SMITH070+

USER INTERFACE SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] If any definitions (e.g. figure reference signs, specialized terms, examples, data, information, etc.) from any related material (e.g. parent application, other related application, material incorporated by reference, material cited, extrinsic reference, etc.) conflict with this application (e.g. abstract, description, summary, claims, etc.) for any purpose (e.g. prosecution, claim support, claim interpretation, claim construction, etc.), then the definitions in this application shall apply.

BACKGROUND AND FIELD OF INVENTION

[0002] Embodiments of the present invention generally relate to consumer electronic devices, particularly cell phones, tablets, and other mobile devices (however, applicability is also relevant to non-mobile devices).

BRIEF SUMMARY

[0003] A system, method, and computer program product are provided for a touch or pressure signal-based interface. In operation, a touch or pressure signal is received in association with a touch interface of a device. To this end, a user experience is altered, utilizing the signal. In different embodiments, various features may be further incorporated in association with the system, method, and computer program product, for improvement purposes.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

[0004] So that the features of various embodiments of the present invention can be understood, a more detailed description, briefly summarized above, may be had by reference to various embodiments, some of which are illustrated in the accompanying drawings. It is to be noted, however, that the accompanying drawings illustrate only embodiments and are therefore not to be considered limiting of the scope of various embodiments of the invention, for the invention may admit to other effective embodiments. The following detailed description makes reference to the accompanying drawings that are now briefly described.

[0005] Figure **1** illustrates a network architecture, in accordance with one embodiment.

[0006] Figure **2** shows a representative hardware environment that may be associated with the servers and/or clients of Figure **1**, in accordance with one embodiment.

[0007] Figure **3** shows a method for altering a user experience based on a received signal, in accordance with one embodiment.

[0008] Figure **4** shows a method for defining a selection made within a user interface based in part on contact pressure, in accordance with one embodiment.

[0009] Figure **5** shows a pressure-sensitive user interface for making a selection, in accordance with one embodiment.

[0010] Figure **6** shows a method for determining the context of a contact pressure-based selection and choosing an appropriate selection function, in accordance with one embodiment.

[0011] Figure **7** shows a device having a backtouch interface, in accordance with one embodiment.

[0012] Figure **8** shows a method for providing feedback to the user of a backtouch interface, in accordance with one embodiment.

[0013] Figure **9** shows a pressure-sensitive user interface for making a selection using a backtouch interface, in accordance with one embodiment.

[0014] Figure **10** shows a user interface for defining settings associated with a backtouch interface, in accordance with one embodiment.

[0015] Figure **11** shows a user interface for defining settings associated with a pressure-sensitive interface, in accordance with one embodiment.

[0016] Figure **12** shows a method for assisting a user in defining touch states, in accordance with one embodiment.

[0017] Figure **13** shows a user interface for assisting a user in defining touch states, in accordance with one embodiment.

[0018] Figure **14** shows a user interface for indicating that a backtouch or pressure-sensitive interface is activated, in accordance with one embodiment.

[0019] Figure **15** shows a user interface for defining a backtouch feedback style, in accordance with one embodiment.

[0020] Figure **16** shows an alternative method for defining a selection made within a user interface based in part on contact pressure, in accordance with one embodiment.

[0021] Figure **17** shows a user interface for performing operations on a selection, in accordance with one embodiment.

[0022] Figure **18** shows a method for utilizing contact pressure-based gestures, in accordance with one embodiment.

[0023] Figure **19** shows an example of a contact pressure-based gesture for scrolling a text field, in accordance with one embodiment.

[0024] Figure **20** shows an example of a multitouch pressure gesture for indicating a direction, in accordance with one embodiment.

[0025] Figure **21** shows an example of a multitouch pressure gesture for indicating a rotation, in accordance with one embodiment.

[0026] Figure **22** shows an example of a front-back gesture, in accordance with one embodiment.

[0027] Figure **23** shows an example of a virtual control knob gesture, in accordance with one embodiment.

[0028] Figure **24** shows a method for gesture differentiation, in accordance with one embodiment.

[0029] Figure **25** shows an example of an "anchor and move" gesture, in accordance with one embodiment.

[0030] Figure **26** shows an example of a twist gesture, in accordance with one embodiment.

[0031] Figure **27** shows a method for determining touch states using a touch-sensitive interface, in accordance with one embodiment.

[0032] Figure **28** shows examples of pattern-based gestures, in accordance with one embodiment.

[0033] Figure **29** shows file sharing triggered by a gesture, in accordance with one embodiment.

[0034] Figure **30** shows an example of touch-based device authentication, in accordance with one embodiment.

[0035] Figure **31** shows an example of multi-device facial recognition, in accordance with one embodiment.

[0036] Figure **32** shows two displays combined to function as one, in accordance with one embodiment.

[0037] Figure **33** shows a user interface for defining custom gestures, in accordance with one embodiment.

[0038] Figure **34** shows a user interface for defining pressure inputs for a custom gesture, in accordance with one embodiment.

[0039] Figure **35** shows a user interface for defining touch inputs for a custom gesture, in accordance with one embodiment.

[0040] Figure **36** shows a user interface for defining surface inputs for a custom gesture, in accordance with one embodiment.

[0041] Figure **37** shows a user interface for summarizing contact inputs for a custom gesture, in accordance with one embodiment.

[0042] Figure **38** shows a user interface for defining system inputs for a custom gesture, in accordance with one embodiment.

[0043] Figure **39** shows a pressure-sensitive user interface containing a virtual trackball, in accordance with one embodiment.

[0044] Figure **40** shows a device with slide pads, in accordance with one embodiment.

[0045] Figure **41** shows an example of a slide pad gesture for performing an auxiliary function, in accordance with one embodiment.

[0046] Figure **42** shows a user interface for providing quick access to favorite operations, in accordance with one embodiment.

[0047] Figure **43** shows a device with a non-uniform display, in accordance with one embodiment.

[0048] Figure **44** shows a light valve technique, in accordance with one embodiment.

[0049] Figure **45** shows a device display mixing color pixels with black and white pixels, in accordance with one embodiment.

[0050] Figure **46** shows a device display mixing color with black and white across a matrix, in accordance with one embodiment.

[0051] Figure **47** shows a modulated display, in accordance with one embodiment.

[0052] Figure **48** shows a device with a display having a non-uniform backlight, in accordance with one embodiment.

[0053] Figure **49** shows a device having a removable back sensor or display, in accordance with one embodiment.

[0054] Figure **50** shows a user interface containing an adaptive soft keyboard, in accordance with one embodiment.

[0055] Figure **51** shows a user interface with context-sensitive backlighting, in accordance with one embodiment.

[0056] Figure **52** shows a 3D layered user interface, in accordance with one embodiment.

[0057] Figure **53** shows a device which may be converted into a 3D layer enabled device, in accordance with one embodiment.

[0058] Figure **54** shows a 3D layered user interface for augmented reality, in accordance with one embodiment.

[0059] Figure **55** shows a device with user gaze tracking sensors, in accordance with one embodiment.

[0060] Figure **56** shows the use of eye tracking to generate 3D depth cues, in accordance with one embodiment.

[0061] Figure **57** shows a layered display, in accordance with one embodiment.

[0062] Figure **58** shows an automultoscopic display, in accordance with one embodiment.

[0063] Figure **59** shows a layered display containing depth fused 3D images, in accordance with one embodiment.

[0064] Figure **60** shows a light field camera, in accordance with one embodiment.

[0065] Figure **61** shows a 3D layered user interface with a floating soft keyboard, in accordance with one embodiment.

[0066] Figure **62** shows a backtouch enabled 3D layered user interface, in accordance with one embodiment.

[0067] Figure **63** shows a tactile feedback enabled display, in accordance with one embodiment.

[0068] Figure **64** shows a watch-based display extension, in accordance with one embodiment.

[0069] Figure **65** shows a wireless storage ring and key, in accordance with one embodiment.

[0070] Figure **66** shows a 3D layered user interface having a clipboard layer, in accordance with one embodiment.

[0071] Figure **67** shows a 3D layered user interface having an operation layer, in accordance with one embodiment.

[0072] Figure **68** shows a user interface for providing touch feedback, in accordance with one embodiment.

[0073] Figure **69** shows a user interface for providing increased security, in accordance with one embodiment.

[0074] Figure **70** shows a user interface for presenting a contextual menu, in accordance with one embodiment.

DETAILED DESCRIPTION

[0075] Figure **1** illustrates a network architecture **100**, in accordance with one embodiment. As shown, a plurality of networks **102** is provided. In the context of the present network architecture **100,** the networks **102** may each take any form including, but not limited to a local area network (LAN), a wireless network, a wide area network (WAN) such as the Internet, peer-to-peer network, etc.

[0076] Coupled to the networks **102** are servers **104** which are capable of communicating over the networks **102**. Also coupled to the networks **102** and the servers **104** is a plurality of clients **106.** Such servers **104** and/or clients **106** may each include a desktop computer, lap-top computer, hand-held computer, mobile phone, personal digital assistant (PDA), tablet computer, peripheral (e.g. printer, etc.), any component of a computer, and/or any other type of logic. In order to facilitate communication among the networks **102**, at least one gateway **108** is optionally coupled therebetween.

[0077] Figure **2** shows a representative hardware environment that may be associated with the servers **104** and/or clients **106** of Figure **1**, in accordance with one embodiment. Such figure illustrates a typical hardware configuration of a mobile device in accordance with one embodiment having a central processing unit **210**, such as a microprocessor, and a number of other units interconnected via a system bus **212**.

[0078] The mobile device shown in Figure **2** includes a Random Access Memory (RAM) **214**, Read Only Memory (ROM) **216**, an I/O adapter **218** for connecting peripheral devices such as disk storage units **220** to the bus **212**, a user interface adapter **222** for connecting a keyboard **224**, a mouse **226**, a speaker **228**, a microphone **232**, and/or other user interface devices such as a touch screen (not shown) to the bus **212**, communication adapter **234** for connecting the mobile device to a communication network **235** (e.g., a data processing network) and a display adapter **236** for connecting the bus **212** to a display device **238**.

[0079] The mobile device may have resident thereon any desired operating system. It will be appreciated that an embodiment may also be implemented on platforms and operating systems other than those mentioned. One embodiment may be written using JAVA, C, Objective C, and/or C++ language, or other programming languages, along with an object oriented programming methodology. Object oriented programming (OOP) has become increasingly used to develop complex applications.

[0080] Of course, the various embodiments set forth herein may be implemented utilizing hardware, software, or any desired combination thereof. For that matter, any type of logic may be utilized which is capable of implementing the various functionality set forth herein.

[0081] Figure **3** shows a method **300**, in accordance with one embodiment. As an option, the method may be implemented in the context of the architecture and environment of any subsequent Figure(s). Of course, however, the method may be implemented in any desired environment.

[0082] As shown in operation **302**, a signal is received in association with a touch interface of a device. In one embodiment, the signal may include a pressure signal which is indicative of a magnitude of pressure. In the context of the present description, such pressure signal may include any signal that is a function of or related to a pressure applied to the device. In one embodiment, such pressure signal may be indicative of a magnitude of pressure being applied to a touch interface of the device. In various embodiments, such pressure signal may be generated by a pressure sensor including, but not limited to those described in the context of subsequently described embodiments, or any other mechanism, for that matter, that is capable of generating the pressure signal.

[0083] In other embodiments, the signal may include a touch signal indicative of a touch interface being touched. In the context of the present description, such touch signal may or may not be indicative of a magnitude of pressure being applied to a touch interface of the device. For example, such touch signal may, in one embodiment, simply

indicate whether applied pressure is sensed, or not, e.g. not necessarily gauge any more than two pressure states, including pressure-sensed and pressure-not-sensed, etc.

[0084] Also in the context of the present description, the aforementioned device may include any device including, but not limited to those described in the context of this and/or subsequently described embodiments. Further, in various embodiments, the touch interface may or may not be combined with a display. For example, if the touch interface is combined with a display, the touch interface may include a touch screen. Thus, the touch interface may, in various embodiments, include, but is not limited to a touch screen or any other interface responsive to touch that is described in the context of this and/or subsequently described embodiments.

[0085] In one optional embodiment, the touch interface may be positioned on a first face of a device that includes a display on a second face thereof. Further, the first face and second face may include any respective faces (e.g. front, back, left side, right side, top side, bottom side, etc.) that are different. Just by way of example, in various embodiments, the first face and the second face may be on opposite faces of the device of the device. Further, the first face may include a side face and the second face may include a front face of the device. Even still, the first face may include a back face and the second face may include a front face of the device. Of course, the touch interface may be positioned on the same face as the display, in other embodiments.

[0086] As indicated in operation **304**, a user experience may be altered, utilizing the signal. In the context of the present description, the user experience may include any aspect of the device that is capable of being directly or indirectly experienced by a user including, but not limited to those described in the context of this and/or subsequently described embodiments.

[0087] For example, in the context of an embodiment involving the pressure signal, the user experience may be altered as a function of the magnitude of the pressure being applied to the touch interface, utilizing the pressure signal. Just by way of example, in various embodiments, the user experience may be altered by causing input to the device, by causing output from the device, by causing processing by the device, etc. In other

embodiments involving a touch signal, the user experience may be altered in any manner, utilizing the touch signal (dependent or independent of any fluctuation of pressure).

[0088] Of course, the foregoing embodiments may be altered by adding and/or removing various features. For example, in one embodiment, the pressure signal may be received which is indicative of the magnitude of pressure being applied to a touch interface positioned on a first face of a device that includes a display on a second face thereof. To this end, the user experience may be altered as a function of the magnitude of the pressure being applied to the touch interface, utilizing the pressure signal.

[0089] In another embodiment, a pressure signal may be received which is indicative of a magnitude of pressure being applied to a touch screen. To this end, the user experience may be altered as a function of the magnitude of the pressure being applied to the touch screen, utilizing the pressure signal. In still yet another optional embodiment, a touch signal may be received which is indicative of a touch interface being touched, where the touch interface is positioned on a first face of a device that includes a display on a second face thereof. To this end, the user experience may be altered, utilizing the touch signal. Again, any of the features described above (or hereinafter, for that matter) may or may not be combined in any desired manner.

[0090] More illustrative information will now be set forth regarding various optional architectures and features with which the foregoing techniques discussed in the context of any of the present or previous figure(s) may or may not be implemented, per the desires of the user. For instance, various optional examples and/or options associated with the operations **302** and/or **304**, and/or other optional features have been and will be set forth in the context of a variety of possible embodiments. It should be strongly noted, however, that such information is set forth for illustrative purposes and should not be construed as limiting in any manner. Any of such features may be optionally incorporated with or without the inclusion of other features described.

[0091] Figure **4** shows a method **400** for defining a selection made within a user interface based in part on contact pressure, in accordance with one embodiment. As an option, the method **400** may be implemented in the context of the architecture and

environment of the previous Figures or any subsequent Figure(s). Of course, however, the method **400** may be carried out in any desired environment. It should also be noted that the aforementioned definitions may apply during the present description.

[0092] As shown, it is determined whether sufficient initial contact pressure is being exerted on an interaction surface. See determination **402**.

[0093] In the context of the present description, an interaction surface refers to a surface through which a user may interact with a device. It may take up the majority of a device face, or just a subset. In various embodiments, an interaction surface may be sensitive to one or more types of interaction, including but not limited to, contact, pressure, and/or proximity. In one embodiment, an interaction surface is receptive to multitouch input. In another embodiment, an interaction surface may be non-planar. In yet another embodiment, an interaction surface may be transparent, and combined with a display. In still another embodiment, an interaction surface may be replaceable. Some embodiments of an interaction surface may have more than one texture, to provide tactile guides to a user when used on the back face of a device. Other embodiments of an interaction surface are non-rectangular.

[0094] In various embodiments, an interaction surface may utilize one or more forms of technology to sense contact. These contact-sensing technologies may include, but are not limited to, capacitive, resistive, optical, surface acoustic wave based, and/or any other contact sensing technologies now known or later developed.

[0095] In various embodiments, an interaction surface may utilize one or more forms of technology to sense proximity. These proximity-sensing technologies may include, but are not limited to, capacitive, resistive, eddy current, optical, ultrasonic, heat, electric field based, and/or any other contact sensing technologies now known or later developed.

[0096] In various embodiments, an interaction surface may utilize one or more forms of technology to sense pressure. These pressure-sensing technologies may include, but are not limited to, conductive, resistive, piezoelectric, and/or any other pressure sensing technologies now known or later developed. In some embodiments, an interaction

surface may be able to only detect the total pressure being exerted on the surface. In other embodiments, an interaction surface may be able to discern the contact pressures associated with one or more points or areas of contact.

[0097] In the context of the present description, contact pressure is the pressure associated with one or more contact points or contact areas, the pressure being exerted on an interaction surface. For example, in one embodiment, contact pressure may be the pressure exerted on an interaction surface by a single finger press.

[0098] In the context of the present description, a contact area refers to the area of an interaction surface which is in physical contact with an implement of interaction. In various embodiments, the contact area may be described, stored, and processed, as a collection of cells that are part of an array. In one embodiment, this array may directly correspond to pixels of a display. In another embodiment, the array may correspond to a grid of discrete areas on an interaction surface which sense contact, pressure, or both. In still another embodiment, the contact area is represented as a primitive shape (e.g. circle, square, etc.) which best fits the actual area in contact with an implement of interaction.

[0099] In the context of the present description, an implement of interaction refers to an object which is detectable by an interaction surface in one or more ways (e.g. proximity, contact, pressure, etc.). In various embodiments, implements of interaction may include, but are not limited to, fingers, skin, a stylus, and/or any other object with which a user may interact with a device.

[00100] In the context of the present description, a contact point refers to a single location on an interaction surface which is associated with physical contact with an implement of interaction. In various embodiments, the contact point may be determined from a contact area. For example, in one embodiment, a contact point may be located in the center of its associated contact area. In another embodiment, a contact point may be located on an edge of its associated contact area.

[00101] Determining whether sufficient initial contact pressure is being exerted on an interaction surface depends on a threshold contact pressure. In some embodiments, the

smallest degree of contact pressure may be sufficient, such that the definition of a selection may be triggered by any contact. In other embodiments, the definition of a selection does not occur until a non-negligible threshold contact pressure has been detected. In one embodiment, this threshold contact pressure may be defined by the user. In this way, accidental definitions of a selection may be avoided.

[00102] In some embodiments, determination **402** may be performed only after certain events have occurred. For example, in one embodiment, the determination may be made only after receipt of an input or combination of inputs. Possible inputs include, but are not limited to, object proximity to the interaction surface, activation of a hardware switch or button, activation of a software switch or button, and/or any other input capable of being detected. In one embodiment, the determination may not be made until the host device is woken from a sleep state. In another embodiment, the determination itself may also wake the host device from a sleep state.

[00103] If the result of determination **402** is that there is sufficient initial contact pressure being exerted on an interaction surface, the current contact pressure level is determined. See operation **404**.

[00104] In the context of the present description, a contact pressure level is the pressure reading generated from the raw data received from a touch sensor. In various embodiments, the contact pressure level may be represented by a number. For example, in some embodiments, the contact pressure level may be represented by an integer. In another embodiment, the contact pressure level may be represented by a floating-point number. In various embodiments, the contact pressure level may be represented by a percentage of the measurable range of contact pressures. In one embodiment, the contact pressure level may be represented by a unit less number. In another embodiment, the contact pressure level may be represented by an actual measure of the contact pressure, associated with a unit of pressure.

[00105] In various embodiments, the contact pressure level may be represented by a touch state. In the context of the present description, a touch state refers to a predefined subset of the range of measurable contact pressures. For example, in one embodiment,

contact pressure levels may be represented as one of four predefined touch states: no touch, light touch, medium touch, and heavy touch. As an option, the lowest touch state may require some form of proximity. In some embodiments, the user may be able to specify how the measurable range of contact pressures is partitioned across a predefined number of touch states. In one embodiment, the number and boundaries of touch states may be defined system-wide, for all applications. In another embodiment, the number and boundaries of touch states may be defined on a per-application basis.

[00106] As shown, the selection area is calculated. See operation **406**. In the context of the present description, the selection area refers to a region of the display, or a region of an object pictured on the display, which has been chosen for selection. The calculation of the selection area may be based upon one or more factors, including, but not limited to, a contact pressure level, a touch state, a contact area, a contact point, a selection boundary, a selection boundary geometry, and/or any other information derived from user interaction. In some embodiments, the selection area may be described as the pixels contained within a selection boundary.

[00107] In the context of the present description, a selection boundary describes the boundary of an associated selection area, and is based upon a selection boundary geometry. Additionally, in the context of the present description, a selection boundary geometry refers to the shape of a selection boundary. In some embodiments, a selection boundary geometry may not have reference to actual size or location. For example, in one embodiment, a selection boundary geometry may be described using fractional coordinates on a unit square.

[00108] In various embodiments, a selection area may be described, stored, and/or processed as the area contained within a selection boundary. A selection boundary may be represented by a selection boundary geometry, a screen location, and one or more transformations, such as scale or orientation. In one embodiment, the screen location is the point on which the transformed selection boundary geometry is centered.

[00109] In some embodiments, the selection area may be described, stored, and/or processed, as a collection of cells that are part of an array. In one embodiment, this array

may directly correspond to pixels of a display. In another embodiment, the array may correspond to a grid based on a coordinate system specific to an object being displayed. In still another embodiment, the selection area may be described, stored, and processed as a mathematical function that defines the boundary of the selection area.

[00110] In one embodiment, the shape of a selection boundary may be one of a plurality of selection boundary geometries predefined within an operating system. In another embodiment, the shape of a selection boundary may be predefined within an application, independent of selection boundary geometries defined within the operating system. In yet another embodiment, the selection boundary geometry may be specified by a user. In still another embodiment, the geometry of the selection boundary may depend upon at least one form of user input, including but not limited to contact pressure, number of fingers in contact with the screen, device orientation, location of user fingers, and/or any other form of user input. Finally, in another embodiment, the geometry of a selection boundary may depend upon the content within or near a location associated with the selection (e.g. contact point, finger location, focal point of user's eyes, cursor location, etc.).

[00111] In some embodiments, selection boundaries may have geometries based upon simple shapes, including, but not limited to, ovals, circles, triangles, squares, rectangles, and/or higher order polygons. In other embodiments, selection boundaries may be based upon regular shapes (e.g. a star, a plus sign, etc.). In one embodiment, a selection boundary may be based upon the geometry of a contact area.

[00112] There are numerous ways in which a selection boundary may be described, stored, and/or processed. In various embodiments, a selection boundary may be represented by the combination of an established selection boundary geometry with a location, a scale, and/or a rotation.

[00113] In one embodiment, a selection boundary may be described using the vertex coordinates of a selection boundary geometry, in conjunction with location, scale, and/or rotation factors. For example, a rectangular selection boundary geometry may be

described by the coordinates of the four corners. As an option, the vertices of a selection boundary geometry may be described using coordinates within the unit square.

[00114] In another embodiment, a selection boundary may be described using a mathematical function representing an established selection boundary geometry, in conjunction with location, scale, and/or rotation factors. For example, an elliptical selection boundary geometry may be described by an ellipse function whose foci depend upon the scale factor.

[00115] In yet another embodiment, a selection boundary may be described using a spline-based representation of a selection boundary geometry, in conjunction with location, scale, and/or rotation factors. For example, a curled selection boundary geometry may be described using splines combined with location, scale, and rotation factors.

[00116] In other embodiments, a selection boundary may be described by a bitmap representation of a selection boundary geometry, in conjunction with location, scale, and/or rotation factors. For example, in one embodiment, a bitmap representation of a contact area may be used as a selection boundary geometry associated with a selection boundary. In another embodiment, a bitmap representation of an irregular shape may be used as a selection boundary geometry associated with a selection boundary. In this way, geometries which may be difficult to describe using vertices, formulas, or splines may be used as the bases for a selection boundary.

[00117] The selection area is calculated by determining the selection boundary. In various embodiments, the selection boundary may be determined by transforming an appropriate selection boundary geometry as a function of user input, including but not limited to, contact pressure level, contact point, contact area, number of fingers detected, device orientation, etc. Once the selection boundary has been determined, the selection area is known.

[00118] In various embodiments, the selection boundary may be defined by transforming the scale of an appropriate selection boundary geometry as a function of

user input. For example, in one embodiment, a selection boundary with a circular geometry may have a larger radius at higher contact pressure levels. In another embodiment, a rectangular selection boundary geometry may be scaled linearly by a factor proportional to the contact pressure level. In some embodiments, the relationship between selection boundary geometry scale and a contact pressure level is proportional. In other embodiments, the relationship between selection boundary geometry scale and a contact pressure level is inversely proportional, such that higher pressure may make the geometry smaller.

[00119] In various embodiments, the selection boundary may be defined by transforming the orientation of an appropriate selection boundary geometry as a function of user input. For example, in one embodiment, a selection boundary with a star-shaped geometry may be rotated further clockwise at higher contact pressure levels. In some embodiments, the relationship between selection boundary geometry orientation and a contact pressure level is proportional, such that an increase in pressure may result in a clockwise rotation. In other embodiments, the relationship between selection boundary geometry orientation and a contact pressure level is inversely proportional, such that higher pressure may cause a counterclockwise rotation.

[00120] In some embodiments, a selection boundary may have an anchor point. In the context of the present description, an anchor point refers to a point located on or near the selection boundary geometry which remains fixed as the geometry is transformed. In one embodiment, a selection boundary may have a scaling anchor point, which remains fixed as the geometry is scaled. For example, a square selection boundary may have a scaling anchor point in the upper left corner, causing the square to expand down and to the right, keeping the upper left corner stationary.

[00121] In another embodiment, a selection boundary may have a rotational anchor point, which is used as an axis of rotation as the selection boundary geometry is rotated. For example, a star-shaped selection boundary may have a rotational anchor point located in the center, such that it rotates along an axis of symmetry.

[00122] In various embodiments, the calculation of the selection area may depend upon the context of the selection. Specifically, the selection boundary may be dependent upon the type of content on or near a point or area associated with the selection, including but not limited to, a contact point, a contact area, a cursor, a user focus, etc. In various embodiments, the selection boundary geometry and the manner in which it transforms as a function of user input may be context dependent. For example, in one embodiment, a selection made on or near a text field may cause the selection boundary geometry to be rectangular, with a scaling anchor point located in an upper corner appropriate for the text language (e.g. upper left corner for English, upper right corner for Arabic, etc.). In another embodiment, a selection on or near a text field may cause the rectangular boundary geometry to scale in increments of the same scale as the height and width of the text. In this way, it may be clear to the user which text has been selected, as there may never exist a selection boundary which contains only half a line or character.

[00123] In another embodiment, a selection on or near a text field may result in a rectangular selection boundary geometry which scales proportionally to a contact pressure level in a manner similar to how text is selected using a cursor. In this way, a user may select text in a familiar manner, while only having to vary the pressure being exerted on a contact point, for example. In one embodiment, the text selection begins to increase once a predefined contact pressure level has been achieved. As an option, the rate at which the selection grows may be a function of the contact pressure level, once the predefined contact pressure level has been achieved and the selection grown initiated. In another embodiment, the text selection scaling may be tiered, such that at one touch state, the selection grows character by character, at an intermediate touch state the selection grows word by word, and so on, such that the user is able to enlarge the text selection in increments of characters, words, lines, paragraphs, and pages, depending on the touch state. As an option, the selection may also be incremented by sentences.

[00124] In other embodiments, the above-mentioned methods for selecting text using a selection which expands in the direction the text would be read in may also be applied to the selection of other linear subject matter. For example, in one embodiment, this method may be used to select a video segment. As an option, the selection scaling

may be tiered, growing from frame-by-frame to second-by-second to minute-by-minute, and so on. In another embodiment, this method may be used in the selection of a portion of chronologically arranged material. Possible material with a temporal arrangement may include, but is not limited to, a collection of photos, messages in an email account, text messages, web browser history, and/or any other collection of items which may be arranged chronologically.

[00125] In various embodiments, a selection made within a graphic (e.g. photograph, drawing, movie still, bitmapped or rasterized text, etc.) may utilize a default selection boundary geometry with a default scaling anchor point. In one embodiment, the user may be able to predefine the default selection boundary geometry and/or default scaling anchor point, to be used when making a selection within a graphic. In still another embodiment, the default selection boundary geometry and/or default scaling anchor point may be predefined within the operating system. As an option, these defaults may be predefined on a per-application basis.

[00126] In various embodiments, a selection made within a photograph may result in the use of a context-specific selection boundary geometry which transforms in a context specific manner. For example, in one embodiment, a selection made within a photograph containing one or more people may utilize the rough shape of the subjects' faces as a selection boundary geometry. As an option, the selection boundary geometry may be composed of the perimeter of all faces within a certain distance of a point related to the selection (e.g. contact point, user focus, etc.), the distance being proportional to a user input (e.g. contact pressure level, touch state, etc.). Face selection may be used to assist the user in establishing the identity of photographed individuals, allowing future identification to be performed automatically using facial recognition. In one embodiment, facial and other forms of recognition may be performed by a host device, a remote computer, a cloud service, or any other computational resource, or combination of these and other resources.

[00127] In another embodiment, a selection made within a photograph on or near an eye may utilize a rough shape of the eye as a selection boundary geometry. As an option,

the exactness of the method used to automatically detect the shape of an eye may vary as a function of a user input, such as contact pressure level. Eye selection may be used to assist the user in removing the "red eye" effect sometimes seen in photographs.

[00128] In one embodiment, a selection made within a graphic may cause the selection boundary geometry to be defined by the collection of pixels near a point associated with the selection (e.g. contact point, user focus, etc.) which are similar in at least one aspect. Possible aspects include, but are not limited to, color, one or more dimensions of an associated color space, brightness, transparency, and/or any other aspect that may be associated with a pixel. As an option, the degree of similarity between pixels required to influence the selection boundary geometry may be a function of a user input, such as a contact pressure level. In this way, a user may be assisted in separating an object within the graphic from a background or other visual elements.

[00129] In various embodiments, there may exist limits to the amount a selection boundary geometry may be transformed as a function of user input. For example, in one embodiment, the selection boundary geometry may increase in size as the contact pressure level increases, until an upper contact pressure threshold is reached. Contact pressure levels beyond the upper threshold may not increase the size of the selection boundary geometry any further. In another embodiment, the scale of the selection boundary geometry may be limited by a lower contact pressure threshold, such that reducing the contact pressure level below the threshold may not reduce the scale of the selection boundary geometry any further. In yet another embodiment, both upper and lower contact pressure thresholds may be implemented. In other embodiments, this concept of thresholds may be applied to other variable aspects of the selection boundary, such as the orientation of the selection boundary geometry, and as a function of other user input, including but not limited to the location of a contact point, the size of a contact area, user focus, and/or any other type of user input.

[00130] In one embodiment, the user may be able to predefine one or more contact pressure thresholds associated with the limits of a transformation of the selection boundary geometry. As an option, the user may also predefine the limits of the

transformation, in addition to the maximum and/or minimum contact pressure level which may affect said transformation. In another embodiment, these contact pressure thresholds and/or transformation limits may be predefined within the operating system. In still another embodiment, the contact pressure thresholds may be automatically optimized over time, by observing the typical range of pressures the user exerts in various use scenarios.

[00131] In one embodiment, the selection boundary geometry and/or transformation of the selection boundary geometry may depend upon the proximity of an object to an interaction surface, rather than an aspect of physical contact with an interaction surface.

[00132] In various embodiments, the selection boundary may depend, at least in part, upon the user focus. In the context of the present description, user focus refers to a point or area on a display which is the visual and/or interactive focus of the user. For example, in one embodiment, the user focus may be determined by tracking the eyes of a user with one or more device-mounted cameras, and calculating where the user is looking. This may also be referred to as user gaze, or user gaze tracking. As an option, eye movement may also be monitored, to ascertain the manner in which the user is using the device (e.g. methodically reading text, quickly scanning text, examining a picture, visually searching for an icon, etc.). In another embodiment, the user focus may be defined as the location of a cursor in a text environment. In still another embodiment, the user focus may be established through user input, such as a finger tap.

[00133] Various aspects of the selection boundary may depend upon the user focus. For example, in one embodiment, the user focus may be used to determine the context of a selection. In another embodiment, the user focus may be used to determine the location of a selection, allowing the user to make a selection without having to change the location of a contact point. In still another embodiment, the tracking of eye movement may be used to determine the type of selection (e.g. reading eye movement may trigger a text selection, etc.).

[00134] Once the selection boundary has been determined using an appropriate selection boundary geometry that has been transformed as a function of a user input, the selection area may be calculated.

[00135] As shown, the selection area is displayed. See operation **408**. In various embodiments, a selection area may be displayed using one or more selection functions. In the context of the present description, a selection function refers to a function of one or more inputs which determines one or more properties of a selection. For example, in one embodiment, possible selection function inputs may include, but are not limited to, touch state, contact pressure level, selection state, finger locations, device orientation, user focus, pressure uncertainty, touch uncertainty, contact point, contact area, gestures, application type and/or context, application flags, operating system flags, and/or any other form of user or system input.

[00136] Within the context of the present description, pressure uncertainty refers to any data associated with the uncertainty inherent in measuring pressure exerted on an interaction surface. Pressure uncertainty may include, but is not limited to, the uncertainty of a measured contact pressure level, a multidimensional array of values representing the variance of contact pressure measured for each cell/pixel in the vicinity of a contact point, an uncertainty value inherent to a particular pressure sensor, and/or any other type of data associated with uncertainty in pressure measurement.

[00137] Within the context of the present description, touch uncertainty refers to any data associated with the uncertainty inherent in sensing contact with an interaction surface. Touch uncertainty may include, but is not limited to, the error range associated with the location of a contact point, a multidimensional array of values representing the probabilities of contact measured for each cell/pixel in the vicinity of a contact point, an uncertainty value inherent to a particular touch sensor, and/or any other type of data associated with uncertainty in sensing contact.

[00138] Within the context of the present description, a selection state refers to whether a selection is 'dynamic' and still able to vary as a function of user input, or 'static', and not currently varying as a function of user input.

[00139] In various embodiments, a selection function may determine one or more presentation properties of a selection. For example, possible presentation properties may include, but are not limited to, shading, color, contrast, brightness, line style, animation routine, transparency, pattern, border style, area style, and/or any other visual property that may describe an object on a display. Examples of animation routines include, but are not limited to, "marching ants", changing colors, flashing, simulated optical distortions, pulsing highlights, and/or any other routine which varies location, transparency, color, content, and/or any other graphical property over time. In some embodiments, an animation routine may be a function of user input.

[00140] In various embodiments, a selection function may vary one or more presentation properties as a function of contact pressure. In this way, the user has visual feedback as to the level of pressure they are exerting on the device. For example, in one embodiment, the selection function may indicate the contact pressure level by shading the selection with a color which is somewhere between a pair of colors which represent the extrema of measurable contact pressure levels.

[00141] In various embodiments, a selection function may determine one or more fundamental properties of a selection. Within the context of this description, fundamental selection properties may include, but are not limited to, selection boundary geometry, transformation(s) to be applied to the selection boundary geometry as a function of user input, anchor point(s), and/or any other property which may be associated with a selection area.

[00142] In various embodiments, the application of a selection function may display a secondary boundary. Within the context of this description, a secondary boundary is any boundary, radius, or area associated with a selection area, which is not the selection boundary. For example, in one embodiment, a secondary boundary may be the area within which facial detection is being performed, as opposed to the selection boundary, which may be the outline of detected faces. In another embodiment, a secondary boundary may be the area within which eye detection is being performed. In yet another

embodiment, the secondary boundary may represent the pressure uncertainty or touch uncertainty associated with a contact area.

[00143] In various embodiments, the secondary boundary may be displayed in addition to, or instead of, the selection boundary. For example, in one embodiment where face detection is being performed as part of calculating the selection area, the selection boundary may be indicated with a solid color line while the secondary boundary may be indicated with a thin dotted line.

[00144] In various embodiments, the area unique to the secondary boundary (i.e. the portion of the enclosed area which does not overlap with the selection area) may have presentation properties that differ from the selection area. For example, in one embodiment, a user may make a text selection of which the intended boundaries are uncertain. The portion of the text which was almost certainly intended to be selected may be shaded in green, while the text which may or may not have been selected intentionally may be shaded in red. The user intention may be extrapolated from uncertainty associated with the contact and/or contact pressure used to make the selection, in accordance with one embodiment.

[00145] In various embodiments, the secondary boundary may be used to indicate uncertainty associated with a form of machine recognition including, but not limited to, facial recognition, OCR, and/or any other form of machine recognition. For example, faces within a selection which have been recognized with a threshold degree of confidence may be outlined in green, while faces whose recognition is suspect (i.e. multiple viable matches, etc.) may be outlined in red (i.e. a secondary border).

[00146] As shown, after the selection area has been displayed, it is determined whether the selection establishment conditions have been met. See determination **410**.

[00147] Within the context of the present description, selection establishment conditions refer to one or more predefined conditions whose satisfaction may cause the selection state to switch from 'dynamic' to 'static'. For example, in one embodiment, once the selection establishment conditions have been satisfied, the selection may no

longer vary with contact pressure levels until determination **402** once again detects sufficient initial contact pressure. In various embodiments, the selection establishment conditions may be defined within a selection function.

