Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2481396 C 2011/05/10

(11)(21) 2 481 396

(12) BREVET CANADIEN
CANADIAN PATENT

(13) C

(22) Date de dépôt/Filing Date: 2004/09/13

(41) Mise à la disp. pub./Open to Public Insp.: 2005/03/16

(45) Date de délivrance/Issue Date: 2011/05/10

(30) Priorité/Priority: 2003/09/16 (US10/662,813)

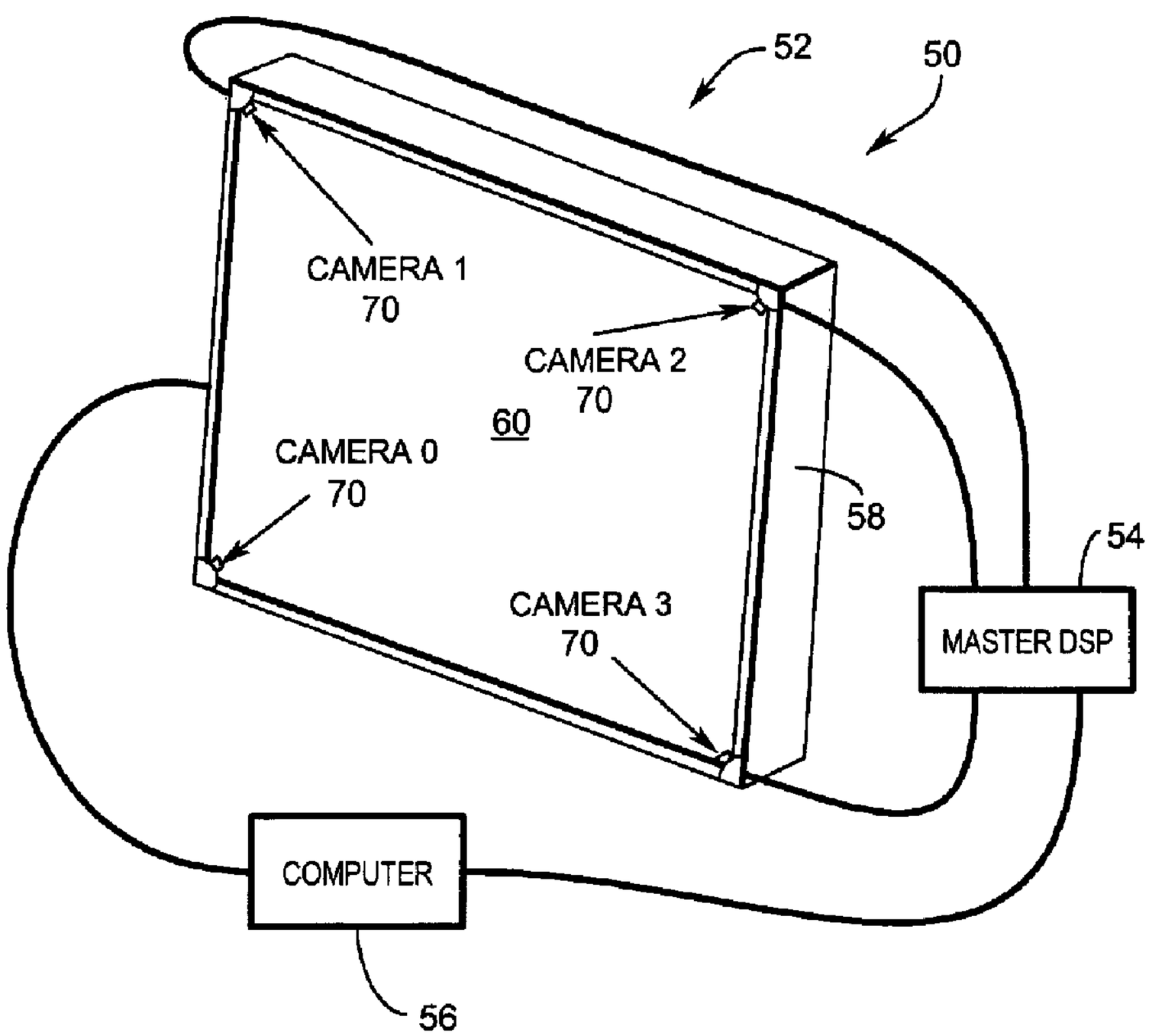
(51) Cl.Int./Int.Cl. *G06K 9/62* (2006.01),
G06F 3/042 (2006.01)

(72) Inventeurs/Inventors:
HILL, DOUGLAS B., CA;
MORRISON, GERALD D., CA

(73) Propriétaire/Owner:
SMART TECHNOLOGIES ULC, CA

(74) Agent: SIM & MCBURNEY

(54) Titre : METHODE DE RECONNAISSANCE GESTUELLE ET SYSTEME TACTILE AINSI EQUIPE
(54) Title: GESTURE RECOGNITION METHOD AND TOUCH SYSTEM INCORPORATING THE SAME



(57) Abrégé/Abstract:
A gesture recognition method includes detecting multiple pointers in close proximity to a touch surface to determine if the multiple pointers are being used to perform a known gesture. When the multiple pointers are being used to perform a known gesture, a command associated with the gesture is executed. A touch system incorporating the gesture recognition method is also provided.

ABSTRACT

A gesture recognition method includes detecting multiple pointers in close proximity to a touch surface to determine if the multiple pointers are being used to perform a known gesture. When the multiple pointers are being used to perform a known gesture, a command associated with the gesture is executed. A touch system incorporating the gesture recognition method is also provided.

GESTURE RECOGNITION METHOD AND TOUCH SYSTEM INCORPORATING THE SAME

Field of the Invention

The present invention relates generally to touch systems and in particular to a gesture recognition method and touch system incorporating the same.

Background of the Invention

Touch systems are well known in the art and typically include a touch screen having a touch surface on which contacts are made using a pointer in order to generate user input. Pointer contacts with the touch surface are detected and are used to generate corresponding output depending on areas of the contact surface where the contacts are made. There are basically two general types of touch systems available and they can be broadly classified as "active" touch systems and "passive" touch systems.

Active touch systems allow a user to generate user input by contacting the touch surface with a special pointer that usually requires some form of on-board power source, typically batteries. The special pointer emits signals such as infrared light, visible light, ultrasonic frequencies, electromagnetic frequencies, etc. that activate the touch surface.

Passive touch systems allow a user to generate user input by contacting the touch surface with a passive pointer and do not require the use of a special pointer in order to activate the touch surface. The pointer can be a finger, a cylinder of some material, or any suitable object that can be used to contact some predetermined area of interest on the touch surface.

Passive touch systems provide advantages over active touch systems in that any suitable pointing device, including a user's finger, can be used as a pointer to contact the touch surface. As a result, user input can easily be generated. Also, since special active pointers are not necessary in passive touch systems, battery power levels and/or pointer damage, theft, or misplacement are of no concern to users.

For example, U.S. Patent No. 6,803,906 issued on October 12, 2004 and International PCT Application No. PCT/CA01/00980 filed on July 5, 2001 and published under No. WO 02/03316 on January 10, 2002, assigned to SMART Technologies Inc., assignee of the present invention, disclose a camera-based touch system comprising a touch screen that includes a passive touch surface on which a

computer-generated image is presented. A rectangular bezel or frame surrounds the touch surface and supports digital cameras at its corners. The digital cameras have overlapping fields of view that encompass and look across the touch surface. The digital cameras acquire images looking across the touch surface from different locations and generate image data. Image data acquired by the digital cameras is processed by digital signal processors associated with the digital cameras to determine if a pointer exists in the captured image data. When it is determined that a pointer exists in the captured image data, the digital signal processors generate pointer information packets (PIPs) and convey the PIPs to a master controller. Each PIP includes a header portion, a data portion and a checksum. The data portion includes a pointer ID field that stores a pointer identifier to allow multiple pointers to be tracked. The data portion also includes a pointer location parameter that identifies a pointer x-position and a pointer tip parameter that identifies a pointer z-position. A contact state field stores a value indicating whether the pointer is in or out of contact with the touch surface allowing pointer hover to be detected.

Upon receipt of the PIPs, the master controller processes the PIPs using triangulation to determine the location of each pointer in the captured images relative to the touch surface in (x,y) coordinates. In this manner, as PIPs are generated in response to captured images, the position and movement of pointers over the touch surface can be tracked. The pointer location data generated by the master controller is conveyed to a computer executing one or more application programs. The computer uses the pointer location data to update the computer-generated image that is presented on the touch surface. Pointer contacts on and pointer movement over the touch surface can therefore be recorded as writing or drawing or used to control execution of application programs executed by the computer.

As will be appreciated, since digital cameras at the corners of the bezels are used to capture image data, the touch system is able to determine when multiple pointers contact and move across the touch surface. This of course provides for enhanced functionality as compared to analog resistive touch systems that are only able to track a single pointer. Although enhanced functionality is provided by the above-described camera-based touch system, to-date, this enhanced functionally has

not been fully exploited. It is therefore an object of the present invention to provide a novel gesture recognition method and touch system incorporating the same.

Summary of the Invention

Accordingly, in one aspect of the present invention there is provided a gesture recognition method comprising the steps of:

displaying an image on a touch surface;

capturing images looking generally across said touch surface;

processing the captured images to detect successive pointer contacts on said touch surface and examining said pointer contacts to recognize a gesture based on relative positions of said pointer contacts and subsequent relative movement of the pointers along the touch surface; and

when a gesture is recognized, updating the displayed image in accordance with said gesture.

