

Cross-slide Gesture to Select and Rearrange

BACKGROUND

[0001] One of the challenges that continues to face designers of devices having user-engageable displays, such as touch displays, pertains to providing enhanced functionality for users, through gestures that can be employed with the devices. This is so, not only with devices having larger or multiple screens, but also in the context of devices having a smaller footprint, such as tablet PCs, hand-held devices, smaller multi-screen devices and the like.

[0002] One challenge with gesture-based input is that of providing secondary actions. For example, in touch interfaces today, it is common to tap on an item to launch the item. This makes it difficult to provide secondary functionality such as an ability to select items. Further, certain challenges exist with so-called pannable surfaces, i.e. surfaces that can be panned and have their content moved. For example, a pannable surface typically reacts to a finger drag and moves the content in the direction of the user's finger. If the surface contains objects that a user might want to re-arrange, it is difficult to differentiate when the user wants to pan the surface or re-arrange the content.

SUMMARY

[0003] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

[0004] Cross slide gestures for touch displays are described. In at least some embodiments, cross slide gestures can be used on content that pans or scrolls in one direction, to enable additional actions, such as content selection, drag and drop operations, and the like.

[0005] In one or more embodiments, a cross slide gesture can be performed by dragging an item or object in a direction that is different from a panning or scrolling direction. The different-direction drag can be mapped to additional actions or functionality. In one or more embodiments, one or more thresholds can be utilized, such as a distance threshold, in combination with the different-direction drag, to map to additional actions or functionality.

[0006] In at least some embodiments, so-called speed bumps, or other perceptible indicia such as visual indicia, can be used to provide a user with an understanding or awareness of the thresholds.

BRIEF DESCRIPTION OF THE DRAWINGS

5 **[0007]** The detailed description is described with reference to the accompanying figures. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The use of the same reference numbers in different instances in the description and the figures may indicate similar or identical items.

10 **[0008]** FIG. 1 is an illustration of an environment in an example implementation in accordance with one or more embodiments.

[0009] FIG. 2 is an illustration of a system in an example implementation showing FIG. 1 in greater detail.

15 **[0010]** FIG. 3 illustrates an example computing device in accordance with one or more embodiments.

[0011] FIG. 4 illustrates an example computing device in accordance with one or more embodiments.

[0012] FIG. 5 is a flow diagram that describes steps in a method in accordance with one or more embodiments.

20 **[0013]** FIG. 6 illustrates an example computing device in accordance with one or more embodiments.

[0014] FIG. 7 illustrates an example computing device in accordance with one or more embodiments.

25 **[0015]** FIG. 8 is a flow diagram that describes the steps in a method in accordance with one or more embodiments.

[0016] FIG. 9 illustrates a cross-slide detection example in accordance with one or more embodiments.

[0017] FIG. 10 is a flow diagram that describes steps in a method in accordance with one or more embodiments.

30 **[0018]** FIG. 11 illustrates distance thresholds in accordance with one or more embodiments.

[0019] FIG. 12 is a flow diagram that describes steps in a method in accordance with one or more embodiments.

[0020] FIG. 13 illustrates distance thresholds in accordance with one or more embodiments.

[0021] FIG. 14 is a flow diagram that describes the steps in a method in accordance with one or more embodiments.

5 [0022] FIG. 15 illustrates a cross-slide gesture in accordance with one or more embodiments.

[0023] FIG. 16 is a flow diagram that describes steps in a method in accordance with one or more embodiments.

[0024] FIG. 17 illustrates an example computing device that can be utilized to
10 implement various embodiments described herein.

DETAILED DESCRIPTION

Overview

[0025] Cross slide gestures for touch displays are described. In at least some embodiments, cross slide gestures can be used on content that pans or scrolls in one
15 direction, to enable additional actions, such as content selection, drag and drop operations, and the like.

[0026] In one or more embodiments, a cross slide gesture can be performed by dragging an item or object in a direction that is different, e.g. orthogonal, from a panning or scrolling direction. Dragging can be performed via a touch-related drag, such as
20 through a finger, stylus, pen and the like, through a mouse/trackpad drag and the like. In the examples described in this document, touch-related dragging is used. The different-direction drag can be mapped to additional actions or functionality. In one or more embodiments, one or more thresholds can be utilized, such as a distance threshold, in combination with the different-direction drag, to map to additional actions or functionality.
25 For example, in the context of a horizontally-scrollable list, dragging an object vertically a short distance and releasing it may mark an object as selected, while dragging the object a larger distance vertically may break the object free from an associated list so that it can be dropped somewhere else.

[0027] In at least some embodiments, so-called speed bumps, or other perceptible
30 indicia such as visual indicia, can be used to provide a user with an understanding or awareness of the thresholds.

[0028] Various embodiments described herein enable an item to be dragged without necessarily entering a mode. A mode can be thought of as an action that is

initiated by a user that is not necessarily related to manipulating an item directly. For example, a mode can be entered by clicking on a particular user interface button to then be exposed to functionality that can be performed relative to an item or object. In the described embodiments, modes can be avoided by eliminating, in at least some instances, user interface elements to access drag functionality.

[0029] In the following discussion, an example environment is first described that is operable to employ the gesture techniques described herein. Example illustrations of the gestures and procedures are then described, which may be employed in the example environment, as well as in other environments. Accordingly, the example environment is not limited to performing the example gestures and the gestures are not limited to implementation in the example environment.

