PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06K 11/06

A2

(11) International Publication Number: **WO 98/07112**

(43) International Publication Date: 19 February 1998 (19.02.98)

(21) International Application Number: PCT/GB97/02147

(22) International Filing Date: 8 August 1997 (08.08.97)

(30) Priority Data:
08/696,046 13 August 1996 (13.08.96) US

(71) Applicant: SYMBIOS LOGIC INC. [US/US]; 2001 Danfield Court, Fort Collins, CO 80525 (US).

(71) Applicant *(for MG only)*: GILL, David, Alan [GB/GB]; W.P. Thompson & Co., Celcon House, 289-293 High Holborn, London WC1V 7HU (GB).

(72) Inventor: BALLARE, Daniel, E.; 2287 Ptarmigan Lane, Colorado Springs, CO 80918 (US).

(74) Common Representative: GILL, David, Alan; W.P. Thompson & Co., Celcon House, 289-293 High Holborn, London WC1V 7HU (GB).

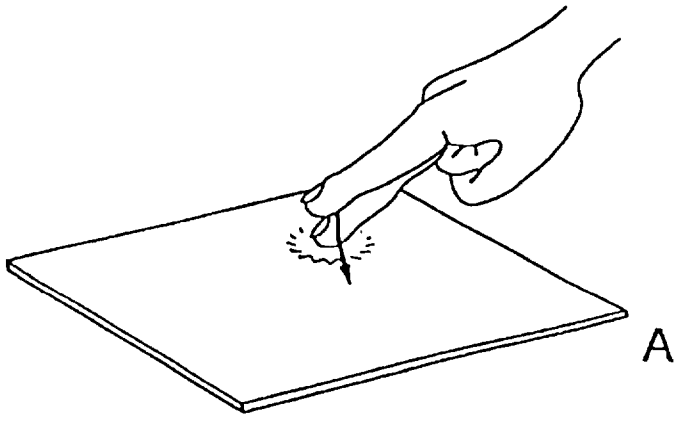
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

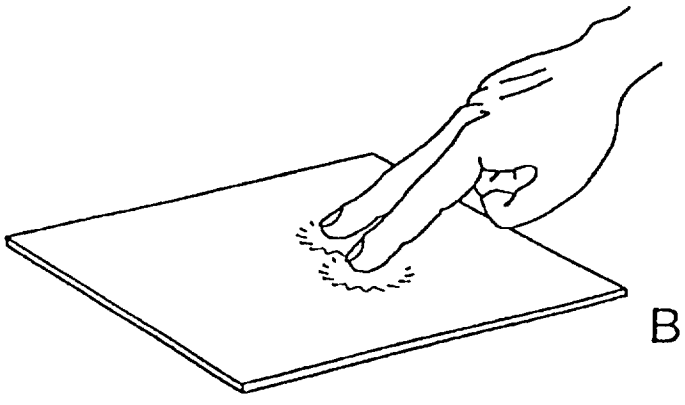
Published
Without international search report and to be republished upon receipt of that report.

(54) Title: DATA INPUT APPARATUS AND METHOD

(57) Abstract

The invention provides for a system for emulating a mouse button event via an absolute coordinate input device and which includes a computer having an operating system capable for receiving mouse type relative coordinate input commands from an absolute coordinte input device adapted to generate coordinate data in response to a finger touch down event. A controller is operably connected to the coordinate input device and adapted to resolve left and right mouse button touch down events and respective coordinate data for such events.





FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

DATA INPUT APPARATUS AND METHOD

BACKGROUND OF THE INVENTION

The present invention relates to a data input apparatus and method. In particular, but not exclusively, the invention relates to coordinate based data input devices and more particularly to a method and apparatus for emulating a supplemental mouse button, for example, a right mouse button or other non-primary feature selection button, in a touch-type coordinate based data input device.

Various methods and apparatuses are known in the art for facilitating the movement of a cursor to a point on the display of a computer or the like. Such methods and apparatuses are useful in assisting electronic device users (such as users of general purpose or dedicated computers) in selecting text, graphics, or menus for subsequent manipulation.

Coordinate based input devices include relative coordinate devices such as mice, track balls and absolute coordinate devices including digitizing tablets, touch screens, and touch pads. A mouse is a hand-held device which typically has between one and three feature buttons (physical electrical switches) for selecting different features related to or useful to a coordinate selected. The mouse buttons are typically arranged horizontally and referred to as left, right and middle mouse buttons. Digitizers and touch screens are utilized in a similar manner and most generally employ either an active or passive stylus. Where an active stylus is employed one or more buttons located on the stylus may be actuated so as to allow mouse button-type feature selection.

Features available for mouse selection can be broken down into the following categories: (a) primary features which are most commonly supported; (b) secondary features which are often supported yet are less common than primary features; and (c) tertiary features which are sometimes supported yet are less common than secondary features.

In the case of a single button mouse, primary features are the only available features. Single button mouse users typically select these features via the index finger pushing the single mouse button.

In the case of a two button mouse, both primary and secondary features are available. A typical right handed user makes primary feature selection via the index finger pushing the left mouse button. In addition, said user makes secondary feature selection via the long finger pushing the right mouse button.

In the case of a three button mouse, primary, secondary and tertiary features are available. A typical right handed user makes primary feature selection via the index finger pushing the left mouse button. In addition, said user makes secondary feature selection via the third finger pushing the right mouse button. Finally, said user makes tertiary feature selection via the long finger pushing the middle mouse button.

Note that users have the option of orienting mouse buttons differently than stated above. However, for the remainder of this document, left mouse button input will be associated with primary feature selection, right mouse button input will be associated with secondary feature selection and middle mouse button input will be associated with tertiary input selection.

Where a passive stylus is utilized other means for selecting mouse button-type features must be employed. Additionally, in absolute coordinate data input devices an initial contact is interpreted as primary mouse (e.g., left mouse button) activation. For example, US-A-5,432,531, teaches a coordinate processor capable of locking a cursor position and allowing differentiation between cursor movement from button click commands without requiring a manually operable push button. However, US-A-5,432,531 does not disclose means for issuing relative coordinate device right mouse button commands.

