

EDGE GESTURE

BACKGROUND

[0001] Conventional techniques for selecting a user interface that is not currently exposed on a display are often confusing, take up valuable display space, cannot be
5 universally applied across different devices, or provide a poor user experience.

[0002] Some conventional techniques, for example, enable selection of a user interface through onscreen controls in a task bar, within a floating window, or on a window frame. These onscreen controls, however, take up valuable display real estate and can annoy users by requiring users to find and select the correct control.

10 [0003] Some other conventional techniques enable selection of a user interface through hardware, such as hot keys and buttons. At best these techniques require users to remember what key, key combination, or hardware button to select. Even in this best case users often accidentally select keys or buttons. Further, in many cases hardware-selection techniques cannot be universally applied, as hardware on computing devices can vary by
15 device model, generation, vendor, or manufacturer. In such cases either the techniques will not work or work differently across different computing devices. This exacerbates the problem of users needing to remember the correct hardware, as many users have multiple devices, and so may need to remember different hardware selections for different devices. Further still, for many computing devices hardware selection forces users to engage a
20 computing device outside the user's normal flow of interaction, such as when a touch-screen device requires a user to change his or her mental and physical orientation from display-based interactions to hardware-based interactions.

SUMMARY

[0004] This document describes techniques and apparatuses enabling an edge gesture.
25 In some embodiments, these techniques and apparatuses enable selection of a user interface not currently exposed on a display through an edge gesture that is easy-to-use and remember.

[0005] This summary is provided to introduce simplified concepts for enabling an edge gesture that are further described below in the Detailed Description. This summary is not
30 intended to identify essential features of the claimed subject matter, nor is it intended for use in determining the scope of the claimed subject matter. Techniques and/or apparatuses enabling an edge gesture are also referred to herein separately or in conjunction as the "techniques" as permitted by the context.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] Embodiments enabling an edge gesture are described with reference to the following drawings. The same numbers are used throughout the drawings to reference like features and components:

5 [0007] Fig. 1 illustrates an example system in which techniques enabling an edge gesture can be implemented.

[0008] Fig. 2 illustrates an example method for enabling edge gestures based on the edge gesture being approximately perpendicular to an edge in which the gesture begins.

10 [0009] Fig. 3 illustrates an example tablet computing device having a touch-sensitive display presenting an immersive interface.

[0010] Fig. 4 illustrates the example immersive interface of Fig. 3 along with example edges.

15 [0011] Fig. 5 illustrates the example immersive interface of Figs. 3 and 4 along with angular variance lines from a perpendicular line and a line from a start point to a later point of a gesture.

[0012] Fig. 6 illustrates the edges of the immersive interface shown in Fig. 4 along with two regions in the right edge.

20 [0013] Fig. 7 illustrates an application-selection interface presented by a system-interface module in response to an edge gesture and over the immersive interface and webpage of Fig. 3.

[0014] Fig. 8 illustrates an example method for enabling edge gestures including determining an interface to present based on some factor of the gesture.

25 [0015] Fig. 9 illustrates an example method enabling expansion of, or ceasing presentation of, a user interface presented in response to an edge gesture or presentation of another user interface.

[0016] Fig 10 illustrates a laptop computer having a touch-sensitive display having a windows-based email interface and two immersive interfaces.

[0017] Fig. 11 illustrates the interfaces of Fig. 10 along with two gestures having a start point, later points, and one or more successive points.

30 [0018] Fig. 12 illustrates the windows-based email interface of Figs. 10 and 11 along with an email handling interface presented in response to an edge gesture.

[0019] Fig. 13 illustrates the interfaces of Fig. 12 along with an additional-email-options interface presented in response to a gesture determined to have a successive point a preset distance from the edge.

[0020] Fig. 14 illustrates an example device in which techniques enabling edge gestures can be implemented.