Example Environment

[0030] FIG. 1 is an illustration of an environment 100 in an example implementation that is operable to employ cross-slide gestures as described herein. The illustrated environment 100 includes an example of a computing device 102 that may be configured in a variety of ways. For example, the computing device 102 may be configured as a traditional computer (e.g., a desktop personal computer, laptop computer, and so on), a mobile station, an entertainment appliance, a set-top box communicatively coupled to a television, a wireless phone, a netbook, a game console, a handheld device, and so forth as further described in relation to FIG. 2. Thus, the computing device 102 may range from full resource devices with substantial memory and processor resources (e.g., personal computers, game consoles) to a low-resource device with limited memory and/or processing resources (e.g., traditional set-top boxes, hand-held game consoles). The computing device 102 also includes software that causes the computing device 102 to perform one or more operations as described below.

[0031] Computing device 102 includes a gesture module 104 that is operational to provide gesture functionality as described in this document. The gesture module can be implemented in connection with any suitable type of hardware, software, firmware or combination thereof. In at least some embodiments, the gesture module is implemented in software that resides on some type of tangible, computer-readable storage medium examples of which are provided below.

[0032] Gesture module 104 is representative of functionality that recognizes gestures, including cross-slide gestures that can be performed by one or more fingers, and

causes operations to be performed that correspond to the gestures. The gestures may be recognized by module 104 in a variety of different ways. For example, the gesture module 104 may be configured to recognize a touch input, such as a finger of a user's hand 106a as proximal to display device 108 of the computing device 102 using touchscreen
5 functionality. In particular, gesture module 104 can recognize cross slide gestures that can be used on content that pans or scrolls in one direction, to enable additional actions, such as content selection, drag and drop operations, and the like.

[0033] For instance, in the illustrated example, a pan or scroll direction is shown as being in the vertical direction, as indicated by the arrows. In one or more embodiments, a
10 cross slide gesture can be performed by dragging an item or object in a direction that is different, e.g. orthogonal, from the panning or scrolling direction. The different-direction drag can be mapped to additional actions or functionality. With respect to whether a direction is vertical or horizontal, a vertical direction can be considered, in at least some instances, as a direction that is generally parallel to one side of a display device, and a
15 horizontal direction can be considered as a direction that is generally orthogonal to the vertical direction. Hence, while the orientation of a computing device may change, the verticality or horizontality of a particular cross slide gesture can remain standard as defined relative to and along the display device.

[0034] For example, a finger of the user's hand 106a is illustrated as selecting
20 an image 112 displayed by the display device 108. Selection 110 of the image 112 and subsequent movement of the finger of the user's hand 106a in a direction that is different from the pan or scroll direction, e.g., generally orthogonal relative to the pan or scroll direction, may be recognized by the gesture module 104. The gesture module 104 may then identify this recognized movement, by the nature and character of the movement, as
25 indicating a "drag and drop" operation to change a location of the image 112 to a point in the display at which the finger of the user's hand 106a is lifted away from the display device 108. Thus, recognition of the touch input that describes selection of the image, movement of the selection point to another location, and then lifting of the finger of the user's hand 106a may be used to identify a gesture (e.g., drag-and-drop gesture) that is to
30 initiate the drag-and-drop operation.

[0035] Although cross-slide gestures are primarily discussed in this document, it is to be appreciated and understood that a variety of different types of gestures may be recognized by the gesture module 104 including, by way of example and not limitation,

gestures that are recognized from a single type of input (e.g., touch gestures such as the previously described drag-and-drop gesture) as well as gestures involving multiple types of inputs. For example, module 104 can be utilized to recognize single-finger gestures and bezel gestures, multiple-finger/same-hand gestures and bezel gestures, and/or multiple-finger/different-hand gestures and bezel gestures.

[0036] For example, the computing device 102 may be configured to detect and differentiate between a touch input (e.g., provided by one or more fingers of the user's hand 106a) and a stylus input (e.g., provided by a stylus 116). The differentiation may be performed in a variety of ways, such as by detecting an amount of the display device 108 that is contacted by the finger of the user's hand 106 versus an amount of the display device 108 that is contacted by the stylus 116.

[0037] Thus, the gesture module 104 may support a variety of different gesture techniques through recognition and leverage of a division between stylus and touch inputs, as well as different types of touch inputs.

[0038] FIG. 2 illustrates an example system showing the gesture module 104 as being implemented in an environment where multiple devices are interconnected through a central computing device. The central computing device may be local to the multiple devices or may be located remotely from the multiple devices. In one embodiment, the central computing device is a "cloud" server farm, which comprises one or more server computers that are connected to the multiple devices through a network or the Internet or other means.

[0039] In one embodiment, this interconnection architecture enables functionality to be delivered across multiple devices to provide a common and seamless experience to the user of the multiple devices. Each of the multiple devices may have different physical requirements and capabilities, and the central computing device uses a platform to enable the delivery of an experience to the device that is both tailored to the device and yet common to all devices. In one embodiment, a "class" of target device is created and experiences are tailored to the generic class of devices. A class of device may be defined by physical features or usage or other common characteristics of the devices. For example, as previously described the computing device 102 may be configured in a variety of different ways, such as for mobile 202, computer 204, and television 206 uses. Each of these configurations has a generally corresponding screen size and thus the computing device 102 may be configured as one of these device classes in this example system 200.

