



US 20080309632A1

(19) **United States**

(12) **Patent Application Publication**
Westerman et al.

(10) **Pub. No.: US 2008/0309632 A1**

(43) **Pub. Date: Dec. 18, 2008**

(54) **PINCH-THROW AND TRANSLATION
GESTURES**

(75) Inventors: **Wayne Carl Westerman**, San
Francisco, CA (US); **Brett G.
Alten**, Cupertino, CA (US)

Correspondence Address:
APPLE C/O MORRISON AND FOERSTER, LLP
LOS ANGELES
555 WEST FIFTH STREET SUITE 3500
LOS ANGELES, CA 90013-1024 (US)

(73) Assignee: **Apple Inc.**, Cupertino, CA (US)

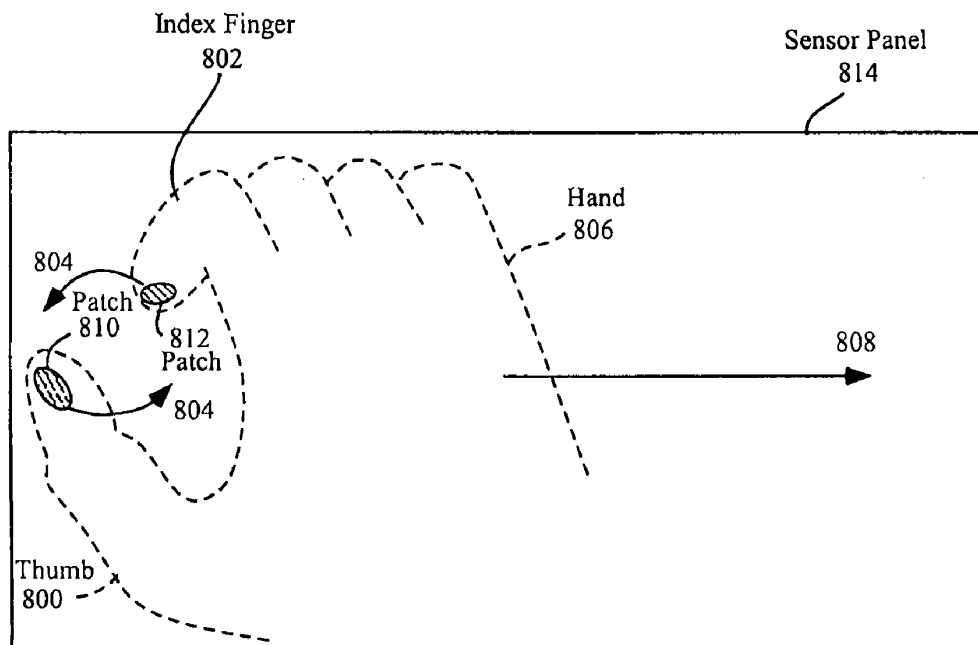
(21) Appl. No.: **11/818,479**

(22) Filed: **Jun. 13, 2007**

Publication Classification

(51) **Int. Cl.**
G06K 9/22 (2006.01)
G06F 3/033 (2006.01)
G06K 9/20 (2006.01)
(52) **U.S. Cl. 345/173; 381/386; 382/195; 455/422.1**
(57) **ABSTRACT**

The detection of finger pinch, rotate, and tap gestures along with a translation and optionally liftoff motion to initiate certain actions is disclosed. To detect both the gesture and the translation, a certain amount of gesture scaling speed can be detected along with a certain amount of translation speed and distance traveled. For a finger pinch gesture, the scaling speed can be computed as the dot product of the velocity vectors of two or more fingers coming together. For a finger rotation gesture, the scaling speed can be computed as a cross product of the velocity vectors of the rotating fingers. The translation speed of a gesture can be computed as the average of the velocity vectors of any fingers involved in the gesture. The amount of gesture scaling speed and translation speed needed to trigger the recognition of a combined gesture with translation can be a predetermined ratio.



