



US 20120084680A1

(19) **United States**

(12) **Patent Application Publication**

Gimpl et al.

(10) **Pub. No.: US 2012/0084680 A1**

(43) **Pub. Date: Apr. 5, 2012**

(54) **GESTURE CAPTURE FOR MANIPULATION OF PRESENTATIONS ON ONE OR MORE DEVICE DISPLAYS**

(75) Inventors: **Martin Gimpl**, Helsinki (FI); **Ron Cassar**, Burlington (CA); **Maxim Marintchenko**, Vaughan (CA); **Nikhil Swaminathan**, Toronto (CA)

(73) Assignee: **IMERJ LLC**, Broomfield, CO (US)

(21) Appl. No.: **13/247,977**

(22) Filed: **Sep. 28, 2011**

tion No. 61/389,117, filed on Oct. 1, 2010, provisional application No. 61/389,087, filed on Oct. 1, 2010.

Publication Classification

(51) **Int. Cl.**
G06F 3/048 (2006.01)

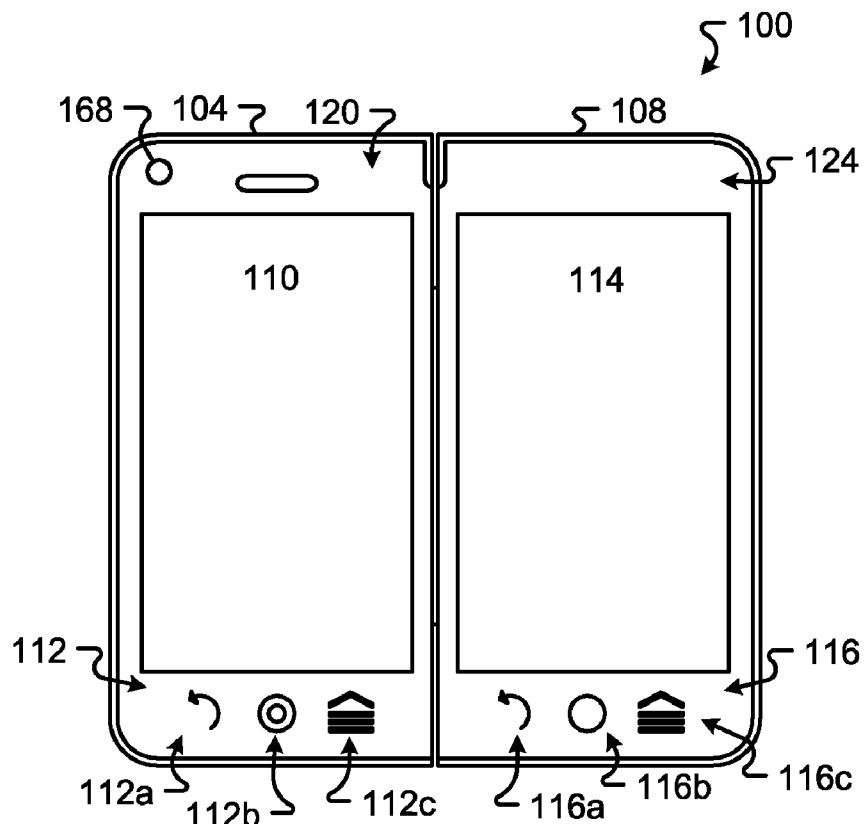
(52) **U.S. Cl.** **715/761**

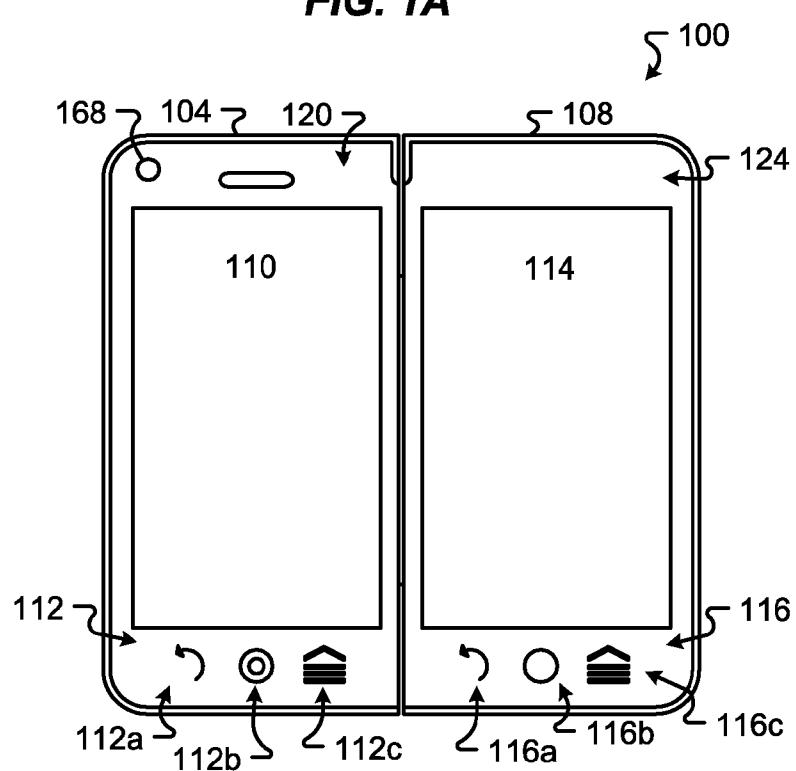
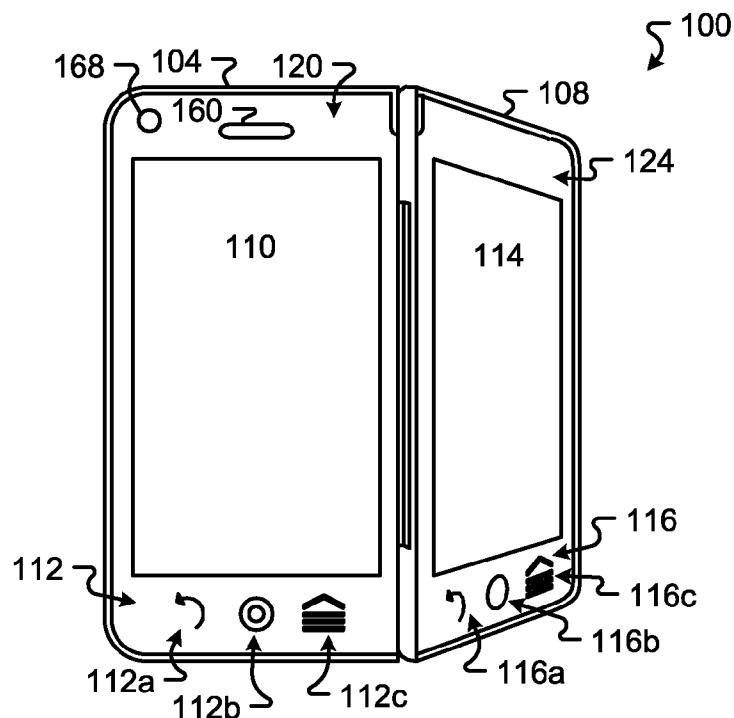
(57) **ABSTRACT**

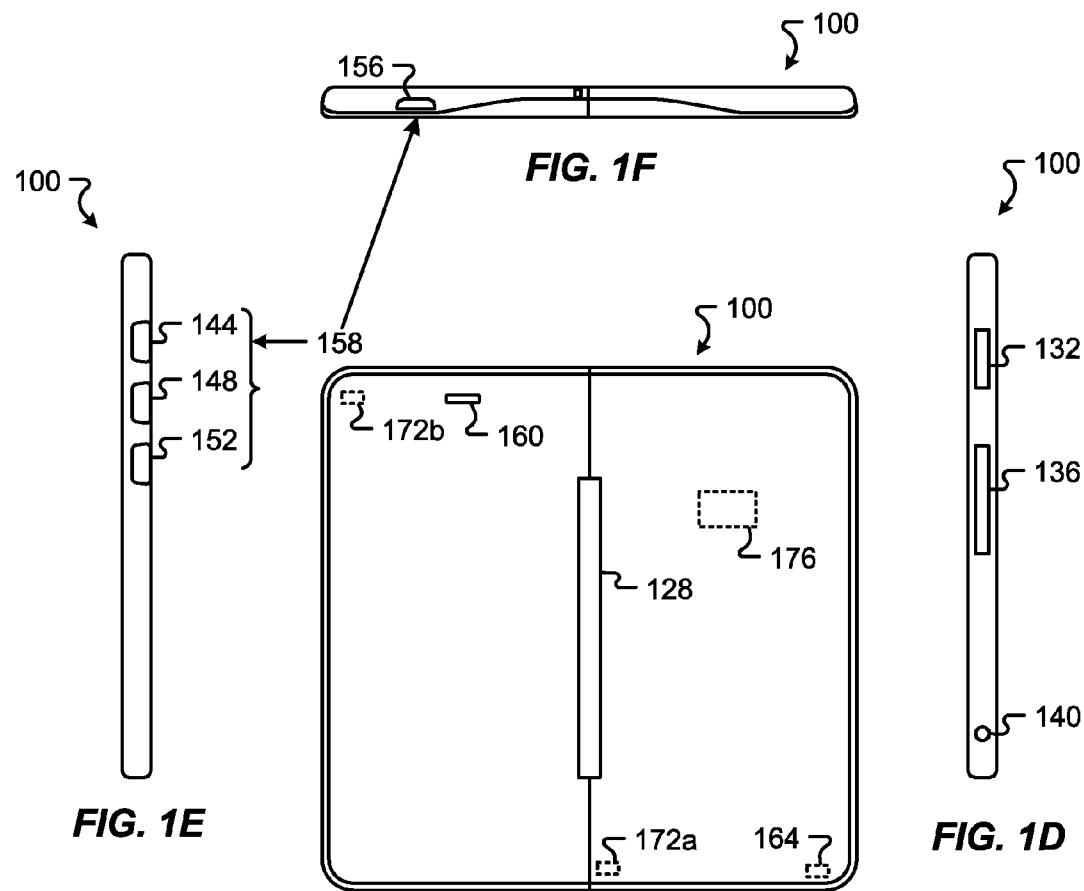
An intuitive technique for inputting user gestures into a handheld computing device is disclosed allowing a user to better manipulate different types of screen display presentations, such as desktops and application windows, when performing tasks thereon, e.g., minimization, maximization, moving between display screens, and increasing/decreasing a display thereof across multiple display screens. For manipulating an application window on a device display screen for performing tasks as identified above, user gestures are input to a corresponding gesture capture area for this display screen, wherein this capture area is separate from this display screen.

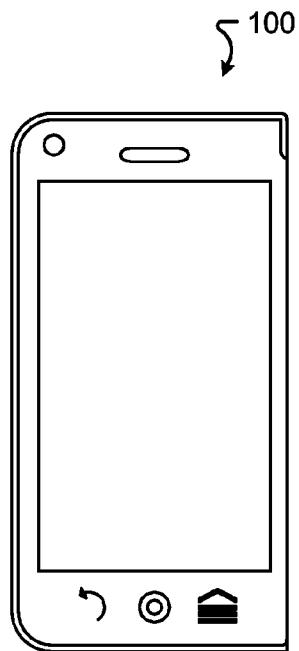
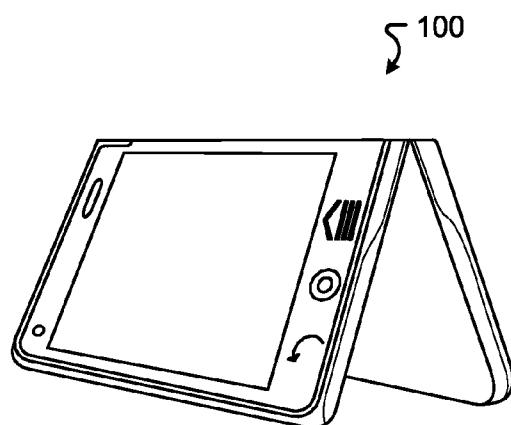
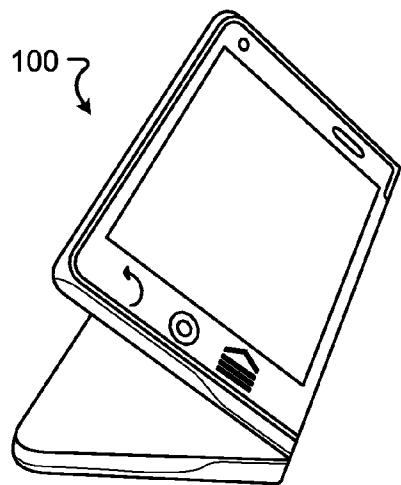
Related U.S. Application Data

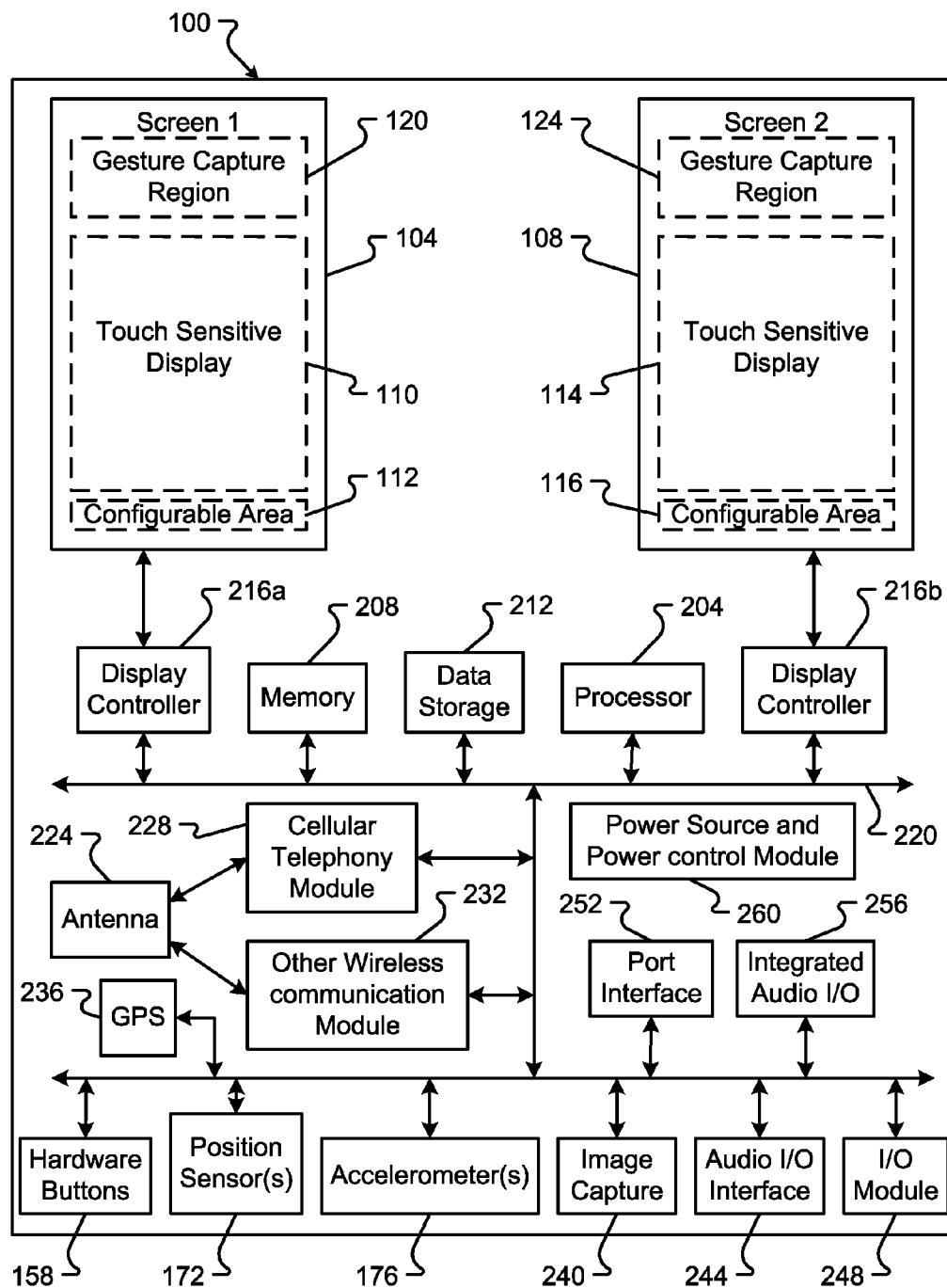
(60) Provisional application No. 61/539,884, filed on Sep. 27, 2011, provisional application No. 61/458,150, filed on Nov. 17, 2010, provisional application No. 61/389,000, filed on Oct. 1, 2010, provisional applica-

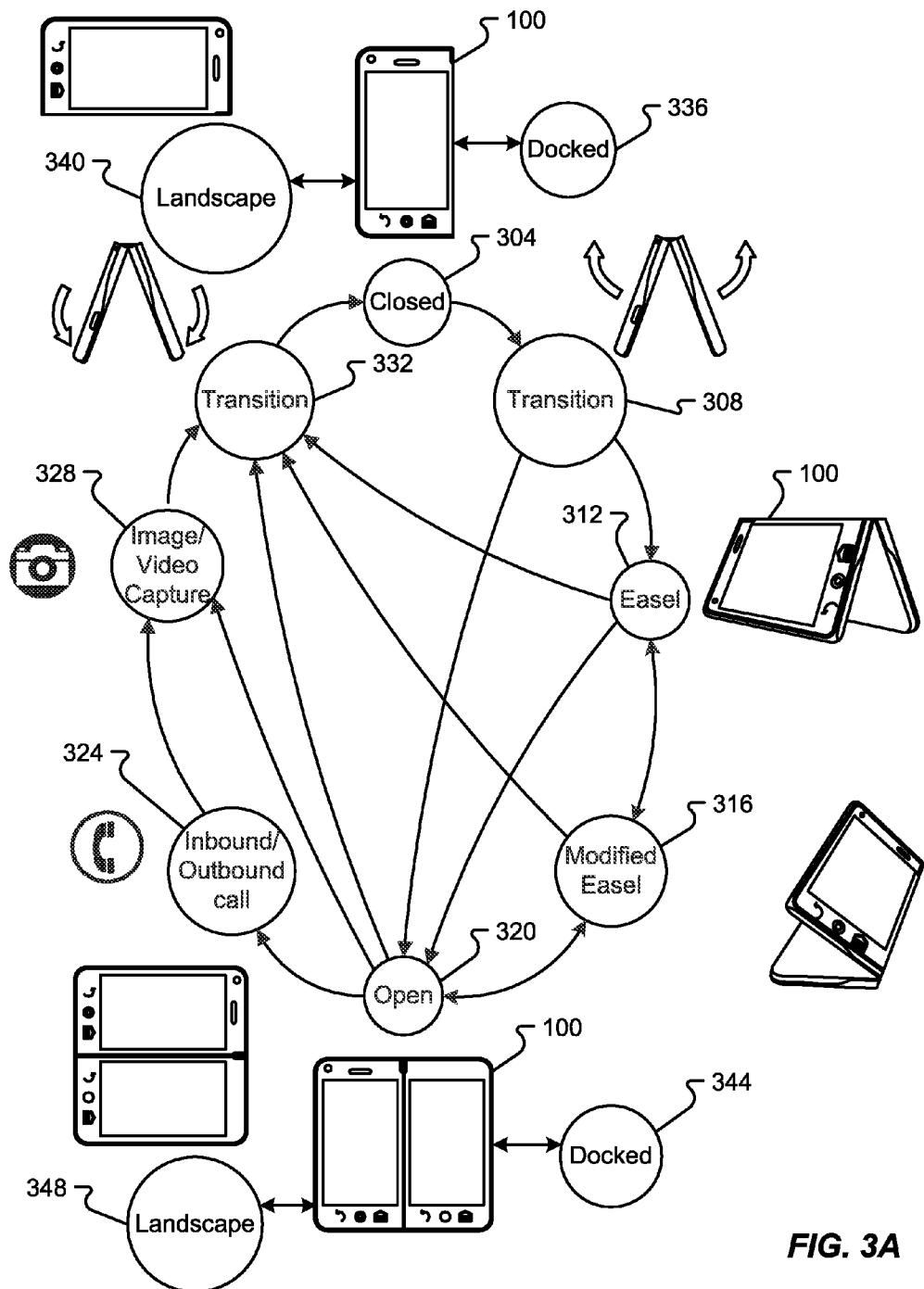






**FIG. 1H****FIG. 1I****FIG. 1J**

**FIG. 2**

**FIG. 3A**

| PORTRAIT | | | | | | | LANDSCAPE | | | | | |
|----------|------------------|--------|-------|-------------------|-------|------------------|-----------|--------|-------|-------------------|-------|------------------|
| | OPEN | CLOSED | EASEL | MODIFIED EASEL | PHONE | IMAGE / VIDEO | OPEN | CLOSED | EASEL | MODIFIED EASEL | PHONE | IMAGE / VIDEO |
| P | OPEN | X | HT | HT | P | I | AT | HT | HAT | HAT | P | I |
| O | CLOSED | HT | X | HAT | P | I | HAT | AT | HAT | HAT | P | I |
| R | EASEL | HT | HT | X | P | I | HAT | HAT | HAT | HAT | P | I |
| T | PHONE | HT | X | HT | X | I | HAT | HAT | HAT | HAT | X | I |
| R | IMAGE / VIDEO | HT | HT | HT | X | P | X | HAT | HT | HAT | X | HAT |
| A | | | | | | | | | | | | |
| T | | | | | | | | | | | | |
| L | OPEN | AT | HAT | HAT | P | I | X | HT | HAT | HAT | P | I |
| A | CLOSED | HAT | AT | HAT | P | I | HT | X | HAT | HAT | P | I |
| N | EASEL | HAT | HAT | HAT | P | I | HT | HT | X | HAT | P | I |
| D | MODIFIED | HAT | HAT | HAT | P | I | HT | HT | HAT | X | P | I |
| S | EASEL | | | | | | | | | | | |
| C | IMAGE / VIDEO | HAT | HAT | HAT | HAT | AT | HT | HT | HAT | P | X | |
| A | | | | | | | | | | | | |
| P | | | | | | | | | | | | |
| E | | | | | | | | | | | | |
| DOCKED | | | | | | | | | | | | |

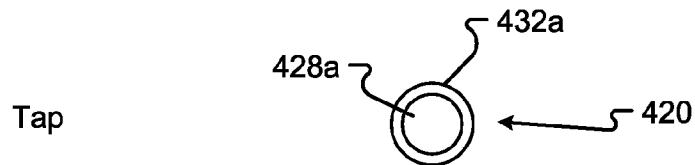
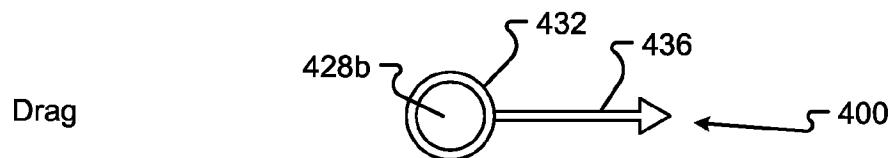
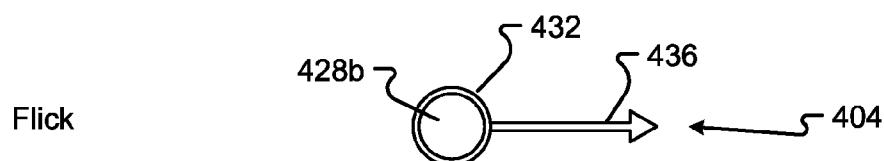
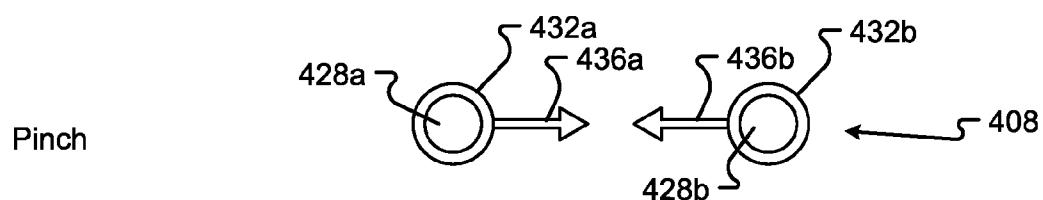
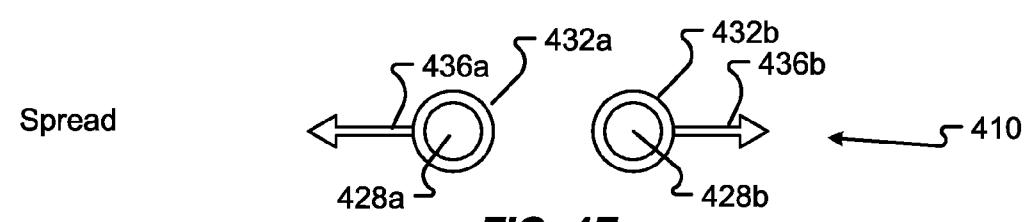
376 { 360 { 364 { 368 { 372 { 352 { 356 { }