Multiple pointer contacts representing a gesture include multiple finger contacts on the touch surface, a finger contact on the touch surface and an object contact on the touch surface and multiple object contacts on the touch surface.

According to another aspect of the present invention there is provided a gesture recognition method comprising the steps of:

capturing images of a pointer input region;

processing the images to detect different successive pointers within said input region to determine if said different successive pointers are being used to perform a known gesture based on relative movement of said pointers within said input region and pointer types; and

when said different successive pointers are being used to perform the known gesture, executing a command associated with said gesture.

Preferably, during the detecting, pointer contacts with or close pointer hovers over the touch surface are detected to determine if a known gesture is being performed and specifically if one of a number of known gestures is being performed, each known gesture being associated with a different command. In a preferred embodiment, the movement of the multiple pointers relative to the touch surface and/or the pointer type determines the gesture being performed.

According to yet another aspect of the present invention there is provided an input detection method in an interactive system capable of detecting movement of multiple pointers within an input region, said method comprising the steps of:

capturing images looking generally across said input region;

analyzing said images to detect different successive pointers within said input region;

when different successive pointers are detected, examining data associated with said different successive pointers to determine if the data represents an input gesture, said data representing the relative movement of said pointers within said input region and types of said pointers; and

when the data represents the input gesture, executing a command corresponding to the input gesture.

According to yet another aspect of the present invention there is provided a touch system comprising:

a touch surface;

at least one imaging device having a field of view looking generally across said touch surface; and

processing structure communicating with said at least one imaging device and analyzing images acquired by said at least one imaging device to determine locations on said touch surface where pointer contacts are made, when said touch surface is contacted by different pointers in succession, said processing structure examining relative positions of said successive pointer contacts and subsequent relative movement of the pointers along the touch surface to recognize a gesture and when the gesture is recognized, said processing structure executing a command associated with said gesture.

According to yet another aspect of the present invention there is provided a gesture recognition method comprising:

detecting different successive pointers contacting a touch surface and subsequent relative movement of the pointers along the touch surface to determine if the different successive pointers are being used to perform a known gesture; and

when the different successive pointers are being used to perform the known gesture, executing a command associated with said gesture.

According to yet another aspect of the present invention there is provided a gesture recognition method comprising the steps of:

capturing images of a pointer input region;

processing the images to detect different successive pointers brought into said input region and subsequent relative movement of the pointers within the input region to determine if said different successive pointers are being used to perform a known gesture; and

when said different successive pointers are being used to perform the known gesture, executing a command associated with said gesture.

According to yet another aspect of the present invention there is provided a gesture recognition method comprising the steps of:

displaying an image on a touch surface;

capturing images looking generally across said touch surface;

processing the captured images to detect different successive pointers contacting said touch surface and subsequent relative movement of the pointers along the touch surface thereby to recognize an input gesture; and

when the input gesture is recognized, updating the displayed image in accordance with said recognized gesture.

According to yet another aspect of the present invention there is provided an event generation method comprising the steps of:

capturing images of a touch surface from different vantages using imaging devices that are proximate to the touch surface and aimed to look generally across said touch surface;

processing the captured images to detect successive pointer contacts on said touch surface and to recognize a gesture based on relative positions of the successive pointer contacts and subsequent relative movement of the pointers along the touch surface; and

generating an event when a gesture is recognized.

According to yet another aspect of the present invention there is provided an event generation method comprising the steps of:

detecting successive pointers brought into contact with and oriented generally normal to a touch surface and examining relative positions of the pointer contacts and subsequent relative movement of said pointers along said touch surface by processing images of the touch surface captured using imaging devices proximate to and aimed to look generally across the touch surface to determine if a gesture has been input; and

generating an event when an input gesture is determined.

According to yet another aspect of the present invention there is provided an event generation method comprising the steps of:

capturing images of a touch surface from different vantages using imaging devices that are proximate to the touch surface and aimed to look generally across the touch surface;

processing the captured images to detect successive pointers brought into contact with said touch surface and to recognize a gesture based on pointer contact positions and subsequent relative movement of the pointers along the touch surface; and

generating an event when a gesture is recognized.

According to yet another aspect of the present invention there is provided an event generation method comprising the steps of:

detecting multiple pointers brought successively into contact with and oriented substantially normal to a touch surface and examining relative positions of the pointer contacts and subsequent relative movement of said pointers along said touch surface by processing images of the touch surface captured using imaging devices that are proximate to and aimed to look generally across the touch surface to determine if a gesture has been input; and

generating an event when an input gesture is determined.

According to still yet another aspect of the present invention there is provided an event generation method comprising the steps of:

capturing images from different vantages using imaging devices that are proximate to and aimed to look generally across a pointer input region;

processing the captured images to detect positions of multiple pointers brought successively into said pointer input region and to recognize a gesture based on the detected pointer positions and the subsequent relative movement of the pointers within said pointer input region; and

generating an event when a gesture is recognized.

The present invention provides advantages in that since gestures represented by multiple pointer contacts on and/or movement over the touch surface are detected and corresponding commands generated, enhanced touch system functionality is provided.

Brief Description of the Drawings

Embodiments of the present invention will now be described more fully with reference to the accompanying drawings in which:

Figure 1 is a schematic diagram of a camera-based touch system in accordance with the present invention;

Figure 2 is a front elevation view of a touch screen forming part of the touch system of Figure 1;

Figures 3a to 3d are front perspective views of the touch screen of Figure 2 showing an input right-click gesture; and

Figures 4a to 4f are front perspective views of the touch screen of Figure 2 showing input up/down and left/right scroll gestures.

Detailed Description of the Preferred Embodiments

Turning now to Figure 1, a camera-based touch system in accordance with the present invention is shown and is generally identified by reference numeral 50. Camera-based touch system 50 is similar to that disclosed in International PCT Application Serial No. WO 02/03316, assigned to SMART Technologies Inc., assignee of the present invention. As can be seen, touch system 50 includes a touch screen 52 coupled to a digital signal processor (DSP) based master controller 54. Master controller 54 is also coupled to a computer 56. Computer 56 executes one or more application programs and provides computer-generated image output that is displayed on the touch screen 52. The coordinate system of the touch system 52 is mapped to the coordinate system of the computer . The touch screen 52, master controller 54 and computer 56 form a closed-loop so that pointer hover or contacts with and pointer movement over or above the touch screen 52 can be recorded as writing or drawing or used to control execution of application programs executed by the computer 56.

Figure 2 better illustrates the touch screen 52. Touch screen 52 in the present embodiment includes a high-resolution display device such as a plasma display 58, the front surface of which defines a touch surface 60. The touch surface 60 is bordered by an illuminated bezel or frame 62 coupled to the display device. Illuminated bezel 62 is of the type disclosed in U.S. Patent No. 6,972,401 issued on December 6, 2005 and includes elongate side frame assemblies 64 that are coupled to the sides of the plasma display 58. Each side frame assembly 64 accommodates a light source (not shown) that projects infrared backlighting across the touch surface

60. The ends of the side frame assemblies 64 are joined by corner pieces 68 that house DSP-based CMOS digital cameras 70. Each digital camera 70 is mounted within its respective corner piece 68 so that its field of view encompasses and looks across the entire touch surface 60.

During operation, the digital cameras 70 acquire images of the touch surface 60 and generate image data. The acquired image data is processed by digital signal processors associated with the digital cameras 70 to determine if a pointer exists in the captured images. When it is determined that one or more pointers exist in the acquired image data, the digital signal processors of the digital cameras 70 generate pointer information packets (PIPs) and convey the PIPs to the digital signal processor (DSP) based master controller 54. Each PIP includes a header portion, a data portion and a checksum. The data portion includes a pointer ID field that stores a pointer identifier to allow multiple pointers to be tracked. The data portion also includes a pointer location parameter that identifies a pointer x-position and a pointer tip parameter that identifies a pointer z-position. A contact state field stores a value indicating whether the pointer is in or out of contact with the touch surface 60 allowing pointer hover to be detected.

Upon receipt of the PIPs, the master controller 54 processes the PIPs using triangulation to determine the location of each pointer in the captured images relative to the touch surface 60 in (x,y) coordinates. In this manner, as PIPs are generated in response to captured images, the position and movement of pointers over the touch surface 60 can be tracked. Since image data is processed to detect the existence of one or more pointers, the pointers may take any suitable form such as for example, a user's finger, a cylinder of material, a passive or active pen tool or erase tool or other appropriate object. Specifics of the manner by which the image data is acquired by the digital cameras 70 and processed by the master controller 54 are described in International PCT Application No. PCT/CA01/00980 filed on July 5, 2001 and published under No. WO 02/03316 on January 10, 2002. Accordingly, such specifics will not be described further herein.

The master controller 54 outputs generated pointer data to the computer 56 that identifies the location of each pointer relative to the touch surface as each pointer approaches and/or contacts and moves over the touch surface 60. A

driver loaded on the computer 56 receives the pointer data and examines the pointer data to determine if the pointer data has been generated in response to a known input gesture stored in a gesture library. Specifically, the driver examines the pointer data to detect the existence of multiple pointers in captured images and then examines the nature of the multiple pointers to determine if a known gesture has been performed such as for example a right-click gesture, a scroll gesture, a rotate gesture etc. When a gesture has been performed, the driver generates a command (event) that is associated with the determined gesture and conveys the command to the active application program being executed by the computer 56.