Systems capable of emulating relative coordinate device right mouse button commands have been disclosed. For example, US-A-5,491,495, teaches a system for simulating a mouse by displaying a mouse icon overlay so as to allow a user to select a coordinate with a stylus and then select a mouse button by moving the stylus to the mouse icon overlay. However, US-A-5,491,495 utilizes relative coordinate location of the stylus input based upon a graphical simulation of a mouse. Therefore, US-A-5,491,495 requires (a) additional processor time to process the mouse icon display, (b) an icon which covers a viewable sized area of the display, and (c) a user to drag the icon into position rather than allowing a user to simply point and act on the coordinate desired.

SUMMARY OF THE INVENTION

The present invention teaches right mouse button emulation in conjunction with an absolute coordinate input device. The present invention discloses a system for receiving input from fingers and for providing right mouse button emulation input to a program capable of receiving

input from a mouse. The system includes a touch screen adapted to receive input from a finger for generating coordinate data in response to a finger, a right mouse button selector in association with the touch screen for determining whether a user has selected a right mouse button feature. Right mouse button selection is determined by either the number of touch events on the touch screen or by the corresponding signal strength produced by the number of fingers touching the touch screen.

According to one aspect of the invention there is provided a data input system including means for receiving mouse-related input signals, the system comprising:

a display screen;

coordinate input means operably associated with said display screen for generating coordinate data in response to touch of a finger;

controller means connected to said display screen and said coordinate input device, said controller means adapted to resolve mouse button events and respective coordinate data for such events; and

an output operably connected to said controller for providing coordinate and mouse button selection data to said means for receiving a mouse-related input.

According to another aspect of the invention there is provided a data input system providing mouse button emulation input to a program capable of receiving input from a mouse, the system comprising:

display means;

coordinate input means adapted to receive input from a finger for generating coordinate data in response

to the touch of a finger;

mouse button emulator means associated with said coordinate input means for determining whether a user has selected a mouse button feature by determining the number of fingers touching said coordinate input means; and

output means for providing finger coordinate and mouse button selection data to said program.

According to yet another aspect of the invention there is provided a data input method for emulating mouse-related input signals within a computer arranged to operate with a program capable of receiving input from a mouse and having a touch screen, comprising the step of determining mouse button selection via the number of fingers touching said touch screen.

The present invention is advantageous in providing a touch screen and touch pad users with full mouse like input and also a system which allows right mouse emulation with touch screens and touch pads. Further, it is another advantage of the present invention that such a system which performs without degrading computer performance can be provided and in particular one which allows both point as well as drag-and-drop selection. Preferably the present invention can provide a system which does not impede the viewable area of a display during operation.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is described further hereafter, by way of example only, with reference to the accompanying drawings in which:

FIG. 1 is a diagrammatic illustration of a coordinate-based touch-input system embodying the present invention;

FIGS. 2A and 2B are diagrammatic perspective views illustrating a system according to an embodiment of the present invention;

FIG. 3 is a diagrammatic perspective view illustrating a system according to another embodiment of the present invention;

FIG. 4 is a flow chart illustrating the basic operation of a preferred embodiment of the present invention;

FIG. 5 is a flow chart illustrating the basic operation of the embodiment illustrated in FIGS. 2A and 2B wherein touch counting is utilized; and

FIG. 6 is a flow chart illustrating the basic operation of the embodiment illustrated in FIG. 3.

DETAILED DESCRIPTION

The following co-pending and co-owned United States Patent Applications are incorporated herein by reference: (1) *Cancellation of Common-Mode Signals in Digitizing Tablet*, United States Patent Application 08/192,078 (filed February 4, 1994); and (2) *Compensation of Stylus Signals in Digitizing Tablet*, United States Patent Application 08/286,720 (filed August 5, 1994).

In an exemplary embodiment right mouse button emulation may be provided in a system including, for example, the SYM93C2000 WriteTouch™ Interface Controller. This controller is available from Symbios Logic, Colorado Springs, Colorado. The WriteTouch™ system is capable of finger touch events by biasing an active indium tin oxide sensor with a voltage and measuring the additional currents required to feed the added finger capacitance. Differences in corner currents reveals the location of the extra charge required directly under the finger. The sensor panel

includes a glass writing surface with an under coating of indium-tin-oxide (ITO). Preferably a polyester spall shield is attached over the ITO coating in order to prevent shattering if the glass is broken. The underside of the shield is also coated with a grounded ITO layer in order to shield the sensor from LCD electrical noise. The top layer of the glass writing surface is provided with a grounded fine grid ITO coating pattern in order to eliminate hand effects. The active ITO layer is slightly resistive and divides the finger current among four corner wires. The corner current signals carried by these wires are proportional to the ITO conductivity between each individual corner and the finger location.

The corner current signals are amplified, filtered and rectified (for example, as illustrated in FIG. 1; C1, C2, C3, and C4). Finger location is then determined by subtracting the normalized corner currents in pairs in order to separately obtain *x* and *y* coordinates. A controller may then calibrate the *x*, *y* coordinates using data preferably specific to each sensor panel (this removes edge effects and non-linearities due to ITO coating variations). Sensor-specific calibration data may be maintained in non-volatile RAM (NVRAM) preferably in the panel cable rather than the controller so as to allow panel replacement. A single serial data line may be utilized to connect the controller to the CPU. A software driver may be utilized to process the data from the controller and, if appropriate, provide the data to an application program to determine whether to use the data to display digital ink via an LCD graphics controller or to use the data as control information.

While the WriteTouch™ system provides a currently

preferred implementation of the system of the present invention, it will be appreciated, a system of the present invention may be assembled which utilizes a membrane, infrared, electromagnetic, or resistive based touch screen sensors or the like.

The source code of a driver program for use with Microsoft Windows 95 is included at the end of this description. This program, by way of example only, may be utilized to implement the system of the present invention where a general purpose computer running Windows 95 is utilized with the WriteTouch™ system. It will be appreciated that the present invention may be implemented with other operating systems and, likewise, with dedicated computers.

