



(12) 发明专利

(10) 授权公告号 CN 102609185 B

(45) 授权公告日 2015. 11. 18

(21) 申请号 201110448471. 5

(22) 申请日 2011. 12. 28

(30) 优先权数据

12/981, 139 2010. 12. 29 US

(73) 专利权人 微软技术许可有限责任公司

地址 美国华盛顿州

(72) 发明人 O·伯克斯 J·H·马修斯三世

A·盖格

(74) 专利代理机构 上海专利商标事务所有限公司

司 31100

代理人 陈斌

(51) Int. Cl.

G06F 3/0488(2013. 01)

G06F 3/041(2006. 01)

(56) 对比文件

CN 101681234 A , 2010. 03. 24, 参见说明书第【0025】-【0080】段, 图 9A-9H.

KR 20080076200 A , 2008. 08. 20, 全文 .

KR 20100027311 A , 2010. 03. 11, 全文 .

KR 20100134339 A , 2010. 12. 23, 全文 .

US 2009207144 A1 , 2009. 08. 20, 参见图 4, 说明书第【0025】-【0040】段 .

WO 2010075136 A2 , 2010. 07. 01, 全文 .

审查员 宋芸芸

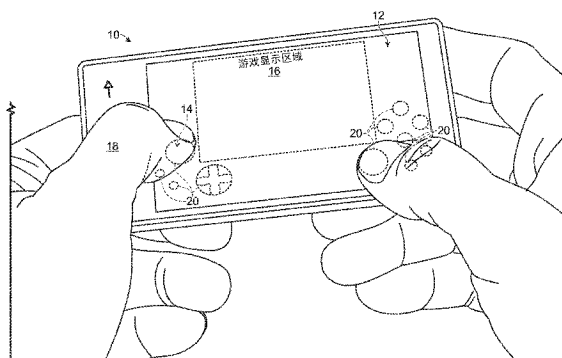
权利要求书3页 说明书6页 附图4页

(54) 发明名称

包括虚拟控制器的计算设备及其输入方法

(57) 摘要

本发明涉及用于触摸显示器的虚拟控制器。提供了用于具有包括被配置成检测用户的手指的触敏的触摸传感器的触敏显示器的计算设备的系统和方法。该方法可包括通过触摸传感器检测所述显示器上的初始手指落下位置, 以及在所述手指落下位置建立虚拟控制器的空档位置。该方法可进一步包括检测手指相对于初始手指落下位置的后续移动, 以及根据手指相对于初始手指落下位置的后续移动确定控制器输入参数。该方法可进一步包括生成指示所确定的控制器输入参数的控制器输入消息。



1. 一种用于计算设备的输入方法,所述计算设备具有包括被配置成检测用户的手指的触摸的触摸传感器的触敏显示器,所述方法包括:

通过所述触摸传感器检测在所述触敏显示器上的初始手指落下位置;

在所述初始手指落下位置建立虚拟控制器的空档位置;

检测手指相对于所述初始手指落下位置的后续移动;

基于手指相对于所述初始手指落下位置的所述后续移动确定控制器输入参数;以及

生成指示所确定的控制器输入参数的控制器输入消息;

检测手指抬起事件;

检测辅助控制区中的后续手指落下位置;

在检测到所述辅助控制区中的后续手指落下位置时,禁止对所述空档位置的重新定位;以及

基于所述辅助控制区中的所述后续手指落下位置确定辅助控制输入参数;

其中,所生成的控制器输入消息包括关于所述辅助控制输入参数的指示。

2. 如权利要求 1 所述的输入方法,其特征在于,还包括:

检测手指抬起事件;

检测不处于所述辅助控制区中的后续手指落下位置;以及

在不处于所述辅助控制区中的所述后续手指落下位置重新定位所述虚拟控制器的所述空档位置。

3. 如权利要求 2 所述的输入方法,其特征在于,所述手指抬起事件是手指与所述显示器不接触达预定阈值时间段。

4. 如权利要求 2 所述的输入方法,其特征在于,还包括:

如果所述初始手指落下位置和 / 或所述后续手指落下位置出现在靠近所述触敏显示器的边界的区域内,则确定最近可接受虚拟控制器位置,其中所述初始手指落下位置和 / 或所述后续手指落下位置出现在靠近所述触敏显示器的边界的区域内会导致所述虚拟控制器的可操纵区域受到所述边界的妨碍。