Turning now to Figures 3a to 4f, examples of gestures that can be recognized by the touch system and the resulting actions that are performed in response to the recognized gestures are shown.

Intuitive Right-Click Gesture

Figures 3a to 3d illustrate an intuitive right-click gesture. When a user contacts the touch surface 60 with a finger over an application displayed on the touch surface, the driver recognizes the contact as a left-click mouse event and injects the left-click mouse event into the application. If the user subsequently contacts the touch surface 60 with another finger while maintaining the contact with the one finger and the subsequent contact is to the right of and close to the initial contact, the driver recognizes the second touch surface contact as a right-click gesture and injects a right-click event into the application. In response to the right-click event, the application opens and displays a drop down menu (not shown). As will be appreciated, this enables a user to invoke a right-click action using a hand gesture that is similar to the action performed when invoking a right-click action using a mouse. Although, Figures 3a to 3d show the intuitive right-click gesture being performed using two fingers on the same hand, it will be appreciated that the right-click gesture can be performed using a finger on different hands.

Scroll Up/Down and Left/Right Gesture

Figures 4a to 4e illustrate up/down and left/right scroll gestures. If the user contacts the touch surface 60 with a pair of fingers simultaneously over an application window displayed on the touch surface and the fingers are closely and generally horizontally spaced, the driver recognizes the simultaneous finger contacts

as a scroll gesture and injects a scroll event into the application. Pointer position data conveyed to the application by the driver in response to subsequent vertical movement of the fingers is interpreted by the application either as scroll up or scroll down commands. In response to the scroll up or down commands, the application moves information presented within the application window in the direction of the vertical movement. Pointer position data conveyed to the application by the driver in response to subsequent horizontal movement of the fingers is interpreted by the application as scroll to side commands. In response to the scroll to side commands, the application moves information displayed within the application window to the side corresponding to the direction of the horizontal movement. Although Figures 4a to 4f show the scroll gestures being performed using two fingers on the same hand, it will be appreciated that the scroll gestures can be performed using a finger on different hands.

Although not illustrated, a number of other gestures can be recognized by the driver and used to generate commands to control an application being executed by the computer 56. Examples of such other gestures will now be described.

Page Up/Down Gesture

If the user contacts the touch surface 60 with three fingers simultaneously over an application window displayed on the touch surface and the three fingers are closely and generally horizontally spaced, the driver recognizes the simultaneous finger contacts as a page gesture and injects a page event into the application. Pointer position data conveyed to the application by the driver in response to subsequent vertical movement of the fingers is interpreted by the application as page up or page down commands depending on the direction of the vertical movement. In response to the page up or page down commands, the application moves information displayed within the window in the appropriate direction.

Rotate Gesture

If the user contacts the touch surface 60 over an object displayed within an application window with one finger and then subsequently contacts the touch surface with another finger and moves that other finger in an arc while maintaining the touch surface contact with the one finger, the driver recognizes the

arcuate movement of the second finger as a rotate gesture. The driver in turn injects a rotate command into the application causing the application to rotate the object about the contact point defined by the first finger in the direction of the arc and by an amount equivalent to the path of the arc.

Zoom Gesture

If the user contacts the touch surface 60 with a pair of closely spaced fingers simultaneously over an application window and expands the distance between the fingers in a generally horizontal direction, the driver recognizes the finger movement as a zoom-out gesture. The driver in turn injects a zoom-out command into the application causing the application to expand the size of the information presented in the application window. If the user contacts the touch surface 60 with a pair of spaced fingers simultaneously over an application window and moves the fingers in a generally horizontal direction towards one another, the driver recognizes the finger movement as a zoom-in gesture. The driver in turn injects a zoom-in command into the application causing the application to reduce the size of the information presented in the application window.

Alternatively, the zoom-out and zoom-in commands may be represented by other gestures. For example, if the user contacts the touch surface 60 with a clawed hand having its fingers bunched together over an application window and expands the hand by extending the fingers outwardly, the driver recognizes the finger movement as the zoom-out gesture.

If the user contacts the touch surface 60 with a generally flat hand having its fingers extended over an application window and contracts the hand by clawing the fingers inwardly to bunch them together, the driver recognizes the finger movement as the zoom-in gesture.

Expand Gesture

If the user contacts the touch surface 60 with a pair of closely spaced fingers simultaneously over an application window and expands the distance between the fingers in a generally diagonal direction, the driver recognizes the finger movement as an increase window size gesture. The driver in turn injects an increase window size command into the application causing the application to expand the size of the application window. If the user contacts the touch surface 60 with a pair of

spaced fingers simultaneously over an application window and moves the fingers in a generally diagonal direction towards one another, the driver recognizes the finger movement as a decrease window size gesture. The driver in turn injects a decrease window size command into the application causing the application to reduce the size of the application window.

Icon Select and Open Gesture

If the user contacts the touch surface 60 with two closely spaced fingers simultaneously over an icon, the driver recognizes the finger contact as a double-click gesture. The driver in turn generates an open application command causing the desktop application running on the computer 56 to open the selected application.

Object/Window Move Gesture

If the user moves a pair of closely spaced fingers above the touch surface 60 and over an object displayed within an application window, the driver recognizes the hovering finger movement as a translate object gesture. The driver in turn injects a translate object command into the application causing the application to translate the displayed object in the direction of and by an amount equal to the distance the fingers are moved.

If the user moves three closely spaced fingers above the touch surface 60 and over an application window, the driver recognizes the hovering finger movement as a translate window gesture. The driver in turn generates a translate window command causing the desktop application running on the computer 56 to translate the application window in the direction of and by an amount equal to the distance the fingers are moved.

Although the above gestures are described as being recognized in response to multiple finger contacts or hovers, the same gestures can be recognized if other objects are used to perform the gestures. For example, multiple pen tools can be used to perform the gestures or alternatively a finger and a pen tool can be used to perform the gestures.

Also, recognized gestures may be enhanced using different pointer characteristics. For example, in the case of scroll gestures, the angle at which the pointers contact the touch surface 60 may be used to determine the rate at which the

displayed information is scrolled. Pointers contacting the touch surface 60 at a steep angle may represent a slow scroll rate whereas pointers contacting the touch surface 60 at a shallow angle may represent a fast scroll rate.

If the touch system is able to differentiate between the type of pointers used to contact the touch surface 60 as is described in U.S. Patent Application Publication No. 2004/0179001 published on September 16, 2004 and/or is able to determine pointer characteristics as is described in U.S. Patent No. 6,951,197 issued on October 11, 2005, different functionality can be assigned to similar gestures that are performed using different pointers. For example, in the case of the rotate gesture described above, if the same gesture is carried out using a finger to initially contact an object within the application window and a pen tool to describe the arc, the driver recognizes the finger contact and pen movement as a pattern fill gesture rather than a rotate gesture. A finger contact and subsequent closely spaced pen tool contact may represent a draw circle gesture rather than a scroll gesture and a finger contact and subsequent closely spaced eraser contact may represent an erase page gesture. As will be appreciated, being able to differentiate between multiple pointers brought into proximity with the touch surface 60 significantly increases the functions that may be invoked by performing the same gestures simply by using discrete pointers that can be differentiated.

Although the driver is described as examining the pointer data to determine if the pointer data is generated in response to a known gesture, it will be appreciated by those of skill in the art that if the active application being executed by the computer has the capability of recognizing gestures, the pointer data may be conveyed to the active application for gesture recognition.

If desired the touch surface 60 may be partitioned into multiple regions to enable multiple users to interact with the touch surface simultaneously without ambiguity between user input. In this case multiple contacts on or hovers over the touch surface that are beyond a threshold distance are treated as multiple user inputs. Multiple contacts on or hovers over the touch surface that are within the threshold distance are treated as multiple contacts made by a single user and are examined to determine if the multiple contacts represent a gesture.

Although preferred embodiments of the present invention have been described, those of skill in the art will appreciate that variations and modifications may be made without departing from the spirit and scope thereof as defined by the appended claims.

What is claimed is:

1. A gesture recognition method comprising the steps of:

displaying an image on a touch surface;

capturing images looking generally across said touch surface;

processing the captured images to detect successive pointer contacts on said touch surface and examining said pointer contacts to recognize a gesture based on relative positions of said pointer contacts and subsequent relative movement of the pointers along the touch surface; and

when a gesture is recognized, updating the displayed image in accordance with said gesture.

2. The method of claim 1 wherein said successive pointer contacts include successive finger contacts on said touch surface.

3. The method of claim 1 wherein said successive pointer contacts include a finger contact on said touch surface and an object contact on said touch surface.

4. The method of claim 1 wherein said successive pointer contacts include successive object contacts on said touch surface.

5. The method of claim 1 wherein said successive pointer contacts include one of successive finger contacts on said touch surface, a finger contact on said touch surface and an object contact on said touch surface, and multiple object contacts on said touch surface.

6. The method of any one of claims 1 to 5 wherein said gesture represents a graphical object manipulation command.

7. A gesture recognition method comprising the steps of:

capturing images of a pointer input region;

processing the images to detect different successive pointers within said input region to determine if said different successive pointers are being used to perform a known gesture based on relative movement of said pointers within said input region and pointer types; and

when said different successive pointers are being used to perform the known gesture, executing a command associated with said gesture.