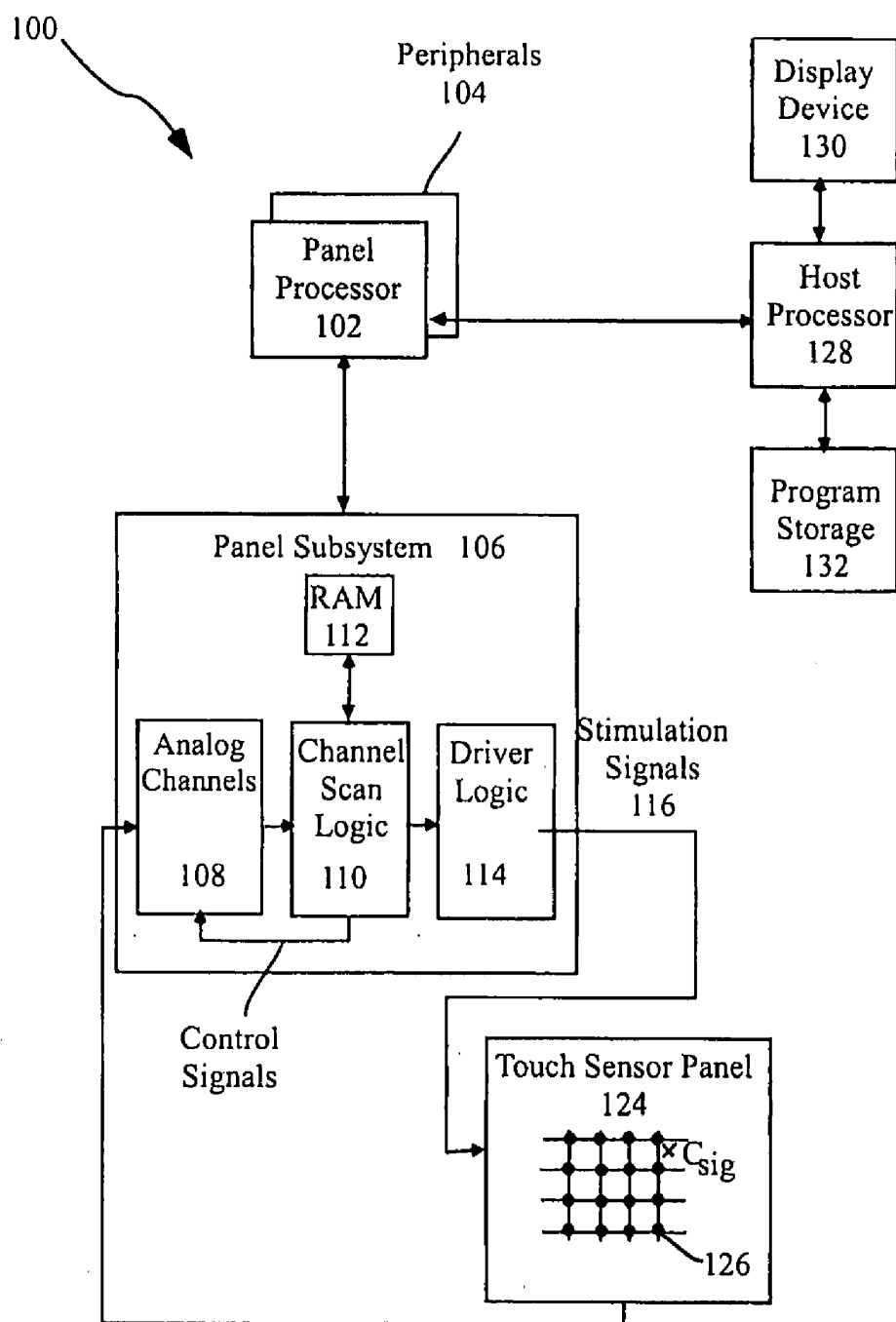


Fig. 1A

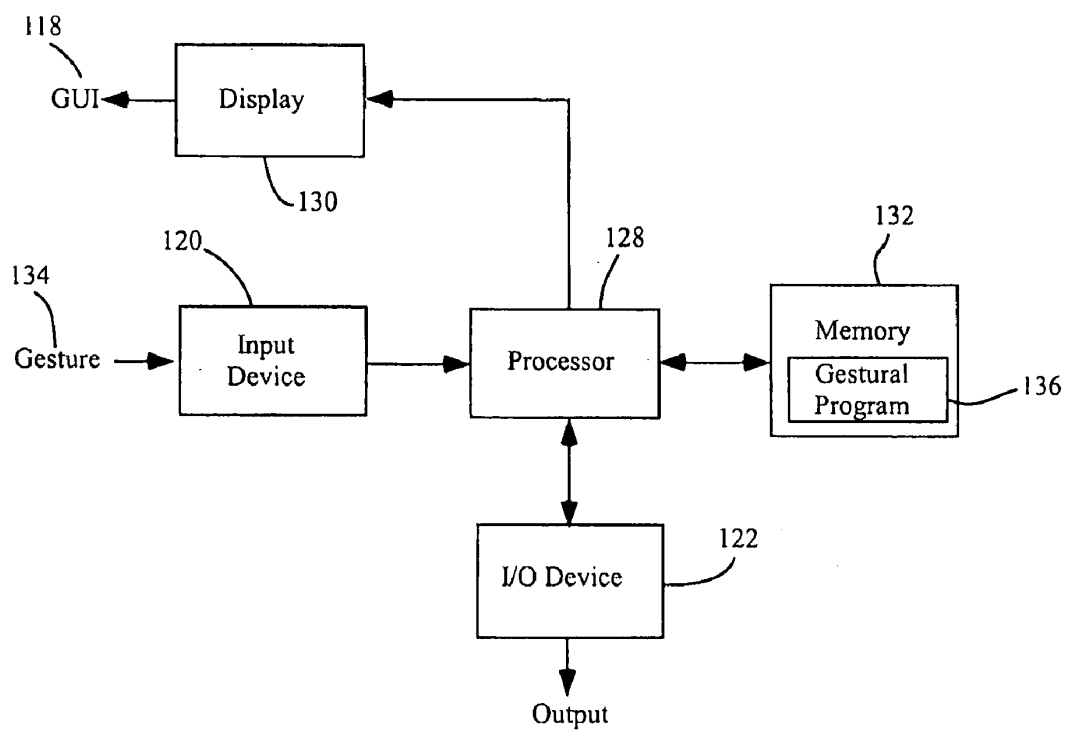


Fig. 1B

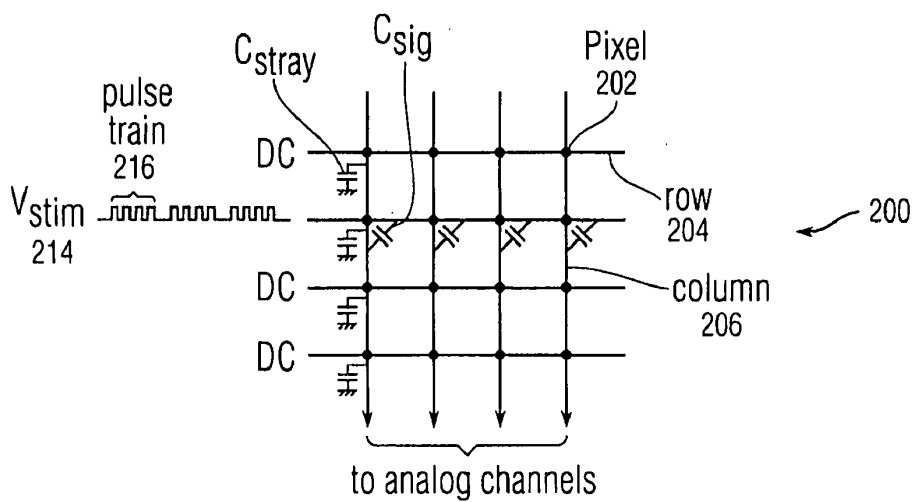


Fig. 2a

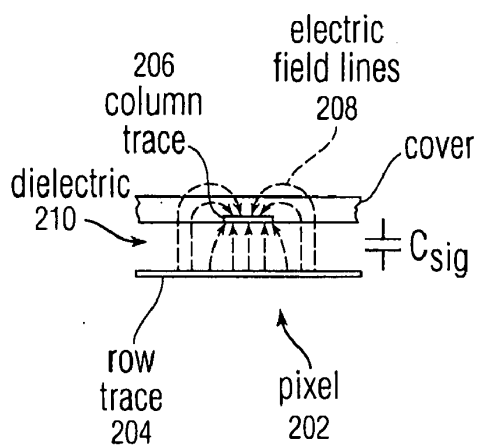


Fig. 2b

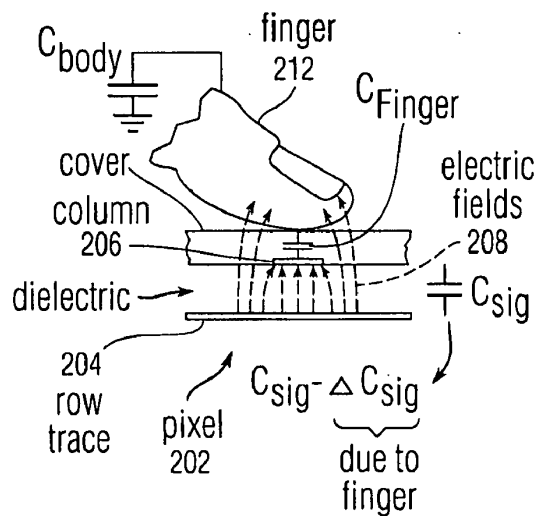


Fig. 2c

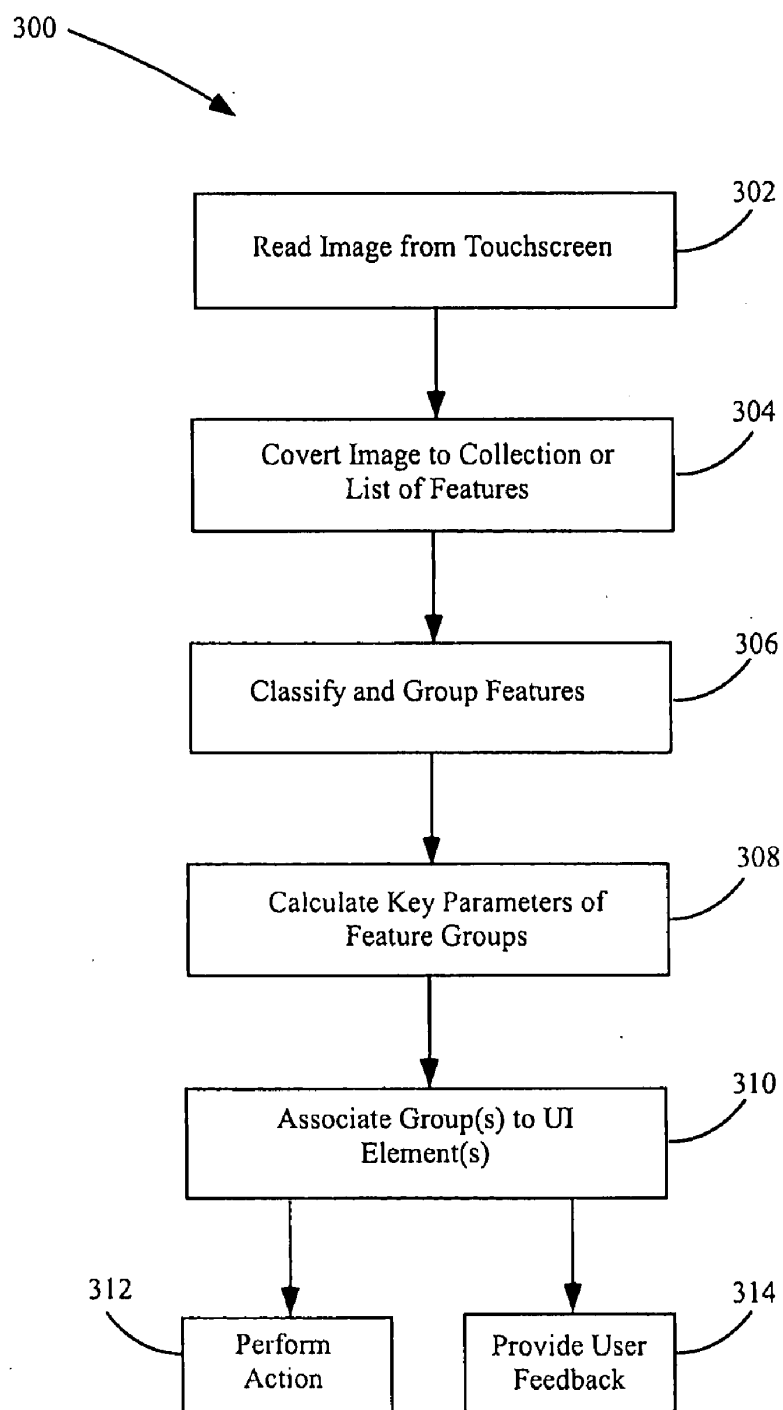


Fig. 3

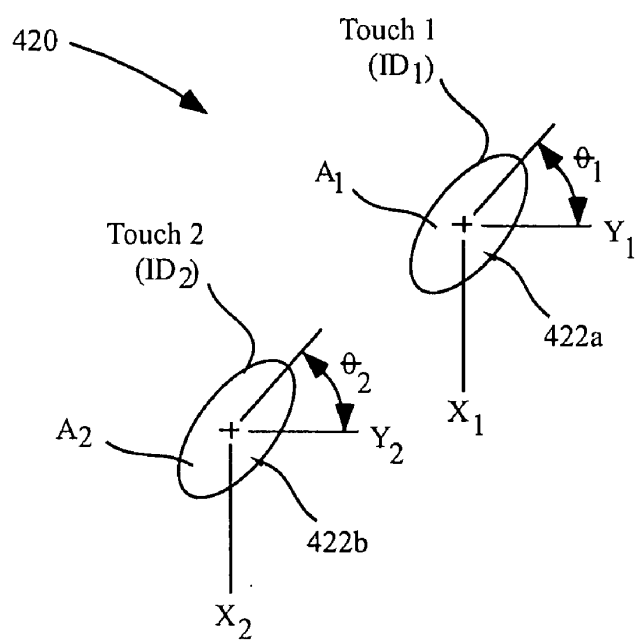


Fig. 4A

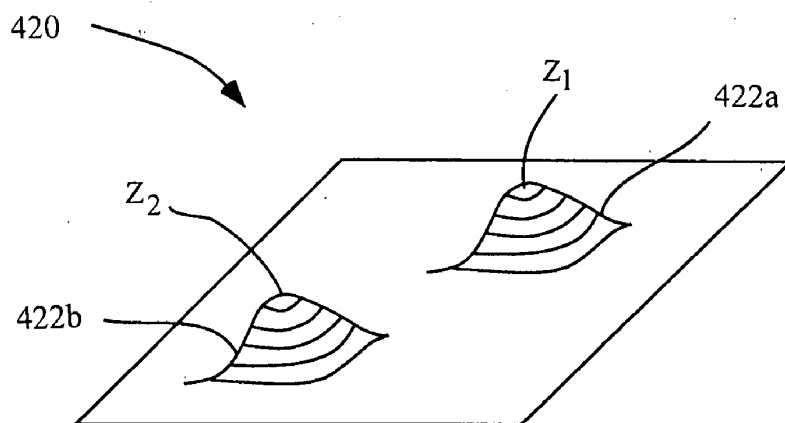


Fig. 4B

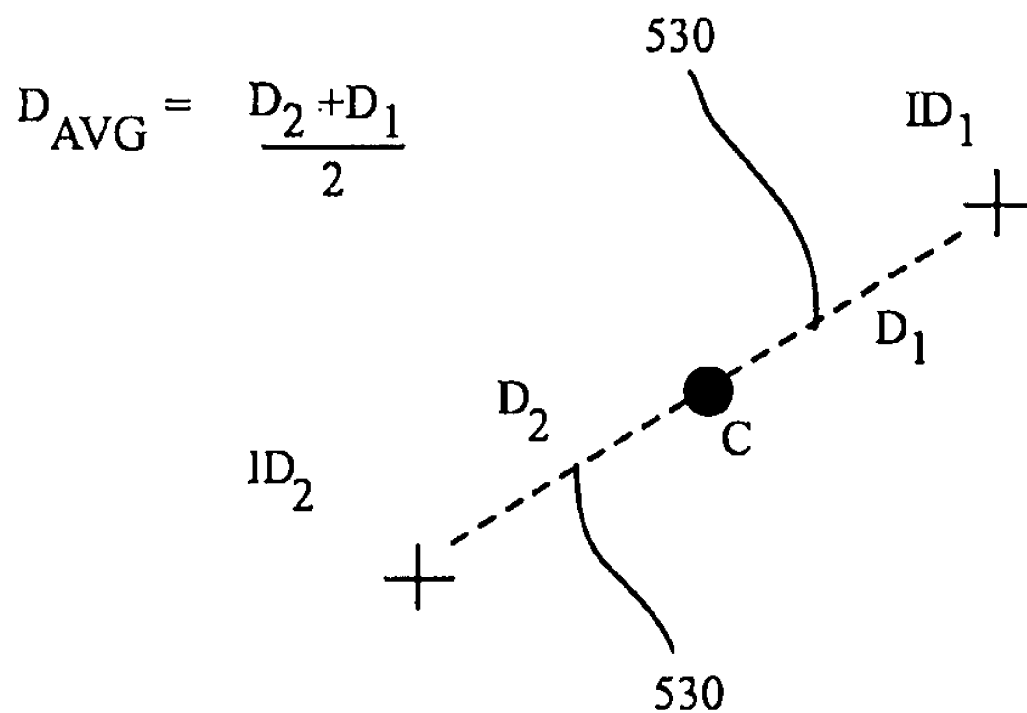


Fig. 5

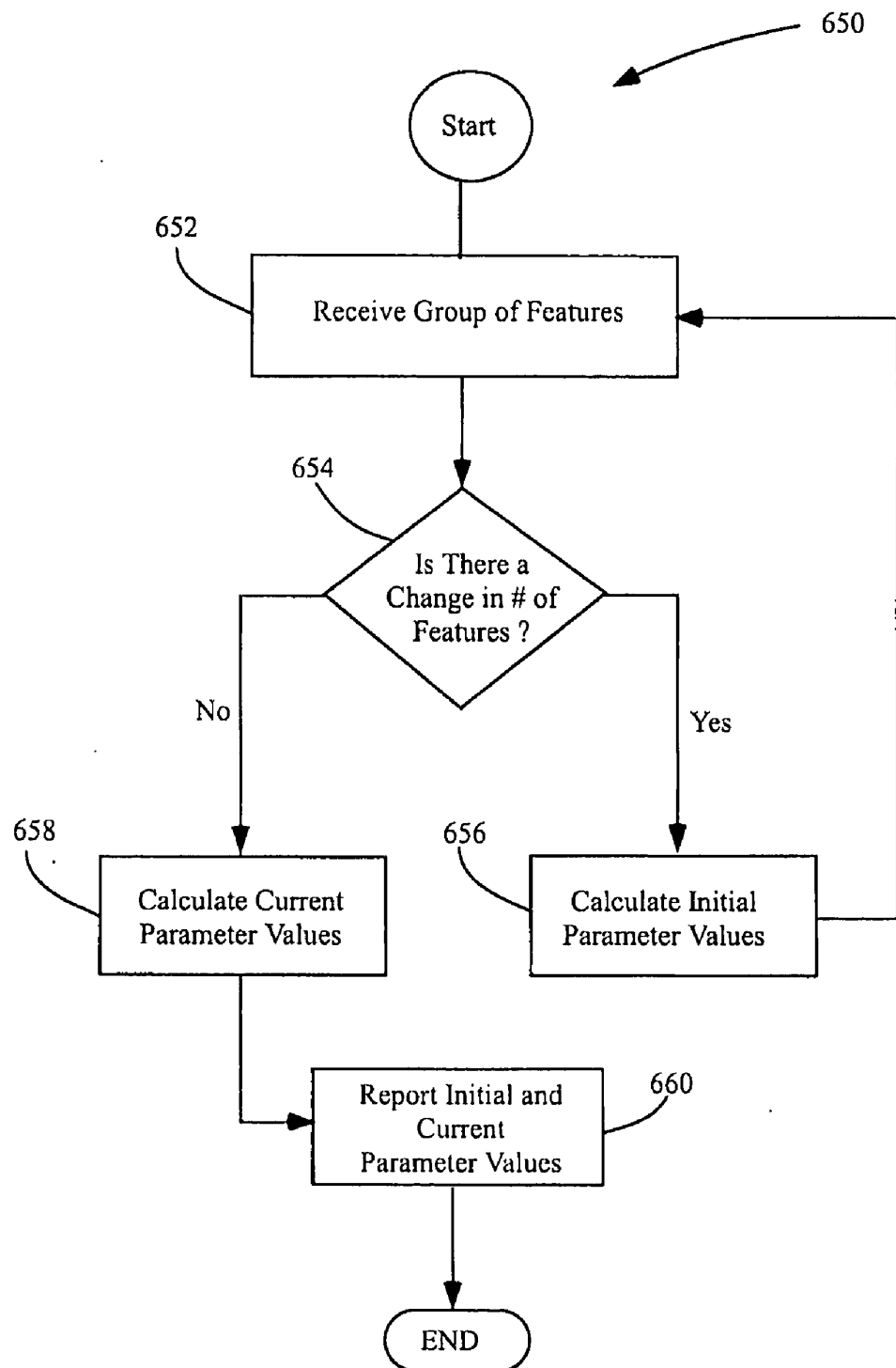


Fig. 6

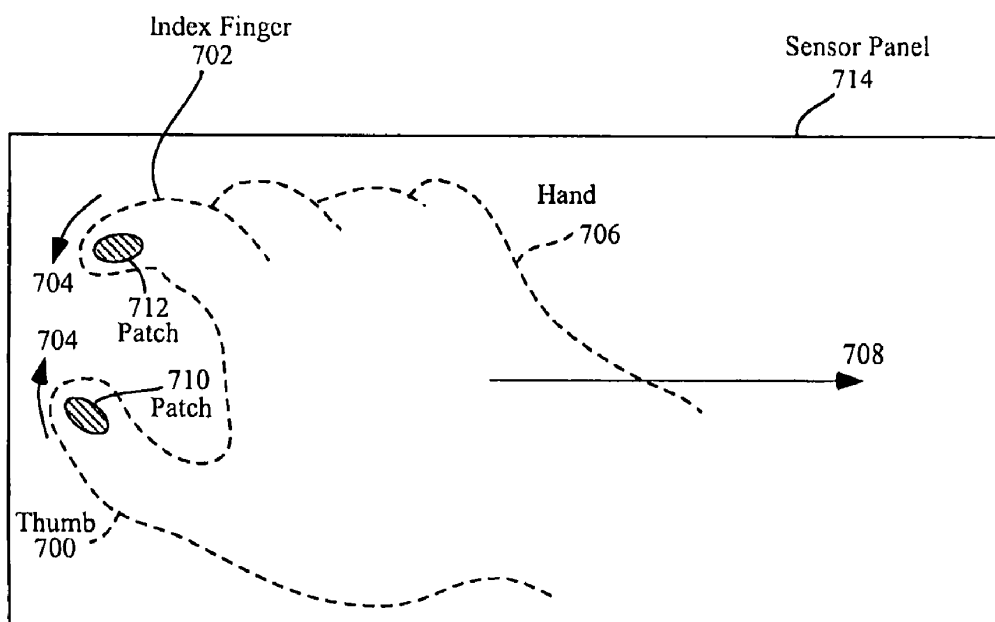


Fig. 7A

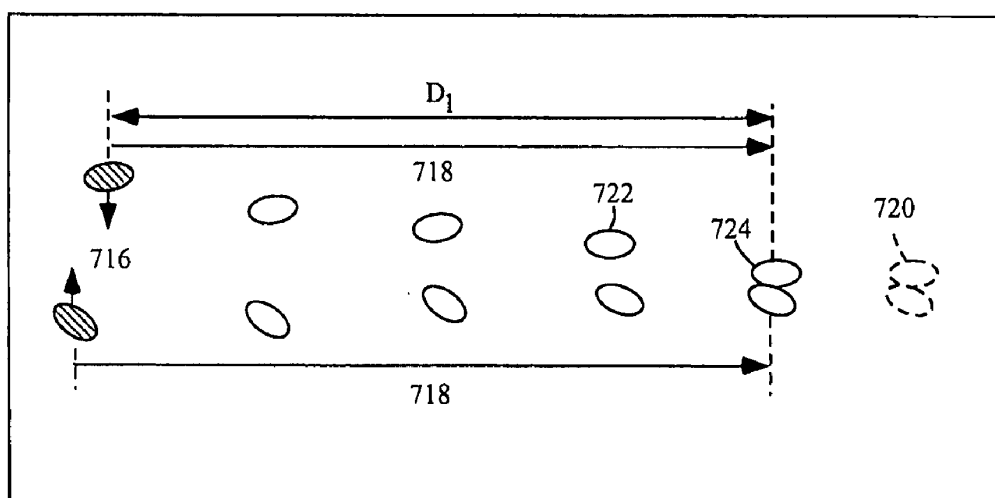


Fig. 7B