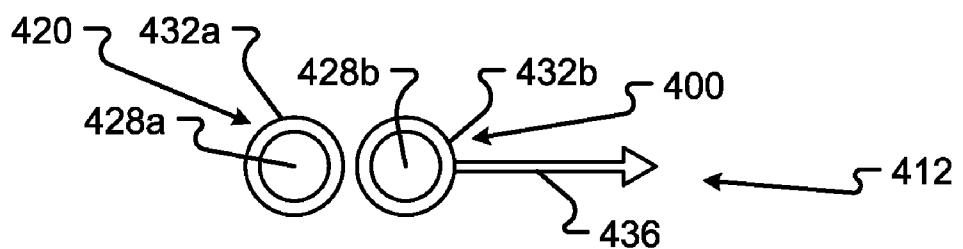
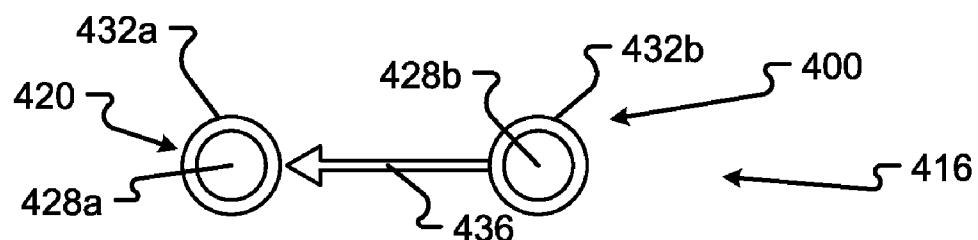
DOCKING SIGNAL

Key:

H - Hall Effect Sensor(s)
a - accelerometer(s)
T - Timer
P - Communications Trigger
I - Image / Video capture Request

FIG. 3B

**FIG. 4A****FIG. 4B****FIG. 4C****FIG. 4D****FIG. 4E****FIG. 4F**

**FIG. 4G****FIG. 4H**

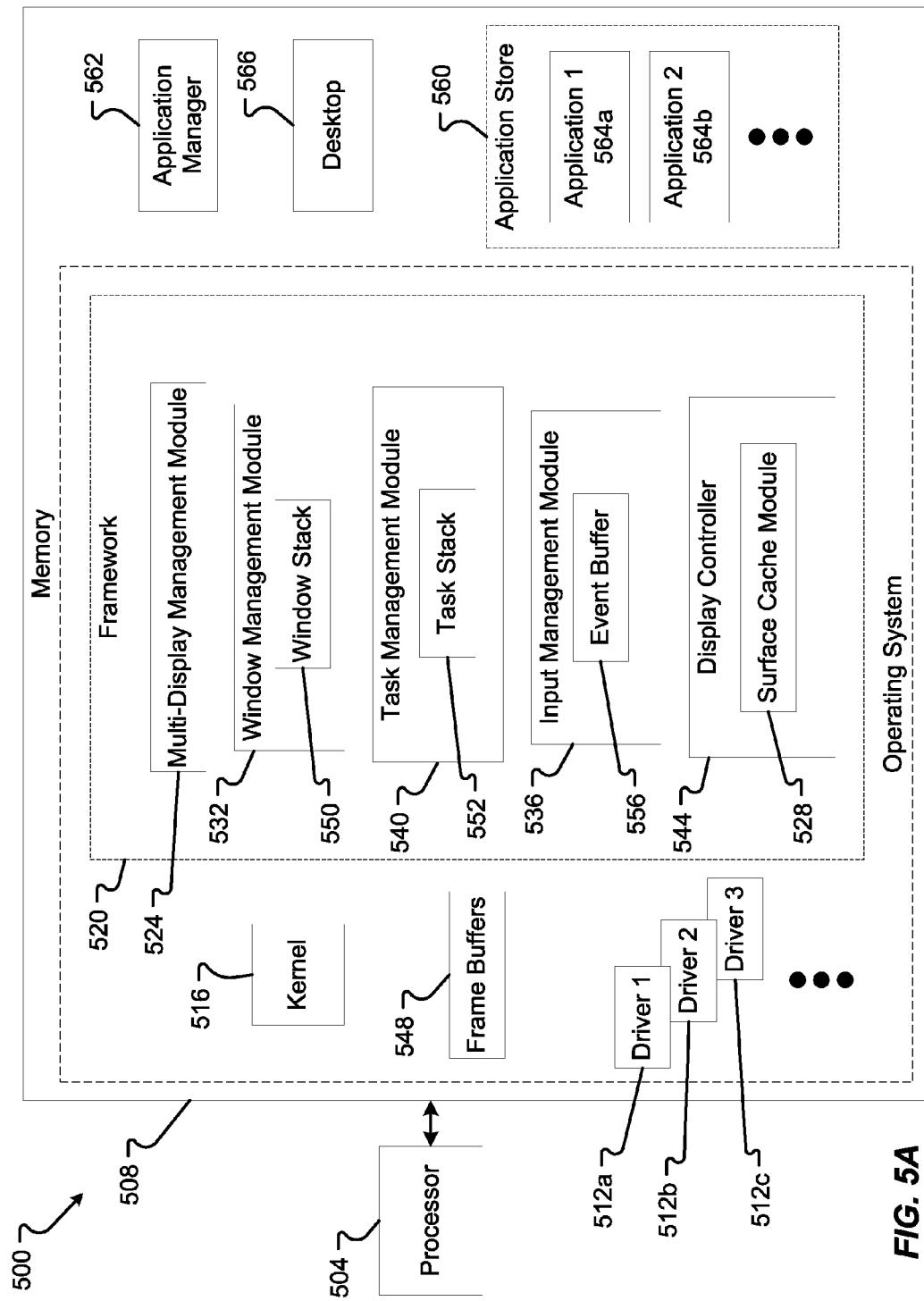
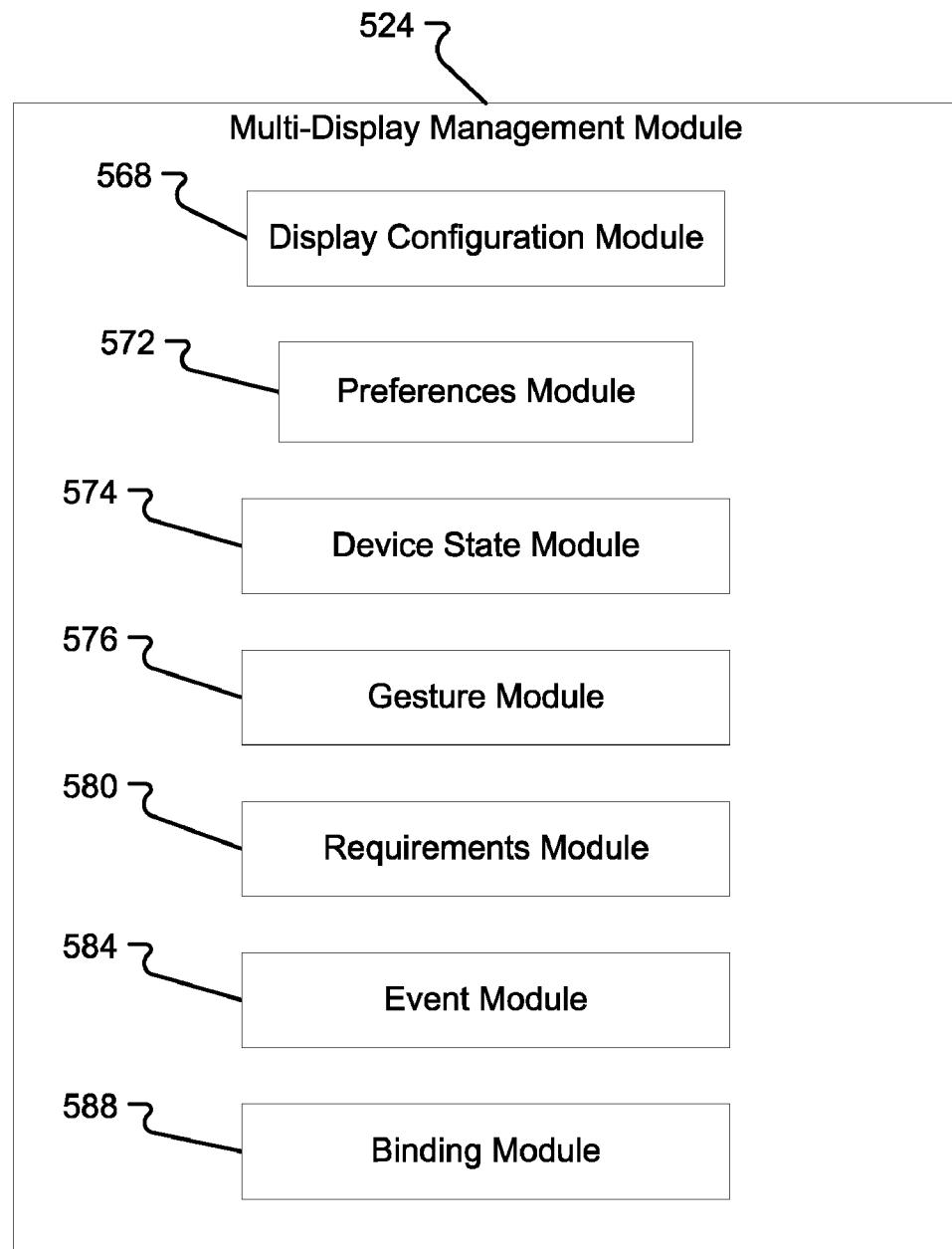


FIG. 54

**FIG. 5B**

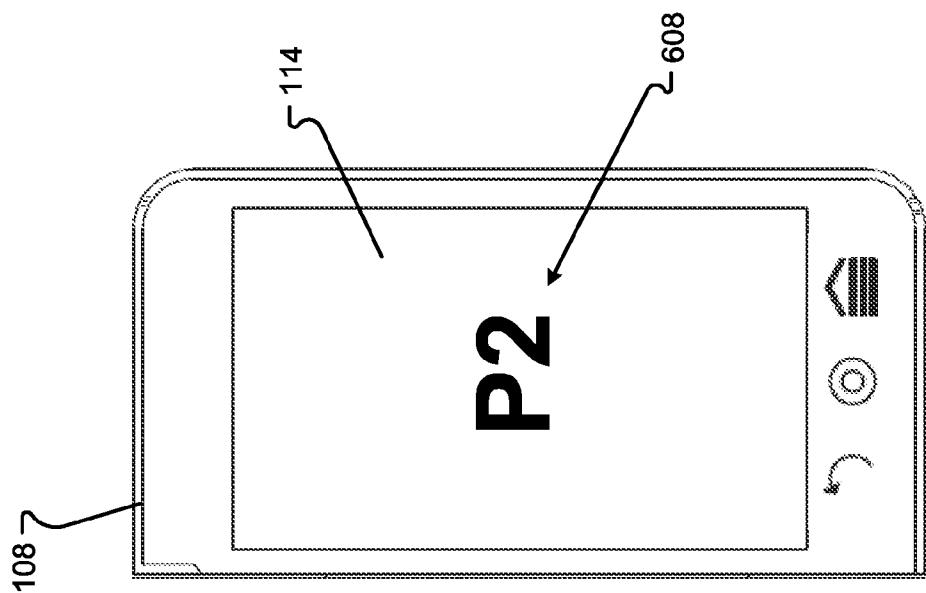


FIG. 6B

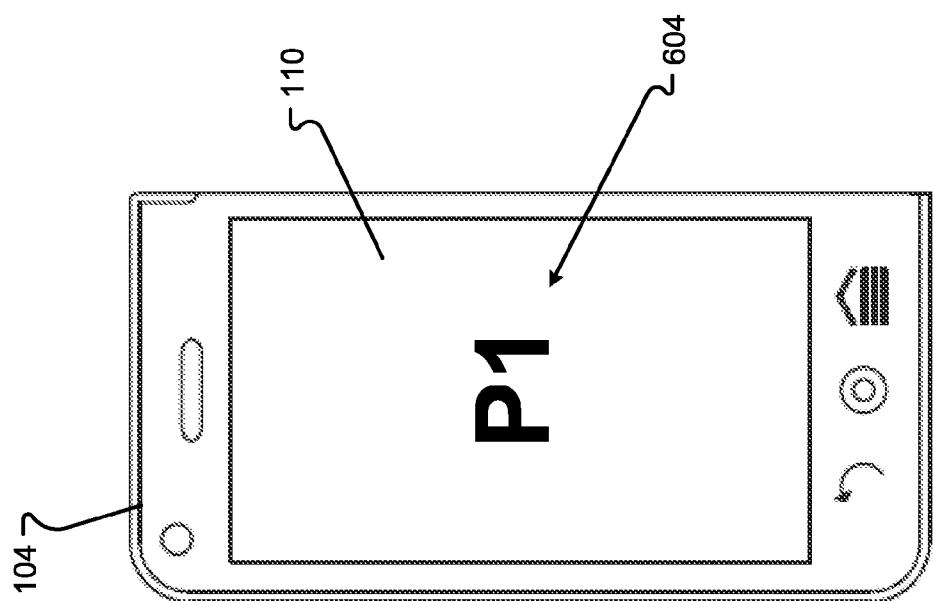


FIG. 6A

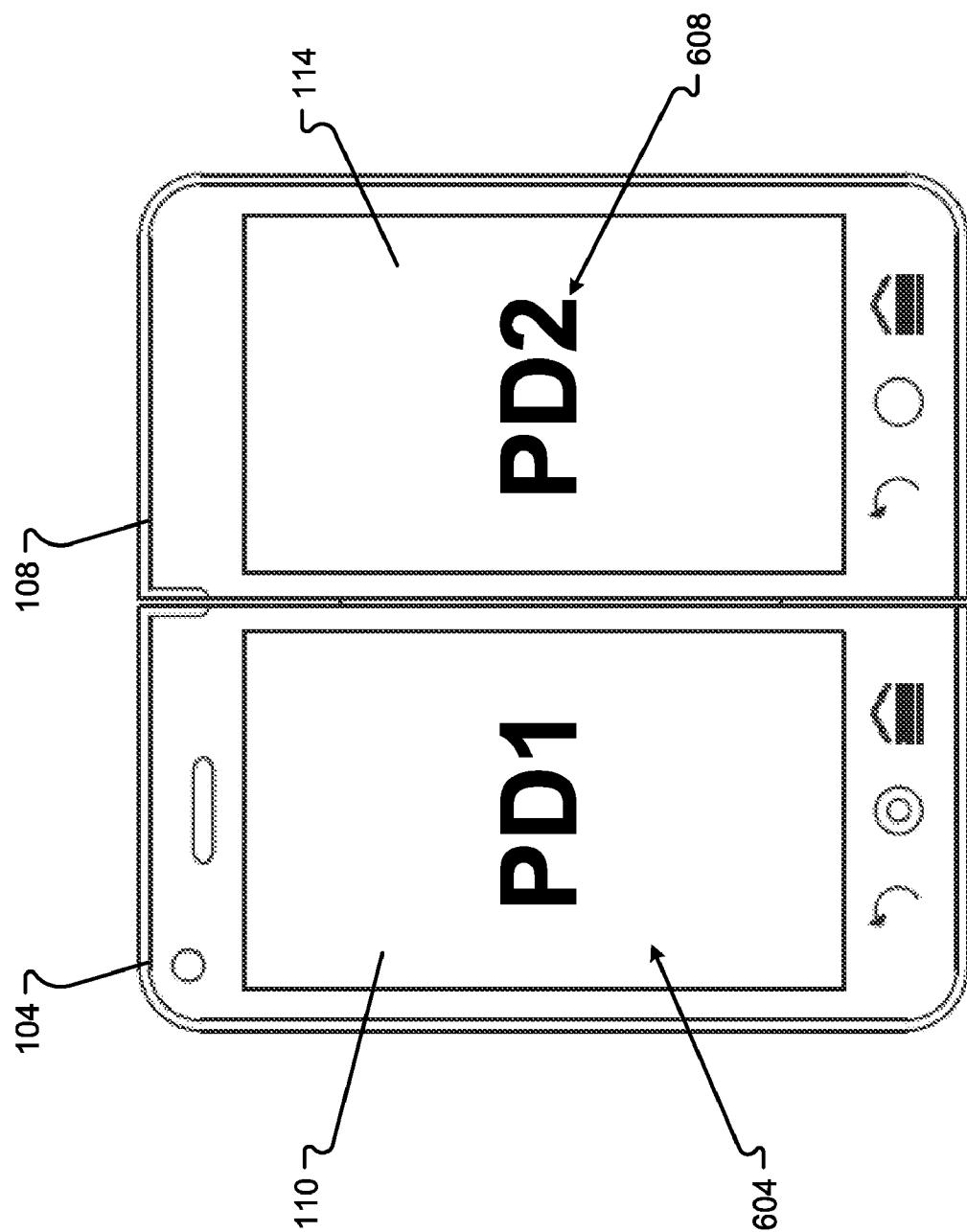


FIG. 6C

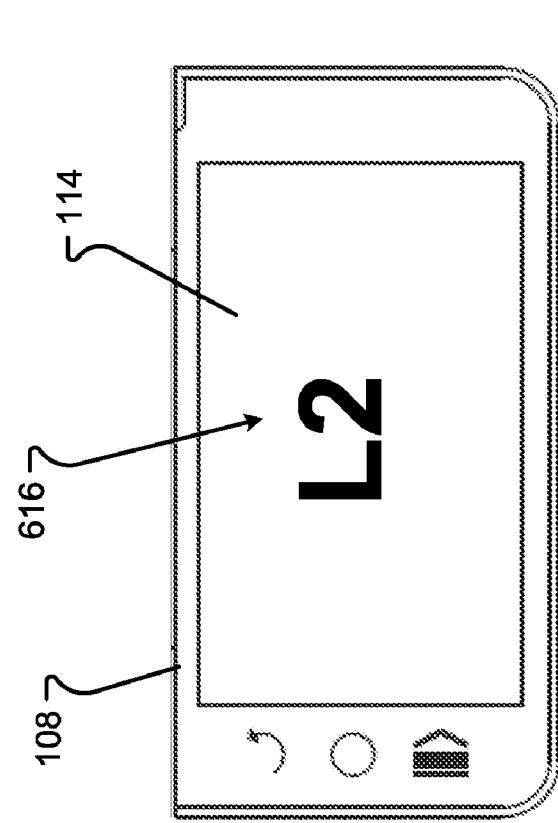


FIG. 6E

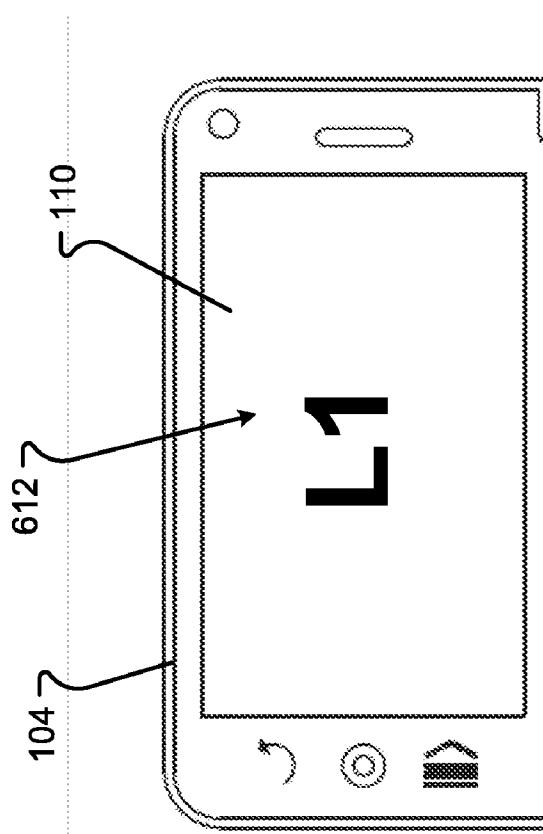


FIG. 6D

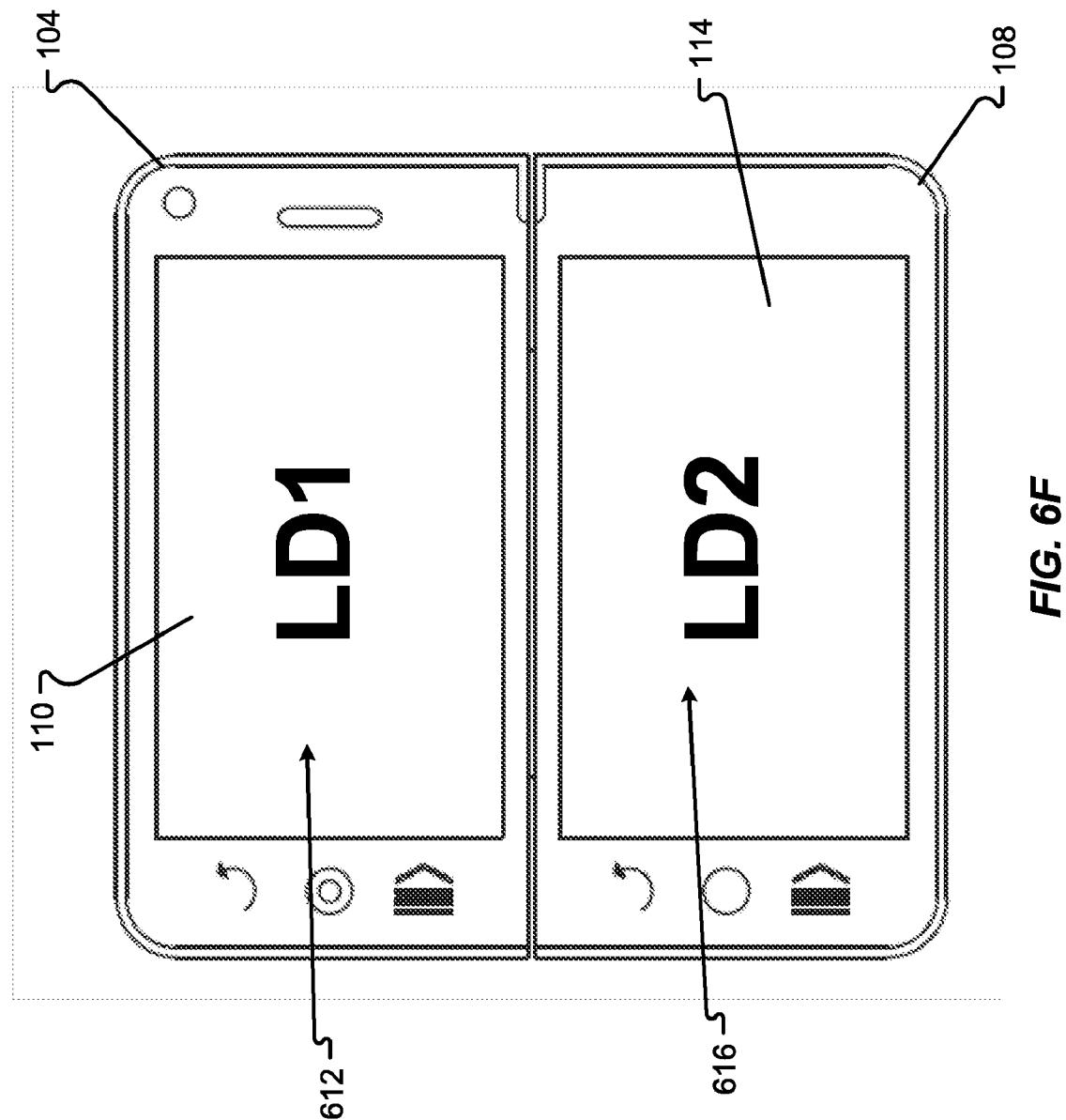
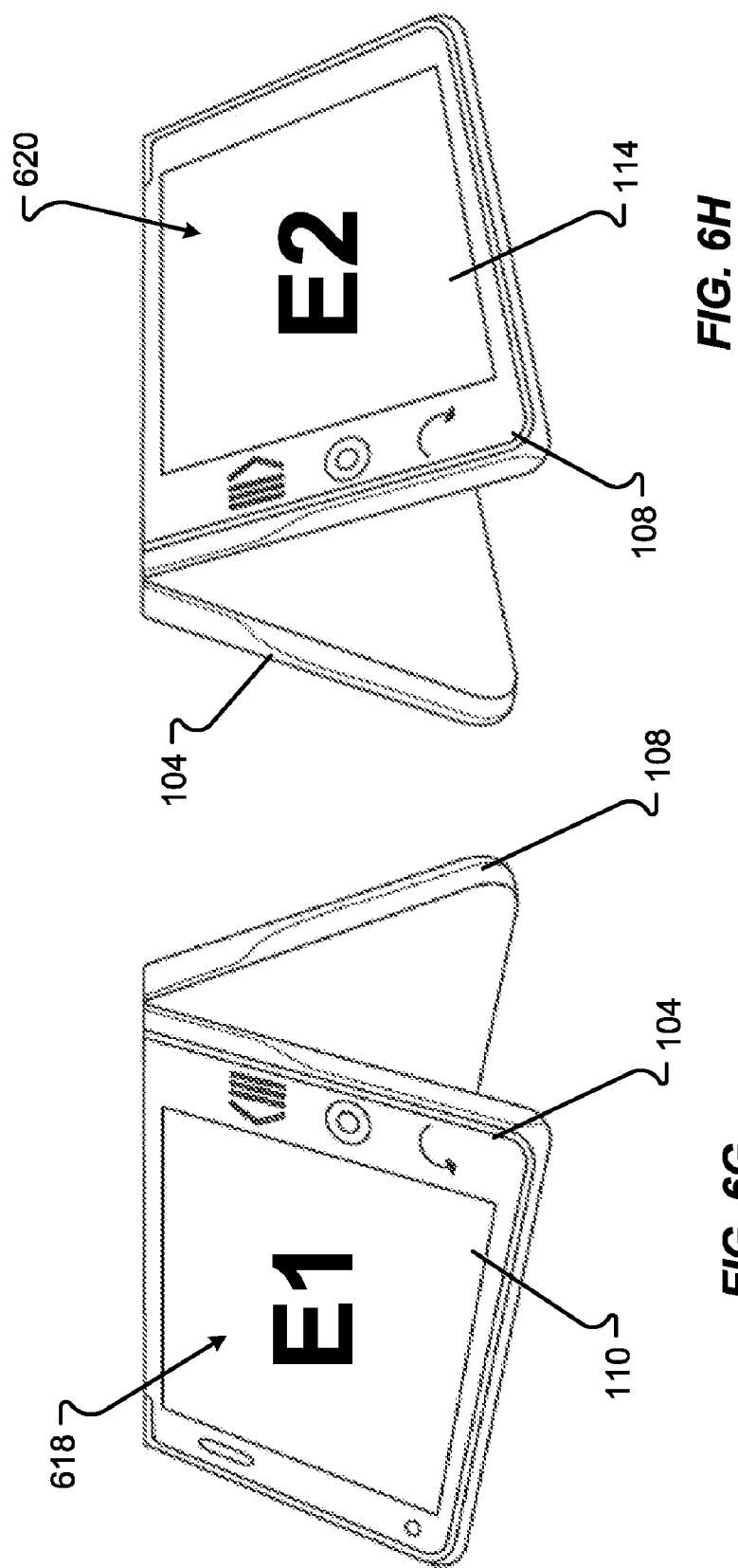