8. The method of claim 7 wherein during said processing, pointer contacts with or close pointer hovers over a touch surface are detected to determine if the known gesture is being performed.

9. The method of claim 8 wherein said successive pointers include multiple fingers, at least one finger and at least one object, and multiple objects in close proximity to said touch surface.

10. The method of claim 8 wherein during said detecting, the successive pointers are examined to determine if one of a number of known gestures is being performed, each known gesture being associated with a different command.

11. The method of claim 10 wherein movement of the successive pointers across the touch surface determines the gesture being performed.

12. The method of claim 10 wherein pointer type determines the gesture being performed.

13. The method of claim 10 wherein movement of the successive pointers relative to the touch surface and pointer type determines the gesture being performed.

14. An input detection method in an interactive system capable of detecting movement of multiple pointers within an input region, said method comprising the steps of:

capturing images looking generally across said input region;

analyzing said images to detect different successive pointers within said input region;

when different successive pointers are detected, examining data associated with said different successive pointers to determine if the data represents an input gesture, said data representing relative movement of said pointers within said input region and types of said pointers; and

when the data represents the input gesture, executing a command corresponding to the input gesture.

15. The method of claim 14 wherein said gesture represents one of a graphical object resize command, a graphical object rotate command and a graphical object translate command event.

16. The method of claim 14 wherein said gesture represents a scroll event.

17. The method of any one of claims 14 to 16 further comprising, prior to said examining, when different successive pointers are detected, determining if the successive pointers are within a threshold distance of one another and if so, performing said examining to determine if the data represents the input gesture, and if not, treating each successive pointer as independent input.

18. A touch system comprising:

a touch surface;

at least one imaging device having a field of view looking generally across said touch surface; and

processing structure communicating with said at least one imaging device and analyzing images acquired by said at least one imaging device to determine locations on said touch surface where pointer contacts are made, when said touch surface is contacted by different pointers in succession, said processing structure examining relative positions of said successive pointer contacts and subsequent relative movement of the pointers along the touch surface to recognize a

gesture and when the gesture is recognized, said processing structure executing a command associated with said gesture.

19. A touch system according to claim 18 wherein each gesture is further represented by pointer type.

20. A gesture recognition method comprising:

detecting different successive pointers contacting a touch surface and subsequent relative movement of the pointers along the touch surface to determine if the different successive pointers are being used to perform a known gesture; and

when the different successive pointers are being used to perform the known gesture, executing a command associated with said gesture.

21. The method of claim 20 wherein during said detecting, pointer contacts with or close pointer hovers over the touch surface are detected to determine if the known gesture is being performed.

22. The method of claim 21 wherein said pointers include one of multiple fingers, at least one finger and at least one object, and multiple objects in contact with or in close proximity to said touch surface.

23. The method of claim 21 wherein during said detecting, the pointers are examined to determine if one of a number of known gestures is being performed, each known gesture being associated with a different command.

24. A gesture recognition method comprising the steps of:

capturing images of a pointer input region;

processing the images to detect different successive pointers brought into said input region and subsequent relative movement of the pointers within the input region to determine if said different successive pointers are being used to perform a known gesture; and

when said different successive pointers are being used to perform the known gesture, executing a command associated with said gesture.

25. The method of claim 24 wherein during said processing, the successive pointers are examined to determine if one of a number of known gestures is being performed, each known gesture being associated with a different command.

26. A gesture recognition method comprising the steps of:

displaying an image on a touch surface;

capturing images looking generally across said touch surface;

processing the captured images to detect different successive pointers contacting said touch surface and subsequent relative movement of the pointers along the touch surface thereby to recognize an input gesture; and

when the input gesture is recognized, updating the displayed image in accordance with said recognized gesture.

27. The method of claim 26 wherein said different pointers comprise different fingers.

28. The method of claim 26 wherein said different pointers comprise a finger and an object.

29. The method of claim 26 wherein said input gesture represents one of a graphical object resize command, a graphical object rotate command and a graphical object translate command event.

30. An event generation method comprising the steps of:

capturing images of a touch surface from different vantages using imaging devices that are proximate to the touch surface and aimed to look generally across said touch surface;

processing the captured images to detect successive pointer contacts on said touch surface and to recognize a gesture based on relative positions of the

successive pointer contacts and subsequent relative movement of the pointers along the touch surface; and

generating an event when a gesture is recognized.

31. The method of claim 30 wherein during said processing, the images are processed to detect movement of multiple pointers towards one another.

32. The method of claim 30 wherein during said processing, the images are processed to detect movement of multiple pointers away from one another.

33. The method of claim 30 wherein during said processing, the images are processed to detect movement of one pointer in a generally arcuate path about another pointer.

34. The method of any one of claims 30 to 33 wherein said event represents a graphical object manipulation command.

35. The method of claim 34 wherein said graphical object manipulation command is one of a graphical object resize command, a graphical object rotate and a graphical object translate command.

36. The method of claim 35 further comprising executing said graphical object manipulation command.

37. The method of any one of claims 30 to 36 further comprising, prior to processing the captured images to recognize the gesture, when successive pointer contacts on said touch surface are detected, determining if the successive pointer contacts are within a threshold distance of one another and if so, performing the processing to recognize the gesture and if not, treating each successive pointer as independent input.

38. An event generation method comprising the steps of:

detecting successive pointers brought into contact with and oriented generally normal to a touch surface and examining relative positions of the pointer contacts and subsequent relative movement of said pointers along said touch surface by processing images of the touch surface captured using imaging devices proximate to and aimed to look generally across the touch surface to determine if a gesture has been input; and

generating an event when an input gesture is determined.

39. The method of claim 38 wherein during said examining, the subsequent relative movement of said pointers is examined to determine if said pointers are moving towards one another.

40. The method of claim 38 wherein during said examining, the subsequent relative movement of said pointers is examined to determine if said pointers are moving away from one another.

41. The method of claim 38 wherein during said examining, the subsequent relative movement of said pointers is examined to determine if one pointer is moving in a generally arcuate path about another pointer.

42. The method of claim 38 wherein during said examining, the subsequent relative movement of said pointers is examined to determine if said pointers are moving in a generally similar direction.

43. The method of any one of claims 38 to 42 wherein said event represents a graphical object manipulation command.

44. The method of claim 43 wherein said graphical object manipulation command is one of a graphical object resize command, a graphical object rotate and a graphical object translate command.

45. The method of claim 44 further comprising executing said graphical object manipulation command.

46. An event generation method comprising the steps of:

capturing images of a touch surface from different vantages using imaging devices that are proximate to the touch surface and aimed to look generally across the touch surface;

processing the captured images to detect successive pointers brought into contact with said touch surface and to recognize a gesture based on pointer contact positions and subsequent relative movement of the pointers along the touch surface; and

generating an event when a gesture is recognized.

47. The method of claim 46 wherein during said processing, the images are processed to detect movement of multiple pointers towards one another.

48. The method of claim 46 wherein during said processing, the images are processed to detect movement of multiple pointers away from one another.

49. The method of claim 46 wherein during said processing, the images are processed to detect movement of one pointer in a generally arcuate path about another pointer.

50. The method of any one of claims 46 to 49 wherein said event represents a graphical object manipulation command.

51. The method of claim 50 wherein said graphical object manipulation command is one of a graphical object resize command, a graphical object rotate and a graphical object translate command.

52. The method of claim 51 further comprising executing said graphical object manipulation command.

53. The method of any one of claims 46 to 52 further comprising, prior to processing the captured images to recognize the gesture, when successive pointer contacts on said touch surface are detected, determining if the successive pointer contacts are within a threshold distance of one another and if so, performing the processing to recognize the gesture and if not, treating each successive pointer as independent input.

54. An event generation method comprising the steps of:

detecting multiple pointers brought successively into contact with and oriented substantially normal to a touch surface and examining relative positions of the pointer contacts and subsequent relative movement of said pointers along said touch surface by processing images of the touch surface captured using imaging devices that are proximate to and aimed to look generally across the touch surface to determine if a gesture has been input; and

generating an event when an input gesture is determined.

55. The method of claim 54 wherein during said examining, the subsequent relative movement of said pointers is examined to determine if said pointers are moving towards one another.

56. The method of claim 54 wherein during said examining, the subsequent relative movement of said pointers is examined to determine if said pointers are moving away from one another.

57. The method of claim 54 wherein during said examining, the subsequent relative movement of said pointers is examined to determine if one pointer is moving in a generally arcuate path about another pointer.

58. The method of claim 54 wherein during said examining, the subsequent relative movement of said pointers is examined to determine if said pointers are moving in a generally similar direction.

59. The method of any one of claims 54 to 58 wherein said event represents a graphical object manipulation command.

60. The method of claim 59 wherein said graphical object manipulation command is one of a graphical object resize command, a graphical object rotate and a graphical object translate command.

61. The method of claim 60 further comprising executing said graphical object manipulation command.

62. An event generation method comprising the steps of:

capturing images from different vantages using imaging devices that are proximate to and aimed to look generally across a pointer input region;

processing the captured images to detect positions of multiple pointers brought successively into said pointer input region and to recognize a gesture based on the detected pointer positions and the subsequent relative movement of the pointers within said pointer input region; and

generating an event when a gesture is recognized.

