



# (12)发明专利申请

(10)申请公布号 CN 107430430 A

(43)申请公布日 2017. 12. 01

(21)申请号 201480084321.2

(74)专利代理机构 中国专利代理(香港)有限公司 72001

(22)申请日 2014.12.22

代理人 李雪娜 张涛

(85)PCT国际申请进入国家阶段日  
2017.06.22

(51)Int.Cl.

G06F 3/01(2006.01)

(86)PCT国际申请的申请数据

PCT/US2014/071797 2014.12.22

(87)PCT国际申请的公布数据

W02016/105329 EN 2016.06.30

(71)申请人 英特尔公司

地址 美国加利福尼亚州

(72)发明人 G·任 L·M·马 H·任

A·库马尔 J·J·瓦拉维

J·M·皮卡多来瓦 K·东格尔

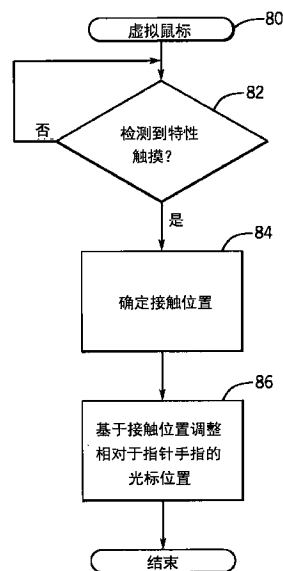
权利要求书2页 说明书7页 附图8页

(54)发明名称

多触摸虚拟鼠标

(57)摘要

根据一些实施例,通过同时利用多于一个手指触摸屏幕,可在鼠标模式中操作触摸输入设备,诸如触摸屏或跟踪板或触摸板。在一个实施例中,可利用三个手指。在一个实施例中三个手指可以是拇指连同食指和中指。然后食指和中指可被用来左击或右击以进入虚拟鼠标命令。



1. 一种计算机实施的方法,其包括:  
检测触摸输入设备上的接触;  
确定所述接触的位置;以及  
在基于所述接触的位置而变化的相对于所述接触的定位处显示光标。
2. 根据权利要求1所述的方法,包括响应于所述接触朝向屏幕边缘移动,将光标从相对于所述接触更中心的第一定位移动至所述接触的较不中心的第二定位。
3. 根据权利要求1所述的方法,包括基于到屏幕边缘的接近度来关于所述接触移动所述光标。
4. 根据权利要求1所述的方法,包括使用供应方独立内核来实现独立于触摸供应方内核进行操作的机制。
5. 根据权利要求1所述的方法,包括在初始化期间加载所述供应方独立内核,在不依赖于任何平台操作系统的情况下在图形处理单元上运行所述供应方独立内核。
6. 根据权利要求1所述的方法,包括通过虚拟鼠标设备对象使鼠标输入事件暴露于操作系统。
7. 根据权利要求6所述的方法,包括使用内核模式驱动器来创建虚拟鼠标设备对象。
8. 根据权利要求1所述的方法,包括检测输入设备是处于触摸模式还是处于虚拟鼠标模式,每个模式与不同人类接口设备分组相关联。
9. 根据权利要求8所述的方法,包括筛选出未被检测到的模式的分组。
10. 根据权利要求1所述的方法,包括使用用于实施虚拟鼠标模式的驱动器。
11. 一个或多个非瞬时计算机可读介质,其存储指令,所述指令被执行来实行包括以下的序列:  
检测触摸输入设备上的接触;  
确定所述接触的位置;以及  
在基于所述接触的位置而变化的相对于所述接触的定位处显示光标。
12. 根据权利要求11所述的介质,所述序列包括响应于所述接触朝向屏幕边缘移动,将光标从相对于所述接触更中心的第一定位移动至所述接触的较不中心的第二定位。
13. 根据权利要求11所述的介质,所述序列包括基于到屏幕边缘的接近度来关于所述接触移动所述光标。
14. 根据权利要求11所述的介质,所述序列包括使用供应方独立内核来实现独立于触摸供应方内核进行操作的机制。
15. 根据权利要求11所述的介质,所述序列包括在初始化期间加载所述供应方独立内核,在不依赖于任何平台操作系统的情况下在图形处理单元上运行所述供应方独立内核。
16. 根据权利要求11所述的介质,所述序列包括通过虚拟鼠标设备对象使鼠标输入事件暴露于操作系统。
17. 根据权利要求16所述的介质,所述序列包括使用内核模式驱动器来创建虚拟鼠标设备对象。
18. 根据权利要求11所述的介质,所述序列包括检测输入设备是处于触摸模式还是处于虚拟鼠标模式,每个模式与不同人类接口设备分组相关联。
19. 根据权利要求18所述的介质,所述序列包括筛选出未被检测到的模式的分组。

20. 根据权利要求11所述的介质,所述序列包括使用用于实施虚拟鼠标模式的驱动器。

21. 一种装置,其包括:

处理器,其用来检测触摸输入设备上的接触,确定所述接触的位置,以及在基于所述接触的位置而变化的相对于所述接触的定位处显示光标;以及

耦合至所述处理器的存储装置。

22. 根据权利要求21所述的装置,所述处理器用来响应于所述接触朝向屏幕边缘移动而将光标从相对于所述接触更中心的第一定位移动至所述接触的较不中心的第二定位。

23. 根据权利要求21所述的装置,所述处理器用来基于到屏幕边缘的接近度来关于所述接触移动所述光标。

24. 根据权利要求21所述的装置,所述处理器用来使用供应方独立内核来实现独立于触摸供应方内核进行操作的机制。

25. 根据权利要求21所述的装置,所述处理器用来在初始化期间加载所述供应方独立内核,在不依赖于任何平台操作系统的情况下在图形处理单元上运行所述供应方独立内核。

## 多触摸虚拟鼠标

### 技术领域

[0001] 这一般涉及使用鼠标命令来控制触摸屏光标。

### 背景技术

[0002] 在常规基于处理器的系统(诸如膝上型计算机、台式计算机、蜂窝电话、诸如游戏设备的媒体播放设备和其他这样的设备)中,触摸屏输入的鼠标命令提供对键盘或鼠标输入的光标命令的使用的替换。例如,鼠标命令可被用来移动光标以便在显示屏上做出选择。常规地,鼠标被拿在用户的手中并且鼠标的移动使光标移动。点击鼠标上的按钮实现对被光标覆盖的显示对象的选择。

