EXHIBIT 2.08

predetermined area may, for example, correspond to the area where the page number is displayed. Following block 604, the GUI operational method 600 proceeds to block 606 where a page turn signal is generated when the object (or objects) is translated in the predetermined region. The translation is configured to simulate a finger turning the page in an actual paper bound book. The direction of the translation indicates whether to go to the next page or previous page in the list of pages. For example, if the finger is swiped right to left, then a page back signal is generated, and if the finger is swiped left to right, then a page up signal is generated. This GUI operational method 600 may be enhanced several ways. For instance, if multiple fingers are swiped, then this may create a paging signal greater than one page. For example, a two finger swipe equals two page turns, three finger swipe equals three page turns, etc. Or a two finger swipe equals ten page turns, three finger swipe equals 50 page turns, etc.

Figs. 21A-21E illustrate a page turning sequence using the GUI operational method 600 described above. As shown in Fig. 21A, which is a close up of a user 510 holding the tablet PC 512, the user swipes their finger over the page number in a direction to the left of the page 630. As shown in Fig. 21B, the tablet PC 512 recognizes the swipe and direction of the swipe in the area of the page number and therefore the tablet PC 512 displays the next page in a group of pages. This can be performed repeatedly to whisk through the group of pages. As shown in Fig. 21C, the user swipes their finger 576 over the page number in a direction to the right of the page 630. As shown in Fig. 21D, the tablet PC 512 recognizes the swipe and direction of the swipe in the area of the page number and therefore the tablet PC 512 displays the previous page in a group of pages. This can be performed repeatedly to whisk through the group of pages.

Fig. 22 is a diagram of a GUI operational method 650, in accordance with one embodiment of the present invention. The GUI operational method 650 is configured for initiating inertia typically during a scrolling or panning operation. Inertia is generally defined as the tendency of a body at rest to remain at rest or of a body in motion to stay in motion in a straight line unless disturbed by an external force. In this particular embodiment, the GUI or some portion thereof is associated with inertial properties, which is its resistance to rate of change in motion. For a GUI with high inertia characteristics, the acceleration of the GUI will be small for a given input. On the other hand, if the GUI has low inertia characteristics, the acceleration will be large for a given input.

The GUI operational method 650 generally begins at block 652 where a graphical image is displayed on a GUI. Following block 652, the GUI operational method 650 proceeds to block 654 where a scrolling or panning stroke on a touch sensitive surface is detected. By way of example, the stroke may be a linear or rotational stroke. During a linear stroke, the direction of scrolling or panning typically follows the direction of the stroke. During a rotational stroke (see Fig. 6), the rotational stroke is typically converted to a linear input where clockwise motion may correspond to vertical up and counterclockwise motion may correspond to vertical down. Following block 654 the process flow proceeds to block 656 where the speed and direction of the scrolling or panning stroke is determined. Following block 656, the GUI operational method 650 proceeds to block 658 where the image is moved in accordance with the speed and direction of the scrolling or panning stroke as well as the associated inertial characteristics. Following block 658, the GUI operational method 650 proceeds to block 660 where the motion of the image continues even when the panning or scrolling stroke is no longer detected. For example, when the user picks up their finger from the touch sensitive surface, the scrolling or panning function continues as if the scrolling or panning stroke was still being made. In some cases, the motion of the image continues infinitely until some braking (stopping or slowing) control is performed. This particular methodology simulates zero gravity. In other cases, the motion of the image is slowed in accordance with the associated inertia GUI operational method 650. Metaphorically speaking, the image may correspond to a piece of paper moving over a desktop. In order to move the piece of paper, the user exerts a force on the paper in the desired direction. When the user lifts their finger off the paper, the paper will continue to slid along the desktop in the desired direction for some period of time. The amount it slides after lifting the finger generally depends on, among other things, its mass, the force applied by the finger, the friction force found between the paper and the desktop, etc. As should be appreciated, traditionally when scrolling and panning are implemented, the scrolling or panning stops when the fingers are picked up. In contrast, using the above mentioned methodology, the scrolling or panning continues to move when the fingers are picked up.

The GUI operational method 650 may additionally include blocks A and B. In block A, an object such as a finger is detected on the touch sensitive surface when the image is moving without the assistance of the object (block 660). In block B, the motion of the image is stopped when the object is detected, i.e., the new touch serves as a braking means. Using the metaphor above, while the piece of paper is sliding across the desktop, the user presses their finger on the paper thereby stopping its motion.

Figs. 23A-23D illustrate an inertia sequence using the method described above. Fig. 23A illustrates a display presenting a GUI 678 including a window 679 having a list 680 of media items 681. The window 679 and list 680 may for example correspond to a control window and music list found in iTunes™ manufactured by Apple Computer, Inc of Cupertino, CA. As shown in Fig. 23B, when the user slides their finger or fingers 576 over the touch screen 520, vertical scrolling, which moves media items up or down through the window, is implemented. The direction of scrolling may follow the same direction as finger movement (as shown), or it may go in the reverse direction. In one particular embodiment, a single finger is used for selecting the media items from the list, and two fingers are used to scroll through the list.

Scrolling generally pertains to moving displayed data or images (e.g., media items 681) across a viewing area on a display screen so that a new set of data (e.g., media items 681) is brought into view in the viewing area. In most cases, once the viewing area is full, each new set of data appears at the edge of the viewing area and all other sets of data move over one position. That is, the new set of data appears for each set of data that moves out of the viewing area. In essence, these functions allow a user to view consecutive sets of data currently outside of the viewing area. In most cases, the user is able to accelerate their traversal through the data sets by moving his or her finger at greater speeds. Examples of scrolling through lists can be found in

U.S. Patent Publication Nos.: 2003/0076303A1, 2003/0076301A1, 2003/0095096A1, which are herein incorporated by reference.

As shown in Fig. 23C, the displayed data continues to move even when the finger is removed from the touch screen. The continuous motion is based at least in part on the previous motion. For example the scrolling may be continued in the same direction and speed. In some cases, the scrolling slow down over time, i.e., the speed of the traversal through the media items gets slower and slower until the scrolling eventually stops thereby leaving a static list. By way of example, each new media item brought into the viewing area may incrementally decrease the speed. Alternatively or additionally, as shown in Fig. 23D, the displayed data stops moving when the finger 576 is placed back on the touch screen 520. That is, the placement of the finger back on the touch screen can implement braking, which stops or slows down the continuous acting motion. Although this sequence is directed at vertical scrolling it should be noted that this is not a limitation and that horizontal scrolling as well as panning may be performed using the methods described above.

Fig. 24 is a diagram of a GUI operational method 700, in accordance with one embodiment of the present invention. The method 700 is configured for simulating a keyboard. The method generally begins at block 702 where a keyboard is presented on the display. Following block 702, the process flow proceeds to block 704 where the presence of a first object over a first key and a second object over a second key at the same time is detected on a touch screen. The touch screen is positioned over or in front of the display. By way of example, the display may be an LCD and the touch screen may be a multipoint touch screen. Following block 704, the process flow proceeds to block 706 where one or more simultaneous control signals are generated when the first object is detected over the first key and when the second object is detected over the second key at the same time.

In one embodiment, only a single control signal is generated when the first object is detected over the first key and when the second object is detected over the second key at the same time. By way of example, the first key may be a shift key and the second key may be a symbol key (e.g., letters, numbers). In this manner, the keyboard acts like a traditional keyboard, i.e., the user is allowed to select multiple keys at the same time in order to change the symbol, i.e., lower/upper case. The keys may also correspond to the control key, alt key, escape key, function key, and the like.

In another embodiment, a control signal is generated for each actuated key (key touch) that occurs at the same time. For example, groups of characters can be typed at the same time. In some cases, the application running behind the keyboard may be configured to determine the order of the characters based on some predetermined criteria. For example, although the characters may be jumbled, the application can determine that the correct order of characters based on spelling, usage, context, and the like.

Although only two keys are described, it should be noted that two keys is not a limitation and that more than two keys may be actuated simultaneously to produce one or more control signals. For example, control-alt-delete functionality may be implemented or larger groups of characters can be typed at the same time.

Figs. 25A-25D illustrates a keyboard sequence using the method described above. Fig. 25A illustrates a display presenting a GUI object 730 in the form of a keyboard. As shown in Fig. 25B, a user positions their fingers 576 over the multipoint touch screen 520 over the keyboard 730 to enter data into a word processing program. By way of example, the user may place one of their fingers 576A on the Q key in order to produce a lower case "q" in the word processing program. As shown in Fig. 25C, when the user decides that a letter should be in upper case, the user places one finger 576B on the shift key and another finger 576A on the desired letter (as indicated by the arrows). As shown in Fig. 25D, in order to continue typing in lower case, the user simply removes their finger 576B from the shift key and places their finger 576A over a desired letter (as indicated by the arrow).

Fig. 26 is a diagram of a GUI operational method 750, in accordance with one embodiment of the present invention. The method 750 is configured for simulating a scroll wheel such as those described in U.S. Patent Publication Nos: 2003/0076303A1, 2003/0076301A1, 2003/0095096A1, all of which are herein incorporated by reference. The method generally begins at block 752 where a virtual scroll wheel is presented on the display. In some cases, the virtual scroll wheel may include a virtual button at its center. The virtual scroll wheel is configured to implement scrolling as for example through a list and the button is configured to implement selections as for example items stored in the list. Following block 752, the method proceeds to block 754 where the presence of at least a first finger and more particularly, first and second fingers (to distinguish between tracking and gesturing) over the virtual scroll wheel is detected on a touch screen. The touch screen is positioned over or in front of the display. By way of example, the display may be an LCD and the touch screen may be a multipoint touch screen. Following block 754, the method proceeds to block 756 where the initial position of the fingers on the virtual scroll wheel is set. By way of example, the angle of the fingers relative to a reference point may be determined (e.g., 12 o clock, 6 o clock, etc.). Following block 756, the method 750 proceeds to block 758 where a rotate signal is generated when the angle of the fingers change relative to the reference point. In most cases, the set down of the fingers associate, link or lock the fingers (or finger) to the virtual scroll wheel when the fingers are positioned over the virtual scroll wheel. As a result, when the fingers are rotated, the rotate signal can be used to rotate the virtual scroll wheel in the direction of finger rotation (e.g., clockwise, counterclockwise). In most cases, the amount of wheel rotation varies according to the amount of finger rotation, i.e., if the fingers move 5 degrees then so will the wheel. Furthermore, the rotation typically occurs substantially simultaneously with the motion of the fingers. For instance, as the fingers rotate, the scroll wheel rotates with the fingers at the same time.

In some cases, the principals of inertia as described above can be applied to the virtual scroll wheel. In cases such as these, the virtual scroll wheel continues to rotate when the fingers (or one of the fingers) are lifted off of the virtual scroll wheel and slowly comes to a stop via virtual friction. Alternatively or additionally, the continuous rotation can be stopped by placing the fingers (or the removed finger) back on the scroll wheel thereby braking the rotation of the virtual scroll wheel.

Figs. 27A-27D illustrates a scroll wheel sequence using the method described above. Fig. 27A illustrates a display presenting a scroll wheel. The scroll wheel may be displayed automatically as part of a program or it may be displayed when a particular gesture is performed. By way of example, during the operation of a music program (such as iTunes manufactured by Apple Computer Inc., of Cupertino, CA), the virtual scroll wheel may appear on the GUI of the music program when two fingers are placed on the touch screen rather than one finger which is typically used for tracking in the music program. In some cases, the virtual scroll wheel only appears when two fingers are placed on a predetermined area of the GUI. As shown in Fig. 27B, a user positions their fingers over the multipoint touch screen 520 over the scroll wheel. At some point, the fingers are locked to the scroll wheel. This can occur at set down for example. As shown in Fig. 27C, when the fingers are rotated in a clockwise direction, the scroll wheel is rotated in the clockwise direction in accordance with the rotating fingers. As shown in Fig. 27D, when the fingers are rotated in a counterclockwise direction, the virtual scroll wheel is rotated in the counter clockwise direction in accordance with the rotating fingers. Alternatively, rotation of the virtual scroll wheel may also be rotated with linear motion of the fingers in a tangential manner.

It should be noted that although a surface scroll wheel is shown, the principals thereof can be applied to more conventional scroll wheels which are virtually based. For example, scroll wheels, whose axis is parallel to the display screen and which appear to protrude through the display screen as shown in Fig. 28. In this particular implementation, however, linear motion of the fingers are used to rotate the virtual scroll wheel.

The various aspects, embodiments, implementations or features of the invention can be used separately or in any combination.

The invention is preferably implemented by hardware, software or a combination of hardware and software. The software can also be embodied as computer readable code on a computer readable medium. The computer readable medium is any data storage device that can store data which can thereafter be read by a computer system. Examples of the computer readable medium include read-only memory, random-access memory, CD-ROMs, DVDs, magnetic tape, optical data storage devices, and carrier waves. The computer readable medium can also be distributed over network-coupled computer systems so that the computer readable code is stored and executed in a distributed fashion.

While this invention has been described in terms of several preferred embodiments, there are alterations, permutations, and equivalents, which fall within the scope of this invention. For example, although the invention has been primarily directed at touchscreens, it should be noted that in some cases touch pads may also be used in place of touchscreens. Other types of touch sensing devices may also be utilized. It should also be noted that there are many alternative ways of implementing the methods and apparatuses of the present invention. It is therefore intended that the following appended claims be interpreted as including all such alterations, permutations, and equivalents as fall within the true spirit and scope of the present invention.

What is claimed is:

1. A computer implemented method for processing touch inputs, said method comprising:

reading data from a touch sensitive device, the data pertaining to touch input with respect to the touch sensitive device, and the touch sensitive device having a multipoint capability; and

identifying at least one multipoint gesture based on the data from the touch sensitive device.

2. A computer implemented method as recited in claim 1 wherein said method further comprises:

performing a computing operation associated with the at least one gesture.

3. A computer implemented method as recited in claim 1 wherein said method further comprises:

associating a user interface element to the at least one gesture.

4. A computer implemented method as recited in claim 3 wherein said method further comprises:

displaying the user interface element on a display.

5. A computer implemented method as recited in claim 1 wherein said method further comprises:

displaying the user interface element on a display prior to said reading; and

subsequently altering the user interface element being displayed based on the at least one gesture.

6. A computer implemented method as recited in claim 1-5 wherein said identifying identifies at least two multipoint gestures based on the data from the touch sensitive device.

7. A computer implemented method as recited in claim 6 wherein the at least two gestures being identified are concurrently performed on the touch sensitive device.

8. A computer implemented method as recited in claim 6 wherein the at least two gestures being identified are contiguously performed on the touch sensitive device.

9. A gestural method, comprising:

detecting multiple touches at different points on a touch sensitive surface at the same time; and segregating the multiple touches into at least two separate gestural inputs occurring simultaneously, each gestural input having a different function.

10. The gestural method as recited in claim 9 wherein touch sensitive surface is associated with a touch screen that overlays a display, and wherein the gestural inputs are selected from zooming, panning, and rotating.

11. A gestural method, comprising:

concurrently detecting a plurality of gestures that are concurrently performed with reference to a touch sensing device;

producing different commands for each of the gestures that have been detected.

12. The method as recited in claim 11 wherein the plurality of gestures are implemented with the same group of touches.

13. The method as recited in claim 11 wherein the plurality of gestures are implemented with different groups of touches.

14. The method as recited in claim 11-13 wherein the gestures include rotate gestures, panning gestures, or zooming gestures.

15 . A gestural method, comprising:

displaying a graphical image on a display screen;

detecting a plurality of touches at the same time on a touch sensitive device; and

linking the detected multiple touches to the graphical image presented on the display screen.

16. The method as recited in claim 15 wherein the graphical image is a user interface element.

17. The method as recited in claim 16 wherein the user interface element is displayed when a particular multitouch configuration is detected.

18. The method as recited in claim 17 wherein the touch sensitive device is a touch screen disposed over the display screen, and wherein the user interface element is displayed in proximity to the multiple touches.

19. The method as recited in claim 18 wherein the user interface element is displayed underneath the multiple touches.

20. The method as recited in claim 16-19 wherein the multiple touches are detected over the displayed user interface element.

21. The method as recited in claim 16-19 wherein the multiple touches are associated with a single user interface element.

22. The method as recited in claim 16-19 wherein a first portion of the multiple touches are associated with a first user interface element and a second portion of the multiple touches are associated with a second user interface element that is different than the first user interface element.

23. The method as recited in claim 15-22 wherein after linking the detected multiple touches to the graphical image, the graphical image changes based on the motion of the linked multiple touches.

24. The method as recited in claim 23 wherein the graphical image is changed in unison with the motion of the linked multiple touches such that the changes in the graphical image continuously follows the motion of the linked multiple touches.

25. The method as recited in claim 23 wherein the size of the graphical image changes when at least one of the touches in the linked multiple touches is translated over the touch sensitive surface.

26. The method as recited in claim 23 wherein the size of features embedded in the graphical image changes when at least one of the touches in the linked multiple touches is translated over the touch sensitive surface.

27. The method as recited in claim 23 wherein the orientation of the graphical image changes when at least one of the touches in the linked multiple touches is rotated over the touch sensitive surface.

28. The method as recited in claim 23 wherein the position of the graphical image changes when the linked multiple touches are translated together over the touch sensitive surface.

29. A method of invoking a user interface element on a display via a multipoint touch sensitive device of a computing system, said method comprising:

detecting and analyzing the simultaneous presence of two or more objects on contact with said touch sensitive device;

based at least in part on said analyzing, selecting a user interface tool, from a plurality of available tools, to display on a display for interaction by a user of said computing system; and

controlling the interface tool based at least in part on the further movement of said objects in relation to said touch sensitive device.

30. The method as recited in claim 29 wherein said selecting is also based in part on the current mode of the computing device at the time of the simultaneous presence of the two or more objects.

31. The method as recited in claim 29 wherein said touch sensitive device is a touch screen disposed over the display and wherein said tool is displayed at a location on the display proximate to the presence of said two or more objects on said touchscreen.

32. A touch-based method, comprising

detecting a user input that occurs over a multipoint sensing device, the user input including one or more inputs, each input having a unique identifier;

during the user input, classifying the user input as a tracking or selecting input when the user input includes one unique identifier or a gesture input when the user input includes at least two unique identifiers;

performing tracking or selecting during the user input when the user input is classified as a tracking or selecting input;

performing one or more control actions during the user input when the user input is classified as a gesturing input, the control actions being based at least in part on changes that occur between the at least two unique identifiers.

33. The method as recited in claim 32 wherein scrolling or panning is performed when the at least two unique identifiers move together in substantially the same direction.

34. The method as recited in claim 32 wherein zooming is performed when the at least two unique identifiers linearly move away or towards one another.

35. The method as recited in claim 32 wherein rotation is performed when the at least two unique identifiers rotate relative to each other or relative to a known point.

36. The method as recited in claim 32 wherein multiple control actions are performed simultaneously during the same user input.

37. The method as recited in claim 32 wherein the user input is a continuous stroke, the stroke maintaining continuous contact on the multipoint sensing device.

38. A touch-based method, comprising:

outputting a GUI on a display;

detecting a user input on a touch sensitive device;

analyzing the user input for characteristics indicative of tracking, selecting or a gesturing;

categorizing the user input as a tracking, selecting or gesturing input;

performing tracking or selecting in the GUI when the user input is categorized as a tracking or selecting input;

performing control actions in the GUI when the user input is categorized as a gesturing input, the actions being based on the particular gesturing input .

39. A touch-based method, comprising:

capturing an initial touch image;

determining the touch mode based on the touch image;

capturing the next touch image;

determining if the touch mode changed between the initial and next touch images;

if the touch mode changed, setting the next touch image as the initial touch image and determining the touch mode based on the new initial touch image; and

if the touch mode stayed the same, comparing the touch images and performing a control function based on the comparison.

40. The method as recited in claim 39 wherein the control function is selected from tracking, panning, zooming, rotating, scrolling, or enlarging.

41. A computer implemented method for processing touch inputs, said method comprising:

reading data from a touch sensitive device, the data pertaining to touch input with respect to the touch sensitive device, and the touch sensitive device having a multipoint capability;

converting the data to a collection of features;

classifying the features;

grouping the features into one or more feature groups;

calculating key parameters of the feature groups; and

associating the feature groups to user interface elements on a display.

42. The computer implemented method as recited in 41 wherein said method further comprises:

recognizing when at least one of the feature groups indicates performance of a gesture relative to its associated user interface element.

43. The computer implemented method as recited in 41 wherein said method further comprises:

providing user feedback when at least one of the feature groups indicates performance of a gesture relative to its associated user interface element.

44. The computer implemented method as recited in 41 wherein said method further comprises:

implementing an action when at least one of the feature groups indicates performance of a gesture relative to its associated user interface element.

45. The computer implemented method as recited in 44, wherein said method further comprises:

providing user feedback in conjunction with the action.

46. The method as recited in claim 41-45 wherein the step of associating comprises:

receiving group of features;

determining if there is a change in number of features;

if there is a change, calculating initial parameter values;

if there is no change, calculating current parameter values; and

reporting both the initial and current parameter values.

47. A computer implemented method, comprising:

outputting a graphical image;

receiving a multitouch gesture input over the graphical image; and

changing the graphical image based on and in unison with multitouch gesture input.

48. The method as recited in claim 47 further comprising:

receiving a second multitouch gesture input over the graphical image; and

changing the graphical image based on and in unison with the second multitouch gesture.

49. The method as recited in claim 47 further comprising:

outputting a second graphical image:

receiving a second multitouch gesture input over the second graphical image; and

changing the second graphical image based on and in unison with the second multitouch gesture.

50. A touch based method, comprising:

receiving a gestural input over a first region;

generating a first command when the gestural input is received over the first region;

receiving the same gestural input over a second region; and generating a second command when the same gestural input is received over the second region, the second command being different than the first command.

51. A method for recognizing multiple gesture inputs, the method comprising:

receiving a multitouch gestural stroke on a touch sensitive surface, the multitouch gestural stroke maintaining continuous contact on the touch sensitive surface;

recognizing a first gesture input during the multitouch gestural stroke; and

recognizing a second gesture input during the multitouch gestural stroke.

52. The method as recited in claim51 further including recognizing a third gesture input during the multitouch gestural stroke

53. A computer implemented method, comprising:

detecting a plurality of touches on a touch sensing device;

forming one or more touch groups with the plurality of touches;

monitoring the movement of and within each of the touch groups; and

generating control signals when the touches within the touch groups are moved or when the touch groups are moved in their entirety.

54. A method for recognizing a zoom gesture made on a multipoint touch sensitive device computing system, comprising:

detecting the relative locations of a first object and a second object at the same time;

detecting a change in the relative locations of said first and second object;

generating a zoom signal in response to said detected change.

55. The method as recited in claim 54 wherein said zoom signal is a zoom out signal.

56. The method as recited in claim 54 wherein said zoom signal is a zoom in signal.

57. The method as recited in claim 54 further comprising:

associating the first and second objects to an image on a GUI interface;

58. The method as recited in claim 57 further comprising zooming out the image when a zoom out signal is generated.

59. The method as recited in claim 57 further comprising zooming in the image when a zoom in signal is generated.

60. The method as recited in claim 54-59 further comprising zooming the image when a zoom signal is generated, and wherein the zooming occurs substantially simultaneously with the change in relative locations of the objects.

61. The method as recited in claim 60 wherein the amount of zooming varies according to the distance between the two objects.

62. A method for recognizing a pan gesture made on a multipoint touch sensitive device, comprising:

detecting the presence of at least a first object and a second object at the same time;

monitoring the position of the said at least first and second objects when the objects are moved together across the touch sensitive device; and

generating a pan signal when the position of the said at least first and second objects changes relative to an initial position.

63. The method as recited in claim 62 further comprising:

associating said at least first and second objects to an image on a GUI interface;

panning the image when a pan signal is generated; and

wherein the panning occurs substantially simultaneously with the motion of the objects.

64. A method for recognizing a rotate gesture made on a multipoint touch sensitive device, comprising:

detecting the presence of at least a first object and a second object at the same time;

detecting a rotation of said at least first and second objects; and

generating a rotate signal in response to said detected rotation of said at least first and second objects.

65. The method as recited in claim 64 further comprising:

associating said at least first and second objects to an image on a GUI interface;

rotating the image when a rotate signal is generated; and

wherein the rotating occurs substantially simultaneously with the motion of the objects.

66. A computer implemented method for initiating zooming targets via a touch screen, the method comprising:

displaying an image on a GUI;

enlarging the image for a period of time when the presence of an object is detected over the image.

67. The method as recited in claim 66 wherein the image is a user interface element.

68. The method as recited in claim 67 wherein the user interface element is a control box containing one or more control buttons, and wherein said enlarging includes enlarging the control box and each of the control buttons.

69. The method as recited in claim 68 wherein the size of the control buttons correspond to the size of the object or larger so that the control buttons can be easily selected by the object.

70. The method as recited in claim 69 further comprising:

in the enlarged state, generating a control signal associated with the control button when the presence of the object is detected over one of the control buttons.

71. The method as recited in claim 66 wherein the object is a finger.

72. The method as recited in claim 66-71 further comprising:

implementing a gesture input when the presence of one or more objects are detected over portions of the GUI other than the image.

73. A computer implemented method of initiating a page turn via a touch sensitive device, the method comprising:

displaying a page from a multitude of pages in a GUI presented on a display;

detecting the presence of an object on the touch sensitive device; and

generating a page turn signal when the object is translated horizontally on the touch sensitive device.

74. The computer implemented method as recited in claim 73 wherein the translation simulates a finger turning the page in a paper bound book.

75. The computer implemented method as recited in claim 73 wherein the translation pertains to a page turn gesture by a user.

76. The computer implemented method as recited in claim 73-75 wherein said method further comprises:

displaying a next page in response to the page turn signal.

77. A computer implemented method of initiating inertia, the method comprising:

displaying an image on a GUI;

detecting a stroke on a touch sensitive surface;

noting the speed and direction of the stroke;

moving the image or features embedded in the image in accordance with the speed and direction of the stroke; and

slowing the motion of the image or features embedded in the image in accordance with inertia principals when the stroke is no longer detected.

78. The method as recited in claim 77 further comprising:

detecting an object on the touch sensitive surface when image or features embedded in the image is slowing down because of inertia;

stopping the motion of the image or features embedded in the image when another object is detected, the another object serving as a braking means to the moving image or features embedded in the image

79. A method of simulating a keyboard, comprising:

providing a display and a touch screen positioned over the display;

displaying a keyboard on the display, the keyboard including at least a first and a second key;

detecting the presence of a first object over the first key and a second object over the second key at the same time; and

generating a single control function in response to the detection of said first object over the first key and said second object over said second key.

80. The method as recited in claim 79 wherein the first and second objects are portions of one or more hands.

81. The method as recited in claim 79 wherein the first key is a modifying key that modifies the second key.

82. The method as recited in claim 81 wherein the first key is a shift key, control key or alt key.

82. The method as recited in claims 9, wherein the touch sensitive surface is part of a touch sensitive device, and wherein the gestural inputs are selected from zooming, panning and rotating.

83. The method as recited in claims 1, 11, 15, 29, 32, 38, 41, 51, 53, 54, 62, 64, 73, 77, wherein the touch sensitive device is a touch screen.

84. The method as recited in claims 1, 11, 15, 29, 32, 38, 41, 51, 53, 54, 62, 64, 73, 77, wherein the touch sensitive device is a touch pad.

