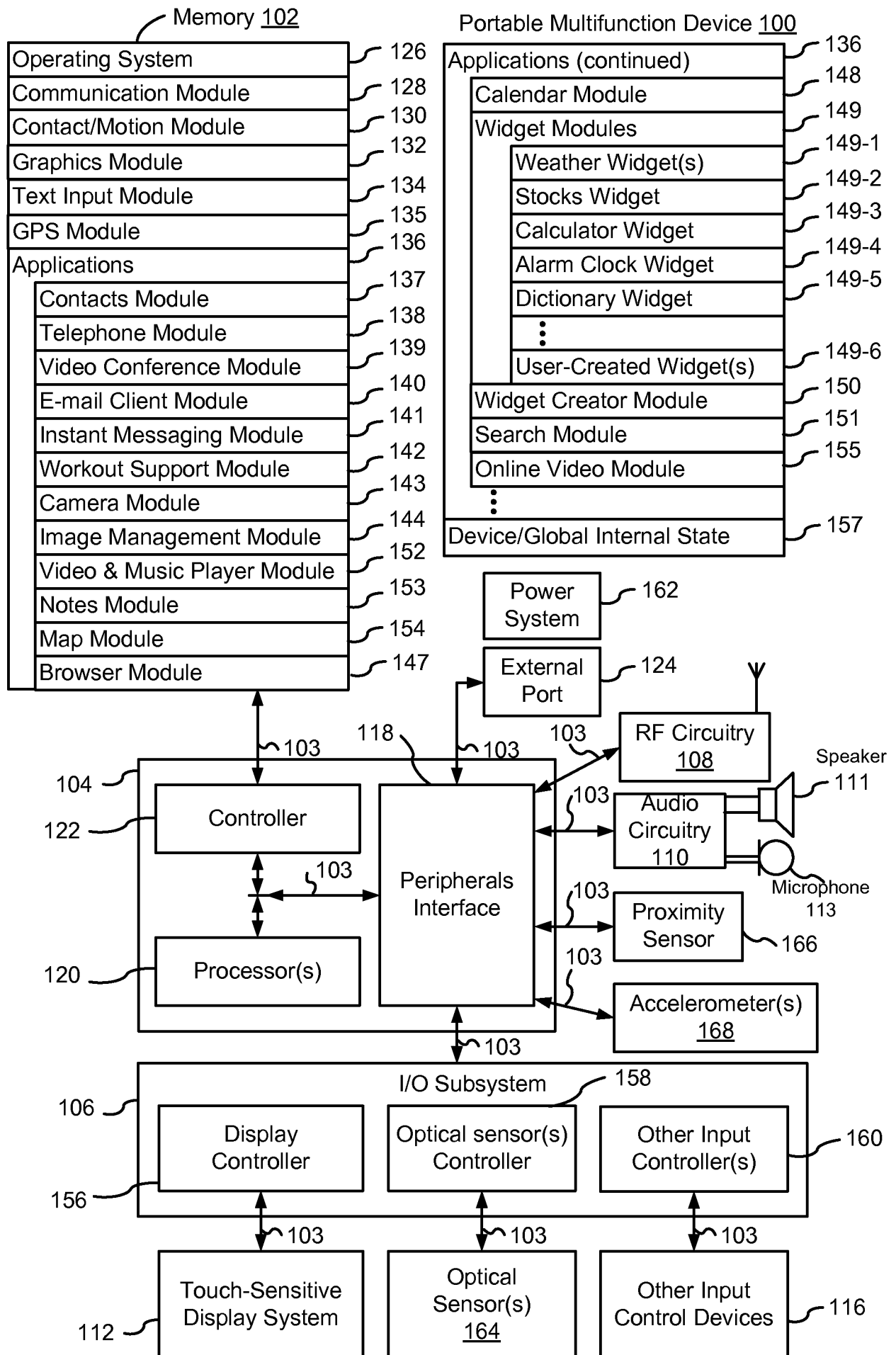


Figure 1B

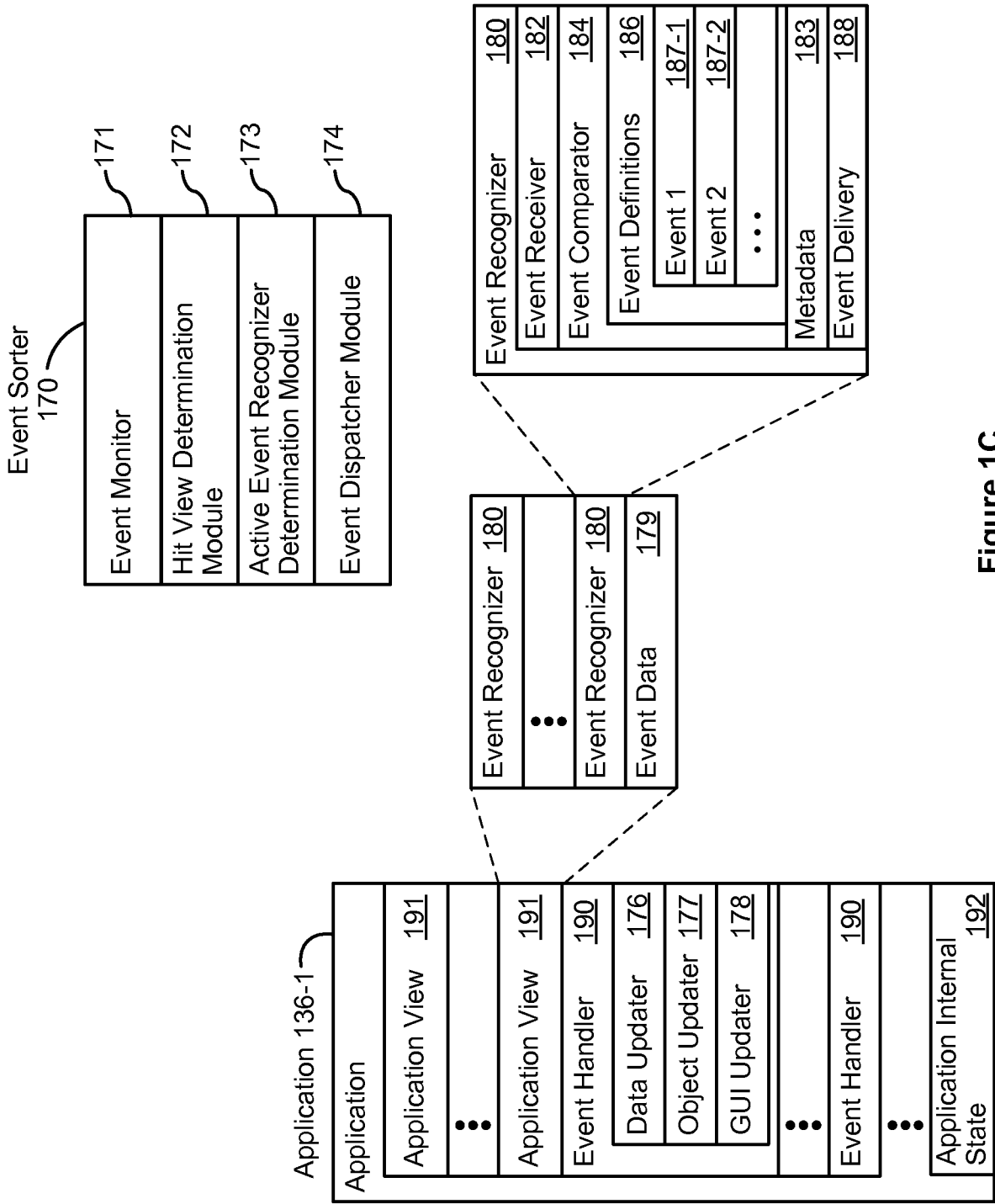


Figure 1C