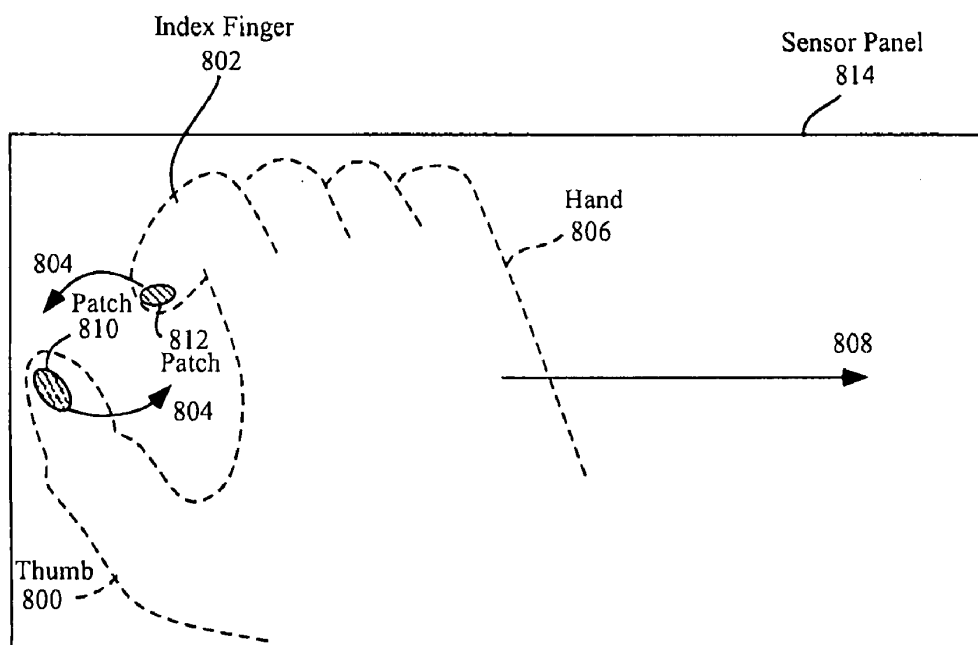


Fig. 8A

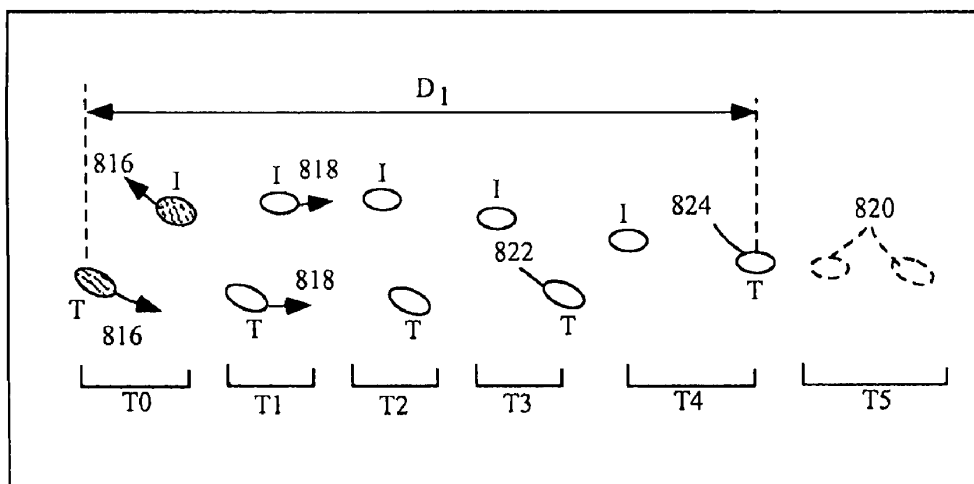


Fig. 8B

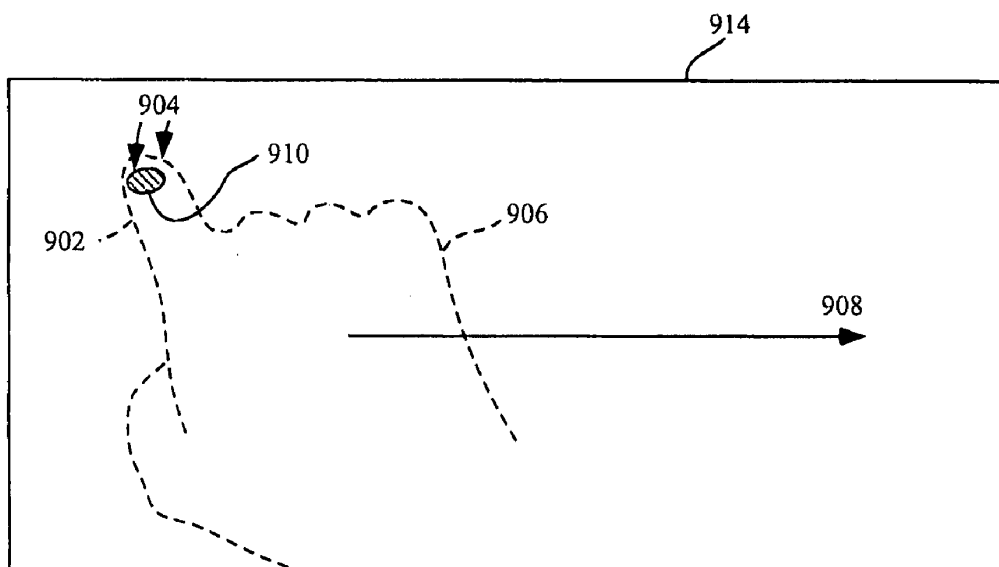


Fig. 9A

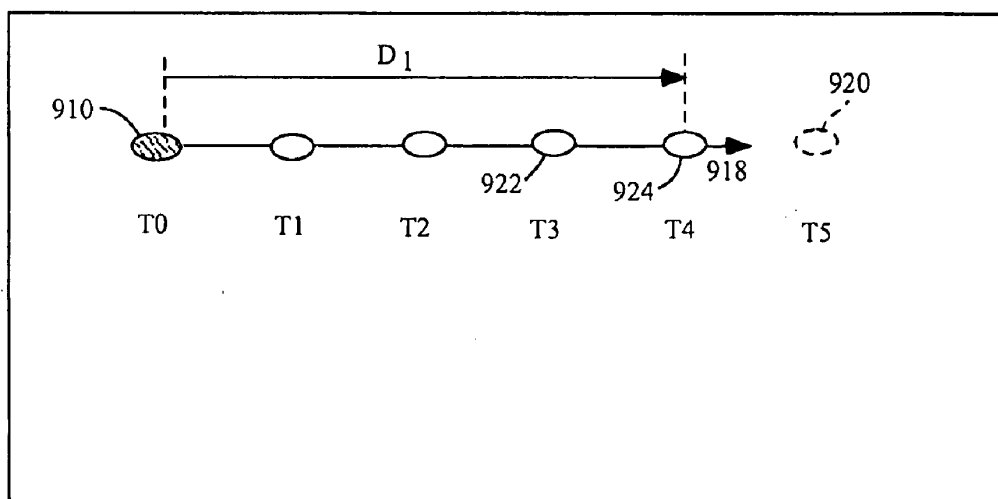


Fig. 9B

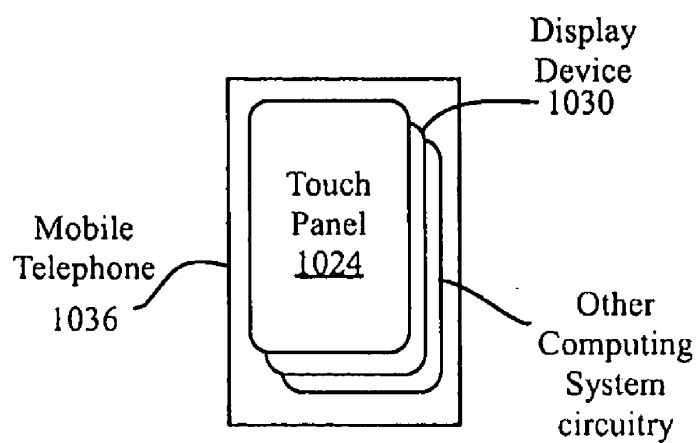


Fig. 10A

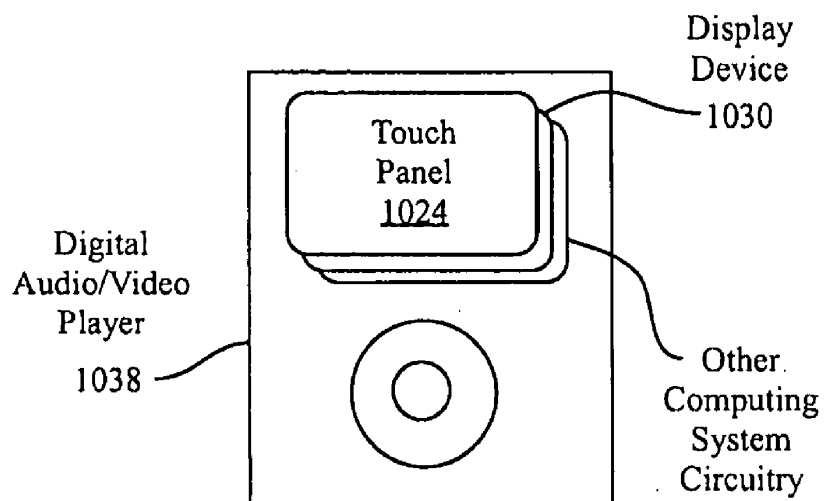


Fig. 10B

PINCH-THROW AND TRANSLATION GESTURES

FIELD OF THE INVENTION

[0001] This relates to finger gestures detectable at the surface of a touch sensor panel, and more particularly, to the detection of finger pinch, rotate, and tap gestures along with translation and optionally liftoff motion to initiate certain actions.