[00148] In various embodiments, one or more selection establishment conditions may depend upon changes in the contact pressure level. For example, in some embodiments, selection establishment may be conditioned upon whether the contact pressure level drops below a predefined level. In one embodiment, this threshold contact pressure level may be defined as being a set amount of pressure less than the current contact pressure level. In this way, a user may reduce the contact pressure a small amount to modify their selection, with further reduction causing a selection establishment condition to be satisfied. In another embodiment, the threshold contact pressure level may be a set pressure amount.

[00149] In various embodiments, a selection establishment condition may depend upon contact pressure velocity, or the rate at which the contact pressure level is changing. For example, in one embodiment, a user may be able to satisfy a selection establishment condition by quickly reducing the contact pressure level to a predefined level within a predefined amount of time. In another embodiment, selection establishment may be conditioned upon whether a user has increased the contact pressure level to a predefined level within a predefined amount of time. In yet another embodiment, a selection may be established by exceeding a predefined contact pressure velocity, without regard to the actual pressure levels. As a specific example, a user may establish their selection by a quick pulse of their finger, without having to reach a particular pressure level.

[00150] In various embodiments, a selection establishment condition may depend upon how much time has elapsed since a particular event. For example, in one embodiment, a selection may be established if the user maintains a contact pressure level for a predefined amount of time. As an option, there may be a predefine tolerance level for pressure variations which may not reset this timing, allowing for small fluctuations in contact pressure which may be difficult for a user to control. In another embodiment, a

selection establishment condition may be satisfied once a predefined amount of time has elapsed since the satisfaction of determination **402**.

[00151] In various embodiments, a selection establishment condition may depend upon forms of user input that are not related to contact pressure. These forms of user input may include, but are not limited to, gestures, a change in the number of fingers in contact with the device, change in device orientation, surpassing a threshold level of accelerations (e.g. wrist flick, etc.), and/or any other form of user input. It should be noted that selection establishment may also be conditioned upon the satisfaction of a combination of any of the previously described conditions.

[00152] If it is determined that the selection establishment conditions have not been met, the selection is updated in a loop through performing operations **404**, **406**, and **408**. The selection may continue to vary as a function of user input until the selection establishment conditions have been met.

[00153] If it is determined that the selection establishment conditions have been met, the selection state is changed from 'dynamic' to 'static'. See operation **412**.

[00154] Setting the selection state to 'static' means the selection is no longer being modified by user input. In various embodiments, the selection function may depend in part on the selection state. For example, in one embodiment, when the selection state is 'static', the secondary boundary may be removed from the display. In another embodiment, a 'static' selection state may cause the selection area to be displayed in a different color, or with a different style of selection boundary. In yet another embodiment, changing a selection state from 'dynamic' to 'static' may cause one or more application and/or operating system flags to be set.

[00155] In various embodiments, changing a selection state from 'dynamic' to 'static' may prompt the user to choose an action to be performed on or with the contents of the selection. For example, in one embodiment, after the desired faces have been highlighted and the user has reduced the contact pressure level below the dynamic selection minimum threshold pressure, they may be prompted to confirm the identity of the

selected faces. In another embodiment, after a text selection has been given a 'static' selection state, the user may be prompted with operations that may be performed on the selected text, including but not limited to copy, cut, paste, spell check, style (e.g. bold, underlined, italic, etc.), font, font size, and/or any other operation which may be performed on text.

[00156] In some embodiments, once a selection has become 'static', it can no longer be modified and may be destroyed upon creation of a new selection. In other embodiments, a user may further modify a static selection by first applying a contact pressure sufficient to satisfy determination **402**. In one embodiment, satisfying determination **402** with a static selection already in existence may change the selection state of the previous selection from static to dynamic, allowing further variation as a function of user input. In another embodiment, satisfying determination **402** with a static selection already in existence may create a new selection, as a function of user input, which is combined with the previously made selection. In yet another embodiment, a user may chose whether to replace a previously made selection or augment (i.e. add to, subtract from, etc.) a previously made selection with a new selection. As an option, the user choice may be indicated through gesture, voice command, toggling a hardware control, toggling a software control, and/or any other form of user input.

[00157] In various embodiments, method **400** may be adapted to allow for the definition, establishment, and modification of selections through other types of user input. Other types of user input may include, but are not limited to, proximity to an interaction surface, number of contact points, gestures, and/or any other form of user input. For example, in one embodiment, a user may expand a text selection by bringing their finger into proximity to the interaction surface.

[00158] Figure **5** shows a pressure sensitive user interface **500** for making a selection, in accordance with one embodiment. As an option, the user interface **500** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **500** may be

implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00159] As shown, the user interface **500** may be utilized in making a selection based in part on pressure exerted by the user upon one or more interaction surfaces located on a device. For example, in one embodiment, a user's finger **502** may exert a small amount of force over contact area **504**, which is centered on contact point **506** located on a pressure sensitive touch screen. If the contact pressure level is sufficient, then a selection area **508** having a selection boundary **510** is calculated and displayed. As the contact pressure is increased, the selection boundary expands to encompass a larger selection area **512**.

[00160] Figure **6** shows a method **600** for determining the context of a contact pressure-based selection and choosing an appropriate selection function, in accordance with one embodiment. As an option, the method **600** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the method **600** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00161] As shown, the context determination zone is defined. See operation **602**. Within the context of the present description, a context determination zone refers to the area that is examined while determining the context of a selection. In various embodiments, the context determination zone may be initially defined as a function of one or more elements of user input upon which the selection itself is based. For example, in one embodiment, the context determination zone may be initially defined to be the contact area. In another embodiment, the context determination zone may be a predefined shape (e.g. circle, square, etc.) of predefined size, centered on the contact point. As an option, the user may be able to specify the initial size and shape of the context determination zone. In yet another embodiment, the context determination zone may be located such that the contact point is off-center, to provide a more intuitive user experience (e.g. the zone may be offset to compensate for the user's view being obstructed by their finger, etc.).

[00162] In still another embodiment, the context determination zone may have a predefined size. As an option, the predefined initial size for a context determination zone may be chosen such that it is smaller than any user interface elements which may be detected. As an additional option, the predefined initial size may be chosen such that it is large enough to assist a user in distinguishing between potential selection targets which are close to each other.

[00163] As shown, once the context determination zone has been defined, it is then determined whether there is a user interface control widget within the context determination zone. See determination **604**.

[00164] Within the context of the present description, a user interface control widget refers to any element within a user interface which may receive user input, which are typically non-textual. For example, in various embodiments, user interface control widgets may include, but are not limited to, buttons, radio boxes, check boxes, drop down lists, sliders, tabs, and/or any other non-textual UI widget.

[00165] In some embodiments, user interface control widgets may also include hyperlinks which are isolated from a text field. For example, text that responds to user interaction like a button, but is not located within a selectable text field. Selectable text fields and hyperlinks embedded within a text field (e.g. links within a web page, links within a word processing document, etc.) are specifically excluded from this definition of user interface control widgets, as they are treated as separate cases within this method.

[00166] In various embodiments, the determination as to whether there is a user interface control widget within the context determination zone begins with determining whether there is a UI control widget located at the contact point upon which the zone is based. In this way, time is not wasted determining the context of a well-aimed contact point.

[00167] If a UI control widget is not located at the contact point, then it may be determined if there is a UI control widget in the vicinity, within the context determination zone. In various embodiments, this determination may be performed using the bounding

boxes associated with user interface elements. Within the context of the present description, a bounding box refers to a rectangle within which a user interface control widget may reside. In some embodiments, a bounding box may be the smallest rectangle which may contain a particular user interface control widget. In this way, detection of user interface control widgets whose shape is not rectangular is simplified.

[00168] For example, in one embodiment, the determination as to whether there is a user interface control widget within the context determination zone may be performed by determining if the boundary of the context determination zone intersects with the bounding box of any user interface control widgets. As an option, if the initial size of the context determination zone is larger than a predefined minimum (e.g. smallest bounding box associated with a typical UI control widget, smallest bounding box present within the present UI, etc.), the determination may also verify the absence of any bounding boxes completely contained within the context determination zone, on the initial pass.

[00169] In another embodiment, the determination may be performed by calculating the distances between the center of the context determination zone and the center of every UI control widget present in the interface, and comparing them with a distance related to the size of the context determination zone.

[00170] If it is determined in **604** that there is at least one user interface control widget within the context determination zone, then a context sensitive UI control widget selection function is chosen. See operation **606**. In various embodiments, once an appropriate selection function is chosen, a selection is defined based upon user input. In one embodiment, the selection is defined using method **4**.

[00171] Within the context of the present description, a context sensitive selection function refers to a selection function adapted for use in making a selection in a context which has been automatically determined. In various embodiments, a context sensitive selection function may include logic to use a different context sensitive selection function to define the user's selection, based upon user input. In this way, a user may be able to indicate that the initial context determination is not correct, and that another selection function should be used.

[00172] For example, in one embodiment, a context sensitive UI control widget selection function may include logic such that, if the contact pressure level exceeds a predefined limit P_0, further definition of the selection is handled by a different context sensitive selection function, such as one adapted for selecting hyperlinks. In another embodiment, a contact pressure level exceeding P_0 may cause the context determination zone to expand, and the process of choosing an appropriate selection function to resume, ignoring the UI control widget previously detected within the zone. As an option, future determinations may ignore all UI control widgets. In this way, a context-aware user selection initiated with a contact point in the vicinity of a UI control widget may be redirected to a different type of subject matter without forcing the user to reposition their finger, for example.

[00173] In various embodiments, a context sensitive UI control widget selection function may depend upon the context determination zone. For example, in one embodiment, a context sensitive UI control widget selection function may use the context determination zone as a secondary boundary. In this way, the selection function may be able to continue searching the context determination zone for additional UI control widgets.

[00174] In various embodiments, a context sensitive UI control widget selection function may include logic to handle the case where there is more that one UI control widget within the context determination zone. In one embodiment, the selection function may choose the UI control widget whose center is closest to the center of the context determination zone. In another embodiment, the selection function may choose the UI control widget whose bounding box has greatest overlap with the context determination zone. In still another embodiment, the selection function may choose the UI control widget whose bounding box has an edge or corner closest to the center of the context determination zone.

[00175] In various embodiments, a context sensitive UI control widget selection function may include logic to allow the user to toggle between multiple widgets found within the context determination zone. For example, in one embodiment, the selection

function may include a list of all UI control widgets found at least partially within the context determination zone. As an option, the widgets may be arranged in the list in order of distance from the center of the context determination zone. In another embodiment, the selection function may traverse the list sequentially, selecting a different widget. As an option, the list traversal may occur as a function of user input, including but not limited to, gestures, contact pressure velocity, accelerometer data, and/or any other user input. In still another embodiment, switching between different widgets found within the context determination may occur after a predefined amount of time has elapsed, and the selection establishment conditions have not yet been satisfied. In yet another embodiment, a context sensitive UI control widget selection function may use user focus to choose between multiple UI control widgets found within the context determination zone.

[00176] In various embodiments, a context sensitive UI control widget selection function may visually indicate the selection of a widget to the user before the selection has been established. For example, in one embodiment, the selection function may highlight the outline of the widget. In another embodiment, the selection function may shade the widget using a predefined mask of the widget. In still another embodiment, the selection function may shade and/or outline the widget bounding box. In yet another embodiment, the selection function may change and aspect of text associated with the widget, aspects including but not limited to font, text style, text color, and/or any other text property. As a specific example, a context sensitive UI control widget selection function may cause the name of a button to be displayed in blue italics if it is the current context-based selection derived from user input. In this way, the user may see what control is about to be activated before satisfying the selection establishment conditions.

[00177] In various embodiments, a context sensitive UI control widget selection function may have selection establishment conditions that depend upon the type of user interface control widget that is being selected. Some user interface control widgets are multi-state in nature (e.g. sliders, drop down lists, etc.), while others are bi-state (e.g. buttons, check boxes, etc.). In some embodiments, it may be assumed that when a user satisfies the selection establishment conditions for a bi-state UI control widget, they

intend for the state of the widget to be toggled. The selection establishment conditions for multi-state UI control widgets, on the other hand, may include the election of a widget state.

[00178] For example, in one embodiment, the selection establishment conditions for a multi-state widget may include one or more primary conditions to establish that the user intends for that particular multi-state widget to be selected, and a secondary condition, to establish the state of that widget.

[00179] In various embodiments, the secondary condition may be satisfied (and the widget state chosen) through user input which includes, but is not limited to, contact pressure. As an option, after the primary selection establishment conditions have been satisfied, the selection function may disregard pre-established pressure triggers (e.g. P_0, global pressure triggers, etc.) until the secondary selection establishment condition has been satisfied.

[00180] For example, in one embodiment, once the primary conditions for selecting a slider widget have been satisfied, a user may select a slider value by varying the contact pressure level between two predefined threshold pressures which represent the slider state extrema. In another embodiment, the contact pressure level at the time of satisfaction of the primary conditions may be used as a zero point, represented by the middle of the slider values. Increasing the contact pressure may increase the slider value, decreasing the contact pressure may decrease the slider value. In other embodiments, these methods may be used to traverse the items within a drop down list, or any other multi-state UI control widgets which may be assigned a value.

[00181] In various embodiments, the secondary selection establishment conditions may be finally satisfied using methods previously discussed, including but not limited to, a contact pressure velocity trigger.

[00182] Once the selection establishment conditions have been satisfied, the state of the UI control widget may be set. For example, a button may be toggled, a check box may be checked or unchecked, a slider value may be set, etc.

[00183] As a specific example of one embodiment, from a user perspective, a user may make contact with and exert pressure upon an interaction surface, with a contact point near a drop down menu. The user may increase the contact pressure until the drop down menu is highlighted. After maintaining that contact pressure level for a few seconds, the list of menu items becomes visible, after which the user may select an item by varying the contact pressure. Once the desired menu item is highlighted, the user may execute a rapid increase in pressure to finalize the selection of that menu item.

[00184] If it is determined in **604** that there is no user interface control widget within the context determination zone, it is then determined whether there is an embedded hyperlink within the context determination zone. See determination **608**.

[00185] Within the context of the present description, an embedded hyperlink refers to a hyperlink which is embedded within a text field. For example, in one embodiment, embedded hyperlinks may include, but are not limited to, text or image links within a web page, text or image links within a word processing document, and/or any other type of link which may be contained within a document. It should be noted that any link which may be classified as a hyperlink may fall within this definition of embedded hyperlink, or within the definition of a UI control widget.

[00186] In various embodiments, the determination as to whether there is an embedded hyperlink within the context determination zone begins with determining whether there is an embedded hyperlink located at the contact point upon which the zone is based.

[00187] If an embedded hyperlink is not located at the contact point, then it may be determined if there is an embedded hyperlink in the vicinity, within the context determination zone. In various embodiments, the methods described for locating a UI control widget within the context determination zone may also be applied to determining whether there is an embedded hyperlink in the zone. In one embodiment, the bounding boxes may be handled by the application displaying the document in which an embedded hyperlink may be found. In another embodiment, the bounding boxes may be handled at a lower level, by the operating system.

[00188] If it is determined in **608** that there is at least one embedded hyperlink within the context determination zone, then a context sensitive embedded hyperlink selection function is chosen. See operation **610**. In various embodiments, once an appropriate selection function is chosen, a selection may be defined based upon user input. In one embodiment, the selection may be defined using method **4**.

[00189] In various embodiments, a context sensitive embedded hyperlink selection function may include logic to use a different context sensitive selection function to define the user's selection, based upon user input. In this way, a user may be able to indicate that the initial context determination is not correct, and that another selection function should be used.

[00190] For example, in one embodiment, a context sensitive embedded hyperlink selection function may include logic such that, if the contact pressure level exceeds a predefined limit P_2, further definition of the selection is handled by a different context sensitive selection function, such as one adapted for selecting text. In another embodiment, a contact pressure level exceeding P_2 may cause the context determination zone to expand, and the process of choosing an appropriate selection function to resume, ignoring the embedded hyperlink previously detected within the zone. As an option, future determinations may ignore all embedded hyperlinks. In this way, a context-aware user selection initiated with a contact point in the vicinity of an embedded hyperlink may be redirected to a different type of subject matter without forcing the user to reposition their finger, for example.

[00191] In various embodiments, a context sensitive embedded hyperlink selection function may depend upon the context determination zone. For example, in one embodiment, a context sensitive embedded hyperlink selection function may use the context determination zone as a secondary boundary. In this way, the selection function may be able to continue searching the context determination zone for additional embedded hyperlinks.

[00192] In various embodiments, a context sensitive embedded hyperlink selection function may include logic similar to that used in a context sensitive UI control widget

selection function to handle the case where there is more that one embedded hyperlink within the context determination zone. Additionally, in various embodiments, a context sensitive embedded hyperlink selection function may include logic similar to that used in a context sensitive UI control widget selection function to allow a user to toggle between multiple embedded hyperlinks found within the context determination zone.

[00193] In various embodiments, a context sensitive embedded hyperlink selection function may visually indicate the selection of an embedded hyperlink before the selection has been established. For example, in one embodiment, the selection function may change the color of an embedded text hyperlink. In another embodiment, the selection function may change the font, style, or font size of an embedded text hyperlink. In yet another embodiment, the selection function may highlight the borders of a linked image. In this way, the user may see which embedded hyperlink is about to be activated before satisfying the selection establishment conditions.

[00194] In various embodiments, a context sensitive embedded hyperlink selection function may have selection establishment conditions which change, depending upon previous user input. For example, in one embodiment, if the contact pressure level surpasses a value of P_1, but not P_2, the selection establishment condition may change to require the selection of an option from a hyperlink contextual menu. The contextual menu may provide options common to web browsers, including but not limited to "open link", "copy link", and/or any other action which may be performed on or with a hyperlink. In various embodiments, the selection of the contextual menu item may be established using methods including, but not limited to, those described for multi-state UI control widgets, or other types of selections.

[00195] In various embodiments, once the selection establishment conditions have been satisfied, the selected embedded hyperlink may be activated.

[00196] If it is determined in **608** that there are no embedded hyperlinks within the context determination zone, it is then determined whether there is a selectable text field within the context determination zone. See determination **612**.

[00197] Within the context of the present description, a selectable text field refers to a field of text which a user may select and perform operations upon (e.g. copy, cut, paste, etc.). In one embodiment, a selectable text field may also include any rasterized text which is recognized using an optical character recognition routine.

[00198] In various embodiments, the determination as to whether there is a selectable text field within the context determination zone begins with determining whether there is a selectable text field located at the contact point upon which the zone is based.

[00199] If a selectable text field is not located at the contact point, then it may be determined if there is a selectable text field in the vicinity of the contact point and within the context determination zone. In various embodiments, the methods described for locating a UI control widget within the context determination zone may also be applied to determining whether there is a selectable text field in the zone, including the use of bounding boxes.

[00200] If it is determined in **612** that there is at least one selectable text field intersecting with or within the context determination zone, then a context sensitive selectable text selection function is chosen. See operation **614**. In various embodiments, once an appropriate selection function is chosen, a selection may be defined based upon user input. In one embodiment, the selection may be defined using method **4**.

[00201] In various embodiments, if there is a selectable text field located at the contact point, the context sensitive selectable text selection function may behave like other text selection functions. For example, in one embodiment, the selection of the text would start at the contact point, and expand with pressure in the direction in which the text would be read.

[00202] In various embodiments, if there is a selectable text field within the context determination zone, but not at the contact point, the selection function may define the selection based upon the entire selectable text field. For example, in one embodiment, the text within selectable text field found in the context determination zone may be selected starting at the beginning of the text field, even if it is not near the contact point.

In another embodiment, the selection of the text begins at the location closest to the contact point.

[00203] In various embodiments, a context sensitive selectable text selection function may depend upon the context determination zone. For example, in one embodiment, a context sensitive selectable text selection function may use the context determination zone as a secondary boundary. In this way, the selection function may be able to continue searching the context determination zone for additional selectable text fields.

[00204] In various embodiments, a context sensitive selectable text selection function may include logic to handle the case where there is more than one selectable text field within the context determination zone. In one embodiment, the selection function may choose the selectable text field which is closest to the contact point, the selection not extending beyond that text field. In another embodiment, the selection function may start the text selection in the selectable text field closest to the contact point; once all of the text in that closest field has been selected, the selection may continue to expand into the next closest text field, starting at the beginning of the field. In still another embodiment, the selection function may include logic similar to that used in a context sensitive UI control widget selection function to handle the case where there is more that one selectable text field within the context determination zone. Additionally, in yet another embodiment, a context sensitive selectable text selection function may include logic similar to that used in a context sensitive UI control widget selection function to allow a user to toggle between multiple selectable text fields found within the context determination zone.

[00205] In various embodiments, a context sensitive selectable text selection function may include logic to use a different context sensitive selection function to define the user's selection, based upon user input. In this way, a user may be able to indicate that the initial context determination is not correct, and that another selection function should be used.

[00206] For example, in one embodiment, a context sensitive selectable text selection function may include logic such that, if the contact pressure level exceeds a predefined

limit P_4, further definition of the selection is handled by a different context sensitive selection function, such as one adapted for selecting graphics. In another embodiment, a contact pressure level exceeding P_4 may cause the context determination zone to expand, and the process of choosing an appropriate selection function to resume, ignoring the selectable text field previously detected within the zone. As an option, future determinations may ignore all selectable text fields. In this way, a context-aware user selection initiated with a contact point in the vicinity of a selectable text field may be redirected to a different type of subject matter without forcing the user to reposition their finger, for example.

[00207] In various embodiments, a context sensitive selectable text selection function may visually indicate the selection of a selectable text field before the selection has been established. For example, in one embodiment, the selection function may shade the selected text within a selectable text field. In another embodiment, the selection function may highlight the borders of an entire selectable text field which has been selected.

[00208] In various embodiments, a context sensitive selectable text selection function may have selection establishment conditions which change, depending upon previous user input. For example, in one embodiment, if the contact pressure level surpasses a value of P_3, but not P_4, the selection establishment condition may change to require the selection of an option from a text selection contextual menu. The contextual menu may provide options including, but not limited to, copy, cut, paste, style, send via email, send via SMS, lookup online, and/or any other action which may be performed on selected text. In various embodiments, the selection of the contextual menu item may be established using methods including, but not limited to, those described for multi-state UI control widgets, or other types of selections.

[00209] If it is determined in **612** that there is no text within the context determination zone, it is then determined whether there is a selectable graphic within the context determination zone. See determination **616**.

[00210] Within the context of the present description, a selectable graphic refers to an image or display, or a portion of an image or display, which a user is able to select and

perform operations upon (e.g. copy, cut, paste, save, etc.). In one embodiment, a selectable graphic may refer to any portion of a user interface which is able to be captured with a screenshot.

[00211] In various embodiments, the determination as to whether there is a selectable graphic within the context determination zone begins with determining whether there is a selectable graphic located at the contact point upon which the zone is based.

[00212] If a selectable graphic is not located at the contact point, then it may be determined if there is a selectable graphic in the vicinity of the contact point and within the context determination zone. In various embodiments, the methods described for locating a UI control widget within the context determination zone may also be applied to determining whether there is a selectable text field in the zone, including the use of bounding boxes. As an option, in addition to determining the boundaries of the bounding boxes within the context determination zone, it may also be determined whether they are able to be selected (e.g. checking a system flag, DRM, copy protection, etc.).

[00213] If it is determined in **616** that there is at least one selectable graphic within the context determination zone, then a context sensitive selectable graphic selection function is chosen. See operation **618**. In various embodiments, once an appropriate selection function is chosen, a selection may be defined based upon user input. In one embodiment, the selection may be defined using method **4**.

[00214] In various embodiments, if there is a selectable graphic located at the contact point, the context sensitive selectable graphic selection function may behave like other graphic selection functions. For example, in one embodiment, the selection of the graphic may be defined using method **4**.

[00215] In various embodiments, if there is a selectable graphic within the context determination zone, but not at the contact point, the selection function may define the selection based upon the entire selectable graphic object. For example, in one embodiment, an image file embedded in a document near the contact point and at least partially within the context determination zone may be selected in its entirety. In another

embodiment, said image file may be partially selected, starting at a location within the image that is closest to the contact point, and scaling as a function of user input and in a manner defined by the selection function.

[00216] In various embodiments, a context sensitive selectable graphic selection function may depend upon the context determination zone. For example, in one embodiment, a context sensitive selectable graphic selection function may use the context determination zone as a secondary boundary. In this way, the selection function may be able to continue searching the context determination zone for additional selectable graphic objects.

[00217] In various embodiments, a context sensitive selectable graphics selection function may include logic to handle the case where there is more than one selectable graphic object within the context determination zone. In one embodiment, the selection function may choose the selectable graphic object which is closest to the contact point.

[00218] In another embodiment, the selection function may include logic similar to that used in a context sensitive UI control widget selection function to handle the case where there is more that one selectable graphic object within the context determination zone. Additionally, in yet another embodiment, a context sensitive selectable graphic selection function may include logic similar to that used in a context sensitive UI control widget selection function to allow a user to toggle between multiple selectable graphic objects found within the context determination zone.

[00219] In various embodiments, a context sensitive selectable graphics function may pass control to another user interface function, depending upon user input. In this way, a user may be able to indicate that the initial context determination is not correct, and that they wish to activate a different user interface routine (e.g. application launcher, favorites list, system settings, etc.).

[00220] For example, in one embodiment, a context sensitive selectable graphic selection function may include logic such that, if the contact pressure level exceeds a predefined limit P_6, further user input is handled by a different context sensitive selection

function, such as one adapted for selecting graphics. In another embodiment, a contact pressure level exceeding P_4 may cause the context determination zone to expand, and the process of choosing an appropriate selection function to resume, ignoring the selectable graphic previously detected within the zone. As an option, future determinations may ignore all selectable graphics. In this way, a context-aware user selection initiated with a contact point in the vicinity of a selectable graphic may be redirected to a different type of subject matter without forcing the user to reposition their finger, for example.

[00221] In various embodiments, a context sensitive selectable graphics selection function may visually indicate the selection of a selectable graphic object before the selection has been established. For example, in one embodiment, the selection function may shade the selected graphic object. In another embodiment, the selection function may highlight the borders of a graphic object which has been selected.

[00222] In various embodiments, a context sensitive selectable graphics selection function may have selection establishment conditions which change, depending upon previous user input. For example, in one embodiment, if the contact pressure level surpasses a value of P_5, but not P_6, the selection establishment condition may change to require the selection of an option from a graphic selection contextual menu. The contextual menu may provide options including, but not limited to, copy, cut, paste, set as wallpaper, send via email, send via SMS, and/or any other action which may be performed on selected graphics. In various embodiments, the selection of the contextual menu item may be established using methods including, but not limited to, those described for multi-state UI control widgets, or other types of selections.

[00223] If it is determined in **616** that there is no selectable graphics within the context determination zone, then the determination zone is expanded. See operation **620**.

[00224] In one embodiment, the context determination zone may be expanded by a fixed number of pixels. In another embodiment, the determination zone may be expanded by a scaling factor. Once the context determination zone has been expanded, the context determinations are performed again.

[00225] In one embodiment, if the context determination zone reaches a predefined threshold size, the user may be informed that a context could not be determined and a selection cannot be made. Other forms of user feedback indicating a failure to determine context include an alert, a prompt, audio feedback, visual feedback, LED light, and/or any other type of feedback.

[00226] In various embodiments, the user may be able to specify which contexts they wish to be detected automatically. In some embodiments, additional types of subject matter may be detectable, including types of text data, such as dates, email addresses, phone numbers, and/or any other type of text information.

[00227] In some embodiments, contextual selections may be performed using method **600** only in situations where a contact pressure based form of user interaction has not been defined. In other embodiments, contextual selections may be made system wide using method **600** or similar methods.

[00228] Figure **7** shows a device **700** having a backtouch interface, in accordance with one embodiment. As an option, the device **700** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the device **700** may be implemented out in any desired environment. It should also be noted that the aforementioned definitions may apply during the present description.

[00229] In the context of the present description, a device with a backtouch interface refers to a device with one or more backtouch sensors or backtouch switches. In one embodiment, a device with a backtouch interface may also include one or more other sensors and/or buttons, including but not limited to cameras, hardware buttons, and/or any other type of device or control. A device with a backtouch interface may also be referred to as a backtouch-enabled device.

[00230] Within the context of the present description, a backtouch sensor refers to an interaction surface located on a device which has reduced user visibility while the device is being operated, because it is obscured by the device itself. For example, in one

embodiment, a backtouch sensor may be located on the back of the device, facing away from the user as they look at a display on the front side. A backtouch sensor may result in a better user interface experience since the user is able to interact with the device without obscuring the front display with an implement of interaction.

[00231] Additional embodiments may be described using the terms 'front' and 'back', but a backtouch-enabled device is not limited to those having parallel surfaces. Within the context of the present description, the terms 'front' or 'front surface' and 'back' or 'back surface' refer to any two surfaces of any portion, piece, section, slice, component, part, and/or face of a device or its parts.

[00232] In various embodiments, a backtouch interface may also be a display. For example, in one embodiment, a device may have high-resolution touchscreens on the front and back. In another embodiment of a multi-display, backtouch-enabled device, the front face may be a high-resolution touchscreen and the back face may be a low-resolution touchscreen. As an option, one or more of the touch screens in these embodiments may be black and white. In still another embodiment, the front and back displays may utilized different display technologies and/or different display properties. For example, in one embodiment, the front display may be a high-resolution trans missive touchscreen and the back display may be a low-resolution reflective touchscreen.

[00233] As shown, device **700** is an example of one embodiment of a backtouch-enabled device, possessing a backtouch sensor **702** and a display **704**. In various embodiments, the display **704** may also be an interaction surface.

[00234] As previously defined, an interaction surface is a surface capable of sensing touch, pressure, proximity, or any combination thereof. In various embodiments, a backtouch sensor may be capable of sensing one or more of touch, pressure, or proximity.

[00235] For example, in one embodiment, a backtouch sensor may be a track pad. Within the context of the present description, a track pad is a touch sensitive pointing device which can detect the motion and position of an implement of interaction (e.g. user finger, stylus, etc.).

[00236] In various embodiments, a backtouch sensor may include a resistive touch sensor. In one embodiment, a resistive touch sensor is made up of two thin, conducting layers separated by a narrow gap. When an implement of interaction presses down on the outer layer, the two layers make contact at that point. A voltage may be applied across the upper layer, from top to bottom. The resistive touch sensor may act as a voltage divider; by measuring the output voltage on the lower layer, the vertical position of the touch position may be sensed. Swapping the voltage to the sides of the upper layer may allow the horizontal touch position to be sensed.

[00237] In various embodiments, a backtouch sensor may include a capacitive touch sensor. In one embodiment, a capacitive touch sensor is made of two layers of parallel conductive lines, separated by an insulator. The layers are arranged such that the conductive lines are perpendicular. When an implement of interaction is placed over the conducting lines, there is a change in capacitance. A high frequency signal may be applied to pairs of conducting lines; the resulting current that passes between the conducting lings is proportional to the capacitance between the conducting lines. This current may be used to sense the touch or proximity of an implement of interaction. A capacitive touch sensor may be capable of detecting more than one contact point or contact area.

[00238] The backtouch-enabled device shown in Figure **7** is one exemplary embodiment. The geometry of a backtouch-enabled device need not be confined to parallel surfaces, with a distinct front and back. For example, in one embodiment, the device may have fold-out sections with interaction surfaces which become backtouch interfaces once the sections are folded out. In another embodiment, the backtouch-enabled device may be composed of curved parts, multiple parts, or complex plan form parts. In still another embodiment, the backtouch-enabled device may not be rectilinear in any outline and/or cross section. In yet another embodiment, the device may have one or more curved or non-planar surfaces, including but not limited to surfaces which are concave or convex. In some embodiments, the device may have one or more surfaces which are rigid or hard. In other embodiments, the device may have one or more surfaces which are soft or flexible. In still other embodiments, the device may change shape,

depending on the mode of operation (e.g. slide-out hardware keyboard, plug-in modules, etc.).

[00239] In one embodiment, a backtouch-enabled device may have one or more backtouch sensors located on plane(s) parallel to the plane of a display. For example, see device **700**. In another embodiment, a backtouch-enabled device may have one or more backtouch sensors located on plane(s) that are at an angle to the plane of a display.

[00240] In one embodiment, a backtouch sensor may be located on a plane which is completely obscured from the display. In another embodiment, a backtouch sensor may be located on a plane which is partially obscured from the display. For example, a backtouch sensor may be located on the back side of a transparent or translucent display.

[00241] In one embodiment, a backtouch interface is an interaction surface which may be made up of one or more touch sensors, one or more pressure sensors, one or more proximity sensors, one or more switches, or any combination of one or more such sensors or switches. In another embodiment, the switches which are part of a backtouch interface may be real or virtual, displayed or mechanical, soft or hard, or any combination of these properties.

[00242] In various embodiments, gestures and interactions described in terms of a backtouch interface may also be performed on sidetouch (left and right sides of a device) and captouch (top and bottom sides of a device) interfaces.

[00243] Figure **8** shows a method **800** for providing feedback to the user of a backtouch interface, in accordance with one embodiment. As an option, the method **800** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the method **800** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00244] As shown, it is determined whether the backtouch interface is enabled. See determination **802**.

[00245] Within the context of the present description, an enabled interface or surface refers to an interface or surface which is not disabled. Within the context of the present description, a disabled interface or surface refers to an interface or surface which is not passing user input to any process, system, application, or routine. In one embodiment, a disabled interface may continue to sense user input, even though it is not used. In another embodiment, a disabled interface may be able to recognize a predefined form or type of user input which then places the interface in an enabled state. As a specific example, in one embodiment, a disabled backtouch interface may not be used to make a selection until it detects a predefined pattern of increases and decreases of contact pressure. In this way, interfaces may be disabled to avoid unwanted input, and easily enabled when desired.

[00246] It is important to note that in various embodiments, an interface or surface may be enabled with respect to one type of user input, and disabled with respect to another type of user input. For example, it may be possible for an interaction surface to be enabled with respect to touch interactions, and disabled with respect to contact pressure or proximity.

[00247] In various embodiments, a backtouch interface may be enabled in response to user input. For example, in one embodiment, a backtouch interface may be enabled in response to a predefined gesture performed on an interaction surface receptive to touch. In another embodiment, a backtouch interface may be enabled in response to a predefined pattern of increases and decreases of contact pressure exerted upon an interaction surface receptive to pressure. In yet another embodiment, a backtouch interface may be enabled by a predefined series of taps or impacts with the device. As an option, the taps or impacts may be detected by an accelerometer. In still another embodiment, a backtouch interface may be enabled by toggling a hardware or software control (e.g. button, switch, etc.).

[00248] In various embodiments, a backtouch interface may be enabled without specific user input. For example, in one embodiment, a backtouch interface may be enabled by a backtouch-enabled application. In another embodiment, a backtouch

interface may be enabled when the device orientation is within a predefined range. In yet another embodiment, a backtouch interface may always be enabled.

[00249] If it is determined that the backtouch interface is enabled, then it is determined whether there is an implement of interaction in proximity of the backtouch interface. See determination **804**.

[00250] In various embodiments, there may be a threshold proximity that may be achieved before determination **804** is satisfied. In one embodiment, a user may define the threshold proximity. In another embodiment, the threshold proximity may be predefined within the operating system. In still another embodiment, the threshold proximity may be predefined on a per-application basis.

[00251] If it is determined that an implement of interaction is in proximity to the backtouch interface, a visual indication of the proximity is displayed. See operation **806**.

[00252] In various embodiments, the location of the implement of interaction in proximity to the backtouch interface may be indicated to the user. For example, in some embodiments, a point associated with the location of the implement may be displayed. In one embodiment, the point may be the centroid of the area with the highest proximity value. As an option, the point may be represented as a circle, a crosshair, and/or any other shape or icon.

[00253] In one embodiment, the point associated with the location of the implement may be displayed using a predefined color. As an option, the user may be able to select the color used. In yet another embodiment, the point may be displayed by inverting the preexisting contents of the display located at that point.

[00254] In various embodiments, an area associated with the location of the implement may be displayed. For example, in one embodiment, the area may be the area with proximity values above a preset threshold value. In another embodiment, the area may be a circle centered upon the centroid of maximum proximity values. As an option, the size of the circle may be similar to a fingertip.

[00255] In one embodiment, the area associated with the location of the implement may be displayed with colored shading. As an option, the user may preset the color of the shading. In another embodiment, the area may be indicated by increasing the color brightness (e.g. moving towards white within a color space, etc.) of the display content within the area. In yet another embodiment, the area may be displayed by reducing the color brightness (e.g. moving towards black within a color space, etc.) of the display content outside the area. As an option, the area may be displayed with a sharp boundary, or with a softened boundary, creating a glow effect.

[00256] In various embodiments, the degree of proximity of the implement of interaction to the backtouch interface may be indicated to the user. For example, in one embodiment, the color of a point or area being displayed may be given a degree of transparency dependent upon the proximity values, allowing the original display content to be viewed. As an option, the proximity value may be averaged over the area, and a single transparency value given to the color applied. In another embodiment, the color of the content displayed within the area may be given a color brightness based upon the proximity, such that the area is white right before contact is made. In yet another embodiment, the color of the content displayed outside the area may be given a color brightness based upon an inverse relationship with the proximity, such that all but the area is black right before contact is made.

