



US 20120131519A1

(19) **United States**

(12) **Patent Application Publication**

Jitkoff

(10) **Pub. No.: US 2012/0131519 A1**

(43) **Pub. Date: May 24, 2012**

(54) **SURFACING OFF-SCREEN VISIBLE
OBJECTS**

(75) Inventor: **John Nicholas Jitkoff**, Palo Alto,
CA (US)

(73) Assignee: **Google Inc.**

(21) Appl. No.: **13/300,438**

(22) Filed: **Nov. 18, 2011**

Related U.S. Application Data

(60) Provisional application No. 61/415,179, filed on Nov. 18, 2010, provisional application No. 61/458,326, filed on Nov. 18, 2010.

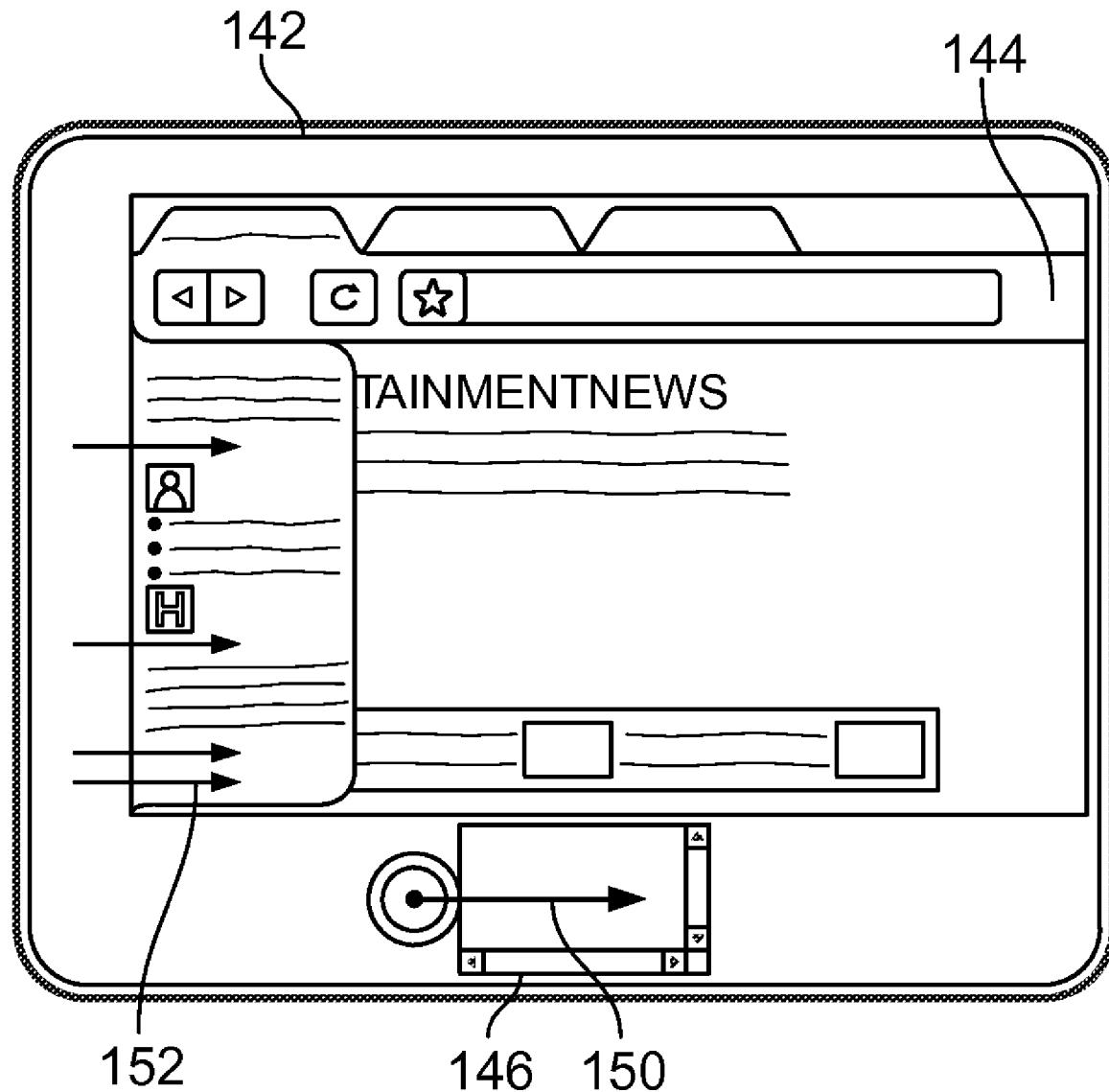
Publication Classification

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

(52) **U.S. Cl.** **715/863**

(57) ABSTRACT

A computer-implemented user input process for a computing device includes receiving, on a touch pad surface over a graphical display, a user input motion dragging across the touch pad surface, identifying the dragging input motion as originating off an edge of the touch pad by identifying a sensed first location for the input motion at a peripheral edge of the touch pad surface, and displaying on the graphical display a sliding graphical element that is animated to move from the edge of the display into a body of the display, over a nonmoving element on the display, in response to identifying the dragging input motion.