BACKGROUND OF THE INVENTION

[0002] There exist today many styles of input devices for performing operations in a computer system. The operations generally correspond to moving a cursor and making selections on a display screen. The operations can also include paging, scrolling, panning, zooming, etc. By way of example, the input devices can include buttons, switches, keyboards, mice, trackballs, touch pads, joy sticks, touch screens and the like. Each of these devices has advantages and disadvantages that can be taken into account when designing a computer system.

[0003] With touch pad instruments such as touch pads on a personal laptop computer, the movement of the input pointer on a display generally corresponds to the relative movements of the user's finger (or stylus) as the finger is moved along a surface of the touch pad. Touch screens, on the other hand, are a type of display screen that can include a touch-sensitive transparent panel (or "skin") that can overlay the display screen. When using a touch screen, a user typically makes a selection on the display screen by pointing directly to objects (such as graphical user interface (GUI) objects) displayed on the screen (usually with a stylus or finger).

[0004] To provide additional functionality, finger and hand gestures have been implemented with some of these input devices. By way of example, in touch pads, selections can be made when one or more taps are detected on the surface of the touch pad. In some cases, any portion of the touch pad can be tapped, and in other cases a dedicated portion of the touch pad can be tapped. In addition to selections, scrolling can be initiated by using finger motion at the edge of the touch pad.

[0005] U.S. Pat. Nos. 5,612,719 and 5,590,219, assigned to Apple Computer, Inc. (which has since changed its name to Apple Inc.) describe some other uses of gesturing. U.S. Pat. No. 5,612,719 discloses an onscreen button that is responsive to at least two different button gestures made on the screen on or near the button. U.S. Pat. No. 5,590,219 discloses a method for recognizing an ellipse-type gesture input on a display screen of a computer system.

[0006] In recent times, more advanced gestures have been implemented. For example, scrolling can be initiated by placing four fingers on the touch pad so that the scrolling gesture is recognized, and thereafter moving these fingers on the touch pad to perform scrolling events. These advanced gestures, however, can be in many instances counterintuitive.

SUMMARY OF THE INVENTION

[0007] This relates to the detection of finger pinch, rotate, and tap gestures along with a translation and optionally liftoff motion to initiate certain actions. These gestures can allow a user to more efficiently and accurately effect intended operations. To detect both the gesture and the translation, a certain amount of gesture scaling speed can be detected, where applicable, along with a certain amount of translation speed and

distance traveled. For a finger pinch gesture, the scaling speed can be computed as the dot product of the velocity vectors of two or more fingers coming together. For a finger rotation gesture, the scaling speed can be computed as a cross product of the velocity vectors of the rotating fingers. The translation speed of a gesture can be computed as the average of the velocity vectors of any fingers involved in the gesture. The amount of gesture scaling speed and translation speed needed to trigger the recognition of a combined gesture with translation can be a predetermined ratio. In addition, the translation distance can be integrated over time, and if the distance exceeds a predetermined threshold, the combined gesture/translation can be recognized as an actionable gesture.

[0008] If finger liftoff is part of the combined gesture/translation, liftoff can be detected as an absence of an image of touch (or a reduced image of touch indicative of fingers now hovering over the sensor panel rather than touching it) in a particular location of the sensor panel as compared to previously captured images of touch. Speed at the time of liftoff can be determined by comparing finger patch positions in frames just prior to liftoff to determine velocity at liftoff. If the velocity of sliding fingers at liftoff exceeds a predetermined threshold, the combined gesture/translation/liftoff can be recognized as an actionable gesture.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] FIG. 1a illustrates an exemplary computing system that can detect finger pinch, rotate, and tap gestures along with translation and liftoff motions to initiate certain actions according to one embodiment of this invention.

[0010] FIG. 1b is a block diagram of the exemplary host processor of FIG. 1a and connected devices according to one embodiment of this invention.

[0011] FIG. 2a illustrates an exemplary mutual capacitance touch sensor panel according to one embodiment of this invention.

[0012] FIG. 2b is a side view of an exemplary pixel in a steady-state (no-touch) condition according to one embodiment of this invention.

[0013] FIG. 2c is a side view of an exemplary pixel in a dynamic (touch) condition according to one embodiment of this invention.

[0014] FIG. 3 is an exemplary multipoint processing method according to one embodiment of this invention.

[0015] FIGS. 4a and 4b illustrate an exemplary image in time according to one embodiment of this invention.

[0016] FIG. 5 illustrates an exemplary group of features according to one embodiment of this invention.

[0017] FIG. 6 illustrates an exemplary parameter calculation method according to one embodiment of this invention.

[0018] FIG. 7a illustrates an exemplary combined finger pinch/translation gesture according to one embodiment of this invention.

[0019] FIG. 7b illustrates exemplary superimposed images of touch at times T0-T4 as a combined finger pinch/translation gesture is being performed according to one embodiment of this invention.

[0020] FIG. 8a illustrates an exemplary combined finger rotation/translation gesture according to one embodiment of this invention.

[0021] FIG. 8b illustrates exemplary superimposed images of touch at times T0-T4 as a combined finger rotation/translation gesture is being performed according to one embodiment of this invention.

[0022] FIG. 9a illustrates an exemplary combined finger tap/translation gesture according to one embodiment of this invention.

[0023] FIG. 9b illustrates exemplary superimposed images of touch or hover at times T0-T4 as a combined finger tap/translation gesture is being performed according to one embodiment of this invention.

[0024] FIG. 10a illustrates an exemplary mobile telephone that can include a touch sensor panel, display device, and other computing system blocks in the computing system of FIG. 1 that can detect the combined gesture/translation/liftoff operations according to one embodiment of this invention.

[0025] FIG. 10b illustrates an exemplary digital audio/video player that can include a touch sensor panel, display device, and other computing system blocks in the computing system of FIG. 1 that can detect the combined gesture/translation/liftoff operations according to one embodiment of this invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0026] In the following description of preferred embodiments, reference is made to the accompanying drawings which form a part hereof, and in which it is shown by way of illustration specific embodiments in which the invention can be practiced. It is to be understood that other embodiments can be used and structural changes can be made without departing from the scope of the embodiments of this invention.

[0027] This relates to the detection of finger pinch, rotate, and tap gestures along with a translation and optionally liftoff motion to initiate certain actions. These gestures can allow a user to more efficiently and accurately effect intended operations. To detect both the gesture and the translation, a certain amount of gesture scaling speed can be detected, where applicable, along with a certain amount of translation speed and distance traveled. For a finger pinch gesture, the scaling speed can be computed as the dot product of the velocity vectors of two or more fingers coming together. For a finger rotation gesture, the scaling speed can be computed as a cross product of the velocity vectors of the rotating fingers. The translation speed of a gesture can be computed as the average of the velocity vectors of any fingers involved in the gesture. The amount of gesture scaling speed and translation speed needed to trigger the recognition of a combined gesture with translation can be a predetermined ratio. In addition, the translation distance can be integrated over time, and if the distance exceeds a predetermined threshold, the combined gesture/translation can be recognized as an actionable gesture.

[0028] If finger liftoff is part of the combined gesture/translation, liftoff can be detected as an absence of an image of touch (or a reduced image of touch indicative of fingers now hovering over the sensor panel rather than touching it) in a particular location of the sensor panel as compared to previously captured images of touch. Speed at the time of liftoff can be determined by comparing finger patch positions in frames just prior to liftoff to determine velocity at liftoff. If the velocity of sliding fingers at liftoff exceeds a predetermined threshold, the combined gesture/translation/liftoff can be recognized as an actionable gesture.

[0029] Although some embodiments of this invention may be described generally herein in terms of touch sensor panels, it should be understood that embodiments of the invention can be applicable to touchscreens and touch tablets (touch

sensor panels without a display device), both of which can include far-field proximity sensors in addition to touch sensors.

[0030] FIG. 1a illustrates exemplary computing system 100 that can detect pinch-throw, rotate, and click gestures along with a translation and liftoff motion to initiate certain actions according to embodiments of the invention. Computing system 100 can include one or more panel processors 102 and peripherals 104, and panel subsystem 106. One or more panel processors 102 can include, for example, ARM968 processors or other processors with similar functionality and capabilities. However, in other embodiments, the panel processor functionality can be implemented instead by dedicated logic, such as a state machine. One or more panel processors 102 or host processor 128 can execute software or firmware implementing the algorithm for distinguishing sloppy taps from sliding motions according to embodiments of the invention. Peripherals 104 can include, but are not limited to, random access memory (RAM) or other types of memory or storage, watchdog timers and the like. Panel subsystem 106 can include, but is not limited to, one or more analog channels 108, channel scan logic 110 and driver logic 114. Channel scan logic 110 can access RAM 112, autonomously read data from the analog channels and provide control for the analog channels. In addition, channel scan logic 110 can control driver logic 114 to generate stimulation signals 116 at one or more frequencies and phases that can be selectively applied to rows of touch sensor panel 124. In some embodiments, panel subsystem 106, panel processor 102 and peripherals 104 can be integrated into a single application specific integrated circuit (ASIC).

[0031] Touch sensor panel 124 can include a capacitive sensing medium having a plurality of row traces or driving lines and a plurality of column traces or sensing lines, although other sensing media can also be used. The row and column traces can be formed from a transparent conductive medium such as Indium Tin Oxide (ITO) or Antimony Tin Oxide (ATO), although other transparent and non-transparent materials such as copper can also be used. In some embodiments, the row and column traces can be perpendicular to each other, although in other embodiments other non-Cartesian orientations are possible. For example, in a polar coordinate system, the sensing lines can be concentric circles and the driving lines can be radially extending lines (or vice versa). It should be understood, therefore, that the terms "row" and "column," "first dimension" and "second dimension," or "first axis" and "second axis" as used herein are intended to encompass not only orthogonal grids, but the intersecting traces of other geometric configurations having first and second dimensions (e.g. the concentric and radial lines of a polar-coordinate arrangement). The rows and columns can be formed on a single side of a substantially transparent substrate separated by a substantially transparent dielectric material, on opposite sides of the substrate, or on two separate substrates separated by the dielectric material.

[0032] At the "intersections" of the traces, where the traces pass above and below (cross) each other (but do not make direct electrical contact with each other), the traces can essentially form two electrodes (although more than two traces could intersect as well) representing a capacitive sensor. Each capacitive sensor can be viewed as picture element (pixel) 126, which can be particularly useful when touch sensor panel 124 is viewed as capturing an "image" of touch over a frame (one complete scan of the panel). (In other words, after

panel subsystem **106** has determined whether a touch event has been detected at each touch sensor in the touch sensor panel, the pattern of touch sensors in the multi-touch panel at which a touch event occurred can be viewed as an “image” of touch (e.g. a pattern of fingers touching the panel.) The capacitance between row and column electrodes appears as a stray capacitance when the given row is held at direct current (DC) voltage levels and as a mutual signal capacitance Csig when the given row is stimulated with an alternating current (AC) signal. The presence of a finger or other object near or on the touch sensor panel can be detected by measuring changes to a signal charge Qsig present at the pixels being touched, which is a function of Csig. Each column of touch sensor panel **124** can drive one or more analog channels **108** (also referred to herein as an event detection and demodulation circuit) in panel subsystem **106**. Each analog channel **108** can generate a value representative of an amount of touch being sensed at the connected column, which can be indicative of a touch event at one or more pixels along that column. Touch sensor panel **124** can include single-touch or multi-touch sensor panels, the latter of which is described in Applicant’s co-pending U.S. application Ser. No. 10/842,862 entitled “Multipoint Touchscreen,” filed on May 6, 2004 and published as U.S. Published Application No. 2006/0097991 on May 11, 2006, the contents of which are incorporated by reference herein.