DETAILED DESCRIPTION

Overview

5 [0021] This document describes techniques and apparatuses enabling an edge gesture. These techniques enable a user to quickly and easily select an interface not currently exposed on the user's device, as well as other operations.

[0022] Consider a case where a user is watching a movie on a tablet computing device. Assume that the movie is playing on an immersive interface occupying all of the display
10 and that the user would like to check her social-networking webpage without stopping the movie. The described techniques and apparatuses enable her to select other interfaces through a simple swipe gesture started at an edge of her display. She may swipe from one edge of her display and drag out a user interface enabling her to select her social networking website. Or instead, assume that she would like to interact with the media
15 application playing the movie in a manner not permitted by the immersive interface, such as to display a menu enabling subtitles or a director's commentary. She may swipe from another edge of her tablet's display and drag out a control menu for the immersive interface and select items and/or commands from this menu quickly and easily.

[0023] In both of these cases valuable real estate used to play the movie was not taken
20 up with on-screen controls, nor was the user required to remember and find a hardware button. Further still, no gesture, other than one starting from an edge, is used by the techniques in this example, thereby permitting the immersive interface to use nearly all commonly-available gestures. Additionally, by considering edge gestures or portions thereof, the techniques do not affect performance of a gesture or touch input system as the
25 edge gestures can be processed before the entire gesture is complete avoiding latency associated with processing entire gestures started elsewhere.

[0024] These are but two examples of the many ways in which the techniques enable and use edge gestures, others of which are described below.

Example System

30 [0025] Fig. 1 illustrates an example system 100 in which techniques enabling an edge gesture can be embodied. System 100 includes a computing device 102, which is illustrated with six examples: a laptop computer 104, a tablet computer 106, a smart phone 108, a set-top box 110, a desktop computer 112, and a gaming device 114, though other computing devices and systems, such as servers and netbooks, may also be used.

[0026] Computing device 102 includes computer processor(s) 116 and computer-readable storage media 118 (media 118). Media 118 includes an operating system 120, windows-based mode module 122, immersive mode module 124, system-interface module 126, gesture handler 128, and one or more applications 130, each having one or more application user interfaces 132.

[0027] Computing device 102 also includes or has access to one or more displays 134 and input mechanisms 136. Four example displays are illustrated in Fig. 1. Input mechanisms 136 may include gesture-sensitive sensors and devices, such as touch-based sensors and movement-tracking sensors (e.g., camera-based), as well as mice (free-standing or integral with a keyboard), track pads, and microphones with accompanying voice recognition software, to name a few. Input mechanisms 136 may be separate or integral with displays 134; integral examples include gesture-sensitive displays with integrated touch-sensitive or motion-sensitive sensors.

[0028] Windows-based mode module 122 presents application user interfaces 132 through windows having frames. These frames may provide controls through which to interact with an application and/or controls enabling a user to move and resize the window.

[0029] Immersive mode module 124 provides an environment by which a user may view and interact with one or more of applications 130 through application user interfaces 132.

In some embodiments, this environment presents content of, and enables interaction with, applications with little or no window frame and/or without a need for a user to manage a window frame's layout or primacy relative to other windows (e.g., which window is active or up front) or manually size or position application user interfaces 132.

[0030] This environment can be, but is not required to be, hosted and/or surfaced without use of a windows-based desktop environment. Thus, in some cases immersive mode module 124 presents an immersive environment that is not a window (even one without a substantial frame) and precludes usage of desktop-like displays (e.g., a taskbar). Further still, in some embodiments this immersive environment is similar to an operating system in that it is not closeable or capable of being un-installed. While not required, in some cases this immersive environment enables use of all or nearly all of the pixels of a display by applications. Examples of immersive environments are provided below as part of describing the techniques, though they are not exhaustive or intended to limit the techniques described herein.

[0031] System-interface module 126 provides one or more interfaces through which interaction with operating system 120 is enabled, such as an application-launching interface, a start menu, or a system tools or options menu, to name just a few.