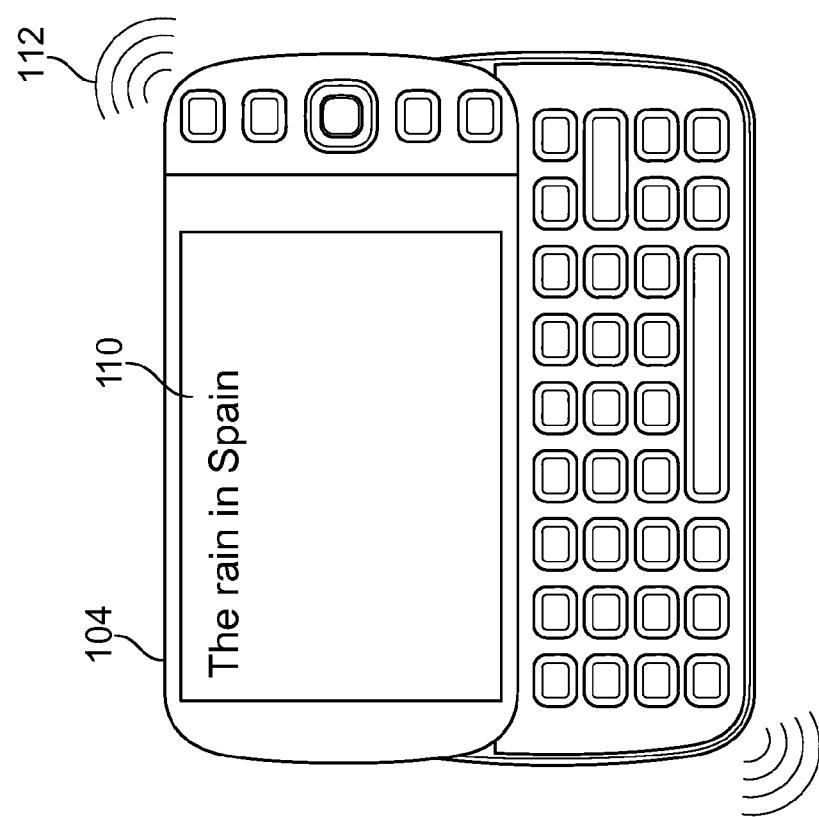


FIG. 1B

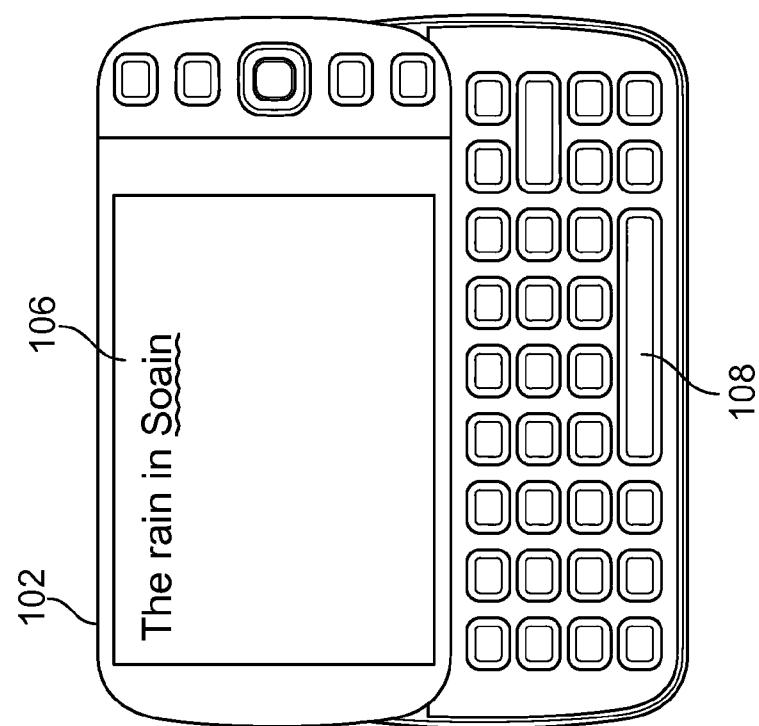
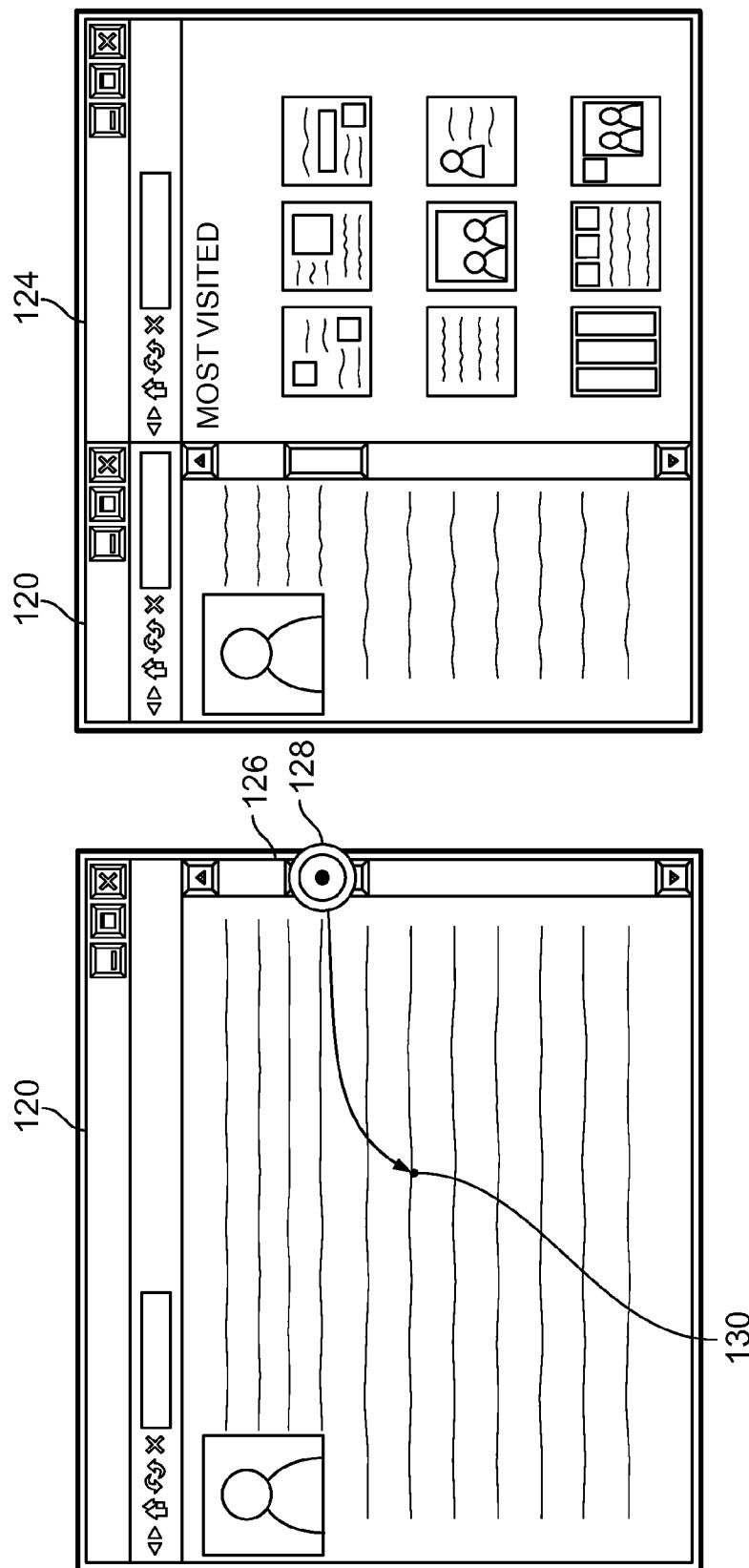
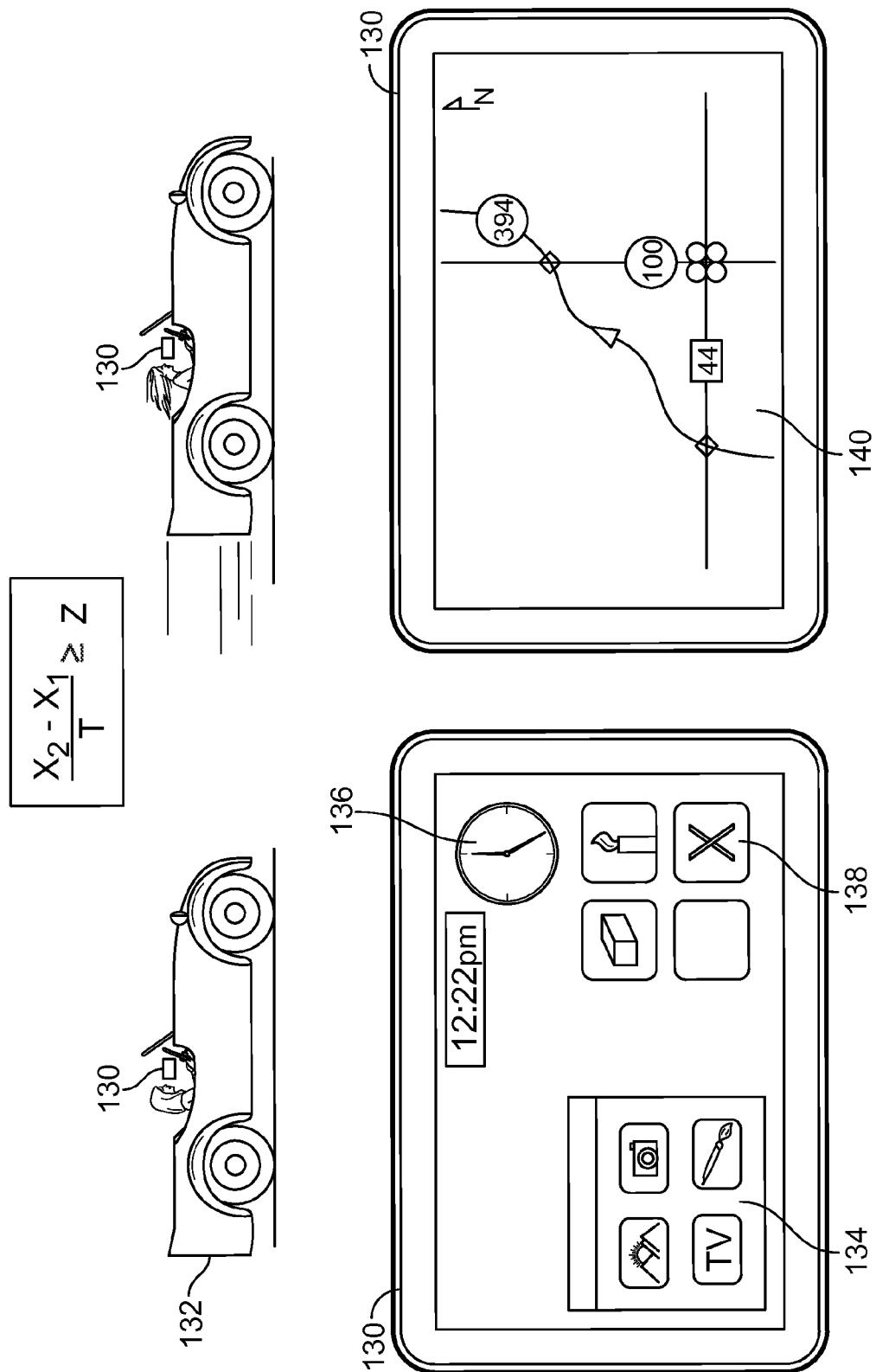