[0003] 在一些情况下,移动用户可能会发现鼠标的使用不方便,因为它需要携带可能比实际基于处理器的设备(诸如蜂窝电话)更大的附加设备。而且,在小屏幕设备(诸如在蜂窝电话中发现的那些)的情况下,可能不存在足够的屏幕空间来选择显示在屏幕上的某些更小的特征。另一问题是在显示屏上的小图标按钮或链接的情况下,对于用户而言可能难以将鼠标光标准确地放置在特定位置处。

### 附图说明

[0004] 关于以下各图来描述一些实施例:

[0005] 图1是根据一个实施例的用户的右手在显示屏上的顶视图;

[0006] 图2是根据一个实施例的用户的右手在显示屏上的顶视图;

[0007] 图3是根据一个实施例的用户的指针手指在显示屏的中心处的顶视图;

[0008] 图4是根据一个实施例的用户的右手击在显示屏的左侧上的顶视图;

[0009] 图5是根据一个实施例的用户的右手在显示屏的右侧上的顶视图;

[0010] 图6是根据一个实施例的用户的右手在显示屏的底部中心上的顶视图;

[0011] 图7是根据一个实施例的显示屏的底部左边缘的顶视图;

[0012] 图8是根据一个实施例的用户的右手在显示器的底部右边缘上的顶视图;

[0013] 图9是根据一个实施例的左鼠标点击操作的顶视图;

[0014] 图10是根据一个实施例的右鼠标点击操作的顶视图;

[0015] 图11是根据一个实施例的筛选器(filter)的示意性描述;

[0016] 图12是根据一个实施例的筛选器驱动器架构的示意性描述;

[0017] 图13是根据一个实施例的图12的筛选器驱动器的示意性描述;

[0018] 图14是根据一个实施例的用于筛选器驱动器状态机的流程图;

[0019] 图15是根据一个实施例的用户激活虚拟鼠标模式的顶视图;

[0020] 图16是根据一个实施例的用户开始光标移动命令的顶视图;

[0021] 图17是根据一个实施例的用户的右手在光标移动命令的过程中的顶视图;

[0022] 图18A是根据一个实施例的左鼠标点击操作的顶视图;

[0023] 图18B是根据一个实施例的右鼠标点击操作的顶视图;以及

[0024] 图19是用于一个实施例的流程图。

### 具体实施方式

[0025] 筛选器可被插入用于触摸手势识别的触摸输入流中。然后在一些实施例中该流可被切换至鼠标模拟器。然而,这些概念还可被扩展至其他输入/输出设备。例如,音频输入流中的筛选器可被用于语音识别,但是然后可将该流切换至执行语音检测变换的键盘模拟器。因此以下在触摸输入流的上下文中的示例不应该被视为限制本公开的范围。

[0026] 在一个实施例中,触摸屏可以以不同模式来操作。在正常模式中,该屏幕对单个手指、多个手指和笔/触笔(stylus)以及如由操作系统定义的所有相关手势进行响应。在检测到具体手势(下面定义的)时,触摸屏进入虚拟鼠标模式。在该模式中,正常触摸响应被禁用,并且触摸屏像虚拟鼠标/触摸板那样起作用。在一个实施例中,当所有手指被抬起时,触摸屏立即返回到正常模式。

[0027] 如本文所使用的,触摸输入设备是检测触摸输入设备的多个手指的多触摸输入设备。

[0028] 在一个实施例中,为了开始虚拟鼠标模式,用户使用三手指手势,利用任何三个手指来触摸屏幕,如图1中所示的。在一个实施例中,用户保持该手势达几毫秒。所述手指中的一个(称为指针手指)控制鼠标光标。在开始三手指手势时,指针手指是手指P,其在触摸屏的三个手指的中间(通过将三个手指的位置的x值进行比较而获得)。指针手指高于其他手指的至少一个,以使得光标C易于被用户可见。用户保持指针手指在屏幕上以保持虚拟鼠标模式中。

[0029] 用户可以通过简单地绕屏幕移动指针手指来移动光标。光标以稍远的距离被定位于指针手指周围,以确保它对用户是可见的,并且以使得它也看起来被连接至指针手指。其确切位置取决于屏幕中的指针手指的位置。

[0030] 将手指接触用作鼠标输入的一个问题是使得光标能够覆盖整个屏幕。这包括边缘和角落。所以,取决于光标在屏幕的哪个部分中,光标被动态地移动至相对于指针手指的不同位置。如果指针手指在屏幕的中心上方(如图2中所示),则光标C被定位在想象中的半椭圆E中心上方,该想象中的半椭圆E在其中心处具有指针手指P。

[0031] 取决于指针手指沿着跨屏幕延伸的x轴的位置,光标被定位在椭圆周围的不同点处,在图2-8中利用圆圈D来表示指针手指的触摸点。当指针手指在屏幕的中心时,如图3中所示光标C被定位在D处的指针手指上方。当指针手指靠近屏幕的左边缘时,如图4中所示光标在指针手指的左侧沿椭圆定位。当指针手指靠近右边缘时,如图5中所示光标在指针手指的右侧沿椭圆定位。如果指针手指在屏幕的中心下方,则如上文所述的那样定位光标,除了将y偏移( $Y_0$ )添加至光标位置的y值之外。这允许光标C到达屏幕的底部,如图6中所示。

[0032] y偏移的值取决于沿着y轴从指针手指到屏幕中心的距离。当指针手指在上面提到的情况之间移动时,光标围绕半椭圆且跨该半椭圆平滑地移动。该方法允许光标在不跳动的情况下到达屏幕中的任何地方(包括角落),如图7和8中所示。在图7中,指针手指在屏幕的底部左侧部分处。在图8中,指针手指在屏幕的底部右侧位置中。

[0033] 当处于虚拟鼠标模式时,用户可以利用除了指针手指之外的任何手指来执行左击和右击。由用户的拇指下面的同心圆圈E所指示的,在指针手指的左侧上的任何触摸被视为