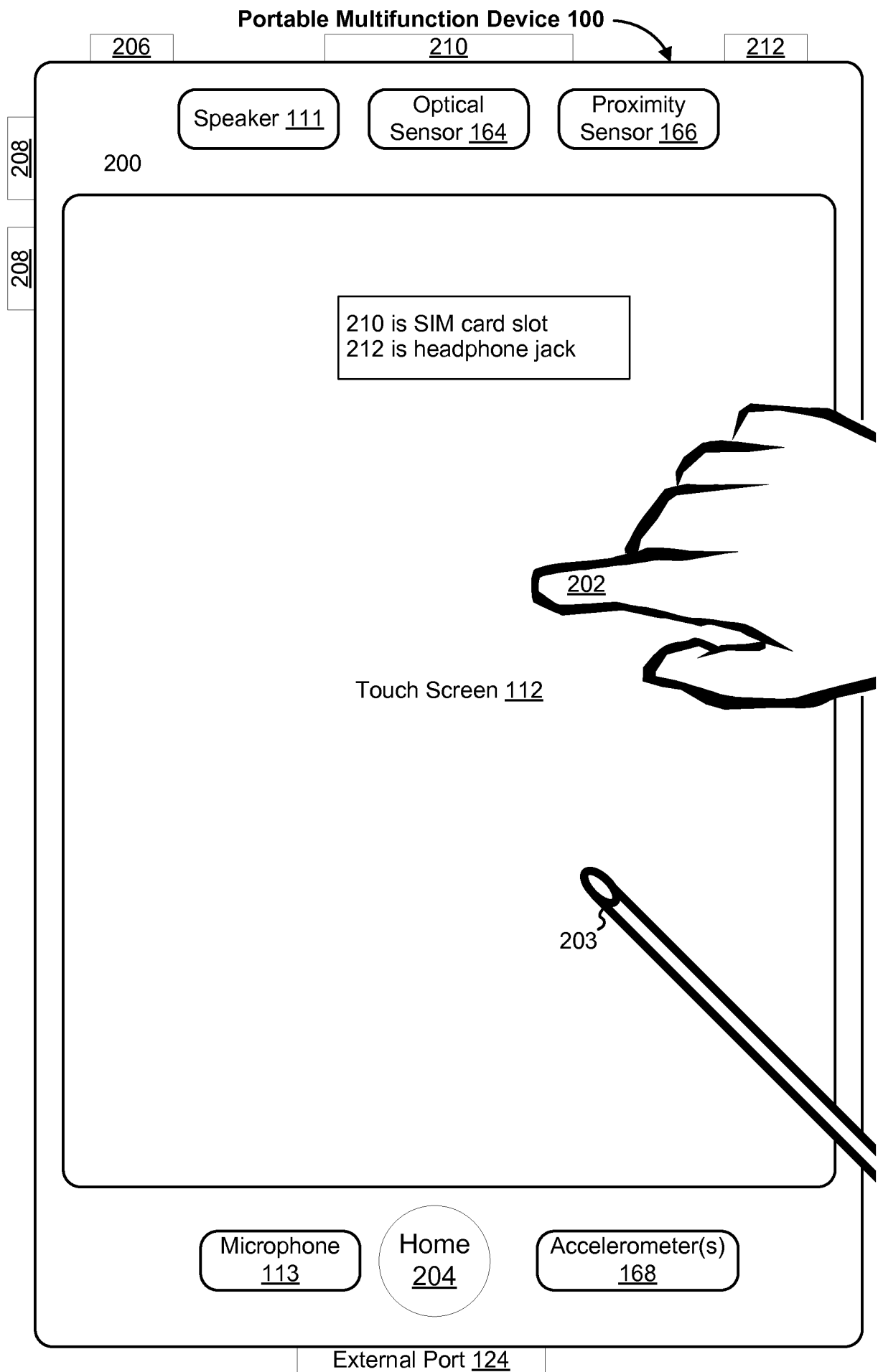


Figure 2
4/24

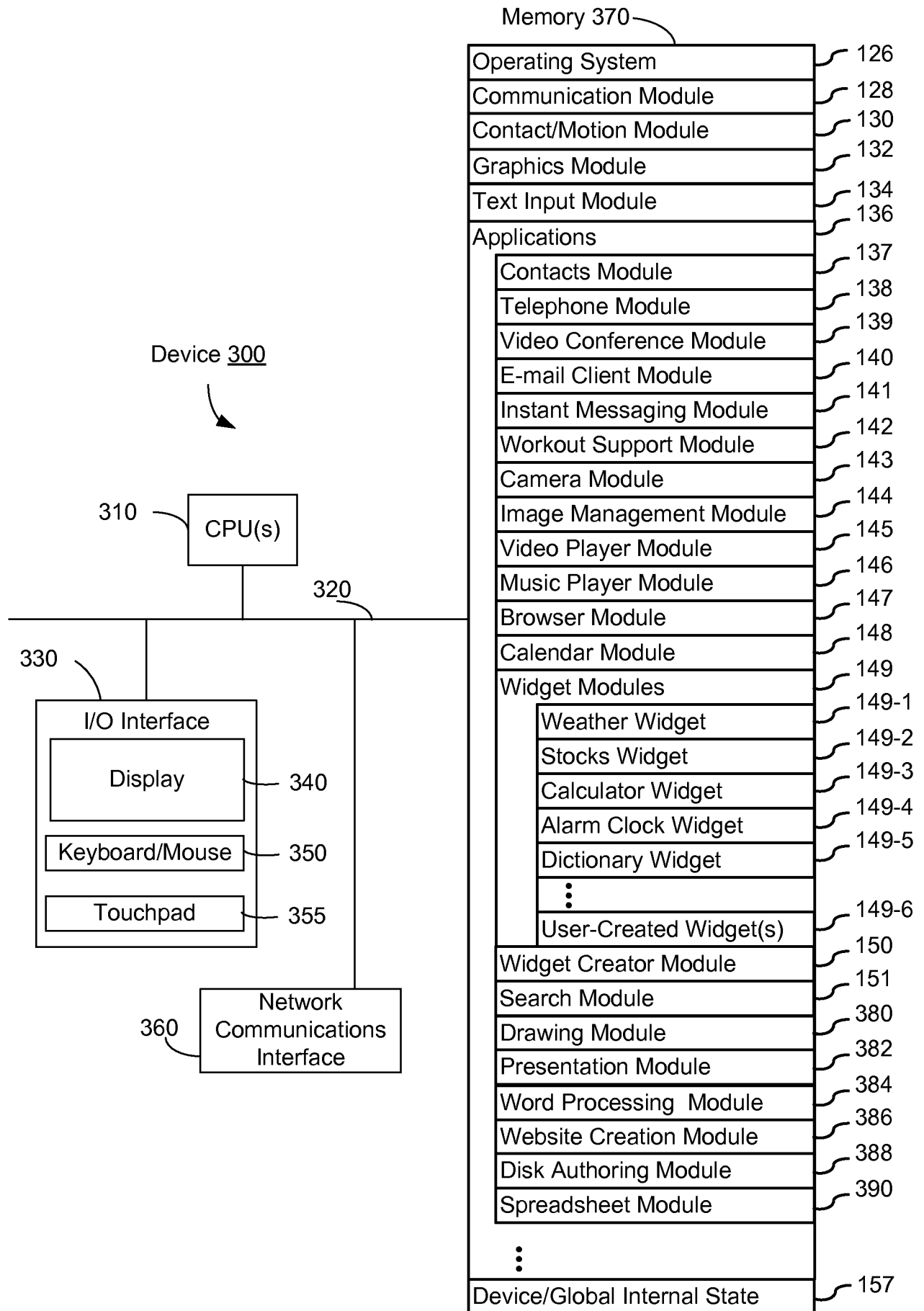
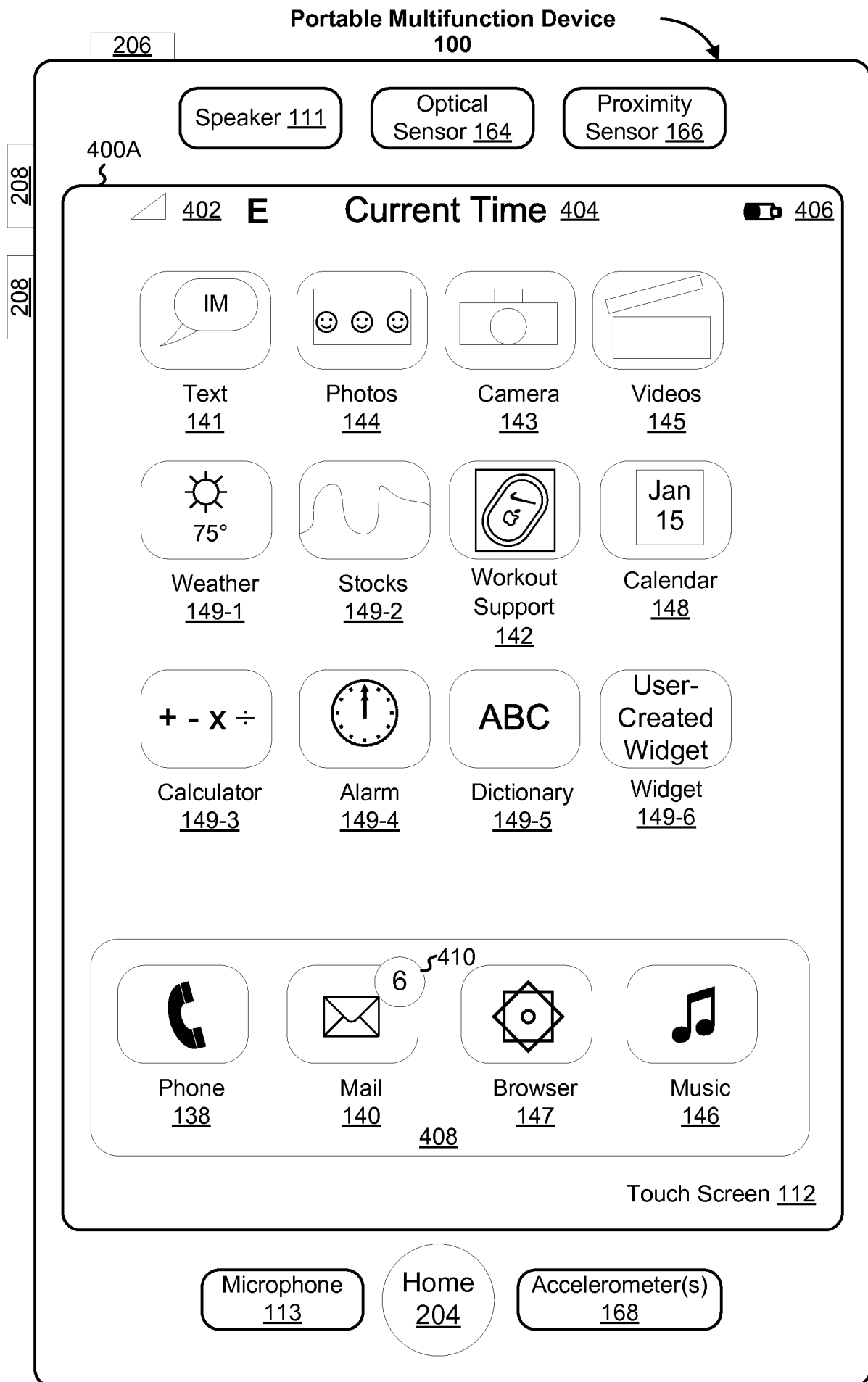