FIG. 1A





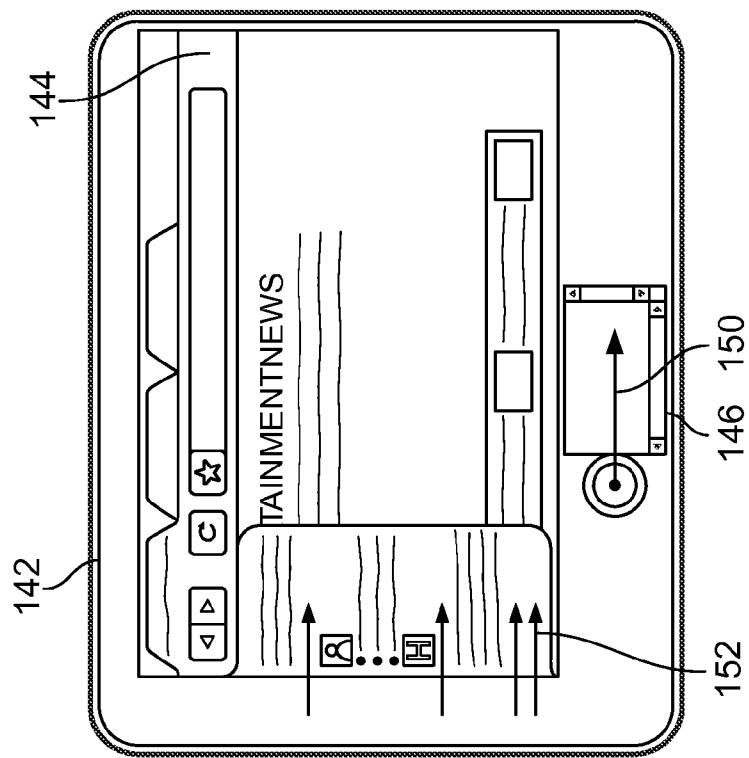


FIG. 4B

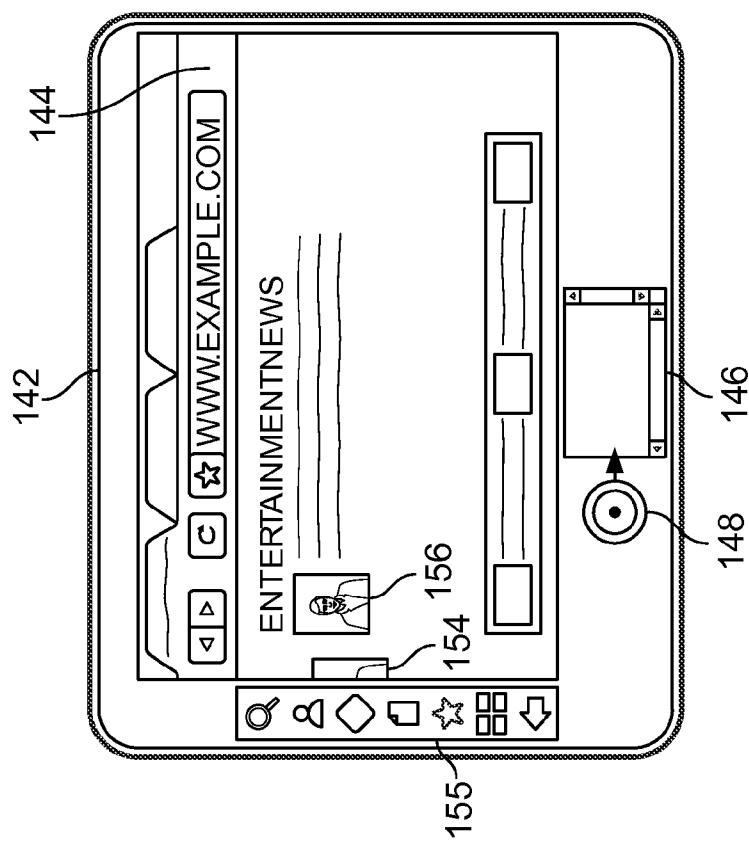


FIG. 4A

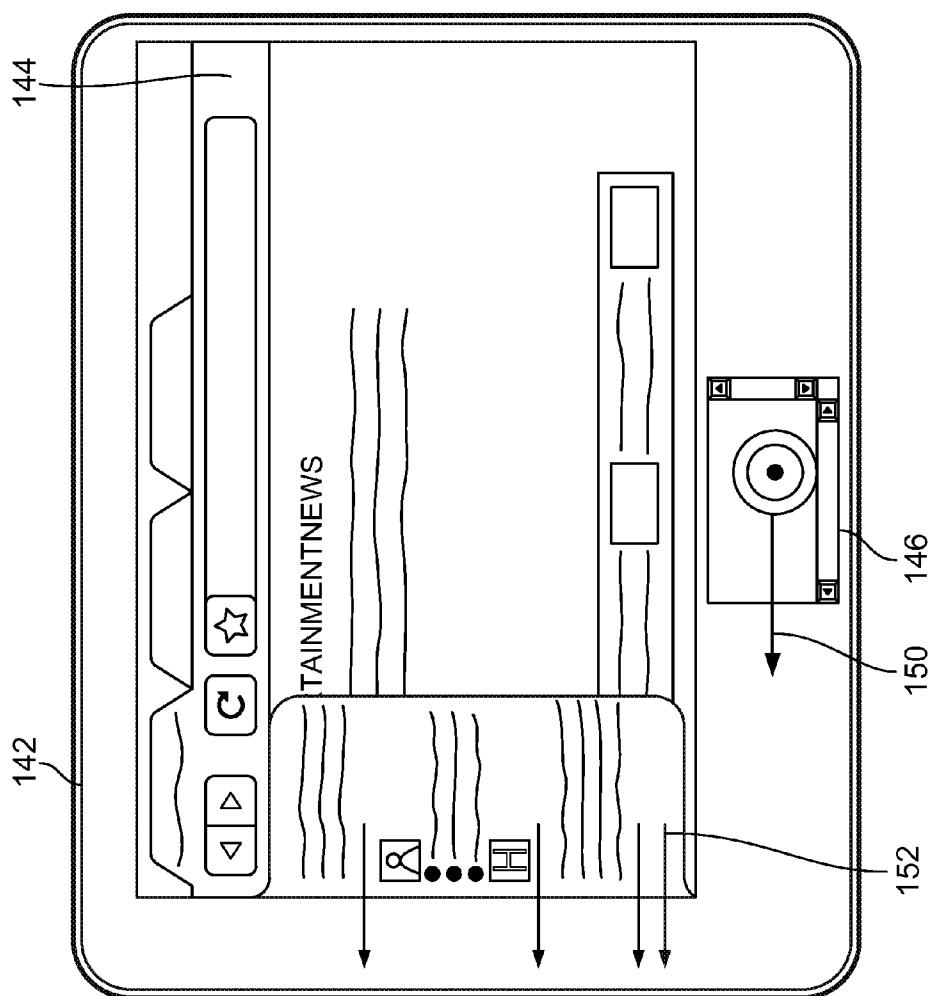


FIG. 4C

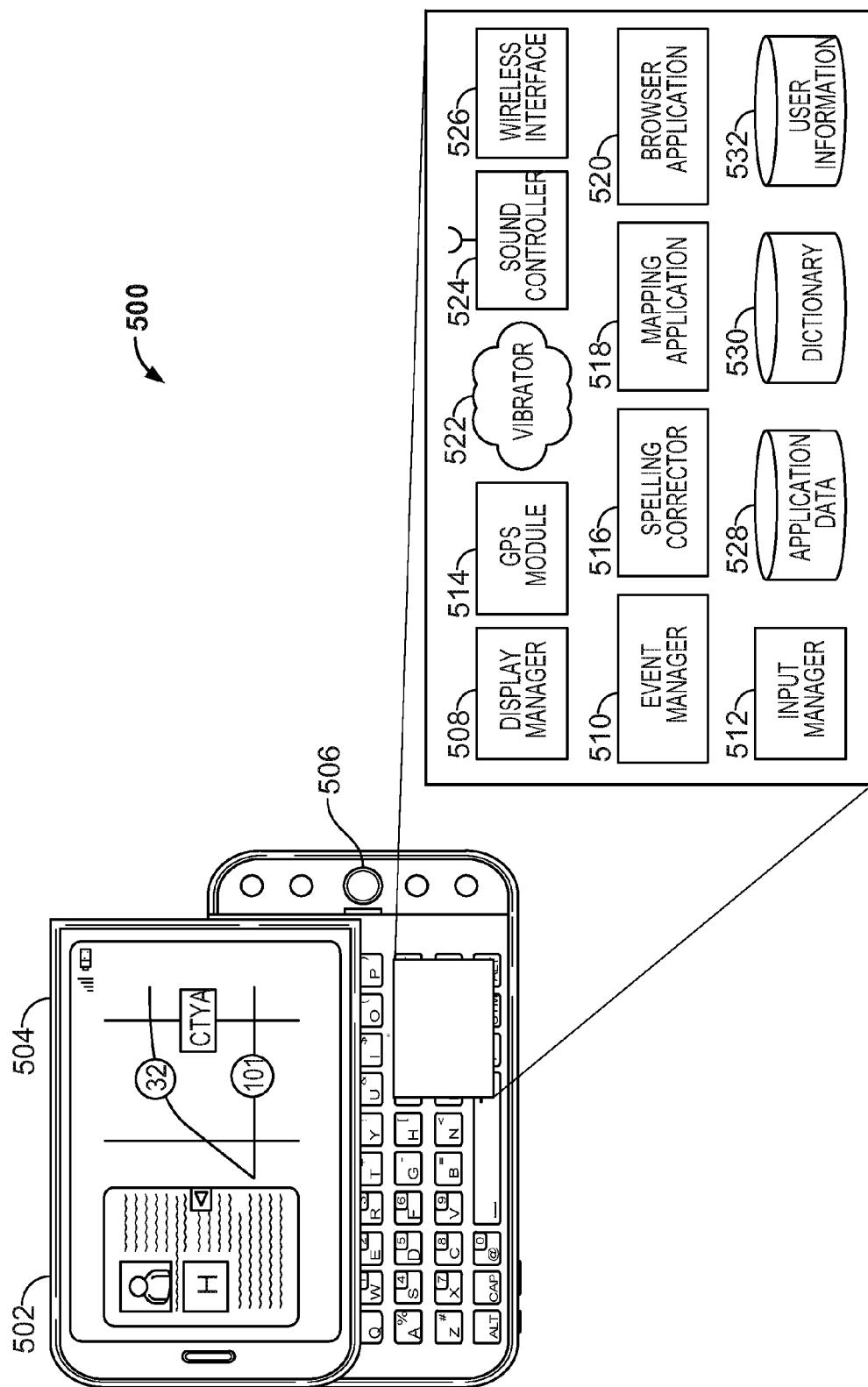


FIG. 5

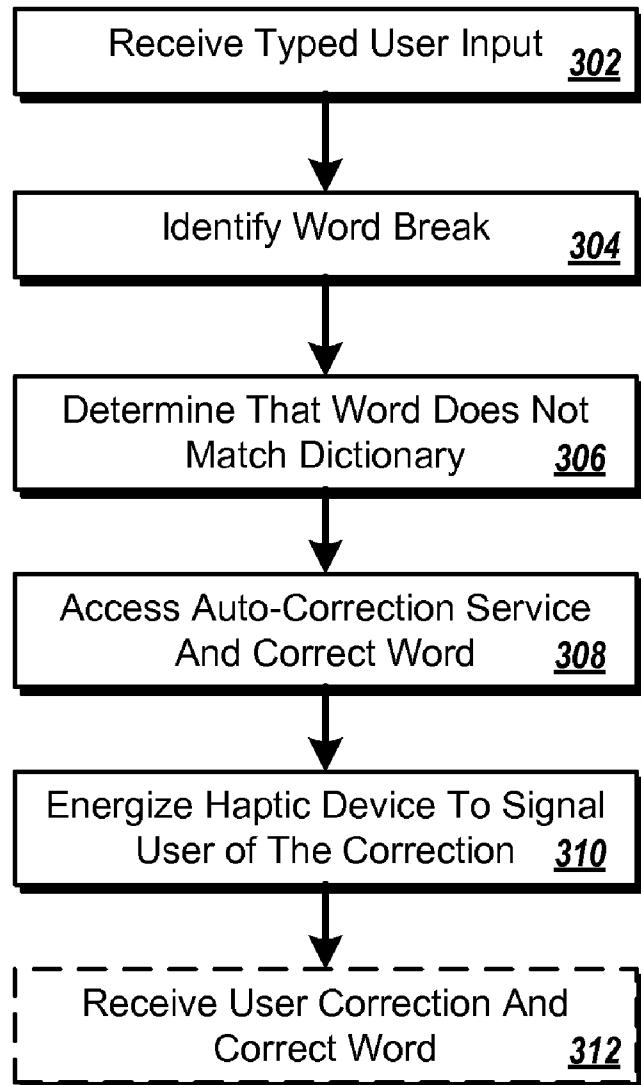


FIG. 6

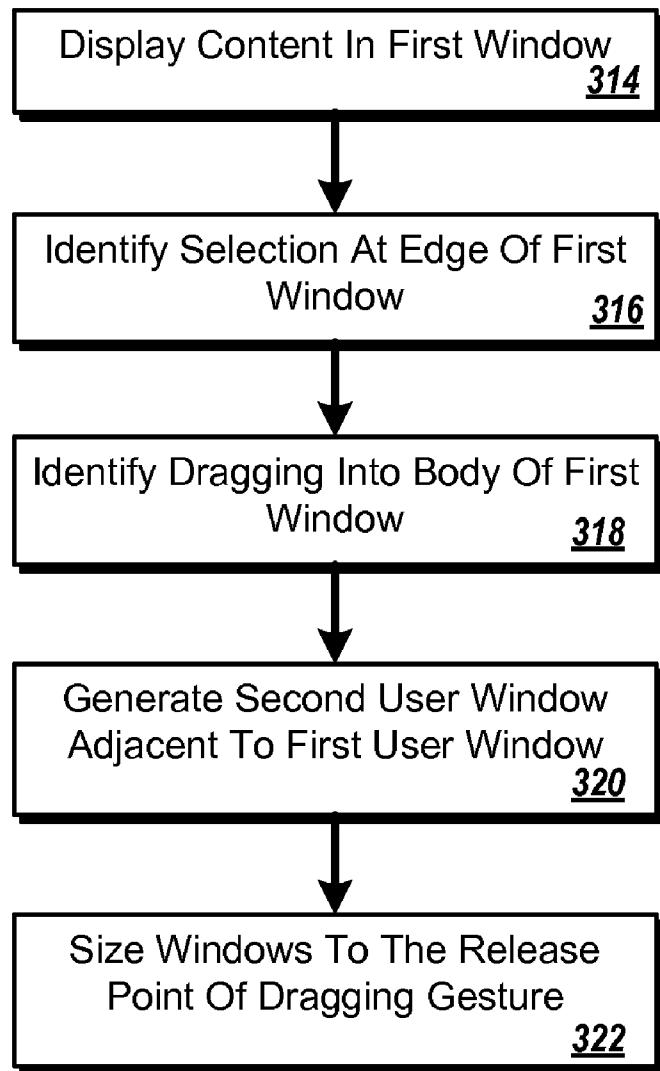


FIG. 7

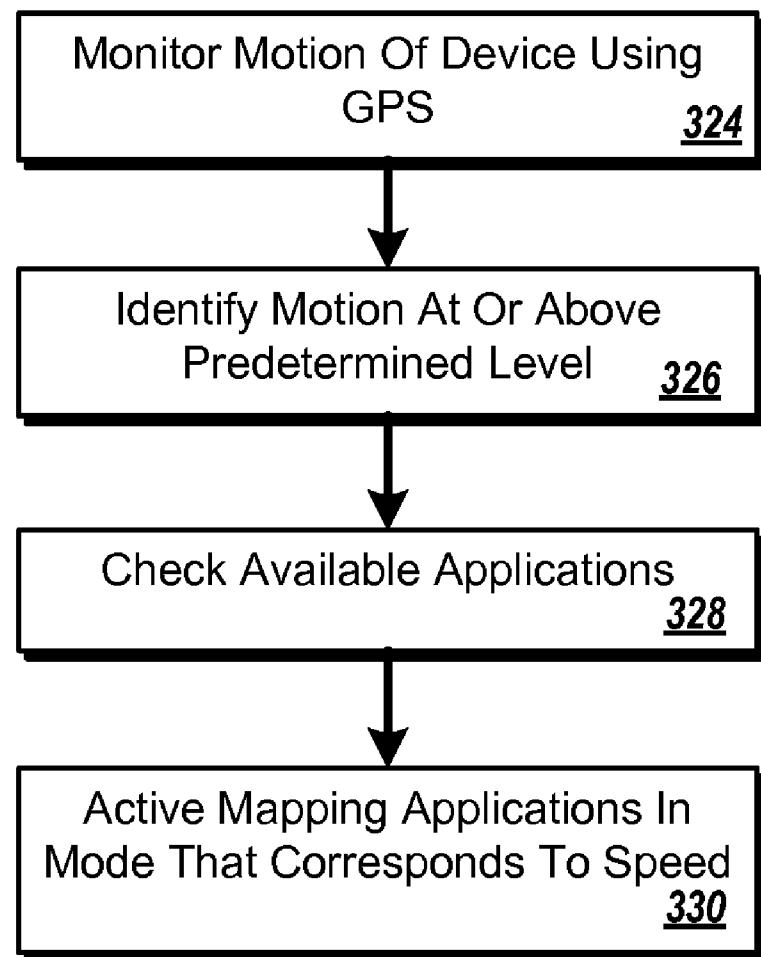


FIG. 8

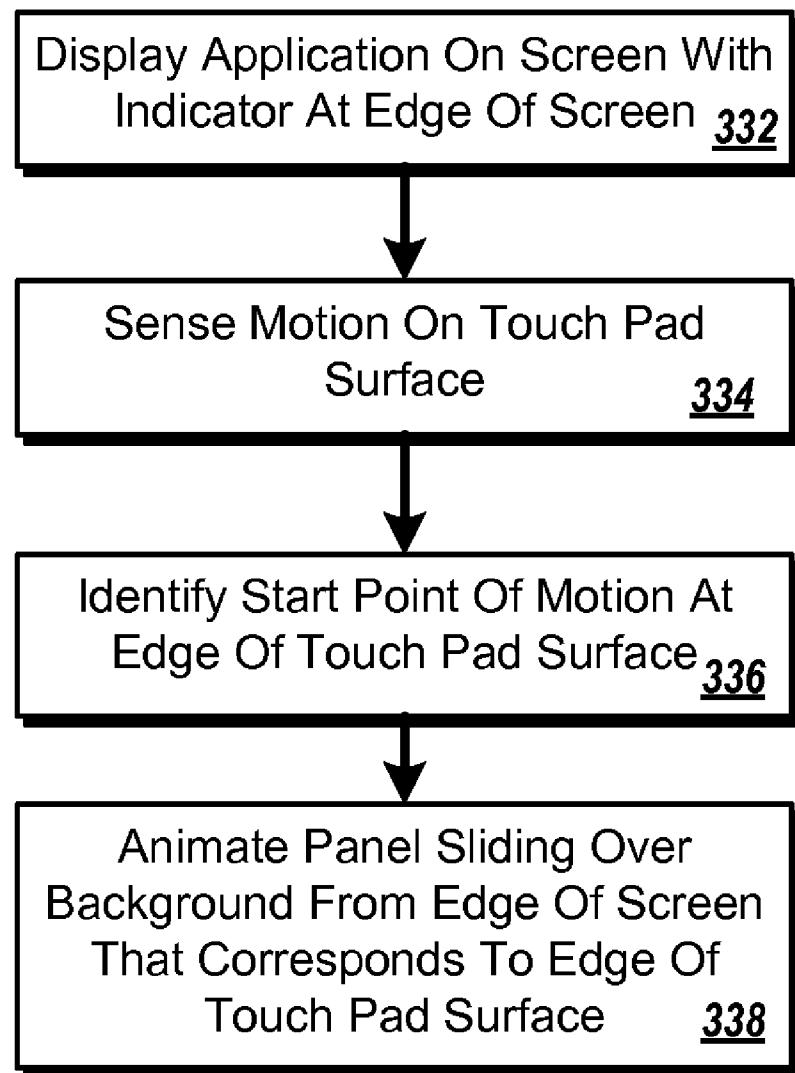


FIG. 9A

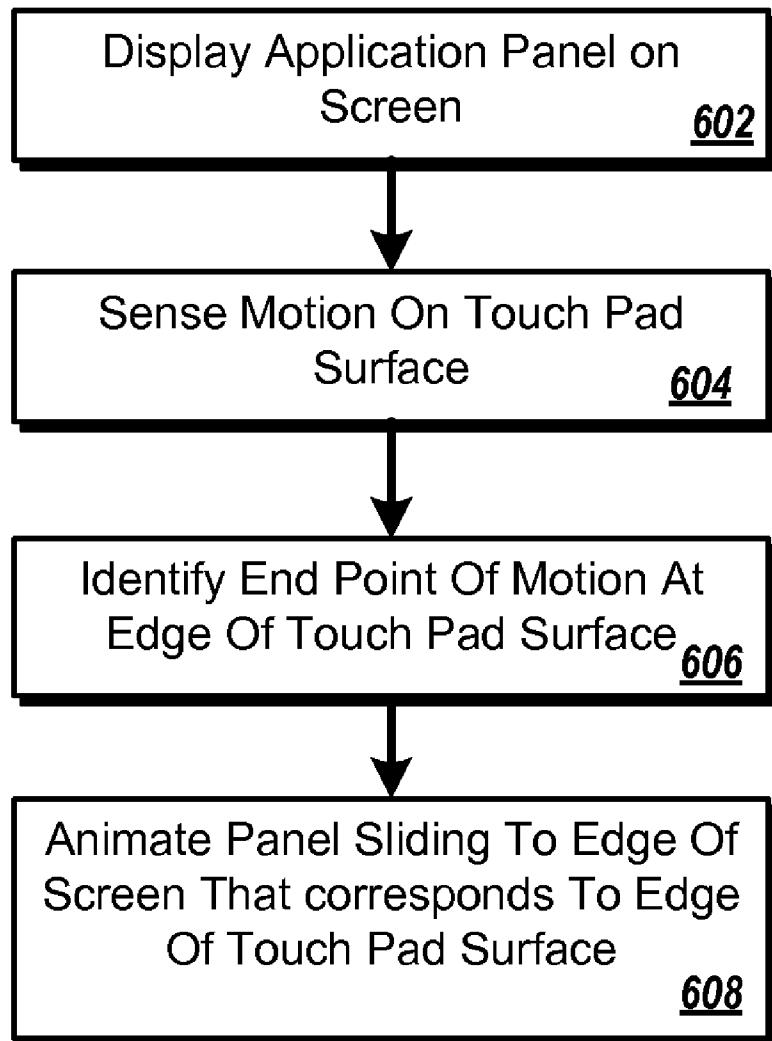
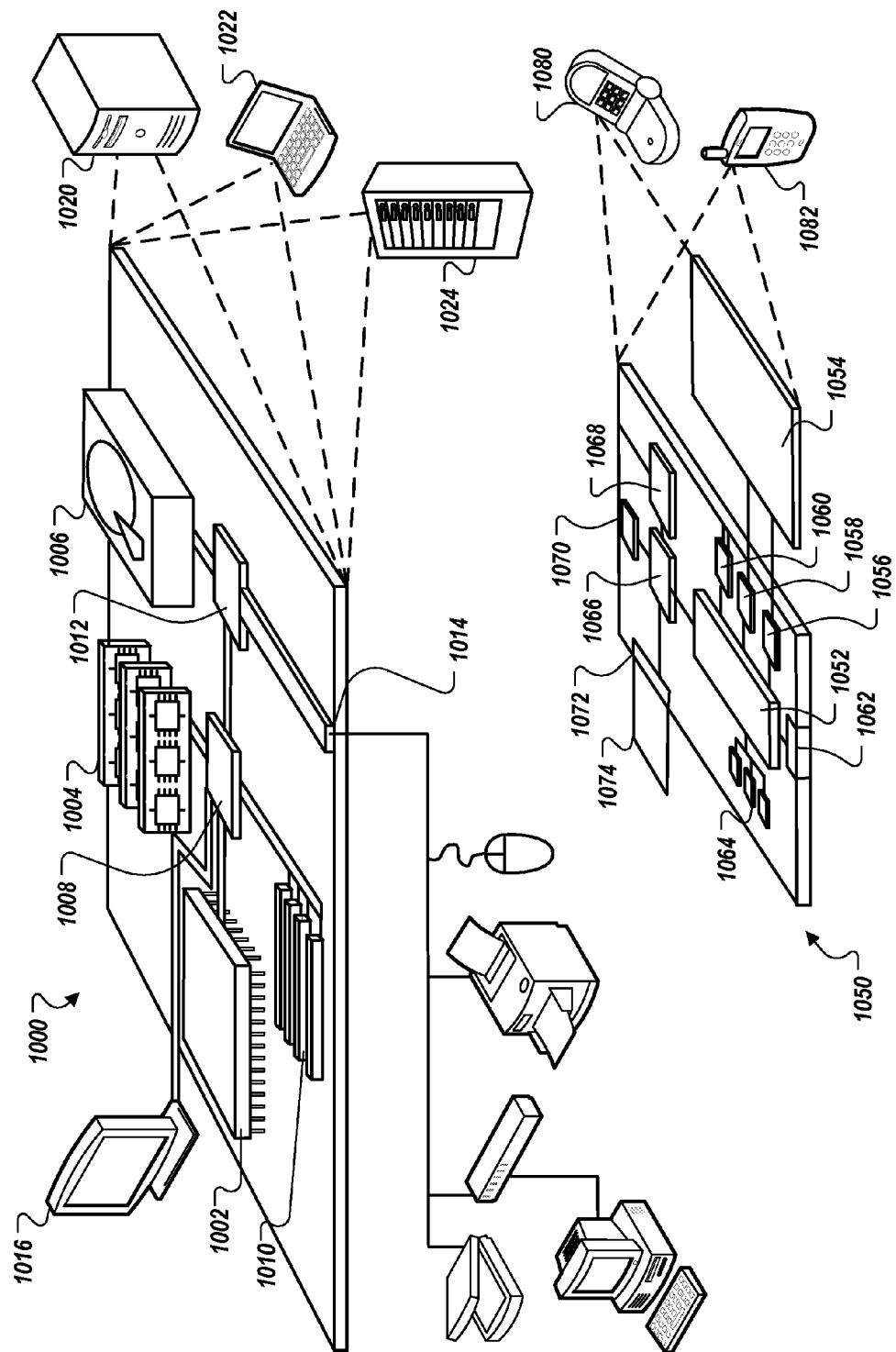


FIG. 9B



SURFACING OFF-SCREEN VISIBLE OBJECTS

CROSS-REFERENCE TO RELATED APPLICATION

[0001] The present application claims priority to U.S. Provisional Application Ser. No. 61/415,179, titled “User Interaction in a Computer Operating System”, filed on Nov. 18, 2010, and U.S. Provisional Application Ser. No. 61/458,326, titled “User Interaction in a Computer Operating System,” filed Nov. 18, 2010, the contents of which are incorporated by reference as if set forth fully herein.

TECHNICAL FIELD

[0002] This document relates to systems and techniques for interacting with users of a computer operating system.

BACKGROUND

[0003] As computers become more present in normal daily activities, the manner in which users interact with computer becomes more important. For example, while early computers used punch cards for interaction with a limited group of users, desktop personal computers led to more prolonged interaction with more users, and to the development of graphical user interfaces. Such interfaces became part of the computer operating systems themselves, and were further extended by applications that ran on the operating systems.

[0004] Computing devices are now frequently portable and carried by a user, such as in the form of smart phones and similar devices. Because such device may be of a limited size that makes interaction via a keyboard less practical than for a desktop or laptop computer, alternative mechanisms for action may be provided, such as in the form of touch screen user interfaces that display graphical content and can also receive user inputs over that content. Other input mechanisms can also be used, such as voice input and gesture-based inputs that can be sensed by accelerometers or compasses in a mobile device.

SUMMARY