FIG. 6F



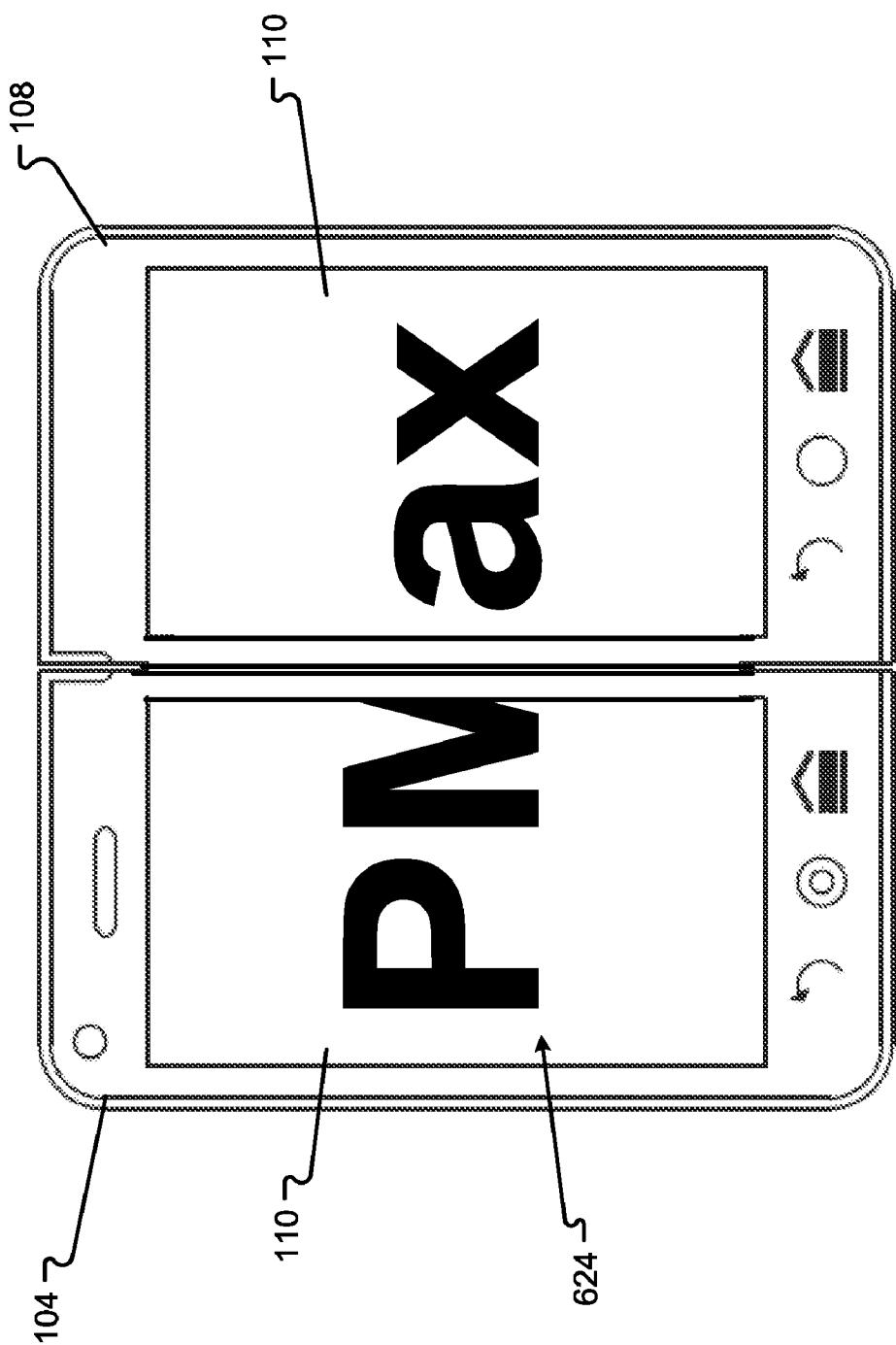
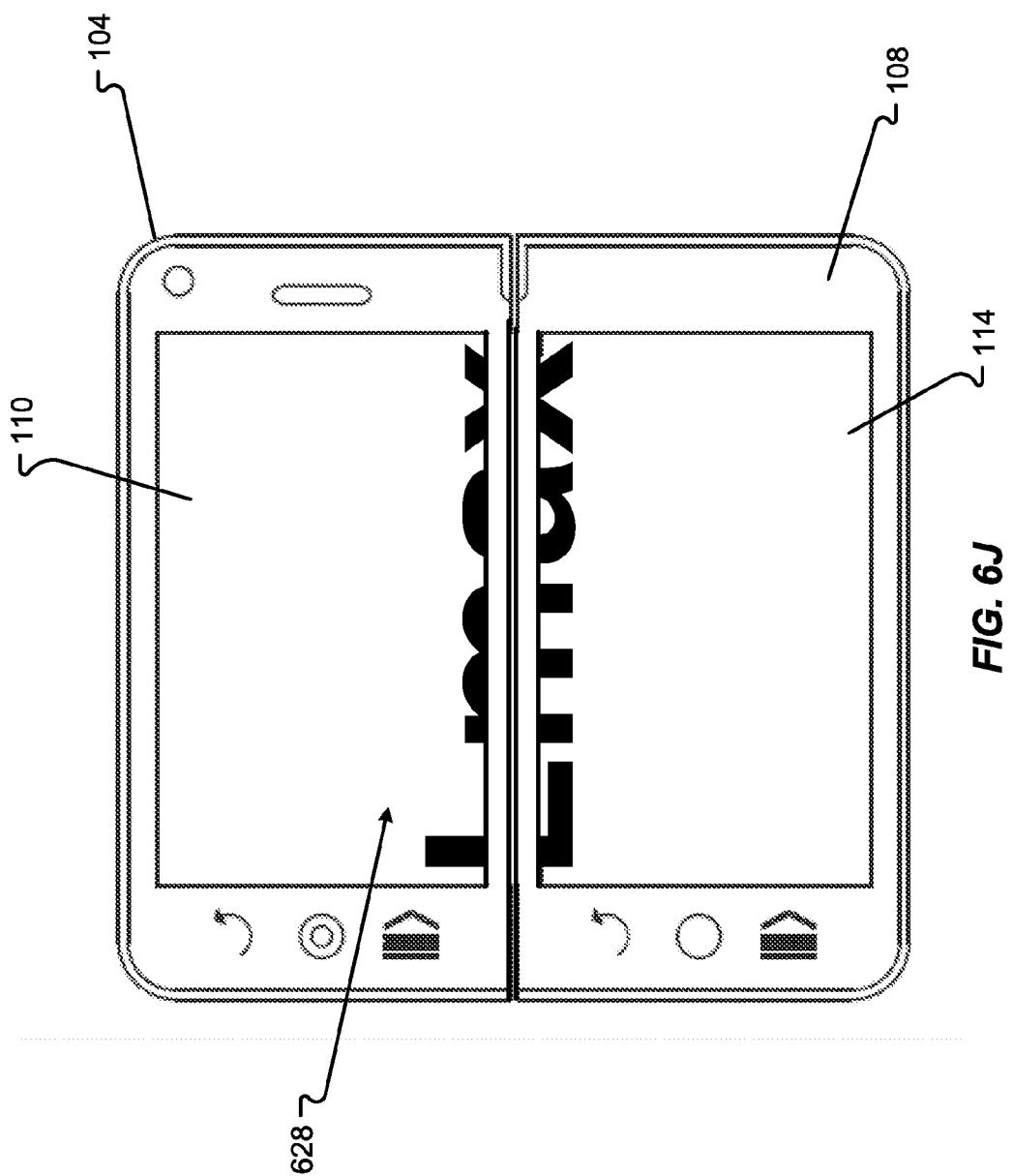
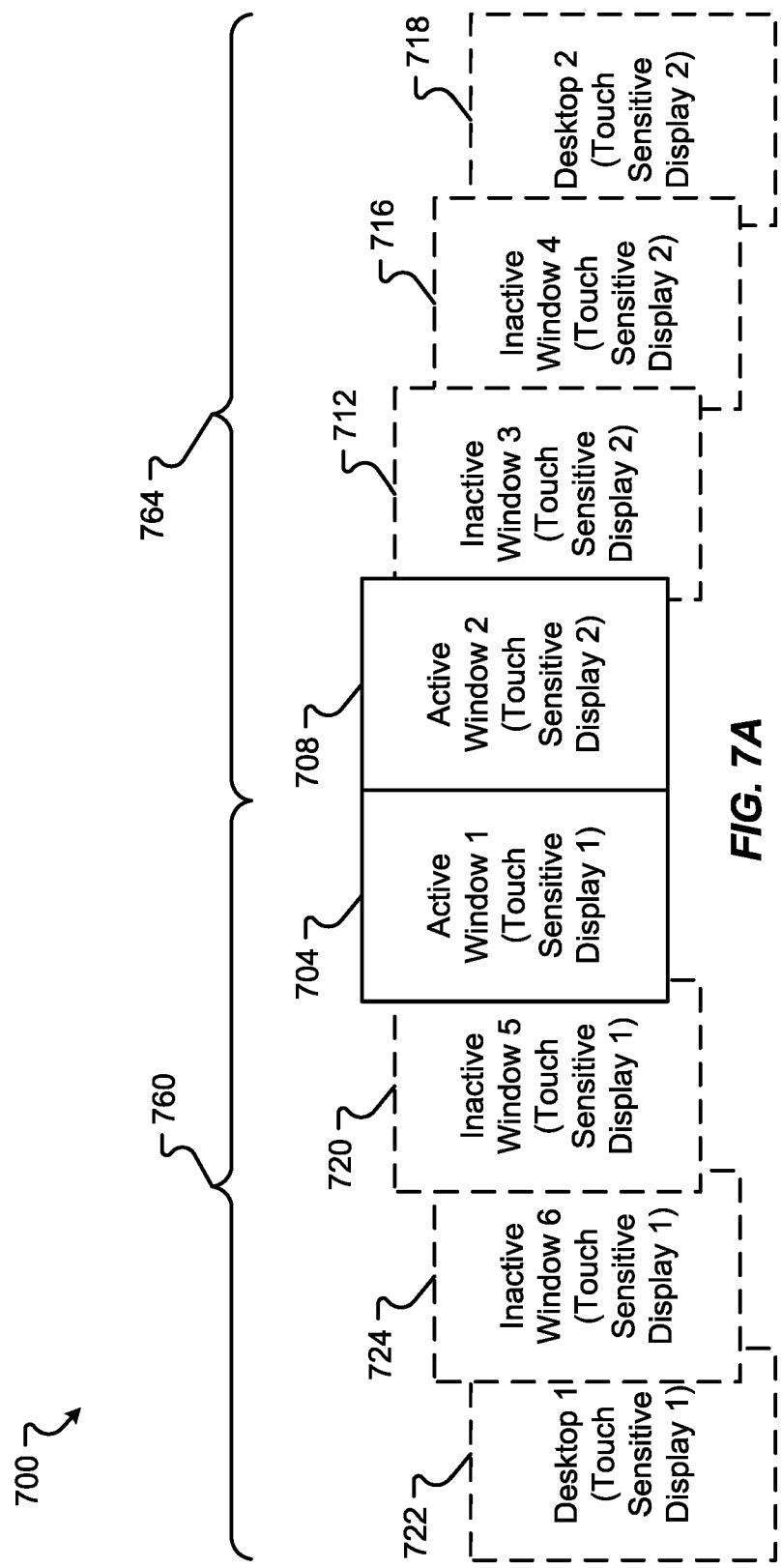
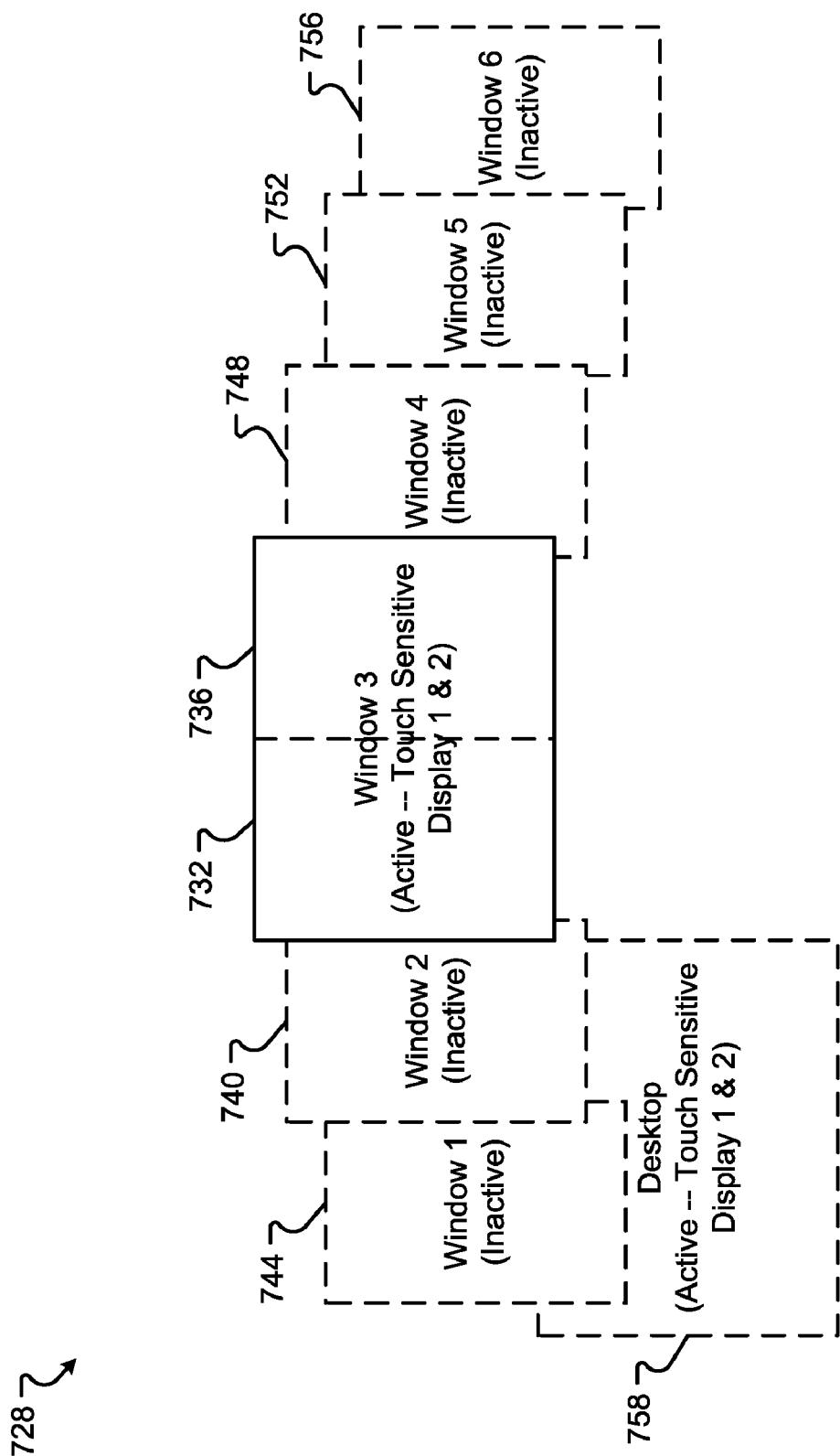