[0032] Operating system 120, modules 122, 124, and 126, as well as gesture handler 128 can be separate from each other or combined or integrated in any suitable form.

Example Methods

[0033] Fig. 2 depicts a method 200 for enabling edge gestures based on the edge gesture being approximately perpendicular to an edge in which the gesture begins. In portions of the following discussion reference may be made to system 100 of Fig. 1, reference to which is made for example only.

[0034] Block 202 receives a gesture. This gesture may be received at various parts of a display, such as over a windows-based interface, an immersive interface, or no interface. Further, this gesture may be made and received in various manners, such as a pointer tracking a movement received through a touch pad, mouse, or roller ball or a physical movement made with arm(s), finger(s), or a stylus received through a motion-sensitive or touch-sensitive mechanism. In some cases, the gesture is received off of or proximate to a physical edge of the display (e.g., as a finger or stylus encounters the edge of the display) by a touch digitizer, a capacitive touch screen, or a capacitive sensor, just to name a few.

[0035] By way of example consider Fig. 3, which illustrates a tablet computing device 106. Tablet 106 includes a touch-sensitive display 302 shown displaying an immersive interface 304 that includes a webpage 306. As part of an ongoing example, at block 202 gesture handler 128 receives gesture 308 as shown in Fig. 3.

[0036] Block 204 determines whether a start point of the gesture is at an edge. As noted above, the edge in question can be an edge of a user interface, whether immersive or windows-based, and/or of a display. In some cases, of course, an edge of a user interface is also an edge of a display. The size of the edge can vary based on various factors about the display or interface. A small display or interface may have a smaller size in absolute or pixel terms than a large display or interface. A highly sensitive input mechanism permits a smaller edge as well. In some instances, an edge may extend beyond an edge of the display or a screen when an input mechanism is able to receive a gesture portion beyond the display or screen. Example edges are rectangular and vary between one and twenty pixels in one dimension and an interface limit of the interface or display in the other dimension, though other sizes and shapes, including convex and concave edges may instead be used.

[0037] Continuing the ongoing example, consider Fig. 4, which illustrates immersive interface 304 and gesture 308 of Fig. 3 as well as left edge 402, top edge 404, right edge 406, and bottom edge 408. For visual clarity webpage 306 is not shown. In this example the dimensions of the interface and display are of a moderate size, between that of smart
5 phones and that of many laptop and desktop displays. Edges 402, 404, 406, and 408 have a small dimension of twenty pixels or about 10-15mm in absolute terms, an area of each shown bounded by dashed lines at twenty pixels from the display limit at edge limit 410, 412, 414, and 416, respectively.

[0038] Gesture handler 128 determines that gesture 308 has a start point 418 and that
10 this start point 418 is within left edge 402. Gesture handler 128 determines the start point in this case by receiving data indicating [X,Y] coordinates in pixels at which gesture 308 begins and comparing the first of these coordinates to those pixels contained within each edge 402-408. Gesture handler 128 often can determine the start point and whether it is in an edge faster than a sample rate, thereby causing little or no performance downgrade
15 from techniques that simply pass gestures directly to an exposed interface over which a gesture is made.

[0039] Returning to method 200 generally, if block 204 determines that the start point of the gesture is not at an edge, method 200 proceeds along a “No” path to block 206. Block 206 passes the gestures to an exposed user interface, such as an underlying interface over
20 which the gesture was received. Altering the ongoing example, assume that gesture 308 was determined not to have a start point within an edge. In such a case gesture handler 128 passes buffered data for gesture 308 to immersive user interface 304. After passing the gesture, method 200 ends.

[0040] If block 204 determines that the start point of the gesture is in an edge, method
25 200 proceeds along a “Yes” path to block 208. Optionally, block 204 may determine a length of a portion of the gesture before method proceeds to block 208. In some cases, determining the length of the portion of the gesture allows the determination of the start point to be made prior to a completion of the gesture. Block 208 responds to the positive determination of block 204 by determining whether a line from the start point to a later
30 point of the gesture is approximately perpendicular from the edge.