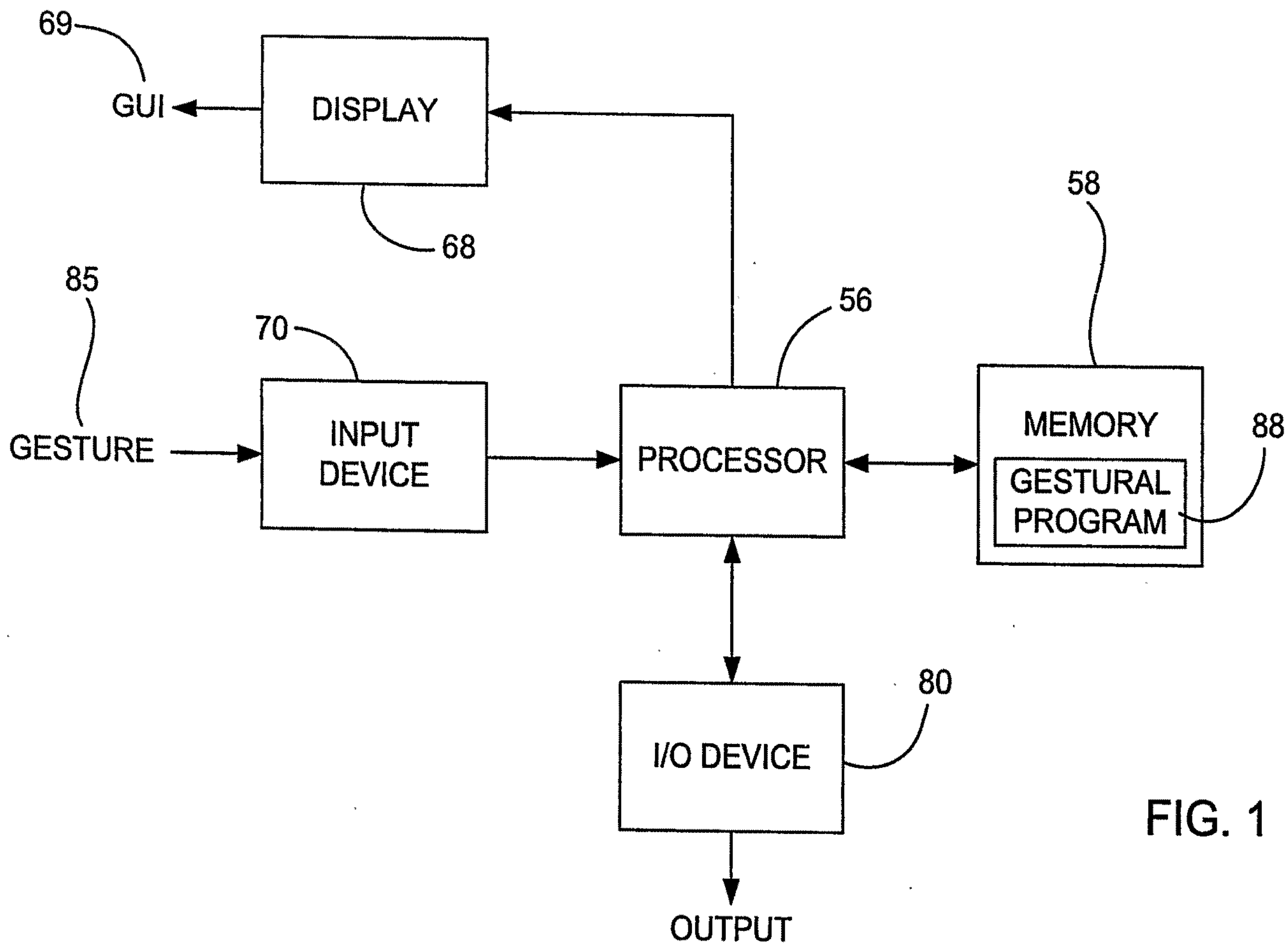

69
GUI
DISPLAY
68
85
70
GESTURE
INPUT DEVICE
PROCESSOR
56
58
MEMORY
GESTURAL PROGRAM
88
I/O DEVICE
80
OUTPUT

FIG. 1

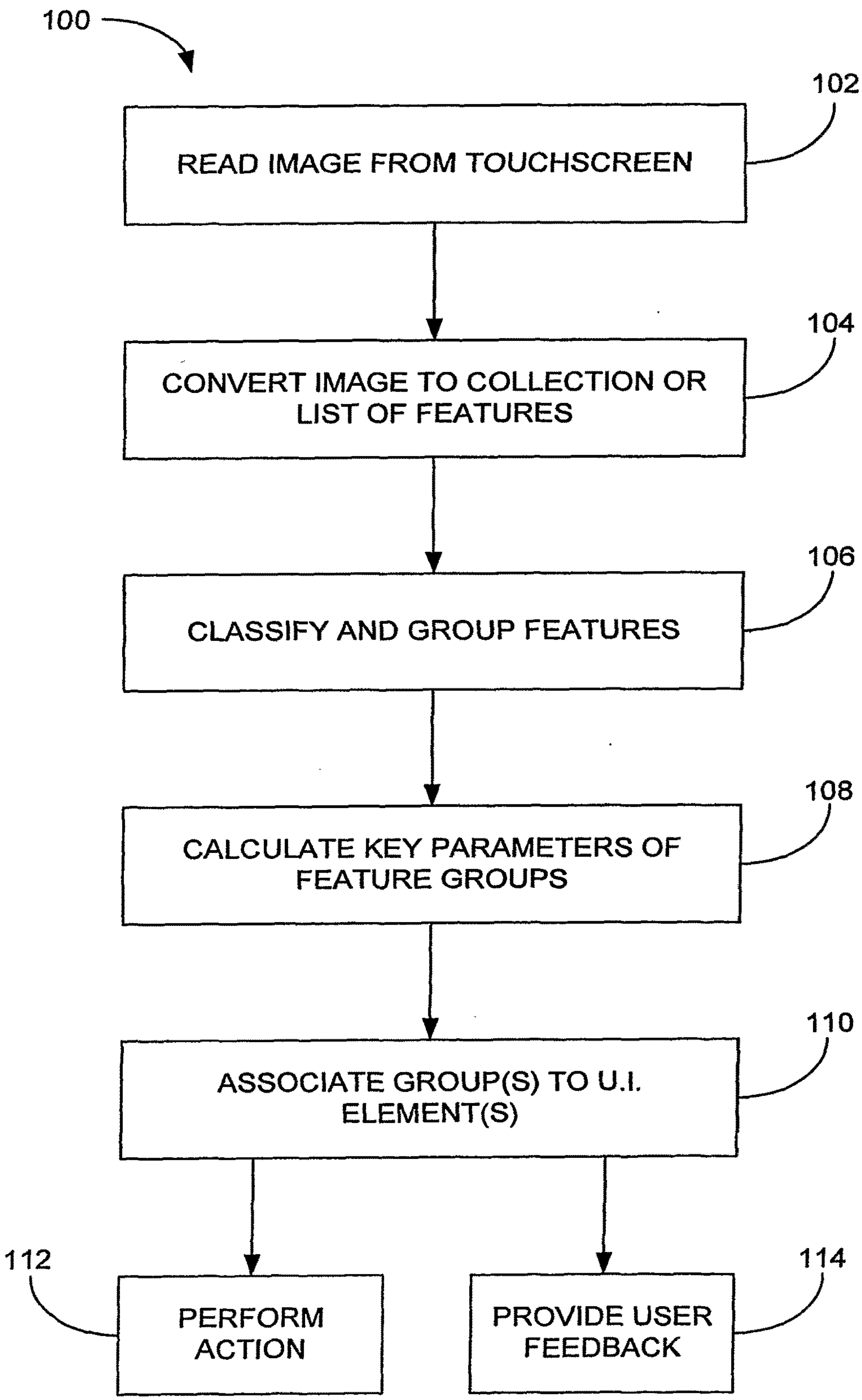

100
102
READ IMAGE FROM TOUCHSCREEN
104
CONVERT IMAGE TO COLLECTION OR LIST OF FEATURES
106
CLASSIFY AND GROUP FEATURES
108
CALCULATE KEY PARAMETERS OF FEATURE GROUPS
110
ASSOCIATE GROUP(S) TO U.I. ELEMENT(S)
112
PERFORM ACTION
114
PROVIDE USER FEEDBACK

FIG. 2

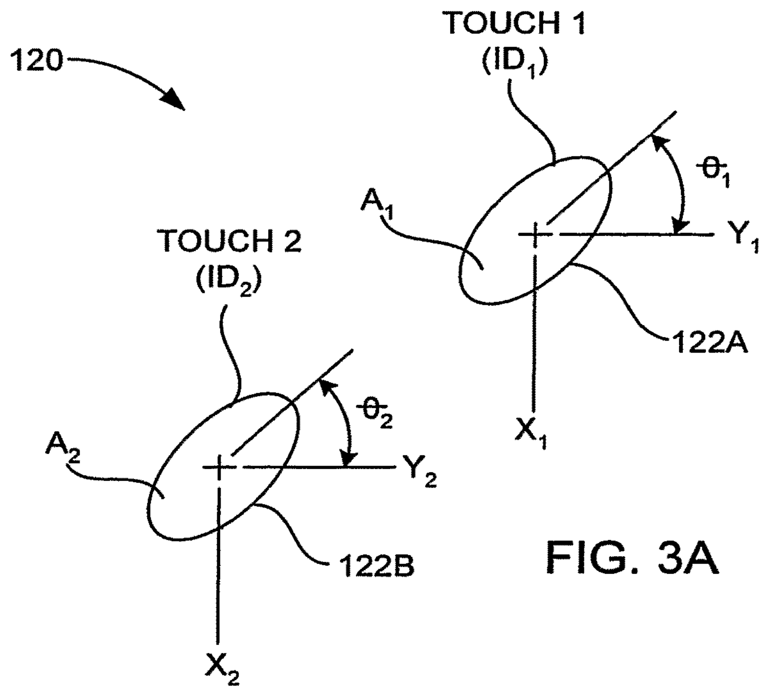

120
TOUCH 1
(ID1)
θ1
A1
Y1
122A
X1
TOUCH 2
(ID2)
θ2
A2
Y2
122B
X2

FIG. 3A

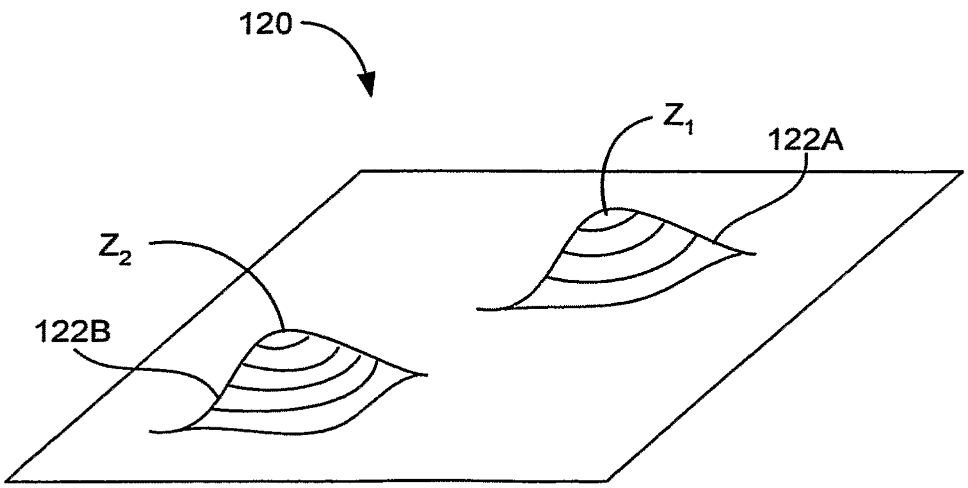

120
Z1
122A
Z2
122B

FIG. 3B

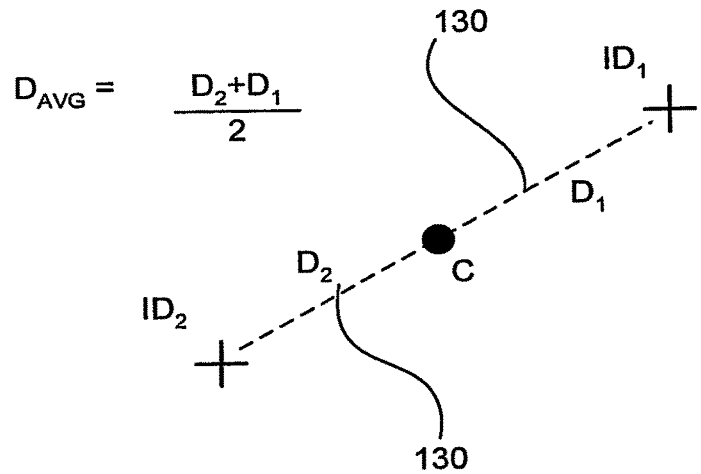


FIG. 4

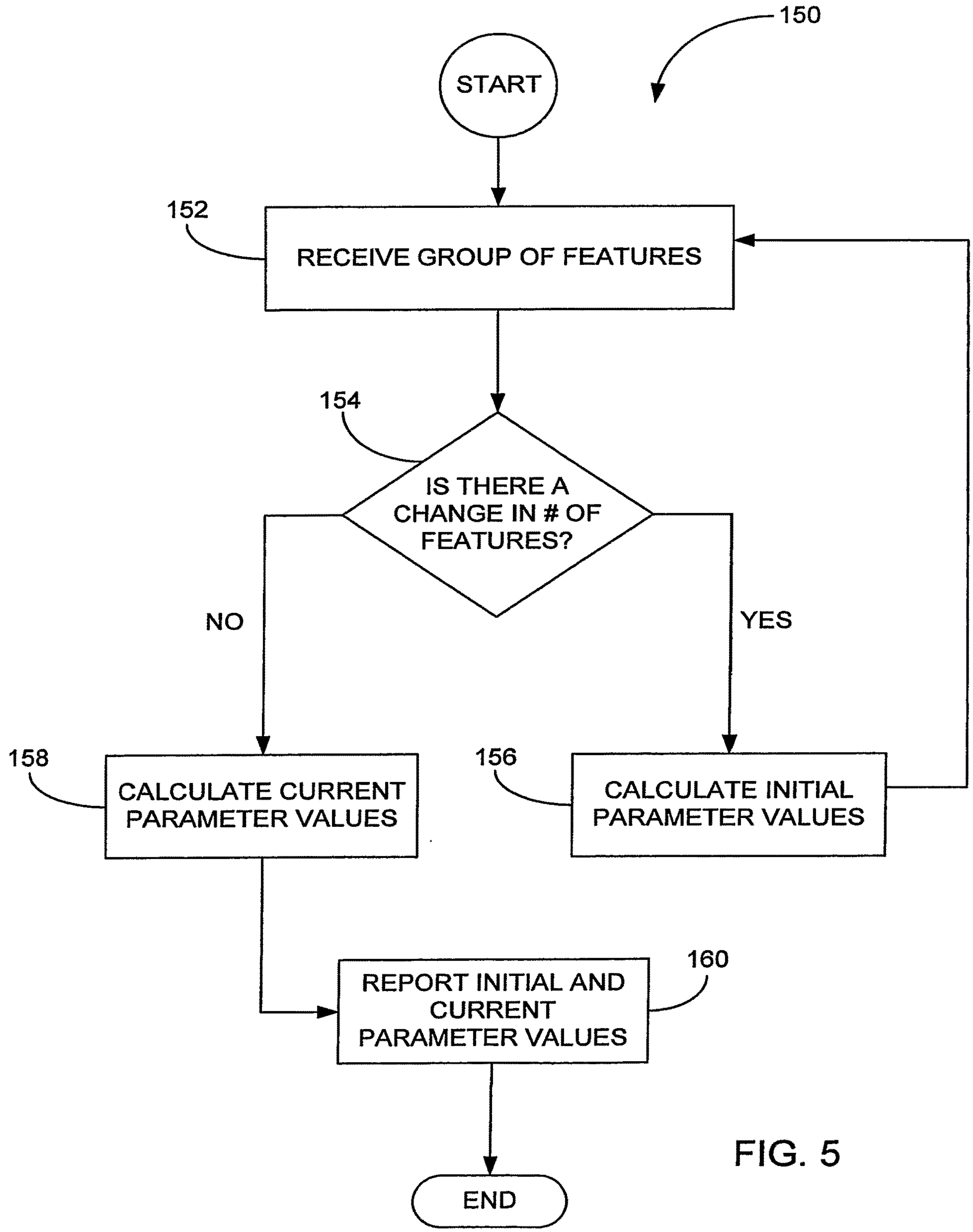

150
START
152
RECEIVE GROUP OF FEATURES
154
IS THERE A CHANGE IN # OF FEATURES?
NO
YES
158
CALCULATE CURRENT PARAMETER VALUES
156
CALCULATE INITIAL PARAMETER VALUES
REPORT INITIAL AND CURRENT PARAMETER VALUES
160
END