For instance, the computing device 102 may assume the mobile 202 class of device which includes mobile telephones, music players, game devices, and so on. The computing device 102 may also assume a computer 204 class of device that includes personal computers, laptop computers, netbooks, and so on. The television 206 configuration
5 includes configurations of device that involve display in a casual environment, e.g., televisions, set-top boxes, game consoles, and so on. Thus, the techniques described herein may be supported by these various configurations of the computing device 102 and are not limited to the specific examples described in the following sections.

[0040] Cloud 208 is illustrated as including a platform 210 for web services 212.
10 The platform 210 abstracts underlying functionality of hardware (e.g., servers) and software resources of the cloud 208 and thus may act as a “cloud operating system.” For example, the platform 210 may abstract resources to connect the computing device 102 with other computing devices. The platform 210 may also serve to abstract scaling of resources to provide a corresponding level of scale to encountered demand for the web
15 services 212 that are implemented via the platform 210. A variety of other examples are also contemplated, such as load balancing of servers in a server farm, protection against malicious parties (e.g., spam, viruses, and other malware), and so on.

[0041] Thus, the cloud 208 is included as a part of the strategy that pertains to software and hardware resources that are made available to the computing device 102 via
20 the Internet or other networks. For example, the gesture module 104 may be implemented in part on the computing device 102 as well as via a platform 210 that supports web services 212.

[0042] For example, the gesture techniques supported by the gesture module may be detected using touchscreen functionality in the mobile configuration 202, track pad
25 functionality of the computer 204 configuration, detected by a camera as part of support of a natural user interface (NUI) that does not involve contact with a specific input device, and so on. Further, performance of the operations to detect and recognize the inputs to identify a particular gesture may be distributed throughout the system 200, such as by the computing device 102 and/or the web services 212 supported by the platform 210 of the
30 cloud 208.

[0043] Generally, any of the functions described herein can be implemented using software, firmware, hardware (e.g., fixed logic circuitry), manual processing, or a combination of these implementations. The terms “module,” “functionality,” and “logic”

as used herein generally represent software, firmware, hardware, or a combination thereof. In the case of a software implementation, the module, functionality, or logic represents program code that performs specified tasks when executed on or by a processor (e.g., CPU or CPUs). The program code can be stored in one or more computer readable memory devices. The features of the gesture techniques described below are platform-independent, meaning that the techniques may be implemented on a variety of commercial computing platforms having a variety of processors.

[0044] In the discussion that follows, various sections describe example cross-slide gestures including re-arrange gestures. A section entitled “Method/Gesture for Cross-Slide Relative to Panning Direction” describes a cross-slide gesture that can be executed relative to a panning direction in accordance with one or more embodiments. Next, a section entitled “Method/Gesture for Re-arranging Items in a Pannable List” describes how items can be arranged and rearranged utilizing a cross-slide gesture in accordance with one or more embodiments. Following this, a section entitled “Detecting Cross-Slide Gestures” describes how cross-slide gestures can be detected in accordance with one or more embodiments. Next, a section entitled “Combining Multiple Interactions” describes how multiple interactions can be combined in conjunction with cross-slide gestures in accordance with one or more embodiments. Following this, a section entitled “Direct Manipulation to Facilitate Threshold to Discernability” describes how direct manipulation feedback can be provided to enable a user to become aware of various thresholds in accordance with one or more embodiments. Next, a section entitled “Interaction Feedback” describes embodiments in which feedback can be provided to a user in accordance with one or more embodiments. Last, a section entitled “Example Device” describes aspects of an example device that can be utilized to implement one or more embodiments.

Method/Gesture for Cross-Slide Relative to Panning Direction

[0045] In one or more embodiments, a cross slide gesture can be performed for causing an object-related action to be performed by dragging an item or object in a direction that is different, e.g. orthogonal, from a scrolling or panning direction.

[0046] As an example, consider FIG. 3 which illustrates an environment 300 in accordance with one or more embodiments. Here, computing device 302 includes a display device 308 whose content can be scrolled or panned in the horizontal direction, as indicated by the double-headed arrow 304, and as suggested by scroll bar 305. Display

device 308 has displayed, thereon, multiple different objects or items 310, 312, 314, 316, 318, 320, 322, 324 which are shown in their entirety, and partial objects or items 326, 328. In this example, a user can affect scrolling or panning in the horizontal direction by using a swipe gesture on the display device 308 in the horizontal direction. Alternately, a user can cause an object-related action to be performed by performing a cross slide gesture, relative to one of the objects or items, in a direction that is different from the scrolling or panning direction.

[0047] As an example, consider the bottom-most illustration of computing device 302. There, a user's hand 306a has touched over item 312 and moved it in a direction that is different from the scrolling or panning direction. In this particular example, the different direction is generally orthogonal to the scrolling or panning direction in a downward direction. It is to be appreciated and understood that, in at least some embodiments, the object can be moved downward and upward or, more generally, bi-directionally, to access the same or different object-related actions. Any suitable type of object-related action can be performed. For example, one type of object-related action can include, by way of example and not limitation, object selection. Notice, in this example, that the selected item is directly manipulated and visual feedback is provided to the user by being able to observe the object move responsive to the user's engagement. Notice also that, in this embodiment and the ones described below, the object-related action is performed without showing additional user interface elements, such as a button to enable a command selection. Other object-related actions can be performed such as object delete and other object manipulation actions.