Figure 3

**Figure 4A**

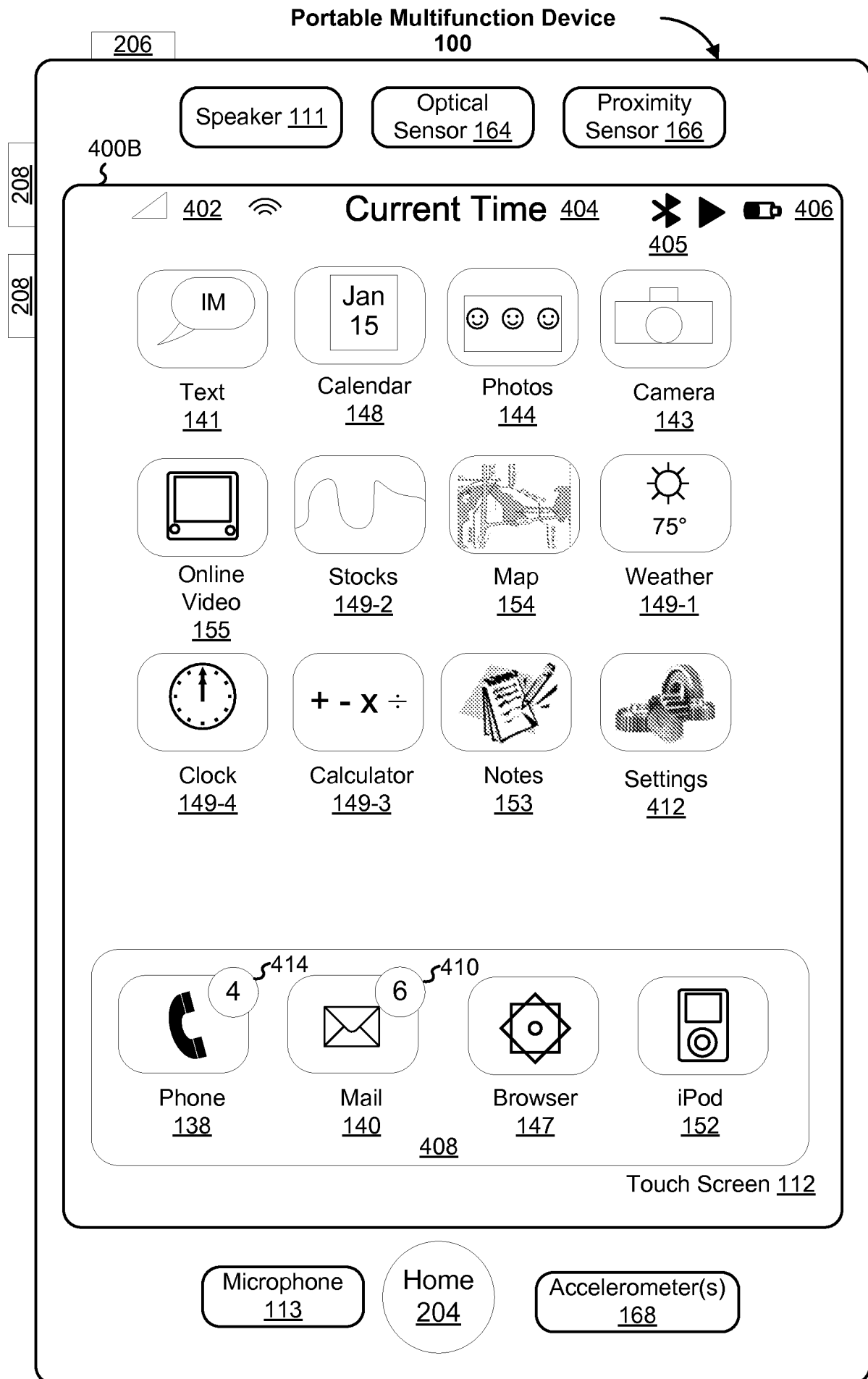


Figure 4B
7/24

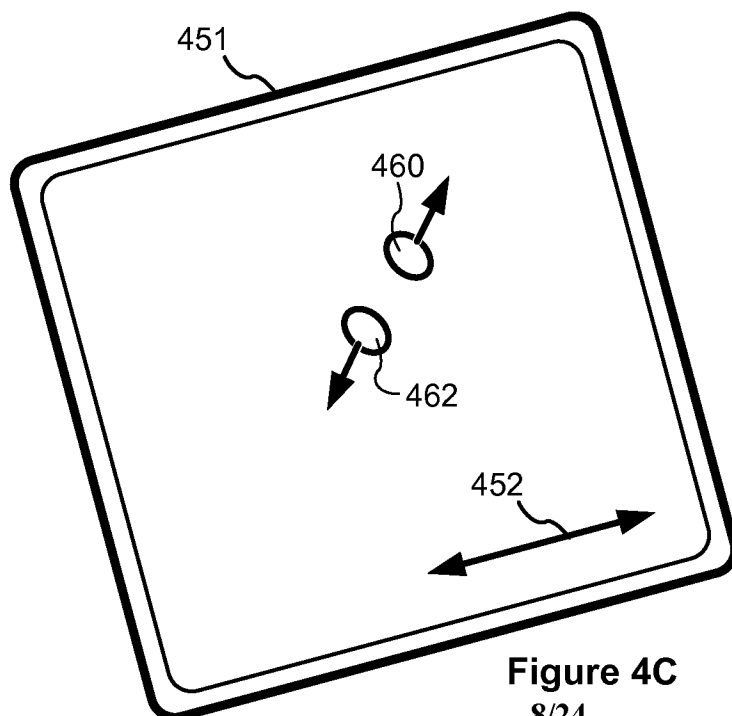
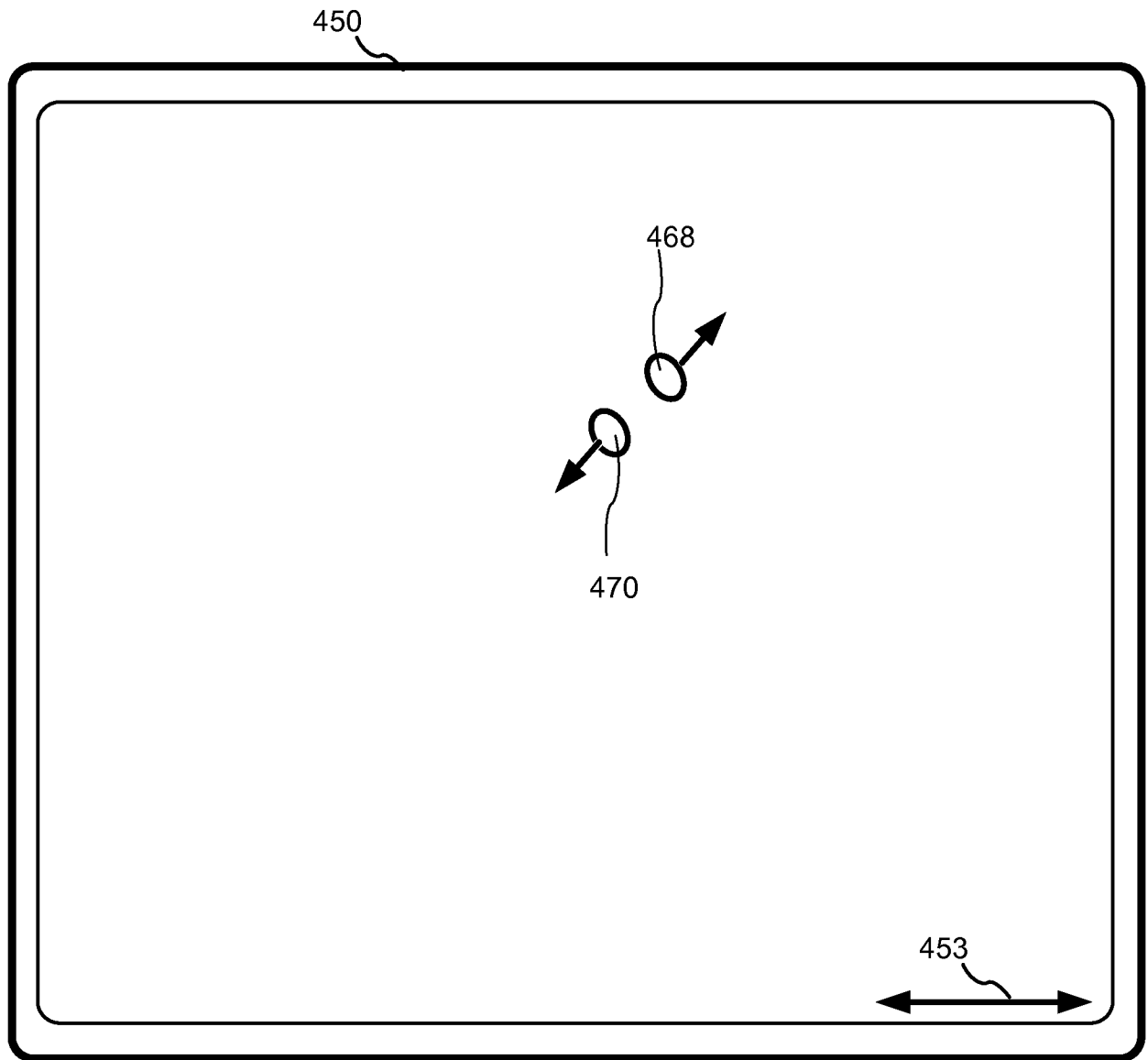
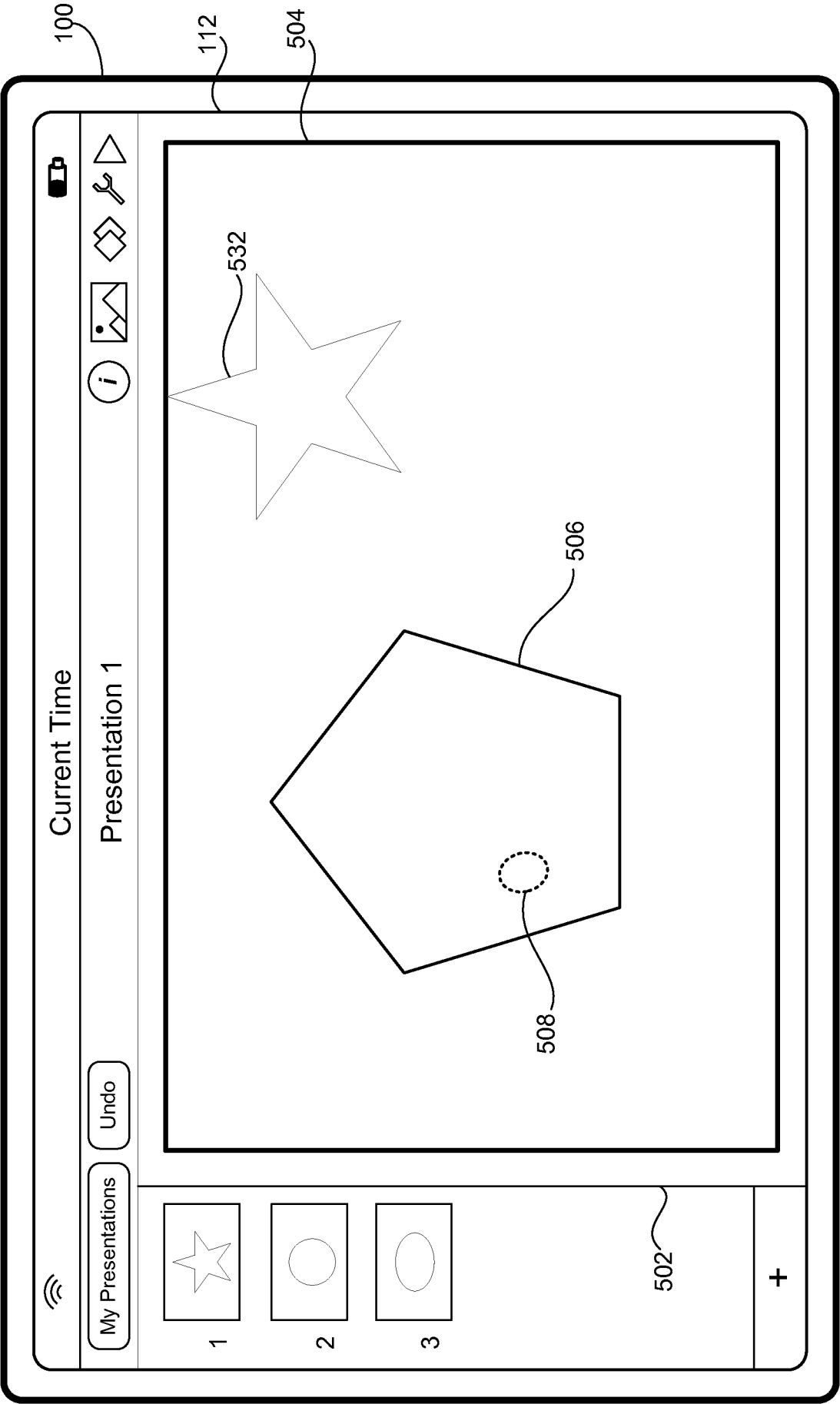
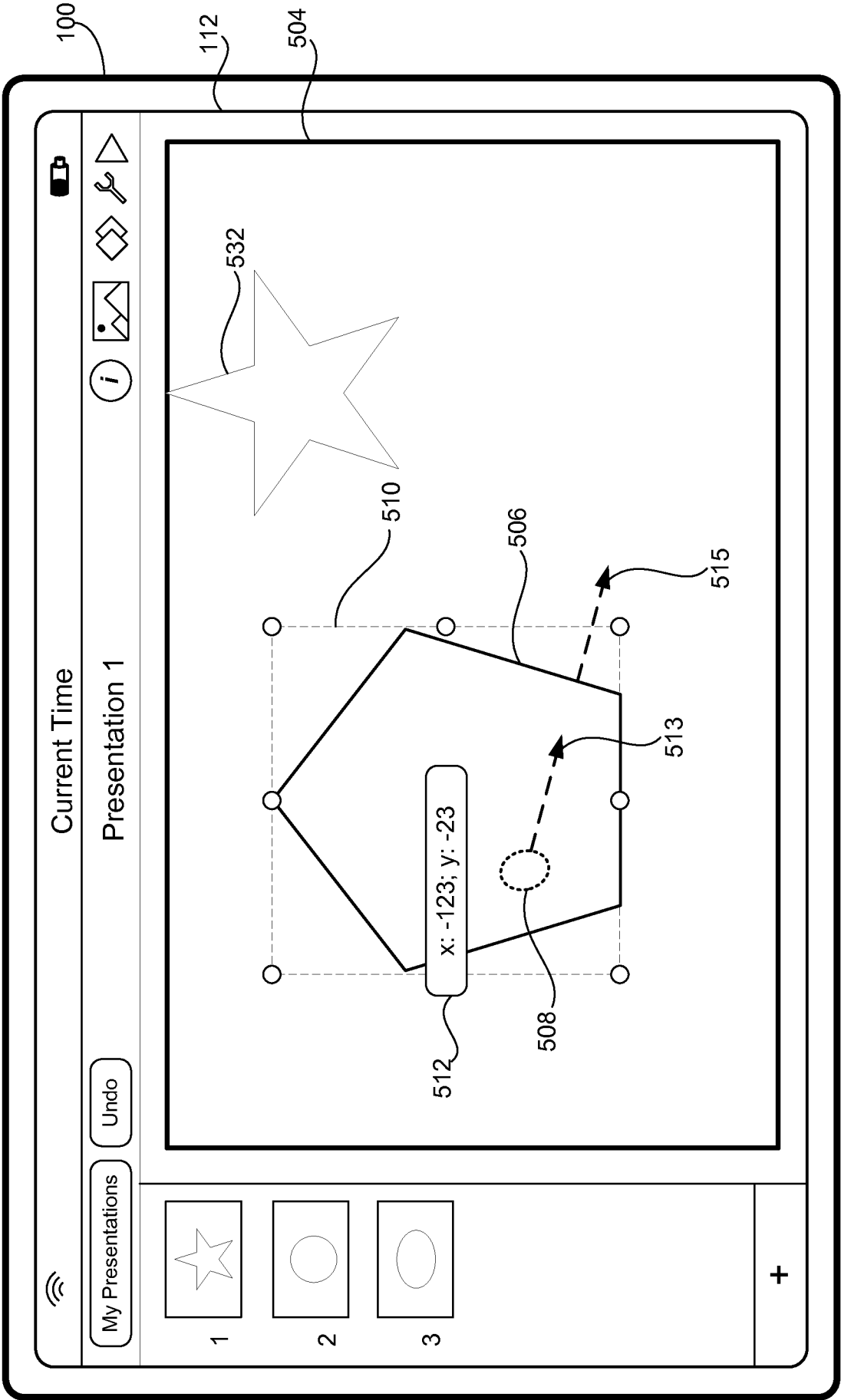