[0005] This document describes systems and techniques that may be used to interact with a user of a computing device, such as a mobile computing device (e.g., a smart phone, netbook, or tablet). The various techniques described here can provide mechanisms for more intuitive user interaction with a device, including by signaling the user with the device and for responding to user inputs provided to the device. User feedback can take the form of audible signals, haptic feedback (e.g., coordinated vibrations of the device) and movement of on screen user interface elements.

[0006] In techniques described in particular detail below, particular user interaction with an off-screen touch input device, such as a capacitive touchpad, may result in particular content being shown on a display of the device. In particular, a device may sense that a user started a dragging motion from off the edge of a touchpad and continued the motion onto the touchpad. (No off-pad contact would be sensed, but the touchpad would note that contact began at its edge, and in a direction from the edge toward the main area of the touchpad). In response to such a sensed input, a device may respond by animating a scrolling action of one or more items moving onto a device display from off the edge of the display, as if the user by pulling off the edge of the touchpad and onto the

touchpad, actually pulled content from off the edge of the display and onto the display. For example, if a user drags his or her finger onto the touchpad from off its left edge, content (e.g., a grid of icons, a number of operating gadgets, or a menu) may appear to be dragged progressive onto the display from off the left edge of the display, and if the user moves his or her finger back and forth, the content may move in synchronization with the finger. Similarly, if the user pulls down from off the top of the touchpad, a notification panel may be pulled down in manner that looks like a roller blind being pulled down, similar to the showing of a notification panel in the ANDROID operating system for mobile devices. Thus, dragging motion sensed to start near the edge of the touchpad may be interpreted as a selection of an item already on the screen and may cause the item (e.g., an icon) to be picked up and moved in coordination with the dragging motion, whereas dragging motion sensed to start off the edge of the touchpad (e.g., it starts at the very edge of the pad and has an existing velocity when it is first sensed) can be used to surface a new visible object that was not previously displayed.