[0033] Computing system **100** can correspond to a personal computer system, such as a desktop, laptop, tablet or handheld computer. Computing system **100** can also correspond to a computing device, such as a mobile telephone, personal digital assistant (PDA), dedicated media player, consumer electronic device, and the like. Computing system **100** can include host processor **128** for receiving outputs from panel processor **102** and performing actions based on the outputs that can include, but are not limited to, moving an object such as a cursor or pointer, scrolling or panning, adjusting control settings, opening a file or document, viewing a menu, making a selection, executing instructions, operating a peripheral device connected to the host device, answering a telephone call, placing a telephone call, terminating a telephone call, changing the volume or audio settings, storing information related to telephone communications such as addresses, frequently dialed numbers, received calls, missed calls, logging onto a computer or a computer network, permitting authorized individuals access to restricted areas of the computer or computer network, loading a user profile associated with a user’s preferred arrangement of the computer desktop, permitting access to web content, launching a particular program, encrypting or decoding a message, and/or the like. Host processor **128** can also perform additional functions that may not be related to panel processing, and can be coupled to program storage **132** and display device **130** such as an LCD display for providing a UI to a user of the device.

[0034] FIG. 1b is a block diagram of the exemplary host processor **128** of FIG. 1a and connected devices according to embodiments of the invention. Host processor **128** can be configured to execute instructions and to carry out operations associated with computing system **100**. For example, using instructions retrieved from program storage **132**, host processor **100** can control the reception and manipulation of input and output data between components of computing system **100**. Host processor **128** can be implemented on a single-chip, multiple chips or multiple electrical components. For example, various architectures can be used for host processor

128, including a dedicated or embedded processor, a single purpose processor, controller, application specific integrated circuit (ASIC), and so forth.

[0035] In most cases, processor **128** together with an operating system (OS) can operate to execute computer code and produce and use data. OSs are generally well known and will not be described in greater detail. By way of example, the OS can correspond to OS/2, DOS, Unix, Linux, Palm OS, and the like. The OS can also be a special purpose OS, such as can be used for limited purpose appliance-type computing devices. The OS, other computer code and data can reside within memory block **132** that is operatively coupled to processor **128**. Memory block **132** can generally provide a place to store computer code and data that are used by computing system **100**. By way of example, memory block **132** can include Read-Only Memory (ROM), RAM, one or more hard disk drives and/or the like. The information can also reside on a removable storage medium and loaded or installed onto computing system **100** when needed. Removable storage mediums can include, for example, CD-ROMs, PC-CARDS, memory cards, floppy disks, magnetic tape, and a network component.

[0036] Computing system **100** can also include display device **130** that can be operatively coupled to processor **128**. Display device **130** can be a liquid crystal display (LCD) (e.g., active matrix, passive matrix and the like). Alternatively, display device **130** can be a monitor such as a monochrome display, color graphics adapter (CGA) display, enhanced graphics adapter (EGA) display, variable-graphics-array (VGA) display, super VGA display, cathode ray tube (CRT), and the like. Display device **130** can also correspond to a plasma display or a display implemented with electronic inks.

[0037] Display device **130** can be configured to display GUI **118** that can provide an easy to use interface between a user of the computer system and the operating system or application running thereon. Generally speaking, GUI **118** can represent programs, files and operational options with graphical images, objects, or vector representations. The graphical images can include windows, fields, dialog boxes, menus, icons, buttons, cursors, scroll bars, etc. Such images can be arranged in predefined layouts, or can be created dynamically to serve the specific actions being taken by a user. During operation, the user can select and/or activate various graphical images in order to initiate functions and tasks associated therewith. By way of example, a user can select a button that opens, closes, minimizes, or maximizes a window, or an icon that launches a particular program. GUI **118** can additionally or alternatively display information, such as non interactive text and graphics, for the user on display device **130**.

[0038] Computing system **100** can also include input device **120** that can be operatively coupled to processor **128**. Input device **120** can be configured to transfer data from the outside world into computing system **100**. Input device **120** can, for example, be used to perform tracking and to make selections with respect to GUI **118** on display **130**. Input device **120** can also be used to issue commands in computing system **100**. Input device **120** can include a touch sensing device such as touch sensor panel **124**, configured to receive input from a user’s touch and send this information to processor **128** through panel subsystem **106**. In many cases, the touch-sensing device can recognize touches as well as the position and magnitude of touches on a touch sensitive sur-

face. The touch sensing device can detect and report the touches to processor 128, and processor 128 can interpret the touches in accordance with its programming. For example, processor 128 can initiate a task in accordance with a particular touch. A dedicated processor can be used to process touches locally and reduce demand for the main processor of the computing system.

[0039] The touch sensing device can be based on sensing technologies including but not limited to capacitive sensing, resistive sensing, surface acoustic wave sensing, pressure sensing, optical sensing, and/or the like. Furthermore, the touch sensing means can be based on single point sensing or multipoint sensing. Single point sensing is capable of only distinguishing a single touch, while multipoint sensing is capable of distinguishing multiple touches that occur at the same time.

[0040] As discussed above, input device 120 can be a touch screen that can be positioned over or in front of display 130, integrated with display device 130, or can be a separate component, such as a touch pad.

[0041] Computing system 100 can also include capabilities for coupling to one or more I/O devices 122. By way of example, I/O devices 122 can correspond to keyboards, printers, scanners, cameras, microphones, speakers, and/or the like. I/O devices 122 can be integrated with computing system 100 or they can be separate components (e.g., peripheral devices). In some cases, I/O devices 122 can be connected to computing system 190 through wired connections (e.g., cables/ports). In other cases, I/O devices 122 can be connected to computing system 100 through wireless connections. By way of example, the data link can correspond to PS/2, USB, IR, Firewire, RF, Bluetooth or the like.

[0042] According to embodiments of the invention, computing system 100 can be designed to recognize gestures 134 applied to input device 120 and to control aspects of computing system 100 based on the gestures. In some cases, a gesture can be defined as a stylized interaction with an input device that can be mapped to one or more specific computing operations. Gestures 134 can be made through various hand, and more particularly finger motions. Alternatively or additionally, the gestures can be made with a stylus. In all of these cases, input device 120 can receive gestures 134, and processor 128 can execute instructions to carry out operations associated with the gestures 134. In addition, memory block 132 can include gesture operational program 136, which can be part of the OS or a separate application. Gesture operation program 136 can generally include a set of instructions that can recognize the occurrence of gestures 134 and can inform one or more software agents of the gestures and/or what action(s) to take in response to the gestures. Additional details regarding the various gestures that can be used as input commands are discussed further below.

[0043] Upon a user performing one or more gestures, input device 120 can relay gesture information to processor 128. Using instructions from memory 132, and more particularly, gesture operational program 136, processor 128 can interpret the gestures 134 and control different components of computing system 100, such as memory 132, display 130 and I/O devices 122, based on the gestures. Gestures 134 can be identified as commands for performing actions in applications stored in memory 132, modifying image objects shown on display 130, modifying data stored in memory 132, and/or for performing actions in I/O devices 122.

[0044] Note that although FIG. 1b illustrates input device 120 and display 130 as two separate boxes for illustration purposes, the two boxes can be realized on one device. It should also be noted that, while FIG. 1a illustrates dedicated panel processor 102, panel subsystem 106 can be controlled directly by the host processor 128. Additionally, it should also be noted that touch sensor panel 124 and display device 130 can be integrated into a single touch screen display device.

[0045] FIG. 2a illustrates exemplary mutual capacitance touch sensor panel 200 according to embodiments of the invention. FIG. 2a indicates the presence of a stray capacitance C_{stray} at each pixel 202 located at the intersection of a row 204 and a column 206 trace (although C_{stray} for only one column is illustrated in FIG. 2 for purposes of simplifying the figure). In the example of FIG. 2a, AC stimuli V_{stim} 214, V_{stim} 215 and V_{stim} 217 can be applied to several rows, while other rows can be connected to DC. V_{stim} 214, V_{stim} 215 and V_{stim} 217 can be at different frequencies and phases, as will be explained later. Each stimulation signal on a row can cause a charge $Q_{sig} = C_{sig} \times V_{stim}$ to be injected into the columns through the mutual capacitance present at the affected pixels. A change in the injected charge (Q_{sig_sense}) can be detected when a finger, palm or other object is present at one or more of the affected pixels. V_{stim} signals 214, 215 and 217 can include one or more pulse trains 216, and each pulse train can include a particular number of a number of pulses. Although pulse trains 216 are shown as square waves, other waveshapes such as sine waves can also be employed. Note that although FIG. 2a illustrates rows 204 and columns 206 as being substantially perpendicular, they need not be so aligned, as described above. As described above, each column 206 can be connected to an analog channel (see analog channels 108 in FIG. 1).

[0046] FIG. 2b is a side view of exemplary pixel 202 in a steady-state (no-touch) condition according to embodiments of the invention. In FIG. 2b, an electric field of electric field lines 208 of the mutual capacitance between column 206 and row 204 traces or electrodes separated by dielectric 210 is shown.

[0047] FIG. 2c is a side view of exemplary pixel 202 in a dynamic (touch) condition. In FIG. 2c, finger 212 has been placed near pixel 202. Finger 212 is a low-impedance object at signal frequencies, and has an AC capacitance C_{finger} from the column trace 204 to the body. The body has a self-capacitance to ground C_{body} of about 200 pF, where C_{body} is much larger than C_{finger} . If finger 212 blocks some electric field lines 208 between the row and column electrodes (those fringing fields that exit the dielectric and pass through the air above the row electrode), those electric field lines are shunted to ground through the capacitance path inherent in the finger and the body, and as a result, the steady state signal capacitance C_{sig} is reduced by ΔC_{sig} . In other words, the combined body and finger capacitance act to reduce C_{sig} by an amount ΔC_{sig} (which can also be referred to herein as C_{sig_sense}), and can act as a shunt or dynamic return path to ground, blocking some of the electric fields as resulting in a reduced net signal capacitance. The signal capacitance at the pixel becomes $C_{sig} - \Delta C_{sig}$, where C_{sig} represents the static (no touch) component and ΔC_{sig} represents the dynamic (touch) component. Note that $C_{sig} - \Delta C_{sig}$ may always be nonzero due to the inability of a finger, palm or other object to block all electric fields, especially those electric fields that remain entirely within the dielectric material. In addition, it should be understood that as a finger is pushed harder or more com-

pletely onto the multi-touch panel, the finger can tend to flatten, blocking more and more of the electric fields, and thus ΔC_{sig} can be variable and representative of how completely the finger is pushing down on the panel (i.e. a range from “no-touch” to “full-touch”).

[0048] Further details of multi-touch sensor detection, including proximity detection by a touch panel, are described in commonly assigned and co-pending (1) U.S. application Ser. No. 10/840,862 entitled “Multipoint Touchscreen,” which was published on May 11, 2006 as U.S. Publication No. US2006/0097991, (2) U.S. application Ser. No. 11/428,522 entitled “Identifying Contacts On A Touch Surface,” which was published on Oct. 26, 2006 as U.S. Publication No. 2006/0238522, and (3) U.S. application Ser. No. 11/649,998 entitled “Proximity and Multi-Touch Sensor Detection and Demodulation,” filed on Jan. 3, 2007, the entirety of each of which is hereby incorporated herein by reference.

[0049] FIG. 3 illustrates multipoint processing method **300** in accordance with embodiments of the invention. Multipoint processing method **300** can, for example, be performed with the system shown in FIG. 1a or FIG. 1b. Multipoint processing method **300** generally begins at block **302** where images can be read from a multipoint input device, and more particularly a multipoint touch screen. Although the term “image” may be used, it should be noted that the data can come in other forms. In most cases, the image read from the touch sensor panel can provide magnitude (Z) as a function of position (X and Y) for each sensing point or pixel of the touch sensor panel. The magnitude can, for example, reflect the capacitance measured at each point.

[0050] Following block **302**, multipoint processing method **300** proceeds to block **304**, where the image can be converted into a collection or list of features. Each feature can represent a distinct input such as a touch. In most cases, each feature can include its own unique identifier (ID), x coordinate, y coordinate, Z magnitude, angle Θ , area A, and the like.

[0051] FIGS. 4a and 4b illustrate an exemplary image **420** in time according to embodiments of the invention. In image **420**, there are two features **422** based on two distinct touches. The touches can for example be formed from a pair of fingers touching the touch screen. As shown, each feature **422** can include unique identifier (ID), x coordinate, y coordinate, Z magnitude, angle Θ , and area A. More particularly, the first feature **422A** can be represented by $ID_1, X_1, Y_1, Z_1, \Theta_1, A_1$ and the second feature **422B** can be represented by $ID_2, X_2, Y_2, Z_2, \Theta_2, A_2$. This data can be outputted for example using a multi-touch protocol.