63. The method of claim 62 wherein during said processing, the images are processed to detect movement of multiple pointers towards one another.

64. The method of claim 62 wherein during said processing, the images are processed to detect movement of multiple pointers away from one another.

65. The method of claim 62 wherein during said processing, the images are processed to detect movement of one pointer in a generally arcuate path about another pointer.

66. The method of claim 62 wherein during said processing, the images are processed to detect movement of multiple pointers in a generally similar direction.

67. The method of any one of claims 62 to 66 wherein said event represents a graphical object manipulation command.

68. The method of claim 67 wherein said graphical object manipulation command is one of a graphical object resize command, a graphical object rotate and a graphical object translate command.

69. The method of claim 68 further comprising executing said graphical object manipulation command.

70. The method of any one of claims 62 to 69 further comprising, prior to processing the captured images to recognize the gesture, when successive pointer contacts on said touch surface are detected, determining if the successive pointer contacts are within a threshold distance of one another and if so, performing the processing to recognize the gesture and if not, treating each successive pointer as independent input.

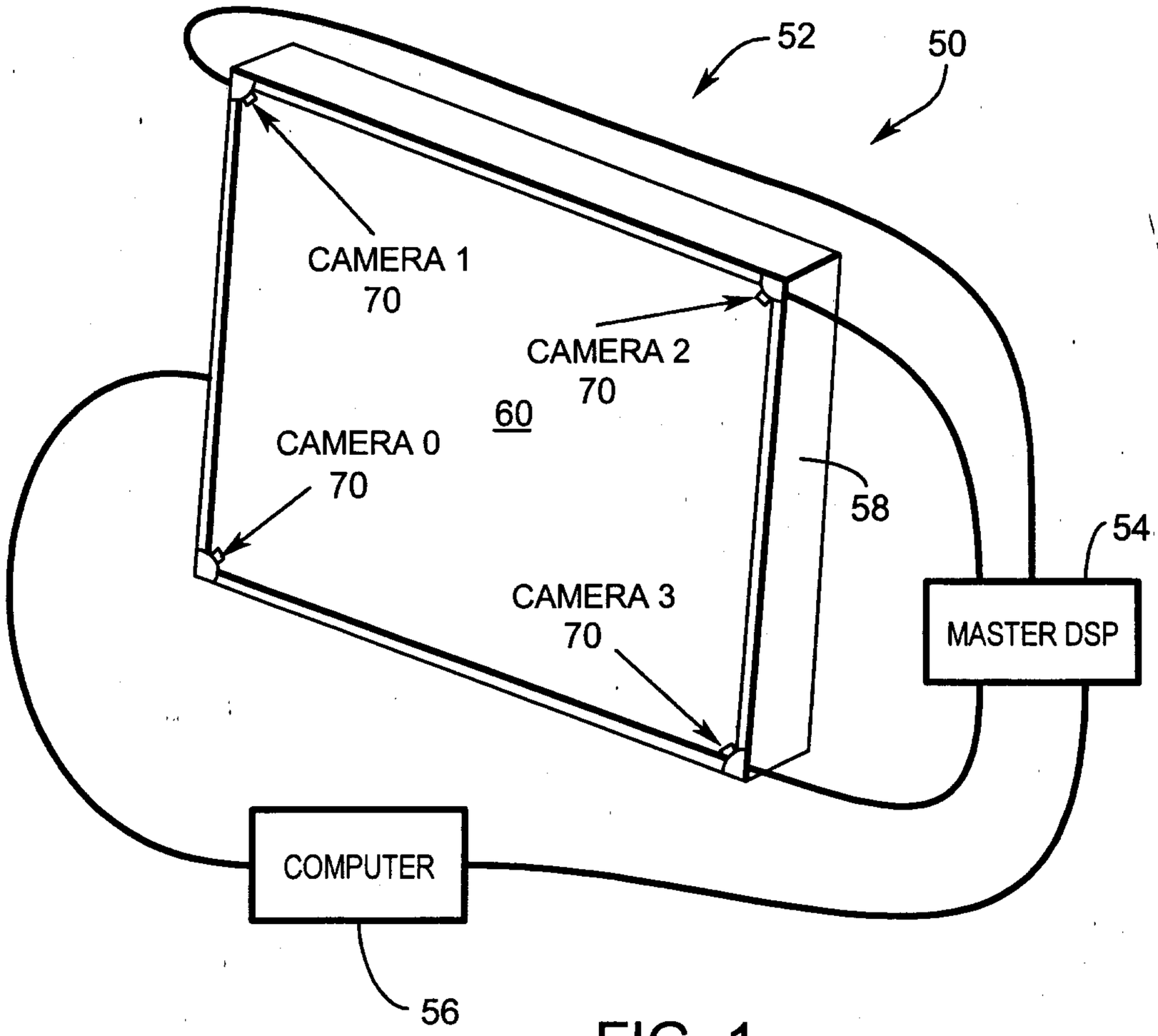

52
50
CAMERA 1
70
CAMERA 2
70
60
CAMERA 0
70
58
54
CAMERA 3
70
MASTER DSP
COMPUTER
56

FIG. 1

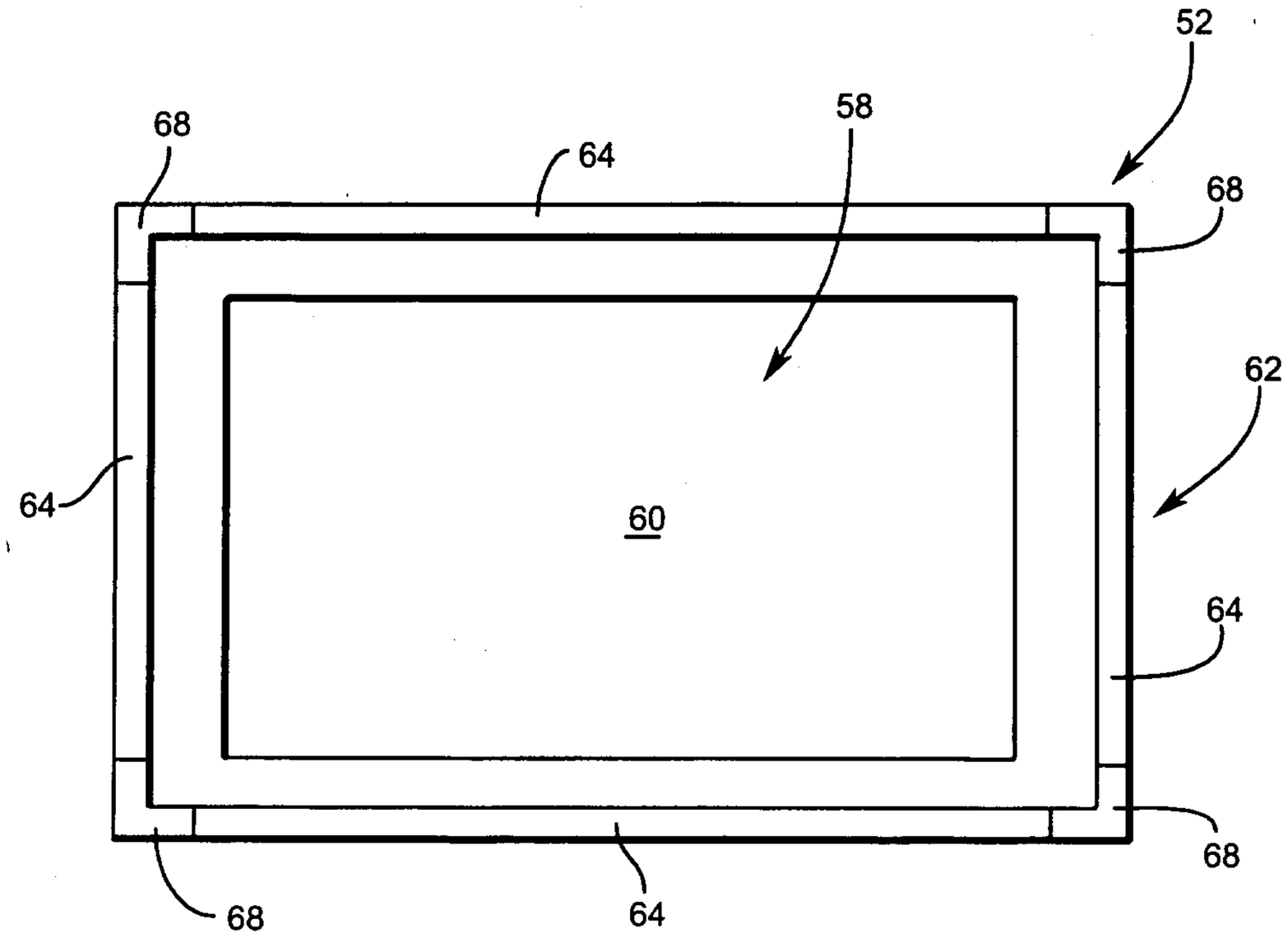

52
68
64
58
68
62
60
64
64
68
68
64

FIG. 2

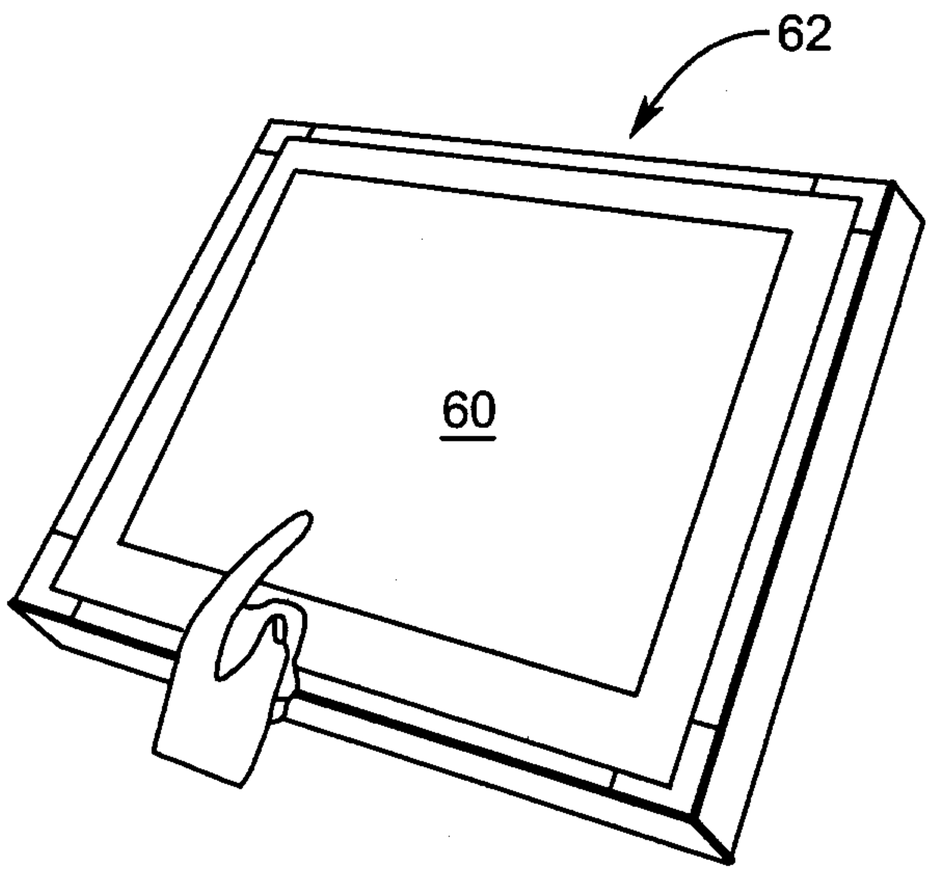


FIG. 3a