In operation, an exemplary embodiment of the present invention may be constructed with a computer, for example, having an IBM PC type architecture and a touch screen connected to the computer as an I/O device. The touch screen system may be, for example, a Symbios Logic™ SYM93C2000 controller, a Scriptel Corporation (Columbus, Ohio) electrostatic sensor panel for the WriteTouch™ system, and a software driver to interface the SYM93C2000 to an operating system such as Microsoft Windows 95. Operation of the SYM93C2000 (WriteTouch Interface Controller, herein after "WTIC"). The computer, for example, may operate application environment software such as Microsoft WinPad in conjunction with an additional driver such as set forth in *Appendix A* at the end of this description.

The source code driver program described later (written in C programming language) allows right mouse button emulation via touch counting (FIGS. 2A, 2B, and 5). With

firmware and driver alterations right mouse emulation may be obtained via touch strength (FIGS. 3, 4, and 6). In either embodiment a user of the system of the present invention may access full mouse type input features, including the hereto before unavailable right mouse button features, with a touch screen.

Where touch counting is to be monitored by the system (FIG. 5) the driver of an exemplary embodiment may first cause the WTIC to send a packet to the host computer. The system then determines whether a touch down event has occurred. The system also determines whether a touch lift event has occurred. A counter is incremented for each touch down event. The counter is reset to zero for each touch lift event. If a touch down event causes the counter to exceed a count of one, then right mouse button data is sent to the operating system (O/S) along with *x*, *y* data from the previous touch down event.

Where touch strength is to be monitored by the system (FIG. 6) the firmware may be modified to monitor touch strength from the relative strength and proximity of the output of the touch screen. In such an embodiment the WTIC sends a new packet to the host and determines whether a touch down event has occurred. If a touch down event has occurred a determination is then made as to whether the touch down event included more than one finger in proximity to another then new *x*, *y* data is calculated and relayed to the operating system along with right mouse button switch data.

Thus, there has been described by way of example two embodiments of the present invention which obtain at least all of the objects of the present invention. It will,

however, be appreciated the present invention is equally applicable to other absolute position pointing devices, operating systems, and computers.

There now follows by way of example only, an illustration of the source code arranged for use within a method, system or apparatus embodying the present invention.

APPENDIX "A"

SOURCE CODE PROVIDED UNDER 37 CFR 1.96(A)(2)(I)

```
[[ PATENT.C : 4125 in PATENT.C ]]
  static int TouchCount;
  static BOOL MarkSpot;
  static int MarkBarrelX;
  static int MarkBarrelY;
  static int EmulateBarrel;
DWORD BuildTouchPkt(LPDRV_PENPACKET lppp) // Handles normal touch mode.
{
 static WORD xyunlocked = 0;
 static WORD touchDown;
 DWORD ret_val;
    ret_val = FF_SUCCESS | FF_PENPACKET;
    if(ByteBuffer[Z_HIGH_BYTE] & 0x10) // touch down event occurred.
    {
      if(BeepEnabled)
       BeepOn(0x100);
      PenStatus &= ~0x1 ;
      touchDown = 1;
      markX = (int)last_X;
      markY = (int)last_Y;
      TouchCount++;
      if (TouchCount > 1)
```

```
        {
         TouchCount = 1;
         if(BeepEnabled)
          BeepOn(0x200);
         EmulateBarrel = 1;
        }
        else
        {
         MarkSpot = TRUE;
         if(BeepEnabled)
          BeepOn(0x100);
        }
      }
      if(ByteBuffer[Z_HIGH_BYTE] & 0x20)  // lift event occurred.
      {
       EmulateBarrel = 0;
       TouchCount = 0;
      }
      if (ByteBuffer[VALIDITY_BYTE] & 0x04)  // proximity bit set ?
      {
        // in prox.
        if (touchDown)
        {
          last_tip_state = PDK_LEFT;
          touchDown = 0;
          xyunlocked = 1;
        }
        else
        {
         if ( !(ByteBuffer[VALIDITY_BYTE] & 0x08))  // valid xy bit not set ?
           xyunlocked = 0;
```

```
        }
      }
      else  // out of prox
      {
        if (EmulateBarrel)
        {
         last_tip_state = PDK_UP;
         EmulateBarrel = 0;
        }
        else
        {
         ret_val = FF_SUCCESS | FF_OORPENPACKET;
         last_tip_state = PDK_OUTOFRANGE;
        }
        xyunlocked = 0;
      }
    if (xyunlocked)
    {
      // read fresh x,y data from buffer.
      wticX = (long)((ByteBuffer[X_HIGH_BYTE] << 7) | ByteBuffer[X_LOW_BYTE]);
      wticY = (long)(ByteBuffer[Y_HIGH_BYTE] << 7) | ByteBuffer[Y_LOW_BYTE];
     // Begin Touch Calibration
     // Caution: the following has been carefully written not to overflow
     //        32 bit signed long numbers with wticX and wticY ranging
     //        from 0-16383 and constants such as channel gains,
     //        sines and cosines  scaled up to the 10000 range.
     // first adjust
      Xadj = (wticX*JtouchCal.Scale_X)/10000L + JtouchCal.Off_X;
      Yadj = (wticY*JtouchCal.Scale_Y)/10000L + JtouchCal.Off_Y;
     // rotate
      Xrot = (Xadj*JtouchCal.Cos_B - Yadj*JtouchCal.Sin_B)/10000L;
```

```
        Yrot = (Xadj*JtouchCal.Sin_B + Yadj*JtouchCal.Cos_B)/10000L;
        calX = (int) Xrot;
        calY = (int) Yrot;
      // end calibrate
            // WTIC puts zero at bottom of screen.
                // Windows puts zero at top of screen.
                    // Ergo, we need to invert the screen....
        last_Y = (WORD)(0x3FFF-calY);
    } // end if xyunlocked
    if (EmulateBarrel == 1)
    {
     // This time, let current x,y data pass through.
     EmulateBarrel++;
    }
    else
    if (EmulateBarrel == 2)
    {
     EmulateBarrel++;
     last_tip_state = PDK_UP; // send Windows mouse button up data to
                    // cancel previous Left mouse down data.
    }
    else
    if (EmulateBarrel == 3)
    {
     last_X = MarkBarrelX;   // use x,y data from first down event.
     last_Y = MarkBarrelY;
     last_tip_state = PDK_RIGHT;  // send Windows right mouse data
    }
    if (MarkSpot)
    {
     MarkSpot = FALSE;
```

```
        MarkBarrelX = last_X;
        MarkBarrelY = last_Y;
      }
      lppp->wPDK = last_tip_state;
      lppp->wTabletX = last_X;
      lppp->wTabletY = last_Y;
      return(ret_val);
    }
```