[0052] The conversion from data or images to features can be accomplished using methods described in copending U.S. application Ser. No. 10/840,862 titled “Multipoint Touchscreen.” As disclosed therein, the raw data can be received in a digitized form, and can include values for each node of the touch screen. The values can be between 0 and 256 where 0 equates to no touch pressure and 256 equates to full touch pressure. Thereafter, the raw data can be filtered to reduce noise. Once filtered, gradient data, which indicates the topology of each group of connected points, can be generated. Thereafter, the boundaries for touch regions can be calculated based on the gradient data (i.e., a determination can be made as to which points are grouped together to form each touch region). By way of example, a watershed algorithm can be used. Once the boundaries are determined, the data for each of the touch regions can be calculated (e.g., X, Y, Z, Θ , A).

[0053] Referring again to FIG. 3, following block **304**, multipoint processing method **300** proceeds to block **306** where feature classification and groupings can be performed. During classification, the identity of each of the features can be determined. For example, the features can be classified as a particular finger, thumb, palm or other object. Once classified, the features can be grouped. The manner in which the groups are formed can widely vary. In most cases, the features can be grouped based on some criteria (e.g., they carry a similar attribute). For example, the two features shown in FIG. 4a and FIG. 4b can be grouped together because each of these features is located in proximity to each other or because they are from the same hand. The grouping can include some level of filtering to filter out features that are not part of the touch event. In filtering, one or more features can be rejected because they either meet some predefined criteria or because they do not meet some predefined criteria. By way of example, one of the features can be classified as a thumb located at the edge of a tablet PC. Because the thumb is being used to hold the device rather than being used to perform a task, the feature generated therefrom can be rejected, i.e., is not considered part of the touch event being processed.

[0054] Following block **306**, multipoint processing method **300** proceeds to block **308** where key parameters for the feature groups can be calculated. The key parameters can include distance between features, X/Y centroid of all features, feature rotation, total pressure of the group (e.g., pressure at centroid), and the like. As shown in FIG. 5, the calculation can include finding the centroid C, drawing a virtual line **530** to each feature from the centroid C, defining the distance D for each virtual line (D_1 and D_2), and then averaging the distances D_1 and D_2 . Once the parameters are calculated, the parameter values can be reported. The parameter values can be typically reported with a group identifier (GID) and number of features within each group (in this case three). In most cases, both initial and current parameter values can be reported. The initial parameter values can be based on set down, i.e., when the user sets their fingers on the touch screen, and the current values can be based on any point within a stroke occurring after set down.

[0055] Referring again to FIG. 3, blocks **302-308** can be repetitively performed during a user stroke thereby generating a plurality of sequentially configured signals. The initial and current parameters can be compared in later steps to perform actions in the system.

[0056] Following block **308**, the process flow proceeds to block **310** where the group can be associated with a user interface (UI) element. UI elements can be buttons boxes, lists, sliders, wheels, knobs, etc. Each UI element can represent a component or control of the user interface. The application behind the UI element(s) can have access to the parameter data calculated in block **308**. In one implementation, the application can rank the relevance of the touch data to the UI element corresponding thereto. The ranking can be based on some predetermined criteria. The ranking can include producing a figure of merit and, whichever UI element has the highest figure of merit, giving it sole access to the group. There can even be some degree of hysteresis as well (e.g., once one of the UI elements claims control of that group, the group sticks with the UI element until another UI element has a much higher ranking). By way of example, the ranking can include determining proximity of the centroid (or features) to the image object associated with the UI element.

[0057] Following block 310, multipoint processing method 300 proceeds to blocks 312 and 314. Blocks 312 and 314 can be performed approximately at the same time. From the user perspective, in one embodiment, blocks 312 and 314 appear to be performed concurrently. In block 312, one or more actions can be performed based on differences between initial and current parameter values, and can also be based to a UI element to which they are associated, if any. In block 314, user feedback pertaining to the one or more action being performed can be provided. By way of example, user feedback can include display, audio, tactile feedback and/or the like.

[0058] FIG. 6 illustrates a parameter calculation method 650 in accordance with embodiments of the invention. Parameter calculation method 650 can, for example, correspond to block 308 shown in FIG. 3. The parameter calculation method 650 generally begins at block 652 where a group of features can be received. Following block 652, the parameter calculation method 650 proceeds to block 654 where a determination can be made as to whether or not the number of features in the group of features has changed. For example, the number of features can have changed due to the user picking up or placing an additional finger. Different fingers can be needed to perform different controls (e.g., tracking, gesturing). If the number of features has changed, the parameter calculation method 650 proceeds to block 656 where the initial parameter values can be calculated. If the number stays the same, the parameter calculation method 650 proceeds to block 658 where the current parameter values can be calculated. Thereafter, the parameter calculation method 650 proceeds to block 660 where the initial and current parameter values can be reported. By way of example, the initial parameter values can contain the average initial distance between points (or Distance (AVG) initial) and the current parameter values can contain the average current distance between points (or Distance (AVG) current). These can be compared in subsequent steps in order to control various aspects of a computer system.

[0059] The above methods and techniques can be used to implement any number of GUI interface objects and actions. For example, finger pinching, rotation and tapping gestures (collectively referred to as finger motions) combined with translation and optionally liftoff can be created to effect user commands to select an object, move an object, scroll an object, and the like. The detection and implementation of such gestures can be performed by a processor executing firmware or software.

[0060] Finger pinch. FIG. 7a illustrates an exemplary combined finger pinch/translation gesture according to embodiments of the invention. In FIG. 7a, a user's thumb 700 and index finger 702 (although other fingers could be used), while in contact with touch sensor panel 714, can be brought together (see arrows 704) at the same time that the entire hand 706 is moved (translated) to the right (see arrow 708) (although the translation can occur in other directions, such as left, up, down, or diagonally). Thumb 700 and index finger 702 can cause patches 710 and 712 to appear in an image of touch gathered from a scan of sensor panel 714.

[0061] FIG. 7b illustrates exemplary superimposed images of touch at times T0-T4 as a combined finger pinch/translation gesture is being performed according to embodiments of the invention. To detect the images of touch as a finger pinch, actual images of touch captured over time can be compared to expected images of touch. For example, in FIG. 7b two actual

images of touch can be expected to move closer together over time. To detect both the pinch gesture and the translation, a certain amount of gesture scaling speed can be detected along with a certain amount of translation speed. The scaling speed can be computed as the dot product of the velocity vectors (see vectors 716) of the two or more fingers coming together (in the Y-direction in the example of FIG. 7b). The translation speed can be computed as the average of the velocity vectors (see vectors 718) of any fingers involved in the gesture (in the X-direction in the example of FIG. 7b). Computation of the translation speed and scaling speed of finger pinch gestures is described in U.S. Pat. No. 6,323,846 entitled "Method and Apparatus for Integrating Manual Input," the contents of which are incorporated by reference herein for all purposes.

[0062] The amount of gesture scaling speed and translation speed needed to recognize a combined finger pinch/translation gesture with can be a predetermined ratio. For example, in one embodiment the scaling speed of the fingers coming together can be required to be at least 30% of the translation speed of the fingers in order for a combined finger pinch/translation gesture to be recognized. In addition, the translation distance (D1 in the example of FIG. 7b) can be integrated over time, and if it exceeds a predetermined threshold, the combined finger pinch/translation gesture can be recognized as an actionable gesture.

[0063] Because a user is unlikely to translate exactly in the X-direction, for example, there can be some allowable jitter in the Y-direction coming through on the dot-products and cross-products (off-axis jitter). In some embodiments, translation can be limited to the X direction, which can be more natural for a user and easier to implement, and any translation in the Y direction can be ignored. Alternatively, translation movements can be limited to a certain axis or cone starting from the original touch areas.

[0064] In some embodiments, the combined finger pinch/translation gesture can be combined with finger liftoff. Liftoff itself can be detected as an absence of an image of touch (or a reduced image of touch indicative of fingers now hovering over the sensor panel rather than touching it) in a particular location of the sensor panel as compared to previously captured images of touch. In the example of FIG. 7b, liftoff can be detected at time T5 when no image of touch or a reduced image of touch (see patches 720) is detected. Speed at the time of liftoff can be determined by comparing finger patch positions in one or more frames (e.g. patches 722 and 724 at times T3 and T4) just prior to liftoff to determine velocity at liftoff. Detecting the velocity of fingers at liftoff is described in U.S. Pat. No. 6,323,846 entitled "Method and Apparatus for Integrating Manual Input." If the velocity of sliding fingers at lift-off is above a predetermined threshold, the combined finger pinch/translation/liftoff gesture can be recognized.

[0065] A combined finger pinch/translation gesture in the proximity of a GUI object can indicate grabbing the GUI object and moving it. For example, a user can pinch the GUI object to grab it while dragging it to a desired location by translating the pinching fingers along the sensor panel surface to a desired drop location.

[0066] Because a combined finger pinch/translation/liftoff gesture is a "dramatic" gesture, it can be used to throw a GUI object in the trash or to another desktop or window, minimize a window (pinch to minimize while translating to send the minimized window down to a corner), or send a file to a printer (pinch to select while translating to a printer icon), and the like. Another embodiment can be a pinch and pick up

gesture, wherein proximity sensors detect the direction of the hand after pickup. A screen may then be activated in that direction (either highlighted or opened).

[0067] A combined finger pinch/translation/liftoff gesture can also be used to continue certain operations, such as the continuation of a scrolling motion. For example, in the case of scrolling, if the velocity of sliding fingers at lift-off is above a predetermined threshold, an artificial momentum vector can be computed (velocity multiplied by mass), where the “mass” of item being scrolled can be assigned in any number of ways, including the number of items in a list, the number of lines in a file, the amount of force being applied by the fingers (as determined by the size of the images of touch) when the scrolling gesture was recognized, and the like. The momentum can then gradually decay over time in an auto-regressively smooth decline until scrolling stops completely, perhaps with some bouncing before stopping.

[0068] Finger rotation. FIG. 8a illustrates an exemplary combined finger rotation/translation gesture according to embodiments of the invention. In FIG. 8a, a user’s thumb 800 and index finger 802 (although other fingers could be used), while in contact with touch sensor panel 814, can be rotated around an imaginary axis (see arrows 804) at the same time that the entire hand 806 is moved (translated) to the right (see arrow 808) (although the translation can occur in other directions, such as left, up, down, or diagonally). Thumb 800 and index finger 802 cause patches 810 and 812 to appear in an image of touch gathered from a scan of sensor panel 814.

[0069] FIG. 8b illustrates exemplary superimposed images of touch at times T0-T4 as a combined finger rotation/translation gesture is being performed according to embodiments of the invention. To detect the images of touch as a finger rotation, actual images of touch captured over time can be compared to expected images of touch. For example, in FIG. 8b each of the two actual images of touch can be expected to rotate in the same direction (clockwise or counterclockwise) over time. To detect both the rotation gesture and the translation, a certain amount of gesture rotation speed can be detected along with a certain amount of translation speed. The rotation speed can be computed as the cross product of the velocity vectors of the rotating fingers (see vectors 816) of the two or more rotating fingers. The translation speed can be computed as the average of the velocity vectors (see vectors 818) of any fingers involved in the gesture (in the X-direction in the example of FIG. 8b). Computation of the translation speed and rotation speed of rotational gestures is described in U.S. Pat. No. 6,323,846 entitled “Method and Apparatus for Integrating Manual Input.”

[0070] The amount of gesture rotation speed and translation speed needed to recognize a combined finger pinch/translation gesture with can be a predetermined ratio. For example, in one embodiment the rotation speed of the rotating fingers can be required to be at least 30% of the translation speed of the fingers in order for a combined finger rotation/translation gesture to be recognized. In addition, the translation distance (D1 in the example of FIG. 8b) can be integrated over time, and if it exceeds a predetermined threshold, the combined finger rotation/translation gesture can be recognized as an actionable gesture.

[0071] In some embodiments, the combined finger rotation/translation gesture can be combined with finger liftoff. In the example of FIG. 8b, liftoff can be detected at time T5 when no image of touch or a reduced image of touch (see patches 80) is detected. Speed at the time of liftoff can be

determined by comparing finger patch positions in one or more frames (e.g. patches 822 and 824 at times T3 and T4) just prior to liftoff to determine velocity at liftoff. If the velocity of sliding fingers at lift-off is above a predetermined threshold, the combined finger pinch/translation/liftoff gesture can be recognized.

[0072] A combined finger rotation/translation/liftoff gesture in the proximity of a GUI object can be used to spin the GUI object off to a virtual desktop.