[0041] Block 208, in some embodiments, determines the later point used. Gesture handler 128, for example, can determine the later point of the gesture based on the later point being received a preset distance from the edge or the start point, such as past edge limit 410 for edge 402 or twenty pixels from start point 418, all of Fig. 4. In some other

embodiments, gesture handler 128 determines the later point based on it being received a preset time after receipt of the start point, such an amount of time slightly greater than used generally by computing device 102 to determine that a gesture is a tap-and-hold or hover gesture.

5 [0042] For the ongoing embodiment, gesture handler 128 uses a later-received point of gesture 308 received outside of edge 402 so long as that later-received point is received within a preset time. If no point is received outside of the edge within that preset time, gesture handler 128 proceeds to block 206 and passes gesture 308 to immersive interface 304.

10 [0043] Using the start point, block 208 determines whether a line from the start point to the later point of the gesture is approximately perpendicular to the edge. Various angles of variance can be used in this determination by block 208, such as five, ten, twenty, or thirty degrees.

[0044] By way of example, consider an angle of variance of thirty degrees from
15 perpendicular. Fig. 5 illustrates this example variance, showing immersive interface 304, gesture 308, left edge 402, left edge limit 410, and start point 418 of Figs. 3 and 4 along with a thirty-degree variance lines 502 from perpendicular line 504. Thus, gesture handler 128 determines that line 506 from start point 418 to later point 508 (which is at about twenty degrees from perpendicular) is approximately perpendicular based on being within
20 the example thirty-degree variance line 502.

[0045] Generally, if block 208 determines that the line is not approximately perpendicular to the edge, method 200 proceeds along a “No” path to block 206 (e.g., a path of a finger is curved). As noted in part above, block 208 may also determine that a later point or other aspect of a gesture disqualifies the gesture. Examples include when a
25 later point is within the edge, such as due to a hover, tap, press-and-hold, or up-and-down gesture (e.g., to scroll content in the user interface), when the gesture is set to be a single-input gesture and a second input is received (e.g., a first finger starts at an edge but a second finger then lands anywhere), or if a tap event occurs during or prior to the gesture (e.g., a finger is already making contact elsewhere or contact is received elsewhere during
30 the gesture).

[0046] If block 208 determines that the line is approximately perpendicular based a later point outside the edge, method 200 proceeds along a “Yes” path to block 210.

[0047] Block 210 responds to the positive determination of block 208 by passing the gesture to an entity other than the exposed user interface. This entity is not a user

interface over which the gesture was received, assuming it was received over a user interface at all. Block 210 may determine to which entity to pass the gesture as well, such as based on an edge or region of an edge in which the start point of the gesture is received. Consider Fig. 6, for example, which illustrates immersive interface 304 and edges 402, 404, 406, and 408 of Fig. 4 but adds top region 602 and bottom region 604 to right edge 406. A start point in top region 602 can result in a different entity (or even a same entity but a different user interface provided in response) than a start point received to bottom region 604. Likewise, a start point in top edge 404 can result in a different entity or interface than left edge 402 or bottom edge 408.

[0048] In some cases, this entity is an application associated with the user interface. In such a case, passing the gesture to the entity can be effective to cause the application to present a second user interface enabling interaction with the application. In the movie example above, the entity can be the media player playing the movie but not the immersive interface displaying the movie. The media player can then present a second user interface enabling selection of subtitles or a director's commentary rather than selections enabled by the interface displaying the movie, such as "pause," "play," and "stop." This capability is permitted in Fig. 1, where one of applications 130 can include or be capable of presenting more than one application user interface 132. Thus, block 210 can pass the gesture to system-interface module 126, the one of applications 130 currently presenting the user interface, or another of applications 130, to name just three possibilities.