FIG. 5

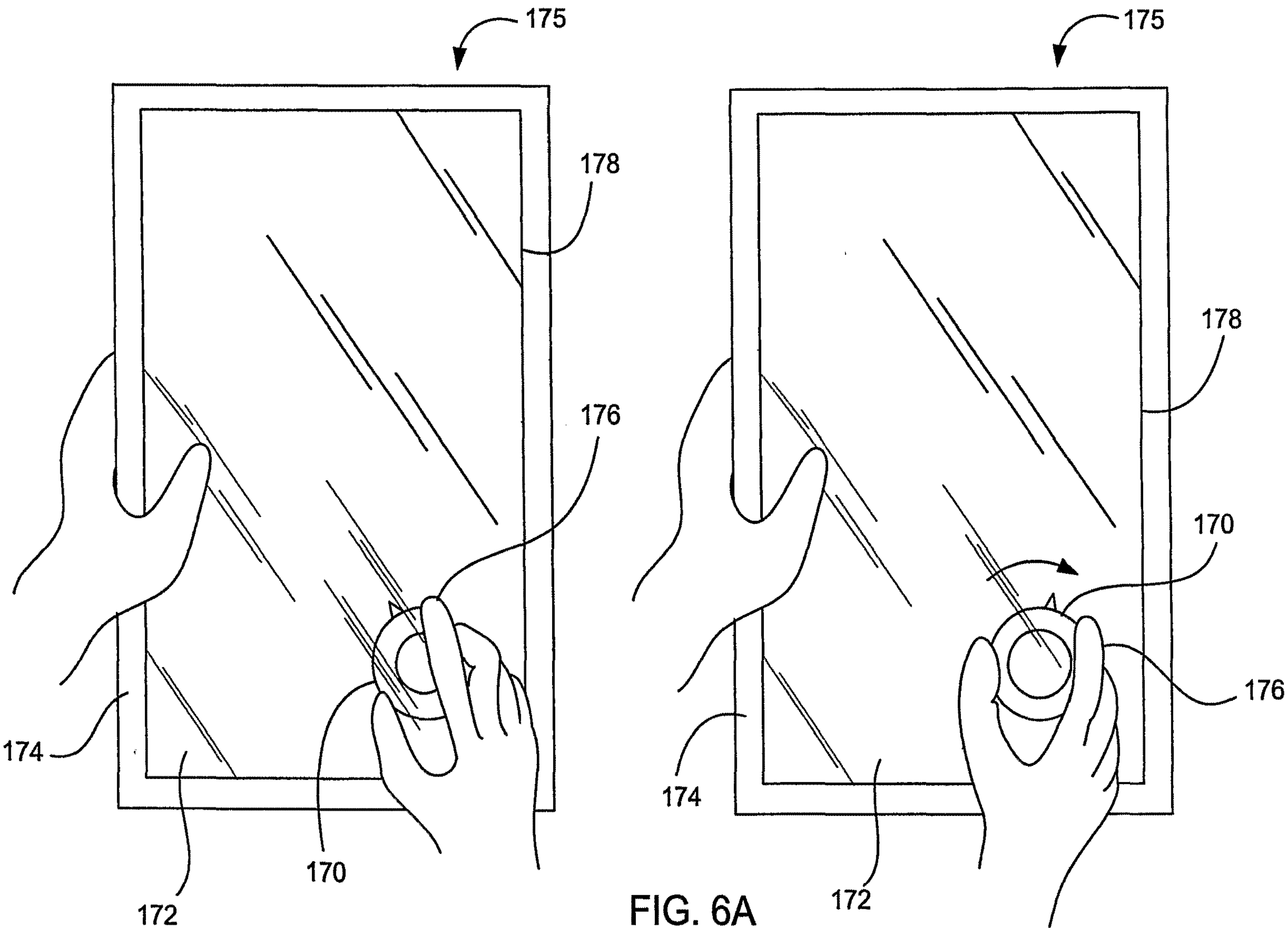


FIG. 6A

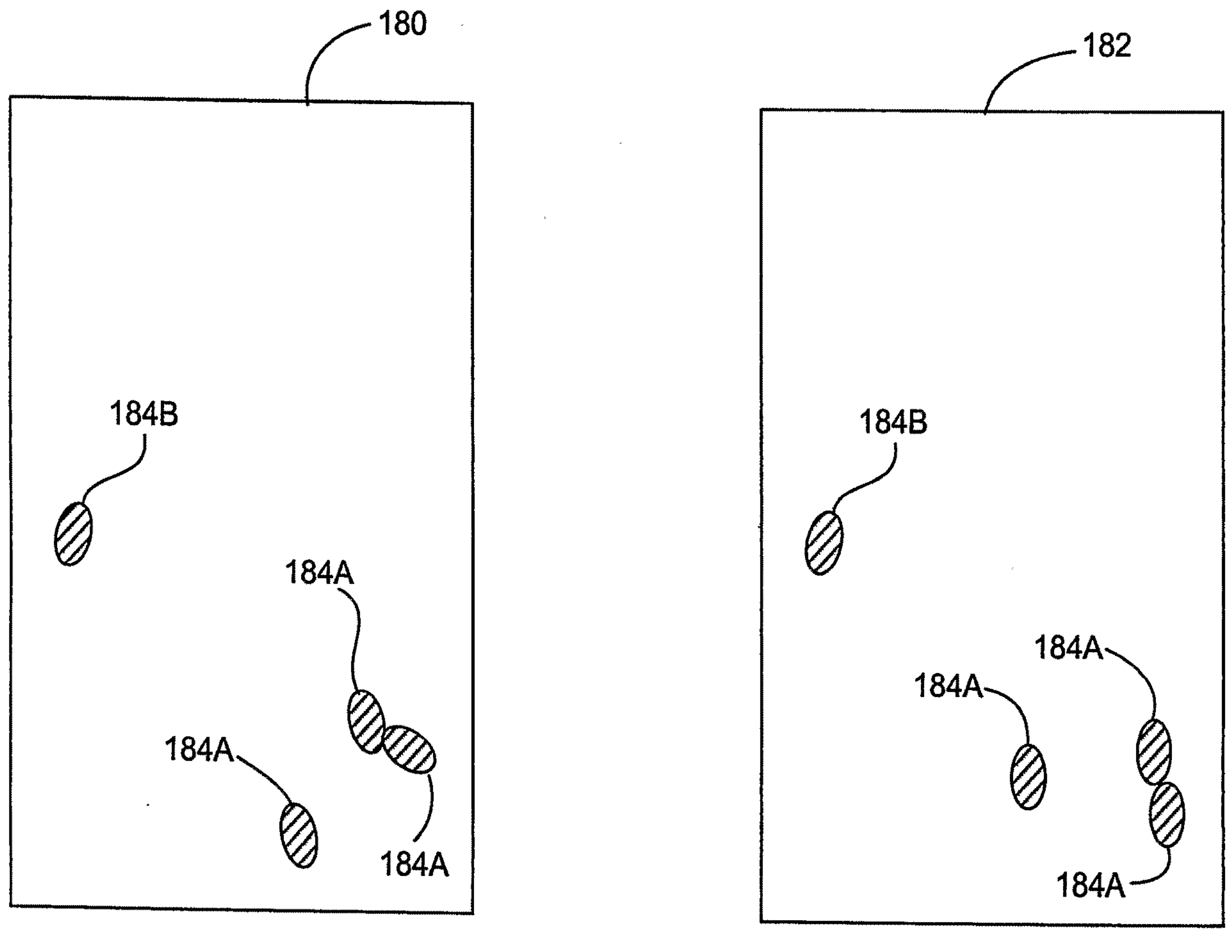


FIG. 6B

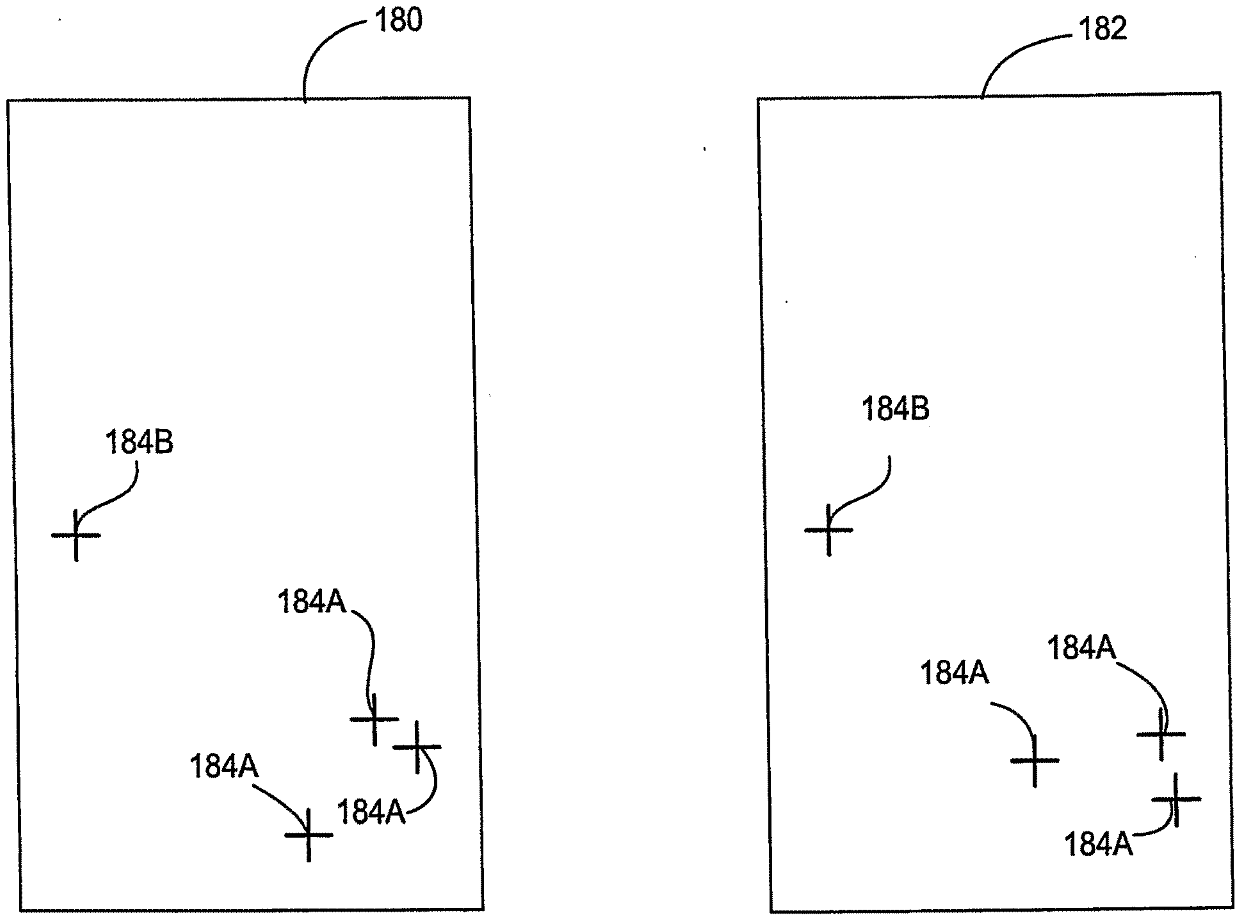


FIG. 6C

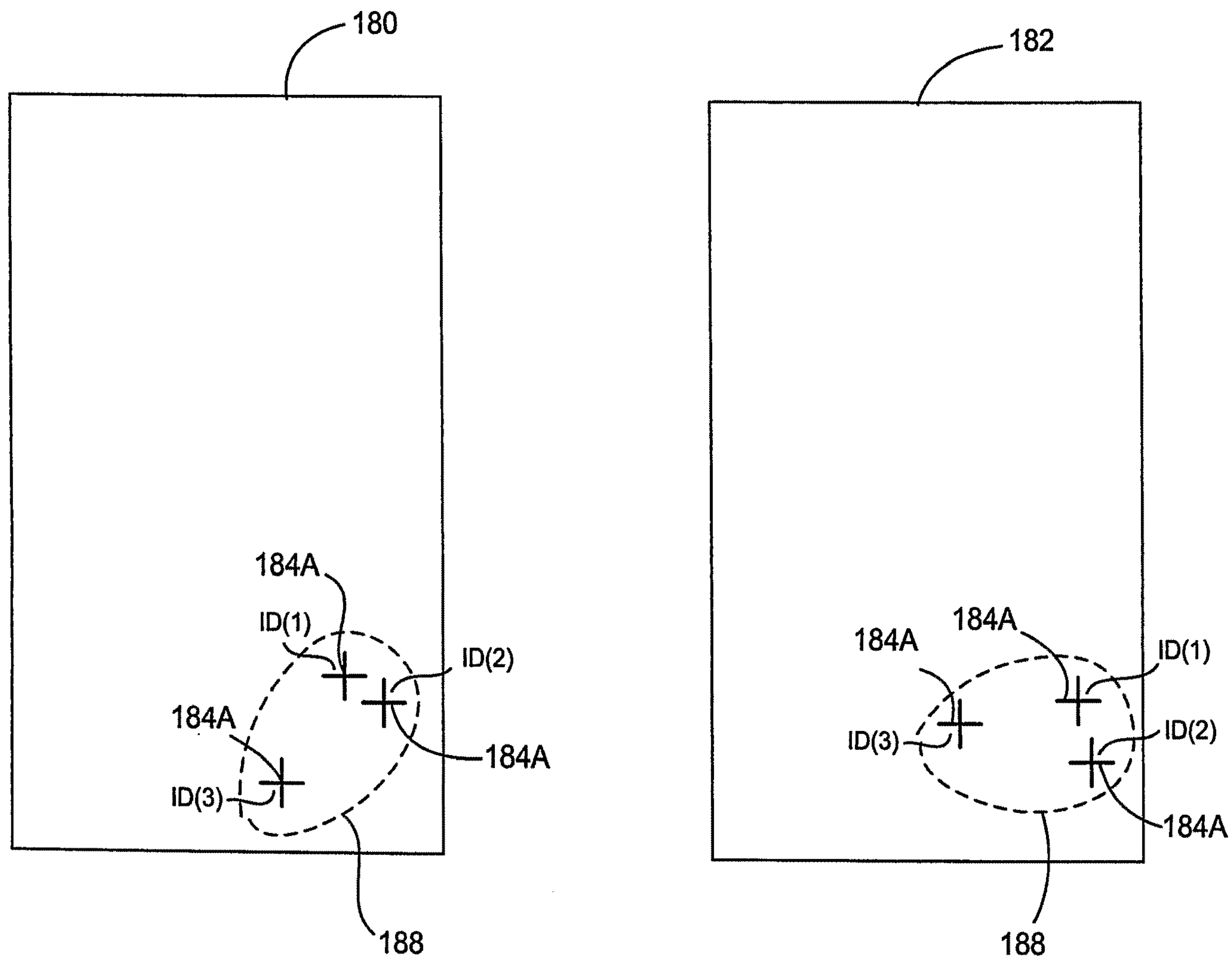

180
184A
ID(1)
ID(2)
184A
184A
ID(3)
188
182
184A
184A
ID(1)
ID(3)
ID(2)
184A
188

FIG. 6D

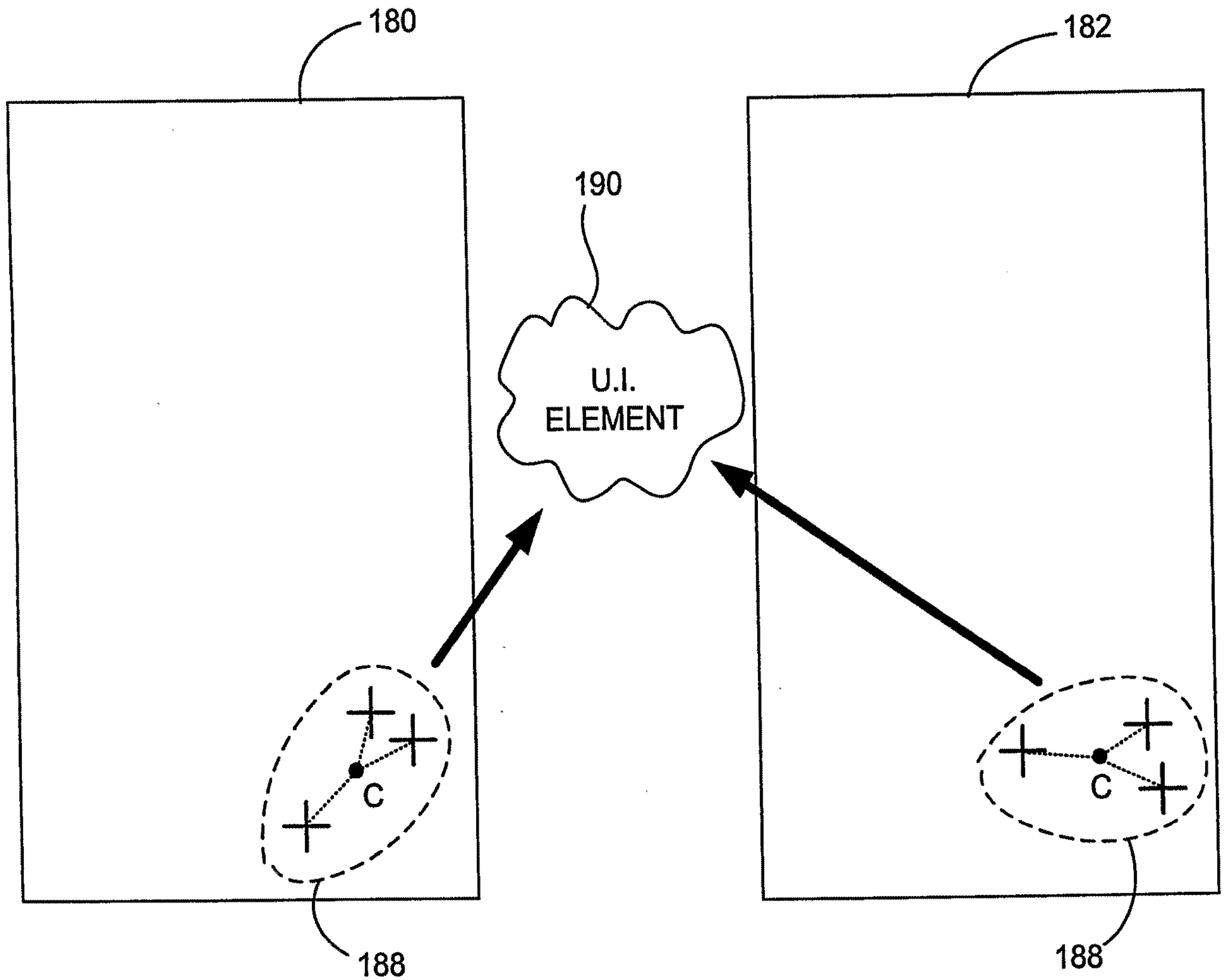


FIG. 6E

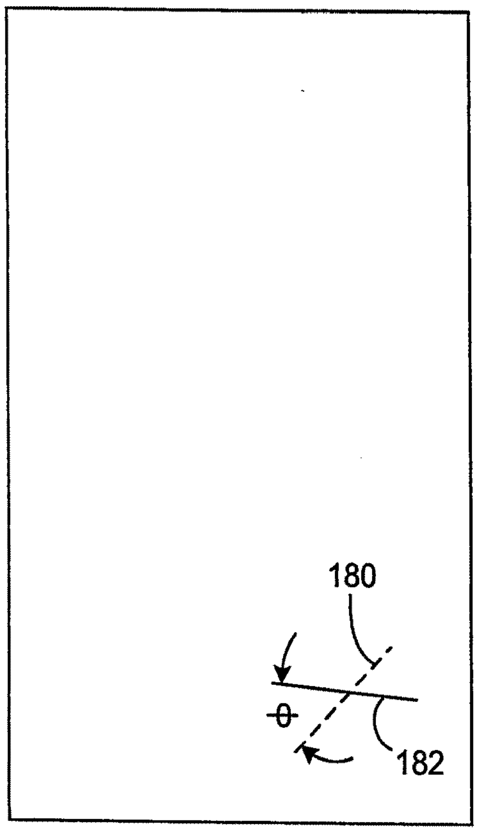


FIG. 6F

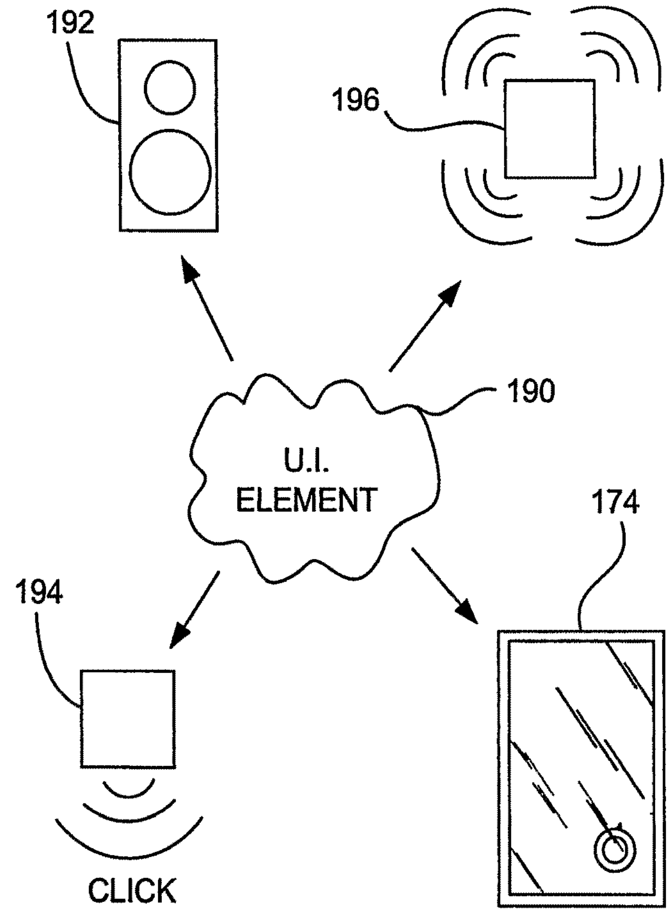


FIG. 6G

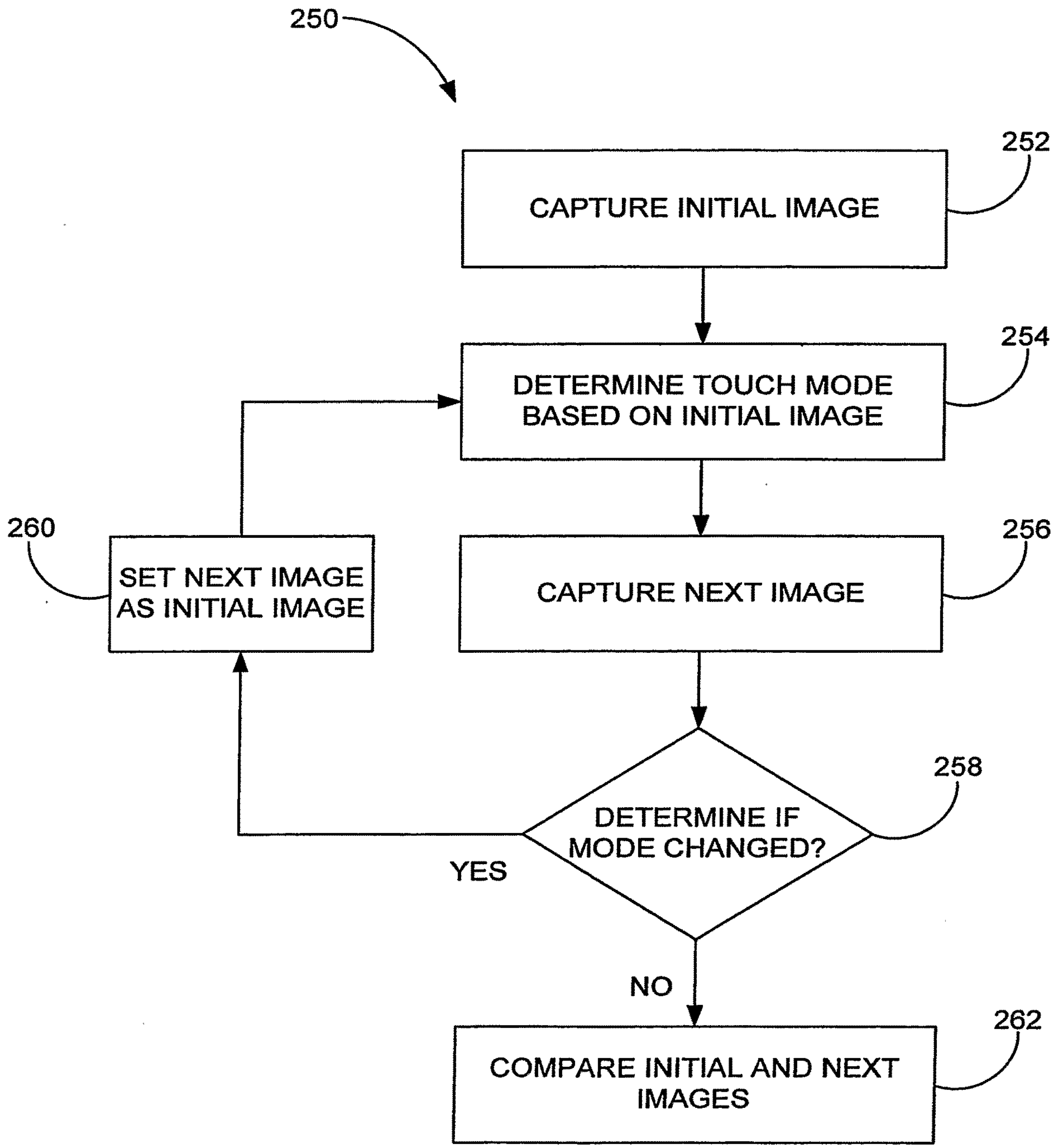

250
CAPTURE INITIAL IMAGE
252
DETERMINE TOUCH MODE BASED ON INITIAL IMAGE
254
260
SET NEXT IMAGE AS INITIAL IMAGE
CAPTURE NEXT IMAGE
256
DETERMINE IF MODE CHANGED?
258
YES
NO
COMPARE INITIAL AND NEXT IMAGES
262

FIG. 8

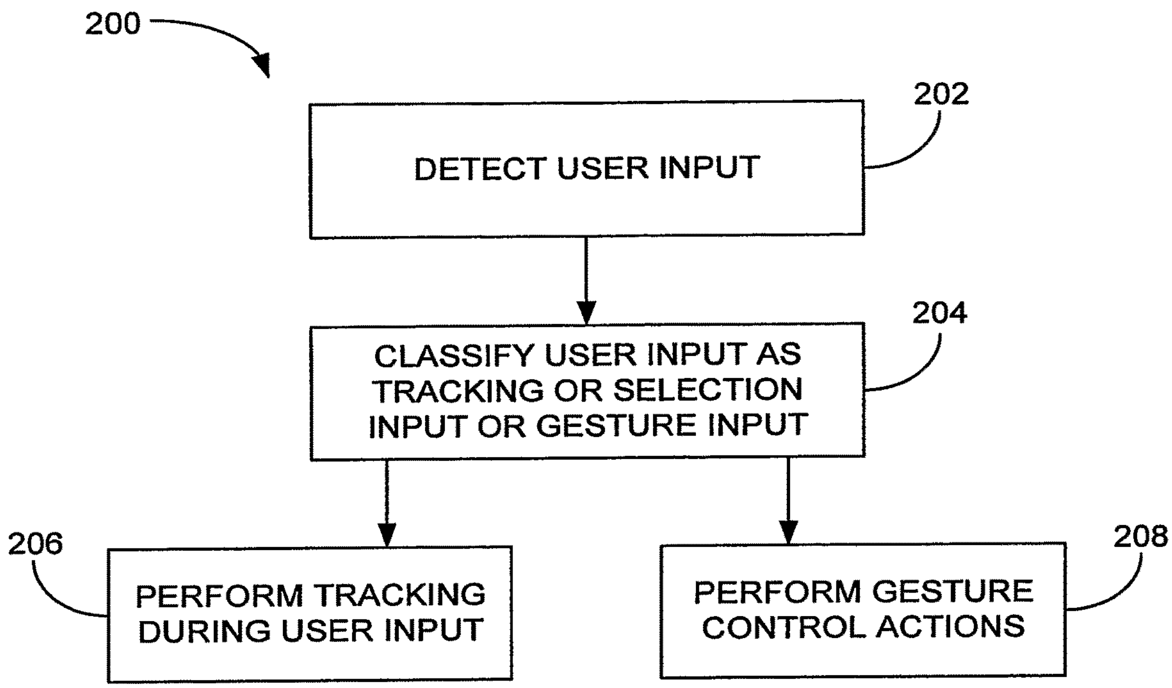

200
202
DETECT USER INPUT
204
CLASSIFY USER INPUT AS TRACKING OR SELECTION INPUT OR GESTURE INPUT
206
PERFORM TRACKING DURING USER INPUT
208
PERFORM GESTURE CONTROL ACTIONS

FIG. 7

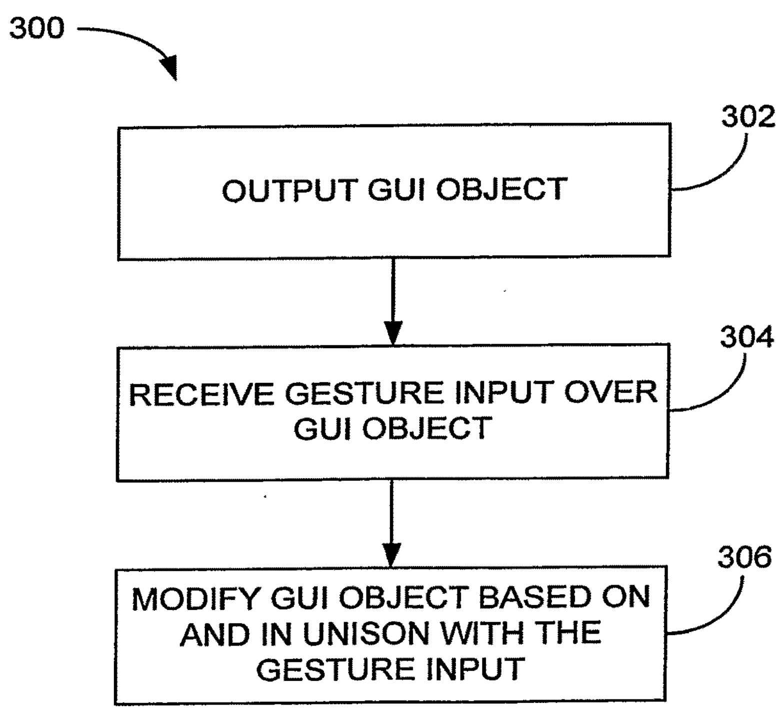

300
302
OUTPUT GUI OBJECT
304
RECEIVE GESTURE INPUT OVER GUI OBJECT
306
MODIFY GUI OBJECT BASED ON AND IN UNISON WITH THE GESTURE INPUT

FIG. 9

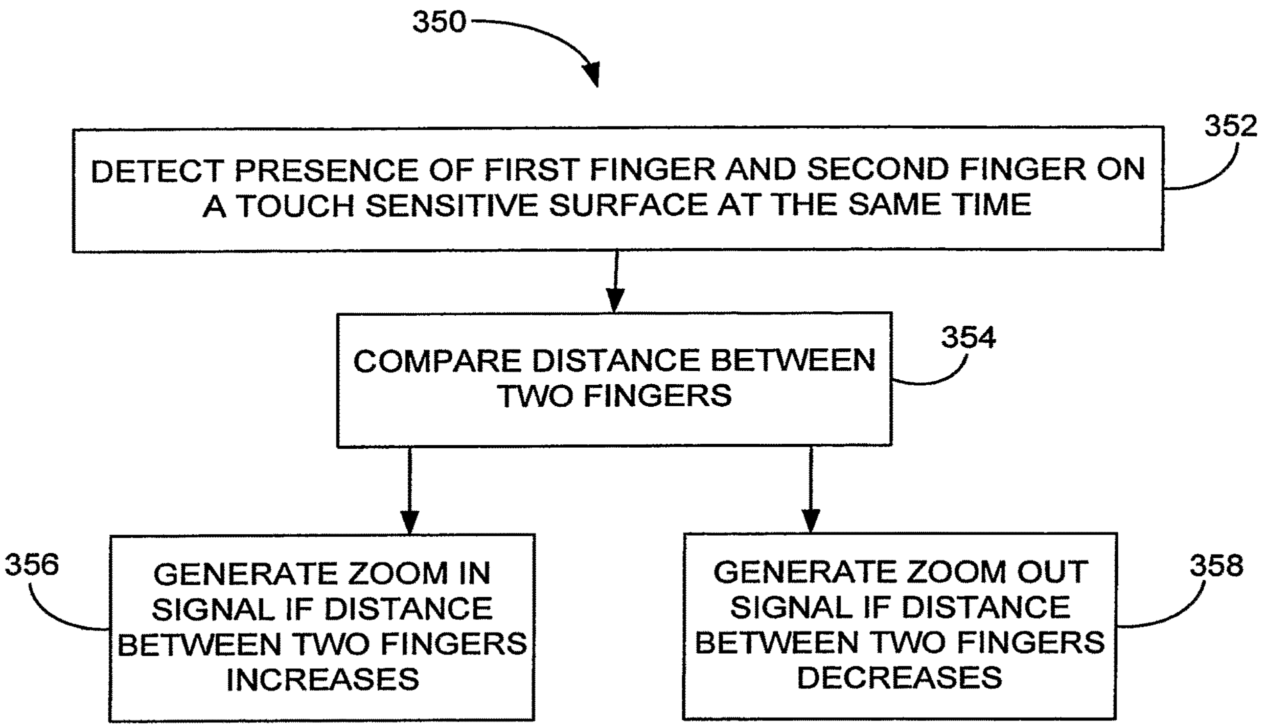

350
DETECT PRESENCE OF FIRST FINGER AND SECOND FINGER ON A TOUCH SENSITIVE SURFACE AT THE SAME TIME
352
COMPARE DISTANCE BETWEEN TWO FINGERS
354
GENERATE ZOOM IN SIGNAL IF DISTANCE BETWEEN TWO FINGERS INCREASES
356
GENERATE ZOOM OUT SIGNAL IF DISTANCE BETWEEN TWO FINGERS DECREASES
358

FIG. 10

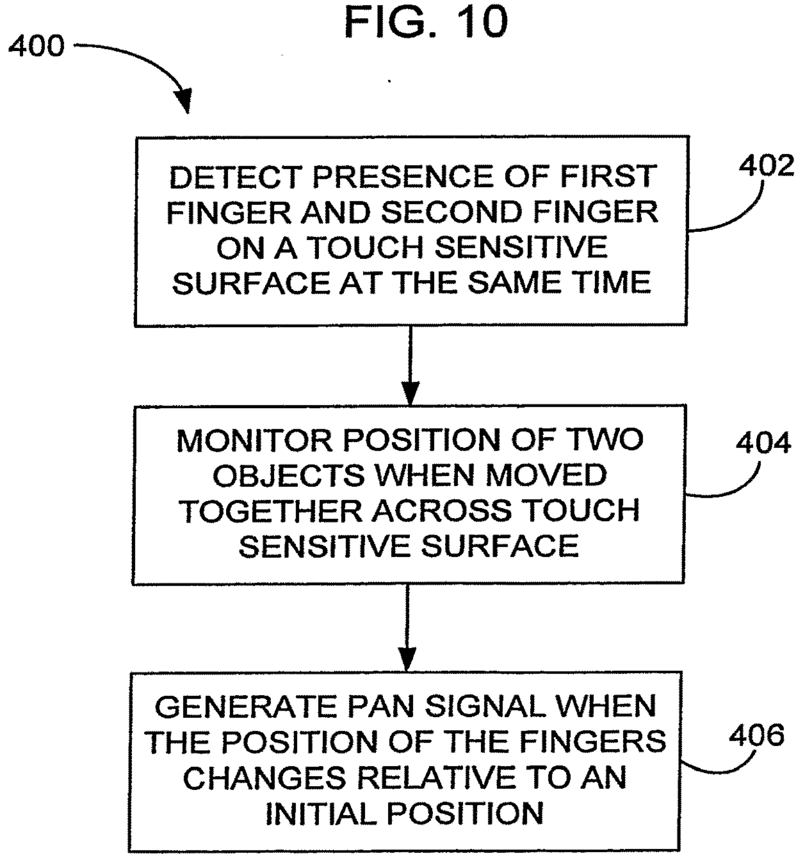

400
DETECT PRESENCE OF FIRST FINGER AND SECOND FINGER ON A TOUCH SENSITIVE SURFACE AT THE SAME TIME
402
MONITOR POSITION OF TWO OBJECTS WHEN MOVED TOGETHER ACROSS TOUCH SENSITIVE SURFACE
404
GENERATE PAN SIGNAL WHEN THE POSITION OF THE FINGERS CHANGES RELATIVE TO AN INITIAL POSITION
406