[0007] In another example, the entry of off-screen content may cause the current screen (whether including a background of a desktop or just the icons on the desktop) to pan off the display and be replaced by the off-screen content, so as to achieve, on a trackpad device, a multi-screen desktop like that implemented on existing smartphone and tablet operating systems.

[0008] In certain implementations, such systems and technique may provide one or more advantages. For example, user interaction with a computing device may be improved and users may be more likely to purchase more devices and recommend them to others. Also, authors of applications may benefit in being able to provide their applications using interfaces like those discussed here, and may also be able to generate new and additional kinds of applications. In more particular advantages of the techniques described in particular detail below, the effective size of a display may be increased by pushing elements that are needed only periodically, off the edge of the display, and showing such objects in response to a convenient and intuitive user input on a touchpad or other similar input device that is not part of the display. Also, such inputs may be distinguished, in certain examples, from inputs that start at or near the edge of the display but do not first enter the touchpad at the very edge and with an existing velocity to indicate that they started off the edge of the touchpad. As a result, a user can have use of all of the normal functionality of the touchpad, in addition to extra functionality be recognizing that certain inputs started off the edge of the touchpad (e.g., by identifying their start position as at the edge, and their start velocity as having a certain speed and a direction onto the body of the touchpad).

[0009] In another implementation, a computer-implemented user input process for a computing device is disclosed, and comprises receiving, on a touch pad surface over a graphical display, a user input motion dragging across the touch pad surface; identifying the dragging input motion as originating off an edge of the touch pad by identifying a sensed first location for the input motion at a peripheral edge of the touch pad surface; and displaying on the graphical display a sliding graphical element that is animated to move from the edge of the display into a body of the display, over a nonmoving element on the display, in response to identifying the dragging input motion. Receiving a user input motion can comprise receiving an input motion that enters the touch pad

surface from a right or left side of the touch pad surface. The method can also include computing a vector for the dragging input motion.

[0010] In other implementations, a computer-implemented user input process for a computing device is disclosed. The process comprises receiving, on a touch pad surface over a graphical display, a user input motion dragging across the touch pad surface; identifying the dragging input motion as originating off an edge of the touch pad by identifying a sensed first location for the input motion at a peripheral edge of the touch pad surface; and displaying on the graphical display a sliding graphical element that is animated to move from the edge of the display into a body of the display, over a nonmoving element on the display, in response to identifying the dragging input motion.

[0011] The details of one or more embodiments are set forth in the accompanying drawings and the description below. Other features and advantages will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

[0012] FIGS. 1A and 1B are diagrams showing haptic feedback in a mobile computing device.

[0013] FIGS. 2A and 2B show screen shots of a technique for adding windows to a graphical user interface.

[0014] FIGS. 3A and 3B show conceptually the activation of a mapping application on a mobile computing device in response to motion of the device.

[0015] FIGS. 4A, 4B, and 4C show screen shots of a computing device having a touch pad input mechanism and on-screen elements responsive to particular gestures on the touch pad input mechanism.

[0016] FIG. 5 is a schematic diagram of a mobile computing device.

[0017] FIG. 6 is a flow chart of a process for providing haptic feedback in coordination with correcting user entry of data on a computing device.

[0018] FIG. 7 is a flow chart of a process for generating display elements in response to user inputs on a window periphery.

[0019] FIG. 8 is a flow chart of a process for activating a navigational application automatically in response to motion of a computing device.

[0020] FIG. 9A is a flow chart of a process for introducing elements to a graphical display in response to user manipulation of a touch pad.

[0021] FIG. 9B is a flow chart of a process for minimizing elements on a graphical display in response to user manipulation of a touch pad.

[0022] FIG. 10 shows an example of a computer device and a mobile computer device that can be used to implement the techniques described here.

[0023] Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

[0024] This document describes systems and techniques for providing interaction with a user of a computing device, such as a mobile smart phone, using, for example, elements of an operating system that runs on the device. The system and techniques may provide various output mechanisms that can be implemented visually, audibly, or via touch, and input mechanisms for controlling the device. Multiple particular

example techniques are described here, and the techniques can be implemented individually, applied in cooperation and integration with each other (e.g., dragging orthogonally on a scroll bar may occur before or after a user surfaces content by dragging onto a touchpad from off its edge), or together with other aspects of an operating system.

[0025] FIGS. 1A and 1B are diagrams showing haptic feedback in a mobile computing device 102. In general, the device 102 takes the form of a smart phone having a slide out keyboard 108. The device 102 also includes a graphical display on its front surface for displaying information to a user, where the display may include a touch screen input mechanism for obtaining input from a user, such as by the user touching the surface of the display with their finger or with a stylus.

[0026] In this example, the device 102 is shown in a first state, in FIG. 1A, and in a second, later state in FIG. 1B. In the first state, the device 102 is shown after the user has started typing a sentence into the device 102, such as to submit the text of the sentence as a search query, to add the text as a sentence to a word processing document or e-mail, or for other similar purposes. The phrase 106 that has been entered by the user includes a spelling error, in that the word Soain has been entered, but the user clearly intended to type the word Spain. In this example, a real-time spell checking system may determine that the word Soain does not appear in a dictionary that is stored on, or is otherwise accessible to (e.g., via a wireless network and the internet), the device 102, and may highlight the term such as by underlining it, in a manner like that performed by various spellchecking systems in existing word processor applications.

[0027] The identification of a misspelled word, in addition to triggering the system to correct the spelling of the word, may also cause the device 102 to provide haptic feedback to a user. For example, depending on the action that the system takes with respect to the misspelled word, one or more distinct types of haptic feedback may be provided. As one example, the device 102 may vibrate once over a relatively short period, such as for a fraction of a second, if the device has been able to locate a spelling correction for the term, such as term 110, where the word Spain has been identified and automatically placed into the string. Alternatively, if the system is unable to identify a correction to the term 106, the device 102 may be made to vibrate two times in close succession to each other.

[0028] The haptic feedback may thus trigger a user of the device to recognize that a problem has been recognized with the text that they entered so that they move their eyes to the screen, or provide additional focus on the screen. If the user feels a single vibration, they might look up to confirm that the replacement word is correct, whereas if they feel two vibrations, they might look up and stop their typing so as to figure out if a manual correction is needed.

[0029] The device 102 may provide other haptic feedback in other embodiments. For example, a small click may be provided or a low tone may be played that the user can feel in his or her hands. Also, patterns of haptic feedback other than a single vibration and a double vibration may be used. For example, the length, time-wise, of a haptic event may signal some status to the user, or the amplitude of the haptic event may be used. As one example, a light vibration can indicate that a correction has been made and a user can probably ignore it, whereas a strong vibration can indicate that an error was identified but a correction could not be made, so that the user should pay more attention to the problem.

[0030] The haptic feedback can also be provided in response to voice input by a user. In such a situation, the system would not be looking for typing errors by the user, but instead, for situations in which the system has a low confidence that a word it has selected matches the word that the user spoke. Haptic feedback may be more helpful than audible feedback, because a user providing spoken input may not normally be looking at the screen to determine whether his or her words are being understood correctly, and audible feedback may be unnecessarily distracting to the user, as it can be hard to talk and listen at the same time. In contrast, haptic feedback would allow a user to hold his or her device, have it in a pocket or other similar location, and be notified about a problem with any speech-to-text conversion in a fairly non-invasive manner.

[0031] FIGS. 2A and 2B show screen shots of a technique for adding windows to a graphical user interface. In general, the technique here involves identifying a user-initiated dragging motion that starts around a periphery of a window in an operating system's graphical user interface (such as over a scroll bar), and moves into the window. Such a user motion is interpreted as an intent to add a new window, so the system creates a new window stacked to the side of the original window, which the system shrinks in size.

[0032] In FIG. 2A, a window 120 is shown with a basic web browser running in the window 120, and a web page displayed in the browser. Suppose that the user reviewing the web page is studying about a person mentioned on the web page, that the user wants to find out additional information about the person, and that the user wants to review that information in parallel with the information on the web page. One useful way to make such a review would be to provide two windows on a display in a side-by-side or similar arrangement.

[0033] A convenient mechanism by which to establish such an arrangement is shown by selection target 128, which represents the pressing by the user over a scroll bar at a right edge of the display area of the browser. An arrow in the figure from selection target 128 to drop target 130 represents a dragging motion between those two points by the user before the user lifts his or her finger (e.g., from a mouse button or a touch screen) at drop target 130. The selection target 128 can be located over a scrolling control 126 on the scroll bar or at another location on the scroll bar. The selection could also be on another peripheral element near the edge of the window 120 that is outside the display area for the browser in which web page content is displayed. Alternatively, the selection target 128 can be a location within a small distance of the periphery of the window 120, regardless of whether there is a visible element there.

[0034] FIG. 2B shows the result of the computing device's interpretation of such inputs by the user. In particular, the window 120 is shown, and a new window 124 is also shown adjacent to the original window 120. The window 120 has been contracted downward in the lateral dimension to the same extent that the user dragged in FIG. 2A, i.e., the right edge of the window 120 in FIG. 2B is approximately in line, vertically, with drop target 130. Also, a new window 124 has been instantiated automatically in response to the user input shown in FIG. 2A. The window 124 can be a standard, generic opening page for a browser, such a page that displays a search box, a page that displays thumbnails of several of the most-visited sites for the user, or other such content.

[0035] Also, the input from the user in FIG. 2A may indicate what page is to be displayed in window 124 when it is instantiated. For example, the selection target 128 may be on an element on a web page, such as a URL (where the drop target is an edge of the screen or an area outside the pane or tab that is currently the focus of the browser), and the web page for that URL may be opened in window 124.

[0036] Although the example here focuses on web browsers and web pages, the techniques may be used with other applications. For example, a user of a word processing application may make a similar dragging motion and have a new document opened in a new window or pane. The new document may be established as a subdocument, along with the original document, of a larger document that contains both subdocuments. For example, such dragging may be used to open a new chapter in a manuscript for a book, where a document for the entire book contains sub-documents for each chapter.

[0037] FIGS. 3A and 3B show conceptually the activation of a mapping application on a mobile computing device 130 in response to motion of the device 130. FIG. 3A shows the device 130 in a standing automobile 132 along with a screen shot of the device 130, while FIG. 3B shows same automobile 132 moving, along with a screen shot of the device 130, so as to highlight an example mode of operation for the device 130.

[0038] In FIG. 3A, the device 130 is mounted to the automobile 132, such as by being placed in a dock on the dash of the automobile 132. The automobile 132 is not moving, and the device 130 simply displays a desktop for an operating system on the device 130. For example, the device may display one or more icons 138 whose selection may cause various applications on the device 130 to launch, and groups 134 of icons. In addition, widgets or gadgets may be displayed on the desktop, such as a clock 136 or other such program. Thus, the display on device 130 is provided in a familiar manner.

[0039] In FIG. 3B the driver of the automobile 132 has started moving, and a GPS module in the device 130 may, as a result, compute a speed for the device 130, and in turn, for the automobile 132. Such a determination may be made according to an equation like that shown between the stationary and moving representations of the automobile 132, by determining a distance between two points over a time period, and dividing by the time period in order to obtain a rate of travel.

[0040] The device 130 in this example with a program that compares that computed speed to a predetermined speed, and when the predetermined speed is met or exceeded, the device 130 changes its display from the desktop display to a navigational display 140 in the form of a map that show the position of the device 130 and automobile 132 on the map. In this manner, the device can change modes automatically based on an inference that the moving user will want to have information about their up-to-date location.

[0041] Although the display 140 here is shown as providing only a navigational display, other information could be maintained on the display. For example, a strip along the periphery of the display may be reserved for displaying incoming messages (e.g., caller ID information for incoming telephone calls, subjects lines from incoming emails, and the like) and the user may select an area on a message alert to have more information about the corresponding message displayed over the navigational display, or may have it replace the navigational display. Also, when a particular event occurs, such as an incoming telephone call, the navigational display may be

pushed to the side in a split screen, or may be removed entirely, while the user takes the call.