Figure 4C
8/24



500A

Figure 5A



500B

Figure 5B

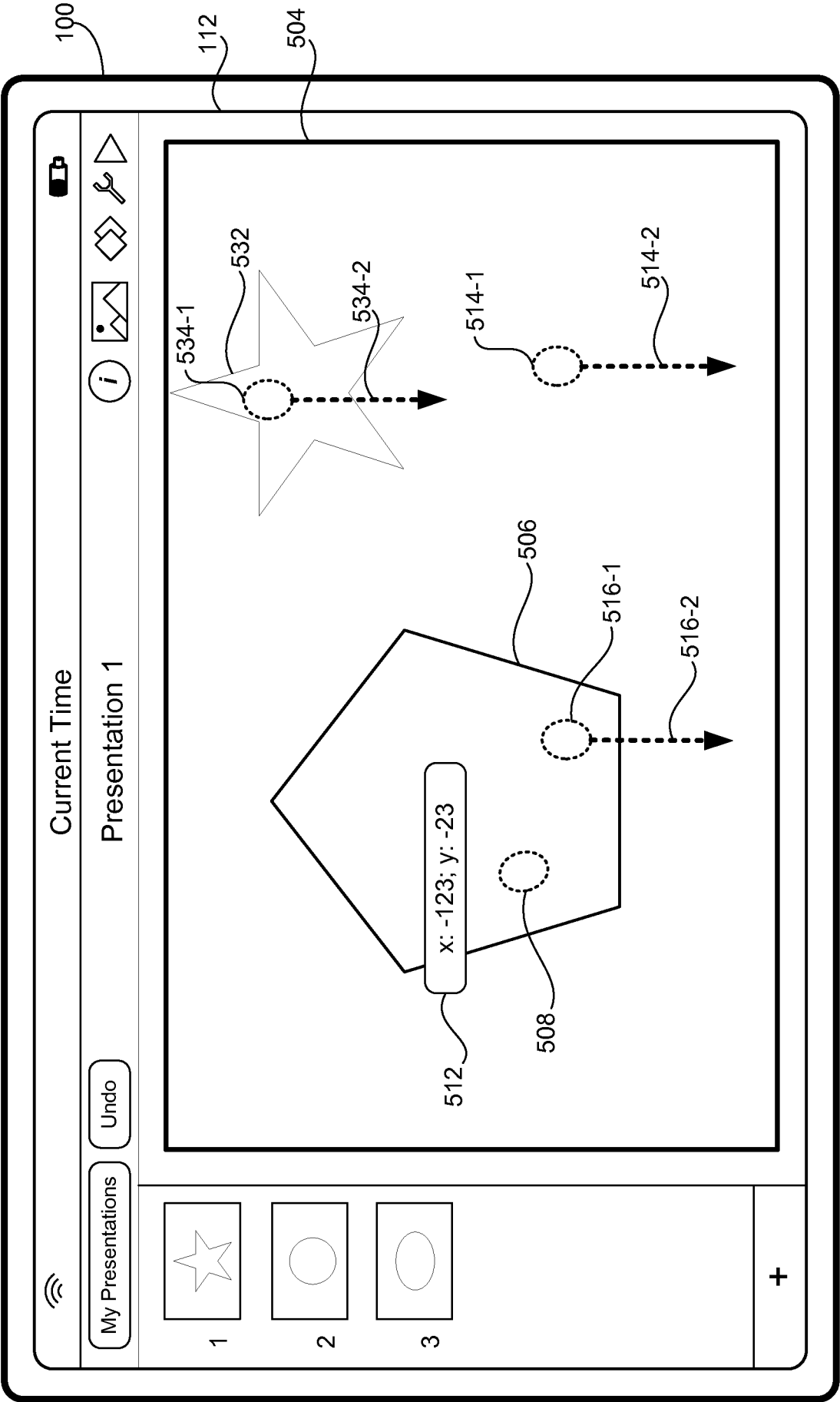
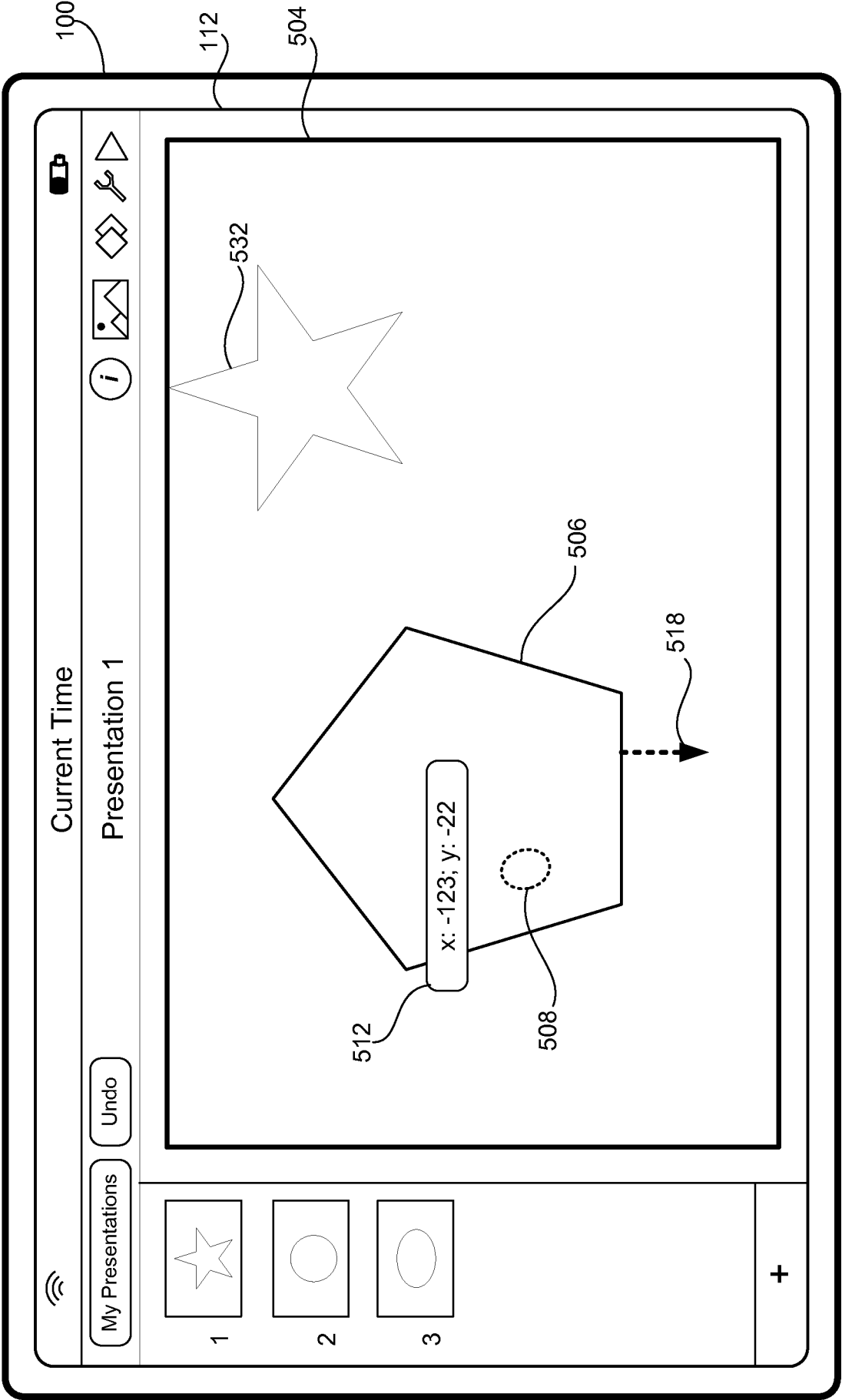