[0048] As another example, consider FIG. 4 which illustrates an environment 400 in accordance with one or more embodiments. Here, computing device 402 includes a display device 408 whose content can be scrolled or panned in the vertical direction, as indicated by the double-headed arrow 404, and as suggested by scroll bar 405. Display device 408 has displayed, thereon, multiple different objects or items 410, 412, 414, 416, 418, 420, 422, 424 which are shown in their entirety. In this example, a user can affect scrolling or panning in the vertical direction by using a swipe gesture on the display device 408 in the vertical direction. Alternately, a user can cause an object-related action to be performed by performing a cross slide gesture, relative to one of the objects or items, in a direction that is different from the scrolling or panning direction.

[0049] As an example, consider the bottom-most illustration of computing device 402. There, a user's hand 406a has touched over item 412 and moved it in a direction that is different from the scrolling or panning direction. In this particular example, the different direction is generally orthogonal to the scrolling or panning direction. Any suitable type of object-related action can be performed, examples of which are provided below. For example, one type of object-related action can include, by way of example and not limitation, object selection. It is to be appreciated and understood that functionality that is accessible through cross slide gestures can be accessed in connection with moving the object or item any suitable threshold distance to invoke the object-related action. In at least some embodiments, there may be no threshold distance to invoke object-related action. In these instances, movement in a different direction other than the pan or scroll direction may be used to invoke the object-related action.

[0050] FIG. 5 is a flow diagram that describes steps in a method in accordance with one or more embodiments. The method can be performed in connection with any suitable hardware, software, firmware, or combination thereof. In at least some embodiments, the method can be performed by a suitably-configured gesture module, such as the one described above.

[0051] Step 500 detects a gestural slide input relative to a display device associated with a computing device. Step 502 ascertains whether the direction of the gestural slide input is different from a panning direction. If the direction is not different from the panning direction, step 504 pans content in the direction of the gestural slide input. If, on the other hand, the direction of the gestural slide input is different from the panning direction, step 506 performs an object-related action. Any suitable type of object-related action can be performed, examples of which are provided below.

Method/Gesture for Re-Arranging Items in a Pannable List

[0052] In one or more embodiments, a cross slide gesture can be performed effective to cause an object-related action, in the form of an object-rearrangement action, to be performed by dragging an item or object in a direction that is different, e.g. orthogonal to or generally not in the direction associated with a scrolling or panning direction.

[0053] As an example, consider FIG. 6 which illustrates an environment 600 in accordance with one or more embodiments. Here, computing device 602 includes a display device 608 whose content can be scrolled or panned in the horizontal direction, as

indicated by the double-headed arrow 604, and as suggested by scroll bar 605. Display device 608 has displayed, thereon, multiple different objects or items 610, 612, 614, 616, 618, 620, 622, 624 which are shown in their entireties, and partial objects or items 626, 628. In this example, a user can affect scrolling or panning in the horizontal direction by using a swipe gesture on the display device 608 in the horizontal direction. Alternately, a user can cause an object-related action, in the form of a rearrangement action, to be performed by performing a cross slide gesture, relative to one of the objects or items, in a direction that is different from the scrolling or panning direction. For example, in the topmost illustration, a user's hand 606a has touched display device 608 over object 612 and dragged the object in a first direction that is generally orthogonal to the scrolling or panning direction, and then in a second direction toward the left bottom corner of display device 608. Here, the first direction is a generally vertical direction. Dragging the object in the first direction indicates to the gesture module that an object is to be rearranged.

[0054] Consider now the bottom-most illustration of computing device 602.

There, the user's hand 606a has dragged object 612 to its illustrated position and dropped it in place. Subsequently, the user's hand touched display device 608 over object 618 and dragged the object in a first direction that is generally orthogonal to the scrolling or panning direction, and then in a second direction toward the middle portion of the display device. Here, the first direction is a generally vertical direction. Once the user's hand is lifted from the touched display device 608, object 618 will be dropped in its illustrated place.

[0055] As another example, consider FIG. 7 which illustrates an environment 700 in accordance with one or more embodiments. Here, computing device 702 includes a display device 708 whose content can be scrolled or panned in the vertical direction, as indicated by the double-headed arrow 704, and as suggested by scroll bar 705. Display device 708 has displayed, thereon, multiple different objects or items 710, 712, 714, 716, 718, 720, 722, 724. In this example, a user can affect scrolling or panning in the vertical direction by using a swipe gesture on the display device 708 in the vertical direction. Alternately, a user can cause an object-related action, in the form of a rearrangement action, to be performed by performing a cross slide gesture, relative to one of the objects or items, in a direction that is different from the scrolling or panning direction. For example, in the topmost illustration, a user's hand 706a has touched display device 708 over object 712 and dragged the object in a direction that is generally orthogonal to the

scrolling or panning direction. Here, the direction is a generally horizontal direction. Dragging an object in this direction indicates to the gesture module that the object is to be rearranged.

[0056] Consider now the bottom-most illustration of computing device 702.

5 There, the user's hand 706a has dragged object 712 to its illustrated position and dropped it in place. Subsequently, the user's hand touched display device 708 over object 710 and dragged the object in a direction that is generally orthogonal to the scrolling or pending direction. Here, the direction is a generally horizontal direction. Once the user's hand is lifted from the touched display device 708, object 710 will be dropped in its illustrated
10 place.