CLAIMS

1. A data input system including means for receiving mouse-related input signals, the system comprising:

a display screen;

coordinate input means operably associated with said display screen for generating coordinate data in response to touch of a finger;

controller means connected to said display screen and said coordinate input device, said controller means adapted to resolve mouse button events and respective coordinate data for such events; and

an output operably connected to said controller for providing coordinate and mouse button selection data to said means for receiving a mouse-related input.

2. A system as claimed in Claim 1, wherein said controller means is arranged to determine whether a user has selected a mouse button feature by counting the number of finger touch down events sensed by said coordinate input means.

3. A system as claimed in Claim 1, wherein said controller means is arranged to determine whether a user has selected a mouse button feature by determining the strength of the output of said coordinate input means.

4. A system as claimed in Claim 2, wherein said controller means is arranged to resolve left and right mouse button touch down events by determining: (a) whether a touch

down event has occurred; (b) if a touch down event has occurred whether more than one touch down event has occurred; (c) whether a touch lift event has occurred; and (d) if a touch lift event has occurred whether the touch lift was in proximity to a touch down event.

5. A system as claimed in Claim 3, wherein said controller means is arranged to resolve left and right mouse button touch down events by determining at least the following: (a) whether a touch down event has occurred; (b) whether a touch lift event has occurred; (c) if a touch down event has occurred whether more than one touch down event has occurred; and (d) if a touch lift event has occurred whether the touch lift was in proximity to a touch down event.

6. A system as claimed in any one of Claims 1 to 5, wherein a left mouse button feature is accessed through an index finger touch down event on the touch screen.

7. A system as claimed in any one of Claims 1 to 6, wherein a right mouse button feature is accessed through a long finger touch down event.

8. A system of as claimed in any one of Claims 1 to 5, wherein a right mouse button feature is accessed through a combined index finger and long finger touch down event.

9. A system as claimed in Claim 1 and for providing mouse button emulation input to a program capable of receiving input from a mouse, and further comprising:

a finger sensor associated with said display screen for sensing a finger, said finger sensor including an output;

wherein the coordinate input means is associated with said finger sensor, said coordinate input means adapted to receive input from said finger sensor, and for generating coordinate data in response to a finger sensed by said finger sensor;

and including a mouse button selector associated with said finger sensor and said coordinate input means for determining whether a mouse button feature has been selected by determining the number of fingers touching said finger sensor.

10. A system as claimed in Claim 9, wherein said mouse button selector is arranged to determine whether a user has selected a mouse button feature by counting the number of fingers sensed by said finger sensor.

11. A system as claimed in Claim 9, wherein said mouse button selector is arranged to determine whether a user has selected a mouse button feature by determining strength of the output of said finger sensor.

12. A data input system providing mouse button emulation input to a program capable of receiving input from a mouse, the system comprising:

display means;

coordinate input means adapted to receive input from a finger for generating coordinate data in response to the touch of a finger;

mouse button emulator means associated with said coordinate input means for determining whether a user has selected a mouse button feature by determining the number of fingers touching said coordinate input means; and

output means for providing finger coordinate and mouse button selection data to said program.

13. A system as claimed in Claim 12, wherein said mouse button emulator means determines whether a user has selected a mouse button feature by counting the number of fingers sensed by said coordinate input means.

14. A system as claimed in Claim 12, wherein said mouse button emulator means determines whether a user has selected a mouse button feature by determining strength of the output of said coordinate input means.

15. A system as claimed in Claim 12, 13 or 14, comprising:

finger sensor means associated with said display means for sensing a finger and for providing an output;

the coordinate input means being associated with said finger sensor means and adapted to receive input from said finger sensor means, and for generating coordinate data in response to a finger sensed by said finger sensor means;

the mouse button emulator means being associated with said finger sensor means and said coordinate input means for determining whether a user has selected a mouse button feature by determining the number

of fingers touching said finger sensor means; and output means for providing finger coordinate and mouse button selection data to said program.

16. A system as claimed in Claim 12, 13 or 14, wherein the display means comprises a touch screen including a transparent sheet material having an indium-tin-oxide coating and being responsive to use by an operative's finger generating coordinate data; the mouse button emulator comprises a touch screen controller exhibiting mouse button emulation for determining whether a user has selected a mouse button feature by determining the number of fingers touching said touch screen; and including a general purpose computer connected to said touch screen via said touch screen controller; and wherein the output means is arranged for providing finger coordinate and mouse button selection data from said controller to said program operating on said general purpose computer.

17. A system as claimed in Claim 16, wherein said touch screen is arranged to sense a finger in operation with said controller by biasing said indium-tin-oxide coating of said sensor with a voltage and then measures additional currents required to compensate for finger capacitance.

18. A system as claimed in Claim 17, wherein said touch screen controller further comprises means for calibrating raw *x*, *y* data using a touch screen profile data set specific to each touch screen.

19. A system as claimed in Claim 16, 17 or 18, wherein

said touch screen controller is arranged to communicate with said computer serially.

20. A data input method for emulating mouse-related input signals within a computer arranged to operate with a program capable of receiving input from a mouse and having a touch screen, comprising the step of determining mouse button selection via the number of fingers touching said touch screen.

21. A method as claimed in Claim 20 wherein the number of fingers touching said touch screen is determined by counting touchdown events.

22. A method as claimed in Claim 20, wherein the number of fingers touching said touch screen is determined by the signal strength of the output of said touch screen.