FIG. 6I





**FIG. 7B**

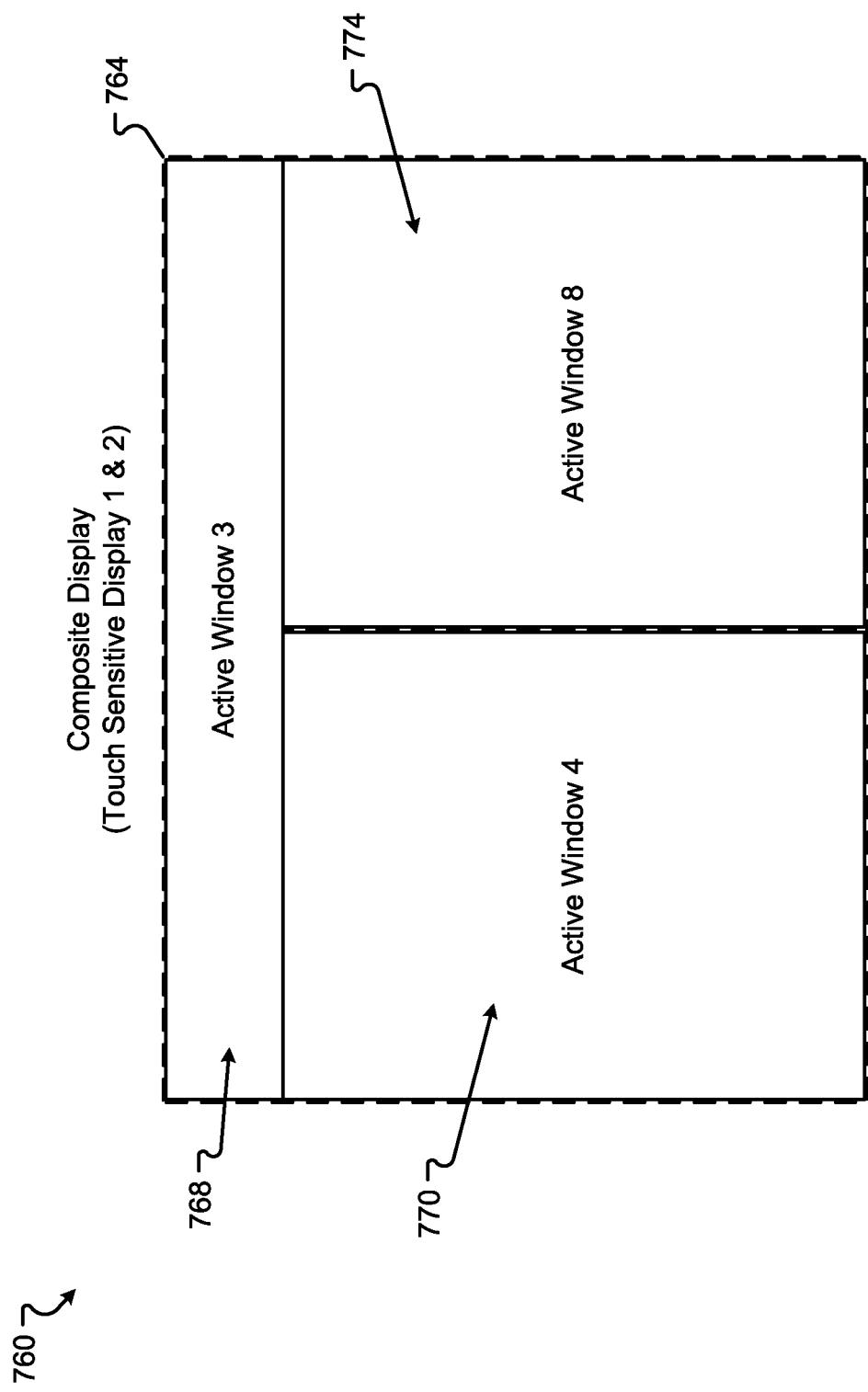


FIG. 7C

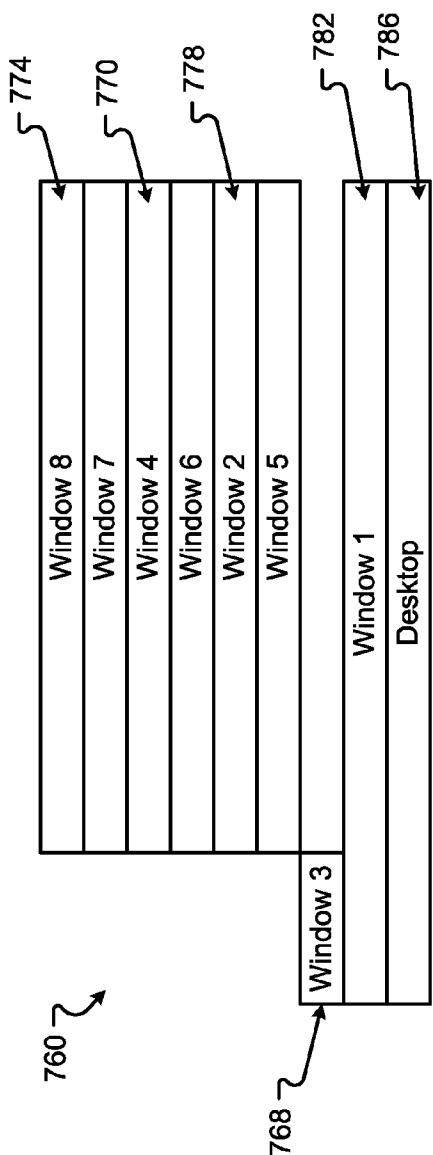


FIG. 7D

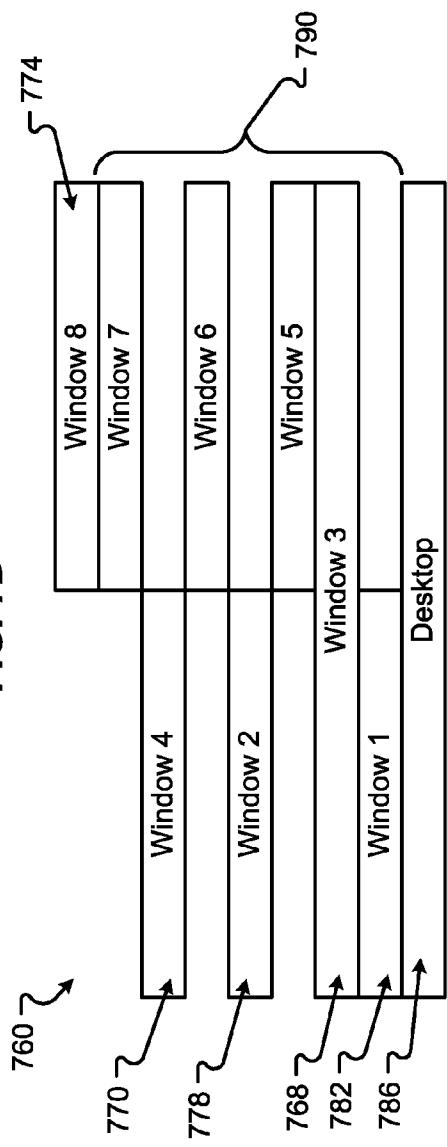


FIG. 7E

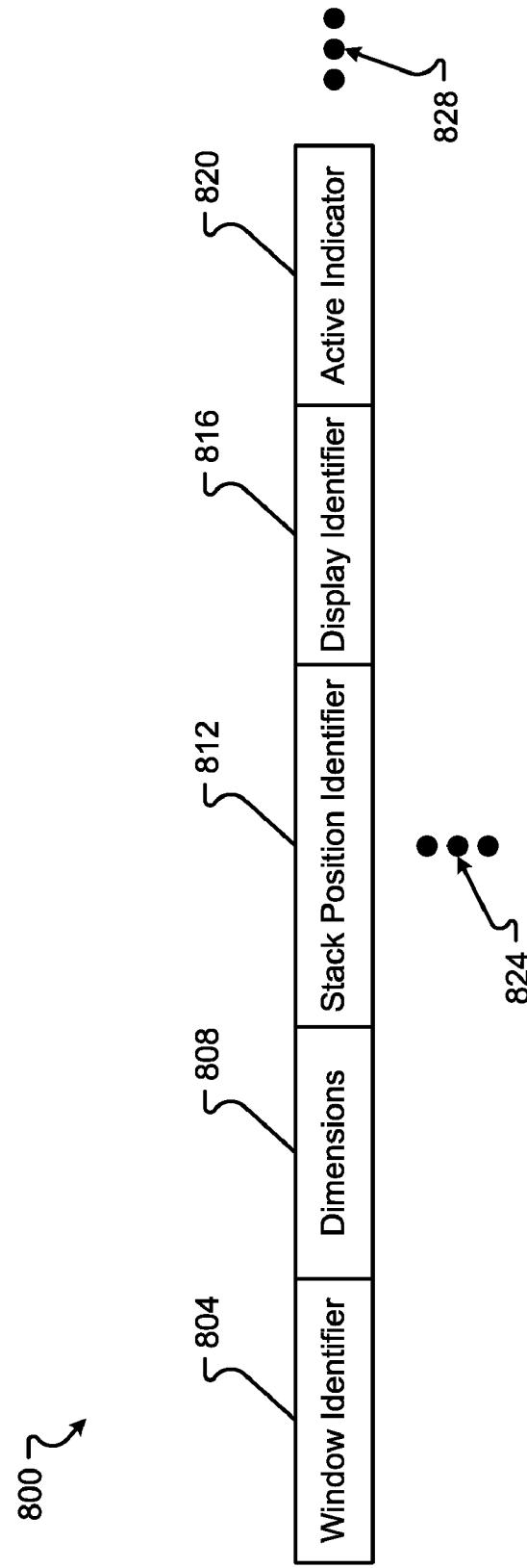
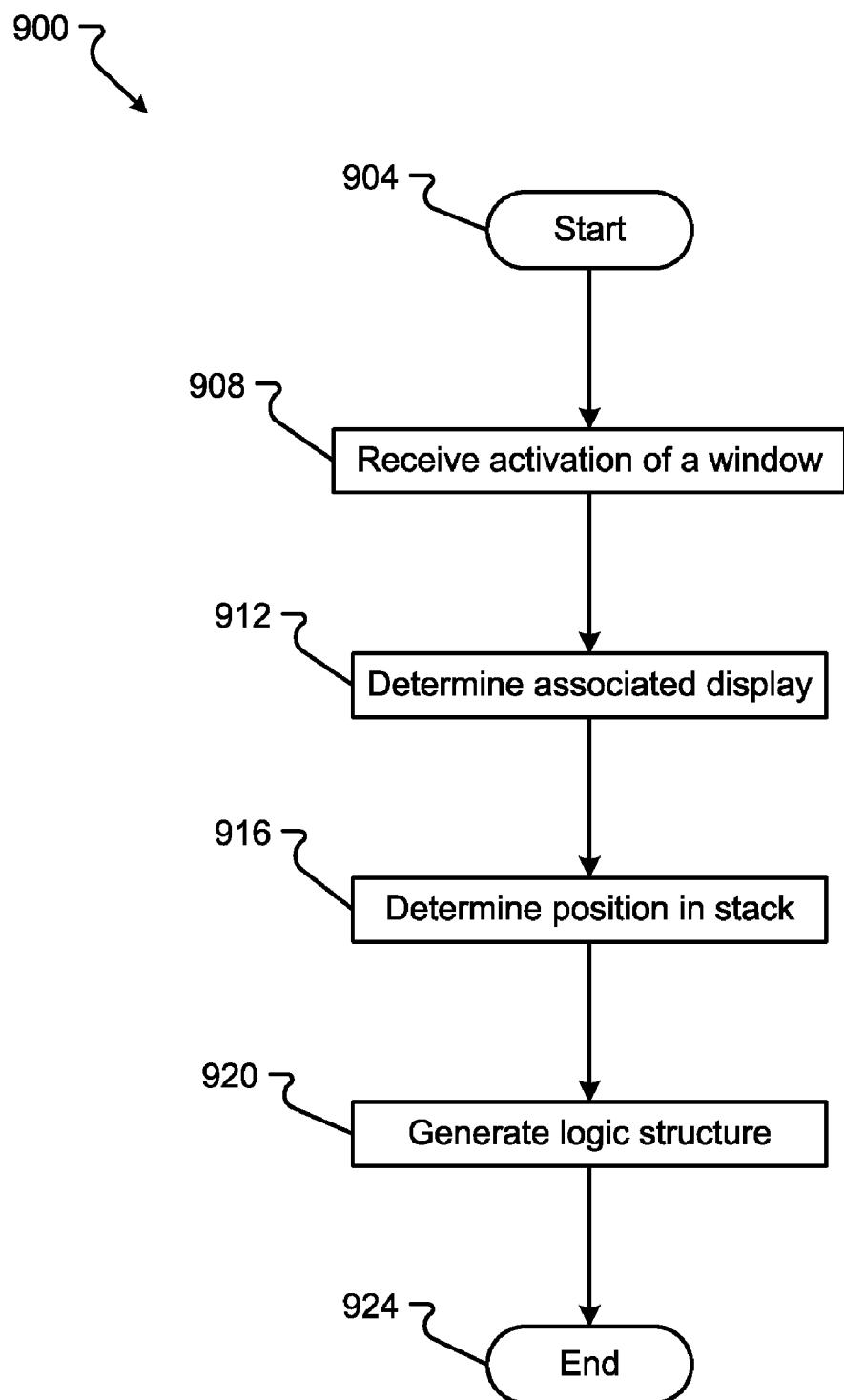


FIG. 8

**FIG. 9**

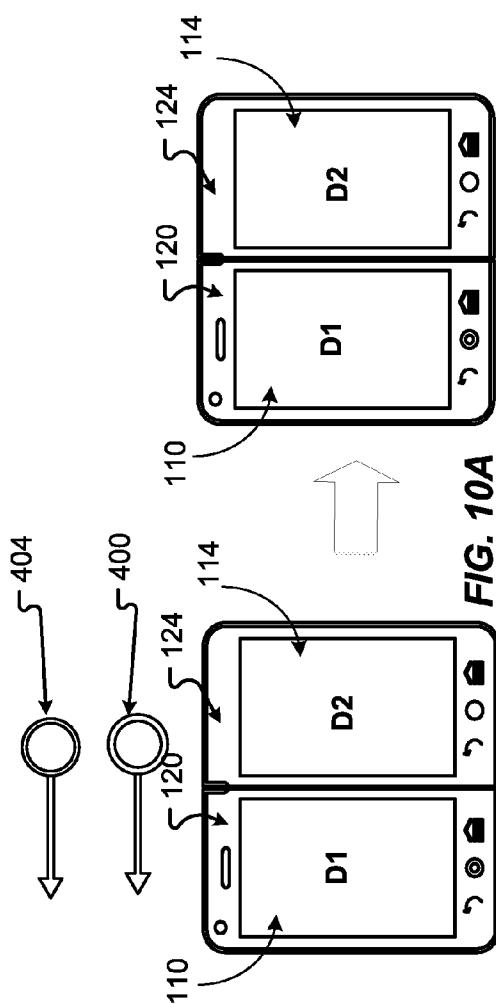


FIG. 10A

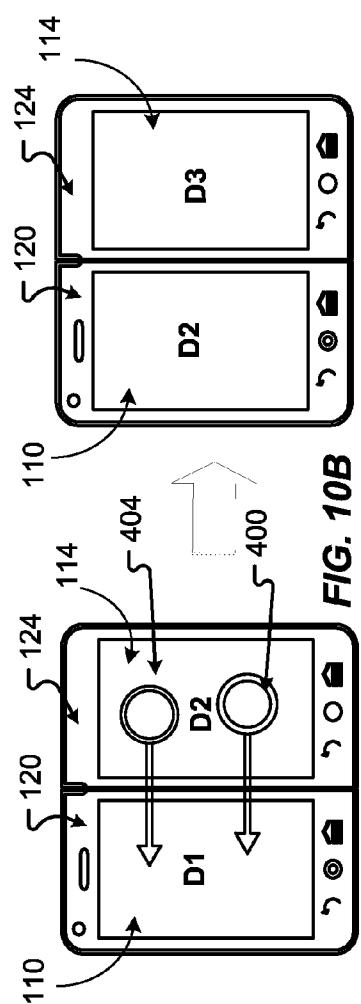


FIG. 10B

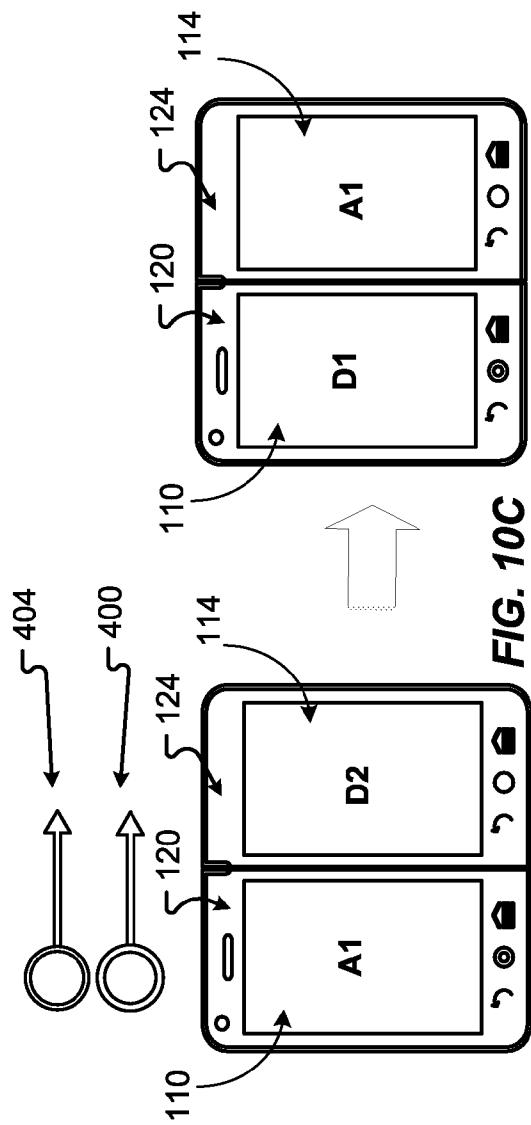


FIG. 10C

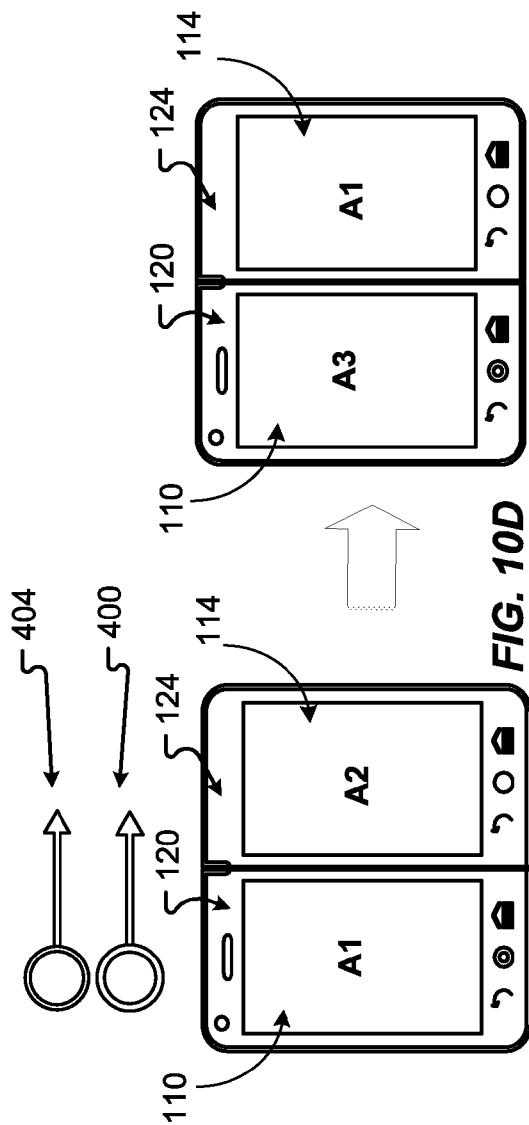
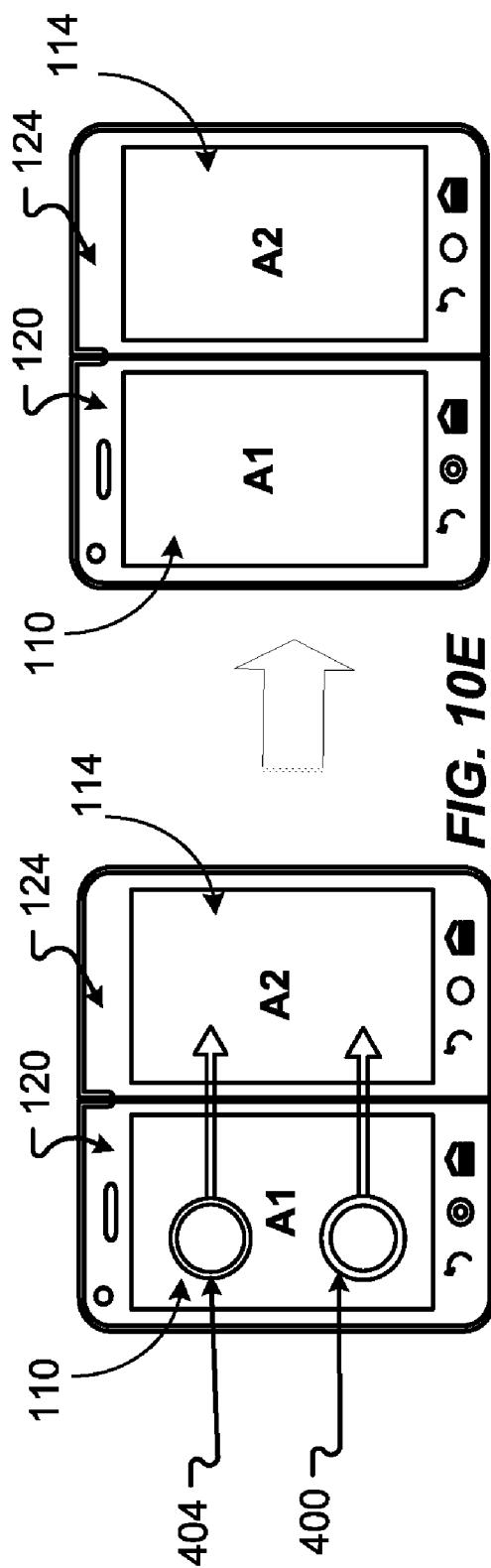


FIG. 10D

**FIG. 10E**

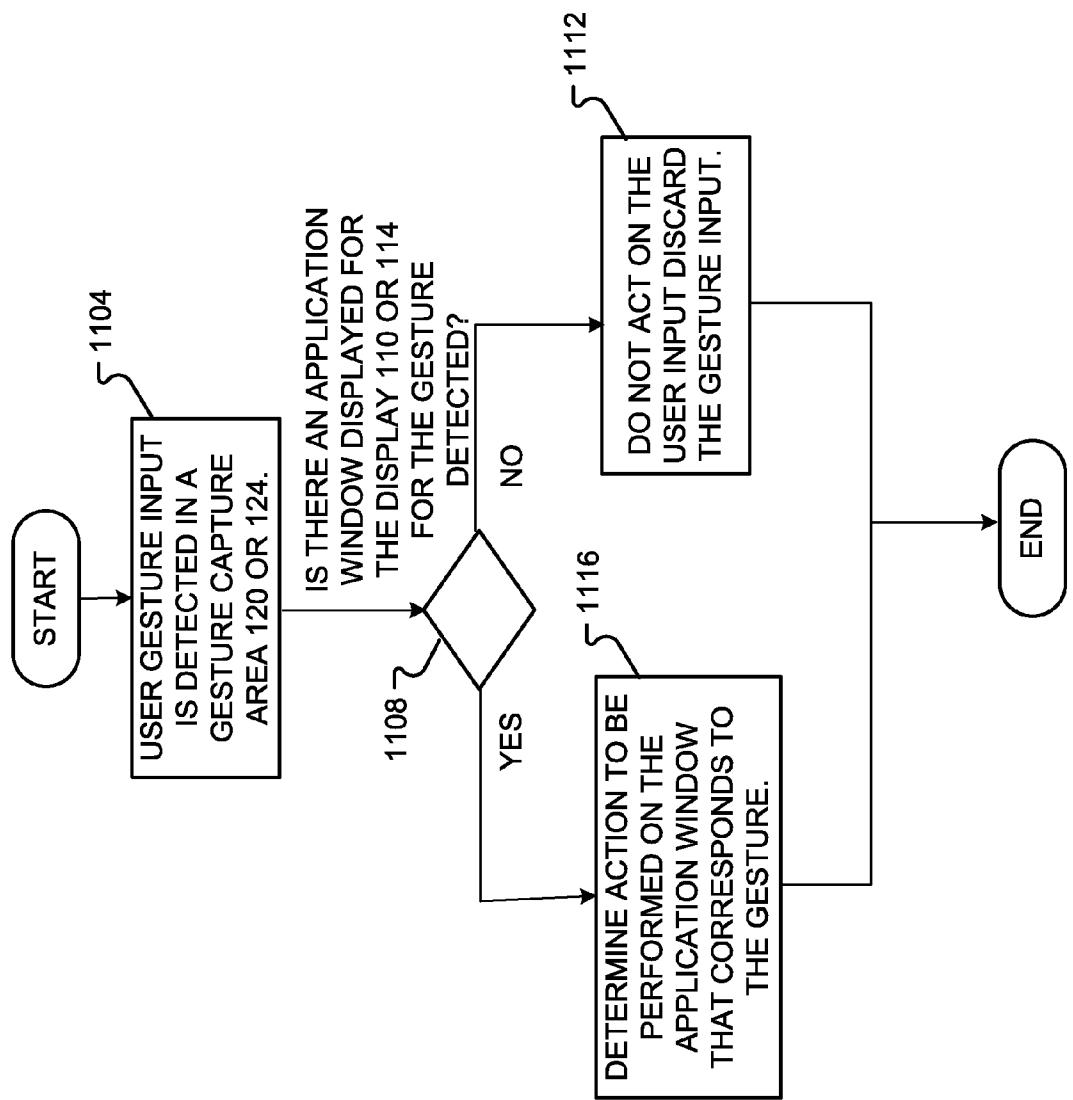


FIG. 11

GESTURE CAPTURE FOR MANIPULATION OF PRESENTATIONS ON ONE OR MORE DEVICE DISPLAYS

CROSS REFERENCE TO RELATED APPLICATION

[0001] The present application claims the benefits of and priority, under 35 U.S.C. §119(e), to U.S. Provisional Application Ser. Nos. 61/389,000, filed Oct. 1, 2010, entitled “DUAL DISPLAY WINDOWING SYSTEM;” 61/389,117, filed Oct. 1, 2010, entitled “MULTI-OPERATING SYSTEM PORTABLE DOCKETING DEVICE;” 61/389,087, filed Oct. 1, 2010, entitled “TABLET COMPUTING USER INTERFACE;” 61/458,150, filed Nov. 17, 2010, entitled “Dual Screen Email Client;” 61/539,884, filed Sep. 27, 2011, entitled “MOBILE DEVICE.” Each of the aforementioned documents is incorporated herein by this reference in their entirety for all that they teach and for all purposes.

BACKGROUND

[0002] A substantial number of handheld computing devices, such as cellular phones, tablets, and E-Readers, make use of a touch screen display not only to deliver display information to the user but also to receive inputs from user interface commands. While touch screen displays may increase the configurability of the handheld device and provide a wide variety of user interface options, this flexibility typically comes at a price. The dual use of the touch screen to provide content and receive user commands, while flexible for the user, may obfuscate the display and cause visual clutter, thereby leading to user frustration and loss of productivity.

[0003] The small form factor of handheld computing devices requires a careful balancing between the displayed graphics and the area provided for receiving inputs. On the one hand, the small display constrains the display space, which may increase the difficulty of interpreting actions or results. On the other, a virtual keypad or other user interface scheme is superimposed on or positioned adjacent to an executing application, requiring the application to be squeezed into an even smaller portion of the display.

[0004] This balancing act is particularly difficult for single display touch screen devices. Single display touch screen devices are crippled by their limited screen space. When users are entering information into the device, through the single display, the ability to interpret information in the display can be severely hampered, particularly when a complex interaction between display and interface is required.

SUMMARY

[0005] There is a need for a dual multi-display handheld computing device that provides for enhanced power and/or versatility compared to conventional single display handheld computing devices. These and other needs are addressed by the various aspects, embodiments, and/or configurations of the present disclosure. Also, while the disclosure is presented in terms of exemplary embodiments, it should be appreciated that individual aspects of the disclosure can be separately claimed. More particularly, there is a need for an intuitive technique for inputting user gestures into a handheld computing device that allows a user to more easily manipulate different types of screen display presentations, such as desktop presentations as opposed to application window presenta-

tions (the terms “desktop” and “application window” being described and distinguished hereinbelow). Since applications may be downloaded from a wide variety of sources and may respond to an application specific set of user gestures, user gesture inputs are neither uniform nor consistent between application windows. Furthermore, since device application windows typically require the user to provide application specific user gesture inputs directly to the displayed application windows, such application specific user inputs prevent the use of a uniform or consistent set of user gesture inputs directly to displayed application windows for performing, e.g., window manipulation tasks such as minimizing, maximizing, and moving application windows from one screen display to another, as well as rearranging in a stack of application windows, activating a display of an application window across multiple screen displays, and/or reducing a display of an application window to a single screen display that is currently displayed across multiple screen displays. Accordingly, user gesture inputs for manipulating application windows according to a consistent or uniform set of window based user gesture inputs has heretofore been problematic. On-the-other-hand, for displays of desktops, since the device typically has control over the operation and manipulation of desktop displays, user gesture inputs for such desktop operation and manipulation can be made or forced to be uniform. Accordingly, the present disclosure describes a handheld device which retains the uniformity that a device typically can impose on the manipulation of desktop displays while also imposing a similar uniformity and consistency on the manipulation of application windows. In one embodiment, the present disclosure is directed to performing the following steps (a) through (h) for configuring a device having at least one display screen for displaying one or more of screen display presentations and for displaying one or more operably different screen display presentations, wherein the screen display presentations are able to be manipulated via user gesture input differently from the operably different screen display presentations:

[0006] (a) receiving a first user gesture input to a gesture capture area associated with the at least one screen, the gesture capture area being separate from the at least one display screen;

[0007] (b) determining whether a first of the one or more of screen display presentations is displayed on the at least one screen;

[0008] (c) when the first screen display presentation is determined to be displayed, perform a step of interpreting the gesture input for determining a predetermined operation to apply to the first screen display presentation;

[0009] (d) when none of the one or more screen display presentations are determined to be displayed in the step of determining, perform a step of preventing the device from using the gesture input to change the at least one display screen;

[0010] (e) receiving a second user gesture input in response to a second user gesture input that includes user contact input directly to the at least one screen;

[0011] (f) second determining whether one of the operably different screen display presentations is displayed on the at least one screen;

[0012] (g) when the one operably different screen display presentation is determined to be displayed on the at least one screen, performing a step of second interpret-

ing the second gesture input for determining a predetermined second operation to apply to the one operably different screen display presentation; and

[0013] (h) when none of the operably different screen display presentations are determined in the step of second determining, perform a step of second preventing the device from using the second gesture input to change the at least one display screen.

[0014] Moreover, is it an aspect of the present disclosure that there is a predetermined collection of a plurality of user gesture input descriptions for recognizing user gesture inputs that are interpretable in the step of interpreting in (c) above, wherein:

[0015] (i) the one or more screen display presentations includes a plurality of application windows,

[0016] (ii) each of a plurality of software applications provide a corresponding one of the plurality of application windows,

[0017] (iii) each of the user gesture input descriptions (UGI) of the predetermined collection is operable for recognizing user gesture inputs for each application window (AW) of the plurality of application windows when: (1) the user gesture input UGI is an instance of the first user gesture input, and (2) the application window AW is an instance of the first screen display presentation for which the user gesture input UGI is an input.

[0018] Note that each of the software applications of the plurality of software applications in (ii) immediately above may be downloaded (via a communications network such as the Internet) to the handheld device from a software vendor that is independent other software vendors from which other of the plurality of software applications can be downloaded.

[0019] The present disclosure can provide a number of advantages depending on the particular aspect, embodiment, and/or configuration. In particular, the present disclosure simplifies a user's experience in providing gesture input to the device. More particularly, for each of the plurality of device displays, the present disclosure provides a corresponding separate area associated therewith for receiving user gesture input, wherein this gesture input (to the separate area) is interpreted as applying to application windows and not to desktop displays. Thus, since user gesture input for manipulating desktops can be input directly to a display presenting the desktop, a user can readily specify that how a displayed application window is to be manipulated by inputting user gestures into the corresponding separate area associated with the display presenting the application window.

[0020] These and other advantages will be apparent from the disclosure.

[0021] The phrases "at least one", "one or more", and "and/or" are open-ended expressions that are both conjunctive and disjunctive in operation. For example, each of the expressions "at least one of A, B and C", "at least one of A, B, or C", "one or more of A, B, and C", "one or more of A, B, or C" and "A, B, and/or C" means A alone, B alone, C alone, A and B together, A and C together, B and C together, or A, B and C together.

[0022] The term "a" or "an" entity refers to one or more of that entity. As such, the terms "a" (or "an"), "one or more" and "at least one" can be used interchangeably herein. It is also to be noted that the terms "comprising", "including", and "having" can be used interchangeably.