[0057] FIG. 8 is a flow diagram that describes steps in a method in accordance with one or more embodiments. The method can be performed in connection with any suitable hardware, software, firmware, or combination thereof. In at least some embodiments, the method can be performed by a suitably-configured gesture module, such
15 as the one described above.

[0058] Step 800 detects a drag direction associated with a drag operation relative to a display device associated with a computing device. Step 802 ascertains whether the drag direction is different from a panning direction. If the drag direction is not different from the panning direction, step 804 pans content in the dragging direction. If, on the
20 other hand, the drag direction is different from the panning direction, step 806 performs an object-rearrangement action. Examples of how this can be done are provided above. In one or more embodiments, rearrangement can occur in any suitable direction.

Detecting Cross-slide Gestures

[0059] Cross-slide gestures can be detected in any suitable way. As but one
25 example of how cross-slide gestures can be detected, consider the following in connection with Fig. 9. In one or more embodiments, to detect if a user is panning or cross-sliding, region detection logic can be applied as graphically illustrated in Fig. 9.

[0060] In this example, consider that the user has displayed a horizontally pannable list of items. When the user puts her finger down on an object, as within the
30 illustrated circle 900, and starts to drag her finger outside the boundary of the circle, region detection can be employed to ascertain the outcome of the drag. For example, in a situation in which there is a drag into one of region 902, the content will be panned in a corresponding direction. However, a drag into one of region 904 will be recognized as a

cross-slide gesture and, accordingly, the functionality associated with the cross-slide gesture can be implemented.

[0061] In the illustrated example, regions 902 and 904 are generally similarly sized. However, based on the scenario, certain actions can be prioritized by changing the entrance angle or range of angles, e.g. angles *a* and *b*, of the different regions. For example, by making angles *a* larger thereby increasing their range, (and angles *b* smaller, thereby decreasing their range), it is easier to start panning without accidentally performing a cross-slide gesture, and vice versa.

[0062] FIG. 10 is a flow diagram that describes steps in a method in accordance with one or more embodiments. The method can be performed in connection with any suitable hardware, software, firmware, or combination thereof. In at least some embodiments, the method can be performed by a suitably-configured gesture module, such as the one described above.

[0063] Step 1000 defines one or more regions associated with a panning gesture. Any suitable region geometry can be utilized, an example of which is provided above. Step 1002 defines one or more regions associated with a cross slide gesture. Again, any suitable region geometry can be utilized, an example of which is provided above. In the FIG. 9 example, the region geometries are generally triangular in shape and converge at a point associated with a touch input. Other geometries can be utilized without departing from the spirit and scope of the claimed subject matter.

[0064] Step 1004 detects a drag operation. This step can be performed by detecting gestural input in the form of a touch gesture, such as a swipe. Step 1006 ascertains an associated region within which the drag operation occurs. If, at step 1008 the region is associated with a panning gesture, step 1010 pans content in an associated direction. If, on the other hand, the region is not associated with a panning gesture, step 1012 performs an operation associated with a cross slide gesture. Any suitable object-related action can be performed, including, by way of example and not limitation, object selection, object deletion, object rearrangement, and the like.

[0065] Having considered how drag operations can be detected and differentiated in accordance with one or more embodiments, consider now a discussion of how multiple interactions can be combined.

Combining Multiple Interactions

[0066] In some cases, it can be desirable to have a threshold that can be utilized to lock into an object-related action, such as a drag threshold that allows locking into a drag direction. Any suitable type of threshold can be utilized including, by way of example and not limitation, distance thresholds, velocity thresholds, directionality thresholds, any combination of the aforementioned thresholds, as well as other thresholds. For example, a combination of distance and velocity threshold can be used to mitigate what might otherwise constitute an accidental or unintended action. For example, when a particular threshold is reached, the velocity of finger movement might be ascertained. If the velocity is below a particular threshold, then a drag action might be invoked. If it is above a particular threshold, then perhaps an object select action is performed.

[0067] This makes it possible for the user to be a less precise at the beginning of their gesture. For example, returning to the FIG. 9 example, notice that box 906 is defined. While the user's finger is within box 906 or, alternatively, within the boundary of circle 900, the corresponding gesture can be in an "undecided" state. Once the finger crosses outside the boundary of the box (or circle), a decision as to the gesture can be made. In practice, this can be handled in a couple of different ways. First, neither a pan operation nor cross-slide functionality can be implemented until the finger has crossed the boundary of box 906. Alternately, both pan and cross-slide operations can be implemented simultaneously while the finger is within the boundary of box 906. As soon as the finger crosses the boundary of the box, the operation associated with that particular region can be maintained, while the other operation can be canceled.

[0068] Once a cross-slide gesture has been detected, different thresholds can be utilized to implement object-related actions. As an example, consider FIG. 11. There, an object or item 1100 is shown. Various distances are shown and are indicated at 1102 and 1104. The distances show the travel distance of object 1100. In one or more embodiments, the first distance 1102 is a threshold which, when passed, results in an action potentially being committed. In this particular example, passing this distance threshold while performing a drag operation causes object 1100 to be selected. To commit this action, the user would lift her finger, and the dragged object would slide back to its original position and change its state to be selected. The area beneath that corresponding to distance 1102 before the threshold of distance 1104 is reached can be treated as a buffer. Thus, releasing the object within this area will still result in object selection.