左击,如图9中所示的。由在用户的中指下面的同心圆圈F所指示的,在指针手指的右侧上的任何触摸被视为右击,如图10中所示的。触摸并保持被视为鼠标按钮向下,并且释放被视为鼠标按钮向上。用户可以通过从屏幕释放指针手指或通过利用任何手指进行四个或更多触摸来返回至触摸模式(退出虚拟鼠标模式)。

[0034] 图11中所示的架构10在图形引擎或图形处理单元核12上执行触摸数字处理。在一些实施例中,这允许以更好性能和可缩放性来运行触摸处理算法。在初始化期间被加载的图形内核中实施触摸处理算法。在一个实施例中,以OpenCL代码14编写(write)这些内核。

[0035] 在流式触摸数据上执行内核序列。触摸集成电路(IC)18供应方提供内核(算法)20来处理来自触摸传感器16的原始触摸数据以在图形处理单元(GPU)12中产生最终触摸X-Y坐标。该数据然后转到操作系统22作为标准触摸人类接口设备(HID)分组24。

[0036] 该架构允许附加的后处理内核的链接,其可以在触摸数据到达OS之前进一步处理该触摸数据。

[0037] 如图11中所示,在后处理内核26中实施虚拟鼠标。该后处理解决方案允许在不考虑供应方内核的情况下运行通用算法。因为在不考虑触摸IC供应方的情况下运行后处理内核,所以它们是独立硬件供应方(IHV)不可知的。为了统一数据格式中的供应商差异,配置数据使跨所有触摸IC供应商的数据对准。因为该固件是在初始化期间加载的,在GPU上运行,并且对操作系统不具有任何依赖,所以它也是操作系统供应方(OSV)独立的。

[0038] 后处理内核遵循链式执行模型,其允许数据从一个内核流到下一内核,从而允许内核在经先前处理的数据上执行。每个内核可被用来适应于特定操作系统或触摸控制器。作为配置的部分,由用户来指定内核的位置。在硬件上运行的能力允许这些算法在不提高(bring up)软件堆栈的情况下运行。后处理内核与供应方内核同时运行,这解除了对拷贝数据或运行后处理内核的任何外部干预的需要。除了虚拟鼠标功能之外,还可以在后处理中实施手势和触摸数据筛选。

[0039] 触摸控制器18获取原始传感器数据并将其转换成干净(clean)、数字触摸点信息,其可以被内核、OS或应用使用。该数据被作为触摸HID分组24发送。在转到OS之前,如上文提到的那样,HID分组经过在GPU上运行的内核序列20。

[0040] 虚拟鼠标内核或触摸/鼠标切换器30像状态机那样表现。它保持内部状态,其存储虚拟鼠标的状态(开或关)和与光标的位置有关的其他信息。

[0041] 虚拟鼠标内核26作为输入获取HID分组流并且执行手势识别25以检测用于开始虚拟鼠标模式的手势。当不处于虚拟鼠标模式时,内核的输出是触摸HID分组24。当处于虚拟鼠标模式时,触摸HID分组被切换器30阻止并且内核的输出是鼠标HID分组32。触摸HID分组或鼠标HID分组被传递给OS 22,其不知道关于切换器30中分组的筛选。然后OS基于应用(APPS)34来处理鼠标和触摸模式。

[0042] 考虑到屏幕上的指针手指的位置而计算针对鼠标的正确坐标的算法被构建在内核26中。

[0043] 实施虚拟鼠标的替换方式是通过驱动器进行触摸数据处理和触摸筛选。手势识别算法和触摸HID分组的筛选将非常类似于上文所述的那些。然而,通过驱动器来做这个将使得算法是OS依赖的。OS依赖涉及与OS供应方协调来实施虚拟鼠标特征。

[0044] 为了使得虚拟鼠标使用对用户甚至更直观,当系统处于虚拟鼠标模式时可显示鼠

标的光透明叠加图像。如果用户将手指放在靠近屏幕的左侧上,则使用触摸悬停能力来检测它,并且光透明图像出现在触摸点附近,向用户暗示该触摸将导致左击。类似地,不同的图像将在手指变得更靠近时出现在右侧上。

[0045] 在上面的替换实施方式中,系统一进入虚拟鼠标模式(即不依赖于悬停能力),叠加图像就指示左击区域和右击区域。

[0046] 作为使用整个屏幕用于虚拟鼠标的替换,较小的透明矩形可出现并像虚拟触摸板那样起作用。该触摸板将被叠加在OS或应用正显示的内容上。用户使用该触摸板来控制鼠标,就像它是物理触摸板一样。还可以提供虚拟左按钮和右按钮。

[0047] 因为虚拟鼠标不区分被用于左击和右击的手指,所以还可能使用双手。当系统进入虚拟鼠标模式时,右手指针手指可以被用来移动光标,并且人可以利用左手来完成左击。这还可以被用于点击并拖动。用户可以利用指针手指光标来选择项目,使用左手来点击,并且将它保持在屏幕上而同时围绕屏幕移动右手指针手指来进行拖动操作。

[0048] 该算法还考虑惯用右手的人和惯用左手的人。该算法基于三手指手势来检测它以进入虚拟鼠标模式。对于惯用左手的人的手指的定位不同于对于惯用右手的人的手指的定位。这是相对于现今如何处理物理鼠标的改进。用户必须作出选择(Windows控制面板)来将鼠标设置用于惯用右手或惯用左手的使用。在虚拟鼠标的情况下这可在运行中被处理。

[0049] 根据使用设备驱动器的一个实施例,内核模式驱动器创建虚拟鼠标设备来与操作系统(OS)对接以便从触摸面板捕获事件,通过虚拟鼠标设备将它们变换成鼠标事件并使它们暴露于OS。而且,一组特定触摸屏手指手势被定义成启用/禁用和控制鼠标活动。

[0050] 在一些实施例中,用户可以在屏幕上更准确地定点、可以触发鼠标让位(Mouse Move Over)事件,并且可以容易地触发右击事件。用户不需要携带外部鼠标。一些实施例的优点包括(1)在鼠标模式和正常触摸面板工作模式之间无缝切换,而不手动运行/停止任何鼠标仿真应用;(2)软件逻辑对OS用户模式模块是透明的,并且不依赖于任何用户模式框架;以及(3)以相同的使用体验无缝地支持Windows经典台式模式和现代(Metro)UI二者。