[0049] Concluding the ongoing embodiment, at block 210 gesture handler 128 passes gesture 308 to system-interface module 126. System-interface module 126 receives the buffered portion of gesture 308 and continues to receive the rest of gesture 308 as it is made by the user. Fig. 7 illustrates a possible response upon receiving gesture 308, showing an application-selection interface 702 presented by system-interface module 126 and over immersive interface 304 and webpage 306 from Fig. 3. Application-selection interface 702 enables selection of various other applications and their respective interfaces at selectable application tiles 704, 706, 708, and 710.

[0050] The example application-selection interface 702 is an immersive user interface presented using immersive mode module 124, though this is not required. Presented interfaces, or a list thereof, may instead be windows-based and presented using windows-based module 122. Both of these modules are illustrated in Fig. 1.

[0051] Block 210 may also or instead determine to pass the gesture to different entities and/or interfaces based on other factors about the gesture received. Example factors are described in greater detail in method 800 below.

[0052] Note that method 200 and other methods described hereafter can be performed in real-time, such as while a gesture is being made and received. This permits, among other things, a user interface presented in response to a gesture to be presented prior to completion of the gesture. Further, the user interface can be presented progressively as the gesture is received. This permits a user experience of dragging out the user interface from the edge as the gesture is performed with the user interface appearing to “stick” to the gesture (e.g., to a mouse point or person’s finger making the gesture).

[0053] Fig. 8 depicts a method 800 for enabling edge gestures including determining an interface to present based on some factor of the gesture. In portions of the following discussion reference may be made to system 100 of Fig. 1, reference to which is made for example only. Method 800 may act wholly or partly separate from, or in conjunction with, other methods described herein.

[0054] Block 802 determines that a gesture made over a user interface has a start point at an edge of the user interface and a later point not within the edge. Block 802 may operate similarly to or use aspects of method 200, such as determining a later point on which to base block 802’s determination. Block 802 may act differently as well.

[0055] In one case, for example, block 802 determines that a gesture is a single-finger swipe gesture starting at an edge of an exposed immersive user interface and having a later point not at the edge but not based on an angle of the gesture. Based on this determination, block 802 proceeds to block 804 rather than pass the gesture to the exposed immersive user interface.

[0056] Block 804 determines which interface to present based on one or more factors of the gesture. Block 804 may do so based on a final or intermediate length of the gesture, whether the gesture is single or multi-point (e.g., single-finger or multi-finger), or a speed of the gesture. In some cases, two or more factors of a gesture determine which interface to present such as a drag-and-hold gesture having a drag length and hold time or a drag-and-drop gesture having a drag length and drop position. Thus, block 804 may determine to present a start menu in response to a multi-finger gesture, an application-selection interface in response to a relatively short single-finger gesture, or a system-control interface permitting selection to shut down computing device 102 in response to relatively

long single-finger gesture, for example. To do so, gesture handler 128 may determine the length of the gesture, speed, or a number of inputs (e.g., fingers).

[0057] In response, block 806 presents the determined user interface. The determined user interface can be any of those mentioned herein as well as a whole new visual such as a new page of an e-book, an additional visual (e.g., a toolbar or navigation bar), or a modified view of a current user interface (presenting text of a current user interface in a different font, color or highlighting). In some cases, visual or non-visual effects may be presented such as actions related to a video game or sound effects associated with the current or presented user interface.

[0058] Assume, by way of example, that gesture handler 128 determines, based on a factor of the gesture, to present a user interface enabling interaction with operating system 120. In response system-interface module 126 presents this user interface. Presentation of the user interface can be similar to manners described in other methods, such as with a progressive display of application-selection user interface 702 of Fig. 7.

[0059] Following method 200 and/or method 800 in whole or in part, the techniques may proceed to perform method 900 of Fig. 9. Method 900 enables expansion of a user interface, presentation of another interface, or ceasing presentation of the user interface presented in response to an edge gesture.