5. 如权利要求 1 所述的输入方法,其特征在于,所述手指是拇指,而所述虚拟控制器是虚拟操纵杆控制器。

6. 如权利要求 1 所述的输入方法,其特征在于,确定控制器输入参数包括确定所述手指相对于所述初始手指落下位置的后续移动的方向、速度和 / 或相对量。

7. 如权利要求 1 所述的输入方法,其特征在于,检测初始手指落下位置、建立空档位置、检测后续移动、确定控制器输入参数以及生成控制器输入消息是由计算设备上执行的应用程序编程接口执行的,所述方法进一步包括:

从所述应用程序编程接口向所述计算设备上执行的程序输出所述控制器输入消息。

8. 如权利要求 1 所述的输入方法,其特征在于,还包括:

检测所述后续移动期间所述手指对所述触敏显示器的压力,其中生成控制器输入消息包括关于所检测到的压力的指示。

9. 如权利要求 8 所述的输入方法,其特征在于,检测压力包括从与所述手指相关联的触摸的区域随时间的变化确定压力,和 / 或从耦合于所述触敏显示器的压力传感器确定压力。

10. 如权利要求 1 所述的输入方法,其特征在于,所述辅助控制区被配置为虚拟按钮或方向键。

11. 一种计算设备,包括:

包括被配置为检测用户的手指的触摸的触摸传感器的触敏显示器;

由所述计算设备的处理器执行的虚拟控制器输入模块,所述虚拟控制器输入模块被配置为:

通过所述触摸传感器检测所述触敏显示器上的初始手指落下位置;

在所述初始手指落下位置建立虚拟控制器的空档位置;

检测所述手指相对于所述初始手指落下位置的后续移动;

根据所述手指相对于所述初始手指落下位置的后续移动确定控制器输入参数;以及

生成指示所确定的控制器输入参数的控制器输入消息;

检测手指抬起事件;

检测在辅助控制区中的后续手指落下位置;

在检测到所述辅助控制区中的所述后续手指落下位置时,禁止对所述空档位置的重新定位;

基于在所述辅助控制区中的后续用户触摸,确定辅助控制输入参数;以及

生成所述控制器输入消息以进一步包括关于所述辅助控制输入参数的指示。

12. 如权利要求 11 所述的计算设备,其特征在于,所述虚拟控制器输入模块被配置为:

检测手指抬起事件;

检测不处于所述辅助控制区中的后续手指落下位置;以及

在不处于所述辅助控制区中的所述后续手指落下位置重新定位所述虚拟控制器的空档位置。

13. 如权利要求 11 所述的计算设备,其特征在于,所述手指是拇指,而所述虚拟控制器是虚拟操纵杆控制器。

14. 如权利要求 11 所述的计算设备,其特征在于,所述虚拟控制器输入模块被配置为显示锚定在所述空档位置的控制图标,所述控制图标向所述用户提供关于控制的定位的可视指示。

15. 如权利要求 11 所述的计算设备,其特征在于,所述计算设备被配置为移动电话或平板计算机。

16. 如权利要求 11 所述的计算设备,其特征在于,所述控制器输入消息包括关于在所述后续移动期间所述手指对所述显示器的压力的指示,其中所述压力是基于从与所述触敏显示器耦合的压力传感器接收的信号或基于与所述手指相关联的触摸的区域随时间的变化而被检测的。

17. 如权利要求 11 所述的计算设备,其特征在于,所述辅助控制区被配置为虚拟按钮或方向键,并且其中所述虚拟控制器输入模块被配置为在所述触敏显示器上在所述辅助控制区中显示辅助控制图标,以提供关于所述辅助控制区的位置的指示。

18. 一种用于计算设备的输入方法,所述计算设备具有包括被配置成检测用户的手指的多个触摸的触摸传感器的多点触摸显示器,所述方法包括:

通过所述触摸传感器检测所述多点触摸显示器上的初始手指落下位置;

在所述初始手指落下位置建立虚拟控制器的空档位置；
检测所述手指相对于所述初始手指落下位置的后续移动；
根据所述手指相对于所述初始手指落下位置的后续移动的方向、速度和 / 或相对量确定控制器输入参数；
生成指示所确定的控制器输入参数的控制器输入消息；
检测手指抬起事件；
检测处于或不处于辅助控制区中的后续手指落下位置；
当检测到所述后续手指落下位置不处于所述辅助控制区中时，在不处于所述辅助控制区中的所述后续手指落下位置重新定位所述虚拟控制器的所述空档位置；
当检测到所述后续手指落下位置处于所述辅助控制区中时，禁止对所述空档位置的重新定位；以及
基于在所述辅助控制区中的后续用户触摸，确定辅助控制输入参数；
其中所生成的控制器输入消息包括所述辅助控制输入参数的指示。