FIG. 12

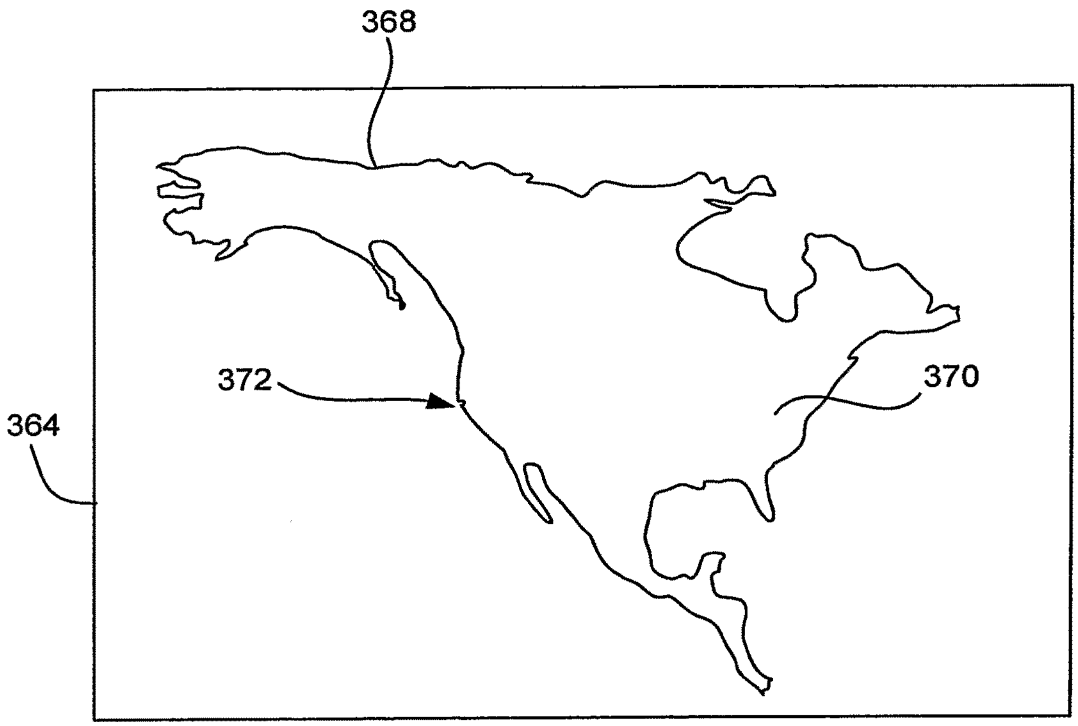


FIG. 11A

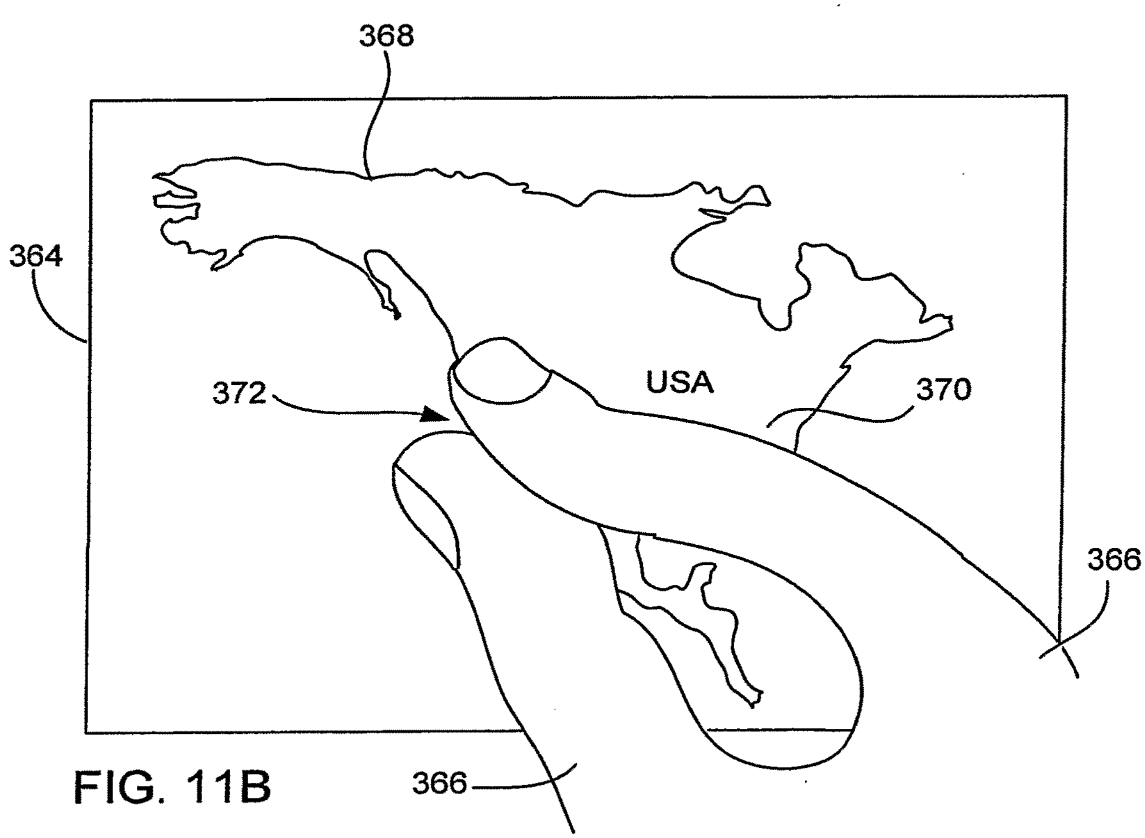


FIG. 11B

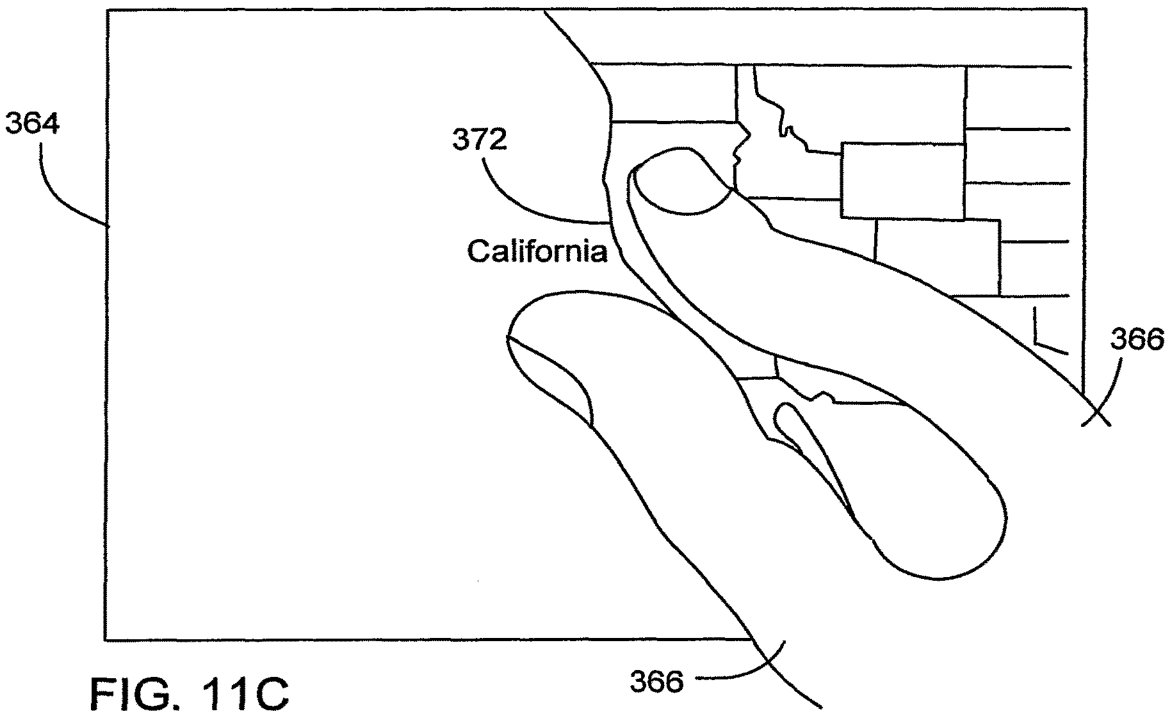


FIG. 11C

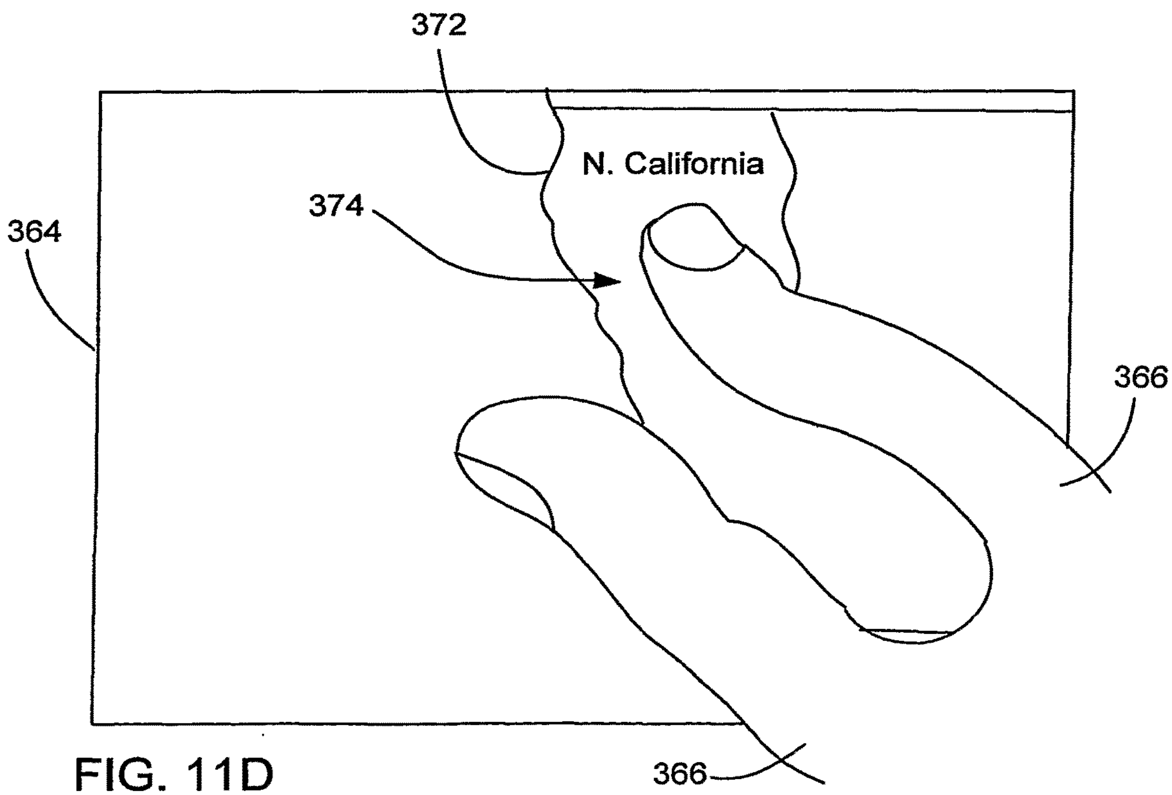


FIG. 11D

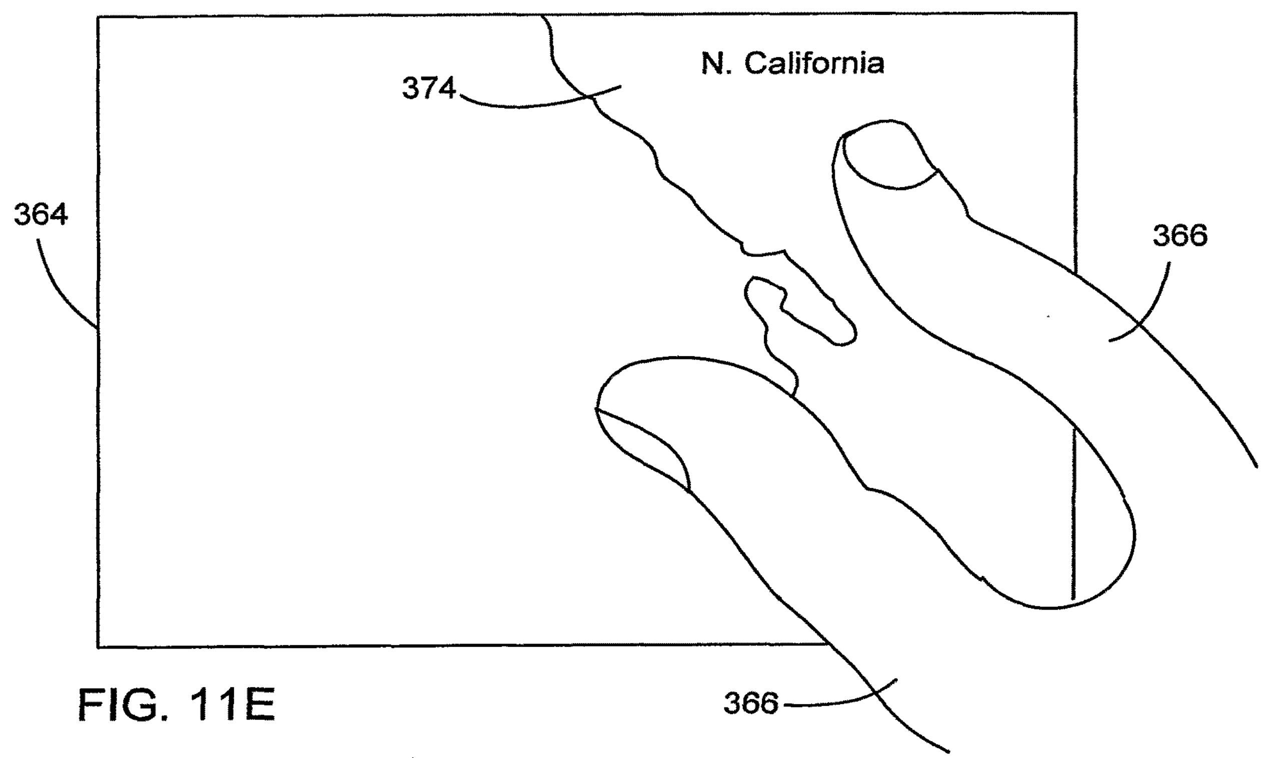


FIG. 11E

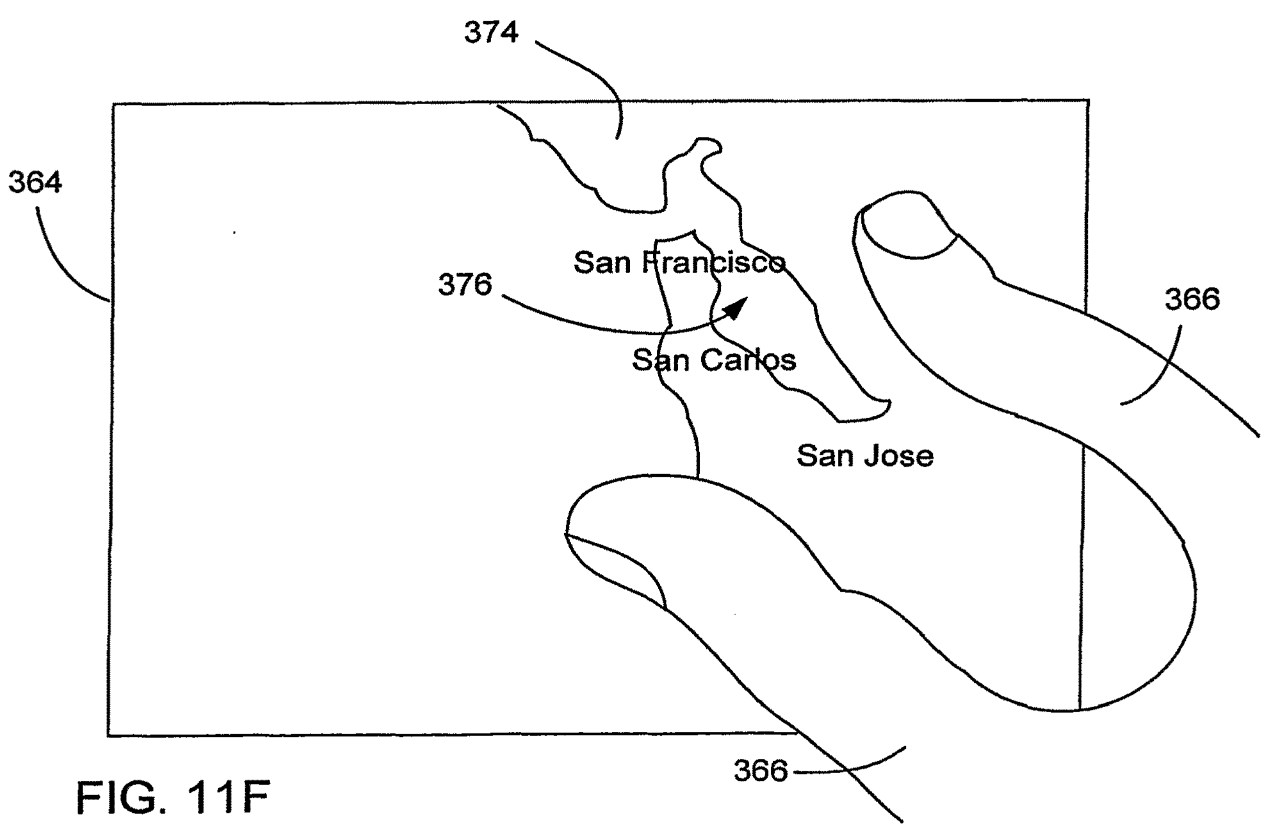


FIG. 11F

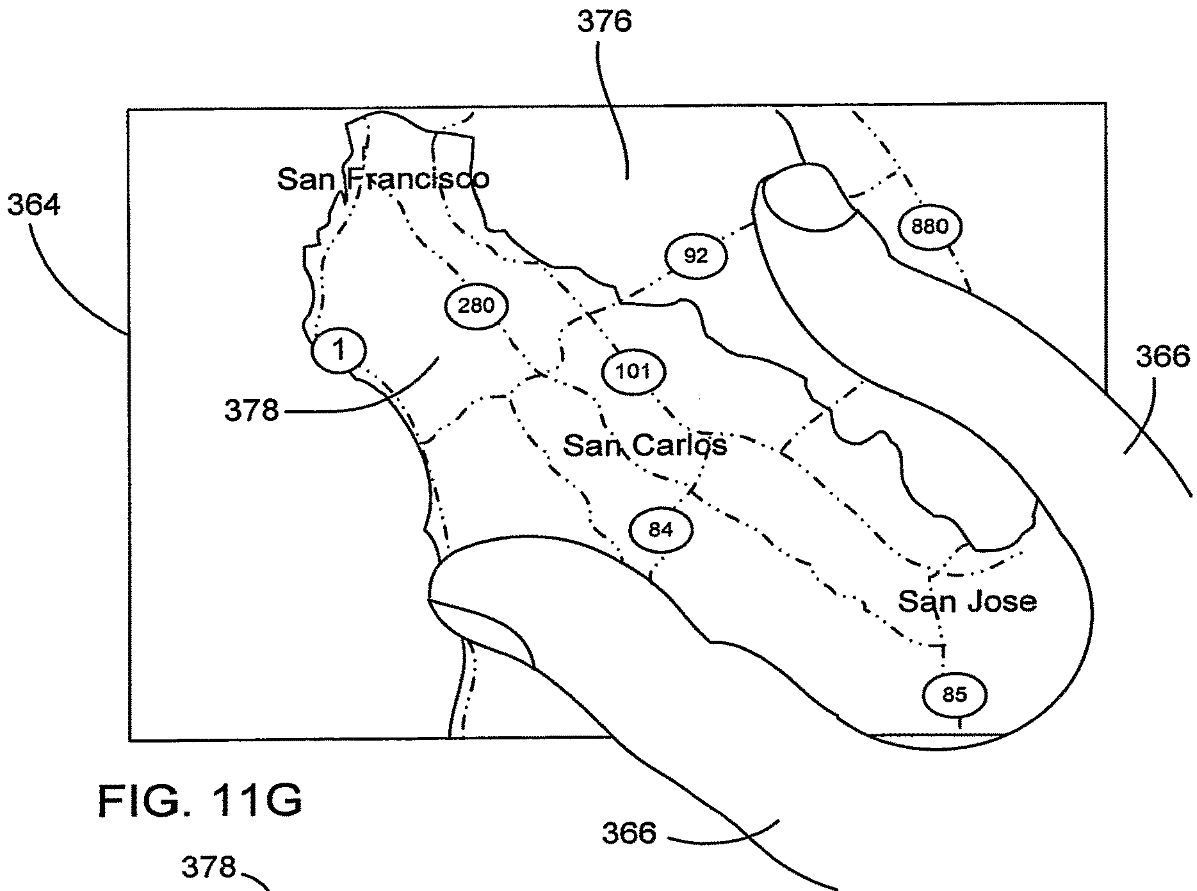


FIG. 11G

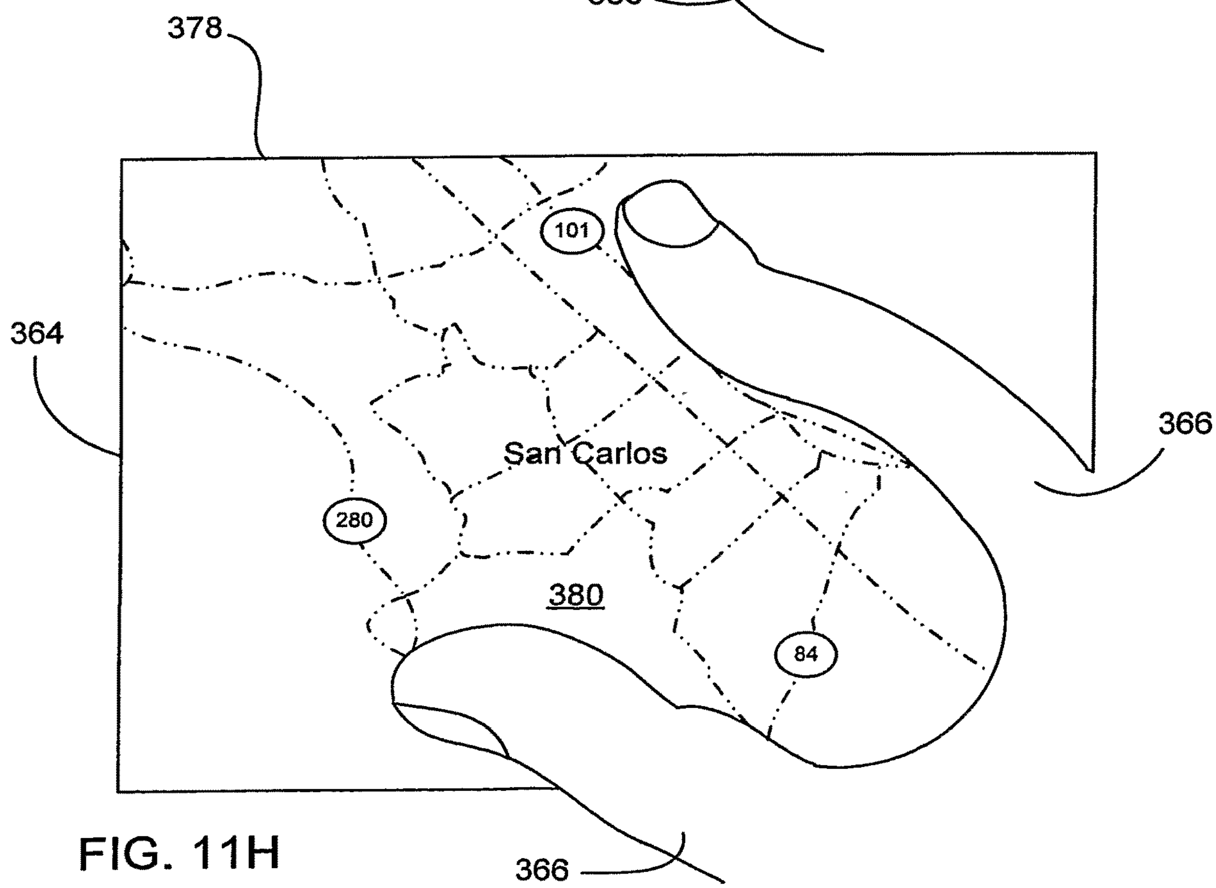


FIG. 11H

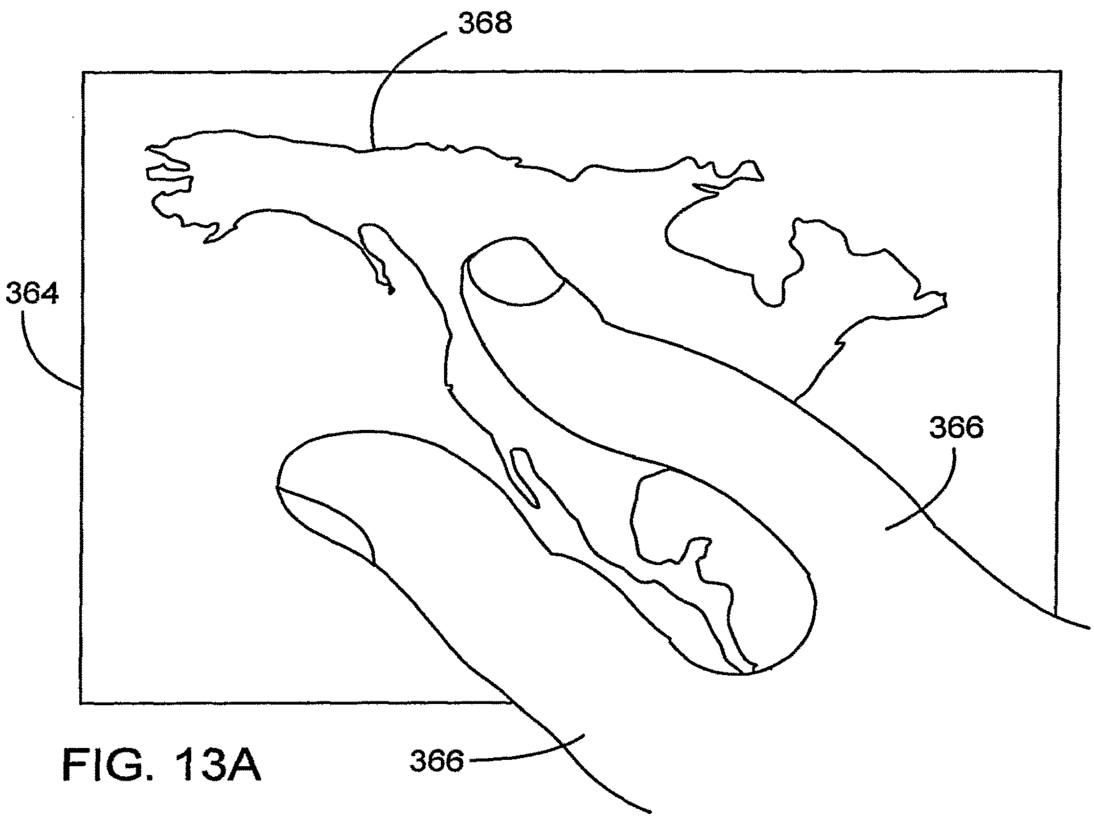


FIG. 13A

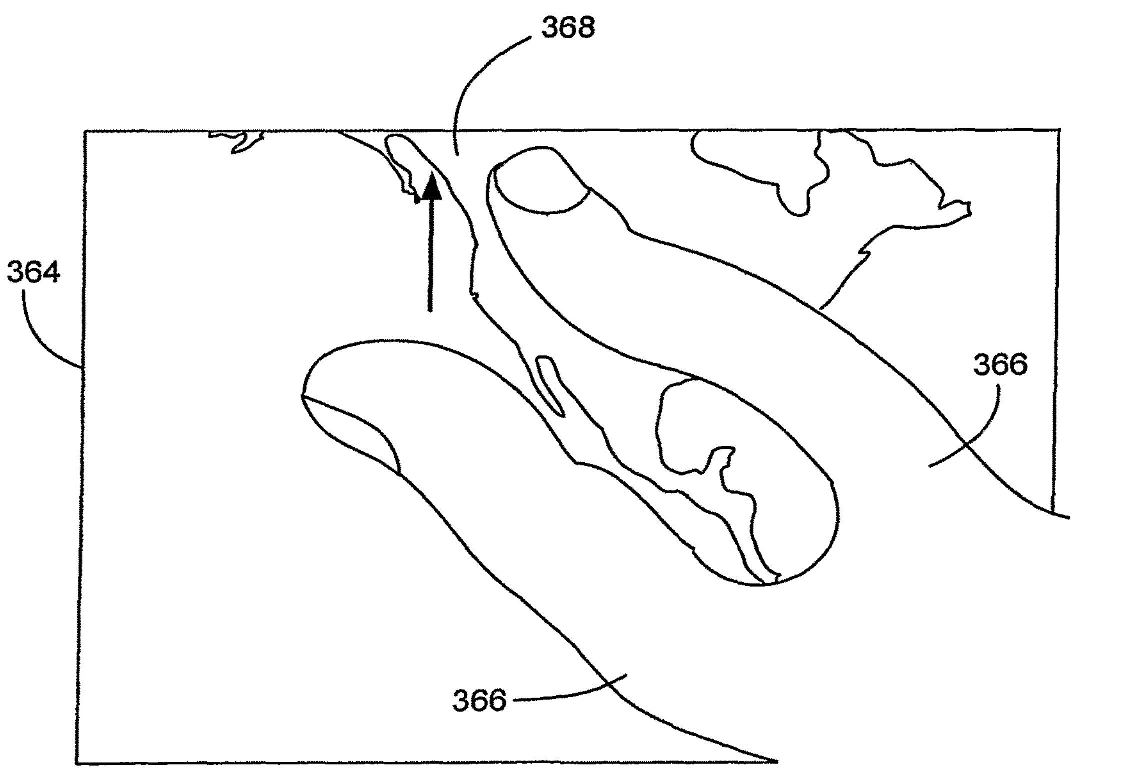


FIG. 13B

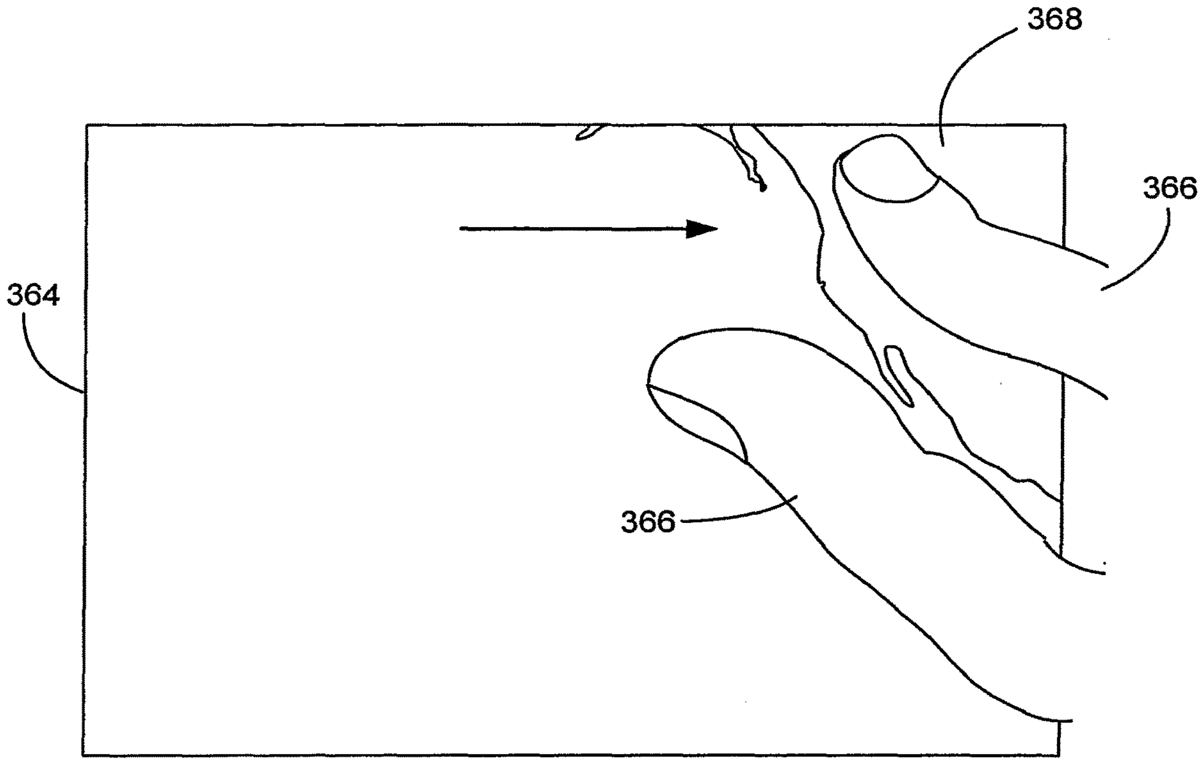


FIG. 13C

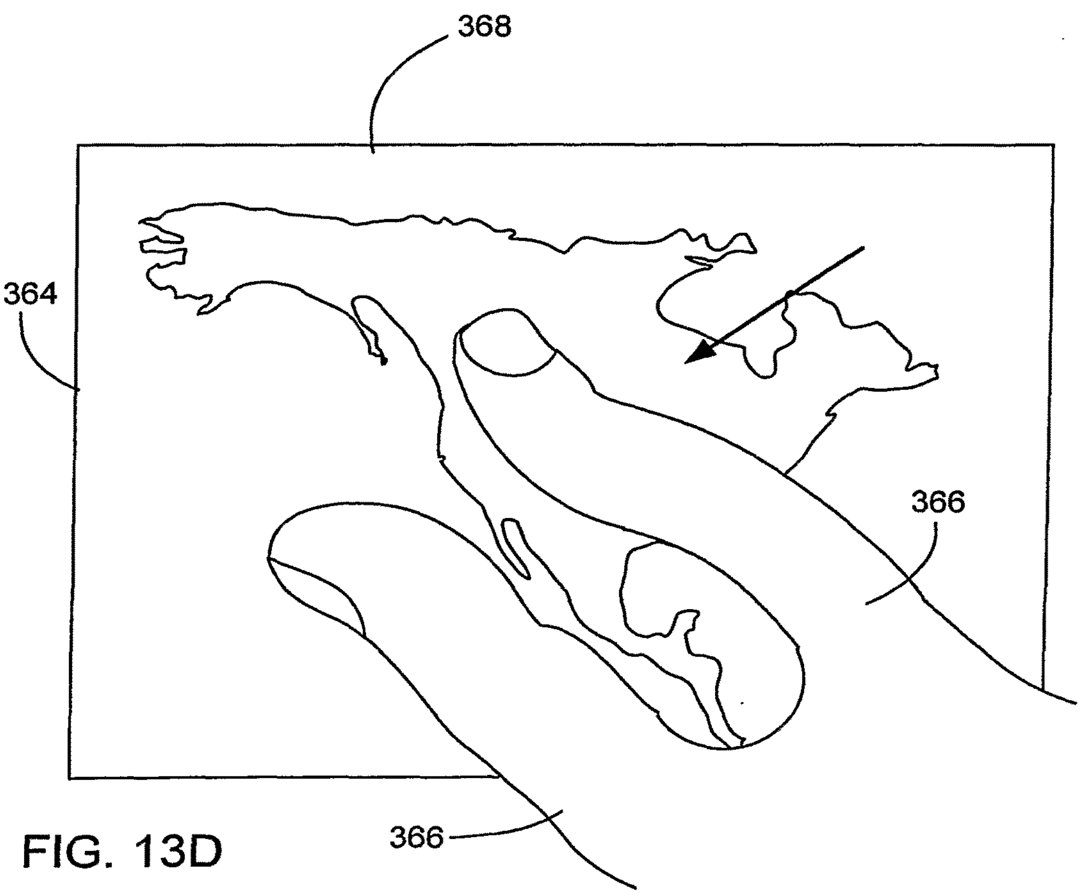


FIG. 13D

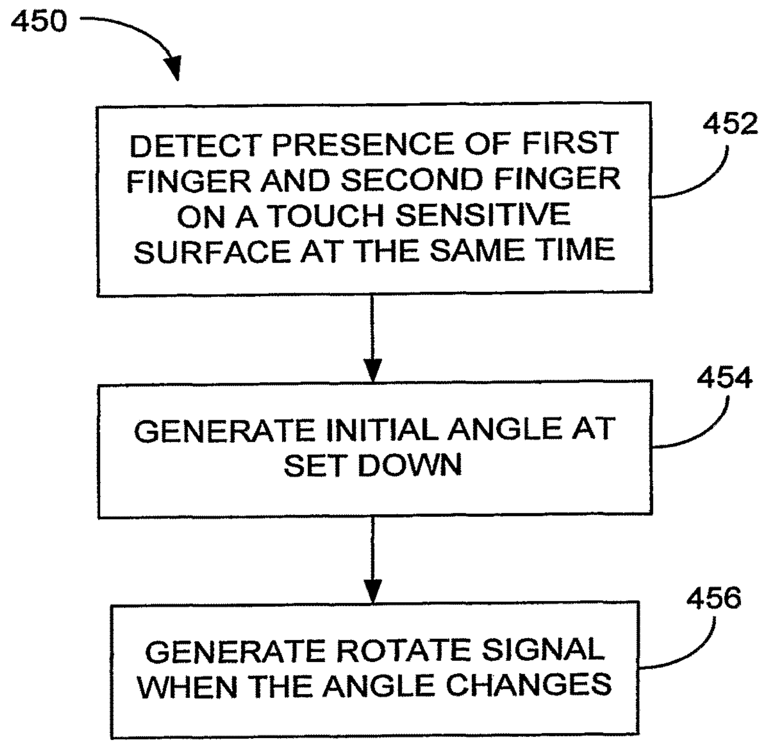

450
DETECT PRESENCE OF FIRST FINGER AND SECOND FINGER ON A TOUCH SENSITIVE SURFACE AT THE SAME TIME
452
GENERATE INITIAL ANGLE AT SET DOWN
454
GENERATE ROTATE SIGNAL WHEN THE ANGLE CHANGES
456

FIG. 14

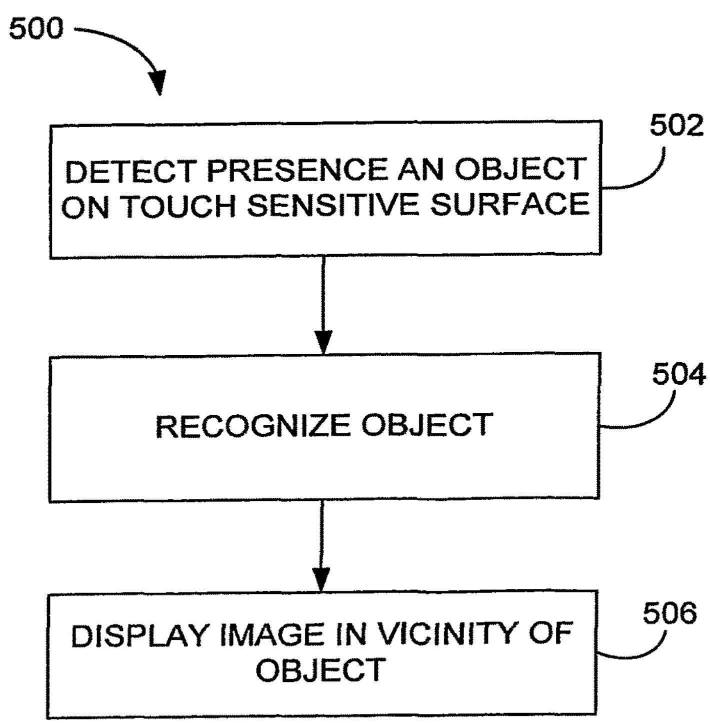

500
DETECT PRESENCE AN OBJECT ON TOUCH SENSITIVE SURFACE
502
RECOGNIZE OBJECT
504
DISPLAY IMAGE IN VICINITY OF OBJECT
506

FIG. 16