[0023] The term "automatic" and variations thereof, as used herein, refers to any process or operation done without

material human input when the process or operation is performed. However, a process or operation can be automatic, even though performance of the process or operation uses material or immaterial human input, if the input is received before performance of the process or operation. Human input is deemed to be material if such input influences how the process or operation will be performed. Human input that consents to the performance of the process or operation is not deemed to be "material".

[0024] The term "computer-readable medium" as used herein refers to any tangible storage and/or transmission medium that participate in providing instructions to a processor for execution. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example, NVRAM, or magnetic or optical disks. Volatile media includes dynamic memory, such as main memory. Common forms of computer-readable media include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, or any other magnetic medium, magneto-optical medium, a CD-ROM, any other optical medium, punch cards, paper tape, any other physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, a solid state medium like a memory card, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read. A digital file attachment to e-mail or other self-contained information archive or set of archives is considered a distribution medium equivalent to a tangible storage medium. When the computer-readable media is configured as a database, it is to be understood that the database may be any type of database, such as relational, hierarchical, object-oriented, and/or the like. Accordingly, the disclosure is considered to include a tangible storage medium or distribution medium and prior art-recognized equivalents and successor media, in which the software implementations of the present disclosure are stored.

[0025] The term "application" as used herein refers to a software program and/or the display therefor on an electronic display screen, wherein the application is computer software designed to help the user to perform one or more predetermined tasks. Examples include enterprise software, accounting software, office suites, graphics software, and media players. An "application" has a user interface for receiving and presenting information from a user. An "application" as used herein typically refers to software (and/or its graphical presentation) wherein: (i) each separately installed additional application (if any) that can be activated from a user activation of the application is interactive with the user for assisting the user in performing a task the user is cognizant of and purposefully activates the application to perform the task, and (ii) the application is not viewed by the user as providing graphical interactions for graphically displaying with a display of the application additional applications such that by user input to the application, the displays of the additional applications can be rearranged relative to the application display, added to the application display, deleted from the application display, and activated from the application display.

[0026] The term "application window" refers to a potentially complex data structure(s) defining a graphical presentation of interrelated information that can be manipulated on a graphical display screen for moving, minimizing, and maximizing as a unit. Each application window has an "application" associated therewith that defines predetermined user input thereto to which the application window is responsive

according to processing performed by the associated application. As referred to herein, an “application window” also refers its graphical presentation when the application window is displayed on an electronic display device such as a “touch screen” as one skilled in the art will appreciate.

[0027] The term “desktop” refers to a metaphor used to portray systems. A desktop is generally considered a “surface” that typically includes pictures, called icons, widgets, folders, etc. that can activate or show applications, windows, cabinets, files, folders, documents, and other graphical items thereon. The icons are generally selectable to initiate a task through user interface interaction to allow a user to execute applications or conduct other operations. More precisely, as used herein a “desktop” refers to a potentially complex data structure for providing graphical interactions for graphically displaying with a display of the desktop a plurality of applications such that by user input to the desktop, the displays of the applications can be rearranged relative to the desktop display, added to the desktop display, deleted from the desktop display, and activated from the desktop display. Each desktop includes or has associated therewith data identifying it as a desktop and not an application and not an application window. Similarly, each application and application window includes or has associated therewith data identifying it has an application or application window and not a desktop.

[0028] The term “screen,” “touch screen,” or “touchscreen” refers to a physical structure that includes one or more hardware components that provide the device with the ability to render a user interface and/or receive user input. A screen can encompass any combination of gesture capture region, a touch sensitive display, and/or a configurable area. The device can have one or more physical screens embedded in the hardware. However a screen may also include an external peripheral device that may be attached and detached from the device. In embodiments, multiple external devices may be attached to the device. Thus, in embodiments, the screen can enable the user to interact with the device by touching areas on the screen and provides information to a user through a display. The touch screen may sense user contact in a number of different ways, such as by a change in an electrical parameter (e.g., resistance or capacitance), acoustic wave variations, infrared radiation proximity detection, light variation detection, and the like. In a resistive touch screen, for example, normally separated conductive and resistive metallic layers in the screen pass an electrical current. When a user touches the screen, the two layers make contact in the contacted location, whereby a change in electrical field is noted and the coordinates of the contacted location calculated. In a capacitive touch screen, a capacitive layer stores electrical charge, which is discharged to the user upon contact with the touch screen, causing a decrease in the charge of the capacitive layer. The decrease is measured, and the contacted location coordinates determined. In a surface acoustic wave touch screen, an acoustic wave is transmitted through the screen, and the acoustic wave is disturbed by user contact. A receiving transducer detects the user contact instance and determines the contacted location coordinates.

[0029] The term “display” refers to a portion of one or more screens used to display the output of a computer to a user. A display may be a single-screen display or a multi-screen display, referred to as a composite display. A composite display can encompass the touch sensitive display of one or more screens. A single physical screen can include multiple displays that are managed as separate logical displays. Thus,

different content can be displayed on the separate displays although part of the same physical screen.

[0030] The term “displayed image” refers to an image produced on the display. A typical displayed image is a window or desktop. The displayed image may occupy all or a portion of the display.

[0031] The term “display orientation” refers to the way in which a rectangular display is oriented by a user for viewing. The two most common types of display orientation are portrait and landscape. In landscape mode, the display is oriented such that the width of the display is greater than the height of the display (such as a 4:3 ratio, which is 4 units wide and 3 units tall, or a 16:9 ratio, which is 16 units wide and 9 units tall). Stated differently, the longer dimension of the display is oriented substantially horizontal in landscape mode while the shorter dimension of the display is oriented substantially vertical. In the portrait mode, by contrast, the display is oriented such that the width of the display is less than the height of the display. Stated differently, the shorter dimension of the display is oriented substantially horizontal in the portrait mode while the longer dimension of the display is oriented substantially vertical.

[0032] The term “composited display” refers to a logical structure that defines a display that can encompass one or more screens. A multi-screen display can be associated with a composite display that encompasses all the screens. The composite display can have different display characteristics based on the various orientations of the device.

[0033] The term “gesture” refers to a user action that expresses an intended idea, action, meaning, result, and/or outcome. The user action can include manipulating a device (e.g., opening or closing a device, changing a device orientation, moving a trackball or wheel, etc.), movement of a body part in relation to the device, movement of an implement or tool in relation to the device, audio inputs, etc. A gesture may be made on a device (such as on the screen) or with the device to interact with the device.

[0034] The term “module” as used herein refers to any known or later developed hardware, software, firmware, artificial intelligence, fuzzy logic, or combination of hardware and software that is capable of performing the functionality associated with that element.

[0035] The term “gesture capture” refers to a sense or otherwise a detection of an instance and/or type of user gesture. The gesture capture can occur in one or more areas of the screen. A gesture region can be on the display, where it may be referred to as a touch sensitive display or off the display where it may be referred to as a gesture capture area.

[0036] A “multi-screen application” refers to an application that is capable of multiple modes. The multi-screen application mode can include, but is not limited to, a single screen mode (where the application is displayed on a single screen) or a composite display mode (where the application is displayed on two or more screens). A multi-screen application can have different layouts optimized for the mode. Thus, the multi-screen application can have different layouts for a single screen or for a composite display that can encompass two or more screens. The different layouts may have different screen/display dimensions and/or configurations on which the user interfaces of the multi-screen applications can be rendered. The different layouts allow the application to optimize the application’s user interface for the type of display, e.g., single screen or multiple screens. In single screen mode, the multi-screen application may present one window pane of

information. In a composite display mode, the multi-screen application may present multiple window panes of information or may provide a larger and a richer presentation because there is more space for the display contents. The multi-screen applications may be designed to adapt dynamically to changes in the device and the mode depending on which display (single or composite) the system assigns to the multi-screen application. In alternative embodiments, the user can use a gesture to request the application transition to a different mode, and, if a display is available for the requested mode, the device can allow the application to move to that display and transition modes.

[0037] A “single-screen application” refers to an application that is capable of single screen mode. Thus, the single-screen application can produce only one window and may not be capable of different modes or different display dimensions. A single-screen application is incapable of the several modes discussed with the multi-screen application.

[0038] The term “window” refers to a, typically rectangular, displayed image on at least part of a display that contains or provides content different from the rest of the screen. The window may obscure the desktop.

[0039] The terms “determine”, “calculate” and “compute,” and variations thereof, as used herein, are used interchangeably and include any type of methodology, process, mathematical operation or technique.

[0040] It shall be understood that the term “means” as used herein shall be given its broadest possible interpretation in accordance with 35 U.S.C., Section 112, Paragraph 6. Accordingly, a claim incorporating the term “means” shall cover all structures, materials, or acts set forth herein, and all of the equivalents thereof. Further, the structures, materials or acts and the equivalents thereof shall include all those described in the summary of the invention, brief description of the drawings, detailed description, abstract, and claims themselves.

[0041] The preceding is a simplified summary of the disclosure to provide an understanding of some aspects of the disclosure. This summary is neither an extensive nor exhaustive overview of the disclosure and its various aspects, embodiments, and/or configurations. It is intended neither to identify key or critical elements of the disclosure nor to delineate the scope of the disclosure but to present selected concepts of the disclosure in a simplified form as an introduction to the more detailed description presented below. As will be appreciated, other aspects, embodiments, and/or configurations of the disclosure are possible utilizing, alone or in combination, one or more of the features set forth above or described in detail below.

BRIEF DESCRIPTION OF THE DRAWINGS

[0042] FIG. 1A includes a first view of an embodiment of a multi-screen user device;

[0043] FIG. 1B includes a second view of an embodiment of a multi-screen user device;

[0044] FIG. 1C includes a third view of an embodiment of a multi-screen user device;

[0045] FIG. 1D includes a fourth view of an embodiment of a multi-screen user device;

[0046] FIG. 1E includes a fifth view of an embodiment of a multi-screen user device;

[0047] FIG. 1F includes a sixth view of an embodiment of a multi-screen user device;

[0048] FIG. 1G includes a seventh view of an embodiment of a multi-screen user device;

[0049] FIG. 1H includes an eighth view of an embodiment of a multi-screen user device;

[0050] FIG. 1I includes a ninth view of an embodiment of a multi-screen user device;

[0051] FIG. 1J includes a tenth view of an embodiment of a multi-screen user device;

[0052] FIG. 2 is a block diagram of an embodiment of the hardware of the device;

[0053] FIG. 3A is a block diagram of an embodiment of the state model for the device based on the device’s orientation and/or configuration;

[0054] FIG. 3B is a table of an embodiment of the state model for the device based on the device’s orientation and/or configuration;

[0055] FIG. 4A is a first representation of an embodiment of user gesture received at a device;

[0056] FIG. 4B is a second representation of an embodiment of user gesture received at a device;

[0057] FIG. 4C is a third representation of an embodiment of user gesture received at a device;

[0058] FIG. 4D is a fourth representation of an embodiment of user gesture received at a device;

[0059] FIG. 4E is a fifth representation of an embodiment of user gesture received at a device;

[0060] FIG. 4F is a sixth representation of an embodiment of user gesture received at a device;

[0061] FIG. 4G is a seventh representation of an embodiment of user gesture received at a device;

[0062] FIG. 4H is an eighth representation of an embodiment of user gesture received at a device;

[0063] FIG. 5A is a block diagram of an embodiment of the device software and/or firmware;

[0064] FIG. 5B is a second block diagram of an embodiment of the device software and/or firmware;

[0065] FIG. 6A is a first representation of an embodiment of a device configuration generated in response to the device state;

[0066] FIG. 6B is a second representation of an embodiment of a device configuration generated in response to the device state;

[0067] FIG. 6C is a third representation of an embodiment of a device configuration generated in response to the device state;

[0068] FIG. 6D is a fourth representation of an embodiment of a device configuration generated in response to the device state;

[0069] FIG. 6E is a fifth representation of an embodiment of a device configuration generated in response to the device state;

[0070] FIG. 6F is a sixth representation of an embodiment of a device configuration generated in response to the device state;

[0071] FIG. 6G is a seventh representation of an embodiment of a device configuration generated in response to the device state;

[0072] FIG. 6H is an eighth representation of an embodiment of a device configuration generated in response to the device state;

[0073] FIG. 6I is a ninth representation of an embodiment of a device configuration generated in response to the device state;

[0074] FIG. 6J is a tenth representation of an embodiment of a device configuration generated in response to the device state;

[0075] FIG. 7A is representation of a logical window stack;

[0076] FIG. 7B is another representation of an embodiment of a logical window stack;

[0077] FIG. 7C is another representation of an embodiment of a logical window stack;

[0078] FIG. 7D is another representation of an embodiment of a logical window stack;

[0079] FIG. 7E is another representation of an embodiment of a logical window stack;

[0080] FIG. 8 is block diagram of an embodiment of a logical data structure for a window stack;

[0081] FIG. 9 is a flow chart of an embodiment of a method for creating a window stack;

[0082] FIG. 10A illustrates processing performed by an embodiment of the Gesture Module 576, wherein a combination flick-drag gesture is applied to the gesture capture area 124, and the Gesture Module 576 ignores the gesture since there are no application windows in the displays 110 and 114 to be operated on according to the gesture;

[0083] FIG. 10B illustrates processing performed by the device 100, wherein a combination flick-drag gesture is applied directly to the desktop of display 114, the desktops are shifted to the left;

[0084] FIG. 10C illustrates processing performed by an embodiment of the Gesture Module 576, wherein a combination flick-drag gesture is applied to the gesture capture area 124, and the Gesture Module 576 since there is an application window in the display 110 to which the gesture can be applied, the Gesture Module 576 shifts the application A1 to the right;

[0085] FIG. 10D is similar to FIG. 10C except that in the window stack for the display 110, there is another application window A3 in the stack which is then displayed in the display 110 when the application window A1 is shifted to the right;

[0086] FIG. 10E illustrates processing performed by an embodiment of the Gesture Module 576, wherein a combination flick-drag gesture is applied to the display 110, and the Gesture Module 576 ignores the gesture since no desktop is displayed in the display 110 to be operated on according to the gesture;

[0087] FIG. 11 is a high level flowchart of the processing performed by the Gesture Module 576 for providing the display changes shown in FIGS. 10A through 10D;

[0088] In the appended figures, similar components and/or features may have the same reference label. Further, various components of the same type may be distinguished by following the reference label by a letter that distinguishes among the similar components. If only the first reference label is used in the specification, the description is applicable to any one of the similar components having the same first reference label irrespective of the second reference label.

DETAILED DESCRIPTION

[0089] Presented herein are embodiments of a device. The device can be a communications device, such as a cellular telephone, or other smart device. The device can include two screens that are oriented to provide several unique display configurations. Further, the device can receive user input in unique ways. The overall design and functionality of the device provides for an enhanced user experience making the device more useful and more efficient.

[0090] Mechanical Features:

[0091] FIGS. 1A-1J illustrate a device 100 in accordance with embodiments of the present disclosure. As described in greater detail below, device 100 can be positioned in a number of different ways each of which provides different functionality to a user. The device 100 is a multi-screen device that includes a primary screen 104 and a secondary screen 108, both of which are touch sensitive. In embodiments, the entire front surface of screens 104 and 108 may be touch sensitive and capable of receiving input by a user touching the front surface of the screens 104 and 108. Primary screen 104 includes touch sensitive display 110, which, in addition to being touch sensitive, also displays information to a user. Secondary screen 108 includes touch sensitive display 114, which also displays information to a user. In other embodiments, screens 104 and 108 may include more than one display area.

[0092] Primary screen 104 also includes a configurable area 112 that has been configured for specific inputs when the user touches portions of the configurable area 112. Secondary screen 108 also includes a configurable area 116 that has been configured for specific inputs. Areas 112a and 116a have been configured to receive a “back” input indicating that a user would like to view information previously displayed. Areas 112b and 116b have been configured to receive a “menu” input indicating that the user would like to view options from a menu. Areas 112c and 116c have been configured to receive a “home” input indicating that the user would like to view information associated with a “home” view. In other embodiments, areas 112a-c and 116a-c may be configured, in addition to the configurations described above, for other types of specific inputs including controlling features of device 100, some non-limiting examples including adjusting overall system power, adjusting the volume, adjusting the brightness, adjusting the vibration, selecting of displayed items (on either of screen 104 or 108), operating a camera, operating a microphone, and initiating/terminating of telephone calls. Also, in some embodiments, areas 112a-C and 116a-C may be configured for specific inputs depending upon the application running on device 100 and/or information displayed on touch sensitive displays 110 and/or 114.

[0093] In addition to touch sensing, primary screen 104 and secondary screen 108 may also include areas that receive input from a user without requiring the user to touch the display area of the screen. For example, primary screen 104 includes gesture capture area 120, and secondary screen 108 includes gesture capture area 124. These gesture capture areas are, e.g., capacitive touch sensitive areas which are able to receive user touch input to be used for recognizing gestures made by a user without the need for the user to actually touch the surface of the display area (i.e., display area 110 or 114). In comparison to touch sensitive displays 110 and 114, the gesture capture areas 120 and 124 are commonly not capable of rendering a displayed image, and in particular, cannot display a desktop or an application window.

[0094] The two screens 104 and 108 are connected together with a hinge 128, shown clearly in FIG. 1C (illustrating a back view of device 100). Hinge 128, in the embodiment shown in FIGS. 1A-1J, is a center hinge that connects screens 104 and 108 so that when the hinge is closed, screens 104 and 108 are juxtaposed (i.e., side-by-side) as shown in FIG. 1B (illustrating a front view of device 100). Hinge 128 can be opened to position the two screens 104 and 108 in different relative positions to each other. As described in greater detail below,

the device 100 may have different functionalities depending on the relative positions of screens 104 and 108.

[0095] FIG. 1D illustrates the right side of device 100. As shown in FIG. 1D, secondary screen 108 also includes a card slot 132 and a port 136 on its side. Card slot 132 in embodiments, accommodates different types of cards including a subscriber identity module (SIM). Port 136 in embodiments is an input/output port (I/O port) that allows device 100 to be connected to other peripheral devices, such as a display, keyboard, or printing device. As can be appreciated, these are merely some examples and in other embodiments device 100 may include other slots and ports such as slots and ports for accommodating additional memory devices and/or for connecting other peripheral devices. Also shown in FIG. 1D is an audio jack 140 that accommodates a tip, ring, sleeve (TRS) connector for example to allow a user to utilize headphones or a headset.

[0096] Device 100 also includes a number of buttons 158. For example, FIG. 1E illustrates the left side of device 100. As shown in FIG. 1E, the side of primary screen 104 includes three buttons 144, 148, and 152, which can be configured for specific inputs. For example, buttons 144, 148, and 152 may be configured to, in combination or alone, control a number of aspects of device 100. Some non-limiting examples include overall system power, volume, brightness, vibration, selection of displayed items (on either of screen 104 or 108), a camera, a microphone, and initiation/termination of telephone calls. In some embodiments, instead of separate buttons two buttons may be combined into a rocker button. This arrangement is useful in situations where the buttons are configured to control features such as volume or brightness. In addition to buttons 144, 148, and 152, device 100 also includes a button 156, shown in FIG. 1F, which illustrates the top of device 100. In one embodiment, button 156 is configured as an on/off button used to control overall system power to device 100. In other embodiments, button 156 is configured to, in addition to or in lieu of controlling system power, control other aspects of device 100. In some embodiments, one or more of the buttons 144, 148, 152, and 156 are capable of supporting different user commands. By way of example, a normal press has a duration commonly of less than about 1 second and resembles a quick tap. A medium press has a duration commonly of 1 second or more but less than about 12 seconds. A long press has a duration commonly of about 12 seconds or more. The function of the buttons is normally specific to the application that is currently in focus on the respective display 110 and 114. In a telephone application for instance and depending on the particular button, a normal, medium, or long press can mean end call, increase in call volume, decrease in call volume, and toggle microphone mute. In a camera or video application for instance and depending on the particular button, a normal, medium, or long press can mean increase zoom, decrease zoom, and take photograph or record video.

[0097] There are also a number of hardware components within device 100. As illustrated in FIG. 1C, device 100 includes a speaker 160 and a microphone 164. Device 100 also includes a camera 168 (FIG. 1B). Additionally, device 100 includes two position sensors 172A and 172B, which are used to determine the relative positions of screens 104 and 108. In one embodiment, position sensors 172A and 172B are Hall effect sensors. However, in other embodiments other sensors can be used in addition to or in lieu of the Hall effect sensors. An accelerometer 176 may also be included as part of

device 100 to determine the orientation of the device 100 and/or the orientation of screens 104 and 108. Additional internal hardware components that may be included in device 100 are described below with respect to FIG. 2.

[0098] The overall design of device 100 allows it to provide additional functionality not available in other communication devices. Some of the functionality is based on the various positions and orientations that device 100 can have. As shown in FIGS. 1B-1G, device 100 can be operated in an “open” position where screens 104 and 108 are juxtaposed. This position allows a large display area for displaying information to a user. When position sensors 172A and 172B determine that device 100 is in the open position, they can generate a signal that can be used to trigger different events such as displaying information on both screens 104 and 108. Additional events may be triggered if accelerometer 176 determines that device 100 is in a portrait position (FIG. 1B) as opposed to a landscape position (not shown).

[0099] In addition to the open position, device 100 may also have a “closed” position illustrated in FIG. 1H. Again, position sensors 172A and 172B can generate a signal indicating that device 100 is in the “closed” position. This can trigger an event that results in a change of displayed information on screen 104 and/or 108. For example, device 100 may be programmed to stop displaying information on one of the screens, e.g., screen 108, since a user can only view one screen at a time when device 100 is in the “closed” position. In other embodiments, the signal generated by position sensors 172A and 172B, indicating that the device 100 is in the “closed” position, can trigger device 100 to answer an incoming telephone call. The “closed” position can also be a preferred position for utilizing the device 100 as a mobile phone.

[0100] Device 100 can also be used in an “easel” position which is illustrated in FIG. 1I. In the “easel” position, screens 104 and 108 are angled with respect to each other and facing outward with the edges of screens 104 and 108 substantially horizontal. In this position, device 100 can be configured to display information on both screens 104 and 108 to allow two users to simultaneously interact with device 100. When device 100 is in the “easel” position, sensors 172A and 172B generate a signal indicating that the screens 104 and 108 are positioned at an angle to each other, and the accelerometer 176 can generate a signal indicating that device 100 has been placed so that the edge of screens 104 and 108 are substantially horizontal. The signals can then be used in combination to generate events that trigger changes in the display of information on screens 104 and 108.

[0101] FIG. 1J illustrates device 100 in a “modified easel” position. In the “modified easel” position, one of screens 104 or 108 is used as a stand and is faced down on the surface of an object such as a table. This position provides a convenient way for information to be displayed to a user in landscape orientation. Similar to the easel position, when device 100 is in the “modified easel” position, position sensors 172A and 172B generate a signal indicating that the screens 104 and 108 are positioned at an angle to each other. The accelerometer 176 would generate a signal indicating that device 100 has been positioned so that one of screens 104 and 108 is faced downward and is substantially horizontal. The signals can then be used to generate events that trigger changes in the display of information of screens 104 and 108. For example, information may not be displayed on the screen that is face down since a user cannot see the screen.

[0102] Transitional states are also possible. When the position sensors 172A and B and/or accelerometer indicate that the screens are being closed or folded (from open), a closing transitional state is recognized. Conversely when the position sensors 172A and B indicate that the screens are being opened or folded (from closed), an opening transitional state is recognized. The closing and opening transitional states are typically time-based, or have a maximum time duration from a sensed starting point. Normally, no user input is possible when one of the closing and opening states is in effect. In this manner, incidental user contact with a screen during the closing or opening function is not misinterpreted as user input. In embodiments, another transitional state is possible when the device 100 is closed. This additional transitional state allows the display to switch from one screen 104 to the second screen 108 when the device 100 is closed based on some user input, e.g., a double tap on the screen 110,114.

[0103] As can be appreciated, the description of device 100 is made for illustrative purposes only, and the embodiments are not limited to the specific mechanical features shown in FIGS. 1A-1J and described above. In other embodiments, device 100 may include additional features, including one or more additional buttons, slots, display areas, hinges, and/or locking mechanisms. Additionally, in embodiments, the features described above may be located in different parts of device 100 and still provide similar functionality. Therefore, FIGS. 1A-1J and the description provided above are nonlimiting.

[0104] Hardware Features:

[0105] FIG. 2 illustrates components of a device 100 in accordance with embodiments of the present disclosure. In general, the device 100 includes a primary screen 104 and a secondary screen 108. While the primary screen 104 and its components are normally enabled in both the opened and closed positions or states, the secondary screen 108 and its components are normally enabled in the opened state but disabled in the closed state. However, even when in the closed state a user or application triggered interrupt (such as in response to a phone application or camera application operation) can flip the active screen, or disable the primary screen 104 and enable the secondary screen 108, by a suitable command. Each screen 104, 108 can be touch sensitive and can include different operative areas. For example, a first operative area, within each touch sensitive screen 104 and 108, may comprise a touch sensitive display 110, 114. In general, the touch sensitive display 110, 114 may comprise a full color, touch sensitive display. A second area within each touch sensitive screen 104 and 108 may comprise a gesture capture region 120, 124. The gesture capture region 120, 124 may comprise an area or region that is outside of the touch sensitive display 110, 114 area, and that is capable of receiving input, for example in the form of gestures provided by a user. However, the gesture capture region 120, 124 does not include pixels that can perform a display function or capability.