[0051] 因此,参考图19,可以以软件、固件和/或硬件来实施序列80。在软件和固件实施例中,其可通过存储在一个或多个非瞬时计算机可读介质(诸如磁性、光学或半导体存储装置)中的计算机执行的指令来实施。

[0052] 参考图19,可以以软件、固件和/或硬件来实施虚拟鼠标序列80。在软件和固件实施例中,其可通过存储在一个或多个非瞬时计算机可读介质(诸如磁性、光学或半导体存储装置)中的计算机执行的指令来实施。

[0053] 如菱形82中所确定的,通过确定是否检测到特性触摸来开始序列80。例如,特性触摸可以是图15中描绘的三手指触摸,其指示期望进入虚拟鼠标模式。如果没有检测到该触摸,则该流程不会继续并且设备停留在常规触摸模式。

[0054] 如果已经检测到触摸模式特性,则确定接触的位置,如在块84中所指示的。具体地,在一个实施例中,检测到中指接触屏幕的屏幕上的位置。在一个实施例中,该位置可以是屏幕的预定区域,包括接近上边缘的区域、接近下边缘的区域、接近右边缘的区域和接近左边缘的区域、以及最终中心区域。

[0055] 然后如在块86中所指示的,基于接触位置来调整相对于指针手指的光标位置。例如,在一个实施例中检测到中心接触并且可如图3中所指示的那样使光标位置取向。如果检

测到在左边缘区域处的接触,则可如图4中所指示的那样调整光标位置。同样地,如果检测到右边缘接触,则可如图5中所指示的那样调整光标位置。如果检测到底边缘接触,则光标位置可如图6中所指示的那样。如果检测到底部左边缘,则可使用图7配置并且如果检测到底部右边缘,则可使用图8中所示的配置。相同的技术可被用于上部左边缘和上部右边缘。当然,除了在显示屏上定义有区别的区域之外或者作为其替换,还可使用其他约定。

[0056] 此外,在一些实施例中,当手指在屏幕的中心下方或上方时添加Y偏移。在一些实施例中,Y偏移的值可取决于沿着Y轴从指针手指到屏幕的中心的距离。

[0057] 根据另一实施例,在图12中示出的内核模式设备筛选器(KMDF)驱动器40位于触摸设备对象物理设备对象(PDO)44和用户层服务46之间。PDO表示Windows操作系统中的逻辑设备。在一些实施例中,筛选器驱动器是触摸供应方不可知的,但是是Windows特定的。

[0058] 架构还可使用驱动器74来通过12C协议支持标准HID。它也可以使用鼠标驱动器70来支持物理鼠标。

[0059] 该筛选器驱动器捕获在触摸设备和用户层服务之间的所有数据事务,尤其是来自外部触摸控制器48的触摸事件数据。它处理该数据并识别触摸屏上的预定义手指手势,并且然后将它们转换成鼠标事件。通过虚拟HID鼠标物理设备对象(PDO)50和HID类驱动器72将这些事件发送至OS。

[0060] 在图13中示出该筛选器驱动器40的内部架构。图13和图11中示出的架构指的是通过触摸解决方案的两个不同鼠标。图13示出基于中央处理器筛选器驱动器的解决方案的架构设计。在运行于CPU上的Windows软件驱动器内部实施该架构设计。它不使用图11中所示的内核。它包括三个主要部分。触摸事件数据捕获回调60是注册到对触摸设备44对象的每个请求内的回调函数,以及一组数据提取函数。每当触摸设备对象完成填充有触摸数据的请求时就调用这些函数。这些函数提取感兴趣的数据并且将该数据发送至下一收件箱模块68,该数据包括X/Y坐标、触摸屏上手指的数目和单独的手指标识符。而且取决于来自数据转换和变换模块62的虚拟鼠标活动的结果(是/否),回调判定是否将原始触摸事件数据发送至OS(菱形66)。

[0061] 触摸数据转换和变换62是筛选器的主要逻辑部分,其识别预定义手指手势、将它们转换成鼠标数据并判定(菱形66)是否进入虚拟鼠标模式。该部分是如图14中所示那样实施的状态机。

[0062] 虚拟鼠标设备对象处理机64接收经转换的鼠标事件数据并且将它封装到HID输入报告中,并且然后通过虚拟鼠标设备对象50将报告发送至OS。

[0063] 在一个实施例中手指手势被定义成与如图15、16、17和18中所示的虚拟鼠标一起工作。如图15中所示,使三个手指停留在触摸屏上而不移动达一时间段(例如三秒)激活触摸到事件变换。这使筛选器驱动器禁用将原始触摸事件数据传递至OS。当触摸至变换有效时,再次将三个手指放在触摸屏上去激活(deactivate)该变换,并允许原始触摸事件数据经由图13中的收件箱模块68传递至OS。

[0064] 如果仅一个手指触摸和移动,如由箭头A所指示的,则鼠标光标在屏幕上如图16中通过箭头B所示那样移动。如果如由箭头C所指示那样两个手指一起触摸和移动,则鼠标光标如由箭头B所指示的那样移动,就像如图17中由箭头D所示出的那样致动左按钮向下(拖动并放置图标I)。



[0065] 如果一个手指触摸(如在图18A和18B中通过圆圈T所示的)并且然后另一手指触摸并然后在一段时间(例如200ms)内被移除(轻敲(tap)),则触发鼠标按钮点击事件。意图是点击右按钮还是点击左按钮的识别取决于轻敲手指F是在左侧(图18A)还是在右侧(图18B)。

[0066] 为了支持如上文讨论的触摸到鼠标事件变换和手势,在一个实施例中在图13的触摸数据转换和变换模块62中实施图14中所示的状态机。

[0067] 在一个实施例中存在图14中所图示的四个状态。在空闲状态90中,没有手指在触摸屏上并且没有生成鼠标事件。在一个手指状态92中,在触摸上检测到一个手指,并且根据该手指在触摸上移动的距离和方向,而将鼠标移动事件发送至OS。在一个手指进入两个手指状态94中,从一个手指状态在触摸上检测到两个手指。然而,不确定这是否是用户手指轻敲事件。所以流程等待点击超时(例如200ms)。如果在该时间耗尽之前再次在触摸屏上检测到仅一个手指,则流程移动回至一个手指状态92并触发左/右按钮点击事件。如果该超时发生,则状态将变成两个手指状态96。在两个手指状态中,在触摸屏上检测到两个手指,并且根据这两个手指在触摸屏上移动的距离和方向,光标移动,其中左按钮向下事件移动被发送至OS。