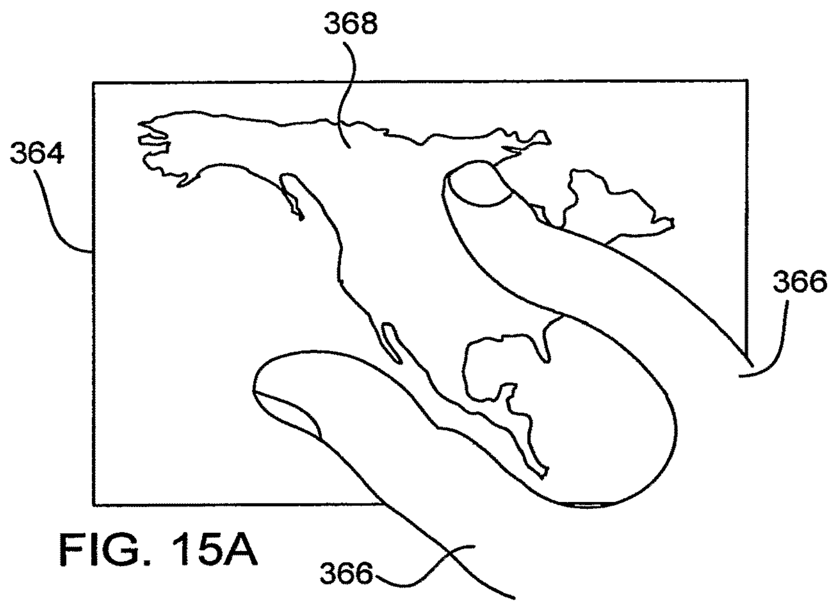


FIG. 15A

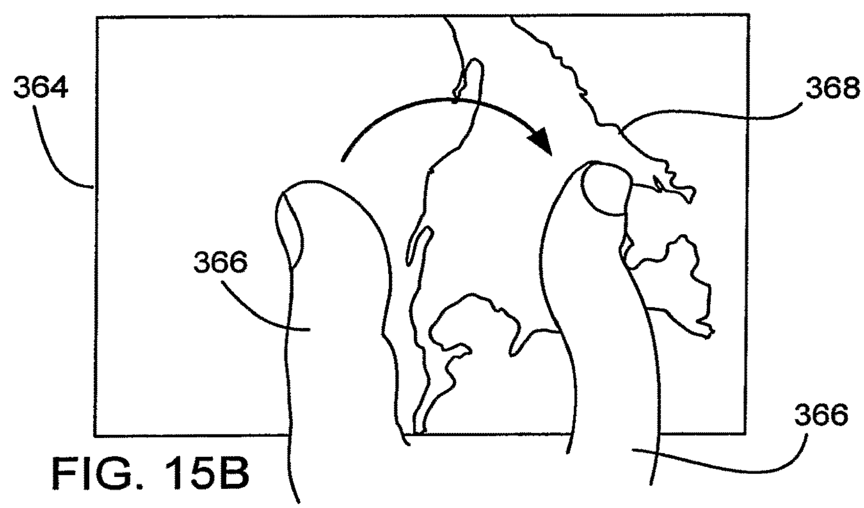


FIG. 15B

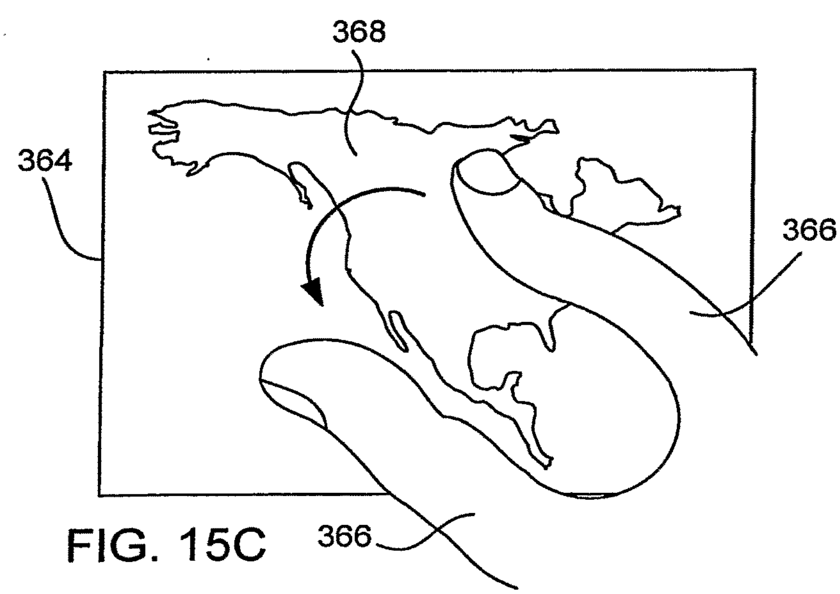


FIG. 15C

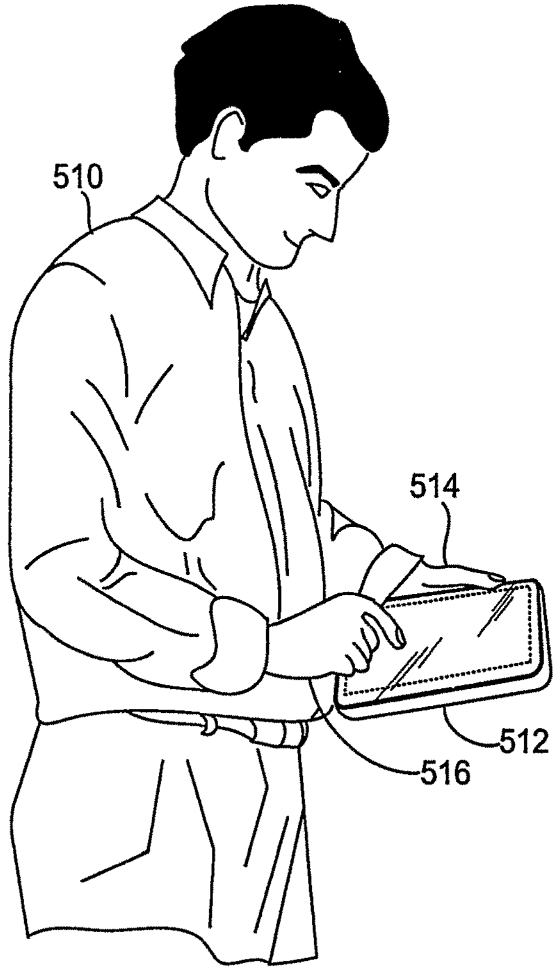


FIG. 17A

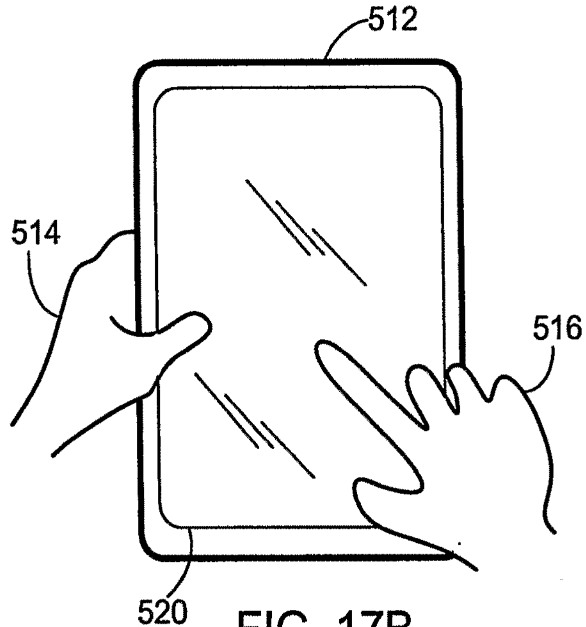


FIG. 17B

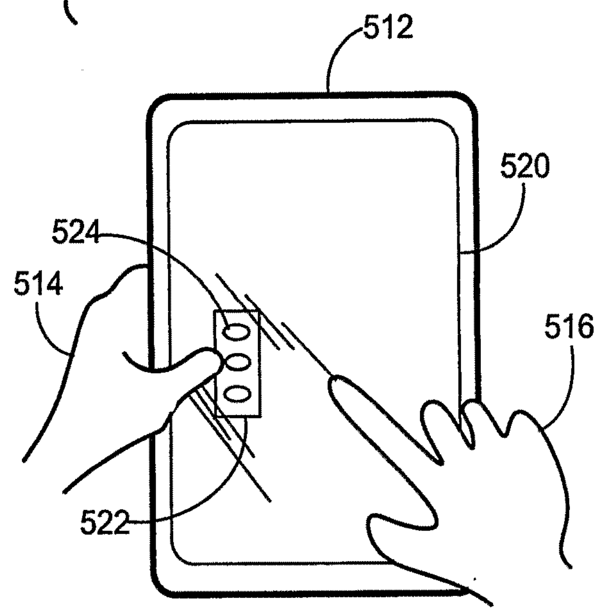


FIG. 17C

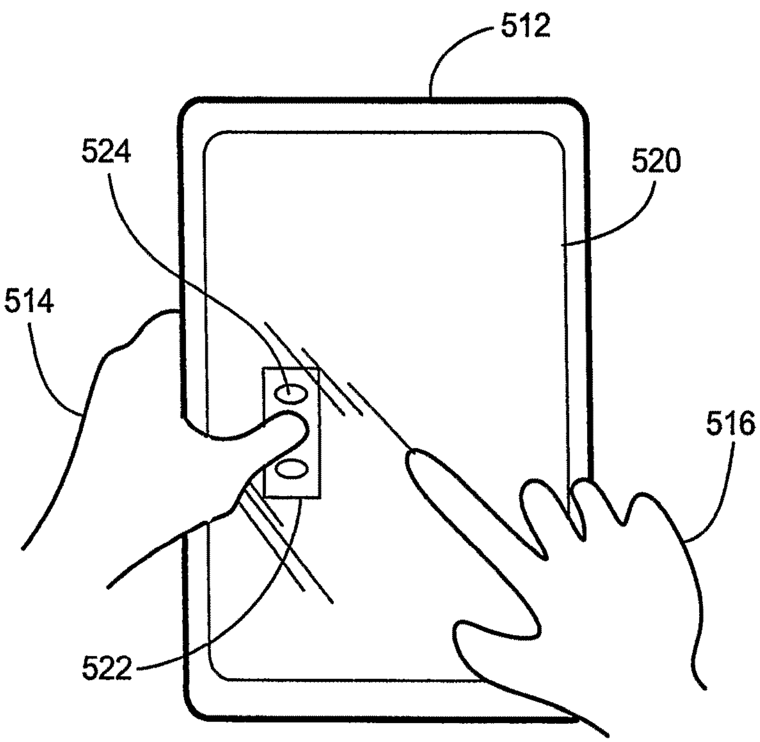


FIG. 17D

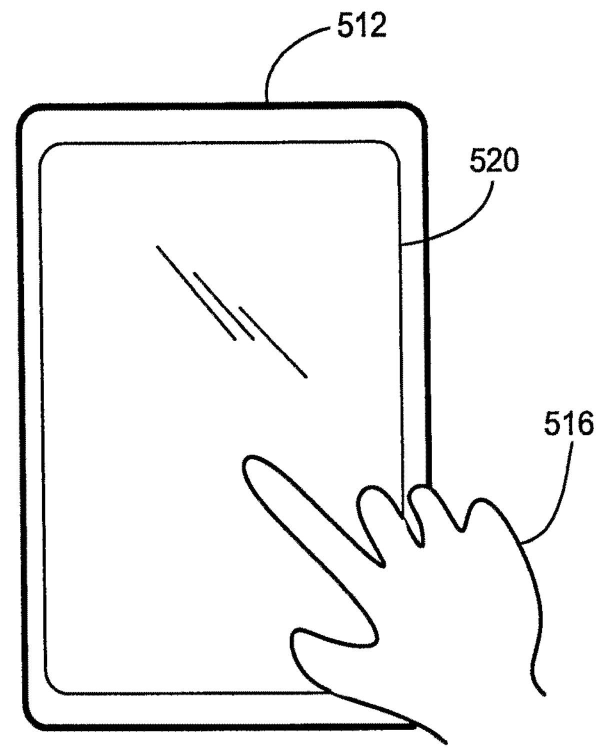


FIG. 17E

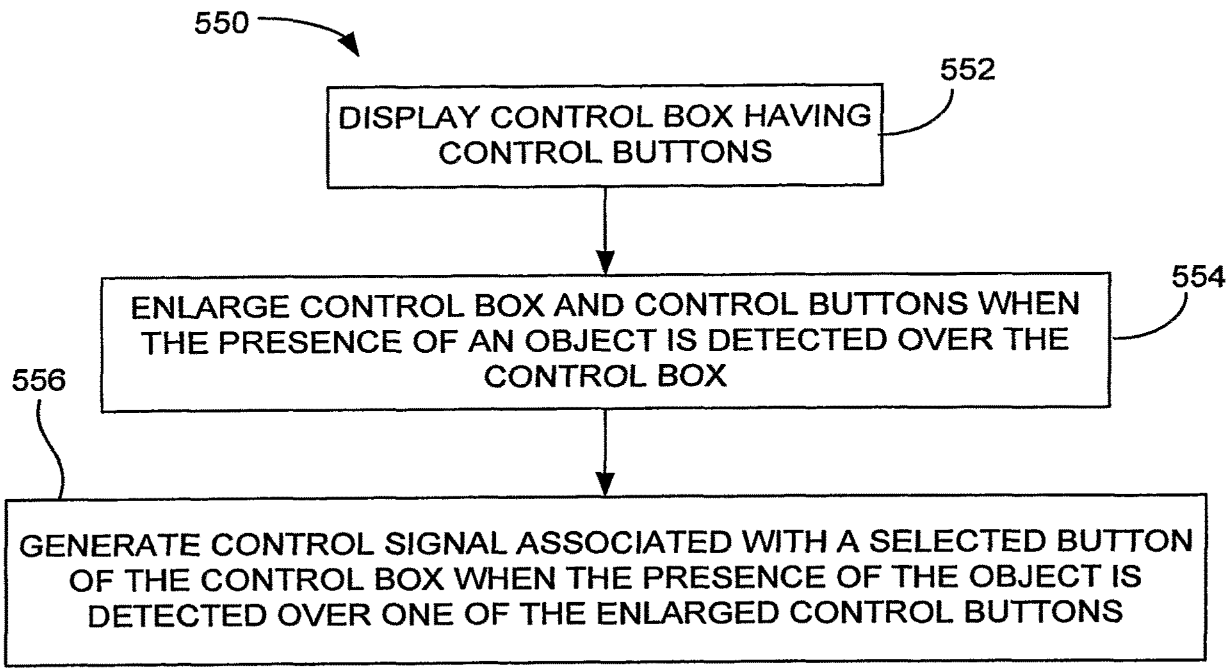

550
552
DISPLAY CONTROL BOX HAVING CONTROL BUTTONS
554
ENLARGE CONTROL BOX AND CONTROL BUTTONS WHEN THE PRESENCE OF AN OBJECT IS DETECTED OVER THE CONTROL BOX
556
GENERATE CONTROL SIGNAL ASSOCIATED WITH A SELECTED BUTTON OF THE CONTROL BOX WHEN THE PRESENCE OF THE OBJECT IS DETECTED OVER ONE OF THE ENLARGED CONTROL BUTTONS

FIG. 18

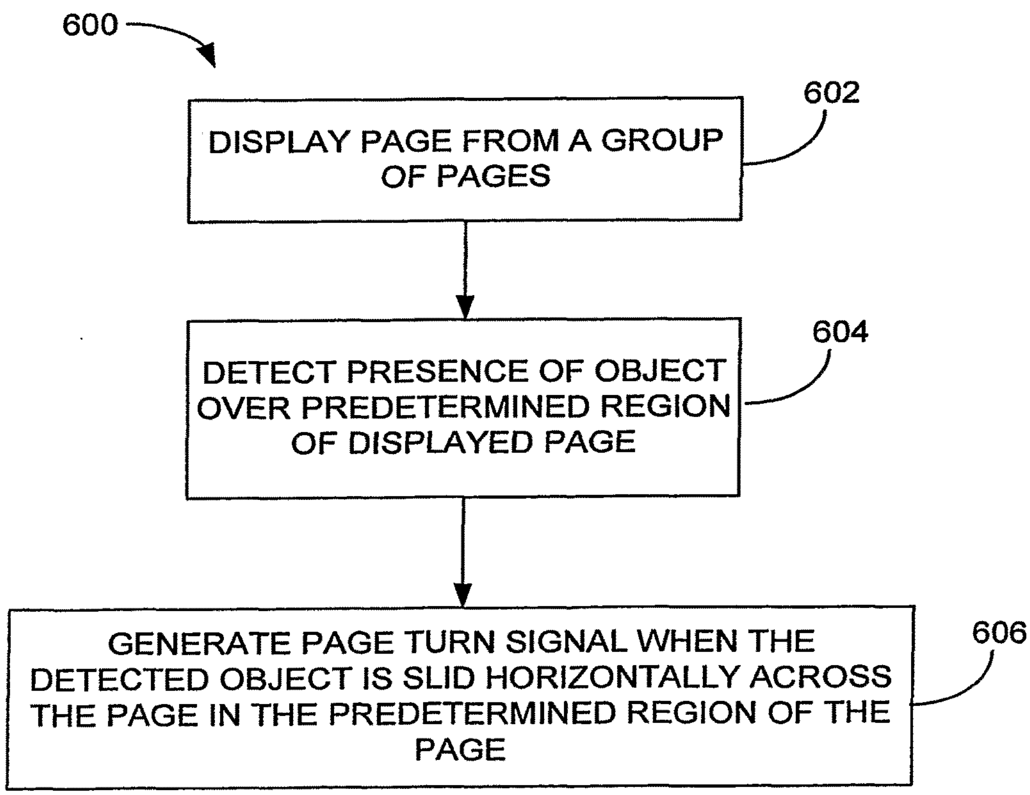

600
602
DISPLAY PAGE FROM A GROUP OF PAGES
604
DETECT PRESENCE OF OBJECT OVER PREDETERMINED REGION OF DISPLAYED PAGE
606
GENERATE PAGE TURN SIGNAL WHEN THE DETECTED OBJECT IS SLID HORIZONTALLY ACROSS THE PAGE IN THE PREDETERMINED REGION OF THE PAGE