Figure 5C

500C



500D

Figure 5D

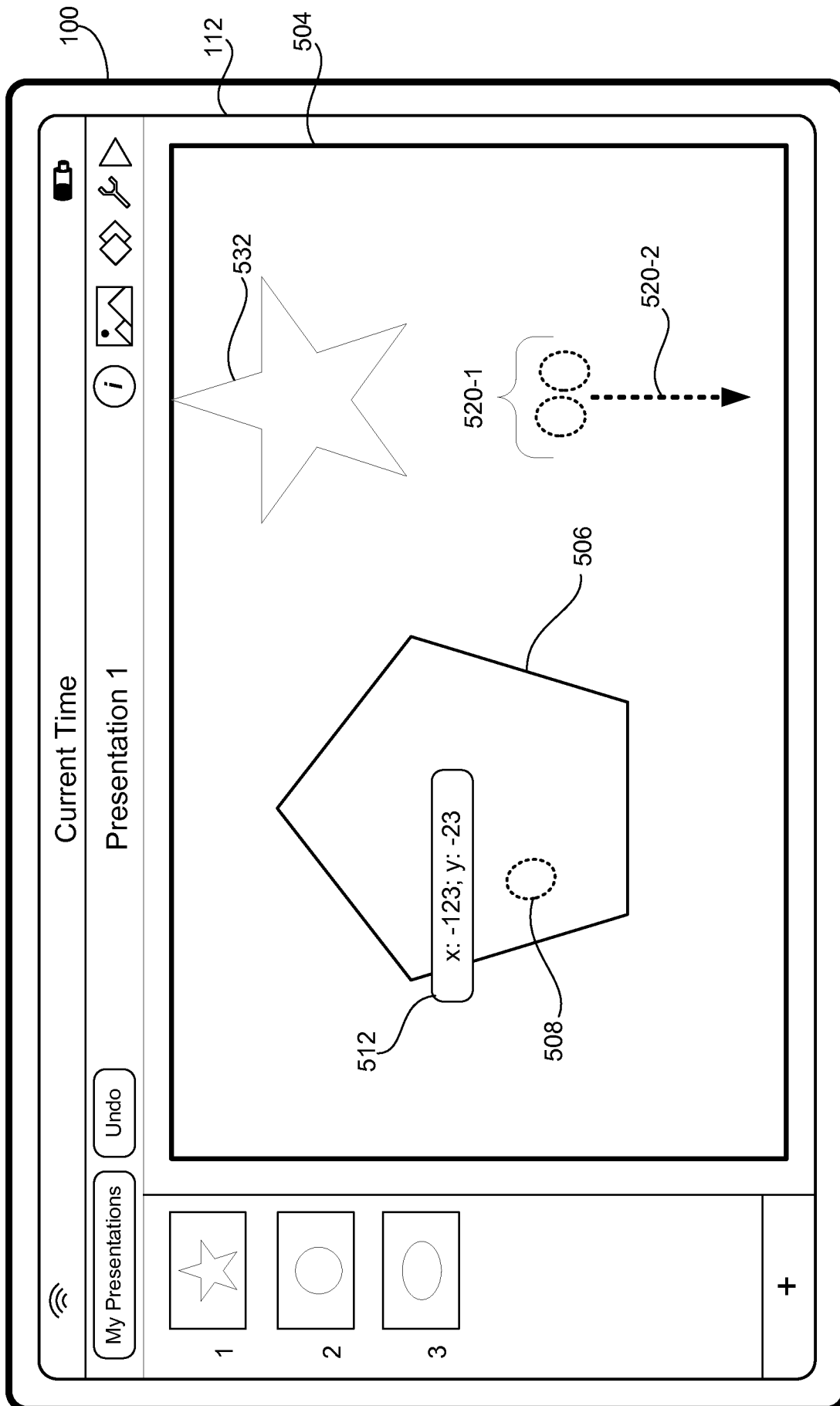


Figure 5E

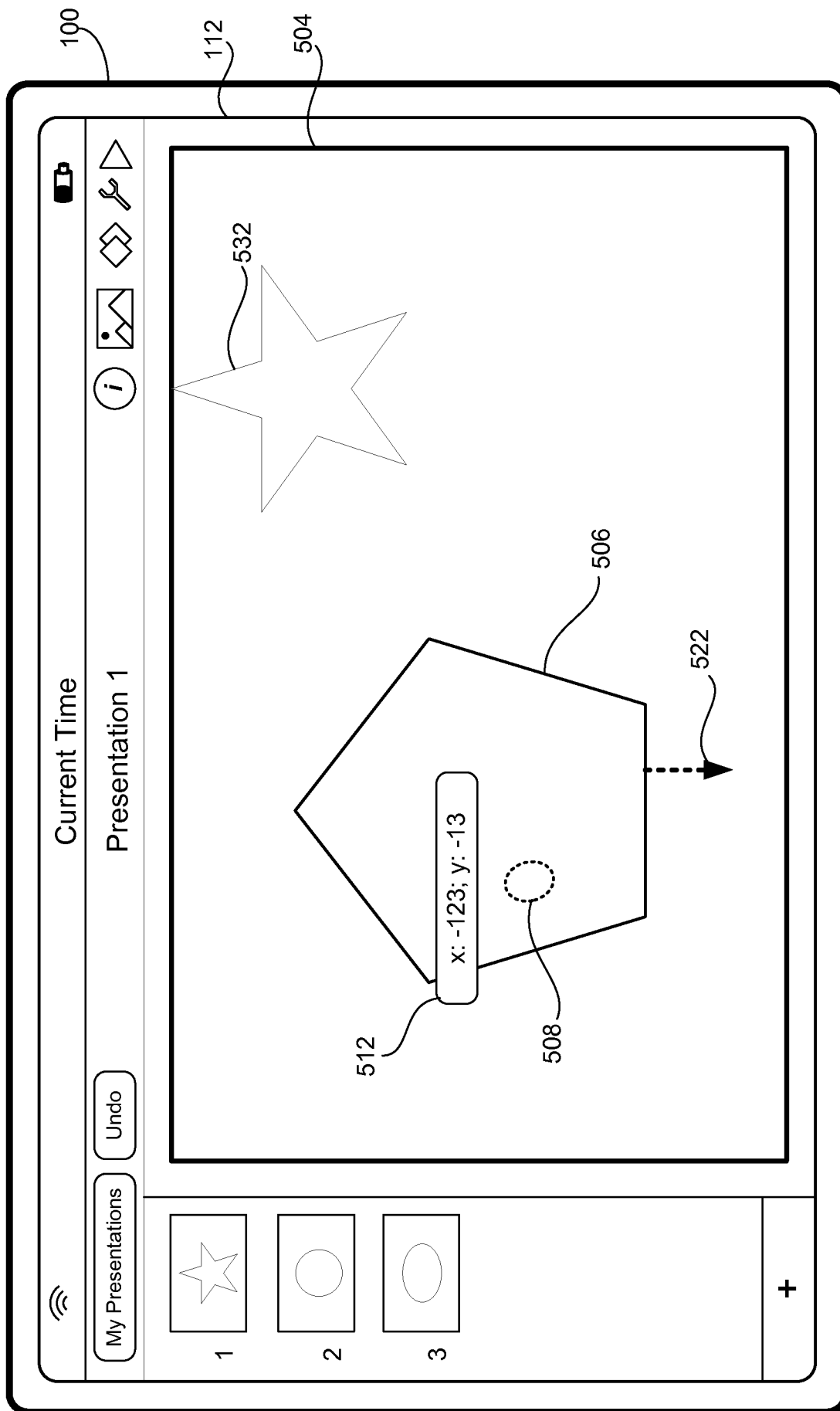
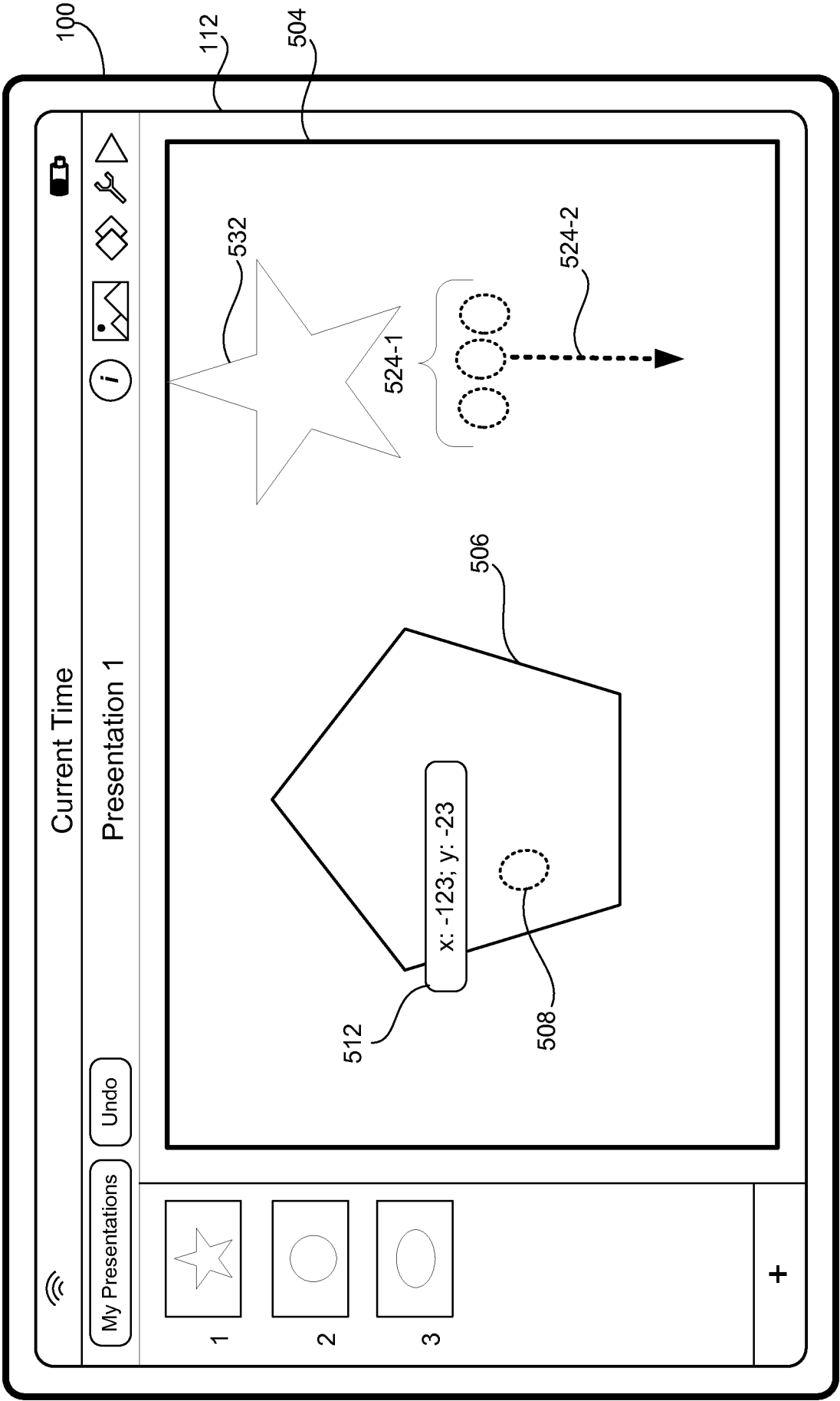
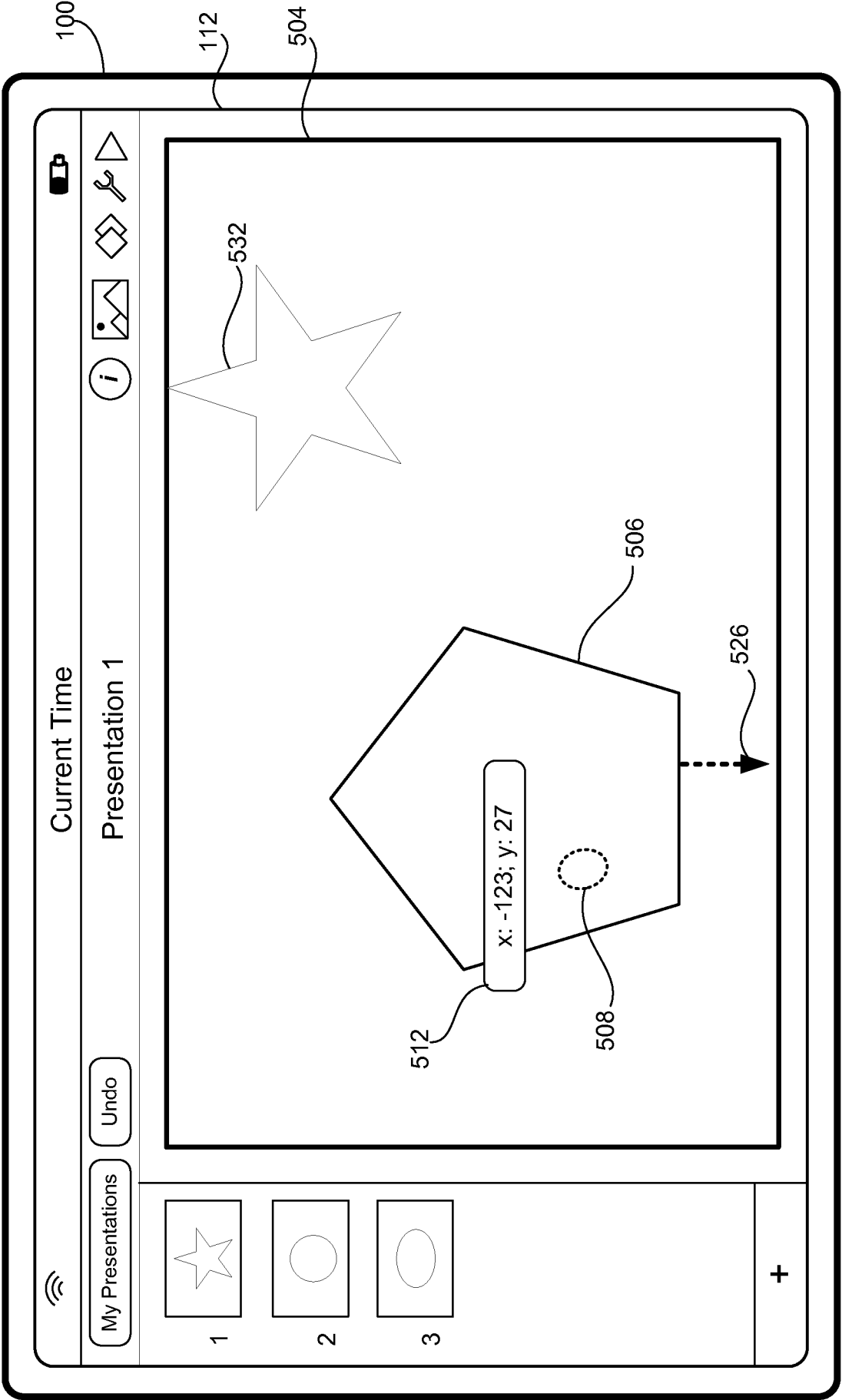