[0068] 此外,在一个实施例中扫描超时(例如20ms)等于触摸扫描间隔的两倍。如果在该扫描超时之后没有接收到触摸事件,则用户已经从屏幕移除所有手指并且流程返回到空闲状态。

[0069] 根据一些实施例,通过同时利用多于一个手指触摸屏,触摸输入设备(诸如触摸屏)可在鼠标模式中操作。在一个实施例中,可利用三个手指。在一个实施例中三个手指可以是拇指连同食指和中指。然后食指和中指可被用来左击或右击以进入虚拟鼠标命令。

[0070] 在一些实施例中,一种系统可检测由多个手指在触摸输入设备上的同时接触。在三手指屏幕触摸命令的情况下,该系统可确定是左手还是右手在设备上以及三个手指的相对位置。这可以完成的一种方式求解由三个接触点定义的三角形的性质以及特别地其形状,并且根据其来确定是用户的左手还是右手在设备上。该手识别可能在确定是左击还是右击被信号通知中是重要的。在一个实施例中可通过在屏幕上轻敲食指或者中指来用信号通知左击或右击,这取决于使用左手或右手中的哪个。在一个实施例中,左手的食指处在右位置中,并且右手的食指处在左位置中。它们两个都是左击。所以在一些实施例中手识别可以是重要的。

[0071] 下面的条款和或示例关于另外的实施例:

[0072] 一个示例实施例可以是一种方法,其包括检测触摸输入设备上的接触,确定所述接触的位置,以及在基于所述接触的位置而变化的相对于所述接触的定位处显示光标。一种方法还可包括:响应于所述接触朝向屏幕边缘移动,将光标从相对于所述接触更中心的第一定位移动至所述接触的较不中心的第二定位。一种方法还可包括基于到屏幕边缘的接近度来关于所述接触移动所述光标。一种方法还可包括使用供应方独立内核来实现独立于触摸供应方内核进行操作的机制。一种方法还可包括在初始化期间加载所述供应方独立内核,在不依赖于任何平台操作系统的情况下在图形处理单元上运行所述供应方独立内核。一种方法还可包括通过虚拟鼠标设备对象使鼠标输入事件暴露于操作系统。一种方法还可包括使用内核模式驱动器来创建虚拟鼠标设备对象。一种方法还可包括检测输入设备是处

于触摸模式还是处于虚拟鼠标模式,每个模式与不同人类接口设备分组相关联。一种方法还可包括筛选出未被检测到的模式的分组。一种方法还可包括使用用于实施虚拟鼠标模式的驱动器。

[0073] 另一示例实施例可包括一个或多个非瞬时计算机可读介质,其存储指令,所述指令被执行来实行包括以下的序列:检测触摸输入设备上的接触;确定所述接触的位置;以及在基于所述接触的位置而变化的相对于所述接触的定位处显示光标。该介质还可包括所述序列包括响应于所述接触朝向屏幕边缘移动,将光标从相对于所述接触更中心的第一定位移动至所述接触的较不中心的第二定位。该介质可包括所述序列包括基于到屏幕边缘的接近度来关于所述接触移动所述光标。该介质可包括所述序列包括使用供应方独立内核来实现独立于触摸供应方内核进行操作的机制。该介质可包括所述序列包括在初始化期间加载所述供应方独立内核,在不依赖于任何平台操作系统的情况下在图形处理单元上运行所述供应方独立内核。该介质可包括所述序列包括通过虚拟鼠标设备对象使鼠标输入事件暴露于操作系统。该介质可包括所述序列包括使用内核模式驱动器来创建虚拟鼠标设备对象。该介质可包括所述序列包括检测输入设备是处于触摸模式还是处于虚拟鼠标模式,每个模式与不同人类接口设备分组相关联。该介质可包括所述序列包括筛选出未被检测到的模式的分组。该介质可包括所述序列包括使用用于实施虚拟鼠标模式的驱动器。

[0074] 在另一示例实施例中,可以是一种装置,其包括:处理器,其用来检测触摸输入设备上的接触,确定所述接触的位置,以及在基于所述接触的位置而变化的相对于所述接触的定位处显示光标;以及耦合至所述处理器的存储装置。该装置可包括所述处理器用来响应于所述接触朝向屏幕边缘移动而将光标从相对于所述接触更中心的第一定位移动至所述接触的较不中心的第二定位。该装置可包括所述处理器用来基于到屏幕边缘的接近度来关于所述接触移动所述光标。该装置可包括所述处理器用来使用供应方独立内核来实现独立于触摸供应方内核进行操作的机制。该装置可包括所述处理器用来在初始化期间加载所述供应方独立内核,在不依赖于任何平台操作系统的情况下在图形处理单元上运行所述供应方独立内核。

[0075] 遍及该说明书对“一个实施例”或“实施例”的参考意指结合该实施例所述的特定特征、结构或特性被包括在涵盖在本公开内的至少一个实施方式中。因此,短语“一个实施例”或“在实施例中”的出现不一定指的是同一实施例。此外,可以以除所图示的特定实施例之外的其他适当形式来建立特定特征、结构或特性并且所有此类形式可被涵盖在本申请的权利要求内。

[0076] 尽管已经描述了有限数目的实施例,但是本领域技术人员将会从其认识到许多修改和变化。意图在于所附权利要求覆盖如落入本公开的真实精神和范围之内内的所有此类修改和变化。