[0060] Block 902 receives a successive point of the gesture and after presentation of at least some portion of the second user interface. As noted in part above, methods 200 and/or 800 are able to present or cause to be presented a second user interface, such as a second user interface for the same application associated with a current user interface, a different application, or a system user interface.

[0061] By way of example, consider Fig. 10, which illustrates a laptop computer 104 having a touch-sensitive display 1002 displaying a windows-based email interface 1004 and two immersive interfaces 1006 and 1008. Windows-based email interface 1004 is associated with an application that manages email, which can be remote or local to laptop computer 104. Fig. 10 also illustrates two gestures, 1010 and 1012. Gesture 1010 proceeds in a straight line while gesture 1012 reverses back (shown with two arrows to show two directions).

[0062] Fig. 11 illustrates gesture 1010 having a start point 1102, a later point 1104, and a successive point 1106, and gesture 1012 having a same start point 1102, a later point 1108, and a first successive point 1110, and a second successive point 1112. Fig. 11 also shows a bottom edge 1114, a later-point area 1116, and an interface-addition area 1118.

[0063] Block 904 determines, based on the successive point, whether the gesture includes a reversal, an extension, or neither. Block 904 may determine a reversal in the direction of the gesture by determining that a successive point is at the edge or is closer to the edge than a prior point of the gesture. Block 904 may determine that the gesture
5 extends based on the successive point being a preset distance from the edge or the later point. If neither of these is determined to be true, method 900 may repeat blocks 902 and 904 to receive and analyze additional successive points until the gesture ends. If block 904 determines that there is a reversal, method 900 proceeds along “Reversal” path to block 906. If block 904 determines that the gesture is extended, method 900 proceeds
10 along an “Extension” path to block 908.

[0064] In the context of the present example, assume that gesture handler 128 receives first successive point 1110 of gesture 1012. Gesture handler 128 then determines that first successive point 1110 is not at edge 1114, is not closer than a prior point of the gesture to edge 1114 (e.g., is not closer than later point 1108), and is not a preset distance from the
15 edge or later point by not being within interface-addition region 1118. In such a case method 900 returns to block 902.

[0065] On a second iteration of block 902, assume that gesture handler 128 receives second successive point 1112. In such a case, gesture handler 128 determines that second successive point 1112 is closer to edge 1114 than first successive point 1110 and thus
20 gesture 1012 includes a reversal. Gesture handler 128 then proceeds to block 906 to cease to present the second user interface previously presented in response to the gesture. By way of example, consider Fig. 12, which illustrates an email handling interface 1202. In this example case of block 906, gesture handler 128 causes the email application to cease to present interface 1202 in response to a reversal of gesture 1012 (not shown removed).

[0066] Block 908, however, presents or causes presentation of a third user interface or expansion of the second user interface. In some cases, presenting the third user interface causes the second user interface to cease to be presented, either through cancelling presentation or hiding the second user interface (e.g., presenting the third user interface over the second user interface). Continuing the ongoing example, consider Fig. 13, which
25 illustrates additional-email-options interface 1302 in response to gesture 1010 determined to have successive point 1106 a preset distance from edge 1104, in this case being within interface-addition region 1118 of Fig. 11. This region and preset distance can be set based on a size of the user interface previously presented in response to the gesture. Thus, a user
30

wishing to add additional controls may simply extend the gesture past the user interface presented in response to an earlier portion of the gesture.

[0067] Method 900 can be repeated to add additional user interfaces or expand a presented user interface. Returning to the example interface 702 of Fig. 7, for example, gesture handler 128 can continue to add interfaces or controls to interface 702 as gesture 308 extends past interface 702, such as by presenting an additional set of selectable application tiles. If gesture 308 extends past the additional tiles, gesture handler 128 may cause system-interface module 124 to present another interface adjacent the tiles to enable the user to select controls, such as to suspend, hibernate, switch modes (immersive to windows-based and the reverse), or shut down computing device 102.