[00257] As shown, it is determined if an implement of interaction is in contact with the backtouch interface. See determination **808**. If it is determined that there is no contact, the proximity may continue to be determined and displayed.

[00258] If it is determined that an implement of interaction is in contact with the backtouch interface, a visual indication of the contact is displayed. See operation **810**.

[00259] In various embodiments, a visual indication of contact with a backtouch interface may differ from a visual indication of contact with a front display, where the contact point and area is usually obscured by an implement of interaction. For example, in one embodiment, the contact point may be displayed, instead of or in addition to the contact area. In another embodiment, the visual representation of the interior of the

contact area may be colored. In yet another embodiment, the contact area may be represented by an iconic version of a fingerprint. In still another embodiment, the contact area may be distorted so as to appear to bulge out of the display. As an option, this distortion may increase as a function of contact pressure.

[00260] As shown, once the user has been given feedback regarding contact with the backtouch interface, further user interaction is handled by a different routine. See operation **8**. For example, methods **4** or **6** may be implemented at this point, utilizing backtouch contact pressure and contact point to define a selection.

[00261] Figure **9** shows a pressure-sensitive user interface **900** for making a selection using a backtouch interface, in accordance with one embodiment. As an option, the user interface **900** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **900** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00262] As shown, the user interface **900** may be utilized in making a selection based in part on pressure exerted by the user upon a backtouch interface. For example, in one embodiment, a user's finger **902** may exert a small amount of force over contact area **904**, which is centered on contact point **906**. In one embodiment, contact area **904** and contact point **906** may be displayed on top of the content already being displayed, to assist the user in making contact at the desired location. In another embodiment, the contact point **906** may be displayed as a symbol, such as a plus sign, which may be rotated to an orientation estimated to match that of the contacting finger. As an option, the finger orientation may be estimated using the size and shape of the contact area, detected location of other fingers, and/or any other user input or observable properties.

[00263] If the contact pressure level is sufficient, then a selection area **908** having a selection boundary **910** and a secondary boundary **912** (indicating uncertainty related to the contact area) is calculated and displayed. As an option, the selection may be defined using method **4** or **6**, or any other method of defining a selection based upon contact pressure.

[00264] Figure **10** shows a user interface **1000** for defining settings associated with a backtouch interface, in accordance with one embodiment. As an option, the user interface **1000** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **1000** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00265] In one embodiment, the user interface **1000** may include a plurality of check boxes **1002** which represent various types of user interactions detectable by the backtouch interface. The types of interactions may include, but are not limited to, pressure, touch, proximity, and/or any other type of user interaction. Through these checkboxes, a user may limit a backtouch interface to only certain types of interactions. As an option, the user interface **1000** may only present checkboxes for interaction types detectable by the particular backtouch interface associated with the settings being defined.

[00266] In one embodiment, the user interface **1000** may include a plurality of check boxes **1004** which represent various methods of enabling a backtouch interface. As an option, in one embodiment, more than one method of activation may be selected.

[00267] In one embodiment, the collection of backtouch interface enablement methods **1004** may include a checkbox **1006** which allows the interface to be activated by a gesture. In one embodiment, the gesture may be predefined within an application or operating system. In another embodiment, the user interface **1000** may include a button **1008** to allow a user to specify a gesture to activate the backtouch interface. As an option, the user may be presented with an interface where they may specify the gesture by performing it. In yet another embodiment, the user may choose from a plurality of predefined gestures.

[00268] In one embodiment, the collection of backtouch interface enablement methods **1004** may include a checkbox **1010** which allows the interface to be activated by a grip. In the context of the present description, a grip refers to the set of sensor data associated with a particular manner of holding a device. The sensor data associated with a grip may

include, but is not limited to, points and areas of contact on one or more interaction surfaces, device orientation, contact pressure levels exerted in one or more contact areas, proximity data (e.g. proximity to portions of a users hand not in contact with the device, etc.), and/or any other sensor data.

[00269] In one embodiment, the backtouch interface-enabling grip may be predefined within an application or operating system. In another embodiment, the user interface **1000** may include a button **1012** to allow a user to specify a grip to enable the backtouch interface. As an option, the user may be presented with an interface where they may specify the grip by performing it. In yet another embodiment, the user may choose from a plurality of predefined grips.

[00270] In one embodiment, the collection of backtouch interface enablement methods **1004** may include a checkbox **1014** which keeps the backtouch interface enabled whenever the device is powered on, similar to a primary touch screen, for example. In another embodiment, the collection of backtouch interface enablement methods **1004** may include a checkbox **1016** which keeps the backtouch interface disabled.

[00271] In one embodiment, the user interface **1000** may include a check box **1018** which allows the criteria for backtouch interface enablement to be set on a per-application basis. As an option, check box **1018** may allow any application to override the preferences set with the plurality of check boxes **1004**.

[00272] In one embodiment, the user interface **1000** may include a plurality of check boxes **1020** which represent various methods of indicating that a backtouch interface is enabled. As an option, in one embodiment, more than one indicator may be selected.

[00273] In one embodiment, the collection of indicator check boxes **1020** may include a checkbox **1022** which indicates that the backtouch interface is enabled by displaying a colored border around the edge of a display. In one embodiment, the colored border may be animated.

[00274] In one embodiment, the collection of indicator check boxes **1020** may include a checkbox **1024** which indicates that the backtouch interface is enabled using an LED.

In one embodiment, the LED may indicate the enabled backtouch interface through a specific color, with other colors being reserved for indicating other device states. In another embodiment, the LED may pulse or flash when the backtouch interface is enabled.

[00275] In one embodiment, the collection of indicator check boxes **1020** may include a checkbox **1026** which indicates that the backtouch interface is enabled by displaying a colored border around a status bar (e.g. a minimal bar common to most user interfaces, etc.). In one embodiment, the colored border may be animated.

[00276] In one embodiment, the collection of indicator check boxes **1020** may include a checkbox **1028** which indicates that the backtouch interface is enabled by displaying an icon. In one embodiment, the icon may be located within a status bar.

[00277] In one embodiment, the collection of indicator check boxes **1020** may include a checkbox **1030** which indicates that the backtouch interface is enabled by changing the color of a status bar. In one embodiment, the color change may be animated.

[00278] In one embodiment, the collection of indicator check boxes **1020** may include a checkbox **1032** which allows the backtouch interface to be enabled without any indication to the user. As an option, this preference may be overridden by an application.

[00279] In various embodiments, the user interface **1000** may allow a user to configure the type and style of feedback provided while an enabled backtouch interface is being used. For example, in one embodiment, the user interface **70.0002.3-00** may include a button **1034** which allows a user to specify whether or not feedback is provided while the backtouch interface is being used. As an option, this preference may be overridden by applications (e.g. an application may be configured such that it provides backtouch interface feedback independent of any system preferences, etc.).

[00280] In one embodiment, the user interface **1000** may include a button **1036**, which presents a user interface which allows a user to specify the style of feedback provided when a backtouch interface is being used. For example, a user may configure the

feedback provided when an implement of interaction is in proximity to, contact with, and/or exerting pressure on the backtouch interface.

[00281] In one embodiment, the user interface **1000** may include a button **1038**, which allows a user to test the style parameters they defined using button **1036**. For example, in one embodiment, button **1038** may present a user interface where a user may experiment with a backtouch interface, observing the feedback styles they have configured. As an option, this interface may provide a quick way to reconfigure the feedback style.

[00282] In various embodiments, the user interface **1000** may allow a user to configure the sensitivity of a backtouch interface. In the context of the present description, the sensitivity of a backtouch interface refers to the threshold level of interaction (e.g. proximity, contact, pressure, etc.) which a backtouch interface may receive before treating the interaction as being intentionally made by a user. In this way, backtouch interactions incidental to device usage may be ignored, while intentional backtouch user interactions may be acted upon.

[00283] In one embodiment, the user interface **70.0002.3-00** may include a plurality of checkboxes **1040** that represent a plurality of backtouch interface sensitivity levels. The backtouch interface sensitivity levels may include, but are not limited to, predefined levels (e.g. “high”, “standard”, “low”, etc.), user defined levels, automatically defined levels, and/or any other type of sensitivity level.

[00284] In one embodiment, the user interface **1000** may allow a user to specify the sensitivity levels for particular aspects of a backtouch interface (e.g. proximity, contact, pressure, etc.). As an option, the user interface may provide the user with the ability to save and load user defined backtouch interface sensitivity profiles.

[00285] In the context of the present description, an interface sensitivity profile refers to a collection of sensitivity parameters associated with various aspects of an interaction interface. The sensitivity parameters may include, but are not limited to, an activation threshold, a maximum signal beyond which interaction is ignored, sample rate(s), and/or any other parameter which may be associated with an interaction interface. As an option,

the sensitivity parameters may be specific to particular types of interaction (e.g. proximity, contact, pressure, etc.).

[00286] In the context of the present description, an activation threshold refers to a threshold signal below which interaction is ignored. An activation threshold may be set for different types of user interaction (e.g. pressure, contact, proximity, etc.). For example, in one embodiment, an interaction surface may have a pressure activation threshold of P, below which all pressure interactions are ignored.

[00287] It is important to recognize the distinction between activation and enablement. In various embodiments, activation serves as a filter of incidental interactions, defining the weakest sensor signal which may not be ignored. Enablement serves a similar purpose, to ignore unwanted interactions; a disabled interface may not act upon any sensor signals except those related to enabling the interface. In other words, activation filters sensor signals based on strength, while enablement filters sensor signals based on the signal itself.

[00288] In one embodiment, the plurality of checkboxes **1040** may include an "automatic" checkbox, representing a backtouch interface sensitivity profile automatically determined by the device. As an option, if an automatic sensitivity profile has not been created, the user may be presented with the option of initiating the process of automatically determining an optimal backtouch interface sensitivity profile.

[00289] In one embodiment, the user interface **1000** may include a button **1042** which allows a user to initiate the process of automatically determining an optimal backtouch interface sensitivity profile. In the context of the present description, an optimal interface sensitivity profile refers to a sensitivity profile which would allow an interface to remain enabled all the time, with minimal accidental inputs. In one embodiment, the process of automatically determining an optimal backtouch interface sensitivity profile may include gathering data over a predefined amount of time. As an option, the time remaining may be displayed in user interface **1000**. During this learning period, the backtouch enablement preferences specified in checkboxes **1004** may be overridden, and instead the user is required to enable the backtouch interface with a specific method, chosen such

that accidental enablement is highly unlikely (e.g. two step activation using gestures, hardware switch with on-screen confirmation, etc.). This facilitates separating intentional interactions from incidental interactions. Once enabled, the backtouch interface may remain enabled for a short period of time. During the learning period, the device may gather data, including but not limited to sensor data from the backtouch interface, changes in orientation, backtouch interface enablements, and/or any other data related to use of the device. Once the learning period has elapsed, the gathered data may be used to estimate the levels of meaningless, incidental backtouch interaction, such as interaction due to holding the device. As an option, in another embodiment, the user may be able to specify the length of the learning period. In yet another embodiment, the user may be able to specify desired accuracy of the sensitivity level, balancing the reduction of unwanted interactions with a possible increase in intentional interactions being missed.

[00290] In one embodiment, the user interface **1000** may include a button **1044** which allows a user to test the specified sensitivity settings. For example, in one embodiment, button **1044** may present a user interface which provides feedback regarding interactions with the backtouch interface. As an option, the testing user interface may utilize a default set of backtouch feedback settings, if the user had previously turned feedback off. In another embodiment, button **1044** may activate a backtouch interface, and present feedback to the user within user interface **1000**. As an option, user interface **1000** may ignore any input received from the backtouch interface, only provide feedback related to the user interactions. In this way, a user may test the settings with the backtouch interface without accidentally interacting with any checkboxes, buttons, or other element of user interface **1000**.

[00291] In various embodiments, the interfaces of Figure **10**, as well as any other previous or subsequent interfaces, may take the form of webpages displayed utilizing a web browser on any desired computer, handheld device, etc. In such case, any of the parameters or other input disclosed herein may be entered without the use of the host device, whereby such parameters or other input (or derivatives thereof) may be pushed to the device for configuration/updating purposes.

[00292] Figure **11** shows a user interface **1100** for defining settings associated with a pressure-sensitive interface, in accordance with one embodiment. As an option, the user interface **1100** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **1100** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00293] In one embodiment, the user interface **1100** may include a checkbox **1102** to allow the user to enable the pressure-receptive aspect of one or more interaction surfaces. In one embodiment, this checkbox may enable all pressure-receptive surfaces. In another embodiment, the user may be able to specify which pressure-receptive surfaces shall be enabled. In various other embodiments, enablement of pressure-receptive surfaces may be accomplished through user input outside of user interface **1100**. Pressure-receptive surface enabling user input may include, but is not limited to, gestures, software buttons, hardware buttons, accelerometer data, contact pressure patterns, and/or any other form of user input.

[00294] In one embodiment, the user interface **1100** may include a plurality of check boxes **1104** which allow a user to specify which types of feedback to provide when the pressure-sensitive interface(s) is activated, if any at all. For example, in one embodiment, the user may specify a sound which plays whenever the pressure-sensitive interface is activated. In another embodiment, the user may specify a form of visual feedback to be given when the pressure-sensitive interface is activated. As an option, the user may select from a plurality of predefined forms of feedback.

[00295] In one embodiment, the user interface **1100** may include a checkbox **1106** which allows a user to specify whether they want visual feedback regarding contact pressure level. As previously described, contact pressure level may be visually represented in a number of ways, including, but not limited to, a color which is part of a two color spectrum, a percentage, a unitless number, and/or any other representation of a value located within a finite range. In one embodiment, user interface **1100** may allow a

user to specify the type of contact pressure level feedback. As an option, a separate user interface may be presented to the user to define the form of the feedback.

[00296] In various embodiments, the user interface **1100** may be used to define the pressure activation threshold. In the context of the present description, a pressure activation threshold is the smallest contact pressure level which may not be ignored. In one embodiment, the current pressure activation threshold level **1108** may be displayed within user interface **1100**. The current pressure activation threshold level may be displayed as a unitless number, a percentage of the maximum measurable contact pressure level, a color, a scale, and/or any other method of representing a value located within a finite range.

[00297] In one embodiment, the user interface **1100** may include a button **1110** which allows a user to manually specify the pressure activation threshold level. In one embodiment, the user may be presented with an interface where they may enter a specific value to be used as the pressure activation threshold. In another embodiment, the user may be prompted to exert the desired threshold pressure on a pressure-sensitive interface, confirming their selection by maintaining the desired contact pressure for a predefined amount of time. As an option, the user may be prompted to repeat this process more than one time, after which the resulting pressure levels may be averages. In yet another embodiment, the user may be given the chance to test the newly specified pressure activation threshold level.

[00298] In one embodiment, the user interface **1100** may include a button **1112** which resets the pressure activation threshold level to a predefined, default value. As an option, the user may be prompted to confirm this action before the default value is applied.

[00299] In one embodiment, the user interface **1100** may include a button **1114** which initiates a process which determines an optimal pressure activation threshold level. In the context of the present description, an optimal pressure activation threshold level refers to an activation threshold which would allow an interface to remain enabled all the time, with minimal erroneous inputs. In one embodiment, the process of automatically determining an optimal pressure activation threshold level may include gathering data

over a predefined amount of time. As an option, the time remaining may be displayed in user interface **1100**.

[00300] During this learning period, the pressure activation threshold may temporarily be set to a level sufficiently high that an accidental activation is highly unlikely. This facilitates separating intentional interactions from incidental interactions. During the learning period, the device may gather data, including but not limited to sensor data from the pressure-sensitive interface, changes in orientation, pressure-sensitive interface activations, and/or any other data related to use of the device. Once the learning period has elapsed, the gathered data may be used to estimate the levels of meaningless, incidental pressure interaction, such as interaction due to holding the device. As an option, in another embodiment, the user may be able to specify the length of the learning period. In yet another embodiment, the user may be able to specify desired accuracy of the pressure activation threshold level optimization, balancing the reduction of unwanted interactions with a possible increase in intentional interactions being missed.

[00301] In one embodiment, the user interface **1100** may display the amount of time remaining in an ongoing learning period. See, for example, text field **1116**. In various embodiments, the time remaining in the learning period may be displayed in user interfaces other than **1100**. User interfaces where the remaining time may be displayed include, but are not limited to, a status bar, a contextual menu, a shortcut menu, a pop-up interface adapted for managing various interaction interfaces, and/or any other user interface.

[00302] In one embodiment, the user interface **1100** may include a plurality of checkboxes **1118** which represent a plurality of forms of feedback associated with the automatic determination of an optimal pressure activation threshold level. During the learning period, it may be helpful to provide the user with feedback every time the pressure-sensitive interface is activated. The forms of feedback include, but are not limited to, a sound, vibration, screen flash, and/or any other form of feedback. In one embodiment, the feedback associated with the automatic determination of an optimal

pressure activation threshold level may override preferences set elsewhere. In another embodiment, the user may specify the parameters of these forms of feedback.

[00303] The user interface **1100** may be used to define and manage touch states. In one embodiment, a preset number of global touch states may be defined, which the operating system and applications may make use of. In another embodiment, the touch states may be defined on a per-application basis.

[00304] In one embodiment, the user interface **1100** may include a button **1120** which presents the user with a user interface that facilitates the definition of touch states. As an option, the user may be able to select from a plurality of predefined touch state definitions. Additionally, the user interface **1100** may display the currently defined touch states. See, for example, text field **1124**. The touch states may be represented as a unitless number, a percentage of the range of allowable contact pressure levels, a color, and/or any other representation of a contact pressure level.

[00305] In one embodiment, the user interface **1100** may include a button **1122** which allows the user to reset the touch state definitions to a predefined default definition. As an option, the user may be prompted for confirmation before resetting the touch state definition to default values.

[00306] Figure **12** shows a method **1200** for assisting a user in defining touch states, in accordance with one embodiment. As an option, the method **1200** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the method **1200** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00307] As shown, the highest touch state is defined. See operation **1202**. In one embodiment, the highest touch state may be predefined within the operating system. In another embodiment, the highest touch state may be specified by the user, in the form of a contact pressure value.

[00308] In various embodiments, the highest touch state may be defined through user interaction with a pressure-sensitive interface. For example, in one embodiment, a user may be prompted to exert a level of contact pressure beyond which the highest touch state may be triggered. As an option, the user may be prompted to exert this pressure multiple times, with the resulting values averaged. In another embodiment, the user may be prompted to exert the highest contact pressure level which is comfortable, and then defining the highest touch state using a contact pressure level slightly lower than that being exerted.

[00309] Once the highest touch state has been defined, the remaining spectrum of contact pressures is partitioned into the remaining touch states and displayed. See operation **1204**.

[00310] In the context of the present description, the spectrum of contact pressures refers to the range of contact pressure levels a user may exert to interact with the device. For example, in one embodiment, the lower end of a spectrum of contact pressures may be equivalent to the pressure activation threshold level. In various embodiments, there may be multiple contact pressure spectrums defined. For example, in one embodiment, there may exist a distinct spectrum of contact pressures for each pressure-sensitive interaction surface. In another embodiment, there may exist a spectrum of contact pressures which is partitioned into touch states (a touch state spectrum), and a different spectrum of contact pressures for use with contact pressure levels (a contact pressure level spectrum).

[00311] In one embodiment, the touch state spectrum may have extrema defined by the user. As an option, the lower bound of the touch state spectrum may be the same as pressure activation threshold level.

[00312] In one embodiment, the touch state spectrum may be partitioned such that all touch states contain the same fraction of the touch state spectrum. In another embodiment, the touch state spectrum may be partitioned such that all touch states have equal portions of the touch state spectrum except for the highest touch state. In yet another embodiment, the touch state spectrum may be partitioned such that the touch

states are weighted, some containing larger fractions of the spectrum than others. As an option, the weight factors may be predefined or may be user definable.

[00313] In one embodiment, the partitioned touch state spectrum may be displayed as a geometric shape (e.g. circle, rectangle, etc.) which has been segmented according to the touch state partitions. In another embodiment, the partitioned touch state spectrum may be displayed as an array of percentages of the total spectrum. In yet another embodiment, the touch state spectrum may be displayed as an array of contact pressure levels associated with the boundaries of the partitions.

[00314] Once the touch state spectrum has been partitioned and displayed, the user is given the opportunity to adjust the touch states. See operation **1206**.

[00315] In one embodiment, the user may adjust the contact pressure levels associated with the touch states by specifying the numerical values of specific contact pressure levels. In another embodiment, the user may adjust the touch states by interacting with a graphical representation of the partitioned touch state spectrum. In yet another embodiment, the user may adjust the touch states by defining new partitions using a pressure-sensitive interface. As an option, the user may exert the desired contact pressure level to set the boundaries of a touch state.

[00316] Once the touch states have been adjusted, the newly partitioned touch state spectrum is verified. See operation **1208**.

[00317] In one embodiment, the touch state spectrum may be verified by asking the user to confirm their choice through a user interface. In another embodiment, the user may verify the touch state spectrum by utilizing it in a testing interface. As an option, the user may be given visual feedback regarding the exact contact pressure level being exerted, in addition to the current touch state.

[00318] Figure **13** shows a user interface **1300** for assisting a user in defining touch states, in accordance with one embodiment. As an option, the user interface **1300** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **1300** may be

implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00319] In one embodiment, the user interface **1300** may include a graphical representation **1302** of a touch state spectrum, hereinafter referred to as touch space spectrum rendering **1302**. As another embodiment, the touch space spectrum rendering **1302** may be colored with the spectrum of colors between two colors representing spectrum extrema. As an option, the same colors used to provide pressure dependent feedback to the user of a pressure-sensitive interface may be used here.

[00320] The touch space spectrum rendering **1302** found in user interface **1300** may be broken up into ranges of contact pressure representing touch states. In various embodiments, the touch space spectrum rendering may be partitioned by one or more touch state dividers **1304**. In one embodiment, the touch state dividers may simply be a line drawn between two touch states. In another embodiment, the touch state dividers may also display the contact pressure level separating those two touch states.

[00321] In various embodiments, a plurality of touch states may be defined. In one embodiment, the user may specify the number of touch states to define. In another embodiment, the number of touch states may defined within the operating system. In still another embodiment, the number of touch states may be defined on a per-application basis, by the applications themselves or by the user (if so allowed by the application). In yet another embodiment, there may exist a set of default touch states of a fixed number which may be overridden by a different set of touch states defined within an application.

[00322] In various embodiments, the touch space spectrum rendering **1302** found in user interface **1300** may have a labeled scale. For example, in one embodiment, the lower boundary **1306** of the lowest touch state (e.g. the touch state associated with the lowest contact pressures) may be displayed. As an option, the lower boundary of the touch space spectrum may be equivalent to the pressure activation threshold, which the user may define using, for example, user interface **1100**.

[00323] In another embodiment, the upper boundary **1308** of the touch space spectrum rendering may be displayed. As an option, upper boundary **1308** may be automatically set to a value slightly higher than the lower boundary of the highest touch state, since the highest touch state is made up of all pressures above that lower boundary. In this way, the scale of the touch space spectrum rendering is not skewed due to the potentially large range of contact pressures making up the highest touch state. In yet another embodiment, the touch space spectrum rendering may change in scale due to user interaction, with upper boundary **1308** being automatically updated.

[00324] In various embodiments, the touch state dividers **1304** found in the user interface **1300** may possess touch state divider handles **1310** with which a user may adjust the partitioning of touch states. For example, in one embodiment, a user may touch and drag a touch state divider handle to change the value of the associated divider. In another embodiment, a user may select a touch state divider handle; after the touch state divider handle has been selected, the divider may be assigned a value equivalent to whatever contact pressure level the user exerts for a predefined amount of time. As a specific example, a user may select a touch state divider handle and then press on an interaction surface. Once the desired contact pressure level is reached, the user maintains the pressure for 3 seconds, after which the value of the touch state divider is updated.

[00325] In one embodiment, the user interface **1300** may include a contact pressure gauge **1312** which reports the contact pressure level currently being exerted, to assist the user in selecting a practical partitioning of the touch state spectrum. In another embodiment, the scale of the contact pressure gauge may match that of touch state spectrum rendering **1302**. As an option, the scale of both contact pressure gauge **1312** and touch state spectrum rendering **1302** may be set to accommodate the larger of two values: the current contact pressure level, or a value slightly higher than the lower boundary of the highest touch state.

[00326] In various embodiments, the touch state dividers **1304** found in the user interface **1300** may each be associated with a plurality of icon buttons **1314** which a user may use to lock a touch state divider at a particular value. In one embodiment, adjusting

one touch state divider may cause all other unlocked dividers to shift such that the remaining portions of the touch state spectrum are partitioned as evenly as possible. As an option, the user interface **1300** may include a button to unlock all touch state dividers. In another embodiment, the user interface **1300** may include a button **1316** which distributes all unlocked touch state dividers evenly throughout the remaining parts of the touch state spectrum.

[00327] In one embodiment, the user interface **1300** may include a button **1318** which allows the user to establish the highest touch state boundary through pressure-based user input. In another embodiment, the user may be prompted to establish this touch state boundary as soon as user interface **1300** is displayed, in accordance with method **1200**. In one embodiment, the user may only specify the highest touch state, with the remaining touch state spectrum being evenly partitioned across the touch states.

[00328] In one embodiment, the user interface **1300** may include a button **1320** which resets all touch state boundaries to default values. In another embodiment, the highest touch state boundary may be preserved, only able to be reset within the interface used to define it that may be displayed in response to activating button **1318**.

[00329] In various embodiments, the user interface **1300** may include a button **1322** which allows a user to test the currently defined touch states. For example, in one embodiment, a user may be asked to achieve various touch states, and maintain them for a certain period of time. In another embodiment, the user may be presented with an interface similar to **1300**, but where the touch state spectrum rendering **1302** and the contact pressure gauge **1312** are more predominantly displayed. As an option, a sound may be played every time a touch state boundary is crossed.

[00330] Figure **14** shows a user interface **1400** for indicating that a backtouch and/or pressure-sensitive interface is activated, in accordance with one embodiment. As an option, the user interface **1400** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **1400** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00331] In one embodiment, the user interface **1400** may include one or more icons **1402** to indicate that a backtouch and/or pressure-sensitive interface has been activated. As an option, the icons may also indicate whether an interface is enabled. For example, an icon may only be visible if the interface is enabled, and only in color if the interface is activated. In another embodiment, the icons may have a color which depends upon the contact pressure level currently being exerted on that particular interface.

[00332] In various embodiments, the user interface **1400** may include a status bar **1404** which provides the user feedback regarding various device functions, in addition to other information, such as the date and/or time. The status bar itself may be used to indicate whether a backtouch and/or pressure-sensitive interface has been activated. For example, in one embodiment, the status bar may have a shading color which depends upon the contact pressure level currently being exerted on an activated interface. This may be done in conjunction with displaying an icon **1402** to indicate which interface has been activated. In another embodiment, the change in color may be restricted to just the border **1406** of the status bar. In yet another embodiment, the type of interface which has been activated may be indicated through an animation routine, including, but not limited to, a pulsing of the border, a cyclical variation of transparency of the status bar shading, "marching ants" along the status bar border, and/or any other animation routine.

[00333] In one embodiment, the status bar **14**04 within the user interface **1400** may include one or more items of textual information **1408**. The items of textual information may include, but are not limited to, date, time, network type, battery level, other device or application status information, and/or any other type of textual information.

[00334] In one embodiment, the activation of a backtouch and/or pressure-sensitive interface may be indicated through the presentation properties of the textual information **1408**. Presentation properties for textual information which may be altered to indicate the activation of an interface include, but are not limited to, font, style, size, color, animation routine (e.g. flashing text, cycling colors, etc.), and/or any other type of presentation property. In another embodiment, activation of a backtouch and/or pressure-sensitive interface may be indicated by temporarily replacing textual information **1408**

with a message, which may include, but is not limited to, the identity of the activated interface.

[00335] In one embodiment, the user interface **1400** may indicate the activation of a backtouch and/or pressure-sensitive interface by displaying a border **1410** around the display. As an option, border **1410** may change color depending upon the current contact pressure level.

[00336] Figure **15** shows a user interface **1500** for defining a backtouch feedback style, in accordance with one embodiment. As an option, the user interface **1500** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **1500** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00337] When interacting with a backtouch interface, it may or may not be important to provide the user with some form of feedback regarding the location and/or activity of their chosen implement of interaction. Providing interaction feedback allows the user to interact with the device using an interaction surface they may not be able to see.

[00338] In one embodiment, the user interface **1500** may include a plurality of checkboxes **1502** which represent various types of feedback which may be provided as an implement of interaction is in proximity to the backtouch interface. The types of feedback may include, but are not limited to, visual (e.g. screen flashes, status bar icons, iconic depictions of the location of the implement, etc.), sound, vibration, and/or any other form of feedback. In another embodiment, one or more of these types of feedback may be configured by the user. For example, in one embodiment, the user may select a sound to be played whenever an implement of interaction is in proximity to the backtouch interface. In yet another embodiment, the feedback may be limited to only those instances where the proximity to the backtouch interface appears intentional, rather than incidental. As a specific example, in one embodiment, feedback might be given for a fingertip, but not for a palm which is in proximity to the backtouch interface due to the user's grip.

[00339] In one embodiment, the user interface **1500** may include a plurality of checkboxes **1504** which represent various types of visual feedback that may indicate that an implement of interaction is in proximity to a backtouch interface. The types of visual feedback include, but are not limited to, graphical representation of probable implement location, status bar icon, screen flash, and/or any other type of visual feedback.

[00340] In various embodiments, a backtouch interface may be able to estimate the location of an implement of interaction in proximity to the backtouch interaction surface. This information can be used to provide useful feedback to the user. For example, in one embodiment, a point associated with the location may be displayed. As an option, the point may be the centroid of the area with the highest proximity value within a predefine threshold. In another embodiment, the point may be replaced or combined with an icon, shape, or symbol.

[00341] In another embodiment, the user may be provided with proximity feedback in the form of a visual representation of the proximity data received from the backtouch interface. As an option, the proximity values may be scaled such that the visual representation has the appearance of a glow, centered on the estimated location of the implement.

[00342] In one embodiment, the user interface **1500** may include a plurality of checkboxes **1506** which represent various styles of visual proximity feedback. The styles include, but are not limited to, dynamic coloring, the feedback described in conjunction with operation **806** of method **800**, and/or any other type or style of visual feedback. In the context of the present description, dynamic coloring refers to coloring which varies as a function of proximity to the backtouch interface. For example, in one embodiment, the color of the visual proximity feedback may vary between two colors which represent the least discernable proximity and the greatest proximity before contact. As an option, the colors may include a transparency factor, so the visual feedback does not obstruct display contents any more than necessary.

[00343] In one embodiment, the user interface **1500** may include a plurality of checkboxes **1508** which represent various types of feedback which may be provided as an

implement of interaction comes in contact with the backtouch interface. The types of feedback may include, but are not limited to, visual (e.g. screen flashes, status bar icons, iconic depictions of the location of the implement, etc.), sound, vibration, and/or any other form of feedback. In another embodiment, one or more of these types of feedback may be configured by the user. For example, in one embodiment, the user may select a sound to be played whenever an implement of interaction makes contact with the backtouch interface. In yet another embodiment, the feedback may be limited to only those instances where the contact with the backtouch interface is estimated to be intentional, rather than incidental. As a specific example, in one embodiment, feedback might be given for contact with a fingertip, but not for a palm which is in contact with the backtouch interface due to the user's grip.

[00344] In one embodiment, the user interface **1500** may include a plurality of checkboxes **1510** which represent various types of visual feedback indicating contact with the backtouch interface. The types of visual feedback include, but are not limited to, graphical representations of contact location and extent (e.g. contact point, contact area, etc.), a status bar icon, screen flash, and/or any other type of visual feedback, including those described in other embodiments.

[00345] In one embodiment, the user interface **1500** may include a plurality of checkboxes **1512** which cause the visual feedback indicating contact with the backtouch interface to fade after a set amount of time. In another embodiment, the user may specify the amount of time before fading begins, how quickly the fade occurs, and/or the limits of the fade (e.g. completely disappear, fade to 70% transparency, etc.). As an option, the fading of the visual feedback may be reversed if the implement of interaction moves beyond a preset threshold while still in contact with the backtouch interface.

[00346] In one embodiment, the user interface **1500** may include a plurality of checkboxes **1514** which allow the user to specify the style of visual feedback associated with making contact with the backtouch interface. The styles may include, but are not limited to, shading of a contact area, line style of the border of a contact area, animation

routine, and/or any other style aspect of visual feedback, including those described in other embodiments.

[00347] In one embodiment, the user interface **1500** may include a plurality of checkboxes **1516** which represent various types of feedback which may be provided as an implement of interaction exerts pressure on the backtouch interface. The types of feedback may include, but are not limited to, visual (e.g. screen flashes, status bar icons, iconic depictions of the contact pressure of the implement, etc.), sound, vibration, and/or any other form of feedback. In another embodiment, one or more of these types of feedback may be configured by the user. For example, in one embodiment, the user may select a sound to be played whenever an implement of interaction exceeds a predefined contact pressure level with the backtouch interface. In another embodiment, the sound may play whenever there is a change in touch state. In yet another embodiment, the feedback may be limited to only those instances where the contact pressure on the backtouch interface is estimated to be intentional, rather than incidental. As a specific example, in one embodiment, feedback might be given for a finger press, but not for a palm squeeze incidental to the user's grip.

[00348] In one embodiment, the user interface **1500** may include buttons **1518** which allow the user to specify the colors which represent the extrema of measurable contact pressure levels. As an option, the colors may be specified with a transparency value.

[00349] In one embodiment, the user interface **1500** may include a plurality of checkboxes **1520** which represent various venues for presenting visual feedback based upon the current contact pressure level. The venues for pressure-related visual feedback include, but are not limited to, an area equivalent to or based upon the contact area, the contact point, the border of the display, the status bar, and/or any other type or example of visual feedback, including those described in other embodiments.

[00350] In one embodiment, the user interface **7.002.8-00** may include a button **1522** which returns all settings defined within user interface **7.002.8-00** to predefined default values. As an option, the user may be prompted to confirm the reset before restoring the default values.

[00351] In one embodiment, the user interface **7.002.8-00** may include buttons **1524** which allow a user to save and load collections of backtouch feedback settings. As an option, the user may be prompted to confirm loading a new collection of settings before overwriting the current collection.

[00352] In one embodiment, the user interface **1500** may include a button **1526** which allows a user to test the current backtouch feedback style settings. In one embodiment, button **1526** may enable the backtouch interface (if not already enabled) and allow the user to experience the current backtouch feedback style without being able to interact with user interface **1500** through the backtouch interface. In other words, the user may be able to expose the backtouch interface to a range of proximity, contact, and pressure interactions without accidentally changing any of the settings found within **1500**.

[00353] In another embodiment, button **1526** may present to the user a user interface which allows the user to experience the current backtouch feedback style settings within a variety of simulated scenarios. The simulated scenarios may include, but are not limited to, web browsing, photo viewing, an application launcher, an eBook, word processing, and/or any other common use scenario. As an option, the simulated use scenarios may be defined such that the user is able to experience the current feedback style in a wide range of color schemes, subject matter, degrees of screen clutter, etc.

[00354] As a specific example of a backtouch feedback style collection in use, in one embodiment, a user may configure the backtouch interface through user interface **1500** such that having a finger in proximity to the backtouch interface causes a faint blue glow to appear on the display, over the location of the finger. The blue glow becomes brighter as the distance between the finger and the backtouch interface decreases, until contact is made. Upon contact, the display shows a small plus sign at the contact point and a thin "marching ants" border around the contact area. The contact area is not shaded. As the user increases contact pressure, the contact area is shaded red, which becomes more opaque as the pressure increases.

[00355] Figure **16** shows an alternative method **1600** for defining a selection made within a user interface based in part on contact pressure, in accordance with one

embodiment. As an option, the method **1600** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the method **1600** may be carried out in any desired environment. It should also be noted that the aforementioned definitions may apply during the present description.

[00356] As shown, it is determined if a touch event has occurred. See determination **1602**.

[00357] In the context of the present description, a touch event refers to an event in which an implement of interaction comes into contact with an interaction surface. For example, in one embodiment, pressing on a pressure-sensitive backtouch interface with a finger may be a touch event. Another example may be making contact with a touch-sensitive display with a stylus.

[00358] If it is determined that a touch event has occurred, then the touch event attributes are determined. See operation **1604**.

[00359] In the context of the present description, touch event attributes refer to one or more measurable aspects of the user interaction which resulted in a touch event. The measurable aspects may include, but are not limited to, touch state, contact area, contact point, contact pressure level, pressure uncertainty, touch uncertainty, and/or any other measurable aspect of user interaction.

[00360] As shown, it is determined whether the system is in a selection mode. See determination **1606**.

[00361] In the context of the present description, a selection mode refers to a system state where a user may create or modify a selection in a display. In various embodiments, a selection mode may be represented by a system flag. Examples of when a system may be in a selection mode include, but are not limited to, instances where a selection already exists, instances where a user has indicated a desire to create a selection.