23. A data input method as claimed in Claim 20, including:

sending a data packet to a host computer for processing, the data packet being received from the touch screen and comprising coordinate and magnitude data relating to the tactile input and a switch variable having values indicative of a left mouse button event, a right mouse button event and no mouse button event;

detecting a finger touch down event based upon the coordinate and the magnitude data;

determining the finger touch down event being produced by more than one finger based upon the magnitude

data upon detecting a finger touch down event;

setting the switch variable to a value indicating a right mouse button event upon determining the finger touch down event being produced by more than one finger;

setting the switch variable to a value indicating a left mouse button event upon determining the finger touch down event not being produced by more than one finger;

in the event that a finger touch down event is not detected,

detecting a touch lift event based upon the coordinate data;

setting the switch variable to value indicating no mouse button event upon detecting a touch lift event;

determining the proximity of the coordinate data to previous coordinate data upon a touch lift event not being detected;

setting the switch variable to a value indicating no mouse button event upon determining the coordinate data not being proximate to previous coordinate data;

thereafter, determining the switch variable being set to a value indicating no mouse button event;

calculating new coordinate data upon determining the switch variable not being set to a value indicating no mouse button event; and

otherwise repeating said sending step by relaying the coordinate and magnitude data and the switch variable to said host computer.

24. A data input method display system capable of receiving tactile input from a touch input device as claimed in Claim 20, including:

sending a data packet to a host computer for processing, the data packet being received from the touch input device and comprising coordinate and touch count data relating to the tactile input, a switch variable having values indicative of a left mouse button event, a right mouse button event and no mouse button event, and a markspot variable marking the coordinate data;

detecting a finger touch down event based upon the coordinate data;

incrementing the touch count data upon detecting the finger touchdown event;

determining the count data being greater than one;

setting the touch count data to two and the switch variable to a value indicating a right mouse button event upon determining the touch count data being greater than one;

setting the switch variable to a value indicating a left mouse button event and the markspot variable to a value marking the coordinate data of the detected finger touch down event upon determining the touch count data not being greater than one;

in the event that a finger touch down event is not detected,

detecting a touch lift event based upon the coordinate data;

setting the switch variable to a value indicating no mouse button event and setting the touch count

data to zero upon detecting a touch lift event;

determining the proximity of the coordinate data to previous coordinate data upon a touch lift event not being detected;

setting the switch variable to a value indicating no mouse button event and setting the touch count data to zero upon determining the coordinate data not being proximate to previous coordinate data;

thereafter,

determining the switch variable being set to a value indicating a left mouse button event;

calculating new coordinate data upon determining the switch variable being set to a value indicating a left mouse button event;

determining the markspot variable being set to a value marking the coordinate data of the detected finger touch down event;

storing the coordinate data and setting the markspot variable not marking the coordinate data of the detected finger touch down event upon determining the markspot being set to a value marking the coordinate data of the detected finger touch down event;

in the event the switch variable is determined to be not set to a value indicating a left mouse button event,

determining the switch variable being set to a value indicating a right mouse button event;

restoring storing previous coordinate data upon determining the switch variable being set to a value indicating a right mouse button event; and

otherwise repeating said sending step by relaying the coordinate and touch count data and the switch and markspot variables to said host computer.

25. Data input apparatus comprising:

(a) a touch screen capable of sensing a finger;

(b) a computer operably connected to said touch screen, and arranged for operation with:

(i) a first program for receiving input from a mouse;

(ii) a second program for generating coordinate and related mouse data from fingers touching said touch screen;

(c) a controller operably connected to said touch screen and said computer, and adapted to resolve left mouse button and right mouse button touch down events and respective coordinate data for such events.

26. Apparatus as claimed in Claim 25, wherein said number of fingers touching said touch screen is resolved by counting touchdown events.

27. Apparatus as claimed in Claim 25, wherein the number of fingers touching said touch screen is resolved by signal strength of the output of said touch screen.

28. Apparatus as claimed in Claim 25, 26 or 27, wherein said computer is a general purpose computer.

29. Apparatus as claimed in Claim 25, 26, 27 or 28, wherein said computer and said first program operate in conjunction with an operating system producing an operating

environment for operating said second program.

30. Apparatus as claimed in any one of Claim 25 to 29, wherein said touch screen includes an electrostatic sensor.

31. Apparatus as claimed in Claim 30, wherein said touch screen includes an indium-tin-oxide coating.

32. Apparatus as claimed in Claim 30, wherein said electrostatic sensor is biased with an AC signal.

33. Apparatus as claimed in Claim 32, wherein a finger is detected by said touch screen by measuring additional currents resulting from the added load of a finger touch on said touch screen.