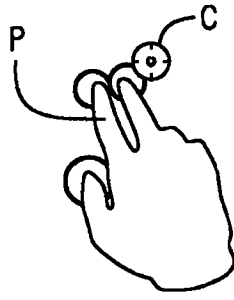


图1

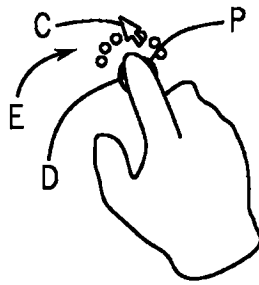


图2

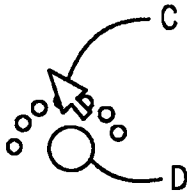


图3



图4



图5

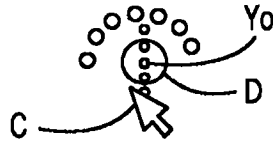


图6

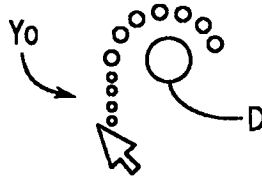


图7

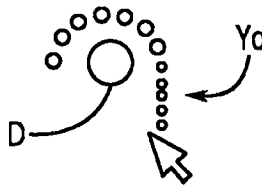


图8

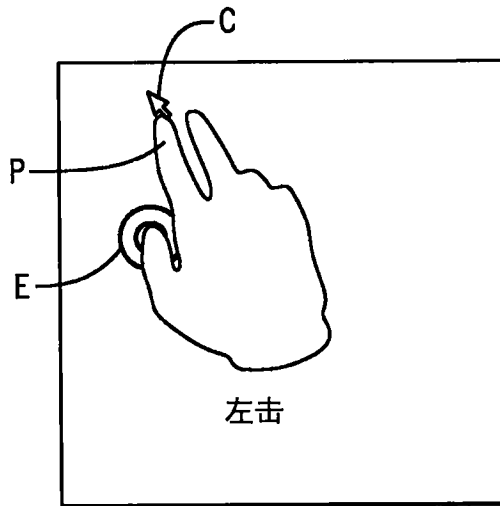
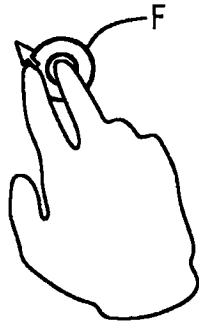