[00362] In various embodiments, a system may be placed in a selection mode through user input. For example, in one embodiment, the user may activate a selection mode by performing a predefined gesture on an interaction surface. In another embodiment, a user may activate a selection mode by exerting contact pressure on an interaction surface for a sufficient amount of time, the contact pressure exceeding a predefined threshold.

[00363] If the system is in a selection mode, then a selection is created and/or modified as a function of the touch event attributes, then displayed. See operation **1608**.

[00364] In one embodiment, the selection may be created, modified, and/or displayed using one or more selection functions. In another embodiment, a display function may be applied in addition to a selection function, to display the selection.

[00365] In the context of the present description, a display function refers to a function of one or more inputs which determines one or more properties of a display. Properties of a display may include, but are not limited to, color values for individual pixels, brightness of a display backlight or portions of a backlight, and/or any other properties of a display. Display functions may apply to an entire display and/or multiple displays.

[00366] In one embodiment, the selection may be displayed with a secondary boundary representing the uncertainty in the selection. As a specific example, there may be displayed a small, pale green tinged transparent circle to show the area that the device is certain the user selected and a surrounding pale-red tinged transparent area (a secondary boundary) representing the area where the device thinks the user may have tried to select, but is not certain.

[00367] Operation **1608** may continue to be performed, updating the selection with each iteration, until there is a change in the touch state. See determination **1614**.

[00368] If it is determined in **1606** that the system is not in a selection mode, then it is determined if a gesture has been detected. See determination **1610**. In one embodiment, a user may be given feedback when a gesture is detected. Possible feedback may include, but is not limited to, sound, flashing display, appearance of an icon, a colored border on the display, and/or any other form of user feedback.

[00369] If a gesture has been detected, then an associated action is performed. See operation **1612**. In one embodiment, any gesture may be mapped to any command or operation of the device. In another embodiment, a user may be provided feedback when the action is performed. Possible feedback may include, but is not limited to, sound, flashing display, appearance of an icon, a colored border on the display, and/or any other form of user feedback.

[00370] A gesture may be associated with a variety of different actions. These actions may include, but are not limited to, scrolling a display or user interface element, zooming in or out, modifying the display brightness, and/or any other action. With respect to making or modifying a selection through method **1600**, it is important to note that, in one embodiment, a gesture may be associated with placing the system in a selection mode.

[00371]

[00372] Figure **17** shows a user interface **1700** for performing operations on a selection, in accordance with one embodiment. As an option, the user interface **1700** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **1700** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00373] After selecting an object, using methods, interfaces, and embodiments described above or others, a user may wish to perform an operation upon the object. Providing easily accessible methods to perform operations on a selected object may enhance the user experience. The availability of multiple interaction surfaces and/or multiple sensitivities increases the number of simple, intuitive ways a user may interact with the device.

[00374] In various embodiments, once a selection is static, a user may perform operations on the selected object. Possible operations include, but are not limited to, replication operations (i.e. copy, cut, paste, etc.), relocation operations, text-specific

operations, image-specific operations, cross-application operations, and/or any other type of operation.

[00375] In various embodiments, a user may perform replication operations on a selected object through simple gestures and other types of user input. For example, in one embodiment, a user may copy a selected object to a clipboard (i.e. temporary storage, etc.) by pressing sharply on a pressure-sensitive front interaction surface, on top of the selected object. In another embodiment, a user may cut a selected object by pinching the object (i.e. applying contact pressure on the object through a front and back interaction surface), then applying greater contact pressure on the front interaction surface than the back interaction surface. In yet another embodiment, a user may perform a paste operation, replacing the selected object with the contents of a clipboard, by pinching the selected object, then applying greater contact pressure on the back interaction surface than the front interaction surface. Of course, in other embodiments, these operations and those following may be assigned to any other gesture, depending upon the location, number, and sensing ability of the host device's interaction surface(s).

[00376] In various embodiments, a user may perform relocation operations on a selected object through simple gestures and other types of user input. For example, in one embodiment, a user may move a selected object by making contact with the object on an interaction surface (front or back) and dragging the contact point to relocate the object. As an option, if the selection was made within a sequential or ordered environment (i.e. word processing document, etc.), the selected object may become transparent or translucent while being relocated, so the user may better see the insertion point (i.e. cursor, etc.). As shown in this example, the same dragging gesture may have different effects, depending on the state of the selection (i.e. static vs. active, etc.).

[00377] In various embodiments, a user may perform text-specific operations on a selected text object using simple gestures and other types of user input. See, for example, user interface **1700** in Figure **17**. As shown, a selection **1702** has been made within a block of text **1704**. In accordance with one embodiment, a user has temporarily magnified the selected text by bringing a finger into proximity to a back interaction

surface, said proximity localized in an area **1706** on the selection. As an option, the degree of magnification may increase as the finger is brought closer to the interaction surface.

[00378] Another example of a text-specific operation is data detection, in accordance with one embodiment. For example, in one embodiment, a user may perform a data detection operation on a selected text object by pressing on the selection with two fingers through a back interaction surface. In various embodiments, the data detection operation may highlight detectable information found within the selected text including, but not limited to, email addresses, phone numbers, dates and times, addresses, web addresses, and/or any other form of information. In another embodiment, the data detection may only highlight types of information which may be further acted upon by the device, such that after detection, a user may initiation further action by touching the highlighted information. For example, in one embodiment, touching a highlighted phone number may dial the number, touching a highlighted address may activate a navigation application, touching a date may create an appointment in a calendar application, and so forth.

[00379] Other examples of text-specific operations may include, but are not limited to, highlighting, modifying the style (i.e. bold, italic, underline, strikethrough, color, shadow, etc.), modifying the font, modifying the font size, translating to a different language, and/or any other operation which may be performed on text. In one embodiment, a user may perform a gesture, such as a two finger, double press on the selected text, to bring up a menu of different text operations (or any other operation in the present description). In another embodiment, each operation may be assigned a different gesture.

[00380] In various embodiments, a user may perform image-specific operations on a selected object through simple gestures intuitively related to the operation being performed. For example, in one embodiment, a user may apply facial recognition and/or red eye removal by simply tapping (on a front or back interaction surface) on peoples faces within the selection. In another embodiment, a user may resize the selected object using the two finger pinch-to-zoom gesture. In still another embodiment, a user may

rotate the selected object by making two points of contact on the object, then twisting the contacting hand. In yet another embodiment, a user may warp or distort a selected object by applying different levels of contact pressure to the selection via a front or back interaction surface. As an option, a front interaction may cause puckering, while a back interaction may cause bulging.

[00381] In various embodiments, a user may perform cross-application operations on a selected object through simple gestures. Cross-application operations may include, but are not limited to, placing the selected object in a new email or SMS message, placing the selected object in a social network posting, sending the object to an application capable of performing further operations (i.e. image editing application, word processing application, etc.), and/or any other operation which may involve another application. In other embodiments, a user may be presented with application-specific operations in a menu, in addition to cross-application operations, upon performing a predefined gesture.

[00382] Figure **18** shows a method **1800** for utilizing contact pressure-based gestures, in accordance with one embodiment. As an option, the method **1800** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the method **1800** may be carried out in any desired environment. It should also be noted that the aforementioned definitions may apply during the present description.

[00383] As shown, it is determined if a touch event has occurred. See determination **1802**.

[00384] If it is determined that a touch event has occurred, then the initial touch event attributes are determined. See operation **1804**.

[00385] As shown, the gesture is identified from gesture input. See operation **1806**.

[00386] In the context of the present description, gesture input refers to the collection of inputs, flags, and signals by which a gesture is identified and parameterized. For example, in one embodiment, a two-finger pinch gesture may be identified by two contact points combined with their motion with respect to each other, and parameterized

by the extent of the motion. Possible gesture inputs may include, but are not limited to, touch event attributes (both initial and over time), system and/or application flags, switches, buttons, states, sensor data, and/or any other possible input.

[00387] For example, in one embodiment, a gesture may be made up of one or more contact points on an interaction surface, each associated with a different contact pressure level. In another embodiment, a gesture may

[00388] In various embodiments, a gesture may be identified from gesture input. Some gestures may be able to be identified solely from initial touch event attributes. Other gestures may only be identified after the gesture has been performed for some period of time. As a specific example, a two finger pinch gesture and a two finger swipe are indistinguishable based solely on initial touch event attributes. However, once the swipe or pinch motion has begun, the gestures are immediately distinguished from each other.

[00389] In various embodiments, gestures may include one or more contact pressures being exerted upon an interaction surface. In some embodiments, pressure may be used to distinguish between two gestures. For example, in one embodiment, two gestures involving a two finger pinch on the front display and one finger contact with the backtouch interface may be defined based on the contact pressure level exerted on the back touch interface. In other embodiments, pressure may be used to specify one or more dimensions of the operation linked to a gesture. For example, in one embodiment, a swipe gesture to control scrolling of a text field may use contact pressure to determine the speed of the scrolling (e.g. higher pressure yields faster scrolling, etc.). In still other embodiments, contact pressure may be used to both identify as well as parameterize a gesture.

[00390] Once the gesture has been identified, the corresponding operation is performed. See operation **1808**.

[00391] In various embodiments, a gesture may be associated with the modification of one or more device audio settings. For example, in one embodiment, a user may adjust

the device volume by applying contact pressure to a back interaction surface, and dragging the contact point up and down to increase or decrease the device volume. As an option, the ringer volume may be adjusted by dragging left and right. In another embodiment, a user may change where audio is sent (i.e. internal speakers, headphone jack, Bluetooth device, etc.) by quickly pressing twice on a back interaction surface, then selecting an audio destination from a pop-up menu presented in response to the rapid double press. In yet another embodiment, a user may mute the audio by applying pressure on front and back interaction surfaces (i.e. a pinching motion), in a predefined corner of the device. In various embodiments, some or all of these device audio related operations may be available through gesture only after the user provides some form of activation input (i.e. touching an icon in the status bar, performing a simple gesture, etc.). In this way, gesture collisions (i.e. assignment of more than one operation to the same gesture, etc.) may be avoided. Of course, these and the following operations may be associated with any gesture or other user input, in various embodiments.

[00392] In various embodiments, a gesture may be associated with the modification of one or more display attributes. For example, in one embodiment, a user may adjust the intensity of a display backlight by applying contact pressure with three fingers on the display of interest. As an option, the backlight intensity may vary as a function of average contact pressure among the three contacts. In another embodiment, a user may activate a "night mode" (i.e. use of a monochrome red/black color space, inverting the display, etc.) by sequentially applying and releasing contact pressure on each corner of a front interaction surface. In various embodiments, some or all of these display attribute related operations may be available through gesture only after the user provides some form of activation input (i.e. touching an icon in the status bar, performing a simple gesture, etc.). In this way, gesture collisions (i.e. assignment of more than one operation to the same gesture, etc.) may be avoided.

[00393] In various embodiments, a gesture may be associated with the modification of one or more device attributes. For example, in various embodiments, a device may have a status bar along one side of a display which indicates the status of device hardware including, but not limited to, Wi-Fi signal strength, cellular network type and signal

strength, Bluetooth status (i.e. on, off, etc.), system volume, battery remaining, etc. In some embodiments, a user may modify the status of these device attributes by interacting with status bar icons. For example, in one embodiment, a user may apply front surface contact pressure, above a predefined threshold for a predefined amount of time, to a status bar icon to turn off the associated device hardware (i.e. Wi-Fi, cellular modem, Bluetooth, etc.). In another embodiment, a user may apply similar pressure via a back interaction surface to a deactivated status bar icon to turn the associated hardware back on. As an option, applying said contact pressure may present the user with a menu of options associated with that device hardware (i.e. Wi-Fi networks to join, Bluetooth devices to pair, activate cellular voice/data/both, etc.).

[00394] In yet another embodiment, a user may apply contact pressure to a status bar battery indicator icon to activate a menu populated with one or more predefined collections of settings for various power scenarios (i.e. extreme cutbacks for low battery, high performance for charged battery, maximum performance while charging, etc.). In this way, a user may activate a power saving mode that allows them to stay connected to a Wi-Fi network while saving power by diming the display (i.e. while web browsing), and a different power saving mode which turns off Wi-Fi without having to dim the display as much (i.e. reading an eBook, etc.).

[00395] In various embodiments, some or all of these device attribute related operations may be available through gesture only after the user provides some form of activation input (i.e. touching an icon in the status bar, performing a simple gesture, etc.). In this way, gesture collisions (i.e. assignment of more than one operation to the same gesture, etc.) may be avoided.

[00396] In various embodiments, a gesture may be associated with application-specific operations. For example, in some embodiments, a user may interact with an email client through multiple interaction surfaces. In one embodiment, a user may scroll through a list of messages and select one for viewing using a back interaction surface. In another embodiment, a user may apply a contact pressure on a back interaction surface, then swipe downward, to forward the current message. In yet another embodiment, a user

may apply a contact pressure on a back interaction surface, then swipe upward, to reply to the current message. As an option, a user may perform this gesture using two fingers to reply to all recipients of the message.

[00397] In other embodiments, a user may interact with a web browser through multiple interaction surfaces. In one embodiment, for example, a user may apply contact pressure on a link within a webpage, then perform a quick flicking gesture, in any direction, to open the link in a background tab or window. In another embodiment, a user may open a link in a new tab or window by pressing on the link through a back interaction surface for a predetermined amount of time.

[00398] In still other embodiments, a user may interact with a camera application through multiple interaction surfaces. For example, in one embodiment, a user may indicate a point to be used for white balancing by making contact with the displayed point through a back interaction surface. In another embodiment, a user may adjust one or more camera properties by applying different amounts of contact pressure on an icon shown on the front display. Possible camera properties may include, but are not limited to, aperture settings, simulated film speed, f-stop, degree of fill flash, and/or any other camera property or setting.

[00399] In even more embodiments, a user may interact with a movie player through a back interaction surface. For example, in one embodiment, a menu may fade into view when a user applies a contact pressure on the back interaction surface. The menu may allow the user to perform one or more movie-related operations, including but not limited to, toggling subtitles, selecting an audio track, selecting a chapter or bookmark, and/or any other movie related operation or setting.

[00400] In other embodiments, a user may interact with a navigation application using a back interaction surface. For example, in one embodiment, a user may cycle through a plurality of map layers by applying different levels of contact pressure on the back interaction surface. Possible map layers may include, but are not limited to, satellite images, street maps, weather maps, traffic maps, points of interest, navigational route, and/or any other type of map or navigational information.

[00401] In various embodiments, some or all of these application-specific operations may be available through gesture only after the user provides some form of activation input (i.e. touching an icon in the status bar, performing a simple gesture, etc.). In this way, gesture collisions (i.e. assignment of more than one operation to the same gesture, etc.) may be avoided.

[00402] In various embodiments, gestures may be associated with operations that are carried out as functions of the gesture input, when the gesture is performed and identified.

[00403] In various embodiments, the operations associated with a gesture may be carried out as a function of the associated gesture dynamics. In the context of the present description, gesture dynamics are aspects of a gesture that may vary without changing the identity of the gesture. Possible aspects may include, but are not limited to, contact point velocity, contact point acceleration, contact pressure velocity, contact pressure acceleration, time to complete gesture, and/or any other aspect which may be associated with a gesture.

[00404] In various embodiments, the association between gesture and operation may be context-dependent. In some embodiments, the association depends on application context (e.g. which application is active, etc.). For example, in one embodiment, a gesture which scrolls a text field in one application may turn a page when performed in a different application.

[00405] In other embodiments, the association may depend on application or state context (e.g. whether or not there is an active selection, whether or not a control has been activated, etc.). For example, in one embodiment, a gesture which may be used to select text may move text if performed when text has already been selected.

[00406] In other embodiments, the association may be independent of context, such that performing the gesture may cause the same operation to be carried out, regardless of what application is active. For example, a gesture which brings up an interface to adjust screen brightness and volume may be defined such that it is always available.

[00407] In various embodiments, a user may be given feedback as a gesture is performed. For example, in one embodiment, a representation of the gesture may be temporarily displayed as the gesture is performed. In this way, a user may be more aware of the form of the gesture they are performing. As an option, the geometry of the displayed gesture may be smoothed or otherwise modified.

[00408] Figure **19** shows an exemplary contact pressure-based gesture **1900** for scrolling a text field, in accordance with one embodiment. As an option, the gesture **1900** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the gesture **1900** may be carried out in any desired environment. It should also be noted that the aforementioned definitions may apply during the present description.

[00409] In one embodiment of a pressure-based gesture, the display properties of one or more display(s) may be altered by the gesture, including the scroll speeds of one or more scrollable objects (e.g. text fields, images, cover art, etc.). Scrolling is a common operation, and is often assigned to a gesture in touch-based devices.

[00410] In one embodiment of a pressure-based gesture, one or more inputs may alter the scrolling speeds of one or more areas on the display. As a specific example, a user may wish to scroll quickly through a very large collection of objects (e.g. contact names, pages of text, photographs, etc.), and desires to control the speed of scrolling in a simple fashion. While performing a classic swipe gesture to scroll through the collection, the user may also apply increasing contact pressure to a pressure-sensitive surface as a means of controlling the scroll speed (e.g. increased contact pressure yields increased scroll speed, etc.). A swipe, slide or other simple gesture may be used.

[00411] See, for example, exemplary pressure-based gesture **1900**. As shown, the gesture starts at touch event **1902**, which involves a single finger is making contact at location L1, at time T1, while exerting contact pressure P1, in accordance with one embodiment. The gesture ends at touch event **1904**, with touch event attributes L2, T2, and P2. In one embodiment, the sliding motion from L1 to L2 may be described as a swipe gesture.

[00412] As shown in plot **1906**, as the contact pressure increases from P1 to P2, the scroll speed increases as well, from S1 to S2. In one embodiment, scroll speed may be a display function (in this case, a function of contact pressure). As an option, pressure-dependent display functions may be step-wise, changing in response to changes in discrete touch states, or continuous functions of a contact pressure level.

[00413] In various embodiments, pressure-based gestures may be classified as either dynamic gestures or static gestures. In the context of the present description, a dynamic gesture is a gesture that requires movement across or in proximity to an interaction surface. For example, in one embodiment, a swipe gesture may be classified as a dynamic gesture.

[00414] In the context of the present description, a static gesture refers to a gesture which does not involve movement across or in proximity to an interaction surface. Examples of static gestures may include, but are not limited to, button presses (real or virtual), multi-finger taps, operating a mechanical switch, a squeeze, and/or any other gesture which does not require motion.

[00415] In one embodiment of a pressure-based gesture that alters display properties, a scroll speed display function may be defined such that scroll speed increases in a non-linear fashion with respect to the contact pressure exerted in a swipe gesture.

[00416] In various embodiments, pressure-based gestures may be associated with different operations. For example, in one embodiment, pressure-based gestures may alter one or more display properties. Display properties may include, but are not limited to, backlight intensity, pixel color, scroll bar sensitivity, slideshow rate, and/or any other property associated with a display or its contents. As a specific example, a device may be in a sleep mode with the display and backlight inactive. A user may move his finger into proximity of a backtouch sensor in order to activate the display and the display backlight intensity to a low intensity level. By increasing the pressure on the backtouch sensor, the backlight intensity may be increased. Alternatively, the backlight intensity may be initially activated at a high level (e.g. depending on ambient light sensor etc.). In this case increasing pressure on the backtouch display may dim the backlight.

[00417] In one embodiment of a pressure-based gesture that alters a scroll speed, as the gesture magnitude is increased, the rate of scroll speed increase is increased. In other words, the scroll acceleration is increased.

[00418] In the context of the present description, gesture magnitude refers to the magnitude of the associated gesture dynamics. For example, in one embodiment, the gesture magnitude of a swipe gesture may include the length of the swipe and/or the contact pressure exerted during the swipe.

[00419] In one embodiment of a pressure-based gesture to control scroll acceleration, the scroll acceleration may depend on the gesture acceleration. Thus, if the contact is such that the contact point is accelerating with time (e.g. positive contact point acceleration), the scroll acceleration may be positive. In another embodiment, if the contact point acceleration is negative then the scroll acceleration may be negative (e.g. scroll speed decreased, etc.).

[00420] In one embodiment of a pressure-based gesture to control scroll acceleration, the scroll acceleration may be both positive and negative, depending on the contact point movement.

[00421] In one embodiment of a pressure-based gesture to control scroll acceleration, other display functions may also be altered. Possible display functions include, but are not limited to, selection(s), color, shape, image and/or text magnification, indicator(s) to provide feedback to user, and/or any other display function.

[00422] The addition of pressure dependence to already established touch gestures may facilitate user interactions with the device. Often, the efficacy of a gesture is limited by the size of the device. For example, on a device which uses the pinch-to-zoom gesture, zooming far in or out may require repetition of the gesture multiple times, due to screen size.

[00423] In one embodiment, the pinch-to-zoom touch gesture may be enhanced as a pressure-based gesture; after performing the pinch motion, the user exerts pressure on the two contact points. So long as the pressure remains above a predefined threshold, the

zoom operation may continue inward/outward without requiring additional pinch motions. As an option, the speed of the zooming may be modified by changing the contact pressure level. As another option, any differences in contact pressure level between the two contact points may be dealt with by using the average pressure.

[00424] In another embodiment, the swipe to scroll touch gesture may be enhanced as a pressure-based gesture; after performing the swipe, the user exerts pressure on the contact point without further motion. So long as the pressure remains above a predefined threshold, the scroll operation may continue without requiring additional swiping motions. As an option, the scroll speed may vary as a function of the contact pressure level.

[00425] Figure **20** shows an exemplary multitouch pressure gesture **2000** for indicating a direction, in accordance with one embodiment. As an option, the gesture **2000** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the gesture **2000** may be carried out in any desired environment. It should also be noted that the aforementioned definitions may apply during the present description.

[00426] As previously discussed, touch gestures may sometimes be modified to be pressure-dependent. In various embodiments, multitouch gestures may also be modified to depend upon pressure. As an option, the modified multitouch gestures may depend upon contact pressure differences between different contact points. In the context of the present description, a multitouch pressure gesture refers to a multitouch gesture whose definition relies in part on differences in contact pressure between different contacts. The definition of multitouch pressure gestures may rely on a variation in pressure differences over time, but it is not required of all multitouch pressure gestures.

[00427] See, for example, multitouch pressure gesture **2000** shown in Figure **20**. This static gesture is based upon two points of contact, **2002** and **2004**. In various embodiments, this gesture may be used to indicate a direction. As depicted in Figure **20**, this multitouch pressure gesture is indicating a left-to-right direction, in accordance with

one embodiment. In other embodiments, this gesture may be associated with any other operation, function, or action.

[00428] In various embodiments, multitouch pressure gesture **2000** may be performed by first exerting a contact pressure upon contact point **2002** which exceeds a contact pressure being exerted upon contact point **2004**. As an option, various embodiments may require that both of these initial contact pressures exceed a predefined threshold contact pressure level. The gesture is completed by altering the contact pressures on the two contact points such that the pressure exerted on **2004** now exceeds that being exerted on **2002**. As an option, a user may be required to maintain this pressure inequality for a predefined amount of time. When this gesture is performed using two digits on the same hand, the user's hand may appear to be rocking from left to right. A right to left direction may be indicated by changing the order of the pressures. This type of multitouch pressure gesture is hereinafter referred to as a rocker pressure gesture.

[00429] In various embodiments, a rocker pressure gesture may be preferable over simply indicating a side of the device through touch or pressure because it is less likely to be accidentally performed.

[00430] In various embodiments, a rocker pressure gesture may be used to modify a selection. For example, in one embodiment, a text selection may be expanded in a particular direction by performing a rocker pressure gesture in the desired direction, and maintaining the contact pressure difference between the two contact points until the text selection is the desired size. As an option, the speed at which the selection expands may be increased/decreased by increasing/decreasing the contact pressure differential. In another embodiment, this gesture may be used to modify the shape of a selection. For example, it may be used to modify the aspect ratio of the selection boundary geometry associated with the selection.

[00431] In various embodiments, a rocker pressure gesture may be used in conjunction with a user interface adapted for the visually impaired. For example, in one embodiment, the rocker pressure gesture may be used to move a selection from one UI element (e.g. button, slider, drop down menu, etc.) to the next. When a UI element is selected, the

assistive interface may speak the object name and/or magnify it such that the visually impaired user understands. In this way, a visually impaired user may operate a user interface displayed on a screen which often provides zero tactile feedback.

[00432] In various embodiments, a rocker pressure gesture may be used to facilitate interaction with the operating system. For example, in one embodiment, this gesture may be used to traverse a list of applications running in a multi-tasking environment. In another embodiment, this gesture may be used to switch between windows in a windowed user environment. In yet another embodiment, a rocker pressure gesture may be used to increase/decrease system settings such as volume and display brightness.

[00433] In various embodiments, a rocker pressure gesture may be advantageous in an application environment. For example, in one embodiment, a rocker pressure gesture may be used to turn the pages of an electronic book, or other multipage document displayed as a book. Using a rocker pressure gesture would allow a user to hold the device with both hands while reading, thumbs in contact with the screen, without danger of accidentally triggering a page turn, nor requiring movement of hands or fingers to turn the page.

[00434] Figure **21** shows an exemplary multitouch pressure gesture **2100** for indicating a rotation, in accordance with one embodiment. As an option, the gesture **2100** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the gesture **2100** may be carried out in any desired environment. It should also be noted that the aforementioned definitions may apply during the present description.

[00435] In various embodiments, multitouch gestures may enhance the user experience with the addition of pressure sensitivity. For example, in one embodiment, the measured difference in contact pressure levels among different contact points associated with a gesture may determine one or more parameters associated with the gesture. As an option, the one or more parameters may vary as a function of the pressure differential. Possible parameters may include, but are not limited to, speed of transformation, direction of

transformation, stringency of a color matching selection, and/or any other parameter which may be associated with any type of action, command, or operation.

[00436] In another embodiment, the presence of a pressure differential among contact points associated with a gesture may change the identity of the gesture itself. For example, see Figure **21**.

[00437] As shown, multitouch pressure gesture **2100** is a static gesture based upon two points of contact, **2102** and **2104**. In various embodiments, this gesture may be used to indicate a rotation. As depicted in Figure **21**, this multitouch pressure gesture is indicating a clockwise rotation, in accordance with one embodiment. In other embodiments, this gesture may be associated with any operation, function, or action.

[00438] In various embodiments, multitouch pressure gesture **2100** may be performed by exerting contact pressure upon contact points **2102** and **2104** such that there is a pressure differential. For example, as seen in Figure **21**, the contact pressure associated with contact point **2104** is greater than that associated with contact point **2102**. As an option, various embodiments may require that both of these contact pressures exceed a predefined threshold contact pressure level. The gesture is completed by ending one or both contacts, or by performing a different gesture. As an option, a user may be required to maintain this pressure inequality for a predefined amount of time. This type of multitouch pressure gesture is hereinafter referred to as a tipping pressure gesture.

[00439] In various embodiments, a tipping pressure gesture may be used to rotate an object or element on a display. For example, in one embodiment, an image or selected portion of an image may be rotated by performing a tipping pressure gesture. As an option, the speed of rotation may depend upon the pressure differential between the two contact points. In another embodiment, the speed of rotation may vary as a function of the distance between the two contact points.

[00440] In one embodiment, the rotation may be performed in small angular increments, continuing until the tipping pressure gesture has ended. In another embodiment, the rotation is performed in 90º increments (e.g. changing the orientation of

photographs, etc.), independent of gesture length or pressure differential. In some embodiments, the rotation may be performed using one of the contact points as the axis of rotation.

[00441] In some embodiments, a tipping pressure gesture may only cause a rotation in one direction (e.g. clockwise, etc.). In other embodiments, the direction of rotation caused by a tipping pressure gesture may depend on the positions of the two contact points with respect to each other. For example, in one embodiment, if the high pressure contact point is to the right of the low pressure contact point, the rotation may be in a clockwise direction, and in a counterclockwise direction otherwise. See, for example, multitouch pressure gesture **2100**. In one embodiment, the axis used to make the left/right determination may be the horizontal axis as defined by the current device orientation. In another embodiment, the horizontal axis used to make the left/right determination is defined by device geometry, and is orientation independent. In still other embodiments, the axis used to determine left/right may be independent of device geometry or orientation (e.g. display diagonal, vertical, etc.).

[00442] In various embodiments, a tipping pressure gesture may be used to interact with user interface elements. For example, in one embodiment, a tipping pressure gesture may be used to increase/decrease the value of a selected slider or scale. In another embodiment, a tipping pressure gesture may be used to cycle through options on a circular pop-up menu. In still another embodiment, a tipping pressure gesture may be used to quickly scroll through a list of items. Possible items may include, but are not limited to, email messages, songs or videos, files, photos, and/or any other object or data.

[00443] In some embodiments, there may exist two or more gestures which are the same or similar in execution. For example, see multitouch pressure gestures **2000** and **2100**. While performing a tipping pressure gesture, if a user allows the contact pressure differential to switch sign (i.e. low pressure contact point becomes the high pressure contact point, and visa versa), a rocker pressure gesture may be recognized by the device. In various embodiments, constraints may be placed upon gestures to assist the operating system in differentiating between similar gestures.

[00444] In various embodiments, similar or identical gestures may be distinguished from each other through the context of their use. For example, in one embodiment, the recognition of a tipping pressure gesture may be limited to instances where there is a rotatable object selected; the rocker pressure gesture would be available in all other instances. By ensuring similar gestures are not recognizable in the same context, user confusion and erroneous gestures may be reduced. These contexts may include, but are not limited to, selection status, device state, active application, system flags, selection subject matter, and/or any other property or attribute by which contexts may be defined.

[00445] In various embodiments, similar or identical gestures may be distinguished from each other by the positions of points of interaction, relative to a common axis system. In the context of the present description, a point of interaction refers to any point on an interaction surface where user interaction may be localized. Points of interaction include, but are not limited to, contact points, localized proximity and/or any other localized user interaction.

[00446] In the context of the present description, a common axis system refers to an axis system upon which all gesture data may be evaluated. For example, in one embodiment, the common axis system may be tied to the device geometry (i.e. the "horizontal" axis is always parallel to the long edge of the main display, etc.). In another embodiment, the common axis system may be tied to device orientation, such that the axis system changes depending on accelerometer data (though it is still common to all gestures).

[00447] Using a common axis system, an examination of the relative positions of points of interaction associated with a gesture may facilitate differentiation. For example, in one embodiment, rocker pressure gestures may only be recognized if a line between the two contact points is within a predefined deviation from the common horizontal axis (and the tipping pressure gesture available in all other cases). In another embodiment, the similar gestures may all be available at all times, but the location of points of interaction with respect to the common axis system may be used to give priority to certain gestures over others in certain situations.

[00448] In various embodiments, similar or identical gestures may be distinguished from each other through the timing associated with their execution. For example, in one embodiment, the performance of a rocker pressure gesture may have to be done within a certain amount of time, otherwise it may be interpreted as multiple tipping pressure gestures. In another embodiment, a tipping pressure gesture may have to be held for a certain amount of time before being recognized, such that the beginning of a rocker pressure gesture may not be identified as a tipping pressure gesture.

[00449] In various embodiments, one or more pressure-based gestures may be combined with other multi-touch and/or pressure-based gestures, such that a user may give multiple gesture-based instructions without breaking contact with one or more interaction surfaces. For example, suppose a user finger is at 3 o'clock and a user thumb at 9 o'clock and finger and thumb are one inch apart, both on a front touchscreen. The user may slide finger and thumb further apart. In one embodiment, this may result in an increase in selection area, for example. The user may then increase finger pressure. This may result in the increased selection area being rotated.

[00450] In other embodiments, other combinations and permutations of various inputs and gestures on various surfaces and using various sensors may be used to simplify the user interface and make pressure-based gestures more intuitive. As an option, such gestures may be pre-programmed, or programmed by the user, or a combination of both.

[00451] Figure **22** shows an exemplary front-back gesture **2200**, in accordance with one embodiment. As an option, the gesture **2200** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the gesture **2200** may be carried out in any desired environment. It should also be noted that the aforementioned definitions may apply during the present description.

[00452] In the context of the present description, a front-back gesture refers to a two surface gesture which includes inputs on interaction surfaces located on both the front and back of the device. For example, in one embodiment, a gesture which is performed on both a front touch-sensitive display and a backtouch interface may be classified as a

front-back gesture. In some embodiments, a front-back gesture may involve multiple inputs on one or both interaction surfaces. In some embodiments, one or more inputs on one or both interaction surfaces may be pressure or proximity-based.

[00453] A front-back gesture is a type of generalized gesture. In the context of the present description, a generalized gesture refers to a gesture that includes input from multiple sources. Possible sources may include, but are not limited to, interaction surfaces, accelerometers, buttons, and/or any other functionality or hardware receptive to user interaction.

[00454] Generalized gestures may provide a more intuitive user experience, as they can sometimes closely mimic familiar actions associated with a particular task. For example, see the front-back gesture shown in Figure **22**.

[00455] As shown, front-back gesture **2200** is a dynamic gesture based upon two points of contact. In various embodiments, this gesture may be used to indicate a rotation. As depicted in Figure **22**, this front-back gesture is associated with rotation about an axis that is both parallel to the interaction surfaces and perpendicular to the motion of the contact points, in accordance with one embodiment. In some embodiments, the rotation axis may only need to be parallel to the front interaction surface (i.e. if the front and back surfaces are not parallel, etc.). In other embodiments, this gesture may be associated with any operation, function, or action.

[00456] In various embodiments, front-back gesture **2200** may be performed by making a single contact with the front interaction surface and a single contact with the back interaction surface. For example, as seen in Figure **22**, contact point **2202** is located on the front interaction surface and contact point **2204** is located on the back interaction surface. The gesture is performed by moving the contact points in opposite directions along parallel vectors. In some embodiments, the motion vectors of the two points of contact may diverge from being parallel to each other by a predefined amount and still be considered parallel. The gesture is completed by ending one or both contacts, or by performing a different gesture. This type of front-back gesture is hereinafter referred to

as a spin gesture. In one embodiment, there may also exist spin gestures which are not front-back gestures.

[00457] In various embodiments, a front-back gesture may have an associated gesture reference. In the context of the present embodiment, a gesture reference refers to a constraint or parameter of one or more dimensions which is related to the operation linked to the gesture. For example, the gesture reference for the front-back gesture **2200** may be the axis of rotation. Other gesture references may include, but are not limited to, an axis for rotation, a plane for movement, and/or any other geometric constraint upon an operation.

[00458] In various embodiments, a spin gesture may be used to rotate an object or element on a display. For example, in one embodiment, a spin gesture may be used to rotate a displayed object within a three dimensional space. As an option, the rotation of the object may be about the same axis as the gesture reference. In another embodiment, a spin gesture may be used to rotate a displayed object about an axis independent of the gesture reference. In yet another embodiment, the gesture reference (i.e. axis of rotation) may not pass through the target object (i.e. the operation may move the object through three dimensional space about an axis, etc.).

[00459] In one embodiment of a front-back gesture, an object transformed by a spin gesture may include an entire image. In another embodiment of a front-back gesture, an object transformed by a spin gesture may include one or more recognized objects in an image.

[00460] In various embodiments, a spin gesture may be used to cycle through a plurality of items or objects. For example, in one embodiment, a spin gesture may be used to cycle through a number of style options which are applied to selected text. Style options include font, type size, bold, italic, underline, predefined style combinations, and/or any other attribute which may be assigned to text.

[00461] In another embodiment, a spin gesture may be used to select a color from a spectrum. In yet another embodiment, a spin gesture may be used to cycle through

system environments. Possible system environments include, but are not limited to, applications running in a multi-tasking environment, screen layers, and/or any other environment or use scenario.

[00462] In various embodiments of a spin gesture, the gesture reference (i.e. axis of rotation) may be parallel to the display plane. For example, see the spin gesture depicted in Figure **22**.

[00463] In various embodiments of a spin gesture, the gesture reference may be perpendicular to the display plane. For example, in one embodiment, a rotation within the plane of the display (i.e. axis of rotation perpendicular to display) may be accomplished by a spin gesture performed on a single interaction surface. A single surface spin gesture is performed by making two contacts with a surface, and moving those contacts such that their motion vectors are in opposite directions, but parallel. This type of gesture may also be called a front-front gesture, to distinguish it from the front-back gestures.

[00464] In various embodiments of a spin gesture, the reference for a spin gesture may be determined by the locations of the interaction surface(s) used. For example, in one embodiment, a device may have faces in the x-plane, y-plane and z-plane. Each face may have an interaction surface. Depending on which faces are used in the spin gesture, a spin gesture operation may be performed about the x-axis, y-axis, or z-axis.

[00465] If the device was represented as a simple cube with six faces numbered according to a standard die, the types of gestures using two interaction surfaces (i.e. front-back gestures, etc.) may be enumerated.

[00466] In one embodiment, the main display (if there is one) may be declared to be the device front, and enumerated as face 1. Face 6 would be the back of the device. Faces 2, 3, 4, 5 are the sides of the device. In another embodiment, the x-, y-, and z-axis are defined to be to be perpendicular, with the x-axis passing through the centers of faces 2 and 5, the y-axis through faces 3 and 4, and the z-axis through faces 1 and 6.