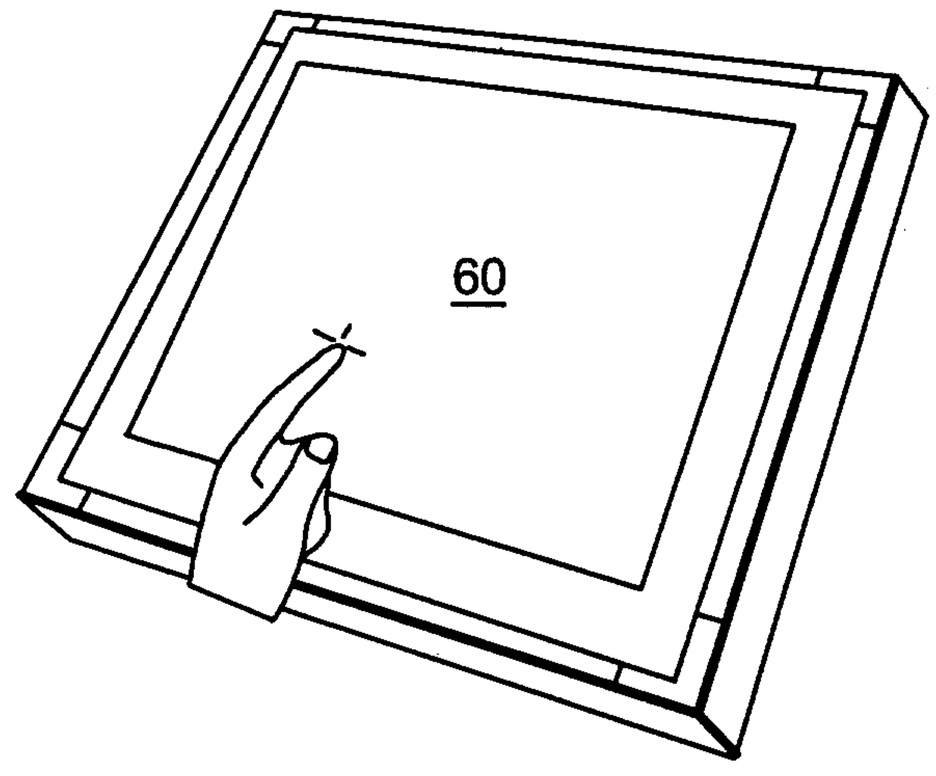


FIG. 3b

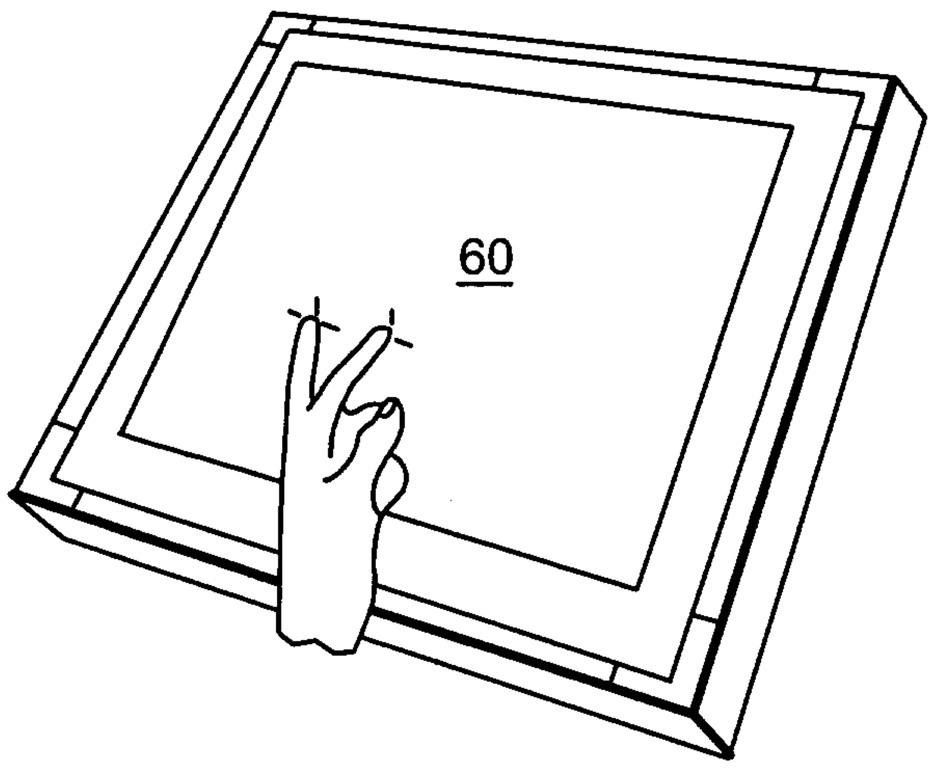


FIG. 3c

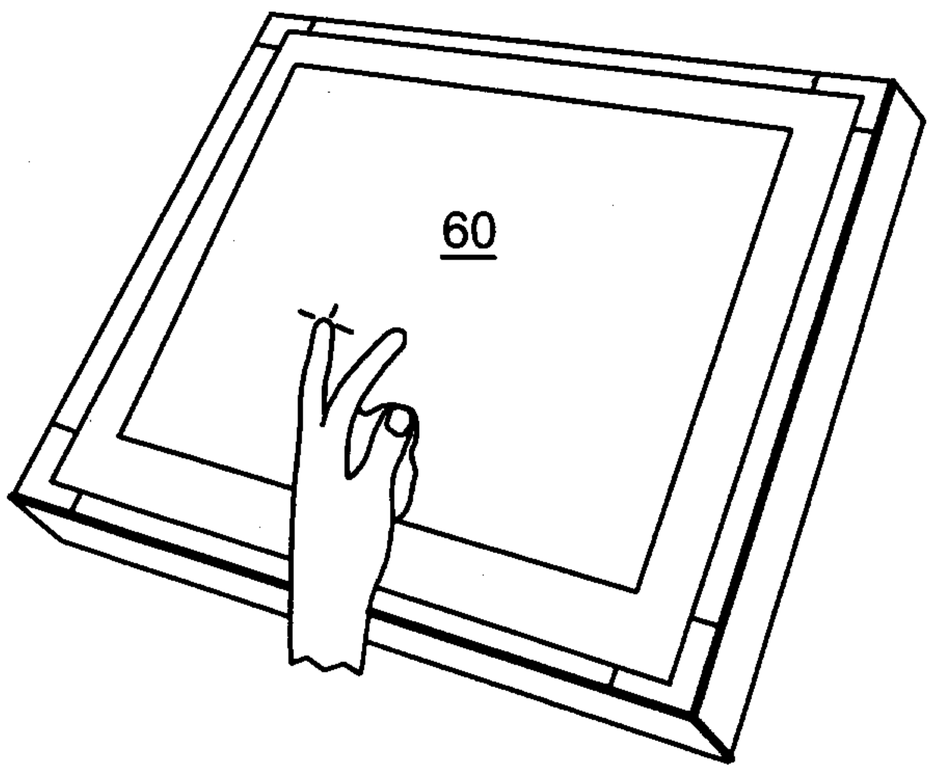


FIG. 3d

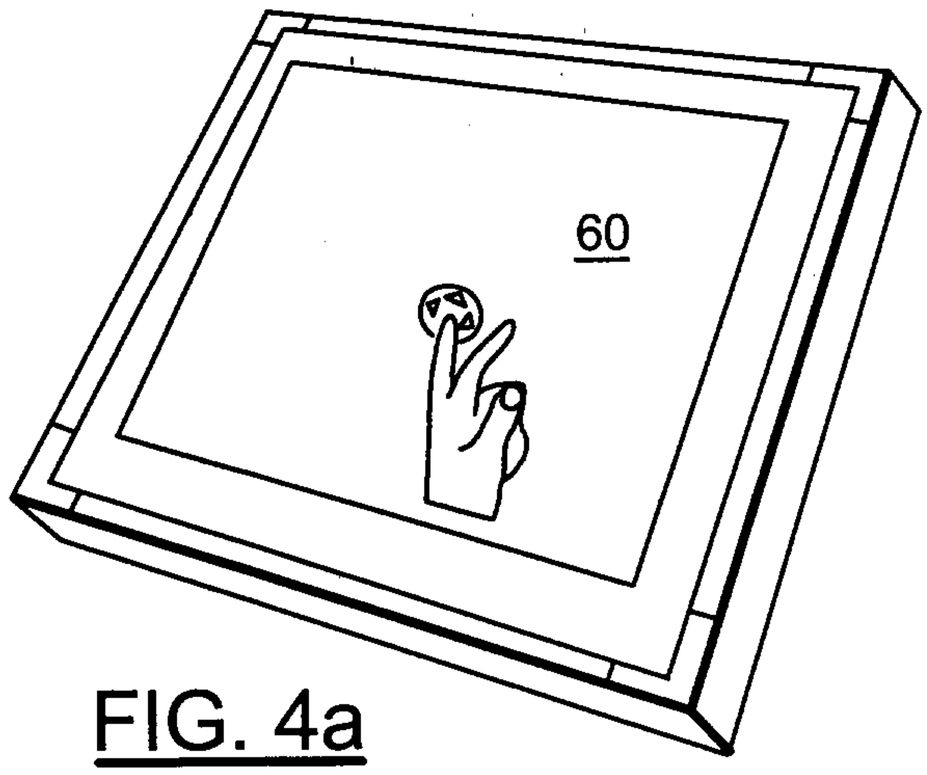


FIG. 4a

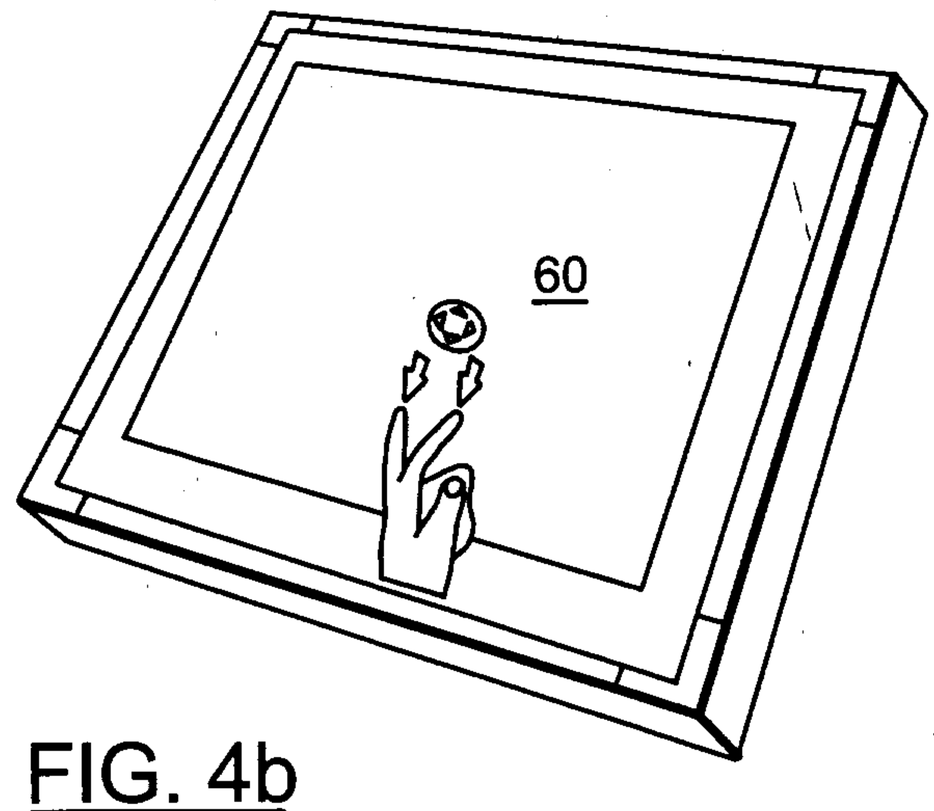


FIG. 4b

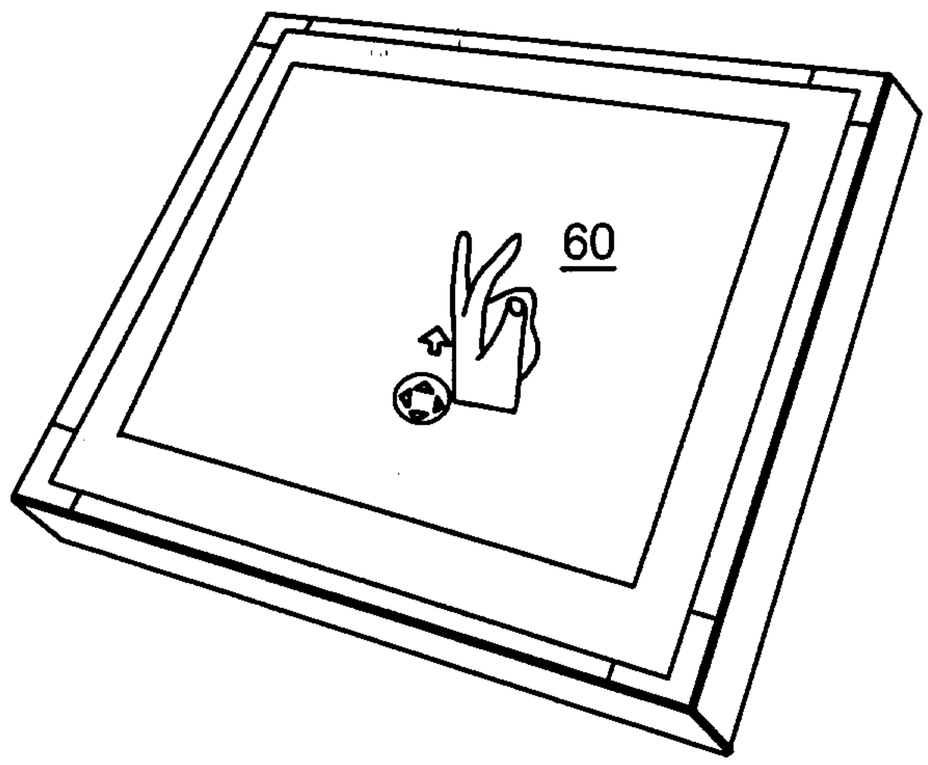


FIG. 4c

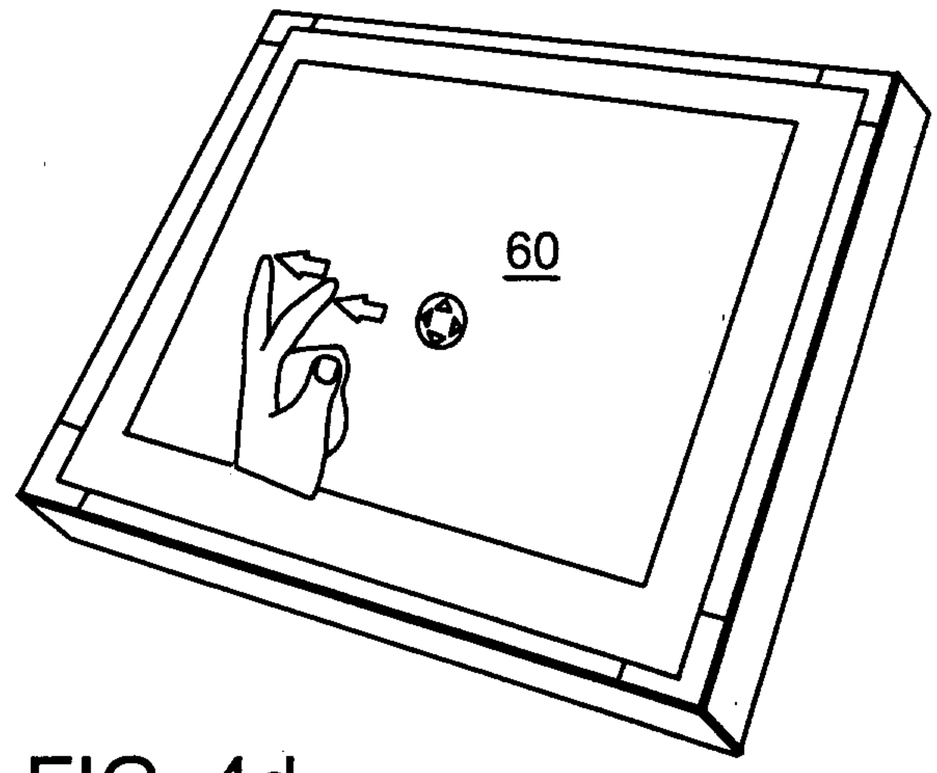


FIG. 4d

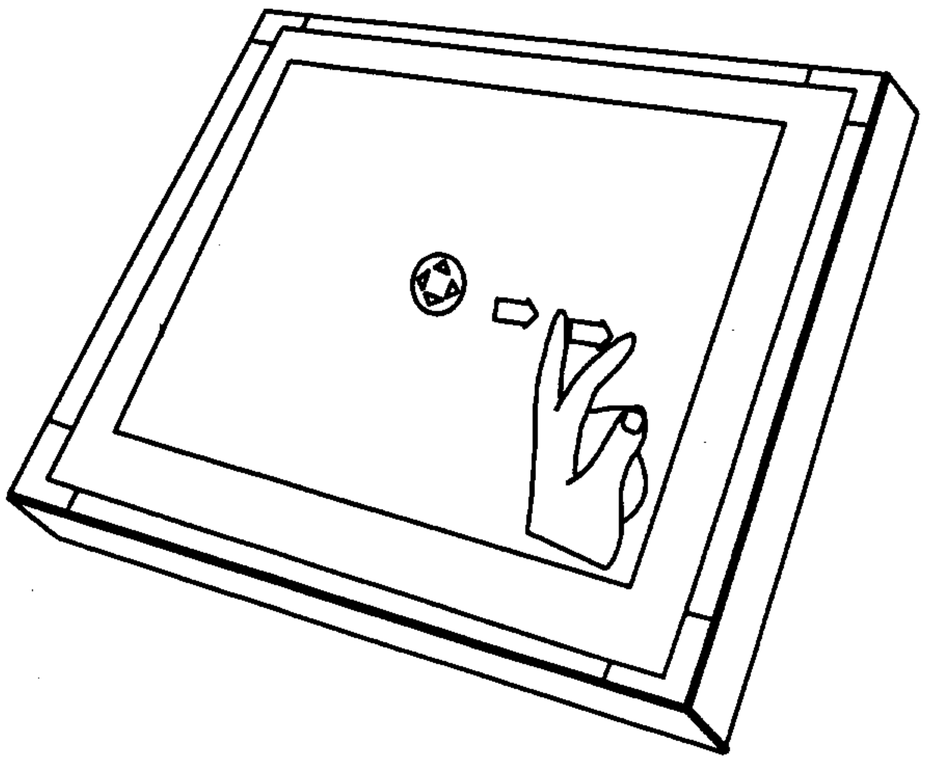


FIG. 4e

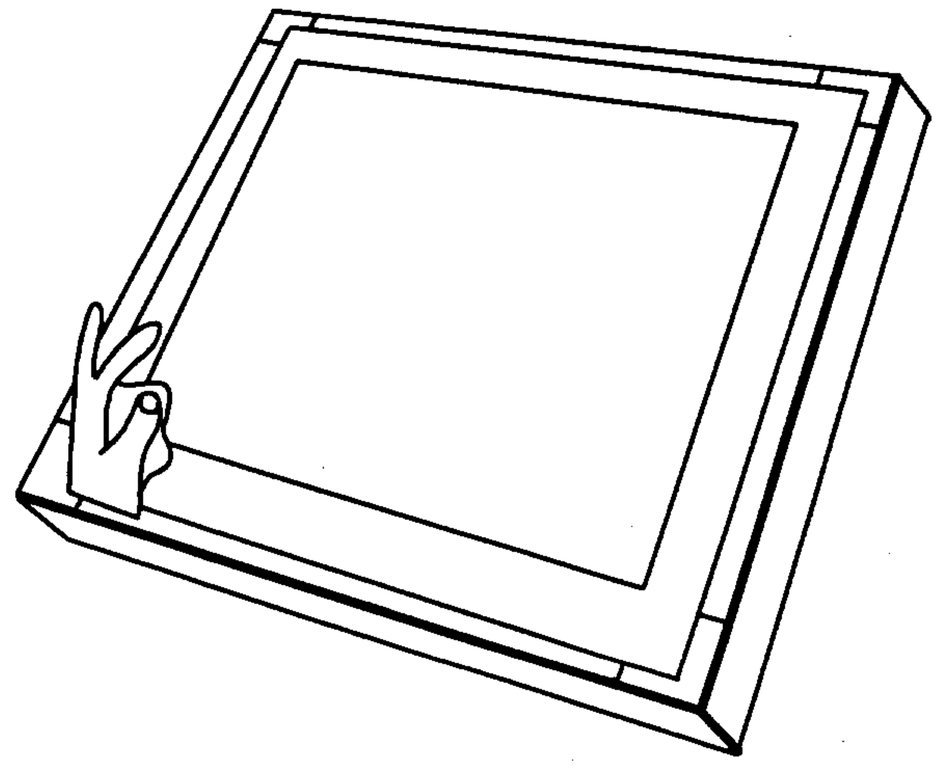
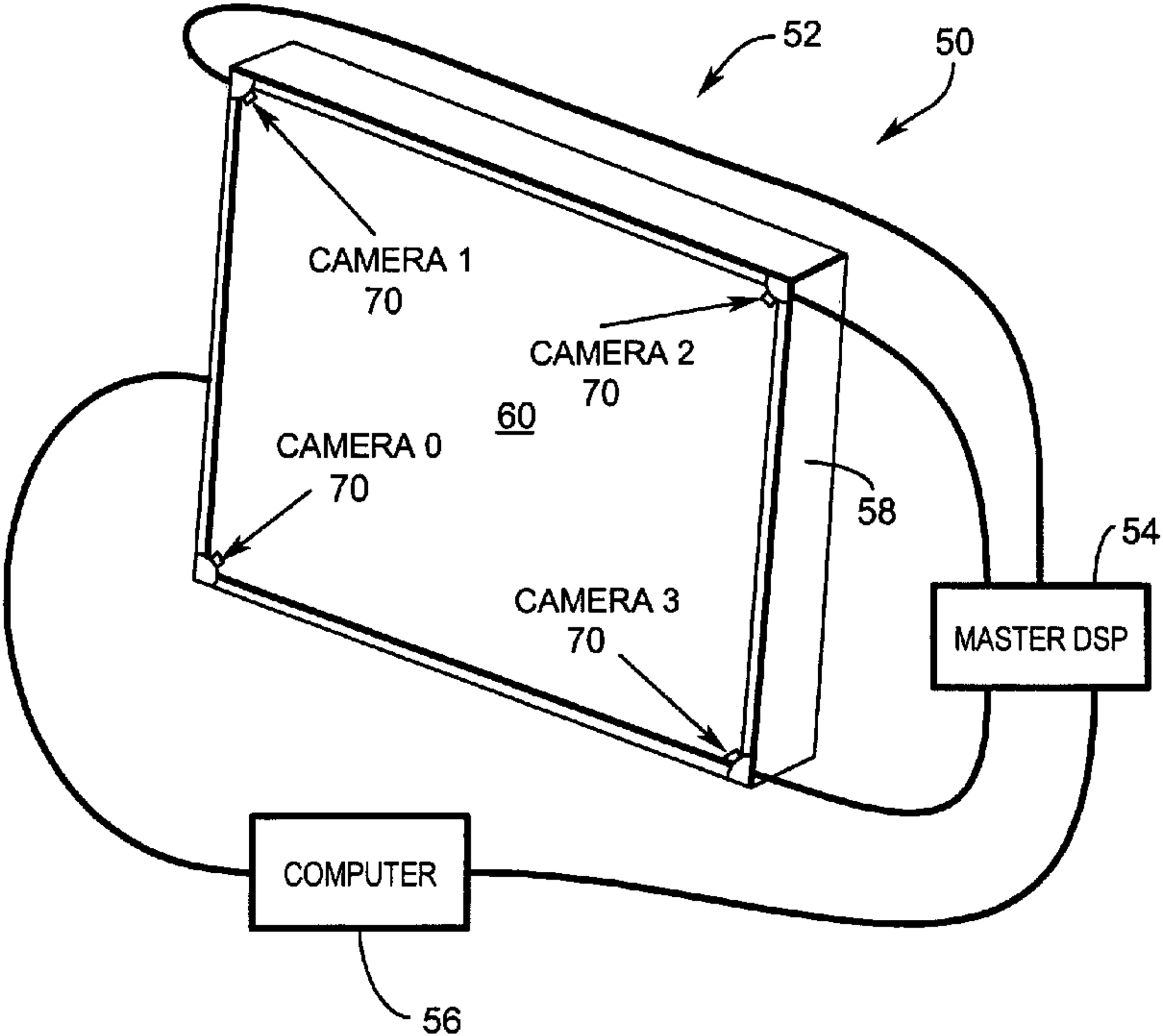


FIG. 4f


52
50
CAMERA 1
70
CAMERA 2
70
60
CAMERA 0
70
58
54
CAMERA 3
70
MASTER DSP
COMPUTER
56