包括虚拟控制器的计算设备及其输入方法

技术领域

[0001] 本发明涉及具有触敏显示器的计算设备,尤其涉及用于触摸显示器的虚拟控制器。

背景技术

[0002] 便携式游戏设备被世界上许多用户所喜爱。一种类型的这样的游戏设备既包括硬按钮控制器又包括触摸屏,其中硬按钮控制器向用户提供触觉反馈来加强手指在控制器上的正确定位,而触摸屏使用户能够使用例如指示笔来与游戏交互。这种设备的一个缺陷是硬按钮设备中的宝贵空间,限制了显示器的大小,且增加了外壳的大小。减小硬按钮的大小是有挑战性的,因为小按钮通常对于用户而言是难以操纵的,特别是在玩快节奏游戏。另外,在也充当多功能平板计算设备或移动电话的游戏设备的上下文中,对于不玩游戏的消费者而言,设备正面上特别专用于玩游戏的专用按钮可能会有损于设备的有用性或外观。

发明内容

[0003] 提供了使用具有触摸感应显示屏的计算设备所需要的方法和系统,该显示屏中包含了用于检测玩家指头宽度的触摸传感器。其中包括通过触摸传感器检测手指在触摸屏上的初次触摸的方法,以及在手指触摸位置确立虚拟控制器空闲位置的方法。还包括手指相对于初始触摸位置相对移动顺序的方法和基于手指相对于初始触摸位置移动确立控制器输入参数的方法。以及生成用于指出已确定控制器输入参数的控制器输入信息。

[0004] 提供本发明内容是为了以简化的形式介绍在以下“具体实施方式”中进一步描述的一些概念。本发明内容不旨在标识出所要求保护的主题的关键特征和必要特征,也不旨在用于限制所要求保护的主题的范围。此外,所要求保护的主题不限于解决本公开中的任何部分所记录的任何或全部缺点的实现方式。

附图说明

[0005] 图 1 是示出用于触摸显示器的虚拟控制器的一个实施例的计算设备的透视图。

[0006] 图 2 是图 1 的计算设备的示意图。

[0007] 图 3 是示出图 1 的虚拟控制器的相对移动的空档位置的图示视图。

[0008] 图 4 是图 1 的虚拟控制器的空档位置的重新定位的图示视图。

[0009] 图 5 是操纵图 1 的控制器图标的拇指抬起事件的图示视图。

[0010] 图 6 是示出用于被配置为输出用于触敏显示器的虚拟控制器的计算设备的方法的流程图。

具体实施方式

[0011] 图 1 是示出用于触敏显示器的虚拟控制器 14 的计算设备 10 的一个实施例的透视图。计算设备 10 可被配置为例如移动电话、平板计算机、或其他手持计算设备。计算设备

10 可包括诸如多触点显示器 12 之类的触敏显示器,该多触点显示器 12 包括被配置成检测用户的手指的多个触摸的触摸传感器,这将关于图 2 更详细地被讨论。另外,计算设备 10 还可以被配置为经该多个触摸从玩家接收输入,并可实现一输入方法,以便控制虚拟控制器 14。这一方法将关于图 6 详细被描述。

[0012] 首先转到图 1,在使得用户能够控制诸如视频游戏之类的程序的配置中,计算设备 10 可以在多点触摸显示器 12 上显示虚拟控制器 14。在所描绘的实施例中,虚拟控制器 14 是包括可用于控制视频游戏的各方面以允许在游戏显示区域 16 中进行游戏的虚拟操纵杆控制器的游戏控制器。通常,虚拟控制器以不会不适当地干扰游戏显示区域的显示的方式被呈现。在另一实施例中,虚拟控制器输入模块可以通过扬声器 / 话简单元 26 提供对应于虚拟控制器输入的听觉反馈。

[0013] 虚拟控制器 14 可以对用户手指的多个触摸作出响应。例如,手指可以为拇指 18,拇指可以在任何常规方向移动,而该移动可由虚拟控制器 14 转换成一个或多个控制器输入参数。控制器输入参数可包括对应于拇指 18 移动的方向、速度、及 / 或相对大小等。结果,在某些实施例中,一个或多个控制图标 20 可向用户提供关于控制的定位的视觉指示,该视觉指示随着用户移动他的或她的手指而更新。