Figure 5F 500E



500G

Figure 5G



500H

Figure 5H

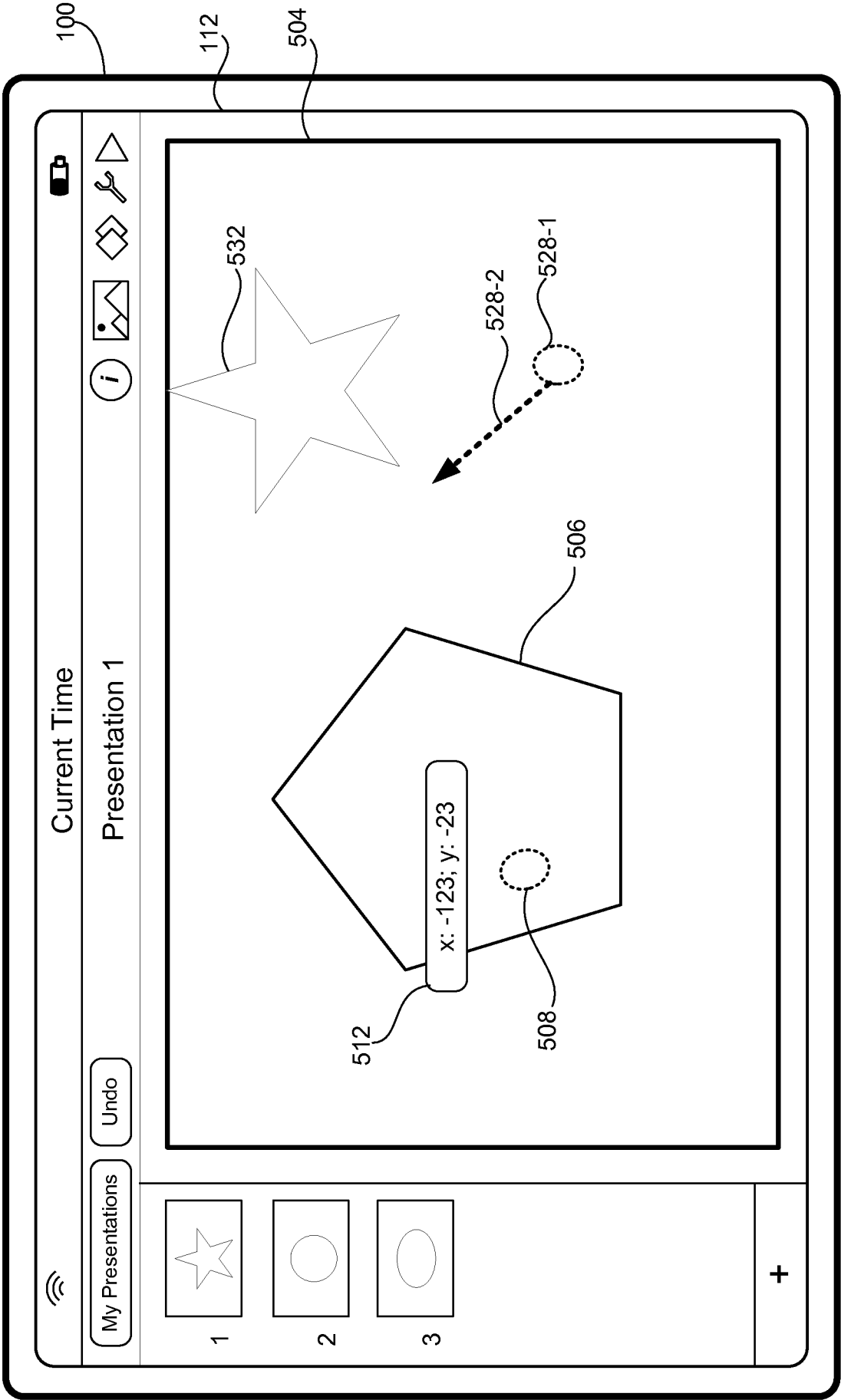
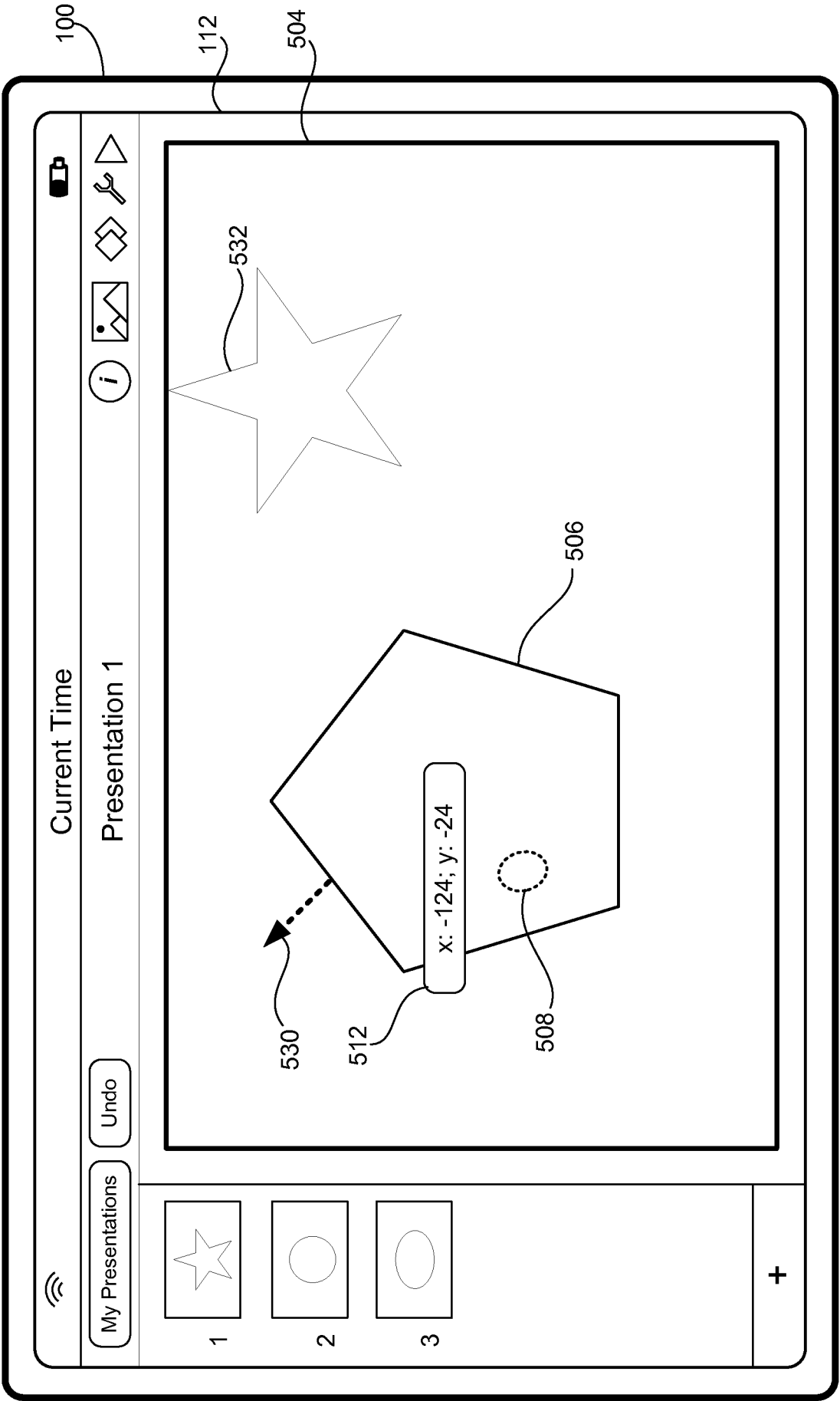