[0068] While the above example user interfaces presented in response to an edge gesture are opaque, they may also be partially transparent. This can be useful by not obscuring content. In the movie example described above, a user interface presented can be partially transparent thereby permitting the movie to be only partially obscured during use of the user interface. Similarly, in the example of Figs. 12 and 13, interfaces 1202 and 1302 may be partially transparent, thereby enabling a user to see the text of the email while also selecting a control in one of the interfaces.

[0069] The preceding discussion describes methods in which the techniques may enable and use edge gestures. These methods are shown as sets of blocks that specify operations performed but are not necessarily limited to the order shown for performing the operations by the respective blocks.

[0070] Aspects of these methods may be implemented in hardware (e.g., fixed logic circuitry), firmware, a System-on-Chip (SoC), software, manual processing, or any combination thereof. A software implementation represents program code that performs specified tasks when executed by a computer processor, such as software, applications, routines, programs, objects, components, data structures, procedures, modules, functions, and the like. The program code can be stored in one or more computer-readable memory devices, both local and/or remote to a computer processor. The methods may also be practiced in a distributed computing environment by multiple computing devices.

Example Device

[0071] Fig. 14 illustrates various components of example device 1400 that can be implemented as any type of client, server, and/or computing device as described with reference to the previous Figs. 1-13 to implement techniques enabling edge gestures. In embodiments, device 1400 can be implemented as one or a combination of a wired and/or

wireless device, as a form of television client device (e.g., television set-top box, digital video recorder (DVR), etc.), consumer device, computer device, server device, portable computer device, user device, communication device, video processing and/or rendering device, appliance device, gaming device, electronic device, and/or as another type of device. Device 1400 may also be associated with a user (e.g., a person) and/or an entity that operates the device such that a device describes logical devices that include users, software, firmware, and/or a combination of devices.

[0072] Device 1400 includes communication devices 1402 that enable wired and/or wireless communication of device data 1404 (e.g., received data, data that is being received, data scheduled for broadcast, data packets of the data, etc.). The device data 1404 or other device content can include configuration settings of the device, media content stored on the device, and/or information associated with a user of the device. Media content stored on device 1400 can include any type of audio, video, and/or image data. Device 1400 includes one or more data inputs 1406 via which any type of data, media content, and/or inputs can be received, such as user-selectable inputs, messages, music, television media content, recorded video content, and any other type of audio, video, and/or image data received from any content and/or data source.

[0073] Device 1400 also includes communication interfaces 1408, which can be implemented as any one or more of a serial and/or parallel interface, a wireless interface, any type of network interface, a modem, and as any other type of communication interface. The communication interfaces 1408 provide a connection and/or communication links between device 1400 and a communication network by which other electronic, computing, and communication devices communicate data with device 1400.

[0074] Device 1400 includes one or more processors 1410 (e.g., any of microprocessors, controllers, and the like), which process various computer-executable instructions to control the operation of device 1400 and to enable techniques enabling and/or using edge gestures. Alternatively or in addition, device 1400 can be implemented with any one or combination of hardware, firmware, or fixed logic circuitry that is implemented in connection with processing and control circuits which are generally identified at 1412.

Although not shown, device 1400 can include a system bus or data transfer system that couples the various components within the device. A system bus can include any one or combination of different bus structures, such as a memory bus or memory controller, a peripheral bus, a universal serial bus, and/or a processor or local bus that utilizes any of a variety of bus architectures.

[0075] Device 1400 also includes computer-readable storage media 1414, such as one or more memory devices that enable persistent and/or non-transitory data storage (i.e., in contrast to mere signal transmission), examples of which include random access memory (RAM), non-volatile memory (e.g., any one or more of a read-only memory (ROM), flash memory, EPROM, EEPROM, etc.), and a disk storage device. A disk storage device may be implemented as any type of magnetic or optical storage device, such as a hard disk drive, a recordable and/or rewriteable compact disc (CD), any type of a digital versatile disc (DVD), and the like. Device 1400 can also include a mass storage media device 1416.