[00467] With this labeling scheme, a front-back gesture would be designated as a 1-6 gesture, for example. The gesture components that together make up a complete gesture may be referred to by their faces. For example, a 1-6 gesture may comprise a 1-gesture (a first gesture component) combined with a 6-gesture (a second gesture component). In the context of the present description, this labeling scheme may be referred to as gesture notation.

[00468] In the context of the present description, gesture components refer to the individual gestures on any single interaction surface (e.g. the movement of a thumb on a touchscreen, the pressure of a finger on a backtouch sensor, etc.) that together make up a complete, compound gesture. In some embodiments, gesture components may be performed simultaneously. In other embodiments, gesture components may all be completed within a predefined window of time, but not necessarily performed simultaneously.

[00469] In various embodiments, compound gestures may be categorized by their components and operations. For example, the set of 1-6 spin gestures is a subset of all 1-6 gestures. As discussed before, a 1-6 spin gesture may be described as a 1-6 gesture in which the 1-gesture and 6-gesture move with opposite though parallel vectors (determined within some threshold of accuracy).

[00470] In one embodiment, gesture components may be characterized by their touch gesture shape, in order to catalog compound gestures. In the context of the present description, touch gesture shape refers to the basic geometry of a touch gesture. Examples of touch gesture shapes include, but are not limited to, lines, circles, arcs, and/or any other geometry.

[00471] In various embodiments, touch gesture shape is an example of a gesture attribute. In the context of the present description, a gesture attribute refers to a characteristic or property which may be associated with a gesture. In one embodiment, all gesture components have at least one gesture attribute. Possible characteristics and properties may include, but are not limited to, touch gesture shape, gesture scale, gesture size, gesture orientation, time, pressure gesture shape, proximity gesture shape, maximum

pressure, and/or any other data which may be associated with a gesture. In the context of the present description, gesture orientation refers to the relative angle of a touch gesture shape, with respect to a predefined axis.

[00472] In the context of the present description, a pressure gesture shape refers to the path a contact point takes through a one dimensional pressure phase space. As an option, it may be correlated with the physical locations that make up a gesture shape. In the context of the present description, a proximity gesture shape refers to the path a detectable implement of interaction takes through a three-dimensional proximity phase space (the third dimension being estimated proximity). In various embodiments, the proximity gesture shape may be represented by the time evolution of local maxima of the data received by a proximity-sensitive interaction surface. In various embodiments, a pressure or proximity gesture shape may be the result of simplifying pressure or proximity data and identifying a matching basic geometry or predefined shape.

[00473] In various embodiments, gesture attributes may also include derived inputs. In the context of the present description, a derived input is an input which is derived or calculated from other inputs. Derived inputs may include, but are not limited to, velocity, acceleration, uncertainty, statistical variation, and/or any other derived input. As an option, these concepts, including but not limited to velocity and acceleration, may also be applied to pressure and/or proximity.

[00474] For example, in one embodiment, the device may sample one or more inputs at time t1 and time t2. The inputs sensed may include a contact point PT(X1, Y1, t1) = PT1 at time t1 and PT(X2, Y2, t2) =PT2 at time t2. The device may then calculate the velocity of part(s) of a gesture by calculating the distance D(t2) = D2 between PT1 and PT2 and dividing the distance D2 by the time difference t2 – t1, to form an estimate of the gesture velocity V(t2) = V2.

[00475] In various embodiments, the gesture attributes may be used to alter the function performed by the gesture, as well as distinguish gestures. For example, in one embodiment, the velocity and acceleration of one or more of the gesture components comprising a spin gesture may modify the speed of the resulting rotation. In one

embodiment, a gesture attribute may be a member of a predefined set (i.e. basic geometries and shapes, etc.) or have a numerical value.

[00476] In another embodiment, gestures may be differentiated by a single gesture attribute. For example, in one embodiment, there may be a 1-6 gesture, where the 1-gesture and 6-gesture are both lines. If the two component gestures have the same gesture orientation (i.e. the lines are in the same direction), the compound gesture may cause an object to move in the direction of the lines. If the two component gestures have opposite orientation (i.e. the lines are in opposite direction), the compound gesture may be a spin gesture and cause an object to rotate about the gesture reference.

[00477] In various embodiments, a touch gesture shape may be determined from a collection of contact point data taken over time. In one embodiment, the touch gesture shape associated with a set of contact points may be determined by constructing a best-fit line. In another embodiment, the touch gesture shape may be determined using a linear regression. In yet another embodiment, the touch gesture shape may be determined by fitting a shape (i.e. circle, square, triangle, etc.), or a fragment of a shape, to a series of averaged contact points. In other embodiments, similar techniques may be used to determine pressure or proximity gesture shapes.

[00478] In various embodiments, a compound gesture may be characterized by the gesture attributes of the component gestures. For example, in one embodiment of a 1-6 spin gesture, the gesture reference may be defined as the y-axis if the 1-gesture has a gesture angle (i.e. angle of the fitted line, etc.) within 30 degrees of the x-axis. As an option, the 1-angle and 6-angle may be averaged.

[00479] In one embodiment, spin gestures may be limited to those whose gesture components occur on parallel planes (i.e. 1-6, 2-5, 3-4, etc.). In another embodiment, spin gestures may also be performed using component gestures occurring on perpendicular planes. For example, a 1-2 spin gesture. In yet another embodiment, the gesture reference (i.e. axis of rotation) for spin gestures occurring on perpendicular planes may be restricted to an axis bisecting the angle between the planes, perpendicular

to their intersection. In this way, the tactile association with spinning may be preserved (i.e. use of other axes may reduce the intuitive nature of the user experience).

[00480] In one embodiment, spin gestures may be constrained, such that the gesture reference is limited to axes parallel to the x-, y-, and z-axes. In other words, each spin gesture occurring on two parallel planes would have only two possible axes of rotation.

[00481] In various other embodiments, the axis of rotation for a spin gesture performed on two parallel planes (i.e. 1-6, etc.) may be less constrained. For example, in one embodiment, the axis of rotation may be defined as the axis which is perpendicular to the motion of the component gestures, and parallel to the component planes. In another embodiment, the constraint may be further reduced, by defining the axis of rotation as the axis which is normal to the plane which contains (within some tolerance) the motion of both component gestures.

[00482] In one embodiment, the axis of rotation associated with a spin gesture may be required to pass through the center of the device. In another embodiment, the axis of rotation may be required to pass through the center of a primary display. In yet another embodiment, the axis of rotation may pass through a point determined by an application and/or through user input.

[00483] In one embodiment, spin gestures may be defined and used in the manners discussed, even if the device has more or less than six faces, not all faces have interaction surfaces, and/or faces are not parallel or perpendicular. As an option, the operation performed by a spin gesture on such a device may be adapted to the device's physical geometry, such that the tactile familiarity of the gesture may be preserved.

[00484] Another example of a 1-6 gesture is the pivot gesture. In the context of the present description, a pivot gesture refers to a 1-6 gesture where one component gesture is a static contact point, and the other component gesture is a circle or arc (i.e. the gesture shape is a circle or arc) which would contain the static contact point if the two faces were combined. In one embodiment, a pivot gesture may be associated with a rotation operation, where an object is rotated about the static contact point, within the plane of the

display. In another embodiment, a pivot gesture may be used to cycle through items in a circular menu, such as a pop-up menu.

[00485] Figure **23** shows an exemplary virtual control knob gesture **2300**, in accordance with one embodiment. As an option, the gesture **2300** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the gesture **2300** may be carried out in any desired environment. It should also be noted that the aforementioned definitions may apply during the present description.

[00486] In various embodiments, a virtual control knob gesture is a subset of the spin gesture. In some embodiments, there may be slight overlap with the set of spin gestures.

[00487] In some embodiments, virtual control knob gestures may provide a user with a more intuitive experience, as they mimic the manner in which a user would operate a real control knob. For example, see virtual control knob gesture **2300**.

[00488] As shown, virtual control knob gesture **2300** is a dynamic gesture based upon two points of contact. In various embodiments, virtual control knob gestures are very similar to spin gestures, except they may be performed upon a knob or knob-like implement displayed on the screen. See for example virtual control knob **2302**.

[00489] In various embodiments, virtual control knob gesture **2300** may be performed by making a single contact with the front interaction surface and a single contact with the back interaction surface, both contacts being on or in close proximity to a control knob on the display. For example, as seen in Figure **23**, contact point **2302** is located on the front interaction surface and contact point **2304** is located on the back interaction surface, both contact points located on top of/below displayed control knob **2306**.

[00490] The gesture is performed by moving the contact points as though the knob is being turned. For example, in Figure **23**, contact point **2302** may be moved upward while contact point **2304** may be moved downward, as indicated. The gesture is completed by ending one or both contacts, performing a different gesture, or when one or both contact points is no longer in proximity to the displayed control knob.

[00491] As a specific example, in one embodiment, the device may be held in a user's right hand. The user's right hand thumb makes contact point **2302** on the front touchscreen, and their right hand index finger makes contact point **2304** on a backtouch interface, both contacts located on the displayed control knob **2306**. As the gesture is performed, the control knob turns, incrementing the associated dial **2308**.

[00492] In another embodiment, a virtual control knob gesture may be performed on a single interaction surface. See, for example, gesture **2310**. Contact points **2312** and **2314** are made on top of displayed control knob **2316**, which rotates on an axis perpendicular to the display. The gesture is performed by moving the contact points in a circular motion, as depicted.

[00493] In another embodiment, the device may be receptive to the performance of two virtual control knob gestures at the same time. For example, a user may turn both control knobs seen in the depiction of gesture **2300**. In yet another embodiment, one or more virtual control knob gestures may be used in combination with other types of gestures. In other embodiments, a virtual control knob gesture may be associated with any other operation, action, or process.

[00494] In some embodiments, virtual control knob gestures may be differentiated from spin gestures because they may be performed upon a knob or knob-like implement displayed on the screen. In other embodiments, virtual control knob gestures differ for spin gestures in that their definition may depend upon which fingers are used. See for example virtual control knob gesture **2318**.

[00495] In various embodiments, gestures may be defined such that particular fingers may be used. In one embodiment, fingers may be identified by the size of the contact area. In another embodiment, fingers may be identified by their relative location on interaction surfaces, taking into account other skin contact. In yet another embodiment, fingers may be identified though a combination of these and other methods. As an option, a user may be prompted to identify their fingers within a training interface.

[00496] As shown in Figure **23**, in one embodiment, gesture **2318** may be performed by making contact point **2320** with an index finger on a front interaction surface, and contact point **2322** with the thumb of the same hand, on a backtouch interface. The gesture is performed by moving the contact points in opposite directions, as depicted in Figure **23** and as described in other embodiments. In one embodiment, this gesture may be distinguished by the same motion and contact points being made with the thumb on the front and the index finger on the back.

[00497] In various embodiments, the user may be provided feedback as they perform a virtual control knob gesture. For example, in one embodiment, a displayed virtual control knob may have markings which make apparent the rotation of the displayed knob as the gesture is performed. In another embodiment, performance of the gesture on a displayed virtual control knob may produce a clicking sound. As an option, the frequency of the clicks may vary as a function of how fast the displayed knob is being turned.

[00498] Figure **24** shows a method **2400** for gesture differentiation, in accordance with one embodiment. As an option, the method **2400** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the method **2400** may be carried out in any desired environment. It should also be noted that the aforementioned definitions may apply during the present description.

[00499] As shown, potentially gesture related input is gathered. See operation **2402**.

[00500] In various embodiments, there may exist a tiered approach to interpreting user input. For example, in one embodiment, it may first be determined whether there is any meaningful user input.

[00501] In various embodiments, each form of user input may have an associated threshold above which user interaction may rise before it may be deemed meaningful. Examples of such thresholds may include, but are not limited to, minimum threshold pressure level, minimum threshold proximity level, maximum context determination

zone, and/or any other threshold meant to prevent acting on unintentional user interactions.

[00502] In one embodiment, upon detecting meaningful user input, it is then determined whether it is related to a basic user interaction (i.e. button press, basic application interaction, etc.). If the meaningful user input is not related to a basic user interaction, then it may potentially be related to a gesture. In one embodiment, that determination may result in the initiation of method **2400**. In another embodiment, system and environmental information may be gathered, in addition to the meaningful user input.

[00503] Once potential gesture input has been received, it is determined if the gesture is possibly a generalized gesture. See determination **2404**.

[00504] A generalized gesture is a gesture that includes input from multiple sources, including but not limited to multiple interaction surfaces. In one embodiment, the possibility of a generalized gesture may be determined by ascertaining the number of sources provided the meaningful input gathered in operation **2402**.

[00505] If it is determined that the gesture is possibly a generalized gesture (i.e. there is meaningful input from more than one source), then possible gesture identities are deduced using gesture attributes, gesture components, and gesture notation. See operation **2406**.

[00506] In various embodiments, the identification of a generalized gesture may begin with the identification of the component gestures. Component gestures may be identified by evaluating associated gesture attributes, as has been previously discussed. In one embodiment, once the component gestures have been identified, gesture notation may be used to narrow the field of potential gestures.

[00507] In one embodiment, a generalized gesture may be identified by first identifying the faces involved, then further limiting the possibilities by identifying characteristic gesture attributes in common, until the number of potential generalized

gestures is either one, or cannot be further reduced. In another embodiment, a decision tree may be used to identify the generalized gesture.

[00508] If it is determined that the gesture is not a generalized gesture (i.e. there is meaningful input from only one source), then possible gesture identities are deduced using gesture attributes. See operation **2408**.

[00509] In various embodiments, the same methods used to identify component gestures in operation **2406** may be used in this operation.

[00510] As shown, it is determined if there is a single possible gesture being performed. See determination **2410**. In one embodiment, if there are no potential gestures, the user may be provided feedback indicating a failure to recognize the gesture. Feedback may be visual, aural, or both.

[00511] If there is more than one possible gesture being performed, then the gesture subject matter context is used to identify the gesture. See operation **2412**.

[00512] In the context of the present description, gesture subject matter context refers to any subject matter which may be related to a gesture. Examples may include, but are not limited to, selected objects, objects displayed at or near the location of the gesture, and/or any other items, objects, or properties which may give context to a gesture.

[00513] In some embodiments, similar gestures may require the use of gesture subject matter context to differentiate. For example, in one embodiment, a spin gesture and virtual control knob gesture may be indistinguishable until it is determined whether there is a displayed control knob at the contact points.

[00514] After using the gesture subject matter context to identify the gesture, it is determined whether there is a single possible gesture being performed. See determination **710.010.1-14**.

[00515] If there is still more than one possible gesture being performed, then a default response is performed. See operation **2416**.

[00516] In one embodiment, the default response to the inability to identify the gesture may be to do nothing, and stop applying method **2400**. In another embodiment, the default response may be to restart method **2400**, in the hope that sufficient additional user input may have been gathered that may assist in gesture differentiation.

[00517] In yet another embodiment, the default response to the inability to identify the gesture may be to prompt the user to clarify their recent input. For example, in one embodiment, the user may be prompted by a pop-up interface which allows the user to select from identified potential gestures, or indicate that none of the candidates are correct. Since such user interaction could possibly interrupt the performance of the gesture, the user may have to repeat the gesture after making an appropriate selection.

[00518] If the identity of the gesture being performed has been determined, it is then determined whether the gesture is meant to trigger some sort of transformation. See determination **2418**. Examples of possible transformations include, but are not limited to, move, slide, scale, rotate, and/or any other spatial or geometric transformation.

[00519] If the gesture is meant to trigger a transformation, then the gesture reference is determined. See operation **2420**.

[00520] In various embodiments, gestures which cause a transformation to occur may have an associated gesture reference. For example, in one embodiment, a spin gesture may have a gesture reference (i.e. axis of rotation). In various embodiments, a gesture reference may be determined using the type or specific identity of the gesture and/or gesture attributes (i.e. gesture orientation, gesture shape, etc.).

[00521] As shown, the intermediate operations associated with the identified gesture are performed. See operation **2422**.

[00522] In the context of the present description, an intermediate gesture operation refers to operations associated with a gesture which may be executed before the gesture is done. For example, a spin gesture which rotates a displayed object may cause rotation to occur and be displayed before the user has stopped performing the gesture. In this way, a

user may receive feedback on their gesture before terminating the interaction. Not all gestures have associated intermediate operations.

[00523] In one embodiment, each execution of intermediate gesture operations may cause an update of the gesture attributes before execution begins. In this way, the feedback given to the user performing the operation is responsive to their performance.

[00524] In another embodiment, each execution of intermediate gesture operations may trigger a display update (i.e. rendering, etc.). As an option, the display update may be localized to the portions of the display affected by the gesture operations, and may be more frequent than display updates for other user interface elements.

[00525] The intermediate operations are continuously performed unless it is determined that the gesture has ended. See determination **2424**.

[00526] In various embodiments, a gesture may be ended by ending one or more user inputs. For example, in one embodiment, a spin gesture may be ended by breaking contact with one or two interaction surfaces. In another embodiment, a proximity-based gesture may be ended by reducing at least one proximity below a predefined threshold. In yet another embodiment, a pressure-based gesture may be ended by reducing the contact pressure level associated with one or more contact points below a predefined threshold level.

[00527] In various embodiments, a gesture may be ended by performing a different gesture. In one embodiment, the performance of a new gesture may be detected by changes observed in one or more gesture attributes. As an option, the changes may have to exceed a predefined threshold before identification of a new gesture is attempted. In some embodiments, a gesture may not necessarily be able to end all other gestures.

[00528] Once it has been determined that the gesture has ended, the final operations associated with the identified gesture are performed. See operation **2426**.

[00529] In various embodiments, gestures may be associated with one or more final operations. In the context of the present description, a final operation refers to an

operation which is executed when a triggering gesture has ended. For example, in one embodiment, a gesture which triggers the display of a pop-up menu and allows a user to cycle through menu items may have as a final operation the selection of the current menu item and removal of the pop-up menu.

[00530] Figure **25** shows an exemplary anchor and move gesture **2500**, in accordance with one embodiment. As an option, the gesture **2500** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the gesture **2500** may be carried out in any desired environment. It should also be noted that the aforementioned definitions may apply during the present description.

[00531] As shown, anchor and move gesture **2500** is a dynamic front-back gesture based upon two points of contact. In various embodiments, this gesture may be used to anchor one object while modifying another. Possible modifications include, but are not limited to, translation, scale, rotate, and/or any other operation that may be performed on a graphic object. As depicted in Figure **25**, this front-back gesture is associated with the anchoring of one graphic object and the translation of another graphic object, in accordance with one embodiment. In other embodiments, this gesture may be associated with other modifications, including but not limited to text modification, user interface interactions, selection modifications, and/or any other type of modification which may be associated with a gesture. In still other embodiments, this gesture may be associated with any operation, function, modification or action.

[00532] In various embodiments, anchor and move gesture **2500** may be performed by making a single contact with the front interaction surface and a single contact with the back interaction surface. For example, as seen in Figure **25**, contact point **2502** is located on the front interaction surface and contact point **2504** is located on the back interaction surface. In one embodiment, the gesture is performed by moving one contact point while keeping the other stationary. For example, as shown, contact point **2502** may be kept stationary while contact point **2504** may move along a line. In various embodiments, this may result in an anchor object **2506** remaining unchanged, and a dynamic object **2508**

being translated in the same direction as the contact point movement. As an option, the dynamic object may be modified as a function of the movement of the non-stationary contact point.

[00533] In the context of the present description, an anchor object refers to an object which remains unchanged by the anchor and move gesture. Additionally, in the context of the present description, a dynamic object refers to an object which is modified by the anchor and move gesture. In some embodiments, there are multiple ways to select the anchor and/or dynamic object.

[00534] In various embodiments, the dynamic and/or anchor object may be selected through pressure-based methods. For example, in one embodiment, one of these two objects may be selected using method **4**. In another embodiment, both objects may be selected using method **4**, after the user has indicated that a dual selection was to be made. As an option, the user may use a gesture to indicate the use of a dual selection. In still another embodiment, one or both objects may be selected using a gesture.

[00535] In various embodiments, the dynamic and/or anchor object may be selected as part of the anchor and move gesture. For example, in one embodiment, a user may select an anchor object by placing the implement of interaction making the stationary contact point on top/beneath it. In another embodiment, the dynamic object may be selected in a similar manner, using the implement of interaction whose interaction with the interaction surface may be dynamic. As an option in these embodiments, the selection of an object may be assisted by recognition routines (i.e. facial recognition, color matching, OCR, etc.). In yet another embodiment, method **6** may be used in conjunction with these embodiments to select one or both objects.

[00536] In various embodiments, the anchor and move gesture may be associated with modifying one object with respect to another. For example, in one embodiment, this gesture may be used to arrange photos in an environment which imitates a light table.

[00537] In some embodiments, the anchor and move gesture may be enhances with pressure sensitivity. For example, in one embodiment, a user may use contact pressure

exerted on the dynamic contact point to modify a different aspect of the dynamic object. Possible aspects include, but are not limited to, scale, orientation, transparency, and/or any other aspect which may be made to vary with pressure.

[00538] In various embodiments, the anchor and move gesture may utilize one or more cues to provide feedback to the user. In the context of the present description, a cue refers to feedback, visual or otherwise, which is provided to the user. Examples of cues include, but are not limited to, the display logic within selection functions, visual representation of contact point movement, proximity feedback, contact pressure level feedback, and/or any other form of user feedback. For example, in one embodiment, the anchor object may be highlighted (e.g. change of color, text highlight, 3D representation, flashing or other dynamic selection area behavior, etc.)

[00539] In various embodiments, cues may include 3D or pseudo-3D effects. For example, in one embodiment, the anchor object may be made to appear to be floating above a background image, using 3D cues.

[00540] In various embodiments, a cue may modify a 2D representation into a pseudo-3D representation responsive to user inputs such as device orientation. For example, in one embodiment, the shadow of an object may be adjusted according to device orientation, to provide the illusion that an object is floating above a background. In another embodiment, the pseudo-3D representation may be responsive to user focus (including, in some embodiments, user gaze).

[00541] In various embodiments, the anchor and/or dynamic object may be text (i.e. letter, word, sentence, paragraph, etc.), graphics, recognized object(s), and/or any part or portion of the display.

[00542] In various embodiments, the anchor and move gesture may be used to allow easy modification of a single object. For example, in one embodiment, a user may select an anchor object, and then modify it as a function of the dynamics of the other contact point. As a specific example, a user may anchor a text selection, and cycle through different fonts by changing the distance between the anchor contact point and the

dynamic contact point. As an option, pressure applied upon the dynamic contact point may cycle through different attributes of the anchor object which are modified as a function dynamic contact point location. Possible attributes include text style, color, transparence, image layer, and/or any other object attribute

[00543] Figure **26** shows an exemplary twist gesture **2600**, in accordance with one embodiment. As an option, the gesture **2600** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the gesture **2600** may be carried out in any desired environment. It should also be noted that the aforementioned definitions may apply during the present description.

[00544] As shown, twist gesture **2600** is a dynamic 1-6 gesture based upon four points of contact. It is intended to mimic a twisting motion performed on the device. In various embodiments, twist gesture **2600** may be performed by making two contacts with the front interaction surface, and two contacts with the back interaction surface. In one embodiment, this is accomplished by holding the device with two hands, as depicted in Figure **26**. As shown, the user's thumbs make front side contact points **2602** and **2604**, and the user's index fingers make back side contact points **2606** and **2608**, in accordance with one embodiment. In other embodiments, different fingers and/or implements of interaction may be used.

[00545] The motion associated with one embodiment of twist gesture **2600** is such that all contact points may be slid in directions parallel to each other and perpendicular to the twisting axis. The contact points on the front interaction surface move in opposite directions, and the movement directions of the contact points on the back interaction surface are opposite those on the front surface. These movements imitate the motion of twisting the device.

[00546] In another embodiment, a twist gesture may be a static 1-6 gesture similar to **2600**, except the thumbs and fingers are located such that like digits form a diagonal across the device. In accordance with one embodiment, this static gesture may be recognized by the application of contact pressure levels to the contact points which

exceed a predefined lower threshold. In yet another embodiment, a twist gesture may involve a combination of contact point movement and contact pressures.

[00547] In various embodiments, a twist gesture may be associated with the deletion of one or more items. Possible items may include, but are not limited to, email messages, files, selected objects, and/or any other item which may be deleted. As an option, the deletion operation associated with a twist gesture may also include a visual routine which makes it appear that the item being deleted is being torn in half by the gesture.

[00548] In various other embodiments, a twist gesture may be associated with operations which are intuitively associated with the action of wringing out a wet cloth. For example, in one embodiment, a twist gesture may cause the compression of a selected file.

[00549] In another embodiment, a twist gesture may trigger the application of data detection routine(s) to selected text. In the context of the present description, a data detection routine refers to a routine which parses text data, creating hyperlinks for actionable information. Types of actionable information may include, but is not limited to, phone numbers, contact names, dates, addresses, and/or any other type of text data which may be associated with an operation or application. In one embodiment, the hyperlinks created by the data detection routine(s) may pass the actionable information to the appropriate application or function (i.e. selecting a phone number may dial it, selecting an address may display a map, etc.).

[00550] In one embodiment, a twist gesture may cause a twisting distortion to be applied to an image. As an option, the magnitude of the distortion may be a function of the average distance traveled by the four contact points. In another embodiment, a twist gesture may be associated with any operation, action, or function.

[00551] Figure **27** shows an exemplary implementation **2700** of a method to determine touch states using a touch-sensitive interface, in accordance with one embodiment. As an option, the gesture **2700** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however,

the gesture **2700** may be carried out in any desired environment. It should also be noted that the aforementioned definitions may apply during the present description.

[00552] Figure **27** shows a method **2700** for determining touch states using a touch-sensitive interface, in accordance with one embodiment. As an option, the method **2700** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the method **2700** may be carried out in any desired environment. It should also be noted that the aforementioned definitions may apply during the present description.

[00553] In various embodiments, method **2700** may be used to provide basic pressure sensitivity for interaction surfaces which are touch-sensitive but not pressure-sensitive. In one embodiment, it may be limited to user interactions involving fingers or thumbs. It is based on the fact that as a digit presses harder on a surface, the contact area increases.

[00554] As shown, a contact area is measured. See operation **2702**.

[00555] In one embodiment, the number of touch states which can be consistently recognized with this method may depend upon how accurate the touch-sensitive interaction surface is able to determine the contact area.

[00556] After measuring the contact area, the digit is identified. See operation **2704**.

[00557] In various embodiments, a digit in contact with an interaction surface may be identified using incidental user interactions. Incidental user interactions include, but are not limited to, skin contact with various touch-sensitive interaction surfaces, detectable proximity to proximity-sensitive interaction surfaces, device orientation, and/or any other data related to how the user is holding or using the device.

[00558] As shown, the touch state is calculated. See operation **2706**.

[00559] In various embodiments, the touch state may be calculated by comparing the measured contact area with a digit pressure response profile. In the context of the present description, a digit pressure response profile refers to a description of how the contact area associated with a digit changes as a function of contact pressure exerted. In one

embodiment, a digit pressure response profile may depend upon a numerical measurement of the contact area. In another embodiment, a digit pressure response profile may also depend upon the shape of the contact area.

[00560] In various embodiments, a collection of one or more digit pressure response profiles may be predefined. In one example, a plurality of response profiles may be created and/or improved through prompting a user to provide examples of different levels of contact pressure exerted by specific digits. As an option, the user may be prompted to provide exemplary digit contact/pressure while holding the device in one of a plurality of predefined device grips (i.e. manner of holding the device).

[00561] In another embodiment, if the device possesses at least one touch-sensitive interaction surface which is also pressure-sensitive, one or more digit pressure response profiles may be created or improved by prompting the user to exert contact pressure on that surface while said surface is situated in a variety of orientations and locations (i.e. as a front surface, as a back surface, as a side surface, etc.). The user may be prompted to exert contact pressures representative of various touch states. As an option, the user may be provided with alternative forms of feedback during this process, as the device may need to be held such that the user cannot see the display. The alternative forms of feedback may include, but are not limited to, vibration, sound, LEDs, and/or any other form of feedback which does not require that the display be visible.

[00562] Some devices may not have a sufficient number of interaction surfaces to accurately determine the identity of a particular digit. In one embodiment, a generic digit pressure response profile may be predefined, to be used in situations where a digit cannot be identified. In another embodiment, a user may be prompted to identify an unrecognizable digit.

[00563] Figure **28** shows exemplary pattern-based gestures **2800**, in accordance with one embodiment. As an option, the gesture **2800** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the gesture **2800** may be carried out in any desired environment. It

should also be noted that the aforementioned definitions may apply during the present description.

[00564] In the context of the present description, a pattern-based gesture refers to a gesture defined, at least in part, by a path through a phase space of one or more dimensions. See, for example, the pattern-based gestures **2800**.

[00565] In some embodiments, pattern-based gestures may be based upon paths through a phase space whose dimensions do not include any measure of location on an interaction surface. In other words, in these embodiments, pattern-based gestures may not depend upon the location or movement of user interaction across an interaction surface. In one embodiment, these gestures may act as gesture components, which may be combined with other gesture components to form generalized gestures which may or may not be based upon location on or movement across an interaction surface.

[00566] See, for example, the pattern-based gesture **2802**, described by a phase space path which represents the time evolution of the touch state associated with a contact point located on a front interaction surface. The location of the contact point upon the interaction surface is of no importance.

[00567] Pattern-based gesture **2802** is based upon three touch state transitions. At time t1, the contact point changes from touch state s0 to s1. Similar changes of state occur at t2 and t3. In various embodiments, these touch state transitions may be represented by ordered 2-tuples. For example, the first transition may be represented as (s1, t1). The pattern-based gesture itself may be represented as an ordered sequence of state transitions. For example, pattern-based gesture **2802** may be represented as {(s1,t1),(s2,t2),(s0,t3)}.

[00568] In other embodiments, pattern-based gestures may be based upon paths through a phase space that may include one or more measures of physical location upon an interaction surface. In other words, there may exist paths through the phase space that describe motion across an interaction surface.

[00569] In various embodiments, a pattern-based gesture may be based upon paths through a phase space which does not include time as a dimension. For example, in one embodiment, a pattern-based gesture may be defined by a series of transitions between touch states, where it does not matter how long the individual touch states are maintained.

[00570] In other embodiments, a pattern-based gesture may depend upon time to the extent that the entire path through phase space may be completed within a predefined amount of time (though still independent of how much time is spent at each state along the path). In still other embodiments, the gesture may be defined, in part, by the time spent on each leg of the path through phase space.

[00571] In one embodiment, a time-independent pattern-based gesture may be combined with a dynamic touch gesture (i.e. a one finger swipe, etc.) to form a generalized gesture.

[00572] In various embodiments, a touch state may be associated with a specific contact point upon a pressure-sensitive interaction surface. In other embodiments, the device may not be able to resolve the specific source of pressure exerted upon an interaction surface; in such cases, the pressure may be represented by surface touch states. In the context of the present description, a surface touch state is a partitioning of the range of measurable pressures exerted upon a surface (similar to touch states associated with contact points, but not contact point specific).

[00573] In various embodiments, there may exist a subset of generalized gestures referred to as pattern-based generalized gestures. In the context of the present description, a pattern-based generalized gesture refers to a pattern-based gesture which includes input from multiple sources. For example, see pattern-based gesture **2806**, which is represented by a path through a two-dimensional phase space.

[00574] As shown, gesture **2806** is a pattern-based generalized gesture based upon the evolution of a touch state on a front interaction surface and a back interaction surface of a device, in accordance with one embodiment. The pattern-based gesture components are **2802** and **2804**.

[00575] As depicted, gesture **2806** is represented by a state transition matrix of touch states, with transitions between touch states marked by directional arrows, in accordance with one embodiment. In the context of the present description, a state transition matrix refers to an iconic depiction of a path through a phase space.

[00576] As shown, **2806** represents the phase space made up of all nine possible combinations of the three possible front and back touch states. In one embodiment, these nine combinations may be referred to as touch state points. In another embodiment, the representation may also be shown as three-dimensional, with time as the third dimension.

[00577] It should be noted that the use of descriptors such as "front" and "back" in this example are for the sake of simplicity. In various embodiments, gesture **2806** may be performed using any two surfaces on a device, even if the device does not have distinct front and back surfaces. Additionally, in other embodiments, pattern-based generalized gestures such as **2806** may include any number of additional dimensions, each describing a form of user input and/or time.

[00578] In various embodiments, a pattern-based generalized gesture may be represented as an ordered sequence of n-tuples. For example, in one embodiment, the gesture **2806** may be represented as a sequence of 2-tuples, (front touch state, back touch state). Specifically, gesture **2806** may be represented as {(s1,s0), (s2,s2),(s0,s1)}. In another embodiment, gesture **2806** may be represented as a sequence of 3-tuples, of the form (front touch state, back touch state, time).

[00579] While exemplary gesture **2806** is based on a path through a touch state phase space, pattern-based gestures may also be based upon types and sources of user input which are not broken into discrete states (i.e. contact pressure level, etc.). Additionally, in some embodiments, pattern-based gestures may be defined in terms of the relative states of input, rather than exact. For example, in one embodiment, a pattern-based gesture may be defined in terms of relative pressure. In other words, rather than require that a fixed pressure be achieved, a user may simply cause a particular change in pressure to occur. In one embodiment, such a pattern-based gesture may be defined in terms of gesture attributes, including derived input such as pressure velocity.

[00580] In various embodiments, the use of pattern-based gestures based upon relative states may be used to trigger some form of user assistance. These patterns are well suited to trigger some form of user assistance, as they may be performed and recognized in the middle of performing a different gesture. For example, in one embodiment, a pattern-based gesture defined as a rapid 20% decrease and increase of contact pressure of any contact, done three times, may present to the user an interface providing help specific to what the user was doing at the time. Possible contexts include, but are not limited to, providing a list of possible gestures when help is requested during performance of a gesture, application help or manual depending on what application is active, a detailed explanation of a user interface control which was being selected when help was requested, and/or any other scenario where a user is operating the device.

[00581] In some embodiments, pattern-based gestures may be recognized in the form of an ordered sequence of tuples using a decision tree. The use of a decision tree, or similar form of logic, has the advantage of being computationally inexpensive. This method may be more appropriate for pattern-based gestures which are represented by paths through an easily enumerated phase space. For example, this method of gesture recognition may be more appropriate for a gesture like **2806** than a similar gesture based upon contact pressure levels, rather than touch states.

[00582] In other embodiments, the pattern-based gestures may be recognized using pattern recognition or pattern matching. For example, in one embodiment, pattern-based gestures may be recognized using the methods, or similar methods, as those used to determine a gesture shape in a gesture component. In various embodiments, the pattern-matching algorithms and methods may search for a match within a predefined margin of error.

[00583] In one embodiment, pattern recognition or pattern matching to identify pattern-based gestures may be performed by the device. As an option, this functionality may be part of the operating system. In another embodiment, the pattern recognition or pattern matching may be performed using compute resources other than those of the device. Alternative compute resources include, but are not limited to, a similar device,

computing resources on a local network, cloud computing resources, and/or any other computing resource. In yet another embodiment, the pattern recognition or pattern matching may be performed using device and/or alternative computing resources combined with help from the user.

[00584] In one embodiment, pattern recognition algorithms and methods may search for a match between input and predefined patterns, within a certain probability. In another embodiment, the pattern recognition algorithms may be probabilistic and use statistical inference to find the best pattern match for a given set of input.

[00585] In one embodiment, probabilistic pattern recognition algorithms may output a probability that the input being described by a given pattern. In one embodiment, this probability may be reported to the user, either specifically (i.e. displayed as a number or percentage, etc.) or in summary (i.e. a label, a color coded border, an icon, etc.). In another embodiment, this probability may be used to alter or modify a selection. In yet another embodiment, this probability may be used to alter or modify one or more display properties.

[00586] In various embodiments, probabilistic pattern recognition algorithms may also determine the n best matches for the given input. In one embodiment, a user may specify the value of n. In another embodiment, these best matches may be reported to the user as a list of gesture names and/or iconic representations of the gestures. As an option, the associated probabilities may be reported as well.

[00587] In another embodiment, this list of probable gestures may be displayed to the user in response to a user input, such as a gesture associated with user assistance. In yet another embodiment, this list of probable gestures may be displayed to the user if the recognition algorithms were not able to recognize a gesture with a probability above a predefined threshold. The user may be able to select the intended gesture from the list; upon selection, the associated operation may be performed. As an option, one or more aspects of the list items may depend upon the associated probability of recognition. Possible aspects may include, but are not limited to, text size, button size, color, and/or any other aspect or property that may be associated with the list.

[00588] As a specific example, in one embodiment, the user may be typing on a soft keyboard and performs a gesture. The device then determines with probability 80% the gesture corresponds to pressing the letter W, with probability 15% the gesture corresponds to pressing Q, and with probability 5% that the user intended to press E. In this situation, the device may enlarge the W key by a large amount, enlarge the W key by a smaller amount, and enlarge the Q key by a still smaller amount.