FIG. 20

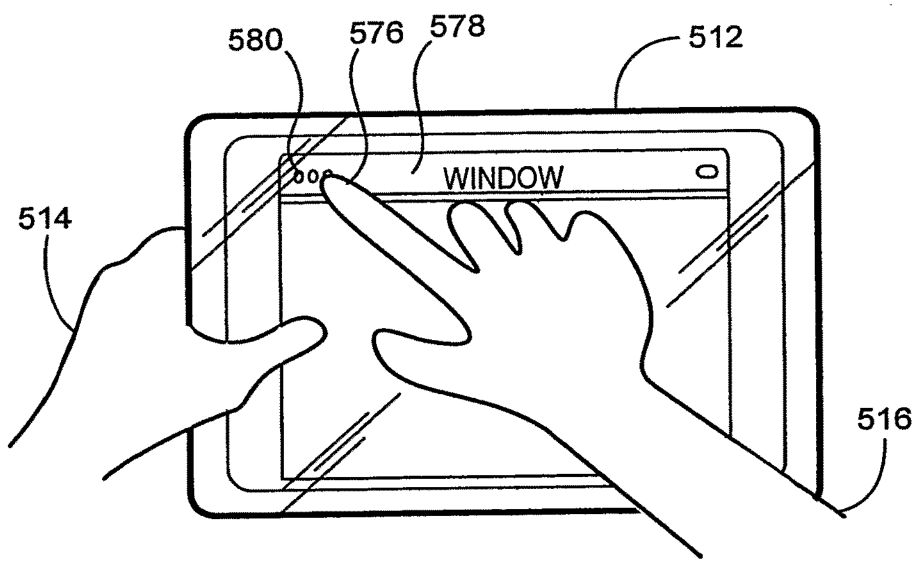


FIG. 19A

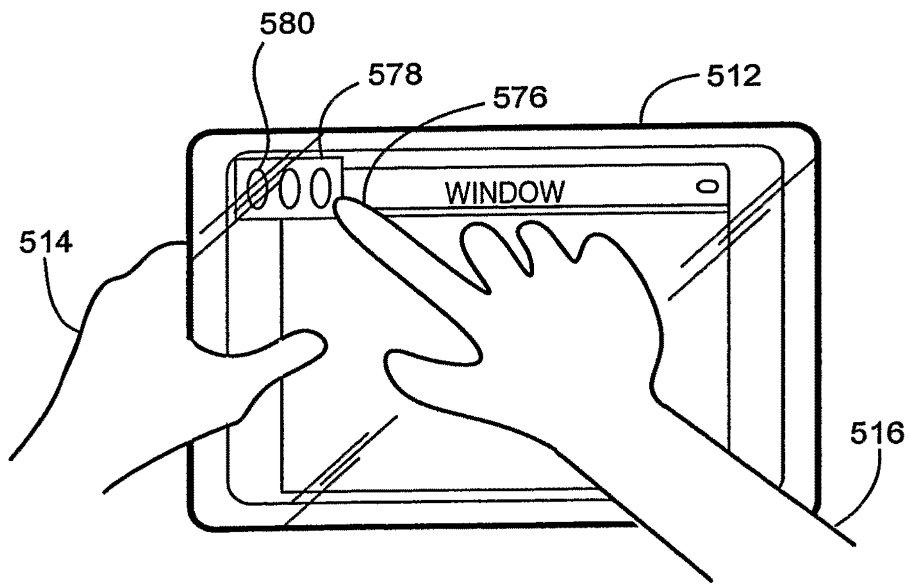


FIG. 19B

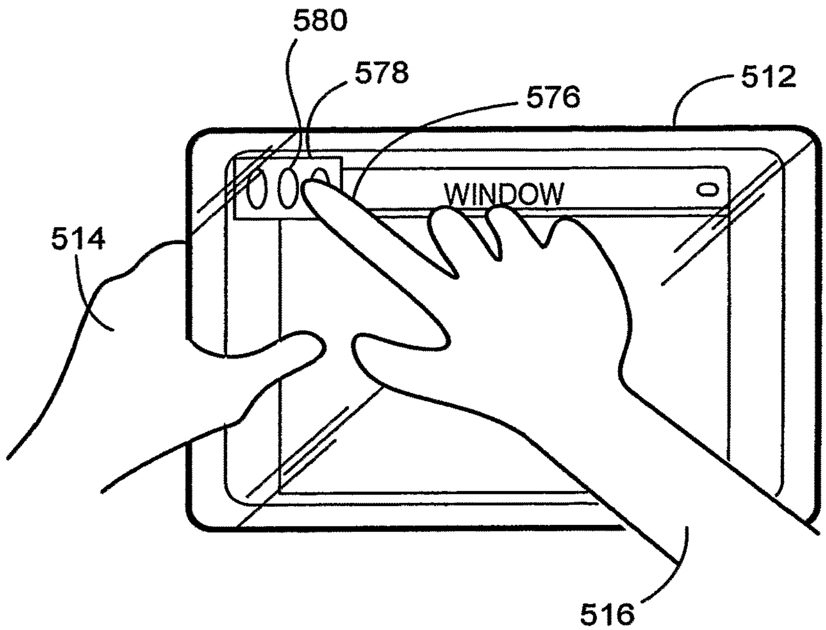


FIG. 19C

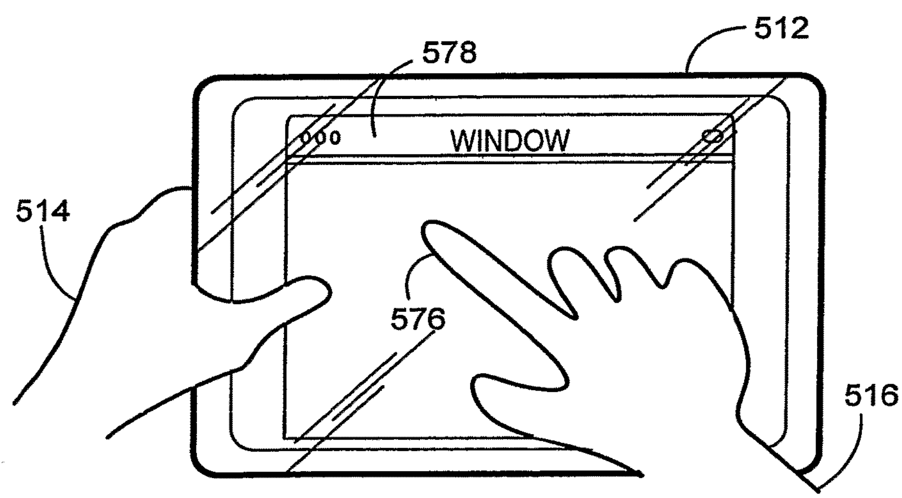


FIG. 19D

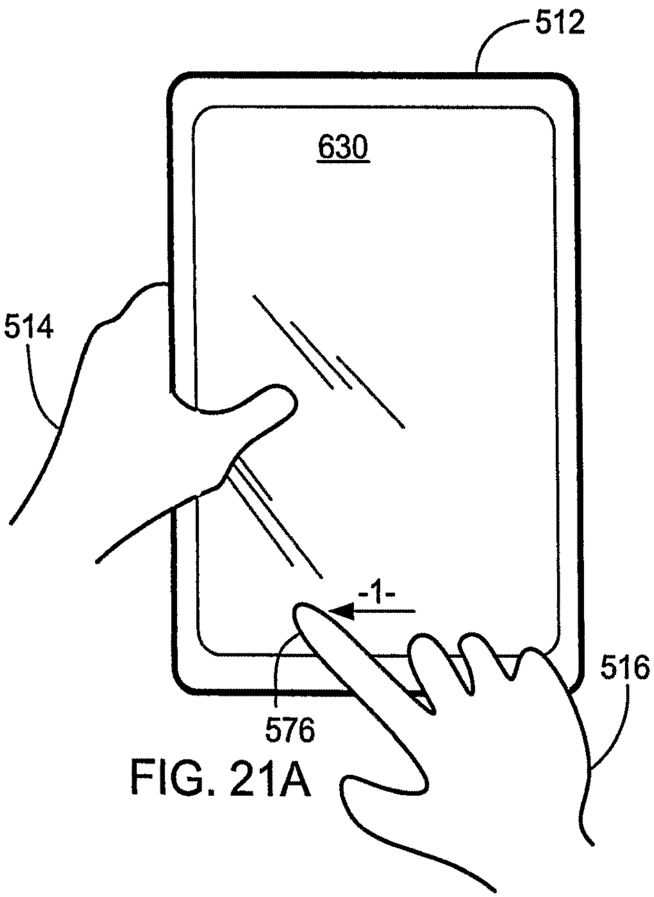


FIG. 21A

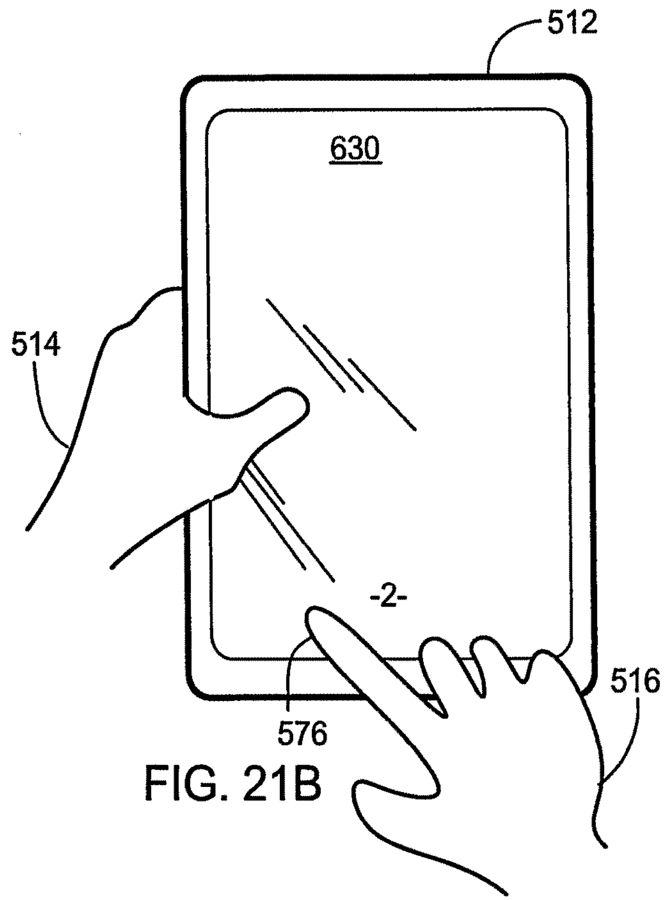


FIG. 21B

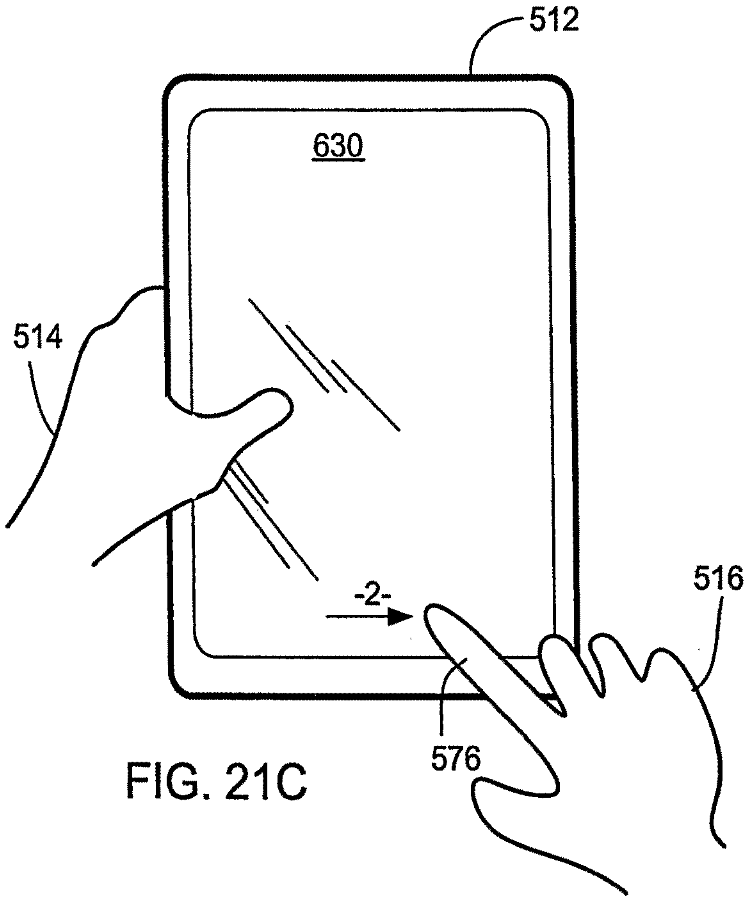


FIG. 21C

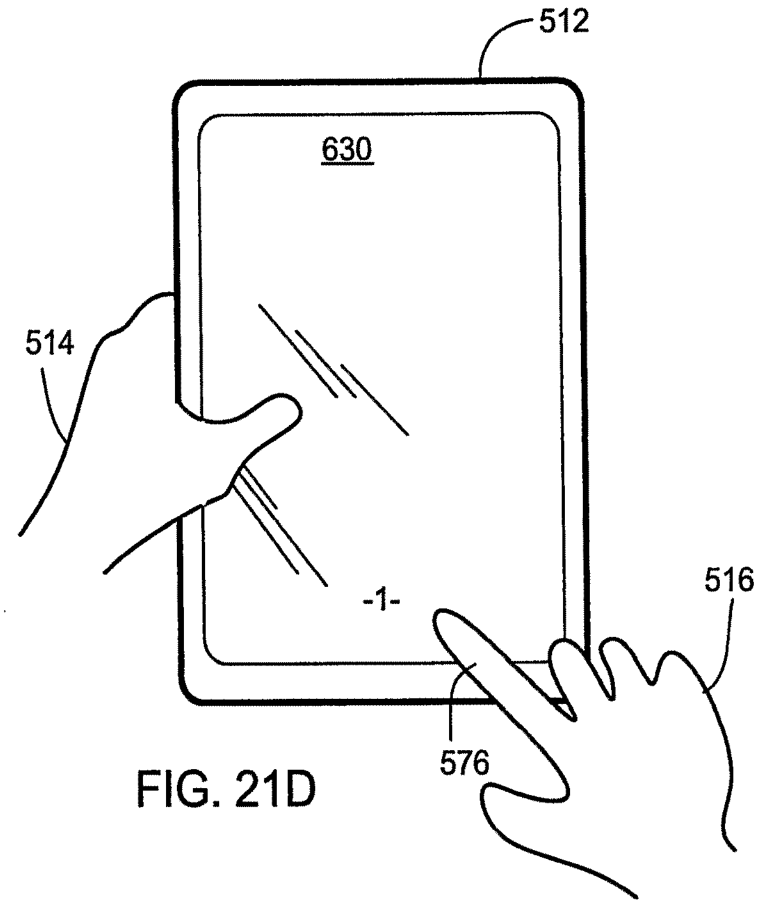


FIG. 21D

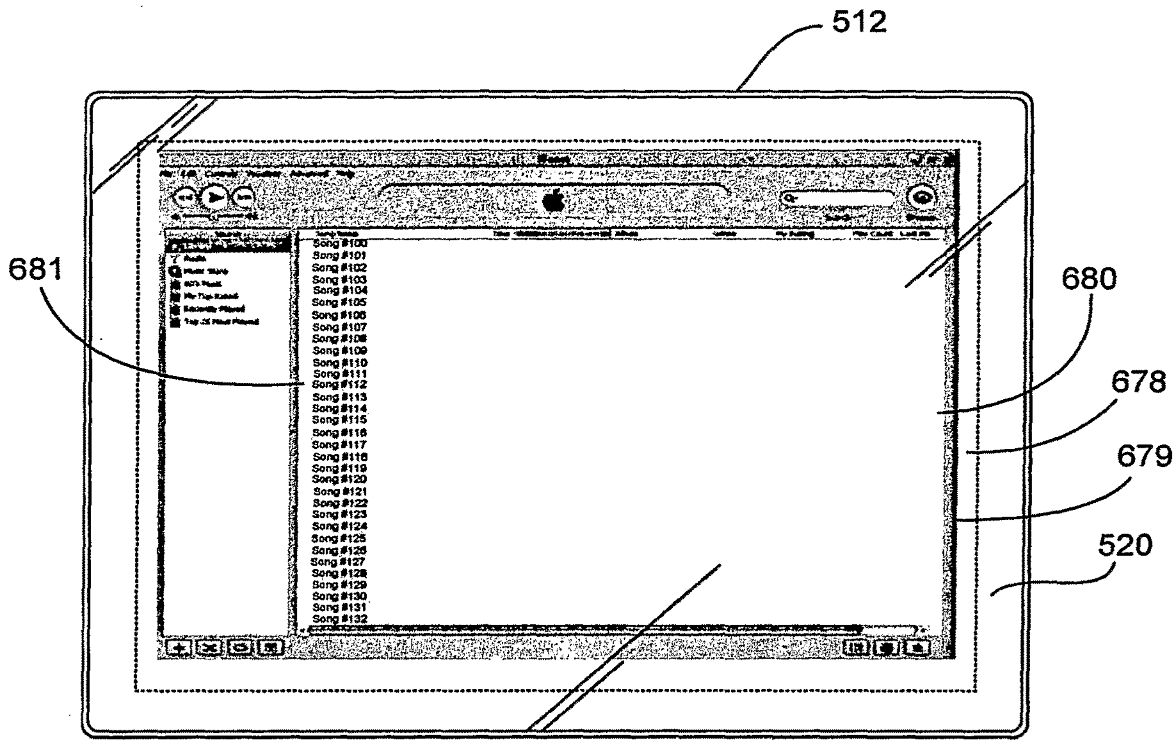


FIG. 23A

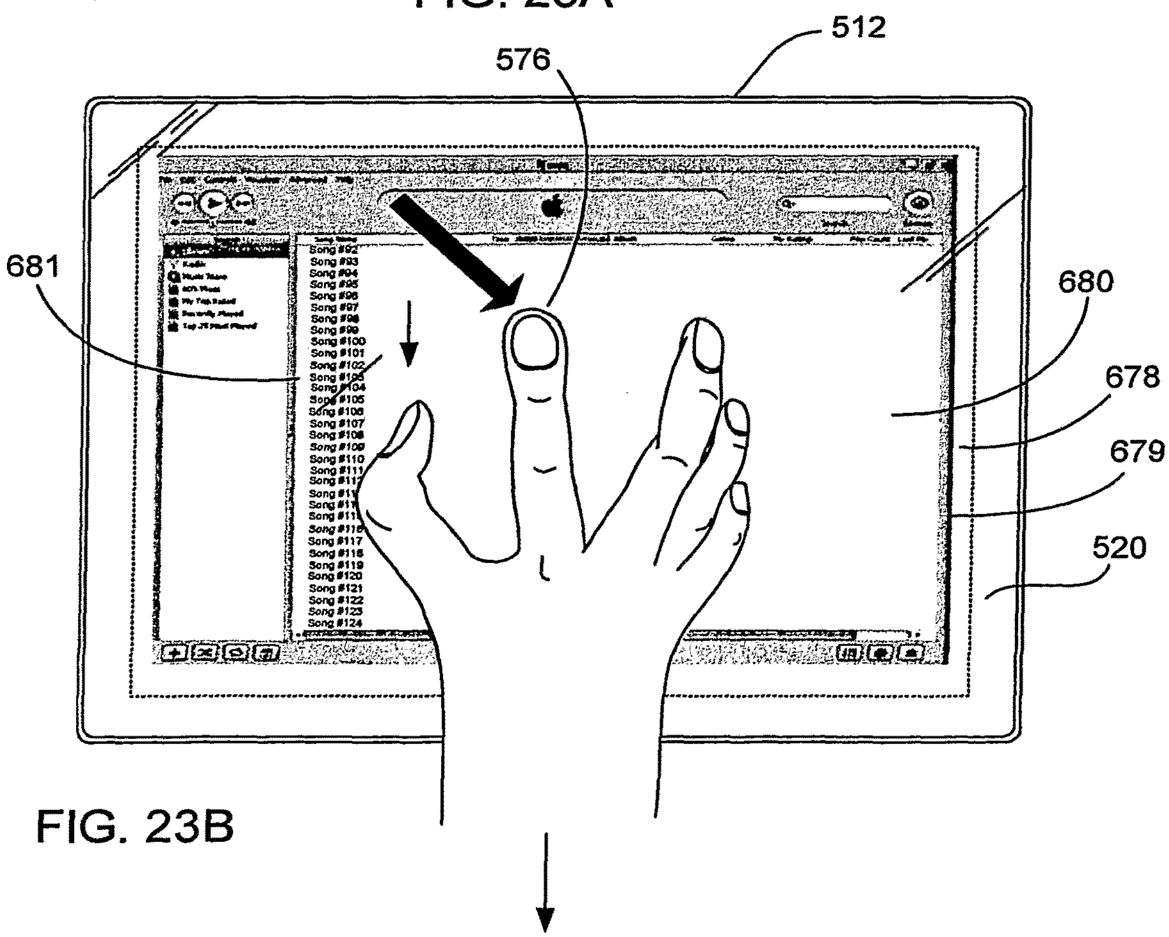


FIG. 23B

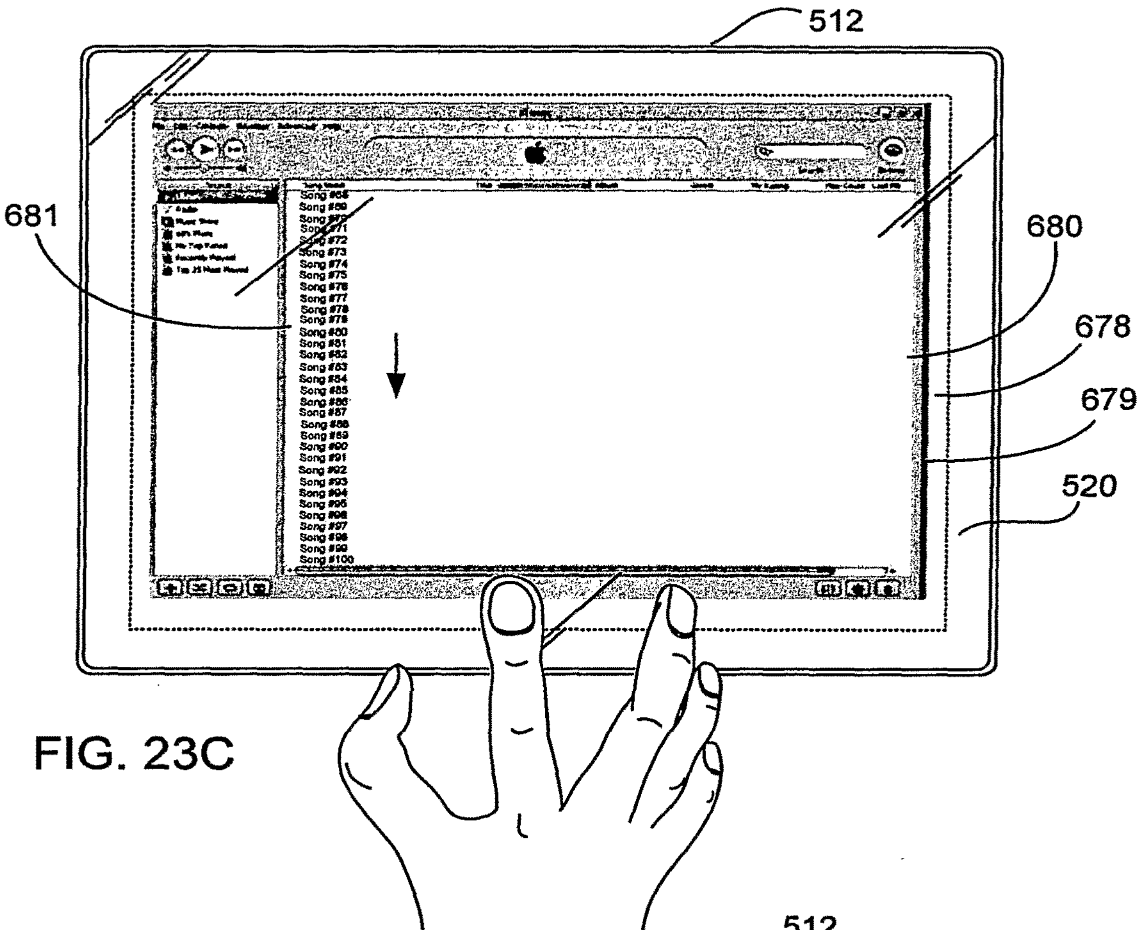


FIG. 23C

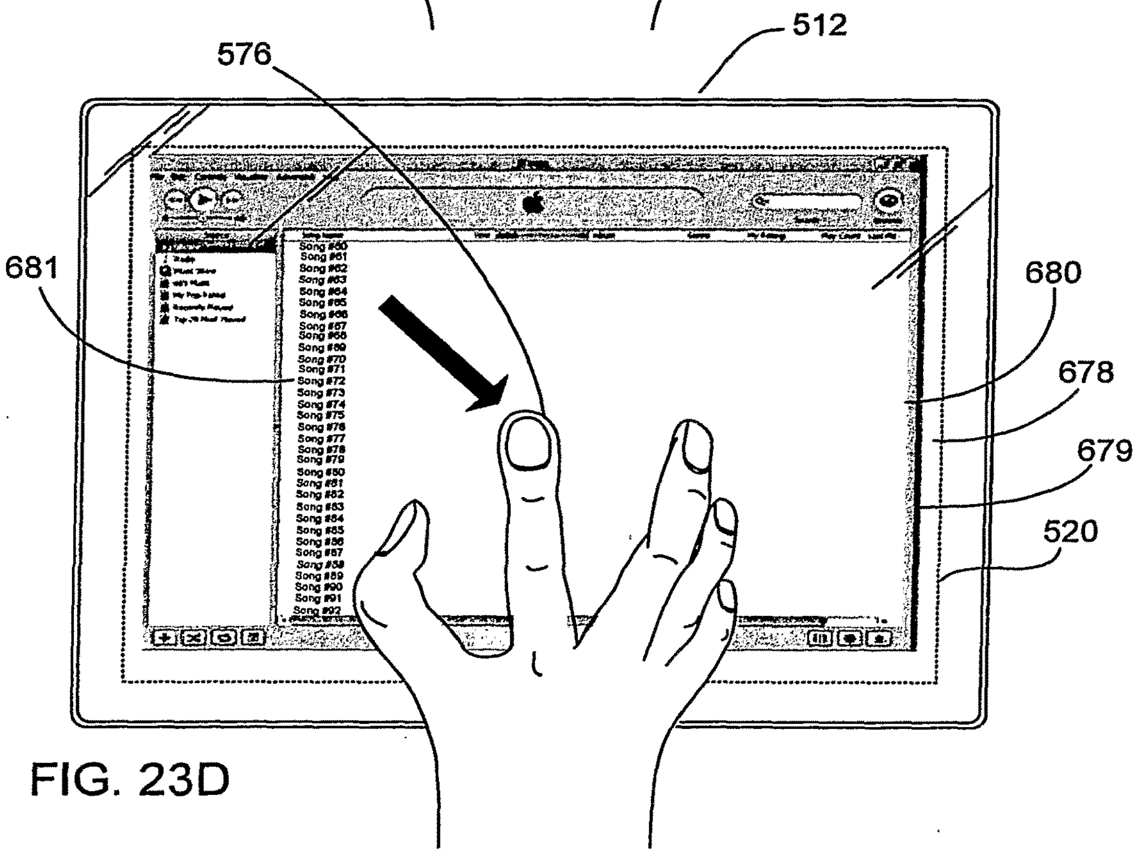


FIG. 23D

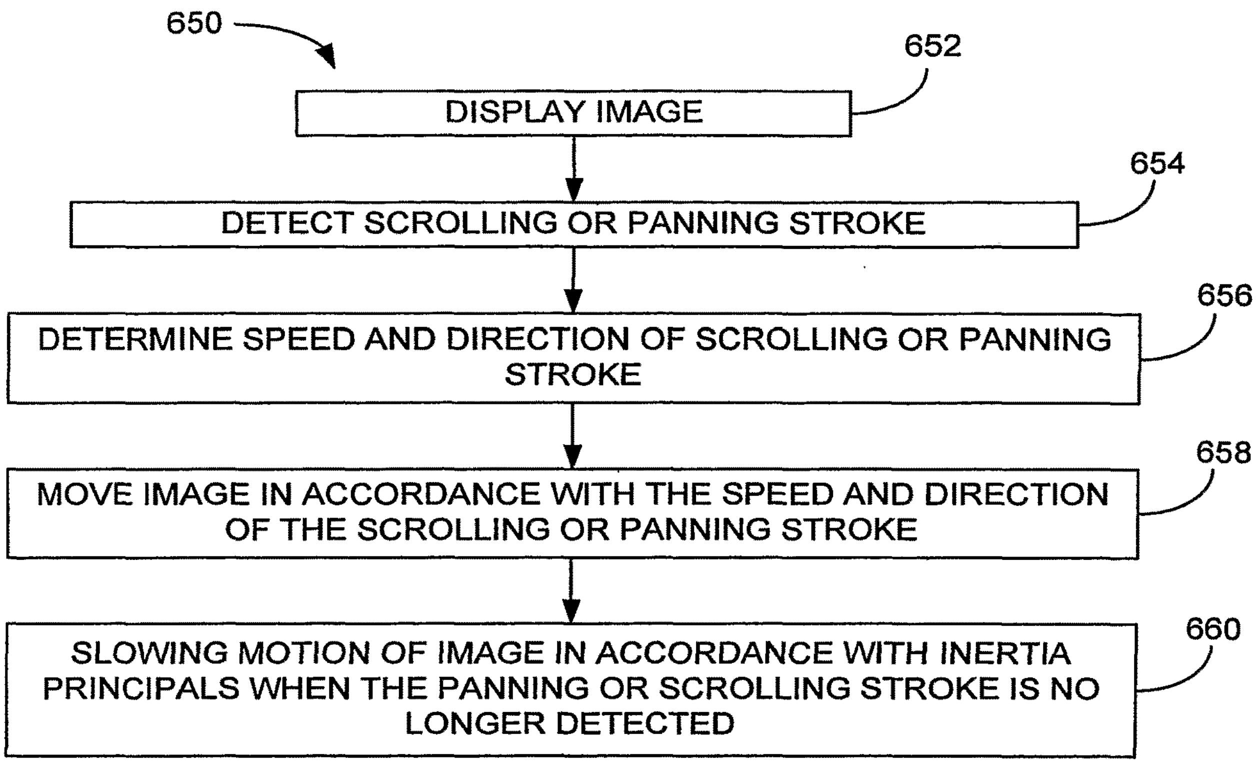

650
652
DISPLAY IMAGE
654
DETECT SCROLLING OR PANNING STROKE
656
DETERMINE SPEED AND DIRECTION OF SCROLLING OR PANNING STROKE
658
MOVE IMAGE IN ACCORDANCE WITH THE SPEED AND DIRECTION OF THE SCROLLING OR PANNING STROKE
660