[0042] The type of navigational display shown to a user may also depend on the speed detected by the device 130. For example, at low speeds, the device 130 may infer that the user is walking or riding a bike, and may show a map with detail that is appropriate to such activity, such as local stores, bus stops, and ATMs. At higher speeds, the display may instead show information relevant to the driver of an automobile, such as filling stations, driving directions, and current traffic information. Certain information may be shown in both modes, such as the locations of coffee shops.

[0043] The zoom level of the navigational display may also change automatically in response to changes in speed of the device 130. For example, if the speed is under ten miles per hour so as to indicate that the user is on foot, the zoom level may be set so that the device 130 shows one or two blocks. If the speed is around sixty miles per hour, the zoom level may be reset automatically so that the device 130 shows several miles of range.

[0044] In the manners shown here, then, a navigational display may be made the focus of a device in response to sensed motion of the device, so that a user need not explicitly request such a display. Also, the particular mode of the navigational display may change, depending on the speed that the device is moving.

[0045] FIGS. 4A and 4B show screen shots of a computing device 142 having a touch pad input mechanism and on-screen elements responsive to particular gestures on the touch pad input mechanism. In this example, the device 142 is in the form of a notepad having a touch pad adjacent the touch screen but would normally be in the form of a netbook or laptop that has two elements connected by a hinge, with a graphical display in one of the elements, and a keyboard with touch pad (and corresponding selection buttons) on the other element.

[0046] In FIG. 4A, the device 142 is shown displaying a web page 156 on a graphical display 144. A selectable tab 154 is shown at the left edge of the display 144. The arrow on the tab 154 indicates to a user of the device 142 that he can select the tab 154, such as with a pointer operating by the touch pad or a mouse, and may drag it to the right so as to drag onto the screen one or more objects that are currently (though merely by inference) located off the left side of the display 144.

[0047] Alternatively, and as shown by contact point 148, a user may indicate an intent to pull an object onto the display 144 from an area to the left of the display by contacting the device 142 to the left of touch pad 146 and then dragging his finger onto the surface of the touch pad 146 and a sufficient distance into the surface to register such an intent. Although the user's finger will not be sensed off the edge of the touchpad, its first presence right at the edge of touchpad 146 can be sensed, along with its vectoring at a right angle to the left edge, in a direction onto the touchpad surface. As shown in FIG. 4B, the input from the user to point 150 on the touchpad 146 causes an information panel 152 to be animated so as to slide across the display 144 in coordination with movement of the user's finger on the touchpad. A corresponding swipe, either from off the right edge of the touchpad 146 and onto its surface, or from an area on the touchpad 146 and off the left edge of the touchpad generally in a horizontal and straight swipe, may indicate an intent to remove the information panel 152 from the display 144 so that the device 142 returns to the look of FIG. 4A. Although the movements are described with

respect to left and right edges with horizontal user swiping motions, top and bottom edges with corresponding vertical user swiping motions are also contemplated.

[0048] In some embodiments, the tab 154 shows a first content when located at the edge of the display, as in FIG. 4A. When panel 152 slides onto the display 144, a second content is shown on the panel 152. In some embodiments, the first content is a sub-set of the second content. For example, the first content may be an abbreviated version of the second content as the tab 154 is smaller than the panel 152 and therefore cannot show as much as the panel 152.

[0049] In this implementation of user inputs, it does not matter whether, when the user swipes their finger on the touchpad 146, the corresponding point on the display 144 would be over the tab 154 (in this situation, at the center of the display 144). Instead, the instance of the motion starting off the touchpad 146 is taken by the device as indicating that the tab 154 be grabbed by the motion, regardless of the vertical location of the finger vis-à-vis the touchpad 146.

[0050] The particular location of the point 150 may or may not affect the reaction of the device 142 to the user input. For example, the point 150 may be required to extend far enough into the touchpad 146 to at least indicate that the user is providing a horizontal input across the touchpad 146. The ultimate location of point 150 may also control how far panel 152 slides onto the display 144. That is, the distance the panel 152 slides from the edge of the display 144 to the body of the display 144 may correspond to the length of the user's dragging motion. For example, the panel 152 may have two vertical columns of information, and one vertical column may be displayed if the user drags less than halfway across the touchpad 146 (or some other distance), while both columns may be displayed if the user drags farther. Alternatively, the distance the panel 152 slides from the edge of the display 144 to the body of the display 144 may correspond to the velocity of the user's dragging motion.

[0051] A touch bar 155 is also shown along a left edge of the display 144, and may be optionally provided in combination with the touchpad 146. The touch bar may be off of the graphical display portion of the device 142, on a bezel of the device 142, and may be a touch sensitive device having a relatively low resolution. As shown, in an appropriate circumstance, the touch bar 155 may have a plurality of zones that are each directed to a web browsing operation, such as showing bookmarks, going to a home page, and moving forward or backward. The areas may each be sized so that a user can easily press their thumb against the relevant selection while pinching their forefinger behind the bezel. The area of the touchbar 155 may also include functionality to identify limited motion by a user, such as a motion dragging their thumb from on top of an icon on the touch bar 155, off the edge of the touch bar 155. Such motions may be assigned to particular functions, and may also or alternatively be programmable by applications running on the device 142.

[0052] Thus, using the touch bar 155 in a bezel of the device 142, a user may be provided with additional quick-input mechanisms, but without having to add keys to a keyboard. Also, various other techniques for receiving and interpreting bezel inputs may be used. For example, the bezel selections may include icons for controlling a music player, and the touch bar 155 may wrap around the device 142, so that some of its buttons (e.g., for web browsing or something else that would occur with a laptop top open) are visible from the inside of a device 142, and its other buttons are visible from

outside the device when it is closed (e.g., music player controls that may be accessed without fully powering up the device 142).

[0053] FIG. 5 is a schematic diagram of a mobile computing device, for performing the techniques described above and below. The device is shown here in the form of a smart phone, but may take various other forms, such as a desktop personal computer, a netbook, or a laptop computer. In this example, the mobile device 502 is a slider-based smart phone having a touch screen display 504 for input and output and a trackball 506 as an alternative input mechanism.

[0054] The display 504 shows an example of a mapping application being displayed in a chrome-free browser in a manner like the pages above, and also an information panel to the left of the display 504 and extending inward from the left edge of the display, like that discussed above.

[0055] A number of components are shown schematically inside the device 502 to indicate components in the device that are associated with the features, and used to implement the techniques, discussed in more detail above and below. One or more of the components may be implemented as being stored in memory on the device and operated on a programmable microprocessor on the device 502, though the particular components are shown separately in this example for clarity of explanation. Other components and arrangements of components are also contemplated.

[0056] The device 502 includes a display manager 508 to control various elements to be shown to a user on the touch-screen display 504, while an input manager 512 manages inputs received from the touch-screen on the display 504 or other mechanisms such as trackball 506. The display manager 508 may receive information from various applications and from other components of the operating system, and may determine what elements to display in relation to other elements on the display 504, which elements to hide, and how to show visible elements in relation to each other.

[0057] The input manager 512 may coordinate with the display manager 508 so as interpret actions that a user takes with the device. For example, a user may place their finger on the touch screen display 504 or may move a pointer on the touch screen display 504 and then make a selection. The input manager 512 may refer to the display manager 508 to determine what item the user selected by such an action, and may generate an event that may be reported to various other components that may be subscribing to such an event, such as by using event manager 510.

[0058] The device 502 may also be provided with various services that may be made available to applications running on the device and/or to the operating system of the device. For example, a GPS module 514, which may be a common commercially-obtained module, may make determinations about a location of the device 504 and may provide such information to various applications, such as applications that provide driving directions, that provide the user's location for reference by friends of the user through a social network, and the like. Such a service may be referenced in relation to techniques like those discussed with respect to FIGS. 3A and 3B above, and FIG. 8 below. In a similar manner, a spelling corrector 516 may provide a service that is generally available to various applications running on the device, so that those applications can correct data that the user has input or otherwise notify the user about problems with the input. Such notification may be via haptic feedback, like that discussed with respect to FIGS. 1A and 1B above, and FIG. 6 below.

[0059] Various physical output mechanisms may also be provided in the device 502. For example, a vibrator 522 may be provided to give haptic feedback to a user. The vibrator 522 may have an associated controller (not shown) that can be called by various applications or services on the device. For example, the spelling corrector 516 may cause the device 504 to vibrate whenever a spelling error is identified by the spelling corrector 516. Also, a sound controller 524 and related audio speaker may be provided to give a user additional feedback, such as in the form of tones or synthesized speech.

[0060] Various applications such as applications 518, 520 may run on the device and may provide data for graphics to be displayed through the display manager 512. The applications may be part of an operating system on the device 502, or may be added by a user of the device 502, such as from an on-line marketplace. In this example, a first application is a mapping application 518 and may run as a standalone application or an application or plug-in that runs in browser application 520. The browser application 520 may be a standalone application or a component that is built into an operating system that runs on the device 502, and may generate displays like those described in detail elsewhere in this document. Other applications may also be provided or installed on the device 502, including by a user of the device 502.

[0061] Various messages for the operating system and the applications 518, 520, may be received over a network such as a wireless communications network connected to the internet, via a wireless interface 526. Information received from the network, such as from one or more remote servers, may be provided to the applications 518, 520, and information may be passed from the applications 518, 520 back to the network.

[0062] In addition, various forms of storage, including volatile storage and persistent storage, may be provided on the device 502. The persistent storage may include various data stores, such as application data 528, dictionary data 530, and user information 532. The dictionary data 530, as described above, coordinates with and is used by the spelling corrector 516, to provide information that permits the spelling corrector 516 to determine whether terms entered by a user are properly spelled. The application data 528 may include whatever data the various applications 518, 520, require in order to perform their functions. Depending on the application, more or less data may be stored locally on the device 502 as opposed to being stored on a remote server system. For example, certain devices may store data for a mapping application locally, whereas others may obtain mapping data (e.g., map tiles that show images of a portion of a geographic area) from a remote server as a user navigates on a map, and may dump the data soon after.

[0063] The user information 532 may include specific information about one or more users of the computing device 502. For example, the user information may include profile information for the user, as well as settings information that controls the interaction of the device 502 with the user, and that can also control the manner in which various applications operate on the device 502. Although three various data stores are shown here for clarity, much other information would be and can be stored on device 502, and need not be arranged as shown here.

[0064] FIG. 6 is a flow chart of a process for providing haptic feedback in coordination with correcting user data entry on a computing device. In general, the process involves automatically shaking or vibrating a mobile device in order to

indicate to a user that a successful or an unsuccessful change (on non-change) has been made to information being input by the user.

[0065] The process begins at box 302, where a computing device receives a typed user input. Such an input may occur through one or more applications running on the device, such a word processing application, e-mail application, web browser applications, and other similar applications. The input may also be spoken input that is converted to text for the application, such as by a speech-to-text convertor. At box 304, the process identifies a word break in the input text, such as a space entered by a user, or the end of a word as determined by the speech-to-text converter. At box 306, a spelling correction application analyzes the entered word, or a phrase made up of multiple words, and determines that the word does not match any word in a dictionary that is stored on the device, or made accessible by the device. Such a determination may be made simply by comparing a word to a list of words in a dictionary or by comparing word in context to better determine the proper spelling of the word.

[0066] At box 308, the process accesses an auto-correction service to correct the word. Such a service may be implemented in a variety of familiar forms, so as to use dictionary data to determine a proper word spelling, or to determine what word a speaking user spoke.