[0073] Finger taps. FIG. 9a illustrates an exemplary combined finger tap/translation gesture according to embodiments of the invention. In FIG. 9a, a user’s index finger 902 (although other fingers could be used) can tap touch sensor panel 914 one or more times (see arrows 904) followed by a translation of the entire hand 906 to the right (see arrow 908) (although the translation can occur in other directions, such as left, up, down, or diagonally). The one or more taps each can indicate different operations. The one or more taps can end with finger 902 raised above touch sensor panel 914, or with the finger touching the panel. Index finger 902 causes patch 910 to appear in an image of touch gathered from a scan of sensor panel 914.

[0074] FIG. 9b illustrates exemplary superimposed images of touch or hover at times T0-T4 as a combined finger tap/translation gesture is being performed according to embodiments of the invention. To detect both the tap gesture and the translation, one or more taps can be detected followed by a certain amount of translation speed. The one or more taps can be detected as one or more unique images of touch appearing and disappearing at about the same location on the sensor panel within a certain period of time (see location 910 in the example of FIG. 9b). The translation of the hand can occur with the finger hovering above sensor panel, in which case the images of touch captured at T1-T4 can represent a hovering finger, or with the finger touching the sensor panel, in which case the images of touch captured at T2-T4 can represent a finger touching the sensor panel. A hovering finger can be detected either by touch sensors having near-field proximity detection capabilities or proximity sensors having far-field proximity detection capabilities. Detection of hover events are described in greater detail in commonly assigned co-pending U.S. application Ser. No. 10/840,862 entitled “Multi-point Touchscreen,” published as U.S. Publication No. US2006/0097991, co-pending U.S. application Ser. No. 11/649,998 entitled “Proximity and Multi-Touch Sensor Detection and Demodulation,” filed on Jan. 3, 2007, co-pending U.S. application Ser. No. 11/619,595 entitled “Multi-Event Input System,” filed on Jan. 3, 2007, and U.S. application Ser. No. 11/619,464 entitled “Multi-Touch Input Discrimination,” filed on Jan. 3, 2007, the contents of each which are incorporated herein by reference.

[0075] The translation speed can be computed as the average of the velocity vector (see vector 918) of any fingers involved in the gesture (in the X-direction in the example of FIG. 9b). In addition, the translation distance (D1 in the example of FIG. 9b) can be integrated over time, and if the translation speed and distance exceeds predetermined thresholds, the combined finger tap/translation gesture can be recognized as an actionable gesture.

[0076] Because a user is unlikely to translate exactly in the X-direction, for example, there can be some allowable jitter in the Y-direction. In some embodiments, translation can be limited to the X direction, which can be more natural for a user and easier to implement, and any translation in the Y

direction can be ignored. Alternatively, translation movements can be limited to a certain axis or cone starting from the original touch areas.

[0077] In some embodiments, the combined finger tap/translation gesture can be combined with finger liftoff. Liftoff itself can be detected as an absence of an image of touch (or a reduced image of touch indicative of fingers now hovering over the sensor panel rather than touching it) in a particular location of the sensor panel as compared to previously captured images of touch. In the example of FIG. 9b, liftoff can be detected at time T5 when no image of touch or a reduced image of touch (see patch 920) is detected after having detected an actual image of touch in preceding frames T1-T4. Speed at the time of liftoff can be determined by comparing finger patch positions in one or more frames (e.g. patches 922 and 924 at times T3 and T4) just prior to liftoff to determine velocity at liftoff. If the velocity of the finger at lift-off is above a predetermined threshold, the combined finger tap/translation/liftoff gesture can be recognized.

[0078] A combined finger tap/translation gesture in the proximity of a GUI object can indicate selecting the GUI object and moving it. For example, a user can tap the GUI object to select it, and then drag it to a desired location by translating the finger along or above the sensor panel surface to a desired drop location.

[0079] Because a combined finger tap/translation/liftoff gesture is a “dramatic” gesture, it can be used to select (tap and hold) and throw a GUI object in the trash or to another desktop or window, minimize a window (tap and hold to select, and then translate to send the minimized window down to a corner), or send a file to a printer (tap to select, and then translate to a printer icon), and the like.

[0080] A combined finger tap/translation/liftoff gesture can also be used to continue certain operations, such as the continuation of a scrolling motion. For example, in the case of scrolling, if the velocity of the sliding finger at lift-off is above a predetermined threshold, an artificial momentum vector can be computed (velocity multiplied by mass), where the “mass” of item being scrolled can be assigned in any number of ways, including the number of items in a list, the number of lines in a file, the amount of force being applied by the finger (as determined by the size of the images of touch) when the scrolling gesture was recognized, and the like. The momentum can then gradually decay over time in an auto-regressively smooth decline until scrolling stops completely, perhaps with some bouncing before stopping.

[0081] FIG. 10a illustrates an exemplary mobile telephone 1036 that can include touch sensor panel 1024, display device 1030, and other computing system blocks in computing system 100 of FIG. 1 that can detect the combined gesture/translation/liftoff operations according to embodiments of the invention.

[0082] FIG. 10b illustrates an exemplary digital audio/video player 1038 that can include touch sensor panel 1024, display device 1030, and other computing system blocks in computing system 100 of FIG. 1 that can detect the combined gesture/translation/liftoff operations according to embodiments of the invention.

[0083] Although embodiments of this invention have been fully described with reference to the accompanying drawings, it is to be noted that various changes and modifications will become apparent to those skilled in the art. Such changes and

modifications are to be understood as being included within the scope of embodiments of this invention as defined by the appended claims.

What is claimed is:

1. A method for recognizing a combined finger motion and translation gesture on a sensor panel, comprising:
 - identifying a type of the finger motion by comparing actual images of touch captured over time to expected images of touch for various finger motion types;
 - computing a translation speed of one or more fingers involved in the finger motion; and
 - recognizing the finger motion and translation as a combined finger motion and translation gesture if the translation speed is greater than a first predetermined threshold.
2. The method of claim 1, further comprising computing the translation speed as an average of velocity vectors for the one or more fingers involved in the finger motion.
3. The method of claim 1, further comprising:
 - determining whether the actual images of touch captured over time are substantially similar to two or more finger patches moving towards each other; and
 - recognizing the finger motion and translation as a combined finger pinch and translation gesture if the actual images of touch captured over time are substantially similar to the two or more finger patches moving towards each other.
4. The method of claim 3, further comprising:
 - detecting a scaling speed by computing a dot product of velocity vectors for the two or more finger patches;
 - computing a ratio of the scaling speed to the translation speed; and
 - recognizing the finger motion and translation as a combined finger pinch and translation gesture only if additionally the ratio is greater than a second predetermined threshold.
5. The method of claim 3, further comprising:
 - determining a translation distance of the one or more fingers involved in the finger pinch; and
 - recognizing the finger motion and translation as a combined finger pinch and translation gesture only if additionally the translation distance is greater than a second predetermined threshold.
6. The method of claim 3, further comprising performing a drag and drop operation if the finger pinch occurs over a graphical user interface (GUI) object and the translation moves the actual images of touch to a drop location.
7. The method of claim 3, further comprising:
 - determining whether a liftoff of the one or more fingers occurred;
 - computing a liftoff velocity of the one or more fingers at a time of the liftoff; and
 - recognizing the finger motion, translation and liftoff as a combined finger pinch, translation and liftoff gesture if the liftoff velocity is greater than a second predetermined threshold.
8. The method of claim 7, further comprising performing a trash operation if the finger pinch occurs over a graphical user interface (GUI) object and the translation moves the actual images of touch towards a trash icon.
9. The method of claim 7, further comprising performing a move to another desktop operation if the finger pinch occurs

over a graphical user interface (GUI) object and the translation moves the actual images of touch towards another desktop.

10. The method of claim 7, further comprising performing a window minimize operation if the finger pinch occurs over a window and the translation moves the actual images of touch towards a corner of a desktop.

11. The method of claim 7, further comprising performing a print operation if the finger pinch occurs over a graphical user interface (GUI) object and the translation moves the actual images of touch towards a printer icon.

12. The method of claim 7, further comprising performing a scrolling operation if the finger pinch occurs over a graphical user interface (GUI) object and the translation moves the actual images of touch in a particular direction to be scrolled.

13. The method of claim 12, further comprising performing a scrolling operation with a gradual decay in a scrolling speed if the finger pinch occurs over a graphical user interface (GUI) object and the liftoff velocity is greater than a third predetermined threshold.

14. The method of claim 1, further comprising:
determining whether the actual images of touch captured over time are substantially similar to two or more finger patches rotating around an imaginary axis; and
recognizing the finger motion and translation as a combined finger rotation and translation gesture if the actual images of touch captured over time are substantially similar to the two or more finger patches rotating around the imaginary axis.

15. The method of claim 14, further comprising:
detecting a rotation speed by computing a cross product of velocity vectors for the two or more finger patches;
computing a ratio of the rotation speed to the translation speed; and
recognizing the finger motion and translation as a combined finger rotation and translation gesture only if additionally the ratio is greater than a second predetermined threshold.

16. The method of claim 14, further comprising:
determining a translation distance of the one or more fingers involved in the finger rotation; and
recognizing the finger motion and translation as a combined finger rotation and translation gesture only if additionally the translation distance is greater than a second predetermined threshold.

17. The method of claim 14, further comprising:
determining whether a liftoff of the one or more fingers occurred;
computing a liftoff velocity of the one or more fingers at a time of the liftoff; and
recognizing the finger motion, translation and liftoff as a combined finger rotation, translation and liftoff gesture if the liftoff velocity is greater than a second predetermined threshold.

18. The method of claim 17, further comprising performing a move to another desktop operation if the finger rotation occurs over a graphical user interface (GUI) object and the translation moves the actual images of touch towards another desktop.

19. The method of claim 1, further comprising:
determining whether the actual images of touch captured over time are substantially similar to one or more finger taps in substantially the same location within a predetermined period of time; and

recognizing the finger motion and translation as a combined finger tap and translation gesture if the actual images of touch captured over time are substantially similar to the one or more finger taps in substantially the same location within the predetermined period of time.

20. The method of claim 19, further comprising:
determining a translation distance of the one or more fingers involved in the finger tap; and

recognizing the finger motion and translation as a combined finger tap and translation gesture only if additionally the translation distance is greater than a second predetermined threshold.

21. The method of claim 19, further comprising performing a drag and drop operation if the finger tap occurs over a graphical user interface (GUI) object and the translation moves the actual images of touch to a drop location.

22. The method of claim 19, further comprising:
determining whether a liftoff of the one or more fingers occurred;
computing a liftoff velocity of the one or more fingers at a time of the liftoff; and

recognizing the finger motion, translation and liftoff as a combined finger tap, translation and liftoff gesture if the liftoff velocity is greater than a second predetermined threshold.

23. The method of claim 19, further comprising performing a trash operation if the finger tap occurs over a graphical user interface (GUI) object and the translation moves the actual images of touch towards a trash icon.

24. The method of claim 19, further comprising performing a move to another desktop operation if the finger tap occurs over a graphical user interface (GUI) object and the translation moves the actual images of touch towards another desktop.

25. The method of claim 19, further comprising performing a window minimize operation if the finger tap occurs over a window and the translation moves the actual images of touch towards a corner of a desktop.

26. The method of claim 19, further comprising performing a print operation if the finger tap occurs over a graphical user interface (GUI) object and the translation moves the actual images of touch towards a printer icon.

27. The method of claim 19, further comprising performing a scrolling operation if the finger tap occurs over a graphical user interface (GUI) object and the translation moves the actual images of touch in a particular direction to be scrolled.

28. The method of claim 27, further comprising performing a scrolling operation with a gradual decay in a scrolling speed if the finger tap occurs over a graphical user interface (GUI) object and the liftoff velocity is greater than a third predetermined threshold.

29. A method for recognizing a combined finger motion and translation gesture on a sensor panel, comprising:

identifying a type of the finger motion by comparing actual images of touch captured over time to expected images of touch for various finger motion types;

computing a translation distance of one or more fingers involved in the finger motion; and

recognizing the finger motion and translation as a combined finger motion and translation gesture if the translation distance is greater than a first predetermined threshold.