SLOWING MOTION OF IMAGE IN ACCORDANCE WITH INERTIA PRINCIPALS WHEN THE PANNING OR SCROLLING STROKE IS NO LONGER DETECTED

FIG. 22

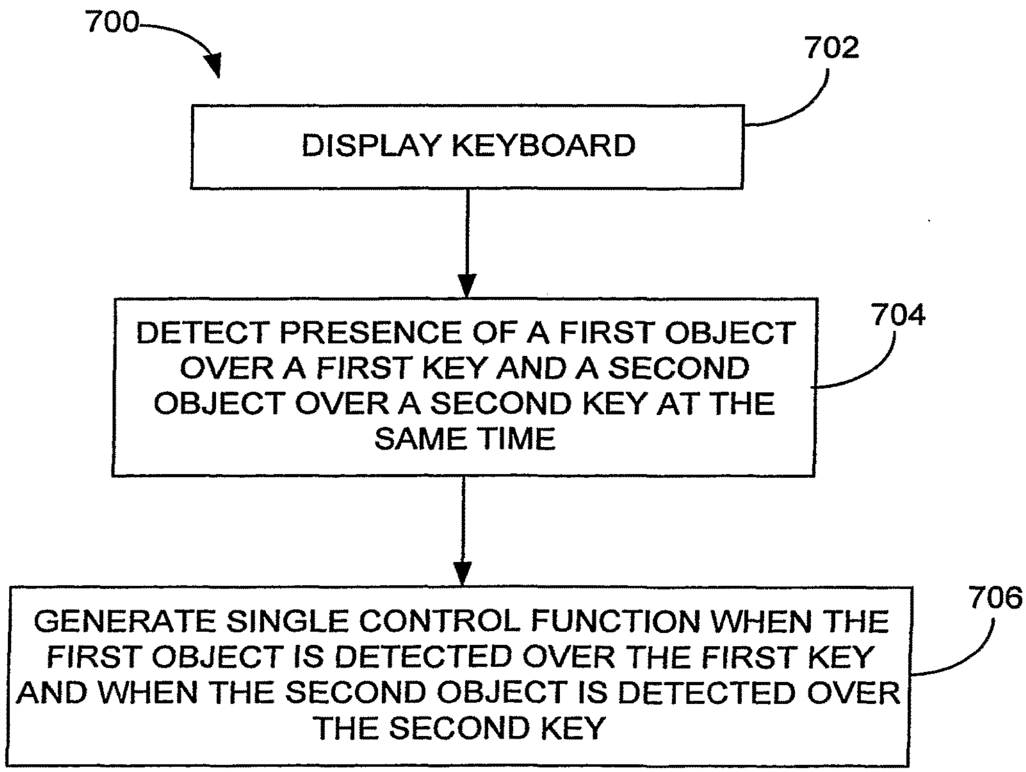

700
702
DISPLAY KEYBOARD
704
DETECT PRESENCE OF A FIRST OBJECT OVER A FIRST KEY AND A SECOND OBJECT OVER A SECOND KEY AT THE SAME TIME
706
GENERATE SINGLE CONTROL FUNCTION WHEN THE FIRST OBJECT IS DETECTED OVER THE FIRST KEY AND WHEN THE SECOND OBJECT IS DETECTED OVER THE SECOND KEY

FIG. 24

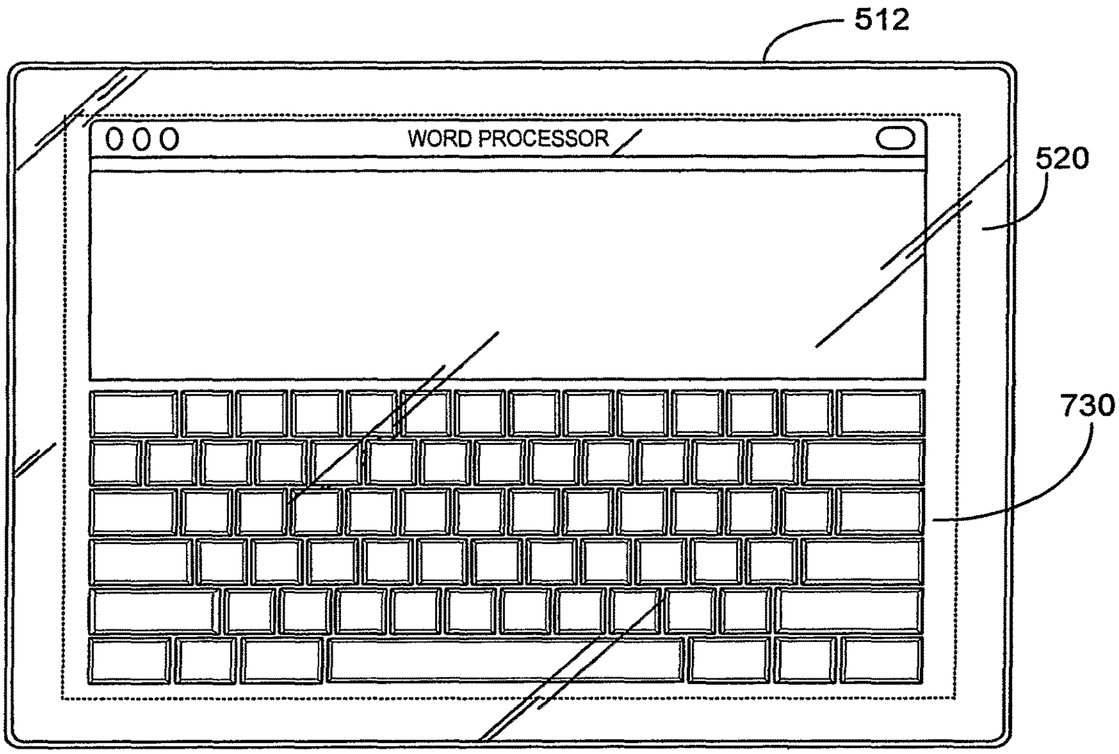


FIG. 25A

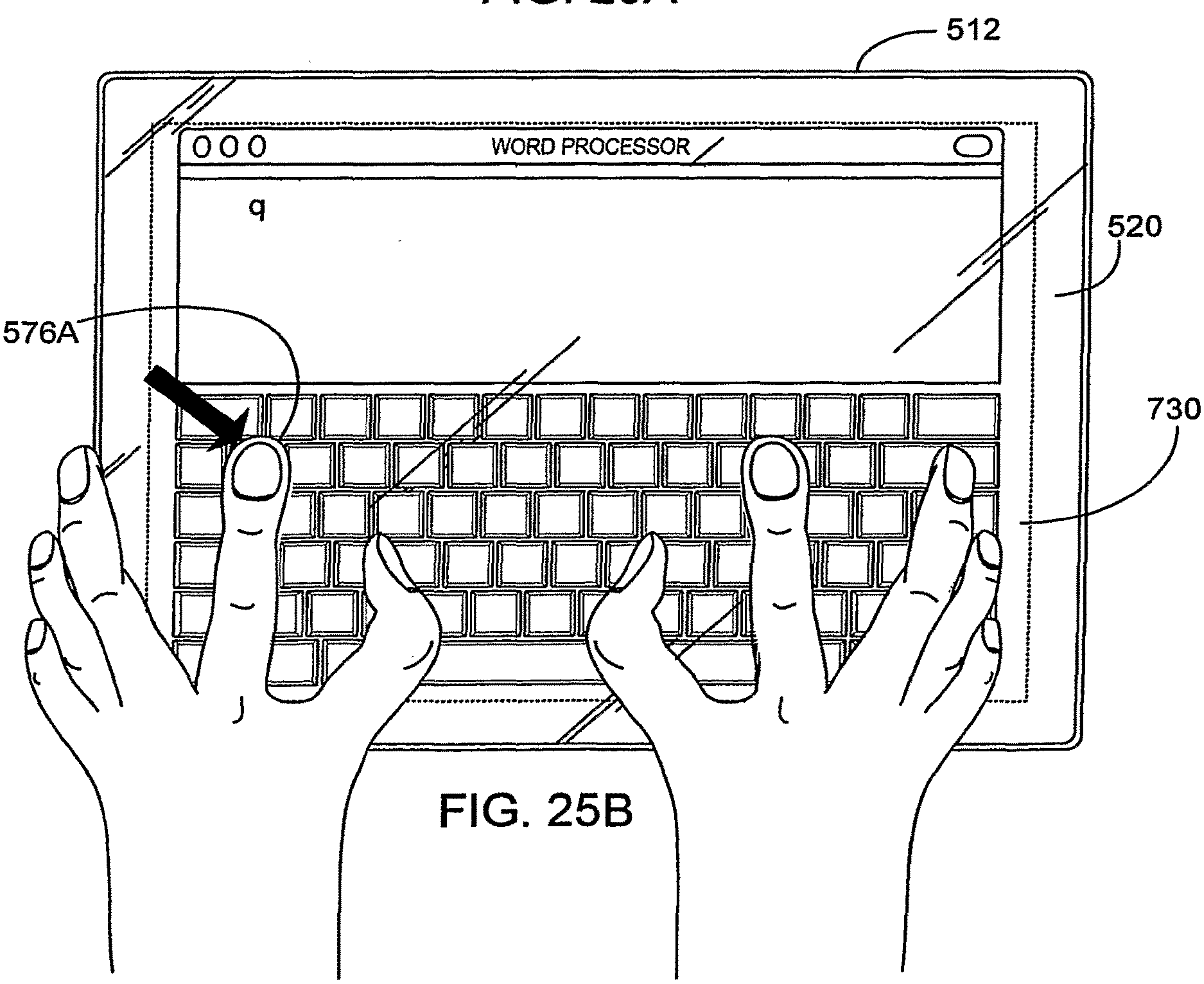


FIG. 25B

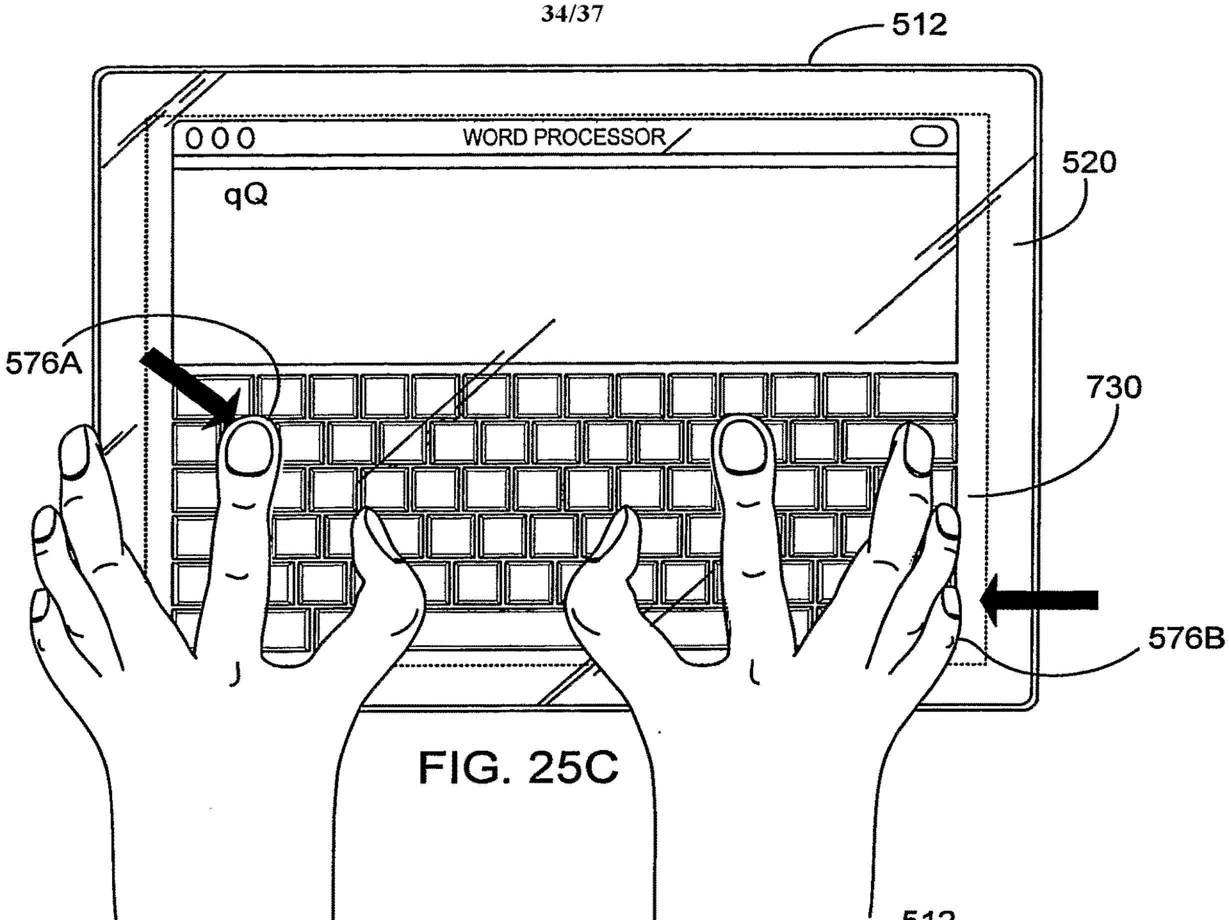


FIG. 25C

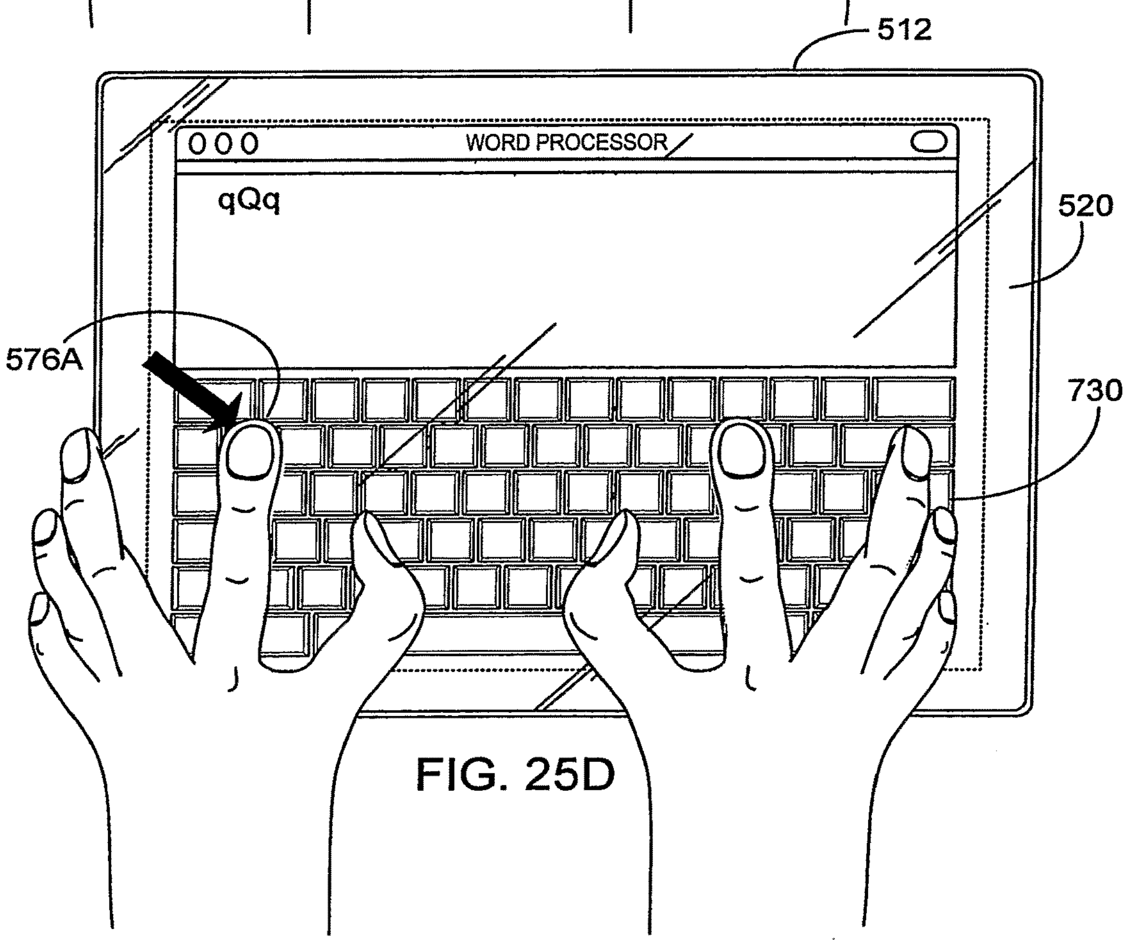


FIG. 25D

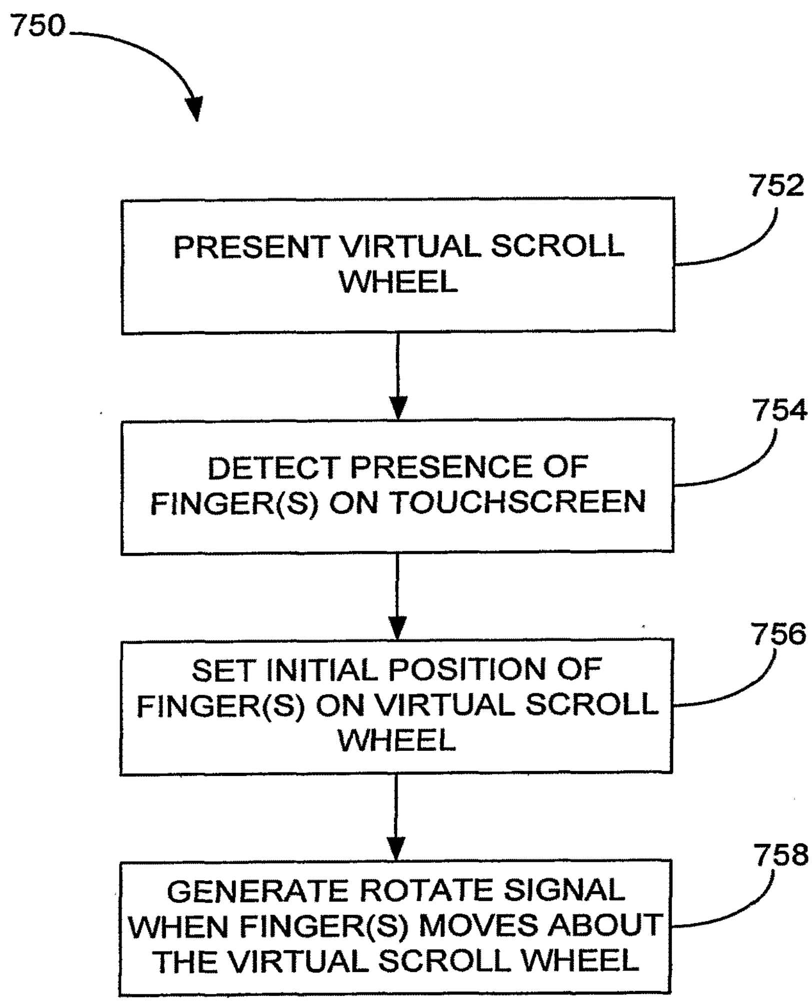

750
PRESENT VIRTUAL SCROLL WHEEL
752
DETECT PRESENCE OF FINGER(S) ON TOUCHSCREEN
754
SET INITIAL POSITION OF FINGER(S) ON VIRTUAL SCROLL WHEEL
756
GENERATE ROTATE SIGNAL WHEN FINGER(S) MOVES ABOUT THE VIRTUAL SCROLL WHEEL
758