[0067] At box 310, the process energizes a haptic device to signal to a user correction of the word. Such signaling may involve energizing the haptic device so that the user lightly feels vibration. After using the device for awhile, the user will inherently understand that the vibration indicates a correction to data being entered on the device. A stronger vibration, or two instances of vibration in close succession, may be used to indicate that the device could not correct the spelling of the term, or could not recognize the term spoken to the device. Other similar haptic feedback mechanisms may also be provided in response to identifying errors in user input, either when the errors in the text correct automatically or when they correct automatically, with a less intrusive signal when corrections can be made automatically, and a more serious signal when they cannot be corrected properly or a word cannot be determined. In the latter case, the more serious signal may cause the user to stop what they are doing and pay more attention to the device to make a manual correction of their input.

[0068] Box 312 shows such a user's manual correction of a word. The box is shown in dashed format because a user need not make such a correction, and can ignore their device, even when the haptic feedback indicating device could not make the correction automatically, such as when the user does not want to stop their input at the current time, and intends to make corrections after they finish entering data.

[0069] In this manner, a computing device may conveniently provide input back to a user who is entering data to the device. The alerting mechanism may be low key, in that it may be provided so as not to interrupt the user is entering the information, and the user need not stop the entry of information to be so notified. The user may thus continuing entering information if they choose, or to address a problem in their entry of information if they choose.

[0070] FIG. 7 is a flow chart of a process for generating display elements in response to user inputs on a window periphery. In general, the process involves opening a new window or similar on-screen elements on a graphical user interface in response to a user selection near a periphery of the

window that is already displayed on the interface. A user selection may begin at a periphery of a window and slide the interior portion of window, essentially indicating a user intent to slide one side of the window to the side, and also to provide a new window in the area vacated by the first open window. In this document, the term window is used to identify a graphical on-screen element that is set off from elements around it by a border and where selections inside the border represent user actions with respect to a particular application.

[0071] The process begins at box 314, where content is displayed in a first window in a graphical user interface. The content may be a web page in a web browser, a document in a word processing application, or of another similar form. In certain instances, a scroll bar may be displayed in a familiar manner along one side of the content, or along the top or bottom of the content, to permit accelerated movement through the content. At box 316, the process identifies a selection at an edge of the first window, such as in an area where a scroll bar is displayed. The selection may occur in a particular area of the scroll bar, such as where a scroll button is displayed in the scroll bar. At box 318, process identifies a dragging into a body of the first window. In particular, the process may determine that a user has first selected along a periphery of the window, such as along a periphery of the graphical display itself, and has dragged inward into a body of a document, such as by a horizontal dragging from left to right or right to left.

[0072] In response to identification of such dragging, the process at box 320 may generate a second user window adjacent to the first user window. Such generation of a second window may occur after the process has contracted the first window to make room on the display or the second window. Such generation of a second window may also await a user release of a pointer button, so that the user can initially slide inward and see outlined images of the windows displayed while they move so that they can see the future size of the two windows as they move.

[0073] A contraction of the first window, and the positioning of the second window, may be placed in a variety of locations, and one such location may be at a point at which the user releases their selection button after starting their dragging motion at a periphery of the window. For example, if a first window fills the entire display, and a user drags from a right hand scroll bar of the window halfway across the display to the left, the first window may fill the left half of the display after such an operation, and the new window may fill a right half of the display, as shown at box 322.

[0074] The information shown in the new window may take a variety of forms. For example, a blank document may be initially displayed in the second window, such as a blank word processing document or a generic first screen for a web browser, which shows a number of prior websites the user has visited or a search box in which the user may enter one or more search query. The new window may alternatively display information related to what is being displayed in the first window, such as a target of a web page link. In such a manner, a user may click on a link on a first page and drag to the periphery of the page or into an area out of the window, and order to open a new window that automatically displays the target of the link. Alternatively, the user motion may start at a periphery of the window and drag to a link on a page in order to open the target of the link in a new window (which may be a tab in a tabbed browser).

[0075] FIG. 8 is a flow chart of a process for activating a navigational application automatically in response to motion of a computing device. In general, the process involves changing a display on a computing device, and generally a mobile computing device, automatically in response to changes in speed of the device.

[0076] The process begins at box 324, where a process monitors motion of a device that uses GPS functionality or other functionality such as by triangulation among cellphone towers or WiFi hot spots, among other techniques. The monitoring may occur in an event handler on the device as part of the device operating system, where various applications or other processes may register an intent to be notified when information is generated by the GPS module of the device.

[0077] At box 326, the process identifies motion at or above a predetermined level. Such a determination may be made by an event handler itself, so that a notification is provided to an application only when the speed is above a certain level. Alternatively, an application in accordance with the process described here may receive any GPS data being generated by device, and may make its own determination that a speed of the device has exceeded a predetermined level.

[0078] At box 328, the process checks available applications, which may be applications that expressed an intent to be notified when the speed of the device exceeds a certain speed. One such application may include a mapping application that may be programmed to automatically instantiate itself and become the focus of the device when the speed of the device exceeds a certain predetermined level, such as a level to indicate that a user is walking, biking, or driving from one place to another. At box 330, the process activates the mapping application, or another application, in a mode that corresponds to the speed determined by the GPS unit. For example, a first mode may relate to a low-speed motion, such as walking by a user, so that the mode displays information that is relevant to a walking user. A second mode may be triggered in response to speeds that indicate driving by the user, and may display information with a mapping application that is relevant to a driver, such as the location of filling stations and fast food restaurants near a road on which the device is moving.

[0079] In this manner, the techniques shown here may be used to automatically change a focus of a mobile device to applications that are relevant to a user who is moving with their mobile device. As a result, the user may receive information without having to expressly ask for the information, and potentially without having to remove the user's attention from walking or driving, as the case may be.

[0080] FIG. 9A is a flow chart of a process for introducing elements to a graphical display in response to user manipulation of a touch pad. In general, the process involves identifying a user motion from off the surface of a touch input mechanism and into an area on the touch input mechanism, where the start of motion is inferred as being off the mechanism by the fact that the mechanism first senses contact or motion at its very edge.

[0081] The process begins at box 332, where the device displays an application on a screen and an indicator at the edge of the screen. In this example, the indicator at the edge of the screen visually identifies to the user of the device that a normally visible element is being docked off the edge of the display of a graphical user interface. The element in this example is shown with an arrow on it so as to naturally indicate to a user that selection of the element will create

some sort of motion in the direction of the arrow. Other mechanisms may also be used to indicate to a user of a graphical element is, in a sense, currently located off an edge of the graphical display.

[0082] At box 334, the process receives motion on a touch pad surface of the device, and at box 336, the device identifies a start point of the motion at an edge of the touch pad surface. The device may also identify that the motion had a velocity at the edge of the touchpad, and thus must have started off the edge of the touchpad. A device may use such determinations to identify that the motion started off the touch pad on the edge where contact is first identified as occurring, and that the input moved onto the touch pad from where it started. The device may then interpret such a motion as involving a user intent to slide an item from off a graphical user display and onto the display, from a side of the display that corresponds to the side of the touchpad from which the user slid his or her finger onto the touch pad. Such a determination may be made even if the point of contact on the touchpad would not correspond to the position on the display of the tab or other element that indicates to a user that there is an off-screen element to be brought onto the display.

[0083] At box 338, the process animates a panel sliding over the background and/or over other displayed applications on the display, from the edge of the screen that corresponds to the edge of the touchpad on which the user's input entered the touchpad. The animation may be a simple sliding into view of the element, or a more complex motion. Also, the visual element that is animated into view may take a variety of forms, such as a ribbon, panel, or toolbar. The user may hide the element again by swiping from a location on the touchpad off the edge of the touchpad, as shown in FIG. 9B.

[0084] FIG. 9B is a flow chart of a process for removing elements from a graphical display or decreasing the size of elements in a graphical display in response to user manipulation of a touch pad. In general, the process involves identifying a user motion from the surface of a touch input mechanism that terminates at an edge of the touch input mechanism, where the end of the motion is inferred as being off the mechanism by the fact that the contact last sensed by the mechanism is motion on or at its very edge.

[0085] The process begins at box 602, where the device displays an application panel on a screen. The application may be fully maximized on the screen or only partially displayed. In most embodiments, the application is displayed near the middle of the screen.

[0086] At box 604, the process receives motion on a touch pad surface of the device, and at box 606, the device identifies an end, or terminating, point of the motion at an edge of the touch pad surface. The device may also identify that the motion had a decelerating velocity at the edge of the touchpad, and thus must have terminating at or near the edge of the touchpad. A device may use such determinations to identify that the motion terminated off the touch pad on the edge where contact is last identified as occurring, and that the input moved from the touch pad from where it started. The device may then interpret such a motion as involving a user intent to slide an item from the graphical user display off the display or to a side of the display that corresponds to the side of the touchpad onto which the user slid his or her finger on the touch pad. Such a determination may be made even if the point of contact on the touchpad would not correspond to the

position on the display of the tab or other element that indicates to a user that there is an on-screen element to be taken off the display.

[0087] At box 608, the process animates a panel sliding to the edge of the touch pad surface, revealing the background and/or other displayed applications that were previously covered by the panel. The panel is animated to slide to the edge of the screen that corresponds to the edge of the touchpad where the user's input terminated on the touchpad. The animation may be a simple sliding of the element from view, or a more complex motion. Also, the visual element that is animated into view may take a variety of forms, such as a ribbon, panel, or toolbar.

[0088] By this process, then, a device may provide a convenient mechanism by which to extend a display, so that certain elements may be moved easily off the actual display and then moved back onto it. A user may quickly retrieve additional content without having to carefully position a pointer or other item, but may instead quickly swipe in a general area around a touch pad or similar input mechanism.

[0089] FIG. 10 shows an example of a generic computer device 400 and a generic mobile computer device 450, which may be used with the techniques described here. Computing device 400 is intended to represent various forms of digital computers, such as laptops, desktops, workstations, personal digital assistants, servers, blade servers, mainframes, and other appropriate computers. Computing device 450 is intended to represent various forms of mobile devices, such as personal digital assistants, cellular telephones, smartphones, and other similar computing devices. The components shown here, their connections and relationships, and their functions, are meant to be exemplary only, and are not meant to limit implementations of the inventions described and/or claimed in this document.

[0090] Computing device 400 includes a processor 402, memory 404, a storage device 406, a high-speed interface 408 connecting to memory 404 and high-speed expansion ports 410, and a low speed interface 412 connecting to low speed bus 414 and storage device 406. Each of the components 402, 404, 406, 408, 410, and 412, are interconnected using various busses, and may be mounted on a common motherboard or in other manners as appropriate. The processor 402 can process instructions for execution within the computing device 400, including instructions stored in the memory 404 or on the storage device 406 to display graphical information for a GUI on an external input/output device, such as display 416 coupled to high speed interface 408. In other implementations, multiple processors and/or multiple buses may be used, as appropriate, along with multiple memories and types of memory. Also, multiple computing devices 400 may be connected, with each device providing portions of the necessary operations (e.g., as a server bank, a group of blade servers, or a multi-processor system).

[0091] The memory 404 stores information within the computing device 400. In one implementation, the memory 404 is a volatile memory unit or units. In another implementation, the memory 404 is a non-volatile memory unit or units. The memory 404 may also be another form of computer-readable medium, such as a magnetic or optical disk.

[0092] The storage device 406 is capable of providing mass storage for the computing device 400. In one implementation, the storage device 406 may be or contain a computer-readable medium, such as a floppy disk device, a hard disk device, an optical disk device, or a tape device, a flash memory or other

similar solid state memory device, or an array of devices, including devices in a storage area network or other configurations. A computer program product can be tangibly embodied in an information carrier. The computer program product may also contain instructions that, when executed, perform one or more methods, such as those described above. The information carrier is a computer- or machine-readable medium, such as the memory 404, the storage device 406, memory on processor 402, or a propagated signal.