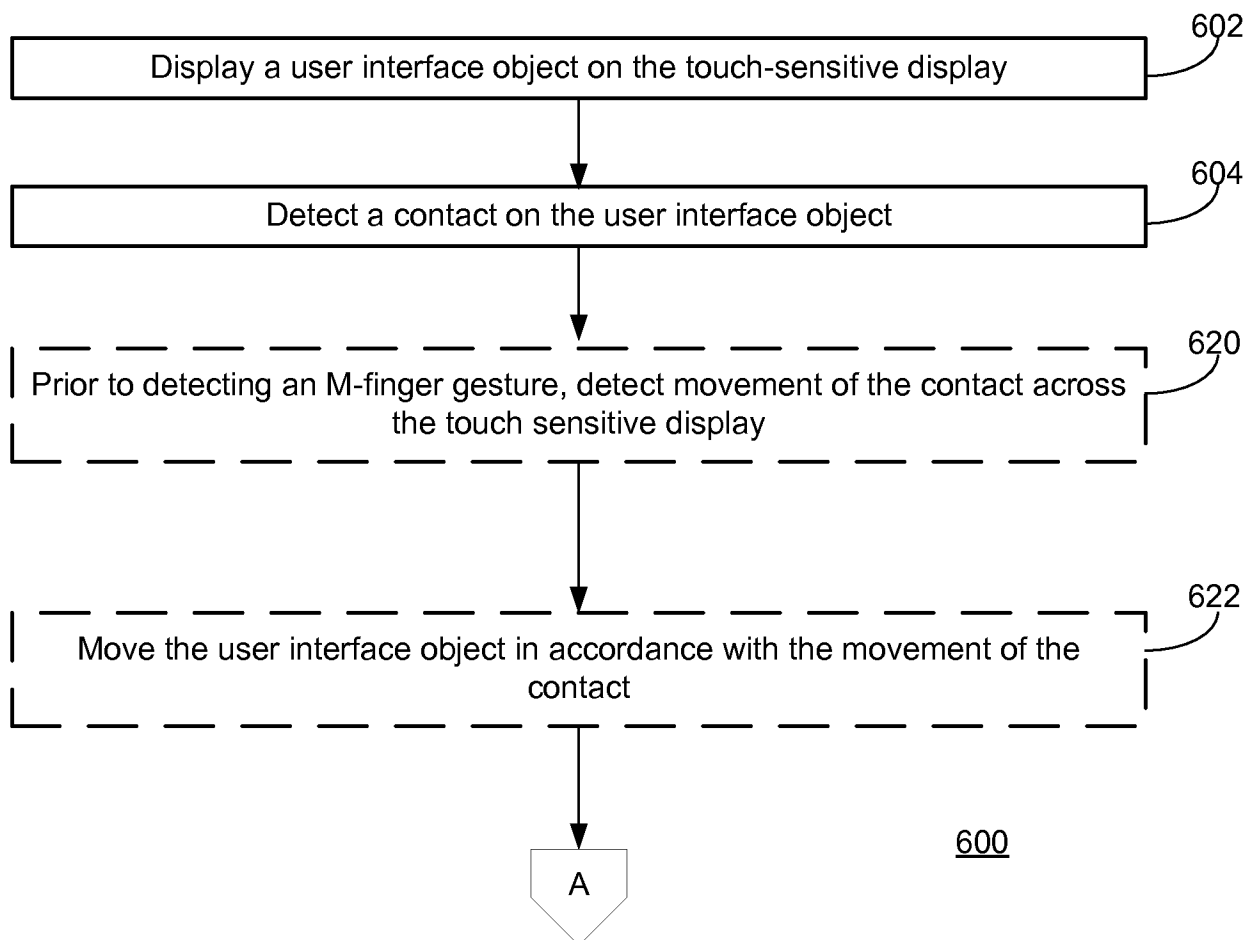
Figure 5I

500I



500J

Figure 5J

**Figure 6A**

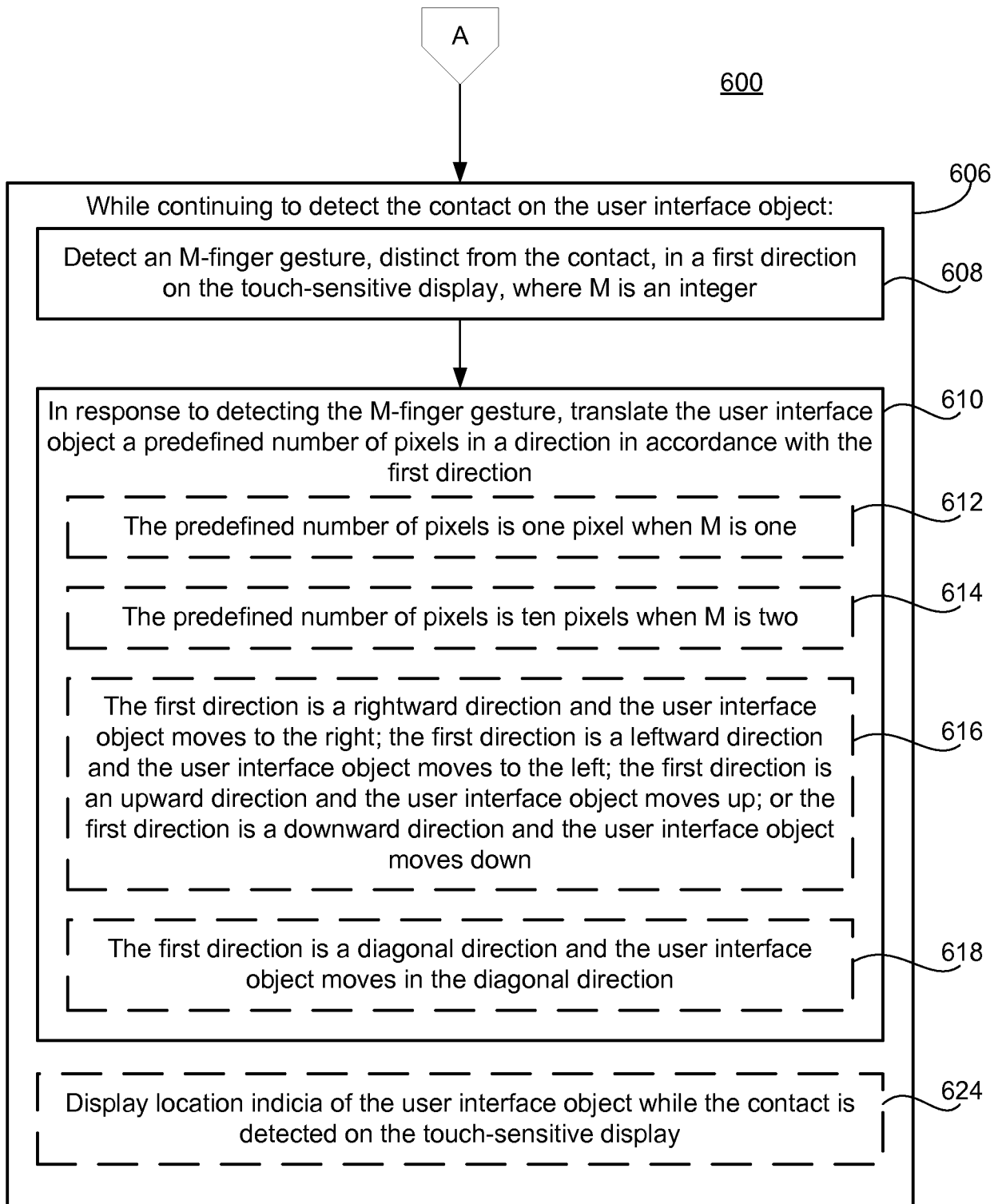


Figure 6B

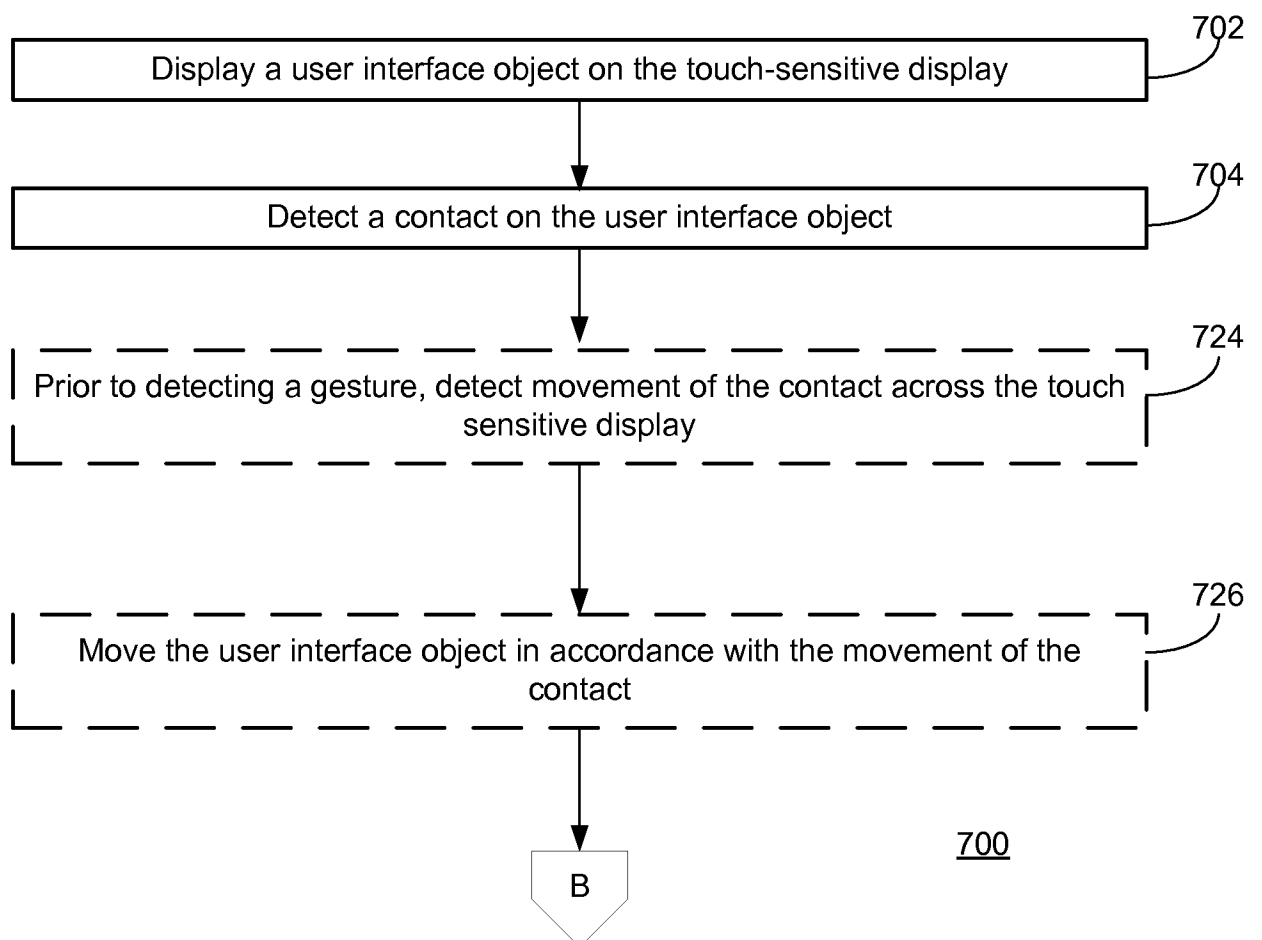


Figure 7A

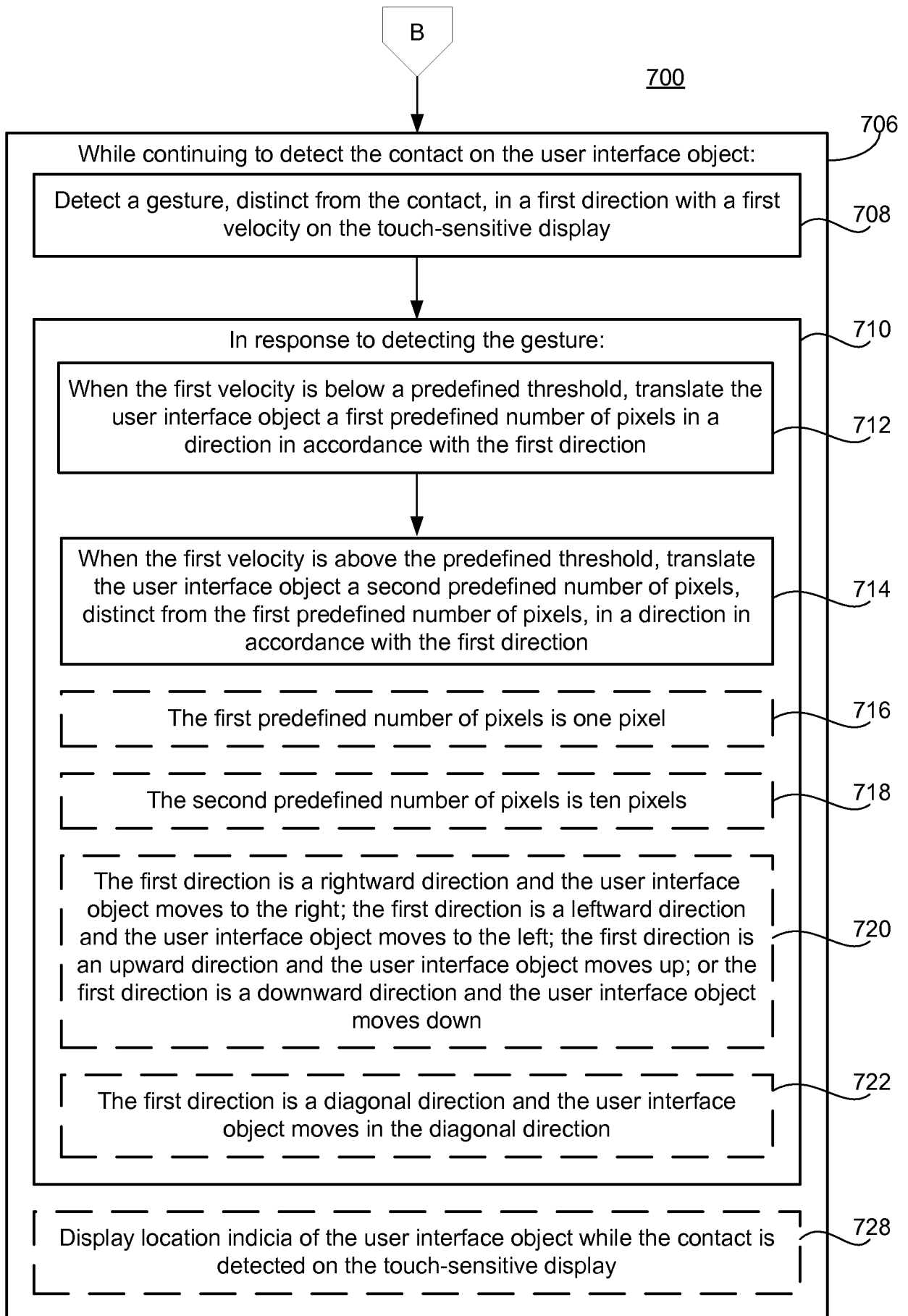
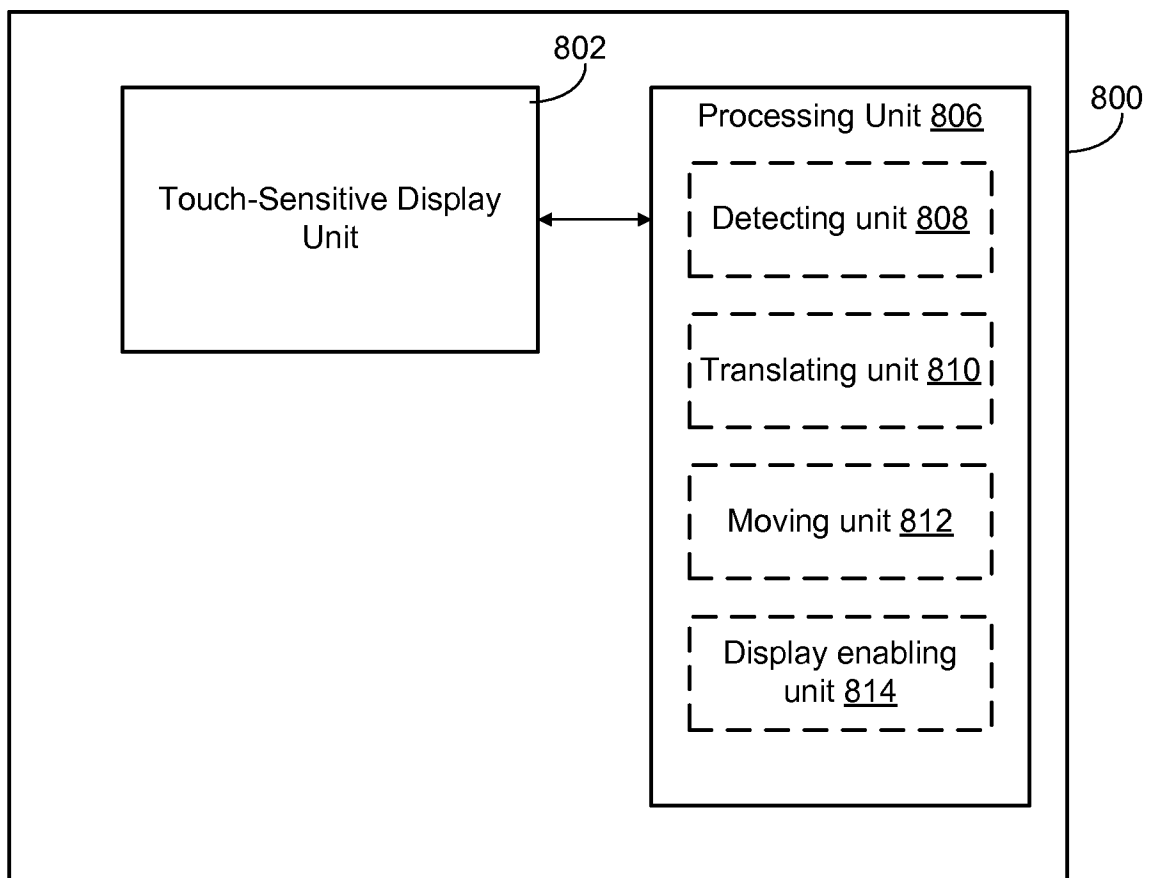


Figure 7B

**Figure 8**

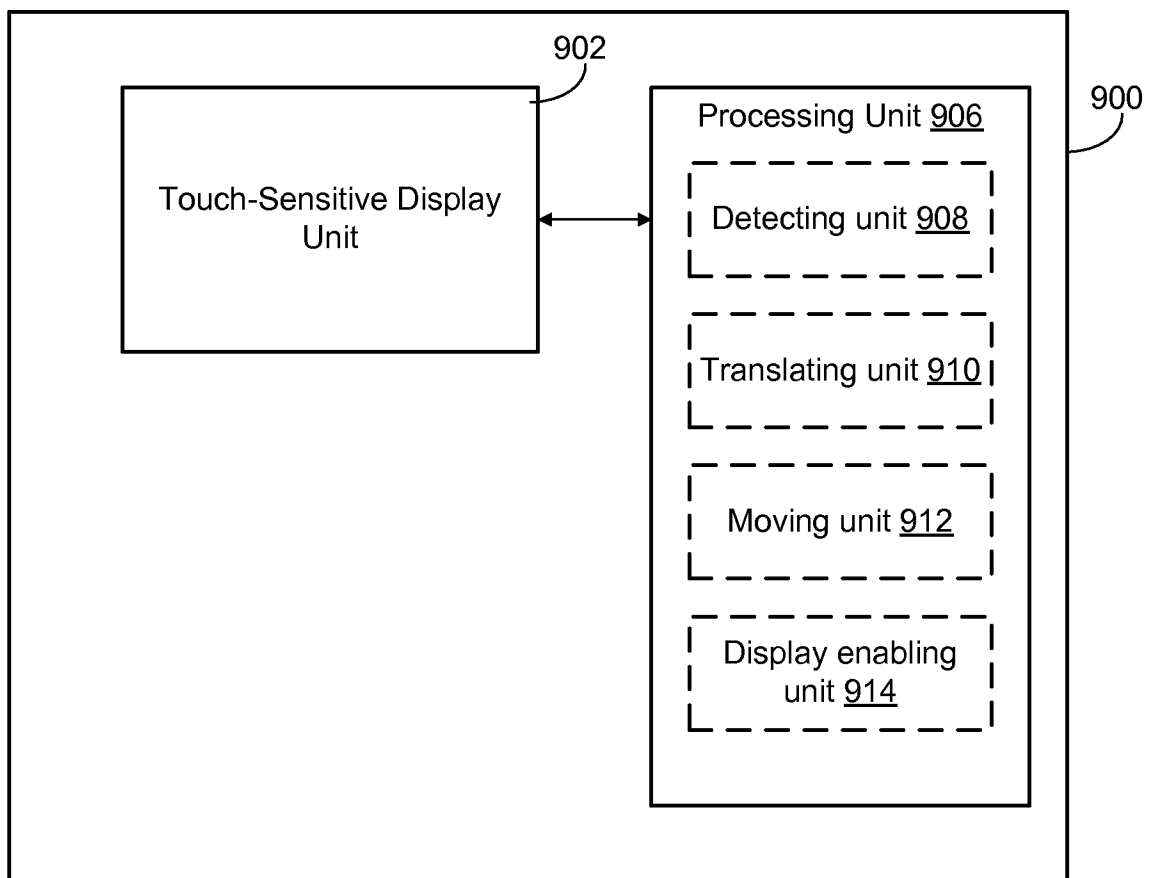


Figure 9

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2011/022525

A. CLASSIFICATION OF SUBJECT MATTER
 INV. G06F3/048
 ADD.

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

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

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

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2008/138046 A1 (RPO PTY LTD [AU]; ATKINS GRAHAM ROY [AU]; MAXWELL IAN ANDREW [AU]) 20 November 2008 (2008-11-20) page 1, line 8 - page 6, line 22 page 7, line 12 - line 33; figures 4A-4B -----	1-55
X	US 2008/150715 A1 (TANG KUAN-CHUN [TW] ET AL) 26 June 2008 (2008-06-26) paragraph [0001] - paragraph [0011] paragraph [0034] - paragraph [0040]; figures 6-8B -----	1-55
X	US 2007/247435 A1 (BENKO HRVOJE [US] ET AL) 25 October 2007 (2007-10-25) paragraph [0001] - paragraph [0014] paragraph [0035] - paragraph [0039] ----- -/-	1-55



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 document 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

19 April 2011

Date of mailing of the international search report

27/04/2011

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
 NL - 2280 HV Rijswijk
 Tel. (+31-70) 340-2040,
 Fax: (+31-70) 340-3016

Authorized officer

Anticoli, Claud

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2011/022525