[0106] A third region of the touch sensitive screens 104 and 108 may comprise a configurable area 112, 116. The configurable area 112, 116 is capable of receiving input and has display or limited display capabilities. In embodiments, the configurable area 112, 116 may present different input options to the user. For example, the configurable area 112, 116 may display buttons or other relatable items. Moreover, the identity of displayed buttons, or whether any buttons are displayed at all within the configurable area 112, 116 of a

touch sensitive screen 104 or 108, may be determined from the context in which the device 100 is used and/or operated. In an exemplary embodiment, the touch sensitive screens 104 and 108 comprise liquid crystal display devices extending across at least those regions of the touch sensitive screens 104 and 108 that are capable of providing visual output to a user, and a capacitive input matrix over those regions of the touch sensitive screens 104 and 108 that are capable of receiving input from the user.

[0107] One or more display controllers 216a, 216b may be provided for controlling the operation of the touch sensitive screens 104 and 108, including input (touch sensing) and output (display) functions. In the exemplary embodiment illustrated in FIG. 2, a separate touch screen controller 216a or 216b is provided for each touch screen 104 and 108. In accordance with alternate embodiments, a common or shared touch screen controller 216 may be used to control each of the included touch sensitive screens 104 and 108. In accordance with still other embodiments, the functions of a touch screen controller 216 may be incorporated into other components, such as a processor 204.

[0108] The processor 204 may comprise a general purpose programmable processor or controller for executing application programming or instructions. In accordance with at least some embodiments, the processor 204 may include multiple processor cores, and/or implement multiple virtual processors. In accordance with still other embodiments, the processor 204 may include multiple physical processors. As a particular example, the processor 204 may comprise a specially configured application specific integrated circuit (ASIC) or other integrated circuit, a digital signal processor, a controller, a hardwired electronic or logic circuit, a programmable logic device or gate array, a special purpose computer, or the like. The processor 204 generally functions to run programming code or instructions implementing various functions of the device 100.

[0109] A communication device 100 may also include memory 208 for use in connection with the execution of application programming or instructions by the processor 204, and for the temporary or long term storage of program instructions and/or data. As examples, the memory 208 may comprise RAM, DRAM, SDRAM, or other solid state memory. Alternatively or in addition, data storage 212 may be provided. Like the memory 208, the data storage 212 may comprise a solid state memory device or devices. Alternatively or in addition, the data storage 212 may comprise a hard disk drive or other random access memory.

[0110] In support of communications functions or capabilities, the device 100 can include a cellular telephony module 228. As examples, the cellular telephony module 228 can comprise a GSM, CDMA, FDMA and/or analog cellular telephony transceiver capable of supporting voice, multimedia and/or data transfers over a cellular network. Alternatively or in addition, the device 100 can include an additional or other wireless communications module 232. As examples, the other wireless communications module 232 can comprise a Wi-Fi, BLUETOOTH™, WiMax, infrared, or other wireless communications link. The cellular telephony module 228 and the other wireless communications module 232 can each be associated with a shared or a dedicated antenna 224.

[0111] A port interface 252 may be included. The port interface 252 may include proprietary or universal ports to support the interconnection of the device 100 to other devices or components, such as a dock, which may or may not include

additional or different capabilities from those integral to the device 100. In addition to supporting an exchange of communication signals between the device 100 and another device or component, the docking port 136 and/or port interface 252 can support the supply of power to or from the device 100. The port interface 252 also comprises an intelligent element that comprises a docking module for controlling communications or other interactions between the device 100 and a connected device or component.

[0112] An input/output module 248 and associated ports may be included to support communications over wired networks or links, for example with other communication devices, server devices, and/or peripheral devices. Examples of an input/output module 248 include an Ethernet port, a Universal Serial Bus (USB) port, Institute of Electrical and Electronics Engineers (IEEE) 1394, or other interface.

[0113] An audio input/output interface/device(s) 244 can be included to provide analog audio to an interconnected speaker or other device, and to receive analog audio input from a connected microphone or other device. As an example, the audio input/output interface/device(s) 244 may comprise an associated amplifier and analog to digital converter. Alternatively or in addition, the device 100 can include an integrated audio input/output device 256 and/or an audio jack for interconnecting an external speaker or microphone. For example, an integrated speaker and an integrated microphone can be provided, to support near talk or speaker phone operations.

[0114] Hardware buttons 158 can be included for example for use in connection with certain control operations. Examples include a master power switch, volume control, etc., as described in conjunction with FIGS. 1A through 1J. One or more image capture interfaces/devices 240, such as a camera, can be included for capturing still and/or video images. Alternatively or in addition, an image capture interface/device 240 can include a scanner or code reader. An image capture interface/device 240 can include or be associated with additional elements, such as a flash or other light source.

[0115] The device 100 can also include a global positioning system (GPS) receiver 236. In accordance with embodiments of the present invention, the GPS receiver 236 may further comprise a GPS module that is capable of providing absolute location information to other components of the device 100. An accelerometer(s) 176 may also be included. For example, in connection with the display of information to a user and/or other functions, a signal from the accelerometer 176 can be used to determine an orientation and/or format in which to display that information to the user.

[0116] Embodiments of the present invention can also include one or more position sensor(s) 172. The position sensor 172 can provide a signal indicating the position of the touch sensitive screens 104 and 108 relative to one another. This information can be provided as an input, for example to a user interface application, to determine an operating mode, characteristics of the touch sensitive displays 110, 114, and/or other device 100 operations. As examples, a screen position sensor 172 can comprise a series of Hall effect sensors, a multiple position switch, an optical switch, a Wheatstone bridge, a potentiometer, or other arrangement capable of providing a signal indicating of multiple relative positions the touch screens are in.

[0117] Communications between various components of the device 100 can be carried by one or more buses 222. In

addition, power can be supplied to the components of the device 100 from a power source and/or power control module 260. The power control module 260 can, for example, include a battery, an AC to DC converter, power control logic, and/or ports for interconnecting the device 100 to an external source of power.

[0118] Device State:

[0119] FIGS. 3A and 3B represent illustrative states of device 100. While a number of illustrative states are shown, and transitions from a first state to a second state, it is to be appreciated that the illustrative state diagram may not encompass all possible states and/or all possible transitions from a first state to a second state. As illustrated in FIG. 3, the various arrows between the states (illustrated by the state represented in the circle) represent a physical change that occurs to the device 100, that is detected by one or more of hardware and software, the detection triggering one or more of a hardware and/or software interrupt that is used to control and/or manage one or more functions of device 100.

[0120] As illustrated in FIG. 3A, there are twelve exemplary “physical” states: closed 304, transition 308 (or opening transitional state), easel 312, modified easel 316, open 320, inbound/outbound call or communication 324, image/video capture 328, transition 332 (or closing transitional state), landscape 340, docked 336, docked 344 and landscape 348. Next to each illustrative state is a representation of the physical state of the device 100 with the exception of states 324 and 328, where the state is generally symbolized by the international icon for a telephone and the icon for a camera, respectfully.

[0121] In state 304, the device is in a closed state with the device 100 generally oriented in the portrait direction with the primary screen 104 and the secondary screen 108 back-to-back in different planes (see FIG. 1H). From the closed state, the device 100 can enter, for example, docked state 336, where the device 100 is coupled with a docking station, docking cable, or in general docked or associated with one or more other devices or peripherals, or the landscape state 340, where the device 100 is generally oriented with the primary screen 104 facing the user, and the primary screen 104 and the secondary screen 108 being back-to-back.

[0122] In the closed state, the device can also move to a transitional state where the device remains closed but the display is moved from one screen 104 to another screen 108 based on a user input, e.g., a double tap on the screen 110, 114. Still another embodiment includes a bilateral state. In the bilateral state, the device remains closed, but a single application displays at least one window on both the first display 110 and the second display 114. The windows shown on the first and second display 110, 114 may be the same or different based on the application and the state of that application. For example, while acquiring an image with a camera, the device may display the view finder on the first display 110 and displays a preview for the photo subjects (full screen and mirrored left-to-right) on the second display 114.

[0123] In state 308, a transition state from the closed state 304 to the semi-open state or easel state 312, the device 100 is shown opening with the primary screen 104 and the secondary screen 108 being rotated around a point of axis coincidence with the hinge. Upon entering the easel state 312, the primary screen 104 and the secondary screen 108 are separated from one another such that, for example, the device 100 can sit in an easel-like configuration on a surface.

[0124] In state 316, known as the modified easel position, the device 100 has the primary screen 104 and the secondary screen 108 in a similar relative relationship to one another as in the easel state 312, with the difference being one of the primary screen 104 or the secondary screen 108 are placed on a surface as shown.

[0125] State 320 is the open state where the primary screen 104 and the secondary screen 108 are generally on the same plane. From the open state, the device 100 can transition to the docked state 344 or the open landscape state 348. In the open state 320, the primary screen 104 and the secondary screen 108 are generally in the portrait-like orientation while in landscaped state 348 the primary screen 104 and the secondary screen 108 are generally in a landscape-like orientation.

[0126] State 324 is illustrative of a communication state, such as when an inbound or outbound call is being received or placed, respectively, by the device 100. While not illustrated for clarity, it should be appreciated the device 100 can transition to the inbound/outbound call state 324 from any state illustrated in FIG. 3. In a similar manner, the image/video capture state 328 can be entered into from any other state in FIG. 3, with the image/video capture state 328 allowing the device 100 to take one or more images via a camera and/or videos with a video capture device 240.

[0127] Transition state 322 illustratively shows primary screen 104 and the secondary screen 108 being closed upon one another for entry into, for example, the closed state 304.

[0128] FIG. 3B illustrates, with reference to the key, the inputs that are received to detect a transition from a first state to a second state. In FIG. 3B, various combinations of states are shown with in general, a portion of the columns being directed toward a portrait state 352, a landscape state 356, and a portion of the rows being directed to portrait state 360 and landscape state 364.

[0129] In FIG. 3B, the Key indicates that "H" represents an input from one or more Hall Effect sensors, "A" represents an input from one or more accelerometers, "T" represents an input from a timer, "P" represents a communications trigger input and "I" represents an image and/or video capture request input. Thus, in the center portion 376 of the chart, an input, or combination of inputs, are shown that represent how the device 100 detects a transition from a first physical state to a second physical state.

[0130] As discussed, in the center portion of the chart 376, the inputs that are received enable the detection of a transition from, for example, a portrait open state to a landscape easel state—shown in bold—"HAT." For this exemplary transition from the portrait open to the landscape easel state, a Hall Effect sensor ("H"), an accelerometer ("A") and a timer ("T") input may be needed. The timer input can be derived from, for example, a clock associated with the processor.

[0131] In addition to the portrait and landscape states, a docked state 368 is also shown that is triggered based on the receipt of a docking signal 372. As discussed above and in relation to FIG. 3, the docking signal can be triggered by the association of the device 100 with one or more other device 100s, accessories, peripherals, smart docks, or the like.

[0132] User Interaction:

[0133] FIGS. 4A through 4H depict various graphical representations of gesture inputs that may be recognized by the screens 104, 108. The gestures may be performed not only by a user's body part, such as a digit, but also by other devices, such as a stylus, that may be sensed by the contact sensing portion(s) of a screen 104, 108. In general, gestures are inter-

preted differently, based on where the gestures are performed (either directly on the display 110, 114 or in the gesture capture region 120, 124). For example, gestures in the display 110, 114 may be directed to a desktop or application, and gestures in the gesture capture region 120, 124 may be interpreted as for the system.

[0134] With reference to FIGS. 4A-4H, a first type of gesture, a touch gesture 420, is substantially stationary on the screen 104, 108 for a selected length of time. A circle 428 represents a touch or other contact type received at particular location of a contact sensing portion of the screen. The circle 428 may include a border 432, the thickness of which indicates a length of time that the contact is held substantially stationary at the contact location. For instance, a tap 420 (or short press) has a thinner border 432a than the border 432b for a long press 424 (or for a normal press). The long press 424 may involve a contact that remains substantially stationary on the screen for longer time period than that of a tap 420. As will be appreciated, differently defined gestures may be registered depending upon the length of time that the touch remains stationary prior to contact cessation or movement on the screen.

[0135] With reference to FIG. 4C, a drag gesture 400 on the screen 104, 108 is an initial contact (represented by circle 428) with contact movement 436 in a selected direction. The initial contact 428 may remain stationary on the screen 104, 108 for a certain amount of time represented by the border 432. The drag gesture typically requires the user to contact an icon, window, or other displayed image at a first location followed by movement of the contact in a drag direction to a new second location desired for the selected displayed image. The contact movement need not be in a straight line but have any path of movement so long as the contact is substantially continuous from the first to the second locations.

[0136] With reference to FIG. 4D, a flick gesture 404 on the screen 104, 108 is an initial contact (represented by circle 428) with truncated contact movement 436 (relative to a drag gesture) in a selected direction. In embodiments, a flick has a higher exit velocity for the last movement in the gesture compared to the drag gesture. The flick gesture can, for instance, be a finger snap following initial contact. Compared to a drag gesture, a flick gesture generally does not require continual contact with the screen 104, 108 from the first location of a displayed image to a predetermined second location. The contacted displayed image is moved by the flick gesture in the direction of the flick gesture to the predetermined second location. Although both gestures commonly can move a displayed image from a first location to a second location, the temporal duration and distance of travel of the contact on the screen is generally less for a flick than for a drag gesture.

[0137] With reference to FIG. 4E, a pinch gesture 408 on the screen 104, 108 is depicted. The pinch gesture 408 may be initiated by a first contact 428 to the screen 104, 108 by, for example, a first digit and a second contact 428b to the screen 104, 108 by, for example, a second digit. The first and second contacts 428a, b may be detected by a common contact sensing portion of a common screen 104, 108, by different contact sensing portions of a common screen 104 or 108, or by different contact sensing portions of different screens. The first contact 428a is held for a first amount of time, as represented by the border 432a, and the second contact 428b is held for a second amount of time, as represented by the border 432b. The first and second amounts of time are generally substantially the same, and the first and second contacts 428

a, b generally occur substantially simultaneously. The first and second contacts **428 a, b** generally also include corresponding first and second contact movements **436 a, b**, respectively. The first and second contact movements **436 a, b** are generally in opposing directions. Stated another way, the first contact movement **436a** is towards the second contact **436b**, and the second contact movement **436b** is towards the first contact **436a**. More simply stated, the pinch gesture **408** may be accomplished by a user's digits touching the screen **104,108** in a pinching motion.

[0138] With reference to FIG. 4F, a spread gesture **410** on the screen **104,108** is depicted. The spread gesture **410** may be initiated by a first contact **428a** to the screen **104,108** by, for example, a first digit and a second contact **428b** to the screen **104,108** by, for example, a second digit. The first and second contacts **428a,b** may be detected by a common contact sensing portion of a common screen **104,108**, by different contact sensing portions of a common screen **104,108**, or by different contact sensing portions of different screens. The first contact **428a** is held for a first amount of time, as represented by the border **432a**, and the second contact **428b** is held for a second amount of time, as represented by the border **432b**. The first and second amounts of time are generally substantially the same, and the first and second contacts **428 a, b** generally occur substantially simultaneously. The first and second contacts **428 a, b** generally also include corresponding first and second contact movements **436a, b**, respectively. The first and second contact movements **436 a, b** are generally in a common direction. Stated another way, the first and second contact movements **436 a, b** are away from the first and second contacts **428a, b**. More simply stated, the spread gesture **410** may be accomplished by a user's digits touching the screen **104,108** in a spreading motion.

[0139] The above gestures may be combined in any manner, such as those shown by FIGS. 4G and 4H, to produce a determined functional result. For example, in FIG. 4G a tap gesture **420** is combined with a drag or flick gesture **412** in a direction away from the tap gesture **420**. In FIG. 4H, a tap gesture **420** is combined with a drag or flick gesture **412** in a direction towards the tap gesture **420**.

[0140] The functional result of receiving a gesture can vary depending on a number of factors, including a state of the device **100**, display **110, 114**, or screen **104, 108**, a context associated with the gesture, or sensed location of the gesture. The state of the device commonly refers to one or more of a configuration of the device **100**, a display orientation, and user and other inputs received by the device **100**. Context commonly refers to one or more of the particular application (s) selected by the gesture and the portion(s) of the application currently executing, whether the application is a single- or multi-screen application, and whether the application is a multi-screen application displaying one or more windows in one or more screens or in one or more stacks. Sensed location of the gesture commonly refers to whether the sensed set(s) of gesture location coordinates are on a touch sensitive display **110, 114** or a gesture capture region **120, 124**, whether the sensed set(s) of gesture location coordinates are associated with a common or different display or screen **104,108**, and/or what portion of the gesture capture region contains the sensed set(s) of gesture location coordinates.

[0141] A tap, when received by an touch sensitive display **110, 114**, can be used, for instance, to select an icon to initiate or terminate execution of a corresponding application, to maximize or minimize a window, to reorder windows in a

stack, and to provide user input such as by keyboard display or other displayed image. A drag, when received by a touch sensitive display **110, 114**, can be used, for instance, to relocate an icon or window to a desired location within a display, to reorder a stack on a display, or to span both displays (such that the selected window occupies a portion of each display simultaneously). A flick, when received by a touch sensitive display **110, 114** or a gesture capture region **120, 124**, can be used to relocate a window from a first display to a second display or to span both displays (such that the selected window occupies a portion of each display simultaneously). Unlike the drag gesture, however, the flick gesture is generally not used to move the displayed image to a specific user-selected location but to a default location that is not configurable by the user.

[0142] The spread gesture, when received by a touch sensitive display **110, 114** or a gesture capture region **120, 124**, can be used to minimize or otherwise increase the displayed area or size of a window (typically when received entirely by a common display), to switch windows displayed at the top of the stack on each display to the top of the stack of the other display (typically when received by different displays or screens), or to display an application manager (a "pop-up window" that displays the windows in the stack). The pinch gesture, when received by a touch sensitive display **110, 114** or a gesture capture region **120, 124**, can be used to maximize or otherwise decrease the displayed area or size of a window, to switch windows displayed at the top of the stack on each display to the top of the stack of the other display (typically when received by different displays or screens), or to display an application manager (typically when received by an off-screen gesture capture region on the same or different screens).

[0143] The combined gestures of FIG. 4G, when received by a common display capture region in a common display or screen **104,108**, can be used to hold a first window stack location in a first stack constant for a display receiving the gesture while reordering a second window stack location in a second window stack to include a window in the display receiving the gesture. The combined gestures of FIG. 4H, when received by different display capture regions in a common display or screen **104,108** or in different displays or screens, can be used to hold a first window stack location in a first window stack constant for a display receiving the tap part of the gesture while reordering a second window stack location in a second window stack to include a window in the display receiving the flick or drag gesture. Although specific gestures and gesture capture regions in the preceding examples have been associated with corresponding sets of functional results, it is to be appreciated that these associations can be redefined in any manner to produce differing associations between gestures and/or gesture capture regions and/or functional results.

[0144] Firmware and Software:

[0145] The memory **508** may store and the processor **504** may execute one or more software components. These components can include at least one operating system (OS) **516**, an application manager **562**, a desktop **566**, and/or one or more applications **564a** and/or **564b** from an application store **560**. The OS **516** can include a framework **520**, one or more frame buffers **548**, one or more drivers **512**, previously described in conjunction with FIG. 2, and/or a kernel **518**. The OS **516** can be any software, consisting of programs and data, which manages computer hardware resources and provides

common services for the execution of various applications **564**. The OS **516** can be any operating system and, at least in some embodiments, dedicated to mobile devices, including, but not limited to, Linux, ANDROID™, iPhone OS (IOS™), WINDOWS PHONE 7™, etc. The OS **516** is operable to provide functionality to the phone by executing one or more operations, as described herein.

[0146] The applications **564** can be any higher level software that executes particular functionality for the user. Applications **564** can include programs such as email clients, web browsers, texting applications, games, media players, office suites, etc. The applications **564** can be stored in an application store **560**, which may represent any memory or data storage, and the management software associated therewith, for storing the applications **564**. Once executed, the applications **564** may be run in a different area of memory **508**.

[0147] The framework **520** may be any software or data that allows the multiple tasks running on the device to interact. In embodiments, at least portions of the framework **520** and the discrete components described hereinafter may be considered part of the OS **516** or an application **564**. However, these portions will be described as part of the framework **520**, but those components are not so limited. The framework **520** can include, but is not limited to, a Multi-Display Management (MDM) module **524**, a Surface Cache module **528**, a Window Management module **532**, an Input Management module **536**, a Task Management module **540**, an Application Model Manager **542**, a Display Controller, one or more frame buffers **548**, a task stack **552**, one or more window stacks **550** (which is a logical arrangement of windows and/or desktops in a display area), and/or an event buffer **556**.

[0148] The MDM module **524** includes one or more modules that are operable to manage the display of applications or other data on the screens of the device. An embodiment of the MDM module **524** is described in conjunction with FIG. 5B. In embodiments, the MDM module **524** receives inputs from the other OS **516** components, such as, the drivers **512**, and from the applications **564** to determine continually the state of the device **100**. The inputs assist the MDM module **524** in determining how to configure and allocate the displays according to the application's preferences and requirements, and the user's actions. Once a determination for display configurations is made, the MDM module **524** can bind the applications **564** to a display. The configuration may then be provided to one or more other components to generate a window with a display.

[0149] The Surface Cache module **528** includes any memory or storage and the software associated therewith to store or cache one or more images of windows. A series of active and/or non-active windows (or other display objects, such as, a desktop display) can be associated with each display. An active window (or other display object) is currently displayed. A non-active windows (or other display objects) were opened and, at some time, displayed but are now not displayed. To enhance the user experience, before a window transitions from an active state to an inactive state, a "screen shot" of a last generated image of the window (or other display object) can be stored. The Surface Cache module **528** may be operable to store a bitmap of the last active image of a window (or other display object) not currently displayed. Thus, the Surface Cache module **528** stores the images of non-active windows (or other display objects) in a data store.

[0150] In embodiments, the Window Management module **532** is operable to manage the windows (or other display

objects) that are active or not active on each of the displays. The Window Management module **532**, based on information from the MDM module **524**, the OS **516**, or other components, determines when a window (or other display object) is visible or not active. The Window Management module **532** may then put a non-visible window (or other display object) in a "not active state" and, in conjunction with the Task Management module Task Management **540** suspends the application's operation. Further, the Window Management module **532** may assign, through collaborative interaction with the MDM module **524**, a display identifier to the window (or other display object) or manage one or more other items of data associated with the window (or other display object). The Window Management module **532** may also provide the stored information to the application **564**, the Task Management module **540**, or other components interacting with or associated with the window (or other display object). The Window Management module **532** can also associate an input task with a window based on window focus and display coordinates within the motion space.

[0151] The Input Management module **536** is operable to manage events that occur with the device. An event is any input into the window environment, for example, a user interface interactions with a user. The Input Management module **536** receives the events and logically stores the events in an event buffer **556**. Events can include such user interface interactions as a "down event," which occurs when a screen **104**, **108** receives a touch signal from a user, a "move event," which occurs when the screen **104**, **108** determines that a user's finger is moving across a screen(s), an "up event," which occurs when the screen **104**, **108** determines that the user has stopped touching the screen **104**, **108**, etc. These events are received, stored, and forwarded to other modules by the Input Management module **536**. The Input Management module **536** may also map screen inputs to a motion space which is the culmination of all physical and virtual display available on the device.

[0152] The motion space is a virtualized space that includes all touch sensitive displays **110**, **114** "tiled" together to mimic the physical dimensions of the device **100**. For example, when the device **100** is unfolded, the motion space size may be 960×800, which may be the number of pixels in the combined display area for both touch sensitive displays **110**, **114**. If a user touches on a first touch sensitive display **110** on location **(40, 40)**, a full screen window can receive touch event with location **(40, 40)**. If a user touches on a second touch sensitive display **114**, with location **(40, 40)**, the full screen window can receive touch event with location **(520, 40)**, because the second touch sensitive display **114** is on the right side of the first touch sensitive display **110**, so the device **100** can offset the touch by the first touch sensitive display's **110** width, which is 480 pixels. When a hardware event occurs with location info from a driver **512**, the framework **520** can up-scale the physical location to the motion space because the location of the event may be different based on the device orientation and state. The motion space may be as described in U.S. patent application Ser. No. 13/187,026, filed Jul. 20, 2011, entitled "Systems and Methods for Receiving Gesture Inputs Spanning Multiple Input Devices," which is hereby incorporated by reference in its entirety for all that it teaches and for all purposes.

[0153] A task can be an application and a sub-task can be an application component that provides a window with which users can interact to do something, such as dial the phone,

take a photo, send an email, or view a map. Each task may be given a window in which to draw a user interface. The window typically fills a display (for example, touch sensitive display 110,114), but may be smaller than the display 110, 114 and float on top of other windows. An application usually consists of multiple sub-tasks that are loosely bound to each other. Typically, one task in an application is specified as the “main” task, which is presented to the user when launching the application for the first time. Each task can then start another task or sub-task to perform different actions.

[0154] The Task Management module 540 is operable to manage the operation of one or more applications 564 that may be executed by the device. Thus, the Task Management module 540 can receive signals to launch, suspend, terminate, etc. an application or application sub-tasks stored in the application store 560. The Task Management module 540 may then instantiate one or more tasks or sub-tasks of the application 564 to begin operation of the application 564. Further, the Task Management Module 540 may launch, suspend, or terminate a task or sub-task as a result of user input or as a result of a signal from a collaborating framework 520 component. The Task Management Module 540 is responsible for managing the lifecycle of applications (tasks and sub-task) from when the application is launched to when the application is terminated.