[0069] Once the dragged object (dragged along the solid line or, any other suitable direction such as along the dashed line) reaches distance 1104 and crosses its threshold, the next object-related action on the cross-slide gesture can be committed. In this particular example, the object-related action can break the object out of its associated list or position on the display device, and thus enable the user to drag and drop the object in any direction. In one or more embodiments, if the object reaches line 1106, such can trigger yet additional object-related actions. For example, crossing this line with the object might trigger additional visual feedback to make it clear to the user that the drag and drop threshold has been reached.

[0070] It is to be appreciated and understood that any suitable number of distance thresholds can be employed and can be associated with object related actions. For example, a first threshold might be defined by the boundary of the illustrated circle within object 1100, a second threshold by the distance 1102, and a third threshold by the distance 1104. In one or more embodiments, movement outside of the first threshold can lock the associated object in the associated movement direction.

[0071] FIG. 12 is a flow diagram that describes steps in a method in accordance with one or more embodiments. The method can be performed in connection with any suitable hardware, software, firmware, or combination thereof. In at least some embodiments, the method can be performed by a suitably-configured gesture module, such as the one described above.

[0072] Step 1200 defines one or more thresholds. This step can be performed in any suitable way utilizing any suitable type of threshold. For example, in the embodiment described above, distance thresholds were employed. It is to be appreciated and understood, however, that other types of thresholds and/or combinations thereof can be utilized without departing from the spirit and scope of the claimed subject matter.

[0073] In the illustrated and described embodiment, the defined thresholds can be utilized in connection with a cross-slide gesture as described above. Step 1202 detects a cross-slide gesture. Examples of how this can be done are provided above. Step 1204 detects or one or more threshold triggers. For example, once a user has touched-engaged an object they can move the object in a particular direction. This step detects when the object has been moved sufficient to trigger one or more thresholds. In embodiments where thresholds are defined in terms of distances, the step can be performed by detecting when an object has been moved a particular distance.

[0074] Step 1206 detects whether a user action indicates that an object-related action is to be committed. This step can be performed in any suitable way. For example, a user action might include lifting their finger off a particular object. If a user action does not indicate that an object-related action is to be committed, step 1208 does not commit
5 the object-related action. For example, the user might terminate the cross-slide gesture in a particular way such that no action is to be committed. For example, the cross-slide gesture can be reversible, i.e. if the user starts dragging an object down, then she can, at any time while still holding the object, slide it back to its original position. By doing this, no cross-slide actions will be taken. Alternately, one or more thresholds might be crossed,
10 without a user having yet indicated that the object-related action is to be committed. In this case, if the cross-slide gesture is ongoing, the method would continue to monitor for threshold triggers as by returning to step 1204. If, on the other hand, a user action indicates that an object-related action is to be committed, step 1210 commits the object-related action associated with a last-triggered threshold. This step can be performed in any
15 suitable way and can include any suitable object-related action, examples of which are provided above.

[0075] In one or more embodiments, the multiple different directions used for cross-slide functionality can either result in the same object-related actions being performed, or different object-related actions being performed. For example, object
20 selection might occur when an object is dragged downward, while a drag and drop action might be performed when the object is dragged upward.

[0076] Having considered the use of various drag thresholds and associated the object-related actions, consider now an additional example that employs thresholds along with indicia to provide feedback of direct object manipulation.

Direct Manipulation to Facilitate Threshold Discernability

[0077] In at least some embodiments, direct manipulation can provide visual feedback so that a user can visually observe an object move and, in accordance with object movement, can be provided with visual affordances to facilitate threshold discernability.

5 Any suitable type of the visual affordance can be employed including, by way of example and not limitation, tool tips, icons, glyphs, and the like. In the example described just below, so-called speed bumps can be used to provide a user with an understanding or awareness of the various thresholds that might be present. As an example, consider FIG. 13.

10 [0078] There, an object or item 1300 is shown. Various distances are shown and are indicated at 1302, 1304, and 1306. The distances show the travel distance of object 1300 or distances through which the object can travel. In one or more embodiments, the first distance 1302 is a threshold which, when passed, results in an action potentially being committed. In this particular example, passing this distance threshold while performing a
15 drag operation causes object 1300 to be selected. To commit this action, the user would lift her finger, and the dragged object would slide back to its original position and change its state to be selected. The area beneath that corresponding to distance 1302 before the region corresponding to distance 1306 is reached can be treated as a buffer. Thus, releasing the object within this area will still result in object selection.

20 [0079] Distance 1306 corresponds to a speed bump region. Movement of object 1300 within the speed bump region is slower than movement of the finger. This presents a visual cue or indication that a new threshold is about to be reached, thus making it easier for the user to commit a particular action without accidentally moving into and over a next distance threshold. For example, within a speed bump region, a user may drag her finger
25 50 pixels in length, while the corresponding object may move five pixels in distance. Releasing the object within this speed bump region will result in an associated action being committed. In this example, the associated action is an object selection.

[0080] Once the dragged object proceeds through the speed bump region corresponding to distance 1306, and reaches distance 1304 and crosses its threshold, the
30 next object-related action on the cross-slide gesture can be committed. In this particular example, the object-related action can break the object out of its associated list or position on the display device, and thus enable the user to drag and drop the object in any direction. In one or more embodiments, if the object reaches line 1308, such can trigger yet

additional object-related actions. For example, crossing this line with the object might trigger additional visual feedback to make it clear to the user that the drag and drop threshold has been reached. In addition, multiple speed bumps can be utilized in connection with the distance thresholds.