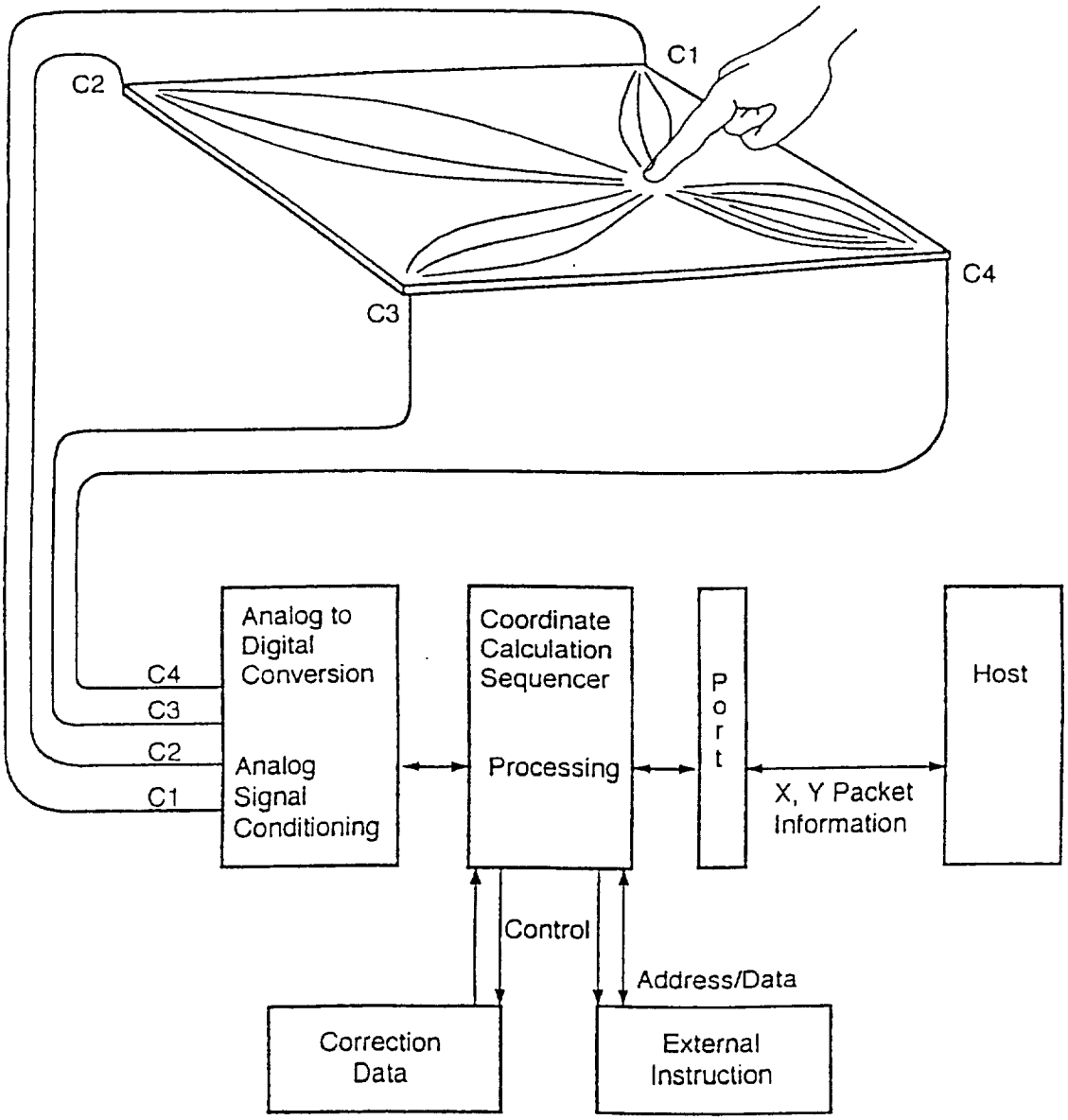

C1
C2
C3
C4
Analog to Digital Conversion
Analog Signal Conditioning
C4
C3
C2
C1
Coordinate Calculation Sequencer
Processing
Port
X, Y Packet Information
Host
Control
Address/Data
Correction Data
External Instruction