[0076] Computer-readable storage media 1414 provides data storage mechanisms to store the device data 1404, as well as various device applications 1418 and any other types of information and/or data related to operational aspects of device 1400. For example, an operating system 1420 can be maintained as a computer application with the computer-readable storage media 1414 and executed on processors 1410. The device applications 1418 may include a device manager, such as any form of a control application, software application, signal-processing and control module, code that is native to a particular device, a hardware abstraction layer for a particular device, and so on.

[0077] The device applications 1418 also include any system components or modules to implement techniques using or enabling edge gestures. In this example, the device applications 1418 can include system-interface module 122, gesture handler 128, and application(s) 130.

Conclusion

[0078] Although embodiments of techniques and apparatuses enabling an edge gesture have been described in language specific to features and/or methods, it is to be understood that the subject of the appended claims is not necessarily limited to the specific features or methods described. Rather, the specific features and methods are disclosed as example implementations enabling and/or using an edge gesture.

CLAIMS

1. A computer-implemented method comprising:
receiving a gesture made through a gesture-sensitive display presenting one or more user interfaces;
5 determining that the gesture has a start point at an edge of the gesture-sensitive display and a later point not at the edge; and
responsive to the determining, passing the gesture to an entity or other user interface other than the one or more user interfaces presented on the gesture-sensitive display.
- 10 2. A computer-implemented method as described in claim 1, wherein the other user interface is a start menu, application-selection interface, or an option interface for the computing device.
3. A computer-implemented method as described in claim 1, further comprising determining a region of multiple regions of the edge, and wherein passing the gesture to
15 the entity indicates the other user interface based on the region.
4. A computer-implemented method as described in claim 1, further comprising determining that the gesture is a first length or a second length, and wherein passing the gesture passes the gesture to the entity if the gesture is determined to be the first length or to the other user interface if the gesture is determined to be the second length.
- 20 5. A computer-implemented method as described in claim 1, further comprising determining whether the gesture is a single-finger gesture or a multi-finger gesture and wherein passing the gesture passes the gesture to the entity or the other user interface is responsive to determining whether the gesture is a single-finger gesture or a multi-finger gesture.
- 25 6. A computer-implemented method as described in claim 1, wherein passing the gesture to the entity causes the entity to present the other user interface enabling interaction with a system of a computing device associated with the gesture-sensitive display.
7. A computer-implemented method as described in claim 6, wherein the second user
30 interface is at least partially transparent.
8. A computer-implemented method as described in claim 1, wherein passing the gesture causes the other user interface to be displayed and further comprising:
receiving a second gesture made through the gesture-sensitive display;

determining that the second gesture has a start point in the other user interface and a later point at the edge; and

responsive to determining that the second gesture has a start point in the other user interface and a later point at the edge, causing the other user interface to cease to be

5 displayed.

9. A computer-implemented method comprising:

receiving a gesture made through a gesture-sensitive display presenting one or more user interfaces;

10 determining whether a start point of the gesture is received at an edge of the gesture-sensitive display;

responsive to determining that the start point is not at the edge of the gesture-sensitive display, passing the gesture to one of the one or more user interfaces over which the start point of another point of the gesture is received; or

15 responsive to determining that the start point is at the edge of the gesture-sensitive display, determining whether a line from the start point to a later point of the gesture is within about thirty degrees of perpendicular from the edge, and

responsive to determining that the line is not within about thirty degrees of perpendicular from the edge, passing the gesture to the one of the one or more user interfaces; or

20 responsive to determining that the line is within about thirty degrees, presenting a system interface enabling interaction with a system entity.

10. A computer-implemented method as described in claim 9, wherein one of the one or more user interfaces is an immersive interface and another of the one or more user interfaces is a windows-based interface.

25