5 **[0081]** It is to be appreciated and understood that any suitable number of distance thresholds and speed bumps can be employed and can be associated with object related actions. Alternately or additionally, other visual indicia can be utilized to indicate thresholds or threshold changes. For example, while dragging an object, one or more lines can be rendered to indicate thresholds and thus, the distance that an object should be
10 dragged to commit different actions. Visuals can also be drawn on the object itself as it gets closer to or crosses a threshold.

[0082] FIG. 14 is a flow diagram that describes steps in a method in accordance with one or more embodiments. The method can be performed in connection with any suitable hardware, software, firmware, or combination thereof. In at least some
15 embodiments, the method can be performed by a suitably-configured gesture module, such as the one described above.

[0083] Step 1400 defines one or more distance thresholds including one or more speed bumps. This step can be performed in any suitable way. In the illustrated and described embodiment, the defined thresholds and speed bumps can be utilized in
20 connection with a cross-slide gesture as described above. Step 1402 detects a cross-slide gesture. Examples of how this can be done are provided above. Step 1404 detects a speed bump crossing. For example, once a user has touched-engaged an object they can move the object in a particular direction. This step detects when the object has been moved sufficient to cross a boundary associated with a speed bump.

25 **[0084]** Step 1406 modifies a user experience within the speed bump region. Any suitable modification of the user experience can be provided. For example, in at least some embodiments, modification of the user experience can entail modifying the user's visual experience. For example, and as noted above, the user's finger may move the faster than the underlying object. Alternately or additionally, other experience modifications can
30 take place including, by way of example and not limitation, providing audible or haptic feedback to indicate presence within a particular speed bump region.

[0085] Step 1408 detects whether a user action indicates that an object-related action is to be committed. This step can be performed in any suitable way. For example,

a user action might include lifting their finger off a particular object. If a user action does not indicate that an object-related action is to be committed, step 1410 does not commit the object-related action. For example, the user might terminate the cross-slide gesture in a particular way such that no action is to be committed. For example, the cross-slide gesture can be reversible, i.e. if the user starts dragging an object down, then she can, at any time while still holding the object, slide it back to its original position. By doing this, no cross-slide actions will be taken. Alternately, one or more thresholds and one or more speed bump regions might be crossed, without a user having yet indicated that the object-related action is to be committed. In this case, if the cross-slide gesture is ongoing, the method would continue to monitor for threshold crossings and additional speed bumps, as appropriate. If, on the other hand, a user action indicates that an object-related action is to be committed, step 1412 commits the object-related action associated with a last-crossed threshold. This step can be performed in any suitable way and can include any suitable object-related action, examples of which are provided above.

[0086] In one or more embodiments, the multiple different directions used for cross-slide functionality can either result in the same object-related actions being performed, or different object-related actions being performed. For example, object selection might occur when an object is dragged downward, while a drag and drop action might be performed when the object is dragged upward.

Interaction Feedback

[0087] In one or more embodiments, visual feedback can be provided to a user to inform the user of a particular object-related action that will be committed, responsive to the detected cross-slide gesture. For example, as a particular object passes different distance thresholds, visual indicia can be provided to inform the user of a particular action that will be committed by releasing the object. Alternately or additionally, visual indicia can further be provided on a particular object-related action that might be next if object dragging continues.

[0088] As an example, consider Fig. 15. There, an object in the form of a picture is shown generally at 1500. In the top-most part of the figure, a user has touched the object to initiate a drag operation. As the user drags the object downward, as shown at 1502, visual indicia 1504 can be presented as by beginning to emerge from underneath the picture. In this particular example, the visual indicia resides in the form of a check box that gradually emerges from underneath the object. Any suitable type of visual indicia

can be utilized without departing from the spirit and scope of the claimed subject matter. For example, the visual indicia might be presented in the form of a line, beneath the picture, and to which the picture is to be dragged to commit a particular action. Once the object has been dragged a particular distance, as shown at 1506, the visual indicia – here, the check box, can be fully exposed thus informing the user that she can release the object to commit an object-related action. In this particular example, the object-related action comprises an object selection. Thus, the fully exposed check box can indicate that the action is completed.

[0089] FIG. 16 is a flow diagram that describes steps in a method in accordance with one or more embodiments. The method can be performed in connection with any suitable hardware, software, firmware, or combination thereof. In at least some embodiments, the method can be performed by a suitably-configured gesture module, such as the one described above.

[0090] Step 1600 detects a drag operation associated with an object. Examples of how this can be done are provided above. For example, the drag operation can be detected in conjunction with a cross-slide gesture. Step 1602 presents a partial portion of visual indicia associated with committing an object-related action. Examples of how this can be done are provided above. Step 1604 ascertains whether the drag operation continues. If the drag operation has not continued, the method can, in an event that a user has not terminated the drag operation, return to step 1602. In the event that user has terminated the drag operation, as by returning the object to its original position, the method can terminate. On the other hand, if the drag operation continues, step 1606 ascertains whether a distance threshold associated with an object-related action has been reached. If not, the method can return to step 1602. By doing so, more of the visual indicia can be exposed in accordance with the distance that the object has been dragged. If, on the other hand, a distance threshold associated with an object-related action has been reached, step 1608 presents a complete visual indicia associated with committing the object-related action. By doing so, the visual indicia can inform the user, visually, that the object-related action can be committed as by the user removing her finger from the object.