Fig. 1

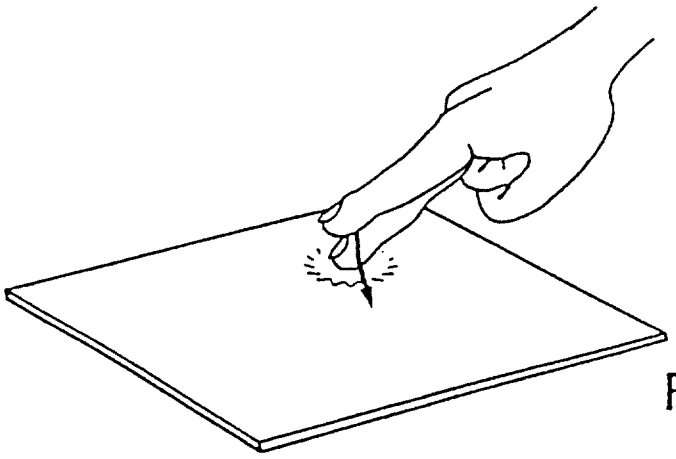


Fig. 2A

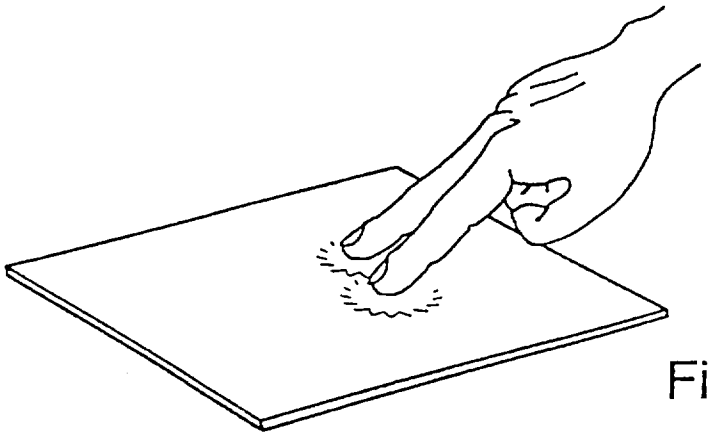


Fig. 2B

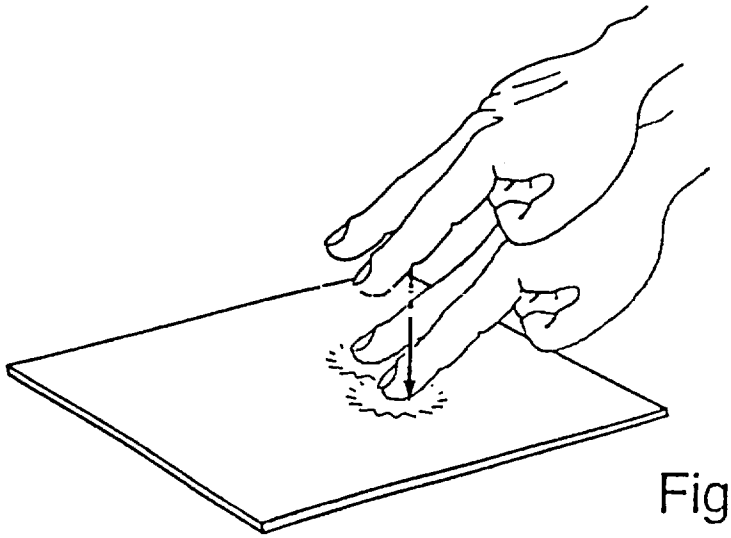


Fig. 3

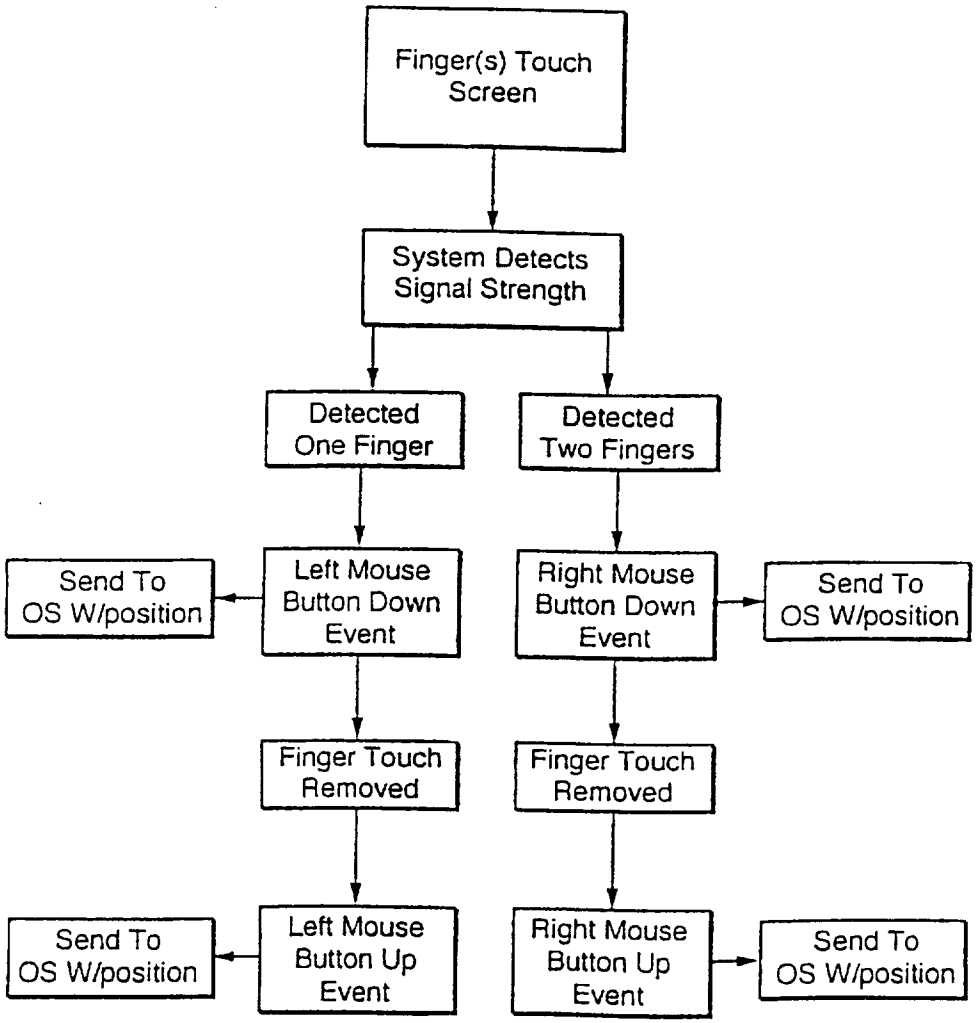

Finger(s) Touch Screen
System Detects Signal Strength
Detected One Finger
Detected Two Fingers
Send To OS W/position
Left Mouse Button Down Event
Right Mouse Button Down Event
Send To OS W/position
Finger Touch Removed
Finger Touch Removed
Send To OS W/position
Left Mouse Button Up Event
Right Mouse Button Up Event
Send To OS W/position