图9



右击

图10

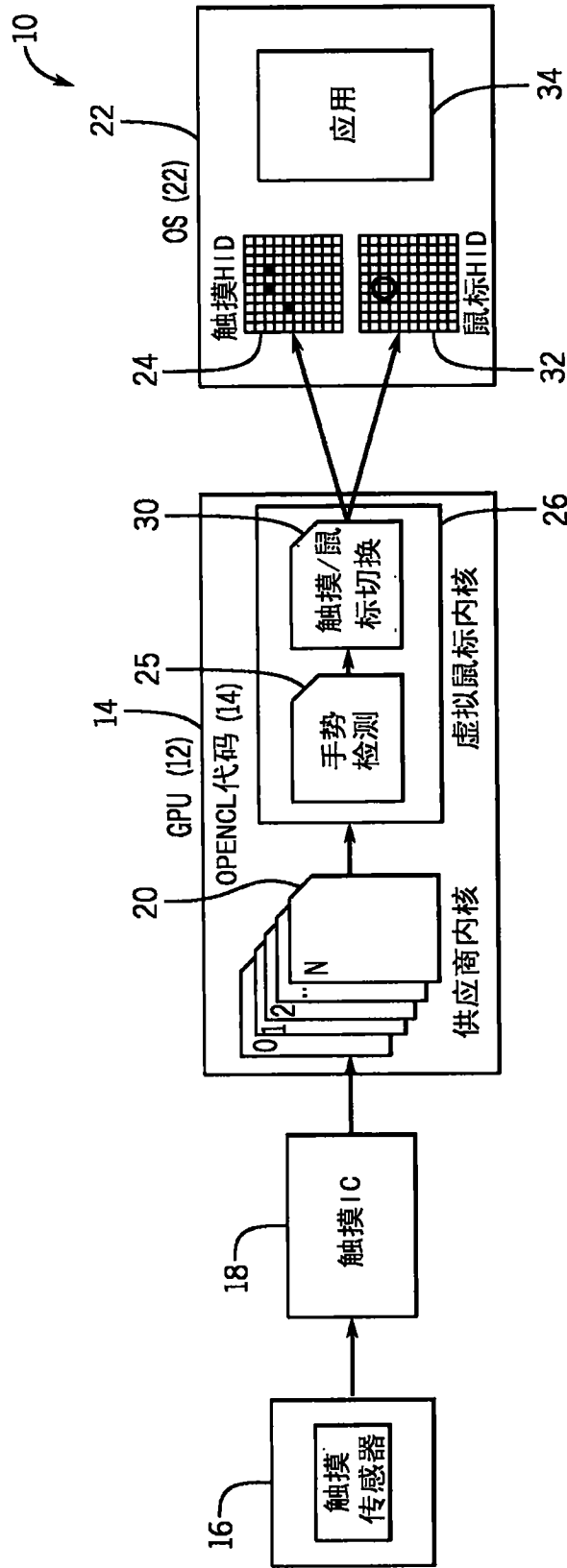


图11

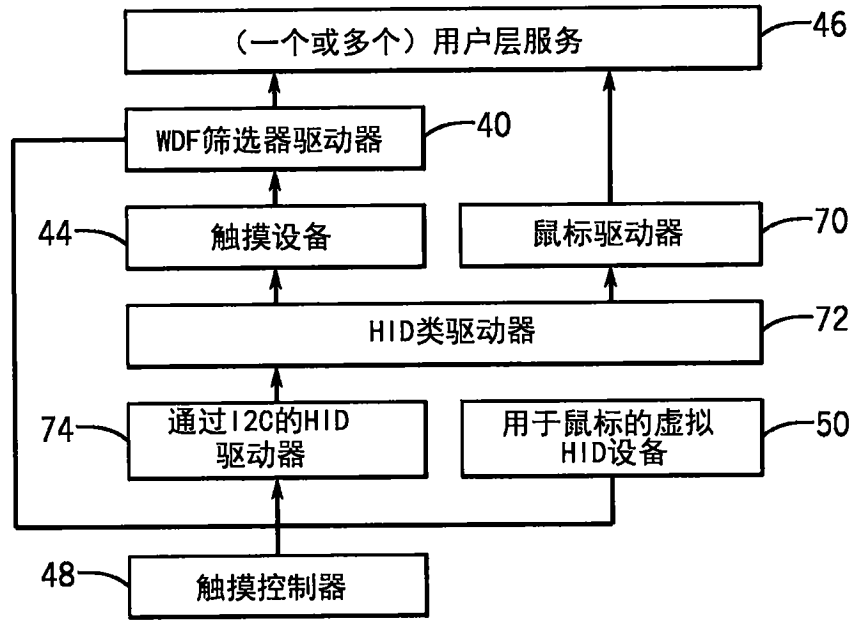


图12

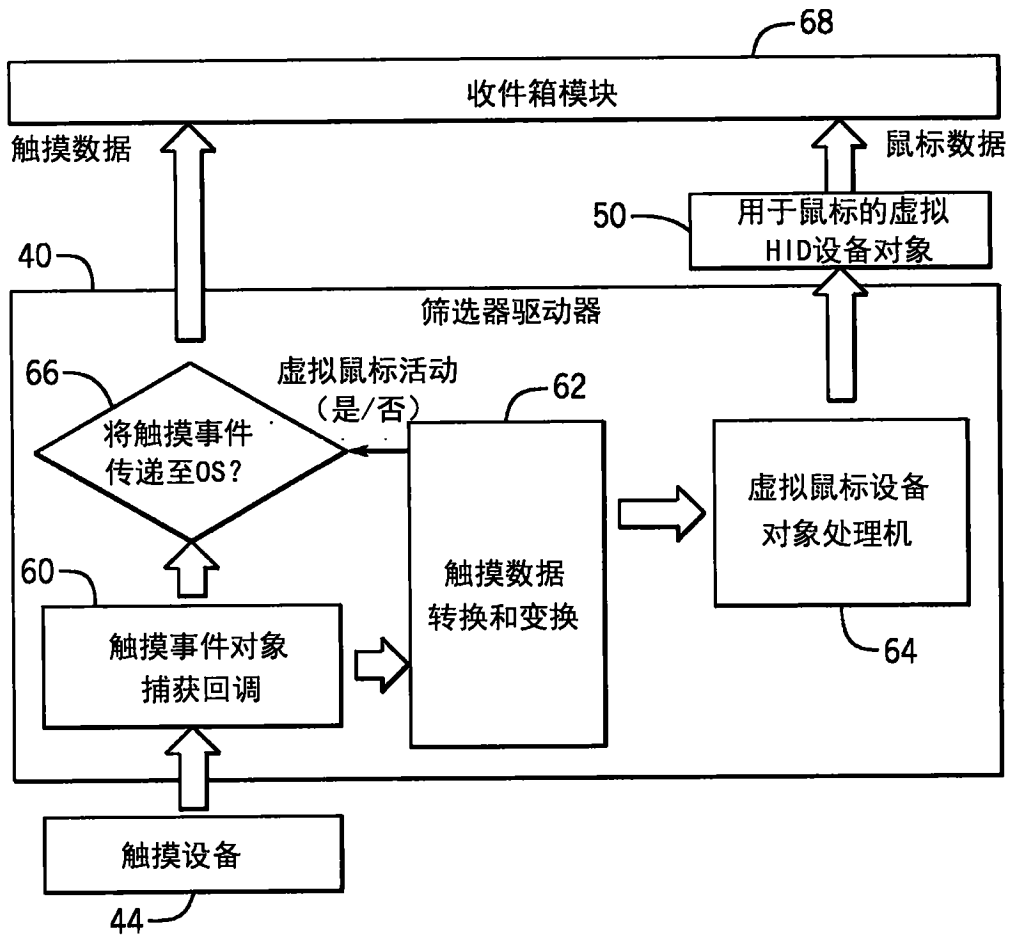


图13

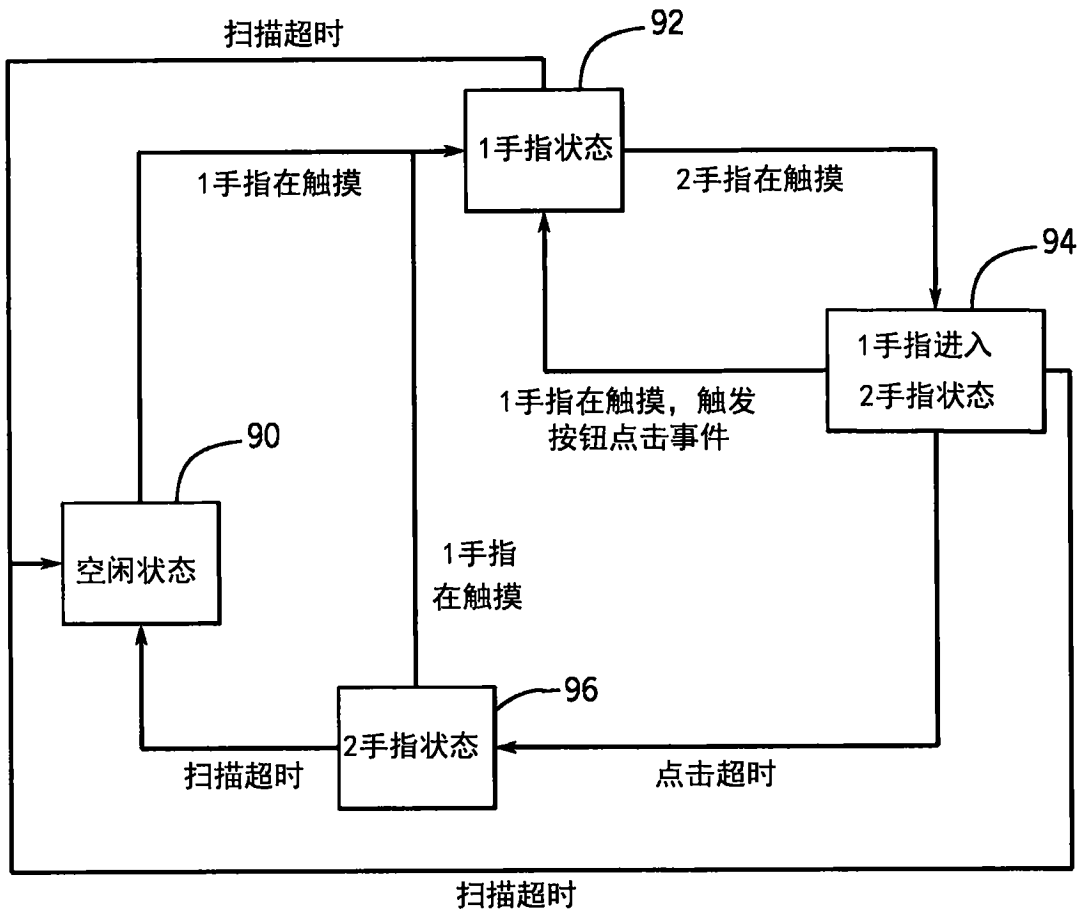


图14

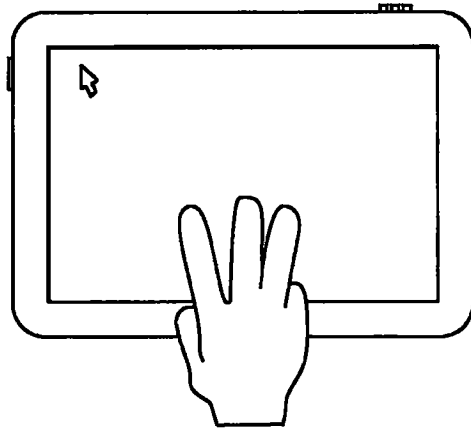