[0091] Having described an example visual indicia associated with committing an object-related action associated with a cross-slide gesture, consider now an example device that can be utilized to implement one more embodiments described above.

Example Device

[0092] FIG. 17 illustrates various components of an example device 1700 that can be implemented as any type of portable and/or computer device as described with reference to FIGS. 1 and 2 to implement embodiments of the gesture techniques described herein. Device 1700 includes communication devices 1702 that enable wired and/or wireless communication of device data 1704 (e.g., received data, data that is being received, data scheduled for broadcast, data packets of the data, etc.). The device data 1704 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 1700 can include any type of audio, video, and/or image data. Device 1700 includes one or more data inputs 1706 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.

[0093] Device 1700 also includes communication interfaces 1708 that 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 1708 provide a connection and/or communication links between device 1700 and a communication network by which other electronic, computing, and communication devices communicate data with device 1700.

[0094] Device 1700 includes one or more processors 1710 (e.g., any of microprocessors, controllers, and the like) which process various computer-executable or readable instructions to control the operation of device 1700 and to implement the gesture embodiments described above. Alternatively or in addition, device 1700 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 1712. Although not shown, device 1700 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.

[0095] Device 1700 also includes computer-readable media 1714, such as one or more memory components, 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 1700 can also include a mass storage media device 1716.

[0096] Computer-readable media 1714 provides data storage mechanisms to store the device data 1704, as well as various device applications 1718 and any other types of information and/or data related to operational aspects of device 1700. For example, an operating system 1720 can be maintained as a computer application with the computer-readable media 1714 and executed on processors 1710. The device applications 1718 can include a device manager (e.g., 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, etc.), as well as other applications that can include, web browsers, image processing applications, communication applications such as instant messaging applications, word processing applications and a variety of other different applications. The device applications 1718 also include any system components or modules to implement embodiments of the gesture techniques described herein. In this example, the device applications 1718 include an interface application 1722 and a gesture-capture driver 1724 that are shown as software modules and/or computer applications. The gesture-capture driver 1724 is representative of software that is used to provide an interface with a device configured to capture a gesture, such as a touchscreen, track pad, camera, and so on. Alternatively or in addition, the interface application 1722 and the gesture-capture driver 1724 can be implemented as hardware, software, firmware, or any combination thereof.

[0097] Device 1700 also includes an audio and/or video input-output system 1726 that provides audio data to an audio system 1728 and/or provides video data to a display system 1730. The audio system 1728 and/or the display system 1730 can include any devices that process, display, and/or otherwise render audio, video, and image data. Video signals and audio signals can be communicated from device 1700 to an audio device and/or to a display device via an RF (radio frequency) link, S-video link, composite video link, component video link, DVI (digital video interface), analog audio connection, or other similar communication link. In an embodiment, the audio system 1728 and/or the display system 1730 are implemented as external components to device 1700.

Alternatively, the audio system 1728 and/or the display system 1730 are implemented as integrated components of example device 1700.

Conclusion

5 **[0098]** Cross slide gestures for touch displays are described. In at least some embodiments, cross slide gestures can be used on content that pans or scrolls in one direction, to enable additional actions, such as content selection, drag and drop operations, and the like.

10 **[0099]** In one or more embodiments, a cross slide gesture can be performed by dragging an item or object in a direction that is different from a scrolling direction. The different-direction drag can be mapped to additional actions or functionality. In one or more embodiments, one or more thresholds can be utilized, such as a distance threshold, in combination with the different-direction drag, to map to additional actions or functionality.

[00100] In at least some embodiments, so-called speed bumps can be used to provide a user with an understanding or awareness of the thresholds.

15 **[00101]** Although the embodiments have been described in language specific to structural features and/or methodological acts, it is to be understood that the embodiments defined in the appended claims are not necessarily limited to the specific features or acts described. Rather, the specific features and acts are disclosed as example forms of implementing the claimed embodiments.

CLAIMS

What is claimed is:

1. A method comprising:
detecting a gesture slide input relative to a display device of a computing
5 device;
ascertaining whether a gestural slide input direction is different from a
panning or scrolling direction; and
responsive to the gestural slide input being in a direction that is different
from the panning or scrolling direction, performing an object-related action.
- 10 2. The method of claim 1, wherein the panning or scrolling direction is
generally vertical along the display device.
3. The method of claim 1, wherein the panning or scrolling direction is
generally horizontal along the display device.
4. The method of claim 1, wherein the object-related action comprises
15 an object selection.
5. The method of claim 1, wherein the object-related action comprises a
re-arrangement action.
6. The method of claim 1, wherein performing an object-related action
comprises performing one of a plurality of object-related actions which are
20 accessible via the gestural slide input being in a direction that is different from the
panning or scrolling direction.
7. The method of claim 1, wherein the direction that is different from
the panning or scrolling direction comprises a direction that is generally orthogonal
relative to the panning or scrolling direction.
- 25 8. The method of claim 1, wherein said performing an object-related
action is performed responsive to detecting a threshold trigger associated with the
gesture slide input.
9. One or more computer readable storage media embodying computer
readable instructions which, when executed, implement a method comprising:
30 detecting a drag direction associated with a drag operation;

ascertaining whether the drag direction is different from a panning direction;
and

responsive to the drag direction being different than the panning direction,
performing an object-rearrangement action.

- 5 10. The one or more computer readable storage media of claim 9,
wherein detecting a drag direction is performed by detecting a drag direction
associated with a touch gesture.