Fig. 4

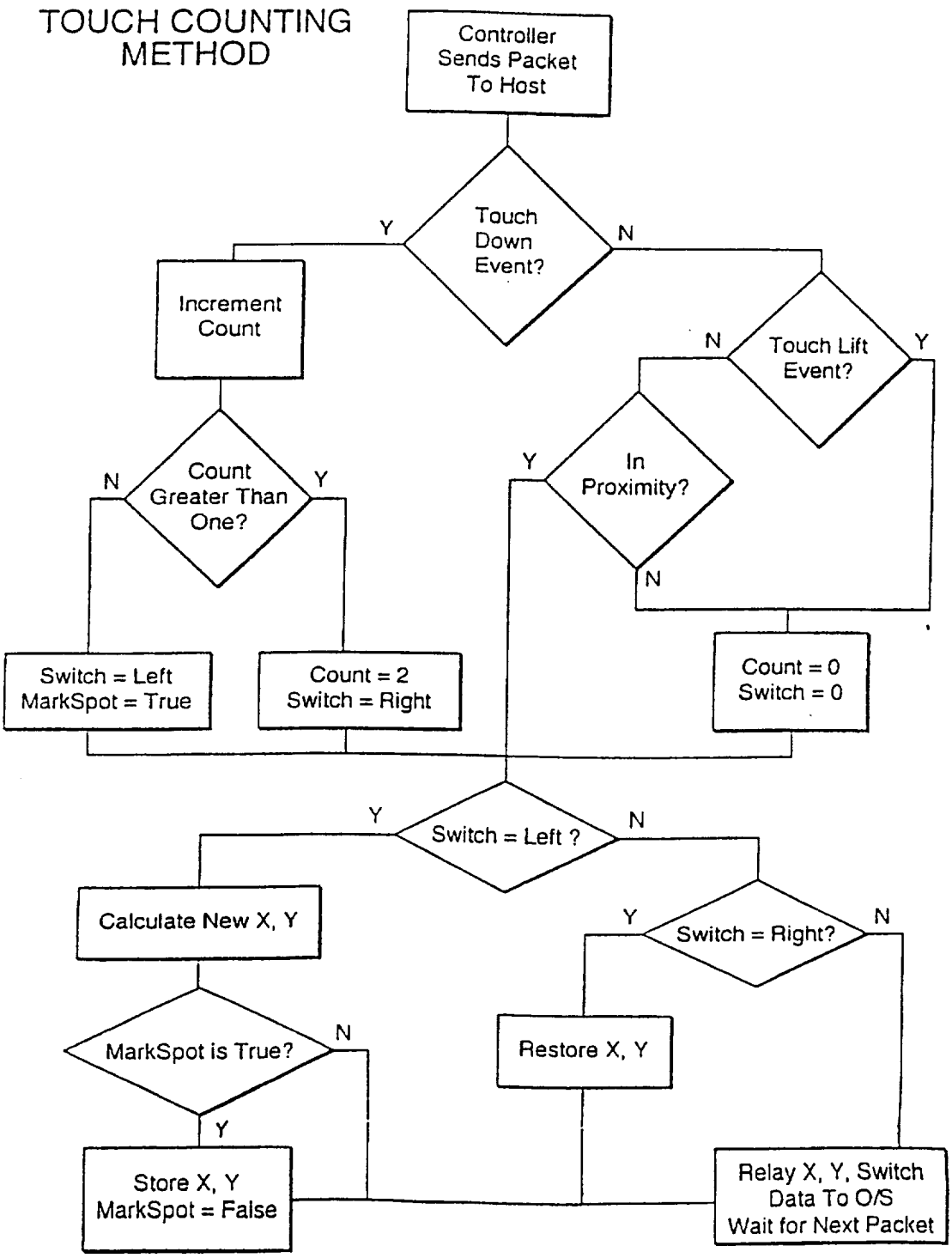

TOUCH COUNTING METHOD
Controller Sends Packet To Host
Touch Down Event?
Y
N
Increment Count
Touch Lift Event?
N
Y
In Proximity?
Y
N
Count Greater Than One?
N
Y
Switch = Left MarkSpot = True
Count = 2 Switch = Right
Count = 0 Switch = 0
Switch = Left ?
Y
N
Calculate New X, Y
Switch = Right?
Y
N
MarkSpot is True?
N
Y
Restore X, Y
Store X, Y MarkSpot = False
Relay X, Y, Switch Data To O/S Wait for Next Packet

FIG. 5

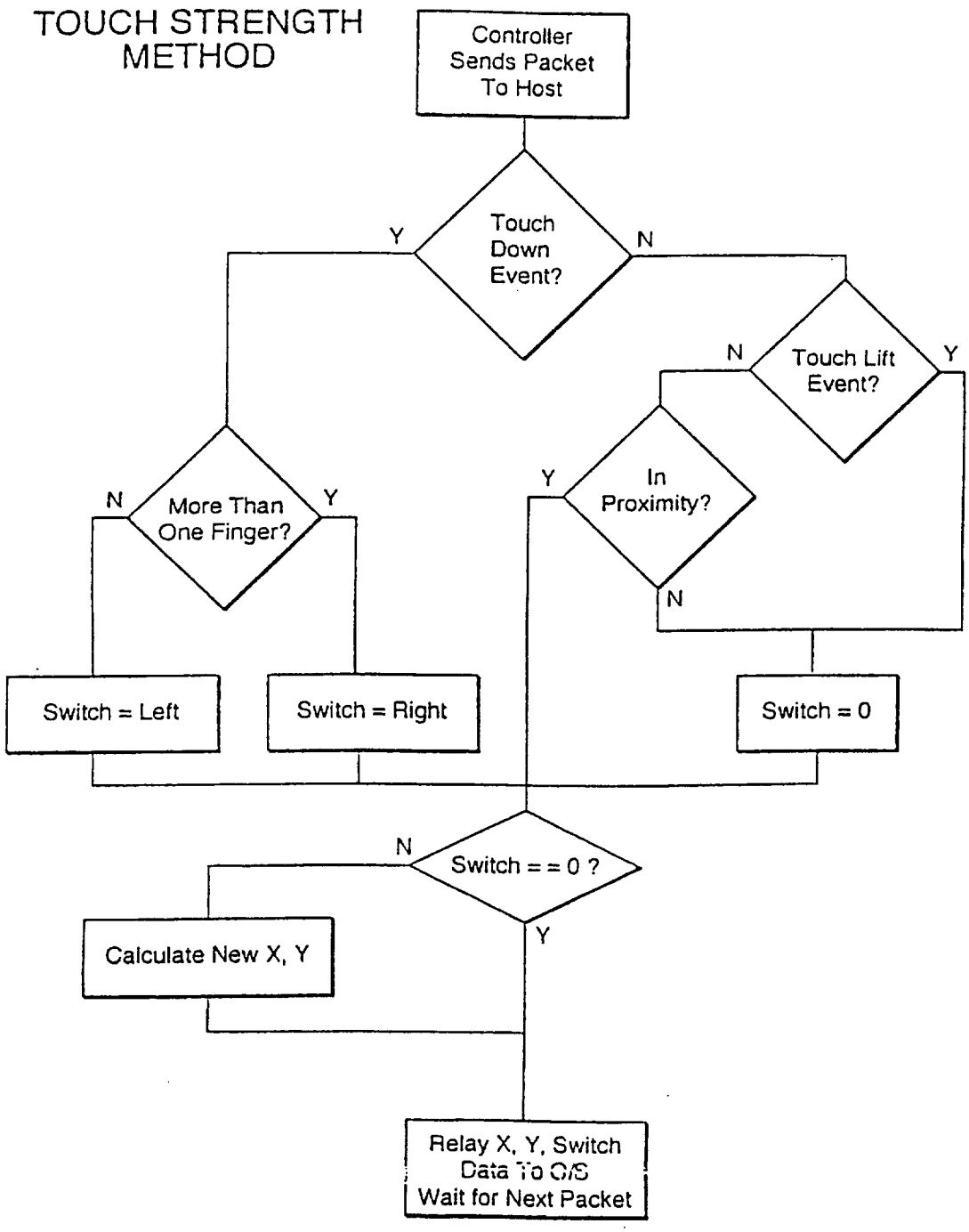

TOUCH STRENGTH METHOD
Controller Sends Packet To Host
Touch Down Event?
Y
N
Touch Lift Event?
N
Y
In Proximity?
Y
N
More Than One Finger?
N
Y
Switch = Left
Switch = Right
Switch = 0
Switch = = 0 ?
N
Y
Calculate New X, Y
Relay X, Y, Switch Data To O/S Wait for Next Packet

FIG. 6