[0155] The processing of the Task Management Module 540 is facilitated by a task stack 552, which is a logical structure associated with the Task Management Module 540. The task stack 552 maintains the state of all tasks and sub-tasks on the device 100. When some component of the operating system 516 requires a task or sub-task to transition in its lifecycle, the OS 516 component can notify the Task Management Module 540. The Task Management Module 540 may then locate the task or sub-task, using identification information, in the task stack 552, and send a signal to the task or sub-task indicating what kind of lifecycle transition the task needs to execute. Informing the task or sub-task of the transition allows the task or sub-task to prepare for the lifecycle state transition. The Task Management Module 540 can then execute the state transition for the task or sub-task. In embodiments, the state transition may entail triggering the OS kernel 518 to terminate the task when termination is required.

[0156] Further, the Task Management module 540 may suspend the application 564 based on information from the Window Management Module 532. Suspending the application 564 may maintain application data in memory but may limit or stop the application 564 from rendering a window or user interface. Once the application becomes active again, the Task Management module 540 can again trigger the application to render its user interface. In embodiments, if a task is suspended, the task may save the task’s state in case the task is terminated. In the suspended state, the application task may not receive input because the application window is not visible to the user.

[0157] The frame buffer 548 is a logical structure(s) used to render the user interface. The frame buffer 548 can be created and destroyed by the OS kernel 518. However, the Display Controller 544 can write the image data, for the visible windows, into the frame buffer 548. A frame buffer 548 can be associated with one screen or multiple screens. The association of a frame buffer 548 with a screen can be controlled dynamically by interaction with the OS kernel 518. A composite display may be created by associating multiple screens

with a single frame buffer 548. Graphical data used to render an application’s window user interface may then be written to the single frame buffer 548, for the composite display, which is output to the multiple screens 104,108. The Display Controller 544 can direct an application’s user interface to a portion of the frame buffer 548 that is mapped to a particular display 110,114, thus, displaying the user interface on only one screen 104 or 108. The Display Controller 544 can extend the control over user interfaces to multiple applications, controlling the user interfaces for as many displays as are associated with a frame buffer 548 or a portion thereof. This approach compensates for the multiple physical screens 104, 108 that are in use by the software component above the Display Controller 544.

[0158] The Application Manager 562 is an application that provides a presentation layer for the window environment. Thus, the Application Manager 562 provides the graphical model for rendering by the Task Management Module 540. Likewise, the Desktop 566 provides the presentation layer for the Application Store 560. Thus, the desktop provides a graphical model of a surface having selectable application icons for the Applications 564 in the Application Store 560 that can be provided to the Window Management Module 556 for rendering.

[0159] Further, the framework can include an Application Model Manager (AMM) 542. The Application Manager 562 may interface with the AMM 542. In embodiments, the AMM 542 receives state change information from the device 100 regarding the state of applications (which are running or suspended). The AMM 542 can associate bit map images from the Surface Cache Module 528 to the tasks that are alive (running or suspended). Further, the AMM 542 can convert the logical window stack maintained in the Task Manager Module 540 to a linear (“film strip” or “deck of cards”) organization that the user perceives when the using the off gesture capture area 120 to sort through the windows. Further, the AMM 542 may provide a list of executing applications to the Application Manager 562.

[0160] An embodiment of the MDM module 524 is shown in FIG. 5B. The MDM module 524 is operable to determine the state of the environment for the device, including, but not limited to, the orientation of the device, whether the device 100 is opened or closed, what applications 564 are executing, how the applications 564 are to be displayed, what actions the user is conducting, the tasks being displayed, etc. To configure the display, the MDM module 524 interprets these environmental factors and determines a display configuration, as described in conjunction with FIGS. 6A-6J. Then, the MDM module 524 can bind the applications 564 or other device components to the displays. The configuration may then be sent to the Display Controller 544 and/or the other components within the OS 516 to generate the display. The MDM module 524 can include one or more of, but is not limited to, a Display Configuration Module 568, a Preferences Module 572, a Device State Module 574, a Gesture Module 576, a Requirements Module 580, an Event Module 584, and/or a Binding Module 588.

[0161] The Display Configuration Module 568 determines the layout for the display. In embodiments, the Display Configuration Module 568 can determine the environmental factors. The environmental factors may be received from one or more other MDM modules 524 or from other sources. The Display Configuration Module 568 can then determine from the list of factors the best configuration for the display. Some

embodiments of the possible configurations and the factors associated therewith are described in conjunction with FIGS. 6A-6F.

[0162] The Preferences Module **572** is operable to determine display preferences for an application **564** or other component. For example, an application can have a preference for Single or Dual displays. The Preferences Module **572** can determine an application's display preference (e.g., by inspecting the application's preference settings) and may allow the application **564** to change to a mode (e.g., single screen, dual screen, max, etc.) if the device **100** is in a state that can accommodate the preferred mode. However, some user interface policies may disallow a mode even if the mode is available. As the configuration of the device changes, the preferences may be reviewed to determine if a better display configuration can be achieved for an application **564**.

[0163] The Device State Module **574** is operable to determine or receive the state of the device. The state of the device can be as described in conjunction with FIGS. 3A and 3B. The state of the device can be used by the Display Configuration Module **568** to determine the configuration for the display. As such, the Device State Module **574** may receive inputs and interpret the state of the device. The state information is then provided to the Display Configuration Module **568**.

[0164] The Gesture Module **576** is shown as part of the MDM module **524**, but, in embodiments, the Gesture module **576** may be a separate Framework **520** component that is separate from the MDM module **524**. In embodiments, the Gesture Module **576** is operable to determine if the user is conducting any actions on any part of the user interface. In alternative embodiments, the Gesture Module **576** receives user interface actions from the configurable area **112,116** only. The Gesture Module **576** can receive touch events that occur on the configurable area **112,116** (or possibly other user interface areas) by way of the Input Management Module **536** and may interpret the touch events (using direction, speed, distance, duration, and various other parameters) to determine what kind of gesture the user is performing. When a gesture is interpreted, the Gesture Module **576** can initiate the processing of the gesture and, by collaborating with other Framework **520** components, can manage the required window animation. The Gesture Module **576** collaborates with the Application Model Manager **542** to collect state information with respect to which applications are running (active or paused) and the order in which applications must appear when a user gesture is performed. The Gesture Module **576** may also receive references to bitmaps (from the Surface Cache Module **528**) and live windows so that when a gesture occurs it can instruct the Display Controller **544** how to move the window(s) across the display **110,114**. Thus, suspended applications may appear to be running when those windows are moved across the display **110,114**.

[0165] Further, the Gesture Module **576** can receive task information either from the Task Management Module **540** or the Input Management module **536**. The gestures may be as defined in conjunction with FIGS. 4A through 4H. For example, moving a window causes the display to render a series of display frames that illustrate the window moving. The gesture associated with such user interface interaction can be received and interpreted by the Gesture Module **576**. The information about the user gesture is then sent to the Task Management Module **540** to modify the display binding of the task.

[0166] The Requirements Module **580**, similar to the Preferences Module **572**, is operable to determine display requirements for an application **564** or other component. An application can have a set display requirement that must be observed. Some applications require a particular display orientation. For example, the application "Angry Birds" can only be displayed in landscape orientation. This type of display requirement can be determined or received, by the Requirements Module **580**. As the orientation of the device changes, the Requirements Module **580** can reassert the display requirements for the application **564**. The Display Configuration Module **568** can generate a display configuration that is in accordance with the application display requirements, as provided by the Requirements Module **580**.

[0167] The Event Module **584**, similar to the Gesture Module **576**, is operable to determine one or more events occurring with an application or other component that can affect the user interface. Thus, the Event Module **584** can receive event information either from the event buffer **556** or the Task Management module **540**. These events can change how the tasks are bound to the displays. The Event Module **584** can collect state change information from other Framework **520** components and act upon that state change information. In an example, an email application receiving an email can cause the display to render the new message in a secondary screen. The events associated with such application execution can be received and interpreted by the Event Module **584**. The information about the events then may be sent to the Display Configuration Module **568** to modify the configuration of the display.

[0168] The Binding Module **588** is operable to bind the applications **564** or the other components to the configuration determined by the Display Configuration Module **568**. A binding associates, in memory, the display configuration for each application with the display and mode of the application. Thus, the Binding Module **588** can associate an application with a display configuration for the application (e.g. landscape, portrait, multi-screen, etc.). Then, the Binding Module **588** may assign a display identifier to the display. The display identifier associated the application with a particular display of the device **100**. This binding is then stored and provided to the Display Controller **544**, the other components of the OS **516**, or other components to properly render the display. The binding is dynamic and can change or be updated based on configuration changes associated with events, gestures, state changes, application preferences or requirements, etc.

[0169] User Interface Configurations:

[0170] With reference now to FIGS. 6A-J, various types of output configurations made possible by the device **100** will be described hereinafter.

[0171] FIGS. 6A and 6B depict two different output configurations of the device **100** being in a first state. Specifically, FIG. 6A depicts the device **100** being in a closed portrait state **304** where the data is displayed on the primary screen **104**. In this example, the device **100** displays data via the touch sensitive display **110** in a first portrait configuration **604**. As can be appreciated, the first portrait configuration **604** may only display a desktop or operating system home screen. Alternatively, one or more windows may be presented in a portrait orientation while the device **100** is displaying data in the first portrait configuration **604**.

[0172] FIG. 6B depicts the device **100** still being in the closed portrait state **304**, but instead data is displayed on the

secondary screen **108**. In this example, the device **100** displays data via the touch sensitive display **114** in a second portrait configuration **608**.

[0173] It may be possible to display similar or different data in either the first or second portrait configuration **604**, **608**. It may also be possible to transition between the first portrait configuration **604** and second portrait configuration **608** by providing the device **100** a user gesture (e.g., a double tap gesture), a menu selection, or other means. Other suitable gestures may also be employed to transition between configurations. Furthermore, it may also be possible to transition the device **100** from the first or second portrait configuration **604**, **608** to any other configuration described herein depending upon which state the device **100** is moved.

[0174] An alternative output configuration may be accommodated by the device **100** being in a second state. Specifically, FIG. 6C depicts a third portrait configuration where data is displayed simultaneously on both the primary screen **104** and the secondary screen **108**. The third portrait configuration may be referred to as a Dual-Portrait (PD) output configuration. In the PD output configuration, the touch sensitive display **110** of the primary screen **104** depicts data in the first portrait configuration **604** while the touch sensitive display **114** of the secondary screen **108** depicts data in the second portrait configuration **608**. The simultaneous presentation of the first portrait configuration **604** and the second portrait configuration **608** may occur when the device **100** is in an open portrait state **320**. In this configuration, the device **100** may display one application window in one display **110** or **114**, two application windows (one in each display **110** and **114**), one application window and one desktop, or one desktop. Other configurations may be possible. It should be appreciated that it may also be possible to transition the device **100** from the simultaneous display of configurations **604**, **608** to any other configuration described herein depending upon which state the device **100** is moved. Furthermore, while in this state, an application's display preference may place the device into bilateral mode, in which both displays are active to display different windows in the same application. For example, a Camera application may display a viewfinder and controls on one side, while the other side displays a mirrored preview that can be seen by the photo subjects. Games involving simultaneous play by two players may also take advantage of bilateral mode.

[0175] FIGS. 6D and 6E depicts two further output configurations of the device **100** being in a third state. Specifically, FIG. 6D depicts the device **100** being in a closed landscape state **340** where the data is displayed on the primary screen **104**. In this example, the device **100** displays data via the touch sensitive display **110** in a first landscape configuration **612**. Much like the other configurations described herein, the first landscape configuration **612** may display a desktop, a home screen, one or more windows displaying application data, or the like.

[0176] FIG. 6E depicts the device **100** still being in the closed landscape state **340**, but instead data is displayed on the secondary screen **108**. In this example, the device **100** displays data via the touch sensitive display **114** in a second landscape configuration **616**. It may be possible to display similar or different data in either the first or second portrait configuration **612**, **616**. It may also be possible to transition between the first landscape configuration **612** and second landscape configuration **616** by providing the device **100** with one or both of a twist and tap gesture or a flip and slide

gesture. Other suitable gestures may also be employed to transition between configurations. Furthermore, it may also be possible to transition the device **100** from the first or second landscape configuration **612**, **616** to any other configuration described herein depending upon which state the device **100** is moved.

[0177] FIG. 6F depicts a third landscape configuration where data is displayed simultaneously on both the primary screen **104** and the secondary screen **108**. The third landscape configuration may be referred to as a Dual-Landscape (LD) output configuration. In the LD output configuration, the touch sensitive display **110** of the primary screen **104** depicts data in the first landscape configuration **612** while the touch sensitive display **114** of the secondary screen **108** depicts data in the second landscape configuration **616**. The simultaneous presentation of the first landscape configuration **612** and the second landscape configuration **616** may occur when the device **100** is in an open landscape state **340**. It should be appreciated that it may also be possible to transition the device **100** from the simultaneous display of configurations **612**, **616** to any other configuration described herein depending upon which state the device **100** is moved.

[0178] FIGS. 6G and 6H depict two views of a device **100** being in yet another state. Specifically, the device **100** is depicted as being in an easel state **312**. FIG. 6G shows that a first easel output configuration **618** may be displayed on the touch sensitive display **110**. FIG. 6H shows that a second easel output configuration **620** may be displayed on the touch sensitive display **114**. The device **100** may be configured to depict either the first easel output configuration **618** or the second easel output configuration **620** individually. Alternatively, both the easel output configurations **618**, **620** may be presented simultaneously. In some embodiments, the easel output configurations **618**, **620** may be similar or identical to the landscape output configurations **612**, **616**. The device **100** may also be configured to display one or both of the easel output configurations **618**, **620** while in a modified easel state **316**. It should be appreciated that simultaneous utilization of the easel output configurations **618**, **620** may facilitate two-person games (e.g., Battleship®, chess, checkers, etc.), multi-user conferences where two or more users share the same device **100**, and other applications. As can be appreciated, it may also be possible to transition the device **100** from the display of one or both configurations **618**, **620** to any other configuration described herein depending upon which state the device **100** is moved.

[0179] FIG. 6I depicts yet another output configuration that may be accommodated while the device **100** is in an open portrait state **320**. Specifically, the device **100** may be configured to present a single continuous image across both touch sensitive displays **110**, **114** in a portrait configuration referred to herein as a Portrait-Max (PMax) configuration **624**. In this configuration, data (e.g., a single image, application, window, icon, video, etc.) may be split and displayed partially on one of the touch sensitive displays while the other portion of the data is displayed on the other touch sensitive display. The Pmax configuration **624** may facilitate a larger display and/or better resolution for displaying a particular image on the device **100**. Similar to other output configurations, it may be possible to transition the device **100** from the Pmax configuration **624** to any other output configuration described herein depending upon which state the device **100** is moved.

[0180] FIG. 6J depicts still another output configuration that may be accommodated while the device **100** is in an open

landscape state **348**. Specifically, the device **100** may be configured to present a single continuous image across both touch sensitive displays **110**, **114** in a landscape configuration referred to herein as a Landscape-Max (LMax) configuration **628**. In this configuration, data (e.g., a single image, application, window, icon, video, etc.) may be split and displayed partially on one of the touch sensitive displays while the other portion of the data is displayed on the other touch sensitive display. The Lmax configuration **628** may facilitate a larger display and/or better resolution for displaying a particular image on the device **100**. Similar to other output configurations, it may be possible to transition the device **100** from the Lmax configuration **628** to any other output configuration described herein depending upon which state the device **100** is moved.

[0181] The device **100** manages desktops and/or windows with at least one window stack **700**, **728**, as shown in FIGS. **7A** and **7B**. A window stack **700**, **728** is a logical arrangement of active and/or inactive windows for a multi-screen device. For example, the window stack **700** or **728** may be logically similar to a deck of cards, where one or more windows or desktops are arranged in order, as shown in FIGS. **7A** and **7B**. An active window is a window that is currently being displayed on at least one of the touch sensitive displays **110**, **114**. For example, windows displayed on the touch sensitive displays **104** and **108** are active windows. An inactive window is a window that was opened and displayed but is now “behind” an active window and not being displayed. In embodiments, an inactive window may be for an application that is suspended, and thus, the window is not displaying active content. For example, windows **712**, **716**, **720**, and **724** (FIG. **7A**) are inactive windows.

[0182] A window stack **700**, **728** may have various arrangements or organizational structures. In the embodiment shown in FIG. **7A**, the device **100** includes a first stack **760** associated with the first touch sensitive display **110** and a second stack **764** associated with the second touch sensitive display **114**. Thus, each touch sensitive display **110** and **114** can have an associated respective window stack **760**, **764**. These two window stacks **760**, **764** may have different numbers of windows arranged in the respective stacks **760**, **764**. Further, the two window stacks **760**, **764** can also be identified differently and managed separately. Thus, the first window stack **760** can be arranged in order from a first window **704** to a next window **720** to a last window **724** and finally to a desktop **722**, which, in embodiments, is at the “bottom” of the window stack **760**. In embodiments, the desktop **722** is not always at the “bottom” since application windows can be arranged in the window stack below the desktop **722**, and the desktop **722** can be brought to the “top” of a stack over other windows as a result of performing a desktop reveal operation. Likewise, the second stack **764** can be arranged from a first window **708** to a next window **712** to a last window **716**, and finally to a desktop **718**, which, in embodiments, is a single desktop area, with desktop **722**, under all the windows in both window stack **760** and window stack **764**. A logical data structure for managing the two window stacks **760**, **764** may be as described in conjunction with FIG. **8** hereinbelow.

[0183] Another arrangement for a window stack **728** is shown in FIG. **7B**. In this embodiment, there is a single window stack **728** for both touch sensitive displays **110**, **114**. Thus, the window stack **728** is arranged from a desktop **758** to a first window **744** to a last window **756**. A window can be arranged in a position among all windows without an asso-

ciation to a specific touch sensitive display **110**, **114**. In this embodiment, each window is in the order of windows. Further, at least one window is identified as being active. For example, a single window may be rendered in two portions **732** and **736** that are displayed, respectively, on the first touch sensitive screen **110** and the second touch sensitive screen **114**. This single window may only occupy a single position in the window stack **728** although it is displayed on both displays **110**, **114**.

[0184] Yet another arrangement of a window stack **760** is shown in FIGS. **7C** through **7E**. The window stack **760** is shown in three “elevation” views. In FIG. **7C**, the top of the window stack **760** is shown. Two sides of the window stack **760** are shown in FIGS. **7D** and **7E**. In this embodiment, the window stack **760** resembles a stack of bricks. The windows are stacked on each other. Looking from the top of the window stack **760** in FIG. **7C**, only the top most windows in the window stack **760** are seen in different portions of the composite display **764**. The composite display **764** represents a logical model for the entire display area of the device **100**, which can include touch sensitive display **110** and touch sensitive display **114**. A desktop **786** (FIGS. **7D** and **7E**) or a window can occupy part or all of the composite display **764**.

[0185] In the embodiment shown, the desktop **786** is the lowest display or “brick” in the window stack **760**. Thereupon, window **1** **782**, window **2** **782**, window **3** **768**, and window **4** **770** are layered. Window **1** **782**, window **3** **768**, window **2** **782**, and window **4** **770** only occupy a portion of the composite display **764**. Thus, another part of the stack **760** includes window **8** **774** and windows **5** through **7** shown in section **790**. Only the top window in any portion of the composite display **764** is actually rendered and displayed. Thus, as shown in the top view in FIG. **7C**, window **4** **770**, window **8** **774**, and window **3** **768** are displayed as being at the top of the display in different portions of the window stack **760**. A window can be dimensioned to occupy only a portion of the composite display **760** to “reveal” windows lower in the window stack **760**. For example, window **3** **768** is lower in the stack than both window **4** **770** and window **8** **774** but is still displayed. A logical data structure to manage the window stack can be as described in conjunction with FIG. **8**.

[0186] When a new window is opened on the device **100**, the newly activated window is generally positioned at the top of the stack. However, where and how the window is positioned within the stack can be a function of the orientation of the device **100**, the context of what programs, functions, software, etc. are being executed on the device **100**, how the stack is positioned when the new window is opened, etc. To insert the window in the stack, the position in the stack for the window is determined and the touch sensitive display **110**, **114** to which the window is associated may also be determined. With this information, a logical data structure for the window can be created and stored. When user interface or other events or tasks change the arrangement of windows, the window stack(s) can be changed to reflect the change in arrangement. It should be noted that these same concepts described above can be used to manage the one or more desktops for the device **100**.

[0187] A logical data structure **800** for managing the arrangement of windows or desktops in a window stack is shown in FIG. **8**. The logical data structure **800** can be any data structure used to store data whether an object, record, file, etc. The logical data structure **800** can be stored in any type of database or data storage system, regardless of protocol

or standard. In embodiments, the logical data structure **800** includes one or more portions, fields, attributes, etc. that store data in a logical arrangement that allows for easy storage and retrieval of the information. Hereinafter, these one or more portions, fields, attributes, etc. shall be described simply as fields. The fields can store data for a window identifier **804**, dimensions **808**, a stack position identifier **812**, a display identifier **816**, and/or an active indicator **820**. Each window in a window stack can have an associated logical data structure **800**. While only a single logical data structure **800** is shown in FIG. 8, there may be more or fewer logical data structures **800** used with a window stack (based on the number of windows or desktops in the stack), as represented by ellipses **824**. Further, there may be more or fewer fields than those shown in FIG. 8, as represented by ellipses **828**.

[0188] A window identifier **804** can include any identifier (ID) that uniquely identifies the associated window in relation to other windows in the window stack. The window identifier **804** can be a globally unique identifier (GUID), a numeric ID, an alphanumeric ID, or other type of identifier. In embodiments, the window identifier **804** can be one, two, or any number of digits based on the number of windows that can be opened. In alternative embodiments, the size of the window identifier **804** may change based on the number of windows opened. While the window is open, the window identifier **804** may be static and remain unchanged.

[0189] Dimensions **808** can include dimensions for a window in the composite display **760**. For example, the dimensions **808** can include coordinates for two or more corners of the window or may include one coordinate and dimensions for the width and height of the window. These dimensions **808** can delineate what portion of the composite display **760** the window may occupy, which may be the entire composite display **760** or only part of composite display **760**. For example, window **4 770** may have dimensions **880** that indicate that the window **4 770** will occupy only part of the display area for composite display **760**, as shown in FIGS. 7C through 7E. As windows are moved or inserted in the window stack, the dimensions **808** may change.

[0190] A stack position identifier **812** can be any identifier that can identify the position in the stack for the window or may be inferred from the window's control record within a data structure, such as a list or a stack. The stack position identifier **812** can be a GUID, a numeric ID, an alphanumeric ID, or other type of identifier. Each window or desktop can include a stack position identifier **812**. For example, as shown in FIG. 7A, window **1 704** in stack **1 760** can have a stack position identifier **812** of 1 identifying that window **704** is the first window in the stack **760** and the active window. Similarly, window **6 724** can have a stack position identifier **812** of 3 representing that window **724** is the third window in the stack **760**. Window **2 708** can also have a stack position identifier **812** of 1 representing that window **708** is the first window in the second stack **764**. However, in FIG. 7B, an alternative stack position numbering of windows is provided; i.e., window **1 744** can have a stack position identifier **812** of 1, window **3**, rendered in portions **732** and **736**, can have a stack position identifier **812** of 3, and window **6 756** can have a stack position identifier **812** of 6. Thus, depending on the type of stack, the stack position identifier **812** can represent a window's location in the stack.

[0191] A display identifier **816** can identify that the window or desktop is associated with a particular display, such as the first display **110** or the second display **114**, or the composite

display **760** composed of both displays. While this display identifier **816** may not be needed for a multi-stack system, as shown in FIG. 7A, the display identifier **816** can indicate whether a window in the serial stack of FIG. 7B is displayed on a particular display. Thus, window **3** may have two portions **732** and **736** in FIG. 7B. The first portion **732** may have a display identifier **816** for the first display while the second portion **736** may have a display identifier **816** for the second display **114**. However, in alternative embodiments, the window may have two display identifiers **816** that represent that the window is displayed on both of the displays **110, 114**, or a display identifier **816** identifying the composite display. In another alternate embodiment, the window may have a single display identifier **816** to represent that the window is displayed on both of the displays **110, 114**.

[0192] Similar to the display identifier **816**, an active indicator **820** may not be needed with the dual stack system of FIG. 7A, as the window in stack position **1** is active and displayed. In the system of FIG. 7B, the active indicator **820** can indicate which window(s) in the stack is being displayed. Thus, window **3** may have two portions **732** and **736** in FIG. 7B. The first portion **732** may have an active indicator **820** while the second portion **736** may also have an active indicator **820**. However, in alternative embodiments, window **3** may have a single active indicator **820**. The active indicator **820** can be a simple flag or bit that represents that the window is active or displayed.

[0193] An embodiment of a method **900** for creating a window stack is shown in FIG. 9. While a general order for the steps of the method **900** is shown in FIG. 9. Generally, the method **900** starts with a start operation **904** and ends with an end operation **928**. The method **900** can include more or fewer steps or can arrange the order of the steps differently than those shown in FIG. 9. The method **900** can be executed as a set of computer-executable instructions executed by a computer system and encoded or stored on a computer readable medium. Hereinafter, the method **900** shall be explained with reference to the systems, components, modules, software, data structures, user interfaces, etc. described in conjunction with FIGS. 1-8.