[0014] 一个或多个控制图标 20 可表示可允许用户能够控制游戏的一个或多个方面,一个或多个虚拟控制器 14 和 / 或辅助控制器。控制图标 20 各自可以是虚拟操纵杆、按钮、D-pad (方向键) 按钮、滑块、以及可被操纵来控制游戏的某一方面的任何其他特征中的一个或多个。在图 1 所示的实施例中,将理解虚拟控制器实现 MICROSOFT® XBOX® 控制器。也可以使用其它控制器实现。

[0015] 通常,虚拟控制器是以不会在视觉上不适当地干扰游戏显示区域的显示的方式被呈现。从而,在某些实施例中,显示画面上不提供控制器图标,以将显示画面上的游戏区域的可视面积最大化。在其他实施例中,图标可部分透明以显露其下面的游戏,或者可以短暂的时间段呈现图标然后逐渐消失。

[0016] 图 2 是示出图 1 的计算设备 10 的软件和硬件组件的示意图。计算设备 10 包括多点触摸显示器 12、处理器 22、存储器单元 24、扬声器 / 话简单元 26、以及大容量存储单元 28,所有这些单元均被配置为通过一个或多个数据总线彼此通信。

[0017] 多点触摸显示器 12 可采用各种显示技术来产生可视图像,且可包括例如图形用户界面 (GUI) 32。多点触摸显示器 12 可被配置为例如液晶显示器 (LCD) 或由多个发光像素组成的有机发光二极管 (OLED) 显示器。多点触摸显示器 12 还可包括触摸传感器 30 和压力传感器 34。

[0018] 与多点触摸显示器 12 耦合的触摸传感器 30 可检测用户手指的多个触摸,例如触摸 36。与多点触摸显示器 12 耦合的压力传感器 34 可以从与手指相关联的触摸 36 的区域随时间的变化或从与增加的压力相关联的电特性的变化检测压力,从而充当应变仪。不背离本公开的范围,多点触摸显示器 12 可包括附加的或替代的传感器。

[0019] 大容量存储单元 28 可包括各种程序和模块,这些程序和模块包括虚拟控制器输入模块 42 和程序 38,可由处理器 22 使用存储器单元 24 和显示器 12 执行以实现本文所述的各种功能。另外,大容量存储单元 28 可存储可被配置成例如发送可控制多点触摸显示器 12 上的 GUI 32 的显示的显示输出 40 的一个或多个程序 38。

[0020] 将理解计算设备 10 可包括操作系统,操作系统包括应用程序编程接口 (API),通过 API 各种程序与操作系统资源通信并利用操作系统资源。在某些实施例中,虚拟控制器输入模块 42 可被实现为计算设备上的 API 48。替代地,虚拟控制器输入模块可被实现为计算设备 10 上执行的程序、库、服务、或其他软件组件。

[0021] 虚拟控制器输入模块 42 可由计算设备 10 的处理器 22 执行。虚拟控制器输入模块 42 可被配置为通过触摸传感器 30、压力传感器 34 和 / 或附加传感器来在多触点显示器 12 上按照触摸检测用户的手指以及对应于这些触摸的移动。以这种方式,虚拟控制器输入模块 42 可转换一个或多个传感器接收到的数据,基于该数据生成控制器输入消息 46,以及将该消息发送到程序 38。

[0022] 例如,虚拟控制器输入模块 42 可以通过触摸传感器 30 检测多点触摸显示器 12 上的初始手指落下位置。另外,虚拟控制器输入模块 42 可在初始手指落下位置处建立虚拟控制器 14 的空档位置。也就是说,虚拟控制器输入模块 42 可以在手指接触多点触摸显示器 12 时检测初始触摸。进一步,虚拟控制器输入模块 42 可被配置为显示锚定到空挡位置或可控地以空挡位置为中心的控制图标。在建立了空挡位置后,虚拟控制器输入模块 42 可被配置为例如检测手指相对于该初始手指落下位置的后续移动。其中,这些触摸和移动将关于图 3 至图 6 被更详细地讨论。