[0093] The high speed controller 408 manages bandwidth-intensive operations for the computing device 400, while the low speed controller 412 manages lower bandwidth-intensive operations. Such allocation of functions is exemplary only. In one implementation, the high-speed controller 408 is coupled to memory 404, display 416 (e.g., through a graphics processor or accelerator), and to high-speed expansion ports 410, which may accept various expansion cards (not shown). In the implementation, low-speed controller 412 is coupled to storage device 406 and low-speed expansion port 414. The low-speed expansion port, which may include various communication ports (e.g., USB, Bluetooth, Ethernet, wireless Ethernet) may be coupled to one or more input/output devices, such as a keyboard, a pointing device, a scanner, or a networking device such as a switch or router, e.g., through a network adapter.

[0094] The computing device 400 may be implemented in a number of different forms, as shown in the figure. For example, it may be implemented as a standard server 420, or multiple times in a group of such servers. It may also be implemented as part of a rack server system 424. In addition, it may be implemented in a personal computer such as a laptop computer 422. Alternatively, components from computing device 400 may be combined with other components in a mobile device (not shown), such as device 450. Each of such devices may contain one or more of computing device 400, 450, and an entire system may be made up of multiple computing devices 400, 450 communicating with each other.

[0095] Computing device 450 includes a processor 452, memory 464, and an input/output device such as a display 454, a communication interface 466, and a transceiver 468, among other components. The device 450 may also be provided with a storage device, such as a microdrive or other device, to provide additional storage. Each of the components 450, 452, 464, 454, 466, and 468, are interconnected using various busses, and several of the components may be mounted on a common motherboard or in other manners as appropriate.

[0096] The processor 452 can execute instructions within the computing device 450, including instructions stored in the memory 464. The processor may be implemented as a chipset of chips that include separate and multiple analog and digital processors. The processor may provide, for example, for coordination of the other components of the device 450, such as control of user interfaces, applications run by device 450, and wireless communication by device 450.

[0097] Processor 452 may communicate with a user through control interface 458 and display interface 456 coupled to a display 454. The display 454 may be, for example, a TFT LCD (Thin-Film-Transistor Liquid Crystal Display) or an OLED (Organic Light Emitting Diode) display, or other appropriate display technology. The display interface 456 may comprise appropriate circuitry for driving the display 454 to present graphical and other information to a user. The control interface 458 may receive commands from

a user and convert them for submission to the processor **452**. In addition, an external interface **462** may be provided in communication with processor **452**, so as to enable near area communication of device **450** with other devices. External interface **462** may provide, for example, for wired communication in some implementations, or for wireless communication in other implementations, and multiple interfaces may also be used.

[0098] The memory **464** stores information within the computing device **450**. The memory **464** can be implemented as one or more of a computer-readable medium or media, a volatile memory unit or units, or a non-volatile memory unit or units. Expansion memory **474** may also be provided and connected to device **450** through expansion interface **472**, which may include, for example, a SIMM (Single In Line Memory Module) card interface. Such expansion memory **474** may provide extra storage space for device **450**, or may also store applications or other information for device **450**. Specifically, expansion memory **474** may include instructions to carry out or supplement the processes described above, and may include secure information also. Thus, for example, expansion memory **474** may be provided as a security module for device **450**, and may be programmed with instructions that permit secure use of device **450**. In addition, secure applications may be provided via the SIMM cards, along with additional information, such as placing identifying information on the SIMM card in a non-hackable manner.

[0099] The memory may include, for example, flash memory and/or NVRAM memory, as discussed below. In one implementation, a computer program product is tangibly embodied in an information carrier. The computer program product contains instructions that, when executed, perform one or more methods, such as those described above. The information carrier is a computer- or machine-readable medium, such as the memory **464**, expansion memory **474**, memory on processor **452**, or a propagated signal that may be received, for example, over transceiver **468** or external interface **462**.

[0100] Device **450** may communicate wirelessly through communication interface **466**, which may include digital signal processing circuitry where necessary. Communication interface **466** may provide for communications under various modes or protocols, such as GSM voice calls, SMS, EMS, or MMS messaging, CDMA, TDMA, PDC, WCDMA, CDMA2000, or GPRS, among others. Such communication may occur, for example, through radio-frequency transceiver **468**. In addition, short-range communication may occur, such as using a Bluetooth, WiFi, or other such transceiver (not shown). In addition, GPS (Global Positioning System) receiver module **470** may provide additional navigation- and location-related wireless data to device **450**, which may be used as appropriate by applications running on device **450**.

[0101] Device **450** may also communicate audibly using audio codec **460**, which may receive spoken information from a user and convert it to usable digital information. Audio codec **460** may likewise generate audible sound for a user, such as through a speaker, e.g., in a handset of device **450**. Such sound may include sound from voice telephone calls, may include recorded sound (e.g., voice messages, music files, etc.) and may also include sound generated by applications operating on device **450**.

[0102] The computing device **450** may be implemented in a number of different forms, as shown in the figure. For example, it may be implemented as a cellular telephone **480**.

It may also be implemented as part of a smartphone **482**, personal digital assistant, or other similar mobile device.

[0103] Various implementations of the systems and techniques described here can be realized in digital electronic circuitry, integrated circuitry, specially designed ASICs (application specific integrated circuits), computer hardware, firmware, software, and/or combinations thereof. These various implementations can include implementation in one or more computer programs that are executable and/or interpretable on a programmable system including at least one programmable processor, which may be special or general purpose, coupled to receive data and instructions from, and to transmit data and instructions to, a storage system, at least one input device, and at least one output device.

[0104] These computer programs (also known as programs, software, software applications or code) include machine instructions for a programmable processor, and can be implemented in a high-level procedural and/or object-oriented programming language, and/or in assembly/machine language. As used herein, the terms "machine-readable medium" "computer-readable medium" refers to any computer program product, apparatus and/or device (e.g., magnetic discs, optical disks, memory, Programmable Logic Devices (PLDs)) used to provide machine instructions and/or data to a programmable processor, including a machine-readable medium that receives machine instructions as a machine-readable signal. The term "machine-readable signal" refers to any signal used to provide machine instructions and/or data to a programmable processor.

[0105] To provide for interaction with a user, the systems and techniques described here can be implemented on a computer having a display device (e.g., a CRT (cathode ray tube) or LCD (liquid crystal display) monitor) for displaying information to the user and a keyboard and a pointing device (e.g., a mouse or a trackball) by which the user can provide input to the computer. Other kinds of devices can be used to provide for interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback (e.g., visual feedback, auditory feedback, or tactile feedback); and input from the user can be received in any form, including acoustic, speech, or tactile input.

[0106] The systems and techniques described here can be implemented in a computing system that includes a back end component (e.g., as a data server), or that includes a middleware component (e.g., an application server), or that includes a front end component (e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the systems and techniques described here), or any combination of such back end, middleware, or front end components. The components of the system can be interconnected by any form or medium of digital data communication (e.g., a communication network). Examples of communication networks include a local area network ("LAN"), a wide area network ("WAN"), and the Internet.

[0107] The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other.

[0108] A number of embodiments have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of

the invention. For example, much of this document has been described with respect to television advertisements, but other forms of future, viewership-based advertisements may also be addressed, such as radio advertisements and on-line video advertisements.

[0109] In addition, the logic flows depicted in the figures do not require the particular order shown, or sequential order, to achieve desirable results. In addition, other steps may be provided, or steps may be eliminated, from the described flows, and other components may be added to, or removed from, the described systems. Accordingly, other embodiments are within the scope of the following claims.

What is claimed is:

1. A computer-implemented user input process for a computing device, the process comprising:

receiving, on a touch pad surface on a computing device, a user input motion dragging across the touch pad surface, the computing device having a graphical display;

identifying the dragging input motion as originating off an edge of the touch pad by identifying a sensed first location for the input motion at a peripheral edge of the touch pad surface; and

displaying on the graphical display a sliding graphical element that is animated to move from the edge of the display into a body of the display, over a nonmoving element on the display, in response to identifying the dragging input motion, wherein in the touch pad surface is separate from the graphical display.

2. The method of claim **1**, further comprising mapping the user input motion received on the touch pad to corresponding points on the graphical display.

3. The method of claim **1**, wherein the sliding graphical element moves in a direction corresponding to the user input motion.

4. The method of claim **1**, wherein receiving a user input motion comprises receiving an input motion that enters the touch pad surface from a right or left side of the touch pad surface.

5. The method of claim **1**, wherein receiving a user input motion comprises receiving an input motion that enters the touch pad surface from a top or bottom side of the touch pad surface.

6. The method of claim **1**, further comprising computing a vector for the dragging input motion.

7. The method of claim **1**, wherein the distance the sliding graphical element moves from the edge of the display into the body of the display corresponds to a length of the user input dragging motion.

8. The method of claim **1**, wherein the distance the sliding graphical element moves from the edge of the display into the body of the display corresponds to a velocity of the user input dragging motion.

9. The method of claim **1**, wherein the sliding graphical element displays a first content prior to moving from the edge of the display and a second content after moving into the body of the display.

10. The method of claim **9**, wherein the first content is a sub-set of the second content.

11. The method of claim **1**, further comprising receiving, on the touch pad surface over the graphical display, a second user input motion dragging across the touch pad surface;

identifying the second dragging input motion as originating on the touchpad and terminating off the edge of the touch pad by identifying a sensed last location for the input motion at a peripheral edge of the touch pad surface; and

displaying on the graphical display a second sliding graphical element that is animated to move to the edge of the display from the body of the display, over a second nonmoving element on the display, in response to identifying the second dragging input motion.

12. The method of claim **11**, wherein the first and second sliding graphical elements are the same.

13. The method of claim **12**, wherein the first and second nonmoving elements are the same.

14. A computer-implemented user input process for a computing device, the process comprising:

receiving, on a touch pad surface over a graphical display, a user input motion dragging across the touch pad surface;

identifying the dragging input motion as originating on the touchpad and terminating off an edge of the touch pad by identifying a sensed last location for the input motion at a peripheral edge of the touch pad surface; and

displaying on the graphical display a sliding graphical element that is animated to move to the edge of the display from a body of the display, over a nonmoving element on the display, in response to identifying the dragging input motion.

15. The method of claim **14**, wherein receiving a user input motion comprises receiving an input motion that exits the touch pad surface at a right or left side of the touch pad surface.

16. The method of claim **14**, wherein receiving a user input motion comprises receiving an input motion that exits the touch pad surface at a top or bottom side of the touch pad surface.

17. The method of claim **14**, wherein the sliding graphical element moves in a direction corresponding to the user input motion.

18. The method of claim **14**, wherein the dragging input motion originates on the touch pad surface in a location on the touch pad surface corresponding to the location of the sliding graphical element on the graphical display.

19. The method of claim **14**, wherein the sliding graphical element is animated to further move from the edge of the display off the display.

20. The method of claim **14**, further comprising mapping the user input motion received on the touch pad to corresponding points on the graphical display.

21. A system comprising:

a graphical display on a computing device;

a nonmoving element on the graphical display; and

a sliding graphical element capable of moving from an edge of the graphical display to a body of the graphical display over the nonmoving element in response to an identified dragging input motion that originated off an edge of a touch pad.

22. The system of claim **21**, wherein movement of the sliding graphical element is based on mapping user input received on a touchpad.

23. The system of claim **21**, wherein movement of the sliding graphical element corresponds to a user input motion.

24. The system of claim **23**, wherein movement of the sliding graphical element further corresponds to one of a length of the user input motion and a velocity of the user input motion.

25. The system of claim **21**, further comprising a first content displayed by the sliding graphical element when the sliding graphical element is in a first position located near the edge of the display.

26. The system of claim **25**, further comprising a second content displayed by the sliding graphical element when the sliding graphical element is in a second position located in the body of the display.

27. The system of claim **26**, wherein a size associated with the second content is greater than a size associated with the first content.

* * * * *