[0194] A multi-screen device **100** can receive activation of a window, in step **908**. In embodiments, the multi-screen device **100** can receive activation of a window by receiving an input from the touch sensitive display **110** or **114**, the configurable area **112** or **116**, a gesture capture region **120** or **124**, or some other hardware sensor operable to receive user interface inputs. The processor may execute the Task Management Module **540** (FIG. 5A) to receive the input. The Task Management Module **540** can interpret the input as requesting an application task to be executed that will open a window in the window stack.

[0195] In embodiments, the Task Management Module **540** places the user interface interaction in the task stack **552** (FIG. 5A) to be acted upon by the Display Configuration Module **568** (FIG. 5B) of the Multi-Display Management Module **524** (FIGS. 5A, 5B). Further, the Task Management Module **540** waits for information from the Multi-Display Management Module **524** to send instructions to the Window Management Module **532** to create the window in the window stack.

[0196] The Multi-Display Management Module **524**, upon receiving instruction from the Task Management Module **540**, determines to which touch portion of the composite display **760**, the newly activated window should be associ-

ated, in step 912. For example, window 4 770 is associated with the composite display 764 (FIG. 7A). In embodiments, the device state module 574 (FIG. 5B) of the Multi-Display Management Module 524 may determine how the device is oriented or in what state the device is in, e.g., open, closed, portrait, etc. Further, the preferences module 572 (FIG. 5B) and/or requirements module 580 (FIG. 5B) may determine how the window is to be displayed. The gesture module 576 (FIG. 5B) may determine the user's intentions about how the window is to be opened based on the type of gesture and the location of where the gesture is made.

[0197] The Display Configuration Module 568 (FIG. 5B) may use the input from these modules and evaluate the current window stack 760 to determine the best place and the best dimensions, based on a visibility algorithm, to open the window. Thus, the Display Configuration Module 568 determines the best place to put the window at the top of the window stack 760, in step 916. The visibility algorithm, in embodiments, determines for all portions of the composite display, which windows are at the top of the stack. For example, the visibility algorithm determines that window 3 768, window 4 770, and window 8 774 are at the top of the stack 760 as viewed in FIGS. 7C through 7E. Upon determining where to open the window, the Display Configuration Module 568 can assign a display identifier 816 and possibly dimensions 808 to the window. The display identifier 816 and dimensions 808 can then be sent back to the Task Management Module 540. The Task Management Module 540 may then assign the window a stack position identifier 812 indicating the windows position at the top of the window stack.

[0198] In embodiments, the Task Management Module 540 sends the window stack information and instructions to render the window to the Window Management Module 532 (FIG. 5A). The Window Management Module 532 and the Task Management Module 540 (FIG. 5A) can create the logical data structure 800, in step 924. Both the Task Management Module 540 and the Window Management Module 532 may create and manage copies of the window stack. These copies of the window stack can be synchronized or kept similar through communications between the Window Management Module 532 and the Task Management Module 540. Thus, the Window Management Module 532 and the Task Management Module 540, based on the information determined by the Multi-Display Management Module 524, can assign dimensions 808, a stack position identifier 812 (e.g., window 1 782, window 4 770, etc.), a display identifier 816 (e.g., touch sensitive display 1 110, touch sensitive display 2 114, composite display identifier, etc.), and an active indicator 820, which is generally always set when the window is at the "top" of the stack. The logical data structure 800 may then be stored by both the Window Management Module 532 and the Task Management Module 540. Further, the Window Management Module 532 and the Task Management Module 540 may thereafter manage the window stack and the logical data structure(s) 800.

[0199] In at least one embodiment of the device 100, the MDM 524 (and its Gesture Module 576) may be operable for interpreting gestures in the gesture capture areas 120 and 124 so that they may be used to manipulate application windows, e.g., to minimize, maximize, and move application windows between the displays 110 and 114, as well as to rearrange the application windows in the window stack, to active an application window across multiple screen displays, and/or reduce an application window to a single screen display that is cur-

rently displayed across displays 110 and 114. However, in various embodiments, user gesture input to the gesture capture areas 120 and 124 is not used to manipulate displayed desktops. Referring to FIG. 10A, assuming the flick gesture 404 and the drag gesture 400 are concurrently performed in, e.g., the gesture capture area 124, wherein each of the displays 110 and 114 display a desktop (respectively, D1 and D2), then as shown in the right-hand portion of this figure, such gestures do not affect the displays 110 and 114. However, referring to FIG. 10B, the same combination of gestures (e.g., flick gesture 404 and drag gesture 400) provided on the display 114 cause the displayed desktops D1 and D2 to shift to the left. More particularly, in FIG. 10B, the flick-drag combination gesture on display 114 shifts the application D1 off the display 110, causes the application D2 to shift from the display 114 to the display D1, and for another desktop, D3, this later desktop shifts onto the display 114. Accordingly, depending on the window stack implementation (e.g., the implementation of FIGS. 7A and 7B, or that of FIGS. 7C-7E), the window stack(s) may be changed to reflect the change in which desktops are displayed to the user. Thus, for the embodiment of the window stacks as provided in FIG. 7A, the desktop D2 changes window stack while desktops D1 and D3 change positions within their respective window stacks, and of course, the logical data structures 800 for each of the desktops D1, D2, and D3 changes accordingly.

[0200] Referring to FIG. 10C, this figure illustrates the same flick-drag combination gesture used in the gesture capture area 120 for moving the application window A1 for an application (not a desktop) from the display 110 to the display 114. Note that in performing such moving of the application window A1, the respective window stacks (assuming a window stack embodiment as per FIG. 7A) for the displays 110 and 114 are changed. In particular, when the application window A1 moves from the display 110 (due to the flick-drag combination gesture in area 120), this window also moves from the top of the window stack for the display 110 to the top of the window stack for the display 114, and is removed from the window stack for the display 110. Thus the movement of application windows between displays 110 and 114 will change their corresponding window stacks. Moreover, as shown in FIG. 10C, assuming the application window A1 were the only application window in the window stack for the display 110, then if a desktop D1 for the display 110 is on the window stack for display 110 immediately below application window A1, then desktop D1 is presented on the display 110. However, as shown in FIG. 10D, if there is an application window A3 in the window stack immediately below application window A1 in the window stack for display 110, then upon activation of the flick-drag combination gesture in area 120, application window A3 is displayed in display 110.

[0201] In a further embodiment shown in FIG. 10E, this figure illustrates that the same flick-drag gesture combination applied directly to a display 110 or 104, wherein an application window (labeled A1 in FIG. 10E) is displayed will yield no change to the displays 110 and 114.

[0202] Note that in addition to the combination flick-drag gesture used illustratively above, other gestures and combination gestures input to the gesture capture areas 120 and 124. In particular, the following gestures and their interpretations (as applied to application windows) are contemplated herein:

| Gesture (or combination gesture) | Interpretation by the Gesture Module 576 |
|--|---|
| Drag 400 | Depending on what application window is active (e.g., displayed on the corresponding display 110 or 124 and has focus) to be manipulated by the drag 400 gesture, the application window may be, e.g., moved, maximized or minimized. For example, if such an application window is displayed but minimized (e.g., as an icon), then a drag may maximize this window. Alternatively, a drag gesture for such an application window that is maximized may minimize the application window. |
| Long Drag | <p>A long-drag gesture is a drag gesture on one of the gesture capture areas 120, 124. A long drag is like a regular drag gesture 400, except longer. In practice what this means is that one is performing several individual drag operations in sequence, as long as the long-drag is being performed.</p> <p>When a regular drag 400 operation is performed, it moves applications, or minimizes or maximizes dual-screen applications. A long-drag gesture will do the exact same things, except several of them in sequence automatically for as long as the drag is performed. Essentially, a long-drag gesture equals several individual drag operations performed sequentially. Therefore, by using a long-drag, it is possible to move several single and/or dual-screen applications, depending on the length of gesture.</p> <p>For example, if 10 applications are displayed on the display screen 114, wherein each of the applications is a single-screen application. In performing a long drag from right to left, the application in view on the display 114 (app A) starts moving towards the display 110. Upon continuing the long drag, eventually the application (app A) is positioned on display 110 and on the display 114, another (app B) is fully exposed. If the user stopped "long drag" at this point, only a regular drag 400 would have been performed. However, by continuing the drag gesture (i.e., a long-drag); the application on the display 114 (app B) starts moving towards the display 110. Thus, on the display 114, a third application (app C) becomes visible. Eventually, app B is on the display 110 (hiding app A), and app C is fully exposed on the display 114. In essence, two "regular" drag operations will have been performed with one "long drag" gesture. If app A had been dual-screen capable, a long-drag gesture could have first maximized it, then minimized it to the display 110.</p> |
| Flick 404 | Moves or minimizes the active application window similarly to a drag 400. However, no maximization of the window is applied. |
| Pinch 408 | Swaps the position of application windows displayed on the corresponding displays 110 or 124. Ignored if the device 100 is closed. Also "ignored" for maximized dual-screen applications. |
| Spread 410 | Displays the application manager (i.e., a "pop-up window" that displays the application windows in the window stack) |
| Pin and Drag combination 416 (FIG. 4H) | <p>Holds a pinned application window in place while moving another application window in the window stack.</p> <p>In more detail, a pin and drag gesture allows for more advanced window navigation where one application is held in view while the window stack is navigated beneath. The pin and drag gesture is performed by holding down one finger on one of the gesture capture areas 120, 124, and performing a drag or flick gesture with another finger.</p> <p>Essentially, Pin and drag allows for keeping one application in view while the rest of the applications in the window stacks are navigated. It's like performing drag gestures but one of the applications in view is not affected by the gestures: it's being "held down" and not allowed to move while the others do. A Pin and drag gesture is used when the device 100 is open (e.g., with both displays 110, 114 being active).</p> |

[0203] The device **100** functionality described immediately above for manipulating only application windows (via user gestures input the gesture capture area **120** or **124**) may be provided a particular embodiment of the MDM **524** that performs the flowchart shown in FIG. 11 and now described. Accordingly, in step **1104** of FIG. 11, the Gesture Module **576** (of the MDM **524**) detects a user gesture input to one (or both) of the gesture capture areas **120** or **124**. In one embodiment, such detection is performed by comparing the input user gesture with a predetermined collection of gesture template

data descriptions resident on the device **100**. Assuming, the Gesture Module **576** is able to identify the gesture input as one of the predetermined user gestures for the collection, this module notifies the Display Configuration Module **568** (of the MDM **524**) of: (i) the gesture identity of the gesture input, and (ii) data identifying from where the gesture input as received (e.g., from one of the gesture capture areas or one of the displays **110** and **114**).

[0204] The Display Configuration Module **568** determines for each gesture input to the device **100** (whether this input is

to one of the displays **110, 114**, or to one of the gesture capture areas **120 or 124**) if the gesture identity (for the gesture input) is applicable or not for initiating a predetermined gesture corresponding action to perform on an application window or a desktop displayed on the display **110** or **114** to which the gesture input corresponds. Thus, if the gesture input is from one of the displays **110** or **114**, then Display Configuration Module **568** will only notify the Window Management Module **532** to modify the display receiving the gesture input if this display displays a desktop. Alternatively, if the gesture input is from one of the gesture capture areas **120** or **124**, then Display Configuration Module **568** will only notify the Window Management Module **532** to modify the display corresponding to the gesture input if this display displays an application window.

[0205] Accordingly, in step **1108**, the Display Configuration Module **568** determines whether there is an application window displayed on the display **110** or **114** corresponding to the gesture capture area **120** or **124** that received the inputted user gesture. If no application window is displayed on the display **110** or **114** corresponding to the gesture capture area **120** or **124** that received the inputted user gesture, then in step **1112**, Display Configuration Module **568** prevents the user gesture input from being used to change the displays **110** and **114**, and instead discards such user gesture input. However, if the display **110** or **114** (corresponding to the gesture capture area receiving the inputted user gesture) displays an application window, then the Display Configuration Module **568** uses the identity of the gesture input for determining how to change the display **110** or **114**.

[0206] Note that it is also within the scope of the present disclosure that the above described different processing performed on one or more desktops as opposed to the processing performed on one or more application windows, according to user gesture input, can be reversed. For example, in an alternative embodiment, for each of the displays **110** and **114**: (i) a desktop displayed on the display (**110** or **114**) is manipulated via gesture input to the corresponding gesture capture area **120** or **124**, (ii) when an application window is displayed on the display (**110** or **114**), such gesture input to the corresponding gesture capture area **120** or **124** is not used for manipulating the displayed application window, (iii) an application window displayed on the display (**110** or **114**) is manipulated via gesture input that includes user input that directly contacts the display (**110** or **114**), and (iv) when a desktop is displayed on the display (**110** or **114**), such a gesture input that includes user input that directly contacts the display (**110** or **114**) is not used for manipulating the desktop.

[0207] The exemplary systems and methods of this disclosure have been described in relation to processing performed for the interpretation of gestures in the gesture capture areas **120** and **124**. However, to avoid unnecessarily obscuring the present disclosure, the preceding description omits a number of known structures and devices. This omission is not to be construed as a limitation of the scopes of the claims. Specific details are set forth to provide an understanding of the present disclosure. It should however be appreciated that the present disclosure may be practiced in a variety of ways beyond the specific detail set forth herein.

[0208] Furthermore, while the exemplary aspects, embodiments, and/or configurations illustrated herein show the various components of the system collocated, certain components of the system can be located remotely, at distant portions of a distributed network, such as a LAN and/or the

Internet, or within a dedicated system. Thus, it should be appreciated, that the components of the system can be combined in to one or more devices, or collocated on a particular node of a distributed network, such as an analog and/or digital telecommunications network, a packet-switch network, or a circuit-switched network. It will be appreciated from the preceding description, and for reasons of computational efficiency, that the components of the system can be arranged at any location within a distributed network of components without affecting the operation of the system. For example, the various components can be located in a switch such as a PBX and media server, gateway, in one or more communications devices, at one or more users' premises, or some combination thereof. Similarly, one or more functional portions of the system could be distributed between a telecommunications device(s) and an associated computing device.

[0209] Furthermore, it should be appreciated that the various links connecting the elements can be wired or wireless links, or any combination thereof, or any other known or later developed element(s) that is capable of supplying and/or communicating data to and from the connected elements. These wired or wireless links can also be secure links and may be capable of communicating encrypted information. Transmission media used as links, for example, can be any suitable carrier for electrical signals, including coaxial cables, copper wire and fiber optics, and may take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0210] Also, while the flowcharts have been discussed and illustrated in relation to a particular sequence of events, it should be appreciated that changes, additions, and omissions to this sequence can occur without materially affecting the operation of the disclosed embodiments, configuration, and aspects.

[0211] A number of variations and modifications of the disclosure can be used. It would be possible to provide for some features of the disclosure without providing others.

[0212] In yet another embodiment, the systems and methods of this disclosure can be implemented in conjunction with a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element(s), an ASIC or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as discrete element circuit, a programmable logic device or gate array such as PLD, PLA, FPGA, PAL, special purpose computer, any comparable means, or the like. In general, any device(s) or means capable of implementing the methodology illustrated herein can be used to implement the various aspects of this disclosure. Exemplary hardware that can be used for the disclosed embodiments, configurations and aspects includes computers, handheld devices, telephones (e.g., cellular, Internet enabled, digital, analog, hybrids, and others), and other hardware known in the art. Some of these devices include processors (e.g., a single or multiple microprocessors), memory, nonvolatile storage, input devices, and output devices. Furthermore, alternative software implementations including, but not limited to, distributed processing or component/object distributed processing, parallel processing, or virtual machine processing can also be constructed to implement the methods described herein.

[0213] In yet another embodiment, the disclosed methods may be readily implemented in conjunction with software using object or object-oriented software development environments that provide portable source code that can be used

on a variety of computer or workstation platforms. Alternatively, the disclosed system may be implemented partially or fully in hardware using standard logic circuits or VLSI design. Whether software or hardware is used to implement the systems in accordance with this disclosure is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software or hardware systems or microprocessor or microcomputer systems being utilized.

[0214] In yet another embodiment, the disclosed methods may be partially implemented in software that can be stored on a storage medium, executed on programmed general-purpose computer with the cooperation of a controller and memory, a special purpose computer, a microprocessor, or the like. In these instances, the systems and methods of this disclosure can be implemented as program embedded on personal computer such as an applet, JAVA® or CGI script, as a resource residing on a server or computer workstation, as a routine embedded in a dedicated measurement system, system component, or the like. The system can also be implemented by physically incorporating the system and/or method into a software and/or hardware system.

[0215] Although the present disclosure describes components and functions implemented in the aspects, embodiments, and/or configurations with reference to particular standards and protocols, the aspects, embodiments, and/or configurations are not limited to such standards and protocols. Other similar standards and protocols not mentioned herein are in existence and are considered to be included in the present disclosure. Moreover, the standards and protocols mentioned herein and other similar standards and protocols not mentioned herein are periodically superseded by faster or more effective equivalents having essentially the same functions. Such replacement standards and protocols having the same functions are considered equivalents included in the present disclosure.

[0216] The present disclosure, in various aspects, embodiments, and/or configurations, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various aspects, embodiments, configurations embodiments, subcombinations, and/or subsets thereof. Those of skill in the art will understand how to make and use the disclosed aspects, embodiments, and/or configurations after understanding the present disclosure. The present disclosure, in various aspects, embodiments, and/or configurations, includes providing devices and processes in the absence of items not depicted and/or described herein or in various aspects, embodiments, and/or configurations hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and/or reducing cost of implementation.

[0217] The foregoing discussion has been presented for purposes of illustration and description. The foregoing is not intended to limit the disclosure to the form or forms disclosed herein. In the foregoing Detailed Description for example, various features of the disclosure are grouped together in one or more aspects, embodiments, and/or configurations for the purpose of streamlining the disclosure. The features of the aspects, embodiments, and/or configurations of the disclosure may be combined in alternate aspects, embodiments, and/or configurations other than those discussed above. This method of disclosure is not to be interpreted as reflecting an intention that the claims require more features than are

expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed aspect, embodiment, and/or configuration. Thus, the following claims are hereby incorporated into this Detailed Description, with each claim standing on its own as a separate preferred embodiment of the disclosure.

[0218] Moreover, though the description has included description of one or more aspects, embodiments, and/or configurations and certain variations and modifications, other variations, combinations, and modifications are within the scope of the disclosure, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative aspects, embodiments, and/or configurations to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method for configuring a device having at least one display screen for displaying one or more of screen display presentations and for displaying one or more operably different screen display presentations, wherein the screen display presentations are able to be manipulated via user gesture input differently from the operably different screen display presentations, comprising:

receiving a first user gesture input to a gesture capture area associated with the at least one screen, the gesture capture area being separate from the at least one display screen;

determining whether a first of the one or more of screen display presentations is displayed on the at least one screen;

when the first screen display presentation is determined to be displayed, perform a step of interpreting the user gesture input for determining a predetermined operation to apply to the first screen display presentation;

performing the predetermined operation;

when none of the one or more screen display presentations are determined to be displayed in the step of determining, perform a step of preventing the device from using the user gesture input to change the at least one display screen;

receiving a second user gesture input in response to a user gesture input that includes user contact directly to the at least one screen;

second determining whether one of the operably different screen display presentations is displayed on the at least one screen;

when the one operably different screen display presentation is determined to be displayed on the at least one screen, performing a step of second interpreting the second user gesture input for determining a predetermined second operation to apply to the one operably different screen display presentation; and

when none of the operably different screen display presentations are determined in the step of second determining, perform a step of second preventing the device from using the second user gesture input to change the at least one display screen.

2. The method of claim **1**, wherein the gesture capture area includes a capacitive touch sensitive area for receiving the user gesture input.

3. The method of claim **2**, wherein the gesture capture area is not capable of displaying any of the screen display presentations, and is not capable of displaying any of the operably different screen display presentations.

4. The method of claim **1**, wherein the first screen display presentation includes an application window.

5. The method of claim **1**, wherein the one operably different screen display presentation includes a desktop.

6. The method of claim **1**, wherein the user gesture input includes one of: a drag, a flick, a pinch, and a spread.

7. The method of claim **1**, wherein the second user gesture input includes one of: a drag, a flick, a pinch, and a spread.

8. The method of claim **1**, wherein each of the first screen display presentation and the one operably different screen display presentation has corresponding data included in a corresponding stack, wherein a user inputs user gestures to the device rearranging entries in each stack.

9. The method of claim **1**, wherein each of the user gesture input and the second user gesture input is for a same user gesture.

10. The method of claim **1**, wherein the at least one display screen is one of a plurality of display screens of the device, and the predetermined operation moves the first screen display presentation from the at least screen to another of the plurality of screens.

11. The method of claim **8**, wherein the predetermined second operation moves the one operably different screen display presentation from the at least screen to another of the plurality of screens.

12. The method of claim **1**, wherein there is a predetermined collection of a plurality of user gesture input descriptions for recognizing user gesture inputs that are interpretable in the step of interpreting,

wherein the one or more screen display presentations includes a plurality of application windows,

wherein each of a plurality of software applications provide a corresponding one of the plurality of application windows,

wherein each of the user gesture inputs (UGI) of the predetermined collection is operable for recognizing user gesture inputs for each application window (AW) of the plurality of application windows when: (1) the user gesture input UGI is an instance of the first user gesture input, and (2) the application window AW is an instance of the first screen display presentation for which the user gesture input UGI is an input.

13. The method of claim **12**, wherein each of the software applications is operable on the device independently of each of the other of the software applications.

14. A computer readable medium for configuring a device having at least one display screen for displaying one or more of screen display presentations and for displaying one or more operably different screen display presentations, wherein the screen display presentations are able to be manipulated via user gesture input differently from the operably different screen display presentations, comprising:

machine instructions for performing the following steps: receiving a first user gesture input to a gesture capture area associated with the at least one screen, the gesture capture area being separate from the at least one display screen;

determining whether a first of the one or more of screen display presentations is displayed on the at least one screen;

when the first screen display presentation is determined to be displayed, perform a step of interpreting the user gesture input for determining a predetermined operation to apply to the first screen display presentation;

performing the predetermined operation;

when none of the one or more screen display presentations are determined to be displayed in the step of determining, perform a step of preventing the device from using the user gesture input to change the at least one display screen;

receiving a second user gesture input in response to a user gesture input that includes user contact directly to the at least one screen;

second determining whether one of the operably different screen display presentations is displayed on the at least one screen;

when the one operably different screen display presentation is determined to be displayed on the at least one screen, performing a step of second interpreting the second user gesture input for determining a predetermined second operation to apply to the one operably different screen display presentation; and

when none of the operably different screen display presentations are determined in the step of second determining, perform a step of second preventing the device from using the second user gesture input to change the at least one display screen.

15. The computer readable medium of claim **14**, wherein the step of interpreting is for rearranging of a stack for the screen display presentations so that an order of presentation of the screen display presentations is rearranged.

16. The computer readable medium of claim **15**, wherein each of the screen display presentations includes an application window.

17. The computer readable medium of claim **14**, wherein the step of second interpreting is for rearranging of a stack for the operably different screen display presentations so that an order of presentation of the operably different screen display presentations is rearranged.

18. The computer readable medium of claim **17**, wherein each of the operably different screen display presentations includes a desktop.

19. A hand-held device, comprising:

at least one display screen for displaying one or more of screen display presentations and for displaying one or more operably different screen display presentations;

a gesture capture area associated with the at least one screen for receiving first user gesture input, the gesture capture area being separate from the at least one display screen;

wherein the screen display presentations are operably manipulated via the first user gesture input received by the gesture capture area;

wherein the at least one display screen is responsive to user touch for receiving second user gestures input for manipulating the operably different screen presentations;

a display manager for determining: (a) whether one of the screen display presentations is displayed on the at least

one screen, and (b) whether one of the operably different screen display presentations is displayed on the at least one screen;

a gesture interpreter for interpreting each of the first and second user gesture inputs for determining an action to perform on a corresponding displayed one of the screen display presentations or a corresponding displayed one of the operably different screen display presentations, wherein the gesture interpreter compares each of the first and second user gesture inputs with a predetermined collection of a plurality of user gesture input descriptions for recognizing user gesture inputs that are interpretable;

wherein for the first user gesture input, if the one screen display presentation is determined to be displayed on the at least one screen, the interpreter interprets the first user gesture input for determining a predetermined first task to apply to the one screen display presentation;

wherein when none of the one or more screen display presentations are determined to be displayed in the display manger prevents the device from using the first user gesture input to change the at least one display screen;

wherein for the second user gestures input, if the one operably different screen display presentation is determined to be displayed on the at least one screen, the interpreter interprets the second user gesture input for determining a predetermined second task to apply to the one operably different screen display presentation; and

wherein when none of the operably different screen display presentations are determined to be displayed in the display manger prevents the device from using the second user gesture input to change the at least one display screen.

a task manager for performing the predetermined first and second tasks.

20. The device of claim 19, wherein the gesture capture area is not capable of displaying any of the screen display presentations, and is not capable of displaying any of the operably different screen display presentations.

21. The device of claim 19, wherein the one screen display presentation includes an application window.

22. The method of claim 19, wherein the one operably different screen display presentation includes a desktop.

* * * * *