[0023] 虚拟控制器输入模块 42 可被配置为基于例如手指相对于该初始手指落下位置的该后续移动来确定控制器输入参数 44。确定控制器输入参数 44 可包括例如确定手指相对于该初始手指落下位置的该后续移动的方向、速度和 / 或相对大小。进一步,可由虚拟控制器输入模块 42 生成指示出所确定的控制器输入参数 44 的控制器输入消息 46,并将其发送到程序 38。控制器输入消息 46 充当虚拟控制器输入模块和程序 38 之间通信通道,程序如上所述可以是视频游戏或其他程序。该消息可包括基于检测到的后续移动的控制参数 44,和 / 或下文讨论的各种其他控制器输入参数。

[0024] 作为另一控制器输入参数 44 的一个示例,控制器输入消息 46 可包括关于在某一后续移动期间手指对多点触摸显示器 12 的压力的指示,其中该压力是 基于从压力传感器 34 接收的信号而被检测的。检测到的压力可基于例如与手指相关联的触摸的区域随时间的变化,或基于应变仪检测到的电特性的变化。

[0025] 另外,虚拟控制器输入模块 42 可以检测用户的手指对其他控制器的多个触摸,例如可控制游戏的各方面的与虚拟控制器 14 不同的辅助控制器。这些辅助控制器可与辅助控制区相关联,将关于图 5 更详细地讨论。基于来自这些辅助控制器的输出的控制器输入参数也可被包括在控制器输入消息 46 中。

[0026] 如上文所述,虚拟控制器输入模块 42 可被实现为应用程序编程接口 (API) 48。从而,例如检测初始手指落下位置、建立空挡位置、检测后续移动、确定控制器输入参数 44 以及生成控制器输入消息 46 可由计算设备 10 上执行的应用程序编程接口 48 执行。另外,应用程序编程接口 48 可将控制器输入消息 46 输出到要在计算设备 10 上被执行的程序 38。在其他实施例中,虚拟控制器输入模块 42 可被实现为驱动器独立可执行程序、库、服务或其他类型的软件组件。

[0027] 图 3 是示出图 1 的虚拟控制器 14 的相对移动的空档位置 50 的图示视图。如所示,用户的手指可做出对应于初始手指落下位置的接触,并在初始手指落下位置建立空档位置

50,如上所述。用户可通过做出相对于空档位置 50 的移动控制虚拟控制器 14。例如,用户可做出后续移动使得手指移动到位置 52,或用户可做出后续移动使得手指移动到位置 54。每次移动可在手指连续接触多点触摸显示器 12 时发生,并可以转换为由程序 36 以不同方式解释的不同的控制输入。例如,到位置 52 的后续移动可导致虚拟车辆向前行驶且向左转向,而到位置 54 的后续移动可导致使虚拟车辆向后开。将理解可在除了所描述的那些位置之外的位置中检测后续移动。

[0028] 图 4 是图 1 的虚拟控制器 14 的空档位置 50 的重新定位的图示视图。用户有时可能从多点触摸显示器 12 的表面抬起手指,且继之以后续的手指落下位置以控制虚拟控制器 14。在不重新定位虚拟控制器的情况下,用户可发现他或她对计算设备的紧握导致后续手指落下位置的徐变。结果,用户可能无意中错误点击虚拟控制器,导致不令人满意的用户体验、视频游戏中的悲剧结果等。

[0029] 为解决这个问题,虚拟控制器输入模块 42 被配置为在检测到这种徐变时重新定位空档位置。该重新定位可以如下被实现。如图 4 所示,用户的手指可做出对应于初始手指落下位置的接触,并在该初始手指落下位置建立虚拟控制器 14 的空档区域,如上所述。虚拟控制器输入模块 42 可被配置为检测手指抬起事件 58,其中,手指抬起事件 58 可以是例如手指不与多点触摸显示器 12 接触达预定的阈值时间段。

[0030] 虚拟控制器输入模块 42 可被配置为检测第一和第二后续手指落下位置,并在后续手指落下位置的中心或相对于后续手指落下位置的其他位置将空档位置重新定位到 50B、50C 处的对应位置,从而导致虚拟控制器被如 14B 和 14C 处所示那样重新定位。