[00589] In addition to gesture recognition, various embodiments may also perform user intent prediction. In the context of the present description, gesture prediction refers to a process of estimating user intent by assigning probabilities to candidate gestures based upon historical and real time use data. For example, in one embodiment,

[00590] In various embodiments, the process of gesture recognition may include the computation of the probability of user intent. In the context of the present description, computation of the probability of user intent refers to the process of assigning probabilities to potential user actions, inputs, and intentions, based upon historical and current use data. Examples of historical data include, but are not limited to, often misspelled words, actions consistently followed by a correction, utilization of particular gestures in different use scenarios, and/or any other data which may establish a use pattern.

[00591] The estimation of user intent may enhance the user experience by making the device more responsive to user input. In various embodiments, computation of the probability of user intent may facilitate the entry of text though an interaction surface. For example, in one embodiment, frequently made spelling mistakes may be automatically corrected. In another embodiment, the user may be prompted with the word determined to be most likely to be what the user is intending to type.

[00592] As an option, the user may be able to accept or decline proposed word completions or proposed spelling corrections through a pressure-based gesture. For example, in one embodiment, a user may accept or decline a proposed completion or correction with a series of taps on a backtouch interface (i.e. one tap for accept, two taps for decline), in addition to keyboard based methods. In another embodiment, a user may

be presented with a plurality of possible words to be used for auto completion. As an option, these words may be ordered depending upon a probability based upon historical usage. The words may be presented to the user in a line, with one word highlighted as a selection. In another embodiment, a user may cycle through proposed words with a backtouch gesture, such as a single finger swipe in the direction the words should be cycled in. Of course, these actions may be associated with any other type of gesture, including pattern-based gestures.

[00593] In various embodiments, gesture recognition may be enhanced by computation of the probability of user intent. For example, in one embodiment, probabilities may be assigned to potential gestures based upon historical gesture usage with similar context (i.e. application, selection mode, etc.). In another embodiment, estimated user intent may give weight to certain gestures during the recognition process. In this way, the gesture recognition process may be performed more efficiently, by considering the most likely gestures early in the process.

[00594] In various embodiments, the computation of the probability of user intent may affect the user interface. In various embodiments, the user interface may adapt in response to the user's intended form of interaction. For example, in one embodiment, upon determining that the probability that the user is performing a virtual control knob gesture has exceeded a predefined probability, and certain context requirements have been satisfied, a virtual control knob may appear for the user to interact with. In this way, the user is given visual feedback that their input is being received in the manner they intended. Examples of context requirements may include, but are not limited to, the user performing the gesture in the vicinity of a user interface object which may be controlled via a virtual control knob (i.e. slider, date, time, etc.), the virtual control knob gesture has not been reassigned to a different function by the active application, and/or any other requirement to ensure that a virtual control knob only appears when the user so intends.

[00595] Pattern-based gestures may be simple to perform and easy for a user to remember. For this reason, they are well suited to a number of applications. One of the

most basic pattern-based gestures is the finger press. In various embodiments, a pattern-based gesture made up of one or more fluctuations of contact pressure between a low level and a higher level at a single contact point may be associated with often used operations. For example, in one embodiment, a double press (i.e. increase then decrease contact pressure twice) may activate an interface to switch between active applications in a multitasking environment. In another embodiment, a triple press may activate a pop-up menu populated with iconic representations of the user's preferred actions, contacts, applications, and scripts. In one embodiment, these gestures may be surface agnostic. For example, the application switching interface may be activated no matter which interaction surface received the double press. In another embodiment, these gestures may be surface specific, allowing other system processes and applications to assign operations to these simple gestures without colliding with a system-wide gesture. For example, a triple press on a side surface may activate a pop-up favorites menu, while a triple press on a back surface may activate an application-specific contextual menu.

[00596] Another simple pattern-based gesture is the pinch gesture. In various embodiments, this gesture is performed by making simultaneous contact with two different interaction surfaces such a line connecting the two contact points is roughly parallel to the force vectors being applied at said points. An example is pinching a device between a finger and a thumb, the finger and thumb contacting different interaction surfaces. In another embodiment, this gesture may be pressure based. Variations on this gesture may incorporate multiple pinches (i.e. fluctuations of contact pressure between a low level and a higher level, similar to the finger presses previously described).

[00597] These simple gestures may be associated with often used operations, including but not limited to, activating device interfaces (i.e. backtouch interface, proximity sensitivity, pressure sensitivity, etc.), displaying a common system interface (i.e. application launcher, system settings, etc.), displaying a menu of user defined "favorites" (i.e. applications, contacts, songs, movies, web bookmarks, recently used documents, etc.), mute volume, and/or any other often used interface or operation. Of course, in other embodiments, these gestures may be associated with any operation, function, or interface.

[00598] Building off the previous gesture, a pinch slide gesture is an example of a pattern-based generalized gesture which combines a pinch with a sliding motion. In some embodiments, it may be described as a 1-6 gesture, though it is not limited to just front/back interaction. In various embodiments, this gesture may be performed by pinching the device, then sliding the pinch across the interaction surfaces while maintaining contact. In one embodiment, a contact pressure above a predefined threshold may also be required. This gesture mimics the tactile experience of pushing or pulling an object such as a lever or cord.

[00599] In one embodiment, a pinch slide gesture may be used to quickly move through an alphabetical list, similar to using tabs to quickly find a contact quickly in a paper address book. In another embodiment, this operation may be adopted to moving quickly through other sorted lists of items, including but not limited to song lists, application lists, eBooks, and/or any other collection of organized data. As an option, the display may invite the use of this gesture by displaying virtual tabs (i.e. the alphabet, numbers, etc.) along a side of the display where the gesture would be performed.

[00600] In another embodiment, a pinch slide gesture may be used to quickly select text. For example, in one embodiment, a selection may be initiated by pinching at the desired location. The selection may be expanded by sliding the pinch to the desired selection end. In another embodiment, this method of selection may be adapted to selecting a portion of an image or graphic (i.e. pinch at one corner of desired selection bounding box and sliding to opposite corner, etc.).

[00601] In another embodiment, a pinch slide may be used to move quickly through a large document (i.e. word processing document, web page, image, etc.). For example, in one embodiment, the user may slide a pinch up and down the device, changing the displayed portion of the document relative to the position of the pinch (i.e. the top of the device represents the start of the document, the bottom of the device represents the end, and the relative location of the pinch becomes the portion of the document displayed. In another embodiment, the pinch slide may have to be initiated at a particular location to activate this document navigation functionality. For example, a web browser may have

an icon near the top of the display that activates pinch slide navigation when the gesture is initiated on the icon. This functionality may be adapted to two dimensional navigation in a large image, displaying the portion of the image represented by the relative location of the pinch.

[00602] In yet another embodiment, a pinch slide gesture may be used to display menus of common operations or applications. For example, in one embodiment, a pinch slide gesture which begins near an edge of a display and then moves inward may cause a menu or window to slide in from that edge of the display. As an option, the menu or window may remain visible until a selection is made. In another embodiment, there may be different menus or windows associated with each edge of a display. In still another embodiment, a window may be closed (i.e. slid back out of view) by using a pinch slide to pull it out a bit farther, then releasing it, similar to rolling up a window shade.

[00603] The pinch slide gesture is well adapted for providing input where one of a plurality of values may be chosen; it is also easily combined with a pinch gesture. For example, in one embodiment, a user may activate a system and/or ringer volume slider with a double press pinch gesture, and then select a volume by sliding the pinch up and down the device. In another embodiment, a similar operation may be performed for backlight brightness. In yet another embodiment, a user may cycle between system volume, ringer volume, and system brightness sliders with a pinch gesture, and be able to interact with said sliders using a pinch slide. In still another embodiment, the selection of system/ringer volume or brightness may be determined by where on the device the pinch slide is being performed (i.e. left, middle, right, etc.).

[00604] A subset of pinch gestures are the side pinch gestures. In various embodiments, the side pinch gesture is performed by applying contact pressure beyond a threshold value to opposite sides of the device, the sides being those other than the front and back. As an option, the user may choose the predefined threshold value. In one embodiment, the pressure may be applied at contact points located on the sides such that a line connecting the contact points is roughly normal to the sides. In another

embodiment, the side pinch gesture may also incorporate "presses" similar to the pinch and finger press gestures.

[00605] The side pinch gesture is a simple gesture well adapted for use with often used operations. For example, in one embodiment, a side pinch gesture may toggle the device mute, or activate an application launcher.

[00606] A variation on the side pinch gesture is the squeeze gesture. In various embodiments, a squeeze gesture is performed by applying contact pressure beyond a predefined threshold to one or more contact points located on opposite sides of the device. For example, in one embodiment, the gesture may be performed by holding the device in one hand, in a natural grip, and applying pressure to two opposite sides by squeezing. As an option, the squeeze gesture may incorporate "presses", similar to the finger press and pinch gestures.

[00607] The squeeze gesture is one of the easiest gestures to perform, and may be reserved for operations which are to be available system wide, independent of what application is running. In various embodiments, the squeeze gesture may be associated with commonly used operations. For example, in one embodiment, a double press squeeze gesture may be used to answer a phone call. In another embodiment, a double press squeeze gesture may be used to indicate an affirmative response to any on screen query (i.e. accept proposed auto correction, accept proposed auto completion, activate "ok" button in dialog boxes, etc.). As an option, a complementary negative response gesture may be defined as well, a gesture such as the combination of a single or double tap on the top side while squeezing the left and right sides.

[00608] In another embodiment, applying a squeeze gesture for a predefined amount of time may be associated with a delete operation, to be performed upon any selected or active object. As an option, the delete operation may be accompanied by a "crumpling paper" animation routine, as well as a prompt for user confirmation. In one embodiment, a user may not use a squeeze gesture to confirm an operation triggered by a squeeze operation (i.e. to prevent accidental implementation, etc.).

[00609] Another class of pattern-based gestures are the "no contact" pressure-based gestures. In some embodiments, the touch sensitive interaction surfaces require an implement of interaction which has a certain degree of conductivity in order to register contact. This class of gestures may be implemented in such devices, which are also pressure-sensitive. In various embodiments, the "no contact" pressure-based gestures may be used to interact with the device in situations where direct contact is not possible (i.e. device is in case, device is in pocket, device is in purse, etc.).

[00610] In various embodiments, the "no contact" pressure-based gestures may include variations of the finger press, pinch, and squeeze gestures, in addition to others. In some embodiments, "no contact" pressure-based gestures may require that the pressures being exerted do not create a contact point (i.e. pressure is exerted, but touch contact is not registered). These gestures are well adapted to situations where contact is not feasible. For example, in one embodiment, a user may silence a ringing phone by performing a rapid, one-finger double press on an interaction surface while the device is in the user's pocket, purse, or any other flexible or semi-flexible vessel. In another embodiment, a user may perform a predefined pattern-based gesture to cause the most recently received communication (i.e. voice mail, email, SMS message, etc.) to be reported to the user through a Bluetooth audio device.

[00611] In various embodiments, gestures may be used to trigger a variety of activities. For example, Figure **29** shows a file sharing operation which was triggered, in part, by a gesture. Figure **30** shows the authentication of two devices through a touch interaction. Figure **31** shows another example of two devices working together. It depicts facial recognition being performed by two devices at the same time. Additionally, Figure **32** shows the displays of two devices combined to function as a single display. Any of the embodiments discussed in the present description may be implemented in any of these environments.

[00612] Figure **33** shows a user interface **3300** for defining custom gestures, in accordance with one embodiment. As an option, the user interface **3300** may be implemented in the context of the architecture and environment of the previous Figures

or any subsequent Figure(s). Of course, however, the user interface **3300** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00613] In various embodiments, gestures may be programmed by the user. In other embodiments, gestures may be learned by the device.

[00614] As shown, the user interface **3300** may be utilized to define a custom gesture. For example, in one embodiment, the user interface may include a gesture name and/or an iconic representation of the gesture. In another embodiment, the iconic representation may be automatically chosen by the system, depending upon defined gesture parameters. In yet another embodiment, the user interface may contain a plurality of buttons **3302** which allow a user to save, load, copy, and/or delete custom gestures. The user interface may also contain a list **3304** of all defined custom gestures. As an option, the custom gestures may be grouped by context within the list.

[00615] In various embodiments, the user interface **3300** may contain a button **3306** to allow the user to define a custom gesture. In one embodiment, selecting button **3306** may initiate a gesture recording session. In the context of the present description, a gesture recording session refers to a period of time wherein the device monitors and records all observable user and system input. In another embodiment, selecting button **3306** may present to the user an interface with instructions before initiating a gesture recording session. Instructions may include, but are not limited to, how to record the gesture, a list of all forms of input upon which a gesture may depend, how to end the recording, and/or any other information which may assist the user. In yet another embodiment, the user may be presented with a short countdown before the gesture recording session is initiated, to give the user time to prepare for the recording. As an option, the device may give some form of visual or auditory feedback to the user to indicate that the recording session has begun.

[00616] In various embodiments, a gesture recording session may be terminated by user input. For example, in one embodiment, a gesture recording session may be terminated by the user through a vocal command, recognized by the system. In another

embodiment, a gesture recording session may be terminated by the user through an sound which exceeds a predefined volume. In yet another embodiment, a gesture recording session may be terminated by the user by performing a gesture. As an option, the termination gesture may be a reserved gesture.

[00617] In the context of the present description, a reserved gesture refers to a gesture which may not be reassigned by a user, application, or process. For example, in one embodiment, there may be a set of gestures, chosen for their simplicity, which are reserved for operations fundamental to the user experience (i.e. a gesture to cycle through active applications running in a multitasking environment, etc.).

[00618] In still another embodiment, a gesture recording session may be terminated automatically after a predefined amount of time has elapsed. As an option, in one embodiment the user may specify the amount of time before the recording session begins.

[00619] In one embodiment, after the termination of a gesture recording session, the user may be prompted to repeat the gesture one or more times. As an option, the user may not have to indicate when the gesture is complete after the first recording session. In another embodiment, the repeated performance of the gesture may be used to better define the gesture parameters. For example, in one embodiment, gesture aspects which do not appear in every performance of the gesture may be indicated to the user as potentially irrelevant. In another embodiment, the system may use each iteration to improve the recorded description of the gesture. For example, the system may calculate the probability of correct recognition, using methods previously discussed, after each performance. As an option, the user may be prompted to repeat the gesture until the calculated probability of correct recognition has exceeded a predefined threshold.

[00620] In various embodiments, the user interface **3300** may include a button **3308** to allow a user to view a depiction of the performance of the custom gesture as currently defined. For example, in one embodiment, selecting button **3308** may display an animated recreation of the gesture in the context of a wireframe model of the device. In another embodiment, different forms of user input may be represented by different colors (i.e. pressure is red, touch is blue, proximity is green, etc.). As an option, changes in

contact pressure level may be represented by changes in the color representing pressure. In another embodiment, the device surfaces may be displayed flattened into a two dimensional plane. In still another embodiment, the gesture replay may also include one or more plots which depict the time evolution of various gesture inputs, such as contact pressure or proximity associated with individual contacts or implements of interaction.

[00621] In various embodiments, the user interface **3300** may include a list **3310** displaying all components of the currently defined gesture. For example, in one embodiment, list **3310** may be a nested hierarchical list, grouped by "expected implement of interaction" and input type (i.e. contact 1, contact 2, proximity 1, etc.). In another embodiment, list **3310** may visually indicate (i.e. italicized, colored, underlined, etc.) recorded gesture aspects which were determined to be potentially irrelevant to the gesture. In yet another embodiment, list **3310** may break down the input from each "expected implement of interaction" into basic input categories. Categories may include, but are not limited to, pressure, touch, surface identity, proximity, proximity location, and/or any other aspect or type of user input.

[00622] The user interface **3300** may be utilized to specifically define or modify aspects of a custom gesture. For example, in one embodiment, the user interface **3300** may include a button **3312** to edit a gesture aspect selected in list **3310**. In another embodiment, selecting button **3312** may present an interface to the user which may allow them to review and alter associated gesture aspects. This method of defining the gesture may not be as intuitive as teaching the system the gesture through repeated performance, but it does allow much greater control over the gesture aspects.

[00623] In one embodiment, the user interface **3300** may include a plurality of buttons **3314** which allow a user to copy, create, and delete gesture aspects associated with the current gesture. In another embodiment, the creation of a new gesture aspect may prompt the user to specify the input type to create, and with which "expected implement of interaction" it should be associated. As an option, a user may create a new "expected implement of interaction".

[00624] In one embodiment, the user interface **3300** may include a collection of check boxes **3316** which allow a user to specify the context in which the custom gesture may be performed. The contexts may include, but are not limited to, universal (i.e. available at all times), text (i.e. active text field, on-screen keyboard visible, etc.), application-specific, home (i.e. application chooser, etc.), selection (i.e. an active selection of an object exists), and/or any other type of context.

[00625] When defining a custom gesture, the associated action may be established. As shown, user interface **3300** may be utilized to specify and configure the action to be associated with the present gesture. For example, in one embodiment, the user interface may include a collection of checkboxes **3318** representing different actions which may be associated with the custom gesture.

[00626] In one embodiment, the collection of actions **3318** may include a checkbox **3320** to indicate that the gesture may trigger the insertion of a text snippet, as defined in a nearby text field. In another embodiment, the user interface may include a collection of buttons **3322** which may be used to format the text snippet. As an option, the text snippet may be defined as plain text, without formatting.

[00627] In another embodiment, the collection of actions **3318** may include a checkbox **3324** to indicate that the gesture may cause an application to launch. The user interface **3300** may also include a button which allows the user to specify which application should be launched when the gesture is performed.

[00628] In another embodiment, the collection of actions **3318** may include a checkbox **3326** to indicate that the gesture may set the system volume to a specified level. As an option, the user may be able to specify whether the gesture sets the system volume or a ringer volume. The user interface **3300** may also include a slider which allows the user to specify the volume to be associated with the present gesture.

[00629] In another embodiment, the collection of actions **3318** may include a checkbox **3328** to indicate that the gesture may trigger the creation of a new email message. In yet another embodiment, the user interface **3300** may include a checkbox

3330 to indicate, if the gesture is performed while an object is selected, whether the selected object should be inserted into the new email message. Possible objects may include, but are not limited to, images, text, video, files, and/or any other selectable media, file, or other kind of object. In still another embodiment, the user interface **3300** may also include a checkbox **3332** to indicate that the new email message should be addressed to a particular contact. As an option, the user may be able to choose the email address from a list of the user's contacts.

[00630] In another embodiment, the collection of actions **3318** may include a checkbox **3334** to indicate that the gesture may trigger the creation of a new SMS message. In yet another embodiment, the user interface **3300** may include checkboxes **3336** and **3338** to indicate the inclusion of a selected object and/or specific destination for the message, similar to the "new email message" action previously described. As an option, if a specific recipient is not specified, the new SMS message may be directed to the most recent SMS message sender.

[00631] In another embodiment, the collection of actions **3318** may include a checkbox **3340** to indicate that the gesture may take advantage of functionality provided by third party applications. In various embodiments, developers of third party applications may be able to provide functionality hooks that may be implemented by other applications or the operating system. For example, in one embodiment, a social networking application may allow the user to associate the custom gesture with the creation of a new posting within the social network.

[00632] In other embodiments, possible actions which may be associated with a user defined gesture may include, but are not limited to, a favorite contact (i.e. gesture displays a pop-up interface with buttons representing different methods of communicating with the contact, including but not limited to telephone, email, SMS, social networks, etc.), a predefine energy saving profile, a predefined set of application preferences, and/or any often repeated activity.

[00633] The user interface **3300** may be utilized to determine whether the custom gesture may conflict with already established gestures. For example, in various

embodiments, the user interface may include a list **3342** of conflicting gestures. In one embodiment, the list of conflicts may be made up of other custom gestures, as well as system defined gestures. In another embodiment, the list may also include application specific gestures.

[00634] In yet another embodiment, items in the conflict list may be formatted to indicate their severity. For example, conflicting reserved gestures may be highlighted in red, while conflicting gestures which only exist in a single application may be slightly greyed out. In various embodiments, the conflict list may be created using pattern recognition methods previously discussed, or other methods and algorithms capable of matching patterns and/or data.

[00635] The user interface **3300** may be utilized to test a custom gesture, in accordance with one embodiment. For example, in various embodiments, the user interface may include a button **3344** which presents the user with an interface which allows them to test the custom gesture as it is currently defined. In one embodiment, the gesture testing interface may provide the user with a plurality of forms of feedback. Feedback may include, but is not limited to, current contact pressure level associated with each contact point, current touch state associated with each contact point, probability of correct gesture recognition, current gesture time, a visual representation of previous contact points, and/or any other form of user feedback.

[00636] In one embodiment, user interface **3300** may include statistics collected as a result of the gesture testing process. These statistics may include, but are not limited to, number of tests run with the current gesture configuration, average probability of correct gesture recognition, best probability of correct gesture recognition, identity of next most likely identified gesture (i.e. next closest gesture), minimum gesture performance time, maximum gesture performance time, average gesture performance time, and/or any other statistic related to the testing process.

[00637] In some embodiments, user defined gestures may be recognized in any context they were intended for. In other embodiments, a user may have to perform a predefined activation gesture before using a custom gesture.

[00638] As a specific example of how custom gestures may be utilized, in one embodiment, a user may define a gesture which is comprised of a triple finger press on a top side, followed by a singer finger swipe across a back interaction surface. The user may associate this gesture with a favorite contact. Upon performing this gesture, the user may be presented with a window showing a picture of the contact in addition to buttons linked to their telephone number and email address. In this way, communication may be established with this contact, at any time, with nothing more than a gesture and a button press.

[00639] Figure **34** shows a user interface **3400** for defining pressure inputs for a custom gesture, in accordance with one embodiment. As an option, the user interface **3400** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **3400** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00640] As shown, the user interface **3400** may be utilized to configure aspects of a pressure-based gesture component. In one embodiment, user interface **3400** may be displayed in response to a user selecting a pressure item in list **3310** and selecting the edit button **3312**, or in response to creating a new pressure element, within user interface **3300.**

[00641] The user interface **3400** may be utilized to define the gesture component in terms of contact pressure levels. For example, in one embodiment, the user interface **3400** may include a collection of check boxes **3402** which allow a user to require upper and lower contact pressure level thresholds. In another embodiment, each check box may be associated with a button **3404** which allows a user to specify a contact pressure level by exerting the desired pressure. In still another embodiment, the contact pressure levels may be entered numerically.

[00642] In one embodiment, the user interface **3400** may include a collection of check boxes **3406** that allow a user to require threshold pressure velocity and/or pressure acceleration thresholds. In another embodiment, each check box may be associated with

a button **3408** which allows a user to specify a contact pressure velocity or acceleration by example. In still another embodiment, the contact pressure velocity and/or acceleration may be entered numerically.

[00643] In various embodiments, a user may define a contact pressure level, velocity, and/or acceleration by example through a user interface. For example, in one embodiment, the user may be presented with an interface which provides continuous feedback regarding the contact pressure being exerted. In another embodiment, the user may be able to indicate that the previously exerted contact pressure level, velocity, and/or acceleration is satisfactory and should be used. In yet another embodiment, the user may be prompted to repeat the pressure exertion one or more times, and the resulting average results are used. In various embodiments, this interface may be shown to the user in response to selecting buttons **3404** or **3408**.

[00644] The user interface **3400** may be utilized to define the gesture component in terms of touch states. For example, in one embodiment, the user interface **3400** may include a check box **3410** which allows a user to require a target touch state. In another embodiment, this check box may be associated with a button **3412** which allows a user to specify a touch state by example. As an option, this may be done using an interface similar to that used to define contact pressure levels, velocities, and/or accelerations. In still another embodiment, the touch state may be entered directly.

[00645] The user interface **3400** may be utilized to define the gesture component in terms of a pattern-based pressure gesture. For example, in one embodiment, the user interface **3400** may include a check box **3414** which allows a user to require the performance of a pattern-based pressure gesture. In another embodiment, this check box may be associated with a button **3416** which allows a user to specify a pattern-based pressure gesture by example. In one embodiment, this may be done using an interface similar to that used to define contact pressure levels, velocities, and/or accelerations. As an option, this interface may plot the touch states that make up the pattern as it is being defined.

[00646] In one embodiment, the user interface **3400** may include a plot **3418** which displays the currently specified pattern-based pressure gesture, plotting touch states versus time. In another embodiment, the touch state transitions **3420** displayed on plot **3418** may be able to be modified. In still another embodiment, user interface **3400** may include buttons which may be used to create and delete touch state transition markers on plot **3418**.

[00647] In another embodiment, the user interface may include a checkbox **3422** which allows a user to specify whether the pattern-based pressure gesture shown in plot **3418** is time dependent or not. As an option, selection of checkbox **3422** may cause the time axis of plot **3418** to display the actual times associated with each touch state transition.

[00648] In one embodiment, the user interface **3400** may include a button **3424** which allows a user to test the pressure-based component gesture as it is presently defined. In another embodiment, selecting button **3424** may display a testing interface to the user. In various embodiments, the testing interface may provide the user with different forms of feedback. Feedback may include, but is not limited to, current contact pressure level and/or touch state, gesture time, probability of correct recognition, and/or any other form of feedback related to performance of a gesture.

[00649] Figure **35** shows a user interface **3500** for defining touch inputs for a custom gesture, in accordance with one embodiment. As an option, the user interface **3500** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **3500** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00650] As shown, the user interface **3500** may be utilized to configure aspects of a touch-based gesture component. In one embodiment, user interface **3500** may be displayed in response to a user selecting a touch item in list **3310** and selecting the edit button **3312**, or in response to creating a new touch element, within user interface **3300.**

[00651] The user interface **3500** may be utilized to define the gesture component in terms of a path of contact points. For example, in one embodiment, user interface **3500** may include a button **3502** which allows a user to define a gesture touch path by performing the gesture. In another embodiment, user interface **3500** may also include a window **3504** where the user may draw a new gesture shape, or manipulate one already defined.

[00652] In one embodiment, user interface **3500** may also include a button **3506** which allows the user to test the defined touch gesture by performing it. As an option, the user may be provided with feedback regarding the performance, feedback which may include, but is not limited to, probability of recognition.

[00653] Figure **36** shows a user interface **3600** for defining surface inputs for a custom gesture, in accordance with one embodiment. As an option, the user interface **3600** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **3600** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00654] As shown, the user interface **3600** may be utilized to select an interaction surface for a gesture component. In one embodiment, user interface **3600** may be displayed in response to a user selecting a surface item in list **3310** and selecting the edit button **3312**, or in response to creating a new surface element, within user interface **3300.**

[00655] The user interface **3600** may be utilized to define the gesture component in terms of a device surface. For example, in one embodiment, user interface **3600** may include a button **3602** which allows a user to specify a surface by giving it three sharp taps with a single finger (or some other gesture). In another embodiment, the user interface may also include a three dimensional representation of the device in a window **3604**, upon which a user may indicate a surface.

[00656] Figure **37** shows a user interface **3700** for summarizing contact inputs for a custom gesture, in accordance with one embodiment. As an option, the user interface

3700 may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **3700** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00657] As shown, the user interface **3700** may be utilized to configure summary aspects of a contact-based gesture component. In one embodiment, user interface **3700** may be displayed in response to a user selecting a contact summary item in list **3310** and selecting the edit button **3312**, or in response to creating a new contact summary, within user interface **3300**.

[00658] The user interface **3700** may be utilized to define how the contact related aspects of a gesture component may combine with each other. For example, in one embodiment, user interface **3700** may include a window **3702** which contains a depiction of the touch path **3704** defined in user interface **3500**. In another embodiment, touch path **3704** may include pressure points **3706** which represent pressure states at different time intervals. As an option, a user may drag the pressure points to different locations on the touch path, indicating the pressure level which may be achieved at those points of the touch path.

[00659] In various other embodiments, there are user interfaces for defining proximity based gesture components, which are similar in many respects to the interfaces used to define the role of contact pressure and touch

[00660] Figure **38** shows a user interface **3800** for defining system inputs for a custom gesture, in accordance with one embodiment. As an option, the user interface **3800** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **3800** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00661] As shown, the user interface **3800** may be utilized to configure system aspects of a generalized gesture. In one embodiment, user interface **3800** may be displayed in

response to a user selecting a system item in list **3310** and selecting the edit button **3312**, or in response to creating a new system element, within user interface **3300**.

[00662] The user interface **3800** may be utilized to define the gesture component in terms of system related variables or inputs. For example, in one embodiment, user interface **3800** may include a plurality of checkboxes **3802** associated with various system inputs including, but not limited to, device orientation, device charging status, presence of headphones, and/or any other system related input or variable. In another embodiment, the user interface **3800** may also include a button **3804** which allows the user to define a target orientation by placing the device in the desired orientation until given an indication (i.e. sound, flash, etc.) that the orientation has been captured.

[00663] Figure **39** shows a pressure-sensitive user interface **3900** which contains a virtual trackball, in accordance with one embodiment. As an option, the user interface **3900** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **3900** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00664] As shown, the user interface **3900** includes a virtual trackball **3902**, which is receptive to user input, in accordance with one embodiment. In various embodiments, user interface **3900** may simulate one or more properties, attributes, or behaviors associated with the operation of a real, three-dimensional trackball user input device.

[00665] For example, in one embodiment, virtual trackball **3902** may simulate the operation of a real trackball by simulating the relationship between applied force and trackball rotation observed in a real trackball.

[00666] In different embodiments, various techniques may be employed to exploit the relationship between applied force and ball rotation exhibited in the operation of a real trackball. The force that causes a real 3D trackball to spin is the component of force applied which is in a direction tangential to the trackball surface. A user typically initially touches a real trackball in the center of the exposed trackball surface. The user

then rotates the trackball by moving a finger horizontally at the desired angle of movement. As the finger moves further away from the initial point of contact with the trackball, the less the trackball may rotate with a given horizontal force FT applied by the finger. This behavior is a result of the gradual reduction of the component of force that moves the trackball as the finger moves a distance D from the center of the trackball. If the trackball radius is RT, and the force applied by the finger in a horizontal direction FT, analysis may show that the force that moves the trackball is FT when D = 0 and 0 when D = RT. Thus, to the user, the rotation of the trackball is reduced (i.e. the force of a finger becomes less effective) as the distance from the center of the trackball increases. To this end, the operation of a real trackball may be simulated in a 2D environment by recreating the relationship between rotation efficacy and distance from the center of the trackball.

[00667] In various embodiments, this effect may be simulated on a 2D screen by applying an input function IF to the inputs. In the context of the present description, an input function refers to a function which modifies one or more properties of one or more input signals.

[00668] In one embodiment, the device may display a virtual trackball on a 2D screen at location (X1, Y1). The force FT that the user wishes to apply may be calculated as a function of contact pressure P(t) and/or gesture velocity. The force may thus be calculated by the device as FT(P(t), X(t), Y(t), etc.). Thus, the function FT may be a first input function IF1 = FT. In order to simulate the behavior of a spherical trackball, a second input function IF2 may be applied. For example, in one embodiment, IF2 may equal (1 – D/RT) for D < RT, and equal 0 otherwise. In various other embodiments, other real objects may be simulated in a 2D environment through the use of one or more input functions.

[00669] Figure **40** shows a device **4000** with slide pads, in accordance with one embodiment. As an option, the device **4000** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of

course, however, the device **4000** may be carried out in any desired environment. The aforementioned definitions may apply during the present description.

[00670] In the context of the present description, a slide pad refers to a type of interaction surface which does not cover the majority of a device face, and which is at least touch-sensitive. In one embodiment, a slide pad may be non-rectangular. In another embodiment, a slide pad may have more than one texture, to provide tactile feedback to a user. In various embodiments, a device may have more than one slide pad on a face.

[00671] As shown, device **4000** includes two slide pads **4002** on the back face **4004**. In other embodiments, one or more slide pads may be located on one or more device faces, including but not limited to side faces, flip-out portions of a device, slider portions, and/or any device face or facet.

[00672] In various embodiments, one or more slide pads may be used in conjunction with one or more displays that are not also interaction surfaces. In one embodiment, the use of slide pads in conjunction with non-interactive displays may provide a number of advantages. These advantages may include, but are not limited to, reduced expense, increased durability, and/or simplification of maintenance. In another embodiment, the user of slide pads may facilitate the incorporation of a non-planar or non-rigid display (i.e. flexible display, etc.). In yet another embodiment, slide pads may be replaced and/or upgraded to provide greater sensitivity to more forms of user input.

[00673] In various embodiments, the use of slide pads may provide an improved user experience. For example, in one embodiment, the size and modularity of a slide pad may make it possible to incorporate an apparatus to provide tactile feedback (i.e. vibration, etc.) to just that slide pad. In this way, a user may receive tactile feedback specific to a single hand, rather than vibrating an entire surface, for example.

[00674] In various embodiments, one or more slide pads may be used as interaction surfaces for the performance of any type or variation of gestures previously or subsequently described. In some embodiments, slide pads may have a geometry which

favors one particular type of touch gesture or contact (i.e. a narrow strip may favor sliding, a ring may favor circular gestures, etc.). In one embodiment, slide pads may be of any shape, contour (i.e. planar, rounded, convex, concave, etc.), and/or size.

[00675] In some embodiments, slide pads may not be limited to being touch-sensitive. For example, in one embodiment, a slide pad may also be sensitive to contact pressure and/or proximity. In some embodiments, a device may utilize multiple slide pads having a variety of sensitivities.

[00676] Figure **41** shows an exemplary slide pad gesture **4100** for performing an auxiliary function, in accordance with one embodiment. As an option, the gesture **4100** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the gesture **4100** may be carried out in any desired environment. It should also be noted that the aforementioned definitions may apply during the present description.

[00677] In the context of the present description, a slide pad gesture is a gesture adapted to the limitations of a slide pad, including but not limited to, limited geometry, limited size, limited sensitivity, and/or any other limitation which may be inherent in a slide pad.

[00678] In the context of the present description, an auxiliary function refers to a frequently used operation, command, or action. In one embodiment, any operation or command may be deemed an auxiliary function. In various embodiments, making auxiliary functions available through gestures may enhance the user experience by allowing the user to intuitively accomplish common tasks.

[00679] Making a distinction between slide pad gestures and other gestures, as well as incorporating slide pads in addition to other interaction surfaces, allows a device to provide a more intuitive user experience by increasing the number of distinct though simple gestures. For example, see Figure **41**.

[00680] As shown, slide pad gesture **4100** is a dynamic gesture based upon a single point of contact **4102** with a slide pad **4104** located on the back of a device, next to a

larger interaction surface **4106**. In various embodiments, this gesture may be associated with an auxiliary function which mimics the functionality of a shift and/or caps lock button on a keyboard. For example, in one embodiment, an upward slide may be the equivalent of holding down a shift key. As an option, this operation may cause a soft keyboard **4108** displayed on the front of the device to be populated with upper case letters and symbols. In another embodiment, a downward slide may be the equivalent of toggling a caps lock key. Assigning these auxiliary functions to these slide pad gestures allows a user, within the same interface, to swipe up on a backtouch interface to scroll through a document, and then swipe up on a slide pad to capitalize a soft keyboard also present on the display.

[00681] In one embodiment, a user may program the behavior of a slide pad. For example, in one embodiment, a user may specify that an upward gesture corresponds to entry of uppercase letters and a down gesture resumes input of lowercase letters. In another embodiment, a user may specify that two upward gestures in succession may correspond to a numeric or symbol entry mode, and two downward gestures in succession may correspond to a normal text entry mode.

[00682] In one embodiment, the auxiliary function associated with a slide bad gesture may depend on one or more contextual aspects, including but not limited to application environment, system flags, selection mode, and/or any other contextual data.

[00683] In one embodiment, a user may define custom slide pad gestures, through a user interface similar to that shown in Figure **33**. As an option, a user may associate a custom slide pad gesture with any auxiliary function, or any other device or system operation.

[00684] In various embodiments, slide pads may be located on the sides of a device, giving an inexpensive way to provide additional methods of one-handed device operation. For example, in one embodiment, a user may slide a thumb up and down a side-located slide pad to simulate the use of a thumb wheel. In another embodiment, the side mounted slide pad may span a significant portion of the device side, such that the virtual thumb wheel may be located wherever the user's thumb happens to contact the

slide pad. In yet another embodiment, the virtual thumb wheel may be activated through a gesture. In still another embodiment, the virtual thumb wheel may be activated by applying contact pressure with the thumb on the slide pad for a predetermined amount of time.

[00685] In various embodiments, a virtual thumb wheel may be used for different operations, depending upon the context of use. For example, in one embodiment, the virtual thumb wheel may be used to scroll up and down an active window, or cycle through items in a list (i.e. messages, files, applications, active applications in a multitasking environment, etc.). In another embodiment, a virtual thumb wheel may be used to adjust one or more device properties including, but not limited to, volume, backlight brightness, and/or any other device property. In yet another embodiment, a virtual thumb wheel may be used to move through a three-dimensional environment which may include, but is not limited to, 3D layered user interfaces, 3D rendered scenes, a 3D virtual organizational space, and/or any other three-dimensional environment.

[00686] In other embodiments, virtual scroll wheels similar to the virtual thumb wheel may be located on any type of interaction surface situated on any face of a device (the only practical difference being the digit used to operate them).