FIG. 26

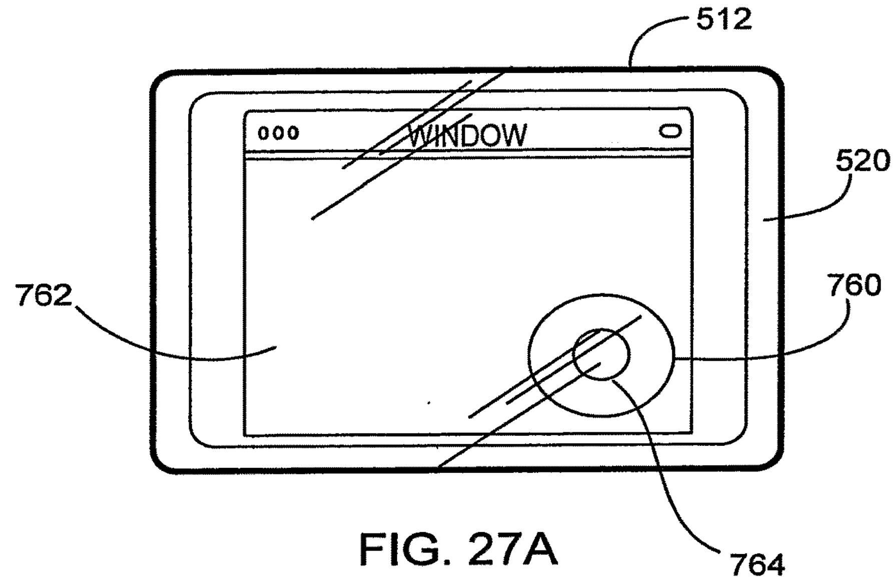


FIG. 27A

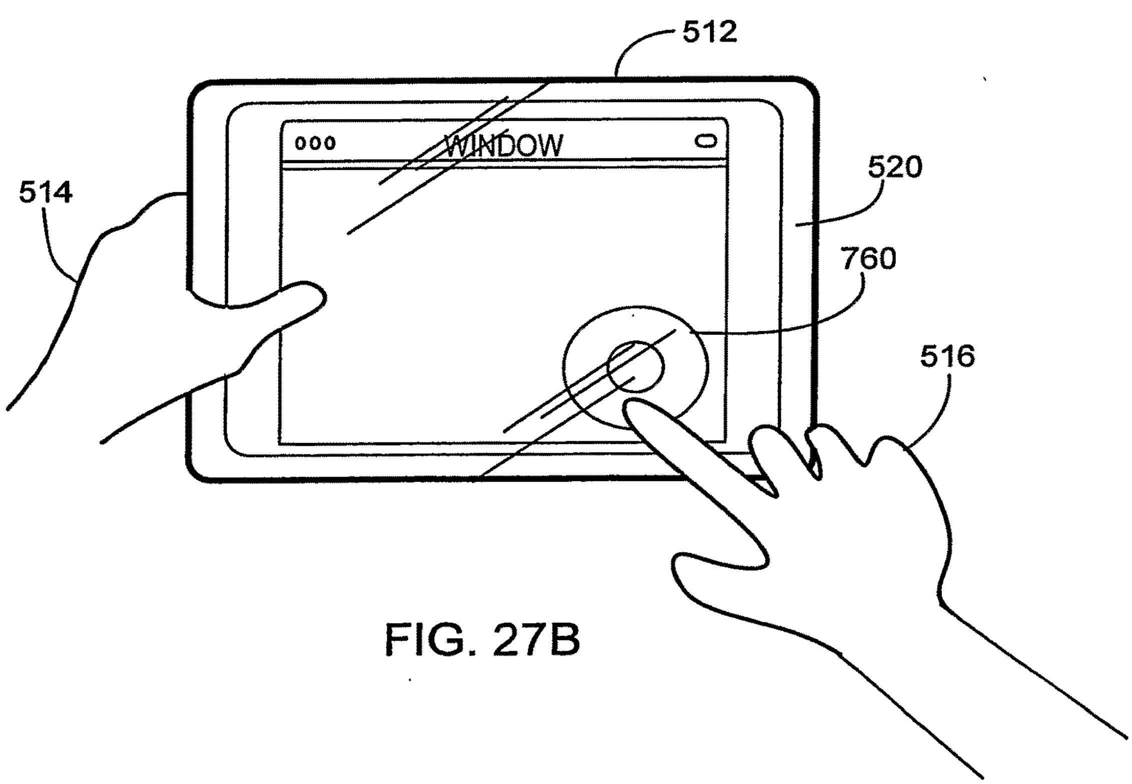


FIG. 27B

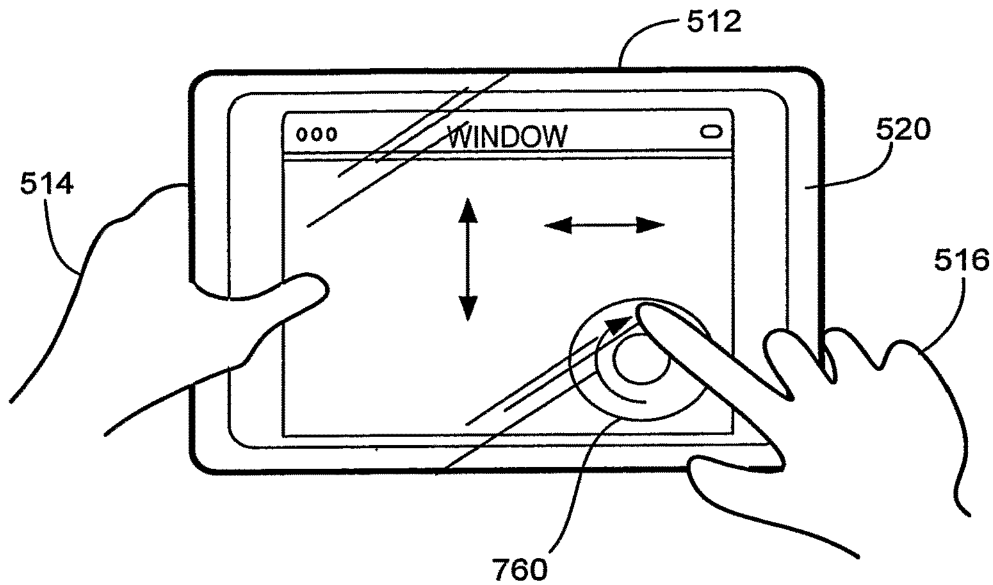


FIG. 27C

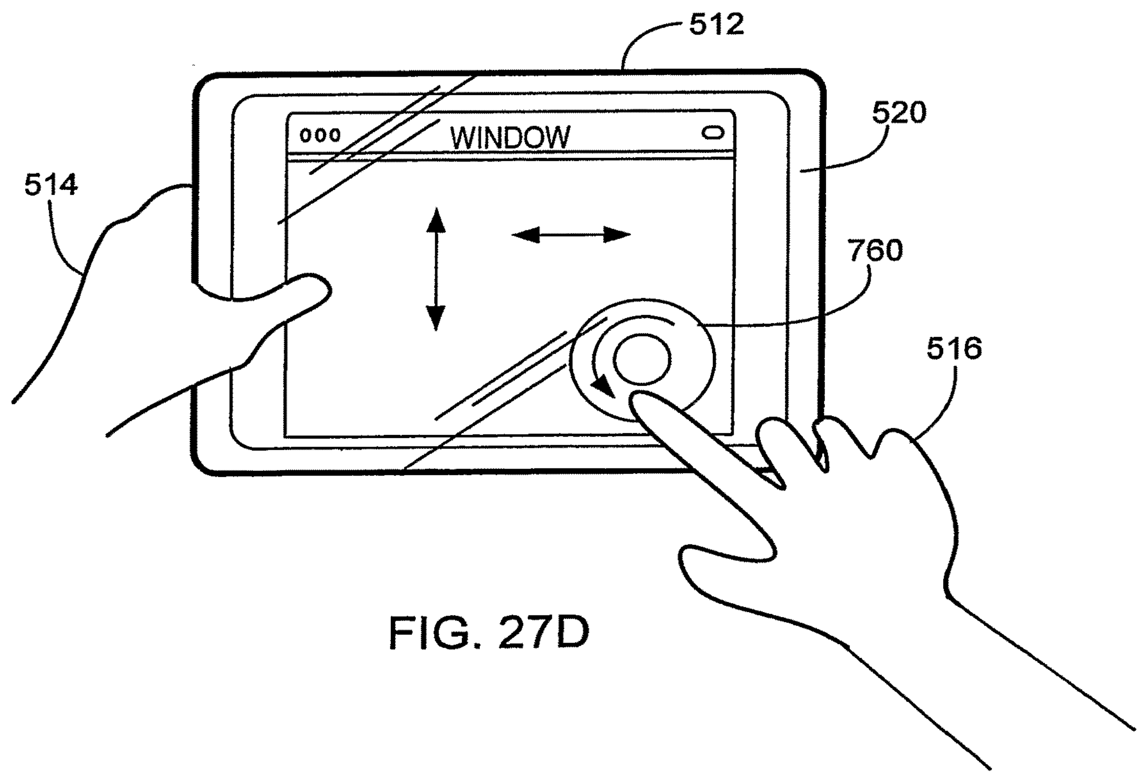


FIG. 27D

IBM® Technical Disclosure Bulletin Vol. 36 No. 04 April 1993

Scroll Control Box

In a windowing software environment, especially graphical interfaces, screen space and screen clutter are continual problems for both the user and the program designer. The need for diverse scrolling operations is common. Diversity includes scroll direction, speed, distance, "other" data chunks, flexibility in operation while visually scanning, etc. In windowing environments, these needs are potentially common for every scrollable window, and even for multiple scrollable objects within a window. Today's graphical or non-graphical scroll operators demand extensive screen space. For example:

- OS/2* scroll bars require about 8% of the useable space of each window.
- Other scroll mechanisms, i.e., PF Keys, Icon Arrows, Screen Push buttons, in the industry are:
  - inconsistently applied
  - are part of cluttered interfaces
  - require learning, relearning
  - require much screen space because packaging of scroll functions are inefficient
- much graphical interface complexity of function in addition to scrolling.
- current scrolling is inefficient and insufficient.
- new scrolling function difficult to add gracefully.

The Scroll Control Box (SCB) is a graphical object shown on a display screen which, singly, is used to do all scrolling operations. The SCB gives the user the capability, with one action (drag), to indicate what is to be scrolled. Also, with single action (point/slide), a user can do all of the following:

- Select one of 10 possible directions of scrolling on the x-y axis 8 compass points (N, NE, E...) and two directions on the y axis (depth), activated by pressure.
- Control the scroll speed or how large the scroll increments.
- Change direction, speed, and distance during this single action.
- Scroll any objects on screen by moving and placement of the SCB.

The user operates the SCB using a touch instrument (finger, stylus), or mouse or mouse-like device, a keyboard or other cursor operating device in much the same way icons and other graphical objects are operated on today. SCB dimensions, i.e., of speed, distance, etc., are customizable.

Results:

- Small footprint of the one SCB compared with other scroll methods, e.g., one or several scroll bars

- Efficient, all direction and dimensional controls in one place with two, three, or more controls for each of 10 directions
- Simple to learn and remember
- Can be placed by user
- Single control for any scrollable area on a display screen
- Flexible design can use combinations that are useful and most usable, e.g., using pressure for speed and sliding for going from one data chunk to another
- Efficient, a "point-to-select" technique

FIG. 1

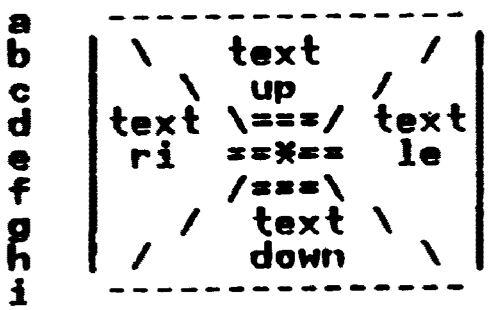
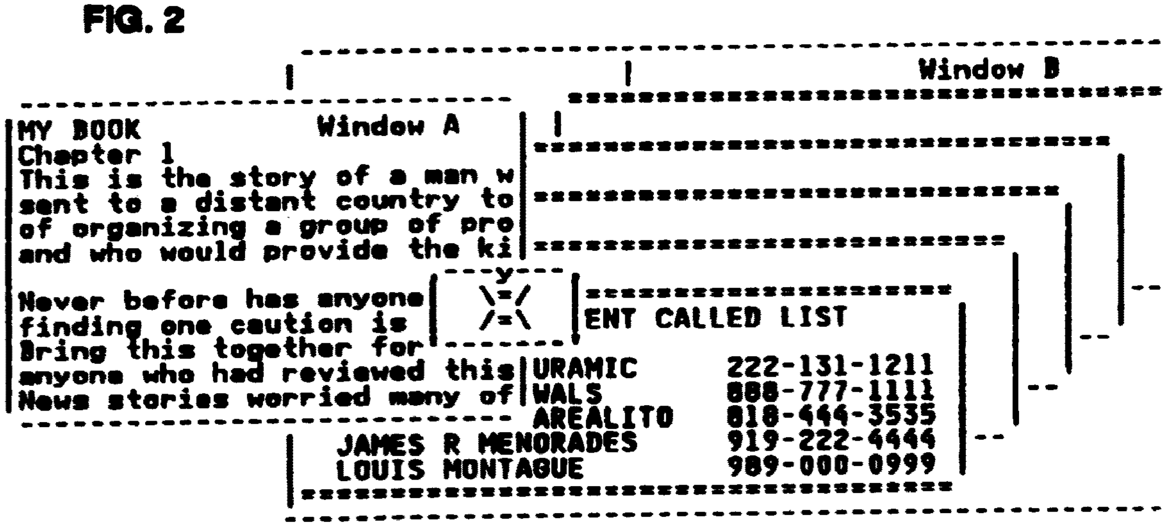


FIG. 2



The quadrant or area in Fig. 1 containing the term "text up" defines where mouse, touch, cursor, or some similar activation method causes scrollable text (or other scrollable objects) to move upward. This will bring into view what is below the visible window, and take from view that within the visible window, in successive timed steps.

Activation of the SCB is on touch down (finger or stylus), mouse point and button down, keyboard caused equivalent action, or other device equivalent, e.g., rollerball, etc. Activation causes either the scrolling movement (after slight delay) of screen content or the SCB-drag state (allows the SCB to be moved around on the screen). This Activation in the center area, as illustrated in Fig. 1 by equal signs (=), allows movement of the SCB. Any other area causes scrolling movement.

"Text" scrolling is illustrated in the following example. Activation, e.g., touch down, near the top perimeter of the box causes scrolling in the slowest of timed steps. Activation closest to the center of the SC box causes the fastest of timed steps. Additionally, once activated, one can slide down to speed up, or slide up to slow the scroll rate. In addition to timing of the scroll, increments can be increased; e.g., from line-by-line, to page-by-page, by control of the application customization programming.

There are many possible choices of scroll increments, line-by-line, full page, full window, window minus one, chapter, etc. Several of these can be operated by the SC box, like line-by-line, window-by-window, and chapter-by-chapter or other increment that has been set up.

Activating on the center area, indicated by equal signs (=) in Fig. 1 initially causes the SCB to become moveable, thereby following any dragging action by the user to position the SCB. The SCB being positioned within any scrollable zone in any window or area defined as having scrollable text or objects within, becomes the scroll control for that zone. Areas can also be excluded from control of the SCB by the application developer.

When the SCB is not within a scrollable zone, or scrolling has not been enabled by the application program, the SCB changes its visual appearance; e.g., it is reduced in size and can be de-emphasized color.

When at the scroll limit (at bottom of text), the "text up" area of the SCB becomes solid in color, obscuring the "text up" and becoming inoperative for that direction.

The terms used within the SCB are for illustration only. They are modifiable and may be symbolic, text, or graphical representations. The illustration in Fig. 1 is only an approximation and art work is needed.

After initial activation within a scroll direction area (up), sliding to an adjacent direction area (left), causes diagonal scroll at the transition point in the averaged direction of the two. This facility can be turned off. The SCB is only present when there is a scrollable areas on screen.

Fig. 2 shows an example user display screen with two windows opened -- window A and window B. Window A overlaps window B and some of window B contents are visible. Window A shows 12 lines of a document that is 150 pages long and having 10 chapters. Window B shows six of a deck of 50 cards, each offset to the right and in back of the previous card. The front card in window B shows seven lines of a list of 36 lines. Window A, then has one scrollable area, and window B has two scrollable area (deck of cards, and first card content).

An example SCB is located in about the center of Fig. 2. It is adjacent to the right of the phrase "Never before has anyone". The SCB is shown in a position where it will, when operated, scroll the text content of window A, it having the majority of its box area within window A.

By touch down in the center drag area, the SCB can be dragged to the right one inch, the SCB would then be in a position in window B to activate scrolling of the list on the front card.

By touching down within the center drag area, the SCB can be dragged directly up to a position still within window B, where most of it was above the front card, the scrolling action would be directed to scrolling the deck of cards.

1. Touching on the SCB at point "Y" (12 o'clock), causes text in window A to move upward at a relatively slow rate.
2. Sliding towards the center of the SCB increases the scroll rate.
3. Sliding away from the center of the SCB, but staying within the SCB, stops scrolling.
4. Sliding outside the SCB stops scrolling.
5. Sliding across a diagonal border within the SCB; e.g., from a position at 12 o'clock to one at 3 o'clock, changes the direction of the text scrolling direction from upward to leftward.
6. Sliding across a diagonal border without continuing into another quadrant (as in previous step) changes scrolling direction so as to make text already in the window to move out of view to the southwest, bring next text into view from the northeast.
7. There is a slight time delay in activating any scrolling movement to prevent unintentional diagonal movement when changing between direction of 90°, or when accidentally touching the diagonal transition zone, or when touching any SCB area unintentionally.

The act of sliding within the SCB direction quadrant to produce faster or slower scrolling can be replaced with the act of increasing pressure, either on a touch display screen directly, or from a stylus, mouse, or other device. Benefits of this can include reducing the area required by the SCB, allowing more change graduations, easier or more efficient to operate.

More Controls: A depth dimension of scrolling can be added with "pressure" selection technology. Keeping in tact with eight directional capabilities of the SCB described above, "pressure" can add depth directions. Following are examples:

- Limits of the SCB - Touching the SCB "Text Up" area scrolls line-by-line. It can be made to scroll faster as the touch is dragged toward the SCB center. Increments closer to the SCB center can also be programmed to scroll chapters or sections, or articles, or documents, but there is a limit to how many delineations can be put into a small SCB quadrant. So, with pressure, dimension function can reduce demands on the quadrant space.
- Working with a document - When an SCB is associated with a document, it may have SCB increments set to scroll text lines, then paragraphs, then pages as one slide towards its center. With depth direction (pressure increments), one may, while scrolling at page 7, press harder to jump to the next chapter, and harder yet to go to the end. Direction scrolling is momentarily suspended when a pressure threshold is detected.

- Working with an appointment book - The SCB is dragged from a document to appointments where SCB increments are set to scroll line-by-line, then sliding towards the SCB center, it scrolls faster line-by-line.then appointment-to- appointment, then day-by-day. With depth direction (pressure increments), one may, while at line 2 of 3:30 a.m. of Tuesday, jump to same time and day week-by-week. More pressure might cause month-by-month, and so on.

* Trademark of IBM Corp.

Patterns for Effective Interaction Design

Designing Interfaces

O'REILLY®

Jenifer Tidwell

Designing Interfaces
by Jenifer Tidwell

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*safari.oreilly.com*). For more information, contact our corporate/institutional sales department: (800) 998-9938 or *corporate@oreilly.com*.

Editors: Andrew Odewahn and Mary O'Brien
Production Editor: Genevieve d'Entremont
Cover Designer: Mike Kohnke
Interior Designer: NOON

Printing History:
November 2005: First Edition.

This book uses RepKover, a durable and flexible lay-flat binding.

ISBN: 0-596-00803-1

[C] [4/06]

85 magnetism

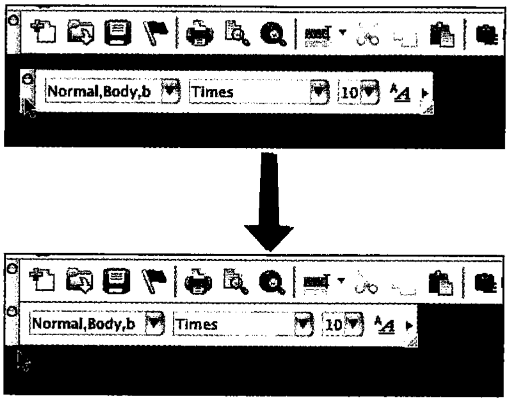


FIGURE 8-14 / A Word toolbar "snapped" into place

what

Make the objects "magnetic" to the things a user positions them against. When the user drags an object very near one of these things, it should stick.

use when

The user needs to position objects very precisely, such as next to other objects or against **Guides**. This often happens in graphic editors, of course, but it's also common in window managers and desktop frameworks, in which a user needs to move windows and palettes, for example.

why

Magnetism helps compensate for users' lack of perfect dexterity with a mouse. If the user really does want the moved object to end up right against another one—and you need to be sure about this—then the computer can help by doing it for her.

What magnetism does is make the mouse "target zone" effectively bigger than it really is. If the user tries to put one thing precisely next to another thing, the target zone is one pixel wide; beyond that, they either overlap or don't touch, and she has to keep trying to position it exactly right. But a one-pixel target is awfully tiny. If magnetism snaps the object into place when the user gets within, say, four pixels of the edge, then the target zone is two times four (i.e., eight) pixels wide. Much easier!

And much faster, too. This device also helps users who do have the dexterity necessary to place objects within one-pixel target zones. Magnetism saves them the time and effort they'd otherwise need to do that, and it makes the application feel more responsive and helpful in general.[2]

how

When the user drags an object close to another object's edge, make it snap to the other object. Likewise, when it's dragged away, keep it there for a few pixels, and then let it move away.

2 Martijn van Welie first described this pattern in his online patterns catalog: *http://www.welie.com/patterns/gui/magnetism.html*.

Objects that can be "magnetic" might include:

- Objects of the same type as the one moved, such as windows to windows or shapes to shapes.
- Canvas edges, margins, and screen edges.
- **Guides** and grid lines—devices that exist strictly for the purpose of aligning objects with one another.
- Easily detectable "hard edges" in picture layers, as seen in Photoshop.

examples

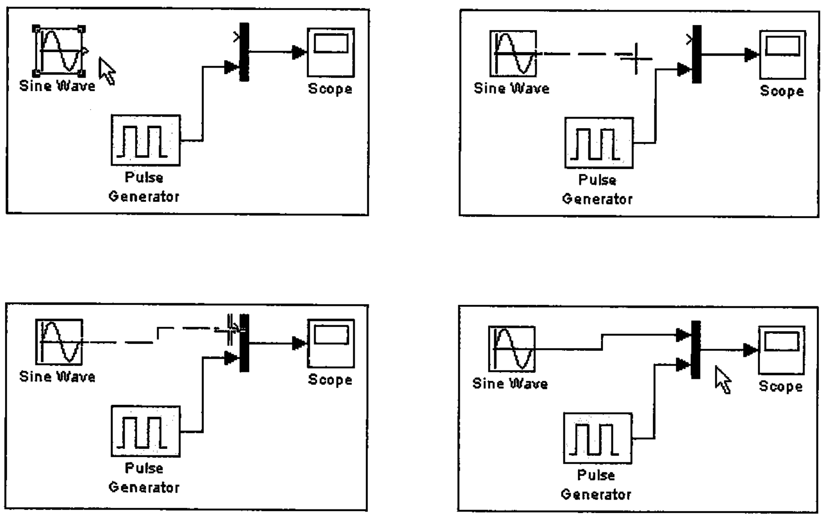


FIGURE 8-15 / Diagram builders and visual programming environments sometimes use magnetism to help the user link together a diagram. Here, you can use the Simulink application to put together a miniature signal-generator simulation. The user needs to connect the output port of the Sine Wave source to an input port of a multiplexer, shown as a dark vertical bar.

Without magnetism, this task would require the user to position the mouse on a very tiny target, click down on the mouse button, and then drag and release the connection to another very tiny target. However, the source output port actually is bigger than the few pixels shown. And as the user drags the connection—illustrated by a dotted line—toward the multiplexer's input port, the connection "snaps" into the input port as the mouse approaches within ten pixels or so. (Simulink even puts corners into the connection automatically.) In the last frame, the user has released the mouse button, finishing the connection.

PATENT COOPERATION TREATY

From the INTERNATIONAL SEARCHING AUTHORITY

PCT

To:

MORGAN LEWIS & BOCKIUS LLP
Attn. Williams, Gary S.
2 Palo Alto Square
3000 El Camino Real, Suite 700
Palo Alto, CA 94306
ETATS-UNIS D'AMERIQUE

INVITATION TO PAY ADDITIONAL FEES AND, WHERE APPLICABLE, PROTEST FEE

(PCT Article 17(3)(a) and Rule 40.1 and 40.2(e))

by fax and post

	Date of mailing *(day/month/year)* 18/07/2008
Applicant's or agent's file reference 63266-5054WO	**PAYMENT DUE** within **ONE MONTH** from the above date of mailing
International application No. PCT/US2008/050292	International filing date *(day/month/year)* 04/01/2008
Applicant APPLE INC.	

1. This International Searching Authority

(i) considers that there are 3 *(number of)* inventions claimed in the international application covered by the claims indicated on an extra sheet:

(ii) therefore considers that **the international application does not comply with the requirements of unity of invention** (Rules 13.1, 13.2 and 13.3) for the reasons indicated on an extra sheet:

(iii) [X] has carried out a partial international search (see Annex) [ ] will establish the international search report on those parts of the international application which relate to the invention first mentioned in claims Nos.:

see extra sheet

(iv) will establish the international search report on the other parts of the international application only if, and to the extent to which, additional fees are paid.

2. Consequently, the applicant is hereby **invited to pay**, within the time limit indicated above, the amount indicated below:

EUR 1.700,00	x	2	=	EUR 3.400
Fee per additional invention		number of additional inventions		currency/total amount of additional fees

3. The applicant is informed that, according to Rule 40.2(c), **the payment of any additional fee may be made under protest**, i.e., a reasoned statement to the effect that the international application complies with the requirement of unity of invention or that the amount of the required additional fee is excessive, where applicable, subject to the payment of a protest fee.
Where the applicant pays additional fees under protest, the applicant is hereby invited, within the time limit indicated above, to pay a protest fee (Rule 40.2(e)) in the amount of EUR 750,00 *(currency/amount)*

Where the applicant has not, within the time limit indicated above, paid the required protest fee, the protest will be considered not to have been made and the International Searching Authority will so declare.

4. [ ] Claim(s) Nos. ______ have been found to be unsearchable under Article 17(2)(b) because of defects under Article 17(2)(a) and therefore have not been included with any invention.

Name and mailing address of the International Searching Authority	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL-2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Matthew Davis

Form PCT/ISA/206 (April 2005)

INVITATION TO PAY ADDITIONAL FEES

International application No.
PCT/US2008/050292

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-42

   Translating an electronic document or list of items in response to gestures on a touch screen display.

   ---

2. claims: 43-47

   Executing rotation commands in response to a multifinger twisting gesture on a touch screen display.

   ---

3. claims: 48-73

   Scaling of an electronic document in response to a gesture on a touch screen display.

The use of gestures on a touch screen display for translation, rotation and scaling is known, e.g., from document WO2006/020305 (see, e.g., figures 13A-13D, 15A-15C, 11A-11H respectively and the associated text of the description).

The remaining features (i.e., special technical features in the sense of Rule 13.2 PCT) of the independent claims, in particular claims 2, 43 and 63, relate to a) translating in a second direction for aligning a document's edge with an edge of a display region, b) executing a 90° or reverse rotation command, depending on the degree of rotation of the gesture and c) displaying a portion of an electronic document in a predefined magnification, i.e., they relate to different aspects, each peculiar to translation, rotation or scaling respectively.

In conclusion, the groups of claims are not linked by common or corresponding special technical features and define three different inventions not linked by a single general inventive concept.

The application, hence does not meet the requirements of unity of invention as defined in Rules 13.1 and 13.2 PCT.

Annex to Form PCT/ISA/206
COMMUNICATION RELATING TO THE RESULTS OF THE PARTIAL INTERNATIONAL SEARCH

International Application No
PCT/US2008/050292

1. The present communication is an Annex to the invitation to pay additional fees (Form PCT/ISA/206). It shows the results of the international search established on the parts of the international application which relate to the invention first mentioned in claims Nos.:
see 'Invitation to pay additional fees'
2. This communication is not the international search report which will be established according to Article 18 and Rule 43.
3. If the applicant does not pay any additional search fees, the information appearing in this communication will be considered as the result of the international search and will be included as such in the international search report.
4. If the applicant pays additional fees, the international search report will contain both the information appearing in this communication and the results of the international search on other parts of the international application for which such fees will have been paid.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2004/021676 A1 (CHEN HUNG-MING [TW] ET AL) 5 February 2004 (2004-02-05) the whole document	1-22
Y	WO 03/060622 A (KONINKL PHILIPS ELECTRONICS NV [NL]) 24 July 2003 (2003-07-24) the whole document	23-42
Y	TIDWELL ET AL: "Designing Interfaces" 20051101, 1 November 2005 (2005-11-01), XP002478404 Section "85. magnetism"	1-42
A	WO 2006/020305 A (APPLE COMPUTER [US]; HOTELLING STEVE [US]; STRICKON JOSHUA A [US]; HUP) 23 February 2006 (2006-02-23) page 24, line 1 - page 28, line 10 figures 10-15C	1-42
A	WO 02/01338 A (INTEL CORP [US]; MILLER ALYSON [US]; MERRILL CINDY [US]; LUNDELL JAMES) 3 January 2002 (2002-01-03) page 3, line 5 - page 4, line 12 figures 2-4	1-42

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
"&" document member of the same patent family

2

Patent Family Annex

Information on patent family members

International Application No
PCT/US2008/050292

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2004021676 A1	05-02-2004	TW 591488 B	11-06-2004
WO 03060622 A	24-07-2003	AU 2002367041 A1	30-07-2003
		CN 1695105 A	09-11-2005
		EP 1459165 A2	22-09-2004
		JP 2005515530 T	26-05-2005
		US 2003122787 A1	03-07-2003
		US 2004125088 A1	01-07-2004
WO 2006020305 A	23-02-2006	DE 202005021427 U1	14-02-2008
		DE 202005021492 U1	08-05-2008
		EP 1774429 A2	18-04-2007
		JP 2008508601 T	21-03-2008
		KR 20070039613 A	12-04-2007
WO 0201338 A	03-01-2002	AU 6830201 A	08-01-2002
		EP 1295198 A1	26-03-2003
		TW 525080 B	21-03-2003
		US 6912694 B1	28-06-2005