- 30.** The method of claim **29**, further comprising:
determining a translation speed of the one or more fingers involved in the finger pinch; and
recognizing the finger motion and translation as a combined finger pinch and translation gesture only if additionally the translation speed is greater than a second predetermined threshold.
- 31.** The method of claim **29**, further comprising:
determining whether the actual images of touch captured over time are substantially similar to two or more finger patches moving towards each other; and
recognizing the finger motion and translation as a combined finger pinch and translation gesture if the actual images of touch captured over time are substantially similar to the two or more finger patches moving towards each other.
- 32.** The method of claim **31**, further comprising:
detecting a scaling speed by computing a dot product of velocity vectors for the two or more finger patches;
computing a ratio of the scaling speed to the translation speed; and
recognizing the finger motion and translation as a combined finger pinch and translation gesture only if additionally the ratio is greater than a second predetermined threshold.
- 33.** The method of claim **29**, further comprising:
determining whether a liftoff of the one or more fingers occurred;
computing a liftoff velocity of the one or more fingers at a time of the liftoff; and
recognizing the finger motion, translation and liftoff as a combined finger pinch, translation and liftoff gesture if the liftoff velocity is greater than a second predetermined threshold.
- 34.** The method of claim **29**, further comprising:
determining whether the actual images of touch captured over time are substantially similar to two or more finger patches rotating around an imaginary axis; and
recognizing the finger motion and translation as a combined finger rotation and translation gesture if the actual images of touch captured over time are substantially similar to the two or more finger patches rotating around the imaginary axis.
- 35.** The method of claim **34**, further comprising:
detecting a rotation speed by computing a cross product of velocity vectors for the two or more finger patches;
computing a ratio of the rotation speed to the translation speed; and
recognizing the finger motion and translation as a combined finger rotation and translation gesture only if additionally the ratio is greater than a second predetermined threshold.
- 36.** A computer-readable medium comprising program code for recognizing a combined finger motion and translation gesture on a sensor panel, the program code for causing performance of a method comprising:
identifying a type of the finger motion by comparing actual images of touch captured over time to expected images of touch for various finger motion types;
computing a translation speed of one or more fingers involved in the finger motion; and
recognizing the finger motion and translation as a combined finger motion and translation gesture if the translation speed is greater than a first predetermined threshold.
- 37.** The computer-readable medium of claim **36**, the program code further for causing performance of a method comprising computing the translation speed as an average of velocity vectors for the one or more fingers involved in the finger motion.
- 38.** The computer-readable medium of claim **36**, the program code further for causing performance of a method comprising:
determining whether the actual images of touch captured over time are substantially similar to two or more finger patches moving towards each other; and
recognizing the finger motion and translation as a combined finger pinch and translation gesture if the actual images of touch captured over time are substantially similar to the two or more finger patches moving towards each other.
- 39.** The computer-readable medium of claim **38**, the program code further for causing performance of a method comprising:
detecting a scaling speed by computing a dot product of velocity vectors for the two or more finger patches;
computing a ratio of the scaling speed to the translation speed; and
recognizing the finger motion and translation as a combined finger pinch and translation gesture only if additionally the ratio is greater than a second predetermined threshold.
- 40.** The computer-readable medium of claim **38**, the program code further for causing performance of a method comprising:
determining a translation distance of the one or more fingers involved in the finger pinch; and
recognizing the finger motion and translation as a combined finger pinch and translation gesture only if additionally the translation distance is greater than a second predetermined threshold.
- 41.** The computer-readable medium of claim **38**, the program code further for causing performance of a method comprising performing a drag and drop operation if the finger pinch occurs over a graphical user interface (GUI) object and the translation moves the actual images of touch to a drop location.
- 42.** The computer-readable medium of claim **38**, the program code further for causing performance of a method comprising:
determining whether a liftoff of the one or more fingers occurred;
computing a liftoff velocity of the one or more fingers at a time of the liftoff; and
recognizing the finger motion, translation and liftoff as a combined finger pinch, translation and liftoff gesture if the liftoff velocity is greater than a second predetermined threshold.
- 43.** The computer-readable medium of claim **42**, the program code further for causing performance of a method comprising performing a trash operation if the finger pinch occurs over a graphical user interface (GUI) object and the translation moves the actual images of touch towards a trash icon.
- 44.** The computer-readable medium of claim **42**, the program code further for causing performance of a method com-

prising performing a move to another desktop operation if the finger pinch occurs over a graphical user interface (GUI) object and the translation moves the actual images of touch towards another desktop.

45. The computer-readable medium of claim **42**, the program code further for causing performance of a method comprising performing a window minimize operation if the finger pinch occurs over a window and the translation moves the actual images of touch towards a corner of a desktop.

46. The computer-readable medium of claim **42**, the program code further for causing performance of a method comprising performing a print operation if the finger pinch occurs over a graphical user interface (GUI) object and the translation moves the actual images of touch towards a printer icon.

47. The computer-readable medium of claim **42**, the program code further for causing performance of a method comprising performing a scrolling operation if the finger pinch occurs over a graphical user interface (GUI) object and the translation moves the actual images of touch in a particular direction to be scrolled.

48. The computer-readable medium of claim **47**, the program code further for causing performance of a method comprising performing a scrolling operation with a gradual decay in a scrolling speed if the finger pinch occurs over a graphical user interface (GUI) object and the liftoff velocity is greater than a third predetermined threshold.

49. The computer-readable medium of claim **36**, the program code further for causing performance of a method comprising:

- determining whether the actual images of touch captured over time are substantially similar to two or more finger patches rotating around an imaginary axis; and
- recognizing the finger motion and translation as a combined finger rotation and translation gesture if the actual images of touch captured over time are substantially similar to the two or more finger patches rotating around the imaginary axis.

50. The computer-readable medium of claim **49**, the program code further for causing performance of a method comprising:

- detecting a rotation speed by computing a cross product of velocity vectors for the two or more finger patches;
- computing a ratio of the rotation speed to the translation speed; and
- recognizing the finger motion and translation as a combined finger rotation and translation gesture only if additionally the ratio is greater than a second predetermined threshold.

51. The computer-readable medium of claim **49**, the program code further for causing performance of a method comprising:—

- determining a translation distance of the one or more fingers involved in the finger rotation; and
- recognizing the finger motion and translation as a combined finger rotation and translation gesture only if additionally the translation distance is greater than a second predetermined threshold.

52. The computer-readable medium of claim **49**, the program code further for causing performance of a method comprising:

- determining whether a liftoff of the one or more fingers occurred;
- computing a liftoff velocity of the one or more fingers at a time of the liftoff; and

recognizing the finger motion, translation and liftoff as a combined finger rotation, translation and liftoff gesture if the liftoff velocity is greater than a second predetermined threshold.

53. The computer-readable medium of claim **52**, the program code further for causing performance of a method comprising performing a move to another desktop operation if the finger rotation occurs over a graphical user interface (GUI) object and the translation moves the actual images of touch towards another desktop.

54. The computer-readable medium of claim **36**, the program code further for causing performance of a method comprising:

- determining whether the actual images of touch captured over time are substantially similar to one or more finger taps in substantially the same location within a predetermined period of time; and

- recognizing the finger motion and translation as a combined finger tap and translation gesture if the actual images of touch captured over time are substantially similar to the one or more finger taps in substantially the same location within the predetermined period of time.

55. The computer-readable medium of claim **54**, the program code further for causing performance of a method comprising:

- determining a translation distance of the one or more fingers involved in the finger tap; and
- recognizing the finger motion and translation as a combined finger tap and translation gesture only if additionally the translation distance is greater than a second predetermined threshold.

56. The computer-readable medium of claim **54**, the program code further for causing performance of a method comprising performing a drag and drop operation if the finger tap occurs over a graphical user interface (GUI) object and the translation moves the actual images of touch to a drop location.

57. The computer-readable medium of claim **54**, the program code further for causing performance of a method comprising:

- determining whether a liftoff of the one or more fingers occurred;
- computing a liftoff velocity of the one or more fingers at a time of the liftoff; and
- recognizing the finger motion, translation and liftoff as a combined finger tap, translation and liftoff gesture if the liftoff velocity is greater than a second predetermined threshold.

58. The computer-readable medium of claim **54**, the program code further for causing performance of a method comprising performing a trash operation if the finger tap occurs over a graphical user interface (GUI) object and the translation moves the actual images of touch towards a trash icon.

59. The computer-readable medium of claim **54**, the program code further for causing performance of a method comprising performing a move to another desktop operation if the finger tap occurs over a graphical user interface (GUI) object and the translation moves the actual images of touch towards another desktop.

60. The computer-readable medium of claim **54**, the program code further for causing performance of a method comprising performing a window minimize operation if the finger tap occurs over a window and the translation moves the actual images of touch towards a corner of a desktop.

61. The computer-readable medium of claim 54, the program code further for causing performance of a method comprising performing a print operation if the finger tap occurs over a graphical user interface (GUI) object and the translation moves the actual images of touch towards a printer icon.

62. The computer-readable medium of claim 54, the program code further for causing performance of a method comprising performing a scrolling operation if the finger tap occurs over a graphical user interface (GUI) object and the translation moves the actual images of touch in a particular direction to be scrolled.

63. The computer-readable medium of claim 63, the program code further for causing performance of a method comprising performing a scrolling operation with a gradual decay in a scrolling speed if the finger tap occurs over a graphical user interface (GUI) object and the liftoff velocity is greater than a third predetermined threshold.

64. A computer-readable medium comprising program code for recognizing a combined finger motion and translation gesture on a sensor panel, the program code for causing performance of a method comprising:

identifying a type of the finger motion by comparing actual images of touch captured over time to expected images of touch for various finger motion types;

computing a translation distance of one or more fingers involved in the finger motion; and

recognizing the finger motion and translation as a combined finger motion and translation gesture if the translation distance is greater than a first predetermined threshold.

65. The computer-readable medium of claim 64, the program code further for causing performance of a method comprising:

determining a translation speed of the one or more fingers involved in the finger pinch; and

recognizing the finger motion and translation as a combined finger pinch and translation gesture only if additionally the translation speed is greater than a second predetermined threshold.

66. The computer-readable medium of claim 64, the program code further for causing performance of a method comprising:

determining whether the actual images of touch captured over time are substantially similar to two or more finger patches moving towards each other; and

recognizing the finger motion and translation as a combined finger pinch and translation gesture if the actual images of touch captured over time are substantially similar to the two or more finger patches moving towards each other.

67. The computer-readable medium of claim 66, the program code further for causing performance of a method comprising:

detecting a scaling speed by computing a dot product of velocity vectors for the two or more finger patches;

computing a ratio of the scaling speed to the translation speed; and

recognizing the finger motion and translation as a combined finger pinch and translation gesture only if additionally the ratio is greater than a second predetermined threshold.

68. The computer-readable medium of claim 64, the program code further for causing performance of a method comprising:

determining whether a liftoff of the one or more fingers occurred;

computing a liftoff velocity of the one or more fingers at a time of the liftoff; and

recognizing the finger motion, translation and liftoff as a combined finger pinch, translation and liftoff gesture if the liftoff velocity is greater than a second predetermined threshold.

69. The computer-readable medium of claim 64, the program code further for causing performance of a method comprising:

determining whether the actual images of touch captured over time are substantially similar to two or more finger patches rotating around an imaginary axis; and

recognizing the finger motion and translation as a combined finger rotation and translation gesture if the actual images of touch captured over time are substantially similar to the two or more finger patches rotating around the imaginary axis.

70. The computer-readable medium of claim 69, the program code further for causing performance of a method comprising:

detecting a rotation speed by computing a cross product of velocity vectors for the two or more finger patches;

computing a ratio of the rotation speed to the translation speed; and

recognizing the finger motion and translation as a combined finger rotation and translation gesture only if additionally the ratio is greater than a second predetermined threshold.

71. A computing system comprising the computer-readable medium of claim 64.

72. A mobile telephone comprising the computing system of claim 71.

73. A digital audio player comprising the computing system of claim 71.

74. A mobile telephone including a computer-readable medium comprising program code for recognizing a combined finger motion and translation gesture on a sensor panel, the program code for causing performance of a method comprising:

identifying a type of the finger motion by comparing actual images of touch captured over time to expected images of touch for various finger motion types;

computing a translation distance of one or more fingers involved in the finger motion; and

recognizing the finger motion and translation as a combined finger motion and translation gesture if the translation distance is greater than a first predetermined threshold.

75. A digital audio player including a computer-readable medium comprising program code for recognizing a combined finger motion and translation gesture on a sensor panel, the program code for causing performance of a method comprising:

identifying a type of the finger motion by comparing actual images of touch captured over time to expected images of touch for various finger motion types;

computing a translation distance of one or more fingers involved in the finger motion; and

recognizing the finger motion and translation as a combined finger motion and translation gesture if the translation distance is greater than a first predetermined threshold.

76. An apparatus for recognizing a combined finger motion and translation gesture on a sensor panel, the apparatus comprising:

means for identifying a type of the finger motion by comparing actual images of touch captured over time to expected images of touch for various finger motion types;

means for computing a translation distance of one or more fingers involved in the finger motion; and
means for recognizing the finger motion and translation as a combined finger motion and translation gesture if the translation distance is greater than a first predetermined threshold.

* * * * *