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2009/183930 A1 (YANG WEI-WEN [TW] ET AL) 23 July 2009 (2009-07-23) paragraph [0002] - paragraph [0017] paragraph [0036] - paragraph [0037]; figures 3A-3B -----	1-55
X	US 2009/307589 A1 (INOSE TSUTOMU [JP] ET AL) 10 December 2009 (2009-12-10) paragraph [0001] - paragraph [0019] paragraph [0054] - paragraph [0079]; figures 4A-5D -----	1-55
E	EP 2 284 675 A2 (LG ELECTRONICS INC [KR]) 16 February 2011 (2011-02-16) paragraph [0001] - paragraph [0008] -----	1,4,6,8, 11,13, 15,18, 20,29, 31,33, 35,37, 40,42, 45,47
A	US 2009/184939 A1 (WOHLSTADTER GIL [IL] ET AL) 23 July 2009 (2009-07-23) paragraph [0002] - paragraph [0036] -----	1-55
A	US 5 825 352 A (BISSET STEPHEN J [US] ET AL) 20 October 1998 (1998-10-20) column 1, line 10 - column 4, line 49 -----	1-55
A	US 2009/051660 A1 (FELAND III JOHN MORGAN [US] ET AL) 26 February 2009 (2009-02-26) paragraph [0001] - paragraph [0006] paragraph [0038] - paragraph [0039]; figures 3-9 -----	1-55

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2011/022525

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2008138046 A1	20-11-2008	US 2011069018 A1	24-03-2011
US 2008150715 A1	26-06-2008	JP 2008159032 A	10-07-2008
US 2007247435 A1	25-10-2007	NONE	
US 2009183930 A1	23-07-2009	NONE	
US 2009307589 A1	10-12-2009	CN 101599002 A	09-12-2009
		JP 2009294857 A	17-12-2009
EP 2284675 A2	16-02-2011	KR 20110016337 A	17-02-2011
		US 2011037714 A1	17-02-2011
US 2009184939 A1	23-07-2009	EP 2243072 A2	27-10-2010
		WO 2009093241 A2	30-07-2009
US 5825352 A	20-10-1998	NONE	
US 2009051660 A1	26-02-2009	NONE	