图15



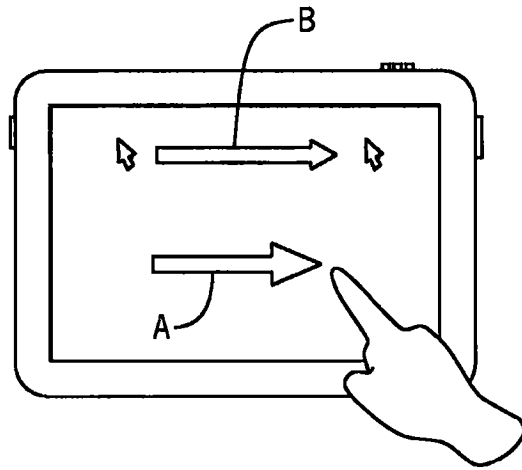


图16

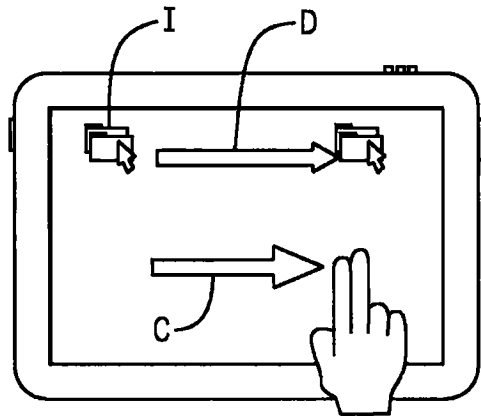


图17

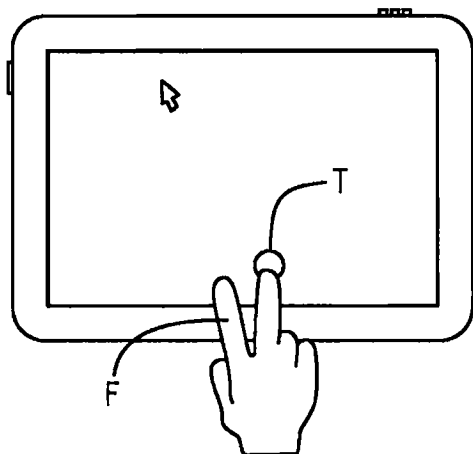


图18A

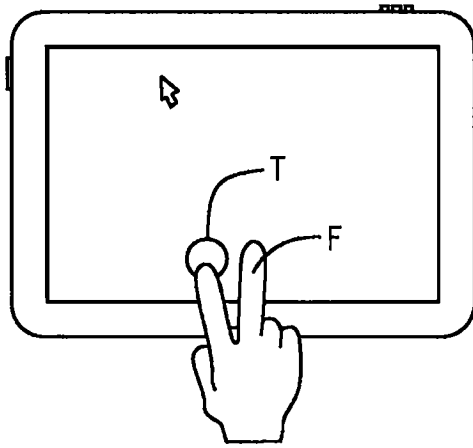


图18B

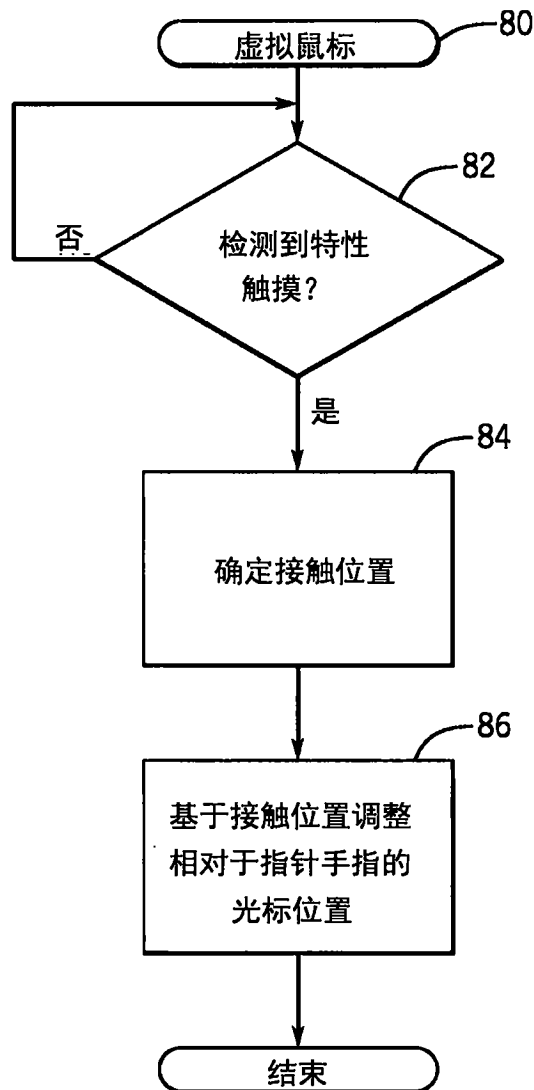


图19