[00687] Figure **42** shows a user interface **4200** for providing quick access to favorite operations, in accordance with one embodiment. As an option, the user interface **4200** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **4200** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00688] As shown, the user interface **4200** may be utilized to provide easy, quick access to a plurality of favorite operations, applications, contacts, documents, and/or other types of objects. In various embodiments, favorites panel **4202** may be displayed in response to a gesture, the selection of a hardware or software button, voice command, accelerometer input (i.e. shaking the device, flicking the users wrist, etc.), and/or any

other form of user input. In one embodiment, once favorites panel **4202** is displayed, a user may select an item using a slide pad **4204** located on the side of the device.

[00689] In one embodiment, a user may configure which items are displayed in the favorites panel. In another embodiment, the favorites panel may be automatically populated by the device, based upon factors which may include, but are not limited to, historical usage, date of last use, frequency of use, popularity information obtained from a remote system, and/or any other factors.

[00690] Figure **43** shows a device **4300** with a non-uniform display, in accordance with one embodiment. As an option, the device **4300** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the device **4300** may be carried out in any desired environment. The aforementioned definitions may apply during the present description.

[00691] In the context of the present description, a non-uniform display refers to a display that is not homogeneous in one or more aspects. Possible aspects may include, but are not limited to, pixel size, pixel shape, resolution, color (i.e. color depth, color versus black and white or grayscale, etc.), brightness, contrast, intensity, display technology (i.e. e-ink, LCD, light emitting polymer, etc.), touch resolution, and/or any other aspect or property associated with a display. In one embodiment, a non-uniform display may be divided into portions, each portion having a different set of display properties. In another embodiment, a non-uniform display may be designed to match the characteristics of the human eye.

[00692] Further, in the context of the present description, an e-ink display refers to any display which is bi-stable. In the context of the present description, bi-stable refers to being capable of holding text or other rendered images even when very little or no power is supplied to the display. In one embodiment, the e-ink display may include a display that has a resolution 150 dpi or better. In another embodiment, the e-ink display may include any technology which may exhibit high contrast, or contrast substantially equal to that of print on paper. For example, the e-ink display may include displays such as bi-stable LCDs, MEMS, cholesteric, pigmented electrophoretic, E Ink-brand display, etc.

Further, in one embodiment, the e-ink display may display visual content in black-and-white or grayscale. In some embodiments, the e-ink display may display visual content in color.

[00693] In various embodiments of uniform displays, a non-negligible amount of resources (i.e. energy expenditure, device component cost, etc.) may be expended on display aspects that may be wasted due to limitations of human physiology, including visual acuity and touch sensitivity.

[00694] In different embodiments, various techniques may be employed to exploit the physiological limits of the human eye. The retina of the eye is approximately 0.5 mm thick and lines the back of the eye. The retina is a circular disc of about 30 mm in diameter and an area of about 700 mm^2. The macula is an oval-shaped highly pigmented yellow spot near the center of the retina. The central retina has a diameter of about 5 mm and an area of about 80 mm^2. There is a very small, high-resolution sensing area in the center of the central retina called the fovea (or fovea centralis). Beyond the central retina is the peripheral retina stretching to the ora serrata, about 21 mm from the fovea.

[00695] The fovea has a diameter of about 1 mm and an area of about 1 mm^2, or about 0.1% of the retina and 1% of the central retina. Approximately half of the nerve fibers in the optic nerve carry information from the fovea, while the remaining fibers carry information from the rest of the retina. The fovea is responsible for sharp central vision (foveal vision and foveal view), which is used for reading, watching television, driving, or any activity where visual detail is of primary importance. To this end, the distinction between sharp central vision and less detailed peripheral vision may be exploited by confining the use of high resolution displays to a smaller area, surrounded by lower resolution display elements, which may be cheaper and consume less power.

[00696] In different embodiments, various techniques may be employed to exploit all types of photoreceptor cells. Rod cells (rods) are photoreceptor cells in the retina that can function in less intense light than cones. Rods are concentrated at the outer edges of the retina and are used in peripheral vision. Rods are more sensitive than cone cells. Rods

are almost entirely responsible for night vision. There are about 100 million rods in the retina.

[00697] Cone cells (cones) are photoreceptor cells in the retina that are responsible for color vision. Cones function best in relatively bright light. Cone cells are densely packed in the fovea, with density decreasing towards the periphery of the retina. Cones are able to perceive finer detail and more rapid changes in images than rods because their response times to stimuli are faster. There are about 3 million cones in the retina, or about 3% of the number of rods.

[00698] Cones are specialized retinal cells containing pigments with different spectral sensitivities. The perception of color is a complex process that starts with the differential output of these cells in the retina and ends in the visual cortex of the brain. Humans usually have trichromatic vision with three kinds of cones with different photopsins, which have different response curves and thus respond to variation in color in different ways. The cones are conventionally labeled according to the ordering of the wavelengths of the peaks of their spectral sensitivities: short (S), medium (M), and long (L) cone types. The three cone types do not correspond closely to particular colors. For example, L cones are referred to as red receptors, but their peak sensitivity is in the greenish-yellow region of the spectrum. The S-cones and M-cones do not directly correspond to blue and green, although they are referred as such. The RGB color model is a convenient means for representing color, but not directly based on the types of cones in the human eye.

[00699] A rod is sensitive enough to respond to a single photon of light and is about 100 times more sensitive to a single photon than cones. Rods require less light to function than cones, they are therefore the primary source of visual information at night (scotopic vision). Cones require tens to hundreds of photons to become activated. Additionally, multiple rods converge on a single interneuron, collecting and amplifying the signals. However, this convergence reduces visual acuity because the pooled information from multiple cells is less distinct than it would be if the visual system received information from each rod individually. The convergence of rods also tends to

make peripheral vision very sensitive to movement, and is responsible for the phenomenon of seeing something vague occur out of the corner of the eye.

[00700] Rods respond more slowly to light than cones do, so stimuli they receive are added over about 100 milliseconds. While this makes rods more sensitive to smaller amounts of light, it also means that their ability to sense changes, such as quickly changing images, is less accurate than that of cones.

[00701] Rods are most sensitive to wavelengths of light around 498 nm (green-blue), and insensitive to wavelengths longer than about 640 nm (red). This fact is responsible for the Purkinje effect: as intensity dims at twilight, the rods take over, and before color disappears completely, peak sensitivity of vision shifts towards the area of rod peak sensitivity (blue-green). To this end, such distinction between rods and cones may be exploited by using techniques that adapt a display to efficiently make use of both types of photoreceptor cells.

[00702] In different embodiments, various techniques may be employed to exploit the different concentration of photoreceptor cells located in different areas of the human eye. The fovea has a high concentration of cones. The center of the fovea is the foveola with a diameter of about 0.3 mm and an area of about 0.1 mm^2. The area of the foveola is about 0.01% of the retina and 0.1% of the central retina. The foveola has a very high concentration of cones but very few or no rods. The fovea and foveola have very compact cones that are thinner and more rod-like than cones elsewhere.

[00703] The central retina contains the parafovea and perifovea. The fovea is surrounded by the parafovea belt and the perifovea outer region. The parafovea belt ganglion cell layer is composed of more than five rows of cells, as well as the highest density of cones. The perifovea region ganglion cell layer contains two to four rows of cells, and is where visual acuity falls below the peak. The perifovea contains about 120 cones/mm versus about 500 cones/mm in the central fovea. The perifovea is surrounded by a larger peripheral area that delivers highly compressed information of low resolution.

[00704] The average number of rods in the eye is about 60 million and the average number of cones is about 3 million. At 40 microns from the foveola, cone density is about 125 thousand cones/mm^2 (about 350 cones/mm). The cone density in the foveal center is 170 thousand cones/mm^2 (about 400 cones/mm). Towards the retinal periphery, cone density decreases from 6 thousand cones/mm^2 (about 80 cones/mm) at a distance of 1.5 mm from the fovea to 2500 cells/mm^2 (50 cones/mm) close to the ora serrata. Cone diameter increases from the center towards the periphery. At 40 microns from the foveola, cone diameter is about 3.3 microns. In the outer retinal regions, cone diameter about 10 microns.

[00705] Rod density is highest in a ring-like area about 3-5 mm from the foveola with an average of 70 thousand rods/mm^2 (about 250 rods/ mm). Rod density peaks at 150 thousand rods/mm^2 (about 400 rods/ mm). Rod density decreases towards the retinal periphery to 35 thousand rods/mm^2 (about 200 rods/mm). Rod diameter increases from about 3 microns to about 6 microns in the periphery. The hexagonal rod and cone inner segments are arranged in a regular honey-comb fashion.

[00706] The optic nerve contains ganglion cell axons connected to the brain. The optic nerve also contains blood vessels that supply the retinal layers and neurons. The ganglion cells are the output neurons of the retina and lie innermost in the retina closest to the lens and front of the eye. The rods and cones are situated outermost in the retina against the pigment epithelium and choroid. Light travels through the retina and strikes the rods and cones. The absorption of photons by the visual pigment of the rods and cones is translated into first a biochemical signal and then an electrical signal that stimulates the neurons of the retina. The retinal signal is transmitted to the brain by spiking discharge patterns of the ganglion cells.

[00707] In the fovea the ratio of ganglion cells to photoreceptors is about 2.5:1; with almost every ganglion cell receiving data from a single cone, and each cone feeding between 1 and 3 ganglion cells. The acuity of foveal vision is determined by the density of the cone matrix, and the fovea is the area of the eye with the highest sensitivity to detail.

[00708] Visual acuity (image resolution, or just vision) measures to the ability to resolve two separated points or lines. The fovea is displaced about 4 to 8 degrees temporal to the optical axis. The fovea sees only the central two degrees of the visual field, roughly equal to the width of a thumbnail at arm's length. Assuming 20/20 visual acuity, the fovea has the ability to differentiate points around 1 arc-minute apart (1/60 of a degree), or about 1/16 of an inch at 20 feet.

[00709] Assuming the user holds a device display at 1 ft. (0.3 m) from their eyes, 1 arc-minute visual acuity corresponds to about 100 microns spacing (89 microns). With a field of vision equal to 2 degrees, the user can see a circular region 120 arc-minutes across. If we assume a pixel spacing of 1 arc-minute, the user can see a maximum area of about 10000 such pixels at the visual acuity limit.

[00710] A user typically focus on a small region of a display (e.g. text, graphics, movie, picture, web page, etc.). We define this small region of user focus as the area of attention. For example the user focus may be an application window, the area of attention may be a tab or web page etc. within the application window (e.g. within the user focus, etc.). Physiologists also refer to the exact point of eye gaze as the fixation point (or fixation). Thus, for example, the fixation may be a pixel within a character within a word within the area of attention within the user focus. To this end, the different zones and levels of visual acuity in the human eye may be exploited by adapting one or more properties associated with a display such that the resources used (i.e. resolution, backlight intensity, etc.) do not exceed the eye's capacity to receive.

[00711] The iPhone 4s has a 960-by-640-pixel display with 326 ppi resolution and 78 micron pixel width, at about the visual acuity limit (89 microns). The iPhone 4s screen thus has 614400 pixels of which only about 10000 or 16% (about one-sixth) can be seen with a 2 degree field of vision. The area of user focus corresponding to highest visual acuity on a typical display while reading text is only about 5 characters wide.

[00712] Most of the area of a conventional high-resolution display is wasted since the eye cannot see at high resolution over the area subtended by the entire display. In various embodiments, energy savings (i.e. battery life, etc.) and cost savings (i.e. battery size,

etc.) may be obtained by more efficiently tailoring device properties (i.e. display resolution, size, pixel properties, brightness, etc.) to the inherent limits of human physiology.

[00713] In one embodiment, one or more regions (e.g. center strip, top, bottom, side, etc.) of a non-uniform display may provide for high-resolution viewing. For example, see Figure **43**.

[00714] As shown, device **4300** includes a display **4302** which is divided into three regions. In one embodiment, two of these display regions, **4304** and **4308**, may be low resolution, while the third region, **4306** may be high resolution.

[00715] In various other embodiments of a non-uniform display, there may be three regions of a display. Region 1 may be the highest display resolution region, and may be intended for reading text. In one embodiment, Region 1 may be sized smaller than a field of view corresponding to an area of about 2 degrees subtended at the eye. In another embodiment, Region 2 may use a lower resolution display than Region 1, and have an area roughly corresponding to about 5 degrees subtended at the eye. In yet another embodiment, Region 3 may be the lowest resolution are of the display, may comprise a black and white display; Region 3 may be larger than an area corresponding to about 5 degrees subtended at the eye (e.g. roughly peripheral vision, etc.).

[00716] In one embodiment, Region 1 may be any shape, including but not limited to, a rectangle, square, circle, or any other shape. In another embodiment, Region 2 may surround Region 1. In yet another embodiment, Region 3 may surround Region 2. Of course, the regions may be any shape or size that allow a display to be divided into regions intended for one or more fields of vision (e.g. central vision, peripheral vision, etc.).

[00717] In one embodiment of a non-uniform display, one or more regions of the non-uniform display may be constructed such that the resolution is below normal visual acuity. In the context of the present description, visual acuity refers to a measure of the ability to resolve two separated points or lines.

[00718] In one embodiment, a device user interface may take advantage of the phenomenon of hyperacuity. In the context of the present description, hyperacuity refers to the eye's ability to resolve, in some special situations, object or lines at a level below normal visual acuity.

[00719] One example of hyperacuity is vernier acuity. In the context of the present description, vernier acuity refers to a measure of the ability to align two line segments. Under optimal conditions, the vernier acuity limit is about 8 arc seconds, about 10% of normal visual acuity. Vernier acuity is thought to be a process of the visual cortex rather than of the retina. Vernier acuity may have the same underlying mechanism that enables humans to resolve very slight differences in the orientations of two lines. Orientation is known to be processed in the visual cortex.

[00720] The stereoscopic acuity is the ability to detect small differences in depth with two eyes. For complex targets, stereoscopic acuity is similar to normal monocular visual acuity, but for simple targets, such as vertical rods, may be as low as only 2 arc seconds, about 3% of normal visual acuity.

[00721] The smallest detectable visual angle produced by a single fine dark line is about 0.5 arc seconds, about 1% of normal visual acuity. The mechanism is the ability to detect very small differences in contrast or illumination and does not depend on the angular width of the line, which cannot be discerned. Thus as a fine line gets finer, it appears to get fainter but not thinner.

[00722] In one embodiment of a non-uniform display, one or more regions of the non-uniform display may be constructed so that the resolution is below normal visual acuity (i.e. a resolution corresponding to hyperacuity, vernier acuity, stereoscopic acuity, etc.).

[00723] In one embodiment, hyperacuity may be used in a device user interface. For example, in one embodiment, two lines may be used to represent the output of an accelerometer (i.e. device orientation, tilt, etc.). When the tilt of the device is adjusted to exactly zero (i.e. the device is level, etc.), the two lines may be moved such that they align. Because the vernier acuity of the human eye is better than normal visual acuity,

the eye is extremely sensitive to this sort of line alignment. This method may thus result in an increased accuracy in leveling the device. In various embodiments, these two lines may be refers to as a hyperacuity widget.

[00724] Other embodiments of the hyperacuity widget may be based upon the alignment of icons or other graphics. For example, in one embodiment, two lines at an angle may be used as a hyperacuity widget to represent the output of a rotation sensor.

[00725] In one embodiment of a non-uniform display, a hyperacuity widget may be displayed on a very high-resolution portion of the display.

[00726] In one embodiment of a non-uniform display, one or more hyperacuity widgets may be utilized to determine optimal display properties (i.e. optimal display resolution for specific lighting conditions, etc.). For example, in one embodiment, a hyperacuity widget may consist of a fine line displayed on a high-resolution portion of a display. The user may adjust the magnification of an image by adjusting the appearance of the fine line hyperacuity widget. When the fine line no longer appears to get narrower, but instead only fainter, the user knows they are no longer able to resolve the thickness of lines in an image, and further increases in magnification may result in diminishing returns.

[00727] In another embodiment of a non-uniform display, the user may adjust backlighting or some other display property (e.g. color gamut, contrast, etc.) according to the appearance of one or more hyperacuity widgets. As an option, the display being adjusted need not be the same display that shows the hyperacuity widget, but may be.

[00728] In yet another embodiment, the device may display one or more hyperacuity widgets to assist the user in adjusting display properties of text. Display properties of text may include, but are not limited to, font, size, style, color, and/or any other text property.

[00729] In one embodiment of a non-uniform display, the user may be presented with an interface containing one or more hyperacuity widgets as well as controls (i.e. buttons, sliders, etc.) associated with various display properties, to assist the user in determining

optimal display settings. In another embodiment, the device may make a series of changes to one or more display properties and then prompt the user to respond with input on how the resolution of the hyperacuity widgets has changed.

[00730] In one embodiment of a non-uniform display, text may be scrolled, moved, translated, or displaced such that the area of attention is positioned in the high-resolution region of the display. In another embodiment, text may be scrolled, moved, translated, or displaced such that the area of attention is positioned in one or more regions of the display which have different display properties.

[00731] In the context of the present description, an area of attention refers to a subset of the user focus on which a user's field of vision is concentrated. For example, in one embodiment, if the user focus is an application window, then the area of attention may be a tab or web page within that application window.

[00732] In various embodiments of a non-uniform display, the display content may be modified such that potential areas of attention are positioned in one or more regions of the display having different display properties. For example, in one embodiment, text may be moved to a text-optimized region of a display (i.e. e-ink, high contrast, black-and-white, high resolution, etc.) while images may be moved to an image-optimized region of a display (i.e. color, high color fidelity, lower resolution, etc.). In one embodiment of a non-uniform display, the area of attention may be magnified.

[00733] In one embodiment of a non-uniform display, one or more displays may comprise adjoining displays of different properties. For example, in one embodiment, a display may consist of one or more smaller displays joined together. As a specific example, a display may comprise a first display which is an e-ink display, joined with a second display which is a LCD display.

[00734] In one embodiment, a display may comprise one or more display regions with different display properties. For example, in another embodiment, a first display region may have a first pattern of display pixels and a second region may have a second different pattern of display pixels.

[00735] In various embodiments of a non-uniform display, one or more displays may have one or more properties that may be modified. For example, in one embodiment, a display may comprise a layered e-ink and LCD display, and may be adjusted such that one or more regions are e-ink only and one or more regions are LCD only. Of course, any combination(s) of display technology may be used in a similar fashion.

[00736] In various embodiments of a non-uniform display, the display content may be manipulated such that potential areas of attention are positioned on one or more displays having different display properties. For example, in one embodiment, text may be moved to a text-optimized display (i.e. e-ink, black-and-white, higher resolution, etc.) and images moved to an image-optimized display (e.g. color, lower resolution, etc.).

[00737] In another embodiment of a non-uniform display, the display content may be manipulated such that potential areas of attention are positioned on one or more displays that may be on different devices. In yet another embodiment, the display content may be manipulated such that potential areas of attention are positioned on one or more displays that may be on a removable portion of the device.

[00738] In various embodiments of a non-uniform display, the display properties of regions of one or more displays may be adjusted such that they are optimized for viewing the one or more potential areas of attention located within them. For example, in one embodiment, contrast may be adjusted for an area of text while brightness may be adjusted for an area of photographs.

[00739] In one embodiment of a non-uniform display, any display property or combination of display properties may be changed in order to optimize any number of regions of the one or more displays.

[00740] In another embodiment, the display scrolling may be performed by the user. For example, in one embodiment, the user may perform the scrolling using a backtouch interface. In yet another embodiment, the scrolling may be performed by the device. As an option, the user may specify a scrolling speed. In still another embodiment, the scrolling may be controlled by detected user focus (including, in some embodiments, user

gaze). In another embodiment, the scrolling speed may be adjusted automatically based upon observed user focus (i.e. if the user focus is tending to one side or another of the display region, the scrolling speed may be adjusted to bring the user focus back to the center of the device).

[00741] Another example of a non-uniform display may be seen in Figure **44**, which shows a display which makes use of a light valve technique.

[00742] In various other embodiments, device efficiency may be increased by taking advantage of the properties associated with rods and cones. For example, Figure **45** shows a display which mixes color pixels with black and white pixels on different display types, to stimulate both rods and cones. Figure **46** shows a display mixing color with black and white across a matrix to stimulate both rods and cones. Finally, Figure **47** shows a display modulated between color and black and white, to simulate sharpness and enhanced color.

[00743] Figure **48** shows a device **4800** with a display having a non-uniform backlight, in accordance with one embodiment. As an option, the device **4800** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the device **4800** may be carried out in any desired environment. The aforementioned definitions may apply during the present description.

[00744] In many cases, the most energy consuming part of a typical device are the display backlights. In various embodiments, energy consumption may be reduced through the use of displays having a non-uniform backlight.

[00745] In the context of the present description, a non-uniform backlight display refers to a display whose backlight may have one or more regions with different backlight intensities. In one embodiment, a non-uniform backlight may be combined with a non-uniform display.

[00746] In one embodiment of a non-uniform backlight display, one or more displays may be divided into one or more regions, each region having at least one different

backlight property. Backlight properties may include, but are not limited to, intensity, color temperature, time until automatic dimming, power saving dim level, and/or any other property of a backlight.

[00747] For example, in one embodiment, some regions of the display may have low intensity backlighting and other regions of the display may have high intensity backlighting. For example, see Figure **48**.

[00748] As shown, device **4800** includes a display **4802** which has a non-uniform backlight. In one embodiment, display **4802** may have two regions, a region **4804** having high intensity backlighting and a region **4806** having low intensity backlighting.

[00749] In various embodiments, one or more aspects or properties of the display backlighting may be variable. For example, in one embodiment, the location and/or size of regions having different backlight properties may be variable.

[00750] In one embodiment of a non-uniform backlight display, high-intensity back lighting may be used for one or more areas of attention. In another embodiment, areas of attention may be moved to regions of high intensity backlighting. As an option, the movement of areas of attention may be performed by the user or by the device. In yet another embodiment, the movement of areas of attention may be between one or more displays having different backlight properties and/or having different backlight settings.

[00751] In some embodiments, the regions of different backlight properties may be separated by distinct boundaries. See, for example, device **4800**. In other embodiments, the backlight properties may vary as a function of display location. As an option, this function may be continuous and differentiable. For example, in one embodiment, the backlight might gradually vary from a high intensity to low intensity.

[00752] In various embodiments, one or more aspects of the display backlighting may vary as a function of user input. For example, in one embodiment, variable backlighting may be used to provide feedback to the user when interacting with a proximity-sensitive backtouch interface. Specifically, in one embodiment, the contrast between the backlighting in the vicinity of an implement of interaction near a proximity-sensitive

surface and the backlighting further away from the implement may increase as a function of proximity, such that right before contact is made the backlight is only activated in the vicinity of the implement of interaction.

[00753] In other embodiments, variable backlighting may be used as other forms of feedback. For example, in one embodiment, the backlighting in the area around a contact point might increase or decrease in intensity as a function of contact pressure. In another embodiment, the backlighting in the areas of a display which have been touched as part of a gesture may be temporarily decreased as the gesture is performed, such that the user may see the path traveled by an implement of interaction in the form of a temporarily darkened trail.

[00754] Figure **49** shows a device **4900** having a removable back sensor or display, in accordance with one embodiment. As an option, the device **4900** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the device **4900** may be carried out in any desired environment. The aforementioned definitions may apply during the present description.

[00755] The use of modular components in a device may provide certain advantages, such as reduced cost and increased ease of repair. Modularity may also introduce new functionality to the device.

[00756] For example, in one embodiment, some modular components may continue to function after being separated from the device. For example, see Figure **49**.

[00757] As shown, device **4900** includes a front display **4902** and a touch-sensitive rear interaction surface located on a detachable device **4904**. In one embodiment, the interaction surface of detachable device **4904** may also be a low power e-ink display.

[00758] In the context of the present description, device **4900** may be referred to as a host device, and device **4904** may be referred to as a detached device.

[00759] In one embodiment, the detached device may be a display, an interaction surface, a display with a backtouch interface, or any other combination of display(s) and

interaction surface(s). In another embodiment, a detached device may have neither a display nor an interaction surface, but instead embody some other form of functionality. Possible functions include, but are not limited to, magnetic data stripe, storage, backlight, camera, speaker, microphone, keyboard, and/or any other type of device or apparatus.

[00760] In one embodiment, the front display of the host device may be removable. In another embodiment, one or more portions of the front display of the host device may be removable. In yet another embodiment, one or more other displays, or one or more portions of other displays, may be removable from the host device. As an option, these removable displays may or may not also include some form of interaction surface.

[00761] In one embodiment, one or more interaction surfaces may be removable from the host device. As an option, the removable interaction surface(s) may or may not have some form of display functionality. In another embodiment, one or more interaction surfaces which also function as displays may be removable from the host device. In yet another embodiment, one or more detached devices may comprise any combination of touchpad(s), touchscreen(s), display(s), touch interfaces, and/or any other form of interaction surface.

[00762] In one embodiment, a detached device may include internal sensors. Internal sensors may include, but are not limited to, accelerometers, digital compasses, RFID sensors, and/or any other type of sensor. In another embodiment, a detached device may include a CPU, one or more forms of storage, a microphone, a speaker, a battery, and/or any other device or component which may also be found in a host device.

[00763] In one embodiment, a detached device may be housed in a transparent or translucent material. In another embodiment, only portions of a detached device may be housed in transparent or translucent material.

[00764] In various embodiments, a detached device may be capable of operating in one of a plurality of roles within a host device. For example, in one embodiment, a detachable touch-sensitive front display may be removed from the front and placed on the

back of the host device, where it may serve as a backtouch interface (i.e. display disabled).

[00765] In some embodiments, a host device may accept different types of detached devices. For example, in one embodiment, a host device may have front and back touch-sensitive LCD displays detached and replaced with a front e-ink display and a backtouch interface.

[00766] In one embodiment, a detached device may be powered via wireless, wired, battery, energy transmission (i.e. mutually coupled coils, other magnetic coupling technology, etc.), supercapacitor, and/or any other type of power source.

[00767] In various embodiments, a detachable device may continue to provide functionality to the host device after being detached. For example, in one embodiment, a detached backtouch interface may serve as a wired or wireless track pad, providing input to the host device. In various embodiments, any data coupling between host device(s0 and detached device(s) may be accomplished through wired, wireless, or a combination wired/wireless connection. Wireless connections may include, but are not limited to, RF, near-field communication (NFC), optical, infra-red, and/or any other form or type of wireless communication or transmission.

[00768] In one embodiment, a detached device may be approximately the size of a credit card. In another embodiment, a detached device may contain a magnetic strip, and may be used in conjunction with a magnetic card reader.

[00769] In various embodiments, a detachable device may be capable of determining its context with respect to a host device. For example, in one embodiment, a detached touchscreen may be able to detect whether it is being attached to the front of a host device or the back. As an option, a detachable device may use this context data to modify gesture definitions. For example, a left-to-right swipe gesture in a front context may be redefined to be a right-to-left gesture when the detachable device is in a rear context.

[00770] In one embodiment, a user, device, or other agent may modify how the device responds to a user. In another embodiment, a user, device, or other agent may modify how the user commands the device.

[00771] Figure **50** shows a user interface **5000** containing an adaptive soft keyboard, in accordance with one embodiment. As an option, the user interface **5000** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **5000** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00772] In the context of the present description, an adaptive soft keyboard refers to an on-screen software keyboard which has at least one aspect which may vary as a function of user input or interaction. Keyboard aspects may include, but are not limited to, key size, key arrangement, key press sensitivity, and/or any other aspect which may be associated with a soft keyboard. For example, see Figure **50**.

[00773] As shown, user interface **5000** contains an adaptive soft keyboard **5002**. In one embodiment, the keys of **5002** may vary in size, depending on the contact area associated with recent or historical key presses. For example, in one embodiment, a small contact area **5004** may result in a soft keyboard with smaller keys, while a large contact area **5006** may result in a soft keyboard with larger keys. In this way, the soft keyboard may adapt to the size of the implement(s) of interaction (i.e. finger, stylus, etc.) being used, to facilitate the interaction.

[00774] In various embodiments, one or more keys on a soft keyboard may change in size. In one embodiment, all keys on the soft keyboard may change in size. In another embodiment, only a subset of keys may change in size. (e.g. most commonly used keys change, vowels change, etc.).

[00775] In one embodiment, the key size may depend on touch pressure. For example, in one embodiment, the user may increase the size of one or more keys by exerting a greater contact pressure.

[00776] In one embodiment, one or more key sizes may depend on a computed metric. Computed metrics may include, but are not limited to, average contact area, average number of key presses over time, frequency of pressing specific keys, and/or any other metric associated with the use of a keyboard.

[00777] In various embodiments, key arrangement may depend upon how the user is holding the device. For example, in one embodiment, if the device determines that the user is holding it with thumbs on the front and fingers on the back, the soft keyboard may arrange the keys such that they are in easy reach of the user's thumbs. This determination may be based upon detected contact points on a rear interaction surface, contact with interaction surfaces located on the sides of the device, or by any other method of estimating user grip.

[00778] Figure **51** shows a user interface **5100** with context-sensitive backlighting, in accordance with one embodiment. As an option, the user interface **5100** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **5100** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00779] In the context of the present description, context-sensitive backlighting refers to a method of backlighting a display where one or more backlight properties may depend upon contextual factors. Contextual factors may include, but are not limited to, type of content displayed, user focus, historical usage data, user input through a backtouch interface, other user input, and/or any other contextual factors. For example, see Figure **51**.

[00780] As shown, user interface **5100** makes use of context-sensitive backlighting such that there are one or more areas of bright backlighting **5102** as well as one or more areas of dim backlighting **710.040-04**. In various embodiments, the size, shape, brightness, and numbers of bright and dim areas within user interface **5100** may depend upon the attributes of a user area of attention. In the context of the present description,

the terms bright and dim are relative terms, indicating a level of backlighting as compared to other regions of the display.

[00781] The size and shape of a user area of attention may vary depending on the type of content being viewed. For example, when browsing a rectangular page of search results, a user typically scans text at the top of a page from left-right, then quickly scans vertically with increasingly shorter left-to-right scans. This may result in a triangular area of attention. This may be true for text in western style alphabets; there may be differences for languages read from right to left.

[00782] In various embodiments of a user interface utilizing context-sensitive backlighting, one or more areas of bright backlighting may be fitted to one or more areas of attention, depending on display content. For example, in one embodiment, a triangular area of the user interface may be brightly backlight when displaying text search results. See, for example, user interface **5106**, having a brightly backlit triangular region **5108** for the text search results, and a dimly backlit area **5110** elsewhere.

[00783] A round area of attention may be associated with viewing movies or images. In another embodiment, a round area of the user interface may be brightly backlit when displaying movies or images. See, for example, user interface **5112**, having a brightly backlit circular region **5114** for a movie, and a dimly backlit area **5116** elsewhere. In another embodiment, the display content may be modified with respect to the backlit area of attention. Modifications may include, but are not limited to, scaling, translation, and/or any other operation which may be performed on image or movie data.

[00784] A rectangular area of attention may be associated with reading a block of text. In yet another embodiment, a rectangular area of the user interface may be brightly backlit when displaying a block of text. See, for example, user interface **5118**, having a brightly backlit rectangular region **5120** for a block of text, and a dimly backlit area **5122** elsewhere. In another embodiment, the text may be modified with respect to the backlit area of attention. Modifications may include, but are not limited to, scaling, translation, reflow, font size, wrapping style, hyphenation style, and/or any other modification which may be performed on text.

[00785] In various embodiments, the areas of attention, and backlight regions, may be of any shape. In one embodiment, the user may select the display properties for different regions of the user interface and/or different areas of attention. In another embodiment, the content type and use context may be determined automatically, using methods and information including, but not limited to, historical use data, user focus tracking (including user gaze tracking), file extensions, analysis of display content, application type, and/or any other possible source of contextual information. As an option, the definition of areas of attention, establishment of backlighting regions, and modification of displayed contents may be performed automatically.

[00786] In various embodiments, one or more aspects of the areas of attention, backlight properties, and/or display properties may be adjusted by the user through a backtouch interface. For example, in one embodiment, a user may move a finger on a backtouch input while reading text to adjust the backlight intensity to be increased over the area of attention.

[00787] In various embodiments, the backlight display properties associated with the area located near a boundary between two backlight regions may be defined as a function of the properties of both regions. For example, in one embodiment, the border between two backlight regions may be defined such that there is a gradient spanning the backlight intensities associated with the two regions. In another embodiment, the border between the two backlight regions may be defined such that there is a sharp change in brightness at the boundary. In yet another embodiment, the backlight display properties associated with the area located near a boundary between two backlight regions may be defined as a function of the other properties of both regions, properties which may include, but are not limited to, type of content being displayed, size of region, region creation time/date, and/or any other property which may be associated with a backlight region.

[00788] Figure **52** shows a 3D layered user interface **5200**, in accordance with one embodiment. As an option, the user interface **5200** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of

course, however, the user interface **5200** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00789] In the context of the present description, a 3D layered user interface refers to a user interface in which displayed objects may exist in one of a plurality of layers. For example, see Figure **52**.

[00790] As shown, user interface **5200** is comprised of two layers, a front layer **5202** and a back layer **5204**. These layers are depicted in a staggered arrangement, to illustrate the depth; it should be noted that in various embodiments, the layers would be aligned. Object **5206** exists in the front layer, while object **5208** exists in the back layer. In various embodiments, a user may interact with the layers separately.

[00791] In various embodiments, a 3D layered user interface may be implemented in a layered display device. In the context of the present description, a layered display device refers to a device with at least one display that is made up of a plurality of overlapping display layers, each layer able to display content independent of other layers, with all layers facing the same direction. For example, in one embodiment of a layered display device, the device may have a display made up of a transparent display located on top of a second display, both facing forward. In another embodiment, the display layers may be separated by a non-opaque material, to enhance the layered appearance.

[00792] In another embodiment of a layered display device, the display layers may include one or more e-ink layers. In yet another embodiment, the display layers may include one or more LCD layers. In still another embodiment, the display layers may include both e-ink and LCD layers. Other embodiments may include any combination of layers embodying any display technology.

[00793] In various embodiments, a 3D layered user interface may be implemented in a device having single layer displays through the use of virtual display layers. In the context of the present description, a virtual display layer refers to a collection of display elements which have all been assigned the same apparent depth within the 3D layered user interface. In various embodiments, a 3D layered user interface may make use of

multiple virtual display layers. For example, in one embodiment, a 3D layered user interface may have a virtual foreground display layer and a virtual background display layer.

[00794] Additionally, in the context of the present description, a display element refers to the region of the display allotted to an identifiable visual object. Identifiable visual objects may include, but are not limited to, digital objects (i.e. images, text boxes, UI control widgets, selections, etc.), subject matter objects (i.e. people within a photograph, letters and/or words within text, etc.), and/or any other type of identifiable visual object. When located in a virtual display layer, display elements may be referred to as 3D display elements.

[00795] In various embodiments, virtual display layers may be given the appearance of depth through the use of 3D depth cues. In the context of the present description, a 3D depth cue refers to an effect, manipulation, transformation, animation, or operation which gives a visual indication of simulated depth. For example, in one embodiment, a 3D depth cue may be a blur operation, such that layers located at successively greater depths may appear blurrier than those closer to the user. In various embodiments, one or more 3D depth cues may be used to give the appearance of depth to one or more virtual display layers. In one embodiment, different 3D depth cues may be used for different virtual display layers. In another embodiment, 3D depth cues may be used in conjunction with a 3D layered user interface implemented in a layered display device, to enhance the layered appearance.

[00796] In one embodiment, a 3D depth cue may include the addition of a shadow element to display elements located within a virtual display layer, making them to appear to float above the next virtual display layer. As an option, the location of the light source(s) casting the shadows may be based in part on data obtained from one or more cameras located on the host device.

[00797] In one embodiment, a 3D depth cue may include the addition of a depth-based fog, giving the appearance that the layers exist in an environment with a thin, uniform fog. In this way, distant objects may be "foggier" than objects close to the user.

[00798] In one embodiment, a 3D depth cue may include a depth-based apparent rate of movement. For example, in a situation where a user is swiping through multiple screens of layered content, the layers closer to the user may appear to move faster than those more distant, giving the appearance of depth.

[00799] In one embodiment, a 3D depth cue may include a time-dependent visual transformation. For example, in one embodiment, a background layer may be transformed such that it appears to be below rippling water, while the foreground layer appears to be floating on the surface. In another embodiment, the visual transformation may be static.

[00800] In one embodiment, a 3D depth cue may include animated visual elements which appear to exist in between layers. Elements may include, but are not limited to, birds or insects flying, shooting stars, tiny people walking, grass blowing in the wind, and/or any other visual element.

[00801] In one embodiment, a 3D depth cue may include moving and/or transforming display elements within a virtual display layer based upon detected user gaze and/or head position. For example, in one embodiment, display elements may be compressed and virtual display layers spread apart if the device determines the user is viewing the display from an angle, the degree of transformation varying with estimated viewing angle. In another embodiment, display elements located on different virtual display layers may be slide around within their respective layers as the user changes their angle of viewing, allowing them to "look around a corner" to see display elements that would be obscured when viewed head on.

[00802] In one embodiment, a 3D depth cue may include moving and/or transforming display elements within a virtual display layer based upon changes in device orientation detected through one or more hardware interfaces (i.e. accelerometer, tilt sensor, compass, etc.).

[00803] In one embodiment of a 3D layered user interface, the 3D cues may be implemented such that the user interface has an apparent depth equal to the thickness of

the device. In another embodiment, the 3D cues may be implemented such that the user interface has an apparent depth equal to the distance between a front display and a backtouch interface.

[00804] In various embodiments, a user may interact with the layers separately. For example, in one embodiment, a front interaction surface may be used to interact with display elements in a foreground display layer, and a back interaction surface may be used to interact with display elements in a background display layer. See, for example, Figure **52**.

[00805] User interface **5200** is being implemented in a device having front and back touch-sensitive interaction surfaces. As shown, a user may manipulate display element **5206**, located in front layer **5202**, through a gesture **5210** performed on the front interaction surface. At the same time, a user may manipulated display element **5208**, located in back layer **5204**, through a gesture **5212** performed on the back interaction surface. In one embodiment, the manipulation of one of these display elements may be done completely independent of the other display element.

[00806] In one embodiment of a 3D layered user interface, a front input interface may be used to move display elements on a virtual foreground display layer. In another embodiment of a 3D layered user interface, a rear input interface may be used to move display elements on a virtual background display layer.

[00807] In various embodiments, there may be display layers with which a user may not interact. For example, in one embodiment, there may be an emphasis display layer situated in front of all other display layers. In the context of the present description, an emphasis display layer refers to a display layer (virtual or physical) which is used to provide visual emphasis to one or more display elements without providing independent means of user interaction. In other words, when a user interacts with a display element located in an emphasis display layer, they might also interact with one or more other layers. For example, in one embodiment, an emphasis display layer may be used to indicate results of a text search within a document (i.e. target words appear to float above the document, etc.). Although located on a different display layer, any text selection

made by the user may incorporate text from both the emphasis display layer and the display layer housing the document. In another embodiment, an emphasis display layer may cause hyperlinks within a webpage to appear to float above the rest of the page, though still able to be selected along with non-hyperlink text.

[00808] In another embodiment, an emphasis display layer may be used to display user avatars floating above the text of a chat session; the user may not interact with the avatars. Other examples of the use of an emphasis display layer may include, but are not limited to, a status bar floating at the top of a display, pop-up volume indicators which appear when the volume is changed, icons within the status bar, and/or any other circumstance where visual emphasis is desired.

[00809] Another example of a display layer with which a user may not interact is an ambiance display layer. In the context of the present description, an ambiance display layer refers to a display layer (virtual or physical) which is used to display decorative display elements with which a user may not directly interact with. For example, in one embodiment, one or more ambiance display layers may be used to create the appearance of three-dimensional structure within the 3D layered user interface (i.e. walls, shelves, backdrop, etc.). In another embodiment, an ambiance display layer may be used to display a wallpaper or some other backdrop which is present behind the elements and controls of the user interface. In yet another embodiment, a user may customize one or more ambiance display layers through a separate user interface, but not through direct interaction.

[00810] In one embodiment of a 3D layered user interface, a rear touch input may be used to control display elements located on a virtual display layer. In another embodiment, a rear touch input may be used together with a front touch input to control one or more display elements on one or more virtual display layers.

[00811] In other embodiments of a 3D layered user interface, a user may interact with one of a plurality of layers through a back interaction surface, the layer being selected using a pressure-sensitive front interface. For example, in one embodiment, a user may select one of a plurality of display layers by applying a predefined level of contact

pressure on the front interaction surface, and interact with the selected layer using a back interaction surface. As an option, 3D depth cues may be used to indicate "moving" through the display layers as the contact pressure is changed (i.e. magnifying and fading display layers as they are passed by, sharpening previously blurry display layers as they grow closer, etc.).

[00812] In one embodiment, one or more display layers associated with a 3D layered user interface may only be interacted with through coordinated user input on multiple interaction surfaces. For example, in one embodiment, a touch input on the front of the device together with a touch input on the back of the device may control the overlap between two or more virtual display layers.

[00813] In one embodiment, display elements may be assigned to virtual display layers, just as they may be assigned to physical display layers in a layered display device. In another embodiment, the process of assigning display elements to virtual display layers may be the same as the process of assigning display elements to physical display layers in a layered display device. In yet another embodiment, a 3D layered user interface may be designed such that it may take advantage of physical display layers, if present, or use virtual display layers if only single layer displays are available. It should be understood that all functionality and embodiments described with respect to virtual display layers may also be implemented in a layered display device.

[00814] In some embodiments, a single layer device may be converted to a layered display device. See, for example, Figure **53**, which shows a device whose rear screen may be reversed to face forward, creating a forward facing display.

[00815] Figure **54** shows a 3D layered user interface **5400** for augmented reality, in accordance with one embodiment. As an option, the user interface **5400** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **5400** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00816] In the context of the present description, augmented reality refers to a view of a physical environment, received from a camera, which is augmented by overlaying additional information from other sources. In various embodiments, the view of a physical environment may be a live video signal from a camera.

[00817] In various embodiments, a 3D layered user interface for augmented reality may display a video signal on one display layer, overlaid with additional information displayed on additional display layers. For example, see Figure **54**.

[00818] As shown, user interface **5400** is comprised of two layers, a back layer **5404** containing a video signal, and a front layer **5402** containing display elements which augment the video signal. These display elements are hereinafter referred to as augmenting display elements. These layers are depicted in a staggered arrangement, to illustrate the depth; it should be noted that in various embodiments, the layers would be aligned.

[00819] In various embodiments, an augmenting display element may include business information. Business information may include, but is not limited to, business name, business contact information (i.e. website, phone number, email address, etc.), business hours, reviews, and/or any other information related to a business. In one embodiment, an augmenting display element may include an iconic and/or text based indication of business type. For example, see restaurant indicator **5406** in user interface **5400**.

[00820] In various embodiments, an augmenting display element may include information obtained from a social network. Social network information may include, but is not limited to, posts tagged with GPS coordinates, posts referring to a specific location, and/or any other information found on a social network. In one embodiment, an augmenting display element may include an iconic and/or text based indication that an individual on a known contacts list made a post at a visible location. For example, see friend post indicator **5408** in user interface **5400**.

[00821] In various embodiments, an augmenting display element may include navigation information. Navigation information may include, but is not limited to route

information, GPS coordinates, estimated time or arrival, street names, compass directions, current address, speed, and/or any other navigation related information. In one embodiment, a augmenting display element may include an iconic and/or text based indication of a route. For example, see route indicator **5410** in user interface **5400**.

[00822] In various embodiments, a 3D layered user interface for augmented reality may use 3D depth cues to visually differentiate augmenting display elements associated with physical locations near the user from those associate with distant physical locations. In one embodiment, 3D depth cues may be used to give augmenting display elements an apparent distance from the user roughly equal to the physical location with which they are associated. In another embodiment, 3D depth cues may be used to augment the video signal such that it has a three dimensional appearance.

[00823] Figure **55** shows a device **5500** with user gaze tracking sensors, in accordance with one embodiment. As an option, the device **5500** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the device **5500** may be carried out in any desired environment. The aforementioned definitions may apply during the present description.

[00824] As discussed previously, user gaze is a type of user focus. In the context of the present description, user gaze tracking refers to a procedure which extrapolates where on a display a user is looking, using data related to the user's eyes.

[00825] In one embodiment, a determination of user gaze may result in a field of eye gaze probabilities. In another embodiment, a determination of user gaze may result in data similar to proximity data obtained from a proximity-sensitive interaction surface. In yet another embodiment, a determination of user gaze may result in a specific location on a display, with an associated uncertainty. As an option, the uncertainty may not be radially symmetric (i.e. greater uncertainty in left-right directions than up-down, etc.).

[00826] In one embodiment, a user gaze tracking sensor may include a camera. In another embodiment, a user gaze tracking sensor may include an infra-red sensor. In various embodiments, a user gaze tracking sensor may also include a light source. In

some embodiments, this light source may be used to observe reflections within a user's eye. In one embodiment, the light source may emit light primarily in the infra-red wavelengths.

[00827] Some embodiments of a device with user gaze tracking sensors may include more than one sensor. For example, see Figure **55**.

[00828] As shown, device **5500** includes a front display **5502** and two user gaze tracking sensors **5504**.

[00829] In one embodiment, the front display **5502** may be a single layer display. In one embodiment, the front display **5502** may be a layered display, having two or more physical display layers. In some embodiments of a layered display device with user gaze tracking sensors, a determination of user gaze may indicate where on the display a user is looking, as well as which layer. As an option, such a determination may also result in uncertainty value(s) for location as well as depth.

[00830] In some embodiments, a device with user gaze tracking sensors may have one sensor positioned for gaze tracking while the device is in a user's hands, and one or more gaze tracking sensors positioned for alternative use scenarios. Possible use scenarios include, but are not limited to, placed in a stand or dock, laying flat on a table, and/or any other use scenarios.

[00831] In some embodiments of a device with user gaze tracking sensors and a single layer display, a determination of user gaze may also be used to determine which virtual display layer a user is looking at. In various embodiments of a 3D layered user interface implemented on a device capable to tracking user gaze, user gaze may be used to select which display layer a user may interact with using an interaction surface. In one embodiment, a device with user gaze tracking sensor(s) may use user gaze tracking to determine one or more areas of attention.

[00832] In various embodiments, a user interface may be sensitive to user gaze. For example, in some embodiments, user gaze may be used to predict what a user wants to do (i.e. likely next step, probable next input, which layer is the area of attention, which

button or key is likely to be pressed, etc.). In one embodiment, the user interface may adapt based upon predicted user intent. Adaptations may include, but are not limited to, resizing of buttons or controls, altering the assigned function of software or hardware controls, adjusting backlighting, changing one or more aspects of one or more display layers (i.e. 3D depth cues, layer order, etc.), and/or any other modification which can be made to a user interface.

[00833] In various embodiments, user gaze may be used as another form of user input. In the embodiments described in the present description, user gaze may be substituted for another form of user input. For example, in one embodiment, a user may be able to perform gestures based upon user gaze. In various embodiments, user gaze tracking may be initiated/terminated through similar or identical methods described for enabling touch, pressure, or proximity sensitive interfaces. In various embodiments, a user may determine if user gaze is being tracked through any of the forms of feedback described to indicate activation of pressure, touch, or proximity-sensitivity.

[00834] In various embodiments, user gaze may be used to generate 3D depth cues. See, for example, Figure **56**, which shows the utilization of user gaze to make a user focus float above a source layer. This and other embodiments may be implemented in a variety of environments, including that shown in Figure **57**, which depicts a two layer display having a transparent front display and opaque back display.

[00835] In various embodiments, other techniques may be used to generate a three-dimensional effect or image. See, for example, Figure **58**, which shows an example of an automultiscopic display. In the context of the present description, an automultiscopic display refers to a type of display which can provide high-resolution stereoscopic images without the use of glasses. In one embodiment, this is done using view-dependent pixels, whose color depends upon viewing angle, combined with parallax barriers.

[00836] In another embodiment, a three dimensional image may be generated through Depth Fused 3D (DFD). See, for example, Figure **59**, which shows how images at an intermediate depth may be produced by controlling the luminances of two stacked 2D images, in accordance with one embodiment.

[00837] In yet another embodiment, a layered display may be utilized for other functions. For example, Figure **60** shows a light-field camera made from a multiple display sensors.

[00838] In these and other embodiments, a user may be provided with different forms of feedback. For example, in one embodiment, one or more embedded force transducers may provide tactile feedback to a user. See, for example, Figure **63**. In another embodiment, feedback may be provided to the user through a secondary device. See, for example, Figure **64**, which shows a watch being used as an extension of a device display.

[00839] Embodiments described herein may be implemented in a variety of environments, including that shown in Figure **65**. Figure **65** shows one embodiment of a wireless storage ring and wireless storage key. Any of the other embodiments may be implemented in this environment.

[00840] Another embodiment of a 3D layered user interface is shown in Figure **61**. In this embodiment, a soft keyboard may be displayed on a lower display layer, beneath an active document (i.e. word processing, email, etc.), such that the words appear to float above the soft keyboard. In another embodiment, a user may interact with the soft keyboard through a back interaction surface. In yet another embodiment, Figure **62** shows a different implementation of a 3D layered user interface on a layered display backtouch-enabled device.

[00841] Figure **66** shows a 3D layered user interface **6600** having a clipboard layer, in accordance with one embodiment. As an option, the user interface **6600** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **6600** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00842] In the context of the present description, a clipboard layer refers to a display layer in a 3D layered user interface that is used to store one or more selectable objects upon which relocation/replication operations may be performed. Relocation/replication

operations may include, but are not limited to, copying, pasting, sending to specified application, inserting into a new message (i.e. email, SMS, social network posting, etc.), and/or any other operation which may be performed on a selectable object. Selectable objects may include, but are not limited to, images, text (plain, rich, hypertext, etc.), video clips, sound clips, files, and/or any other type of object.

[00843] In various embodiments, a 3D layered user interface with a clipboard layer may insert the clipboard layer immediately after the active display layer. For example, see Figure **66**.

[00844] As shown, user interface **6600** is comprised of two layers, an active layer **6602** and a clipboard layer **6604**. These layers are depicted in a staggered arrangement, to illustrate the depth; it should be noted that in various embodiments, the layers would be aligned. In one embodiment, user interface **6600** may include additional layers. In another embodiment, active layer **6602** may contain one or more display elements, one or more of which either is, or contains, a selectable object.

[00845] In various embodiments, an object may be selected before it may be placed on the clipboard layer. Objects may be selected using any of the methods in the present description, or any other methods or procedures for indicating a selection. In one embodiment, a screen capture may be placed on the clipboard layer without having to be selected.

[00846] Once there is an active selection, the selected object may be moved or copied to the clipboard layer. In various embodiments, the clipboard layer may be activated before an object may be moved or copied to it. In one embodiment, the clipboard layer may be activated through selecting an icon located in the status bar, or elsewhere on the display.

[00847] In various embodiments, the clipboard layer may be activated by performing a gesture. For example, in one embodiment, the clipboard layer may be activated with a 1-6 pinch slide gesture, where the user "pulls" the clipboard layer into view from one edge of the display. In various other embodiments, the clipboard layer may only be active

while a gesture is being performed, with the gesture's final operation being hiding the clipboard layer. For example, in one embodiment, the clipboard layer may be activated after a squeeze gesture (i.e. a double-press squeeze gesture, etc.) is initiated, and remains active until the contact pressures fall below a threshold level, ending the gesture.

[00848] In various embodiments, an activated clipboard layer is placed directly after the active display layer (i.e. the layer which contains the selection, etc.). In various embodiments, the activation of a clipboard layer may change one or more visual properties of the active display layer. For example, in one embodiment, the active display layer may become translucent, allowing the user to see the contents of the clipboard layer. In some embodiments, the degree of translucency may be a function of a user input. For example, in one embodiment, the active display layer may become more translucent as the user increases a contact pressure (i.e. a contact pressure associated with an activating gesture, backtouch pressure, etc.). In one embodiment, the active selection in the active display layer may be less translucent than the rest of the active display layer.

[00849] In various embodiments, once the clipboard layer is activated, objects may be copied or moved to and from the clipboard layer as a result of user input. For example, in one embodiment, an object may be cut (i.e. removed from the active display layer and placed on the clipboard layer) by performing a 1-6 pinch gesture, where the contact pressure on the front interaction surface is greater than the contact pressure on the back interaction surface. This provides the user with the sensation of pushing the selected object through the active display layer, into the clipboard layer.

[00850] In one embodiment, a selected object may be copied from the active display layer by performing a press gesture on the front interaction surface, where the user applies a contact pressure level on the selected object that exceeds a predefined threshold. See, for example, Figure **66**. In another embodiment, the user may be provided with one or more visual indications that the object has been copied to the clipboard layer.

[00851] As shown in Figure **66**, a user has performed a press gesture by applying sufficient contact pressure on contact area **6606**, situated on selected object **6608** which is

located in the active display layer **6602**. The press gesture triggered a copy operation, placing a copy **6610** of the selected object to the clipboard layer **6604**.

[00852] In one embodiment, a clipboard layer may only hold one object. Cutting or copying a second object may replace the first clipboard object. In another embodiment, a clipboard layer may contain multiple objects.

[00853] In one embodiment, an object may be copied from the clipboard layer and pasted into the active display layer by performing a press gesture on a back interaction surface, behind the object. This provides the user with the sensation of pushing a copy of the object up from the clipboard layer to the active display layer. As an option, the object may remain in the clipboard layer after the copy operation.

[00854] In one embodiment, an object may be cut from the clipboard layer and pasted into the active display layer by performing a 1-6 pinch gesture, where the contact pressure on the back interaction surface is greater than the contact pressure on the front interaction surface. This provides the user with the sensation of pulling the clipboard object up to the active display layer. As an option, the object may be removed from the clipboard layer after the cut operation.

[00855] In various embodiments, a clipboard layer may contain multiple objects. In one embodiment, the clipboard layer may operated as a first-in-last-out type container, such that when a user causes a new object to be placed on the clipboard layer, previous objects get incremented in order. In such an embodiment, repeated paste operations from the clipboard to the active display layer may eventually transfer all clipboard objects to the active display layer, while repeated copy operations from the clipboard to the active display layer may create multiple copies of the active clipboard object.

[00856] In the context of the present description, an active clipboard object refers to the clipboard object which is visible in an activated clipboard layer. In various embodiments, the default active clipboard object is the last object placed in the clipboard layer. In one embodiment, a user may cycle through various clipboard objects until the desired object is visible. In another embodiment, the clipboard layer may only display

objects which are able to be copied or pasted into the current active display layer (i.e. a video clip may not be displayed in the clipboard layer, even though it has been copied, if the active display layer is a text-only environment).

[00857] In various embodiments, a user may remove an object from an activated clipboard layer by selecting an icon associated with deletion. See, for example, icon **6612** in Figure **66**. As an option, the user may be prompted to confirm this operation before removing the object from the clipboard.

[00858] In one embodiment, the user may move all clipboard objects to the active display layer by performing a predefined gesture. In another embodiment, a user may select certain objects to always be available in the clipboard layer. Such objects may include, but are not limited to, email signatures, text snippets, emoticons, avatars, digital signatures, and/or any other often used object.

[00859] Of course, in various other embodiments, these clipboard layer actions (i.e. copy, cut, paste, etc.) may be triggered through any form of user input including, but not limited to, user interface controls, gestures, voice commands, and/or any other form of user input.

[00860] Figure **67** shows a 3D layered user interface **6700** having an operation layer, in accordance with one embodiment. As an option, the user interface **6700** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **6700** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00861] In the context of the present description, an operation layer refers to a display layer in a 3D layered user interface that may be used to apply operations on display elements located in display layers beneath the operation layer. Operations may include, but are not limited to, image manipulations, data detection, image recognition, OCR, movie filters, sound filters, display modifications, and/or any other type of operation which may be applied to one or more display elements, or portions thereof.

[00862] In various embodiments, a 3D layered user interface with an operation layer may insert the clipboard layer immediately above the active display layer. For example, see Figure **67**.

[00863] As shown, user interface **6700** is comprised of two layers, an operation layer **6702** and one or more display layers **6704**. These layers are depicted in a staggered arrangement, to illustrate the depth; it should be noted that in various embodiments, the layers would be aligned. In one embodiment, user interface **6700** may include additional layers. In another embodiment, display layer **6704** may contain one or more display elements, such as graphic **6706**.

[00864] In various embodiments, an operation layer may be activated by performing a gesture. For example, in one embodiment, an operation layer may be activated with a 1-6 pinch slide gesture, where the user "pulls" the operation layer into view from one edge of the display. In various other embodiments, an operation layer may only be active while a gesture is being performed, with the gesture's final operation being hiding the operation layer. For example, in one embodiment, the operation layer may be activated after a squeeze gesture (i.e. a triple-press squeeze gesture, etc.) is initiated, and remains active until the contact pressures fall below a threshold level, ending the gesture.

[00865] In various embodiments, an operation layer may have an operation window which defines the boundaries within which the operation may be applied. See, for example, operation window **6708** in Figure **67**.

[00866] In one embodiment, a user may resize the operation window. In another embodiment, the operation window may be automatically scaled, translated, and/or reshaped to fit a display element which is in or near the user focus. In yet another embodiment, an operation layer may include a button which enlarges the operation window to fit the entire display.

[00867] In various embodiments, an operation window may be able to perform a plurality of operations. In some embodiments, operations may be divided into subject matter categories. In one embodiment, a user may select an operation by first selecting a

subject matter category, then selecting an operation from a list. In various embodiments, once an operation has been selected by the user, it is applied to the contents of the operation window.

[00868] In various embodiments, possible subject matter categories may include, but are not limited to, image, text, video, audio, display, and/or any other category of subject matter. Possible image operations may include, but are not limited to, color operations (i.e. desaturate, replace color, saturate, etc.), changes in color space (i.e. grayscale, sepia, custom 2-tone space, etc.), processing effects (i.e. halftone, pixelate, posterize, etc.), distortions (i.e. twirl, warp, motion blur, etc.), enhancements (i.e. sharpen, blur, image repair, remove red eye, etc.), and/or any other possible image operation.

[00869] One example of a text operation may include data extractors. In the context of the present description, a data extractor refers to logic which parses text data, identifying information fragments which can trigger further action. For example, in one embodiment, a data extractor operation may highlight all email addresses in the text inside the operation window. As an option, touching or otherwise selecting a highlighted email address may create a new email message to that address. Other examples of information fragments may include, but are not limited to, phone numbers, contact names, dates, times, addresses, appointments, web addresses, and/or any other information fragment which may be used in a related action. Other examples of text operations may include, but are not limited to, foreign language translation, profanity removal, optical character recognition, and/or any other text operation.

[00870] In various embodiments, possible video operations may include, but are not limited to, color adjustment, resolution reduction, sound volume adjustment, text overlay, graphics overlay, and/or any other possible video operation. In one embodiment, image operations may also be applied to video.

[00871] In various embodiments, possible audio operations may include, but are not limited to, volume normalization (i.e. softening loud portions, amplifying soft portions), application of equalizer settings, audio effects (i.e. echo, envelope, reverberation, etc.),

and/or any other possible audio operation. In one embodiment, audio operations may also be applied to video.

[00872] In the context of the present description, a display operation refers to an operation which affects one or more display properties for the display region delineated by the operation window. For example, in one embodiment, a display operation may include adjustment of backlight intensity within the operation window. Other possible display operations may include, but are not limited to, application of power saving backlight profiles (i.e. highlighting only areas of interest, etc.), activation/deactivation of display layer types (i.e. only using e-ink within operation window, etc.), modifying 3D depth cues within operation window, and/or any other possible display operation.

[00873] As a specific example of an operation layer in use, see Figure **67**. As shown, a user has selected the "Image" subject matter category from a plurality of categories **6710**, and the "Invert" operation from a plurality of operations **6712**. As a result, an "Invert" operation has been applied to the contents of operation window **6708**.

[00874] In one embodiment, a user may apply multiple operations within a single operation window. As an option, the operations may come from different subject matter categories. In another embodiment, certain operations may only be available if certain conditions are satisfied, such as the operation window spanning entire display.

[00875] In various embodiments, once the user has selected one or more operations, and has adjusted the operation window to their liking, they may choose to apply the operation. In one embodiment, application of the operation may be elected by selecting a button. See, for example, button **6714**.

[00876] Upon application, in one embodiment, the operation may modify all the content within the operation window. In another embodiment, application of the operations may create a modified copy of the content within the operation window. In still other embodiments, the user may be prompted to choose between modifying the original and creating a copy. In one embodiment, certain operations perform non-destructive modifications, such as backlighting modifications. In another embodiment,

some operations may not make permanent changes, but rather are only useful within the operation layer (i.e. certain text operations, etc.).

[00877] In one embodiment, the selected operation(s) may be cancelled and the operation layer hidden by selecting a button. See, for example, button **6716**.

[00878] In various embodiments, operations may be provided by the operating system. In another embodiment, operations may also be provided by third party developers through applications. As an option, these applications may run in the background of a multitasking environment.

[00879] Figure **68** shows a user interface **6800** for providing touch feedback, in accordance with one embodiment. As an option, the user interface **6800** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **6800** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00880] In the context of the present description, touch feedback refers to any form of visual feedback given to the user to indicate the location and uncertainty associated with their current or historical contact with a touch-sensitive interaction surface.

[00881] In various embodiments, touch feedback may be given to the user in the form of a crosshairs. For example, see Figure **68**.

[00882] As shown, user interface **6800** includes a soft keyboard **6802**. In the course of interacting with the soft keyboard, the user makes contact with the interaction surface over contact area **6804**. In one embodiment, the user may be provided with touch feedback indicating the contact point and uncertainty associated with the recent touch through crosshairs **6806**.

[00883] In one embodiment of touch feedback, a crosshairs may be used to indicate contact point and/or touch uncertainty. Crosshairs **6806** are shown as across with lines horizontal and vertical. In other embodiments, different forms of crosshairs may be used.

Possible forms of crosshairs may include, but are not limited to, lines at 45 degrees to the x-axis and y-axis, a star pattern made of three or more lines and/or any other type of crosshairs.

[00884] In various embodiments, the length of one or more lines in the crosshairs may indicate uncertainty associated with that contact event. For example, in one embodiment, the length of the lines may indicate measurement uncertainty along the direction of the line. In another embodiment, the length of one or more lines may indicate the degree of uncertainty associated with other touch-based operations, including, but not limited to, text selection, image selection, control operation, gesture training, gesture recognition, and/or any other type of touch event.

[00885] In one embodiment, the described forms of touch feedback may be used in conjunction with any of the forms and methods of feedback in the present description. In another embodiment, these forms of touch feedback may be used with any touch sensitive interaction surface located on any face of a device. In yet another embodiment, touch feedback may be implemented using a 3D layered user interface, placing the feedback display elements on a display layer above other display layers.

[00886] Figure **69** shows a user interface **6900** for providing increased security, in accordance with one embodiment. As an option, the user interface **6900** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **6900** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00887] In various embodiments, a host device may be capable of being secured with a numeric passcode. Numeric passcodes are easier to remember, but their simplicity and their reliance on a limited keypad make them vulnerable to observers. In various embodiments, a contact pressure on a back interaction surface of the device may serve as an additional input for passcode definition. See, for example, user interface **6900**.

[00888] As shown, user interface **6900** includes a numeric keypad **6902** for passcode entry. In various embodiments, when a user applies contact pressure on contact area **6904**, located on a back interaction surface, the numeric keypad changes. The exemplary interface in Figure **69** allows for four different sets of numeric keys, differentiated by small dots in the corners of the keys **6906**. The addition of a contact pressure component to passcode definition increases security because it is very difficult for an observer to ascertain how much pressure is being exerted by a finger placed on the back of the device.

[00889] Figure **70** shows a user interface **7000** for presenting a contextual menu, in accordance with one embodiment. As an option, the user interface **7000** may be implemented in the context of the architecture and environment of the previous Figures or any subsequent Figure(s). Of course, however, the user interface **7000** may be implemented in any desired environment. The aforementioned definitions may apply during the present description.

[00890] As shown, user interface **7000** may be utilized to present a context based menu for the user. In one embodiment, a user may interact with a display element **7002** by exerting contact pressure on a front interaction surface over contact area **7004**. At the same time, the user may be in contact with a back interaction surface, over contact area **7006**. In various embodiments, contextual menu **7008** may be displayed near the backtouch contact, after the user performs a predefined static gesture within contact area **7004**. The user may then select a menu item **7010** using small movements and contact pressure applied on the back interaction surface.

[00891] In the various embodiments described above, different combinations of user input have been associated with different actions. It should be understood that the described user inputs may be associated with any action, function, operation or interface, just as the described actions may be triggered by any combination of user inputs, according to different embodiments.

[00892] The various embodiments set forth herein may be implemented in a variety of devices including, but not limited to, consumer devices, phones, cell phones, internet

phones, music players, video players, cameras, social interaction devices, radios, TVs, watches, personal communication devices, electronic wallets, smart jewelry, personal computers, tablets, laptop computers, embedded systems, electronic glasses, and/or any other device that includes one or more sensors or inputs. Possible inputs may include, but are not limited to, keyboard, mouse, touchscreen(s), touchpad(s), interaction surfaces, a joystick, touchwheel, touchknob, touchbutton, touchball, trackball, scroll wheel, thumb-mouse, switch, button, wheel, knob, ball, pedal, voice recognition, audio command, audio prompt, gaze control, eye tracking, head position sensing, facial and/or gestural and/or postural expression recognition, and/or other inputs and combinations of these. Possible sensors may include, but are not limited to, range sensors, scanners, magnetometers, GPS receivers, accelerometers, cameras, depth sensors, light-field cameras, ambient light sensors, pressure sensors, infra-red (IR) sensors, UV sensors, touch and/or proximity sensor(s), grasp sensors, material sensors, chemical sensors, physical sensors, electrical sensors, biological sensors, weight sensors, force sensors, mass sensors, gas sensors, vapor sensors, particle sensors, wireless sensors, RF and other electromagnetic sensors, and/or other sensors and combinations of these.

[00893] In various embodiments, the device may include one or more different forms of displays and/or other outputs including, but not limited to, physical or other actuators, motors, tactile feedback or other tactile output, weight or mass feedback, force feedback or other force outputs, mechanical outputs, audio outputs, alarms, horns, bells, indicators, dials, meters, barcodes and/or other display patterns, vibrations, ultrasonic outputs, wireless and RF outputs, optical and light outputs, avatars, visual outputs, multiscreen displays, multilayer displays, 3D displays, holographic projections and outputs, laser displays and other laser outputs, projection systems, heads-up displays, and/or other outputs or combinations of these.

[00894] In various embodiments, the device may support one or more types of applications including, but not limited to, search applications, contacts and/or friends applications, messaging applications, telephone applications, video conferencing applications, e-mail applications, communications applications, voice recognition applications, instant messaging (IM) applications, blog and/or blogging applications,

photographic applications, shopping applications, payment applications, digital camera applications, digital video camera applications, web browsing and browser applications, digital music player applications, digital video player applications, cloud applications, office productivity applications, and/or other types of applications or combinations or multiple instances of these.

[00895] In other embodiments, different applications on the device may use one or more interface devices that may be physical interface devices or virtual interface devices including, but not limited to, touchscreens, touchpads, soft inputs, hard inputs, keyboards, buttons, knobs, sliders, switches, and/or any other kind of interface device.

[00896] In various embodiments, the device may have a common physical architecture (i.e. display, touchscreen, etc.) such that the device may be used with different applications with one or more common user interfaces with one or more common properties (i.e. easy to use, simple, intuitive, transparent, etc.).

[00897] In various embodiments, user interfaces may include one or more physical interface devices (i.e. keyboard, joysticks, buttons, sliders, switches, knobs, other hard inputs, etc.) and/or one or more virtual or soft interface devices (i.e. soft keyboard, programmable buttons, UI sliders, UI knobs, check boxes, text fields, other soft inputs, etc.).

[00898] In various embodiments, the device may implement one or more types of keyboard (i.e. soft keyboard, physical keyboard, virtual keyboard, keypad, etc.). In one embodiment, the keyboard(s) may include standard keyboard arrangements (i.e. QWERTY configurations, etc.), and/or non-standard arrangements and/or symbols on the keys (i.e. soft keys, buttons, pixels, displayed icons etc.). In another embodiment, the keyboard(s) may include a reduced number of symbols (i.e. letters, icons, soft keys, etc.) compared to the number of keys on a conventional physical keyboard. In yet another embodiment, one or more alternative keyboards may make it easier for users to select one or more symbols in the keyboard, and thus, one or more corresponding representations (i.e. letter, number, special symbol, graphics, emoticon, abbreviation, slang, acronym, etc.). In still another embodiment, one or more keyboards and their associated contents

and/or layouts may be modified. For example, in one embodiment, one or more displayed symbols on the keyboard may be modified in accordance with application, application state, device state, settings (i.e. language, etc.), and/or user actions.

[00899] In various embodiments, one or more keyboards may be customized to one or more applications and/or users. For example, in one embodiment, one or more keyboard embodiments may be customized to one or more users based on factors including, but not limited to, application states, stored application history, web cookies, a word usage history (i.e. lexicography, slang, individual usage, etc.), language, country, and/or any other factor. In another embodiment, one or more keyboards may be modified to reduce the probability of user error when selecting one or more keys when using the soft keyboards.

[00900] In various embodiments, the device may have multiple functions or roles including, but not limited to, telephone, video conferencing, e-mail, instant messaging, blogging, digital photography, digital video, web browsing, digital music playing, social interaction, shopping, searching, and/or any other function or role. As an option, instructions for performing the device functions may be included in a computer readable storage medium, or as a computer program product configured for execution by one or more processors.

[00901] In various optional embodiments, the features, capabilities, and/or technology, etc. of the tablets, mobile devices, computing devices, networks, hardware, and/or software, etc. disclosed in the following patents/applications may or may not be incorporated into any of the embodiments disclosed herein: US7479949; US7748634; US20060017692; US20100188268; US20110145692; US7954101; US20070103454; US20100210329; US20070091825; US20060013219; US7916166; US20090213205; US20070296805; US20100205148; US20100188473; US7441196; US7894641; US7966578; US7479949; US Provisional Application NO. 61/470,336, filed 03-31-2011, titled "SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR UTILIZING IMAGE RECOGNITION TO PERFORM AN ACTION"; US Provisional Application NO. 61/470,391, filed 03-31-2011, titled "SYSTEM, METHOD, AND

COMPUTER PROGRAM PRODUCT FOR ENABLING A PERIPHERAL DEVICE TO UTILIZE FUNCTIONALITY ASSOCIATED WITH A MOBILE DEVICE"; and US Provisional Application NO. 61/569,213 , filed 12-09-2011 , titled "SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR MODIFYING CONTENT." Each of the foregoing patents/applications are hereby incorporated by reference in their entirety for all purposes.

[00902] While various embodiments have been described above, it should be understood that they have been presented by way of example only, and not limitation. Thus, the breadth and scope of a preferred embodiment should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

CLAIMS

What is claimed is:

1. A computer program product embodied on a non-transitory computer readable medium, comprising:

code for receiving a pressure signal indicative of a magnitude of pressure being applied to a touch interface positioned on a first face of a device that includes a display on a second face thereof; and

code for altering a user experience as a function of the magnitude of the pressure being applied to the touch interface, utilizing the pressure signal.

2. The computer program product as set forth in claim 1, wherein the touch interface includes a touch screen.

3. The computer program product as set forth in claim 1, wherein the first face and the second face are on opposite faces of the device of the device.

4. The computer program product as set forth in claim 1, wherein the first face includes a side face and the second face includes a front face of the device.

5. The computer program product as set forth in claim 1, wherein the first face includes a back face and the second face includes a front face of the device.

6. The computer program product as set forth in claim 1, wherein the user experience is altered by causing input to the device.

7. The computer program product as set forth in claim 1, wherein the user experience is altered by causing output from the device.

8. The computer program product as set forth in claim 1, wherein the user experience is altered by causing processing by the device.

9. A device, comprising:

a touch interface positioned on a first face of the device, for receiving a pressure signal indicative of a magnitude of pressure being applied to the touch interface;

a display positioned on a second face of the device; and

a processor for altering a user experience as a function of the magnitude of the pressure being applied to the touch interface, utilizing the pressure signal.

10. A method, comprising:

receiving a pressure signal indicative of a magnitude of pressure being applied to a touch interface positioned on a first face of a device that includes a display on a second face thereof; and

altering a user experience as a function of the magnitude of the pressure being applied to the touch interface, utilizing the pressure signal.

11. A computer program product embodied on a non-transitory computer readable medium, comprising:

code for receiving a pressure signal indicative of a magnitude of pressure being applied to a touch screen; and

code for altering a user experience as a function of the magnitude of the pressure being applied to the touch screen, utilizing the pressure signal.

12. A device, comprising:

a touch screen for receiving a pressure signal indicative of a magnitude of pressure being applied to the touch screen; and

a processor for altering a user experience as a function of the magnitude of the pressure being applied to the touch screen, utilizing the pressure signal.

13. A method, comprising:

receiving a pressure signal indicative of a magnitude of pressure being applied to a touch screen; and

altering a user experience as a function of the magnitude of the pressure being applied to the touch screen, utilizing the pressure signal.

14. A computer program product embodied on a non-transitory computer readable medium, comprising:

code for receiving a touch signal indicative of a touch interface being touched, where the touch interface is positioned on a first face of a device that includes a display on a second face thereof; and

code for altering a user experience, utilizing the touch signal.

15. A device, comprising:

a touch interface positioned on a first face of the device, for receiving a touch signal indicative of the touch interface being touched;

a display positioned on a second face of the device; and

a processor for altering a user experience, utilizing the touch signal.

16. A method, comprising:

receiving a touch signal indicative of a touch interface being touched, where the touch interface is positioned on a first face of a device that includes a display on a second face thereof; and

altering a user experience, utilizing the touch signal.

ABSTRACT

A system, method, and computer program product are provided for a touch or pressure signal-based interface. In operation, a touch or pressure signal is received in association with a touch interface of a device. To this end, a user experience is altered, utilizing the signal.