[0031] 在某些情形中,用户可这样定位手指使得空档位置可能相当靠近多点触摸显示器 12 的边缘,且可能因此导致部分虚拟控制器。为解决这个问题,可确定空档位置的最近可接受位置,并可在最近可接受位置显示控制图标。从而虚拟控制器输入模块 42 可被配置为如果手指落下位置和 / 或后续手指落下位置出现在靠近触敏显示器的边界(这会导致虚拟控制器的可操纵区域被边界阻碍),则确定最近可接受虚拟控制器位置。在某些实施例中,该最近可接受位置可被看作是最靠近用户的手指落下位置、使虚拟控制器的整个可操纵区域位于显示画面上的位置。该特征可任选地被包括在例如新手模式期间,以允许用户变得自在掌握虚拟控制器。

[0032] 在另一示例中,可拒绝检测到的空档位置而无需确定最近可接受位置,从而控制图标不被显示在该最近可接受位置处。在另一示例中,可允许部分虚拟控制器。例如,如果用户选择高级模式,可启用该部分虚拟控制器,使得可用减少的功能来控制游戏,因为边界阻碍了虚拟控制器的可操纵区域的一部分。在这种情况下,虚拟控制器可以在虚拟控制器的可操纵区域中包括控制图标的部分显示。从而,部分虚拟控制器可允许仅来自用户可见的虚拟控制器的那些部分的控制。类似地,虚拟控制器的被显示器的边界遮挡的部分可能在视图外且不可操纵,导致减少的功能。

[0033] 图 5 是在用户尝试在使用诸如上述的虚拟操纵杆之类的主控制图标之后操纵一辅助控制图标时可检测到的拇指抬起事件的图示视图。如图 5 所示,可以建立空档位置 50 来控制虚拟控制器 14。一旦建立,虚拟控制器输入模块 42 即可检测手指抬起事件 58,并可进一步检测后续手指落下位置,例如在辅助控制区 56 内的手指触摸 60 处。

[0034] 辅助控制区 56 可被配置为虚拟按钮或方向键(D-pad)控制器 62,或其他用户输入

机制。从而,辅助控制图标可被显示在对应于虚拟按钮、D-pad 等的辅助控制区 56 中。通过显示这一辅助控制图标,虚拟控制器输入模块 42 可以向用户提供关于辅助控制区 56 的位置的指示。进一步,虚拟控制器输入模块 42 可被配置为基于用户手指在辅助控制区中的后续位置或相对移动来确定辅助控制输入参数。因此,虚拟控制器输入模块 42 可生成控制器输入消息 46,以便进一步包括关于辅助控制输入参数的指示。将理解,辅助控制区 56 可替代地被配置成各种其他配置,以允许与其他程序相关的用户输入。

[0035] 在一些实施例中,在检测到辅助控制区中的后续手指落下位置时,虚拟控制器输入模块 42 可被配置为禁止空档位置的重新定位。在这种情况下,虚拟控制器输入模块确定用户手不是因为无意徐变而移动,而是由于玩家控制离用作主控制的虚拟控制器预定距离和方向的辅助控制的意图。在这一情况下,虚拟控制器输入模块阻止对虚拟控制器的重新定位,而代之以使用户能够与位于辅助控制区的辅助控制进行交互。以此方式,用户可在例如主操纵杆控制器和辅助 D-pad 控制器之间来回切换他的或她的拇指。

[0036] 图 6 是示出用于被配置为通过多点触摸显示器的虚拟控制器 14 接收输入的计算设备的方法 600 的流程图。方法 600 可以是用于具有诸如包括被配置成检测用户的手指的多个触敏的触摸传感器的多触点显示器之类的触敏显示器的计算设备的输入方法。将理解,方法 600 可以使用上文所述的计算设备 10 的软件和硬件组件或使用其他合适的组件来执行。

[0037] 方法 600 开始于 602,并包括通过触摸传感器检测触敏显示器上的初始手指落下位置。方法 600 前进至 604,并包括如上所述在手指落下位置建立虚拟控制器的空档位置。

[0038] 方法 600 继续至 606,且包括检测手指相对于初始手指落下位置的后续移动。方法 600 前进到 608 且包括基于手指相对于初始手指落下位置的后续移动确定控制器输入参数。例如,控制器输入参数可包括与到诸如多点触摸显示器上显示的虚拟操纵杆之类的控制图标的后续移动相关联的方向、速度和 / 或大小。还可以检测其他控制器输入参数,如上所述。

[0039] 方法 600 还可以继续至 610,并可包括检测后续移动期间手指对多点触摸显示器的压力。因此,方法 600 可继续到 612,并包括生成指示所确定的控制器输入参数的控制器输入消息并可进一步包括关于所检测到的压力的指示。

[0040] 从 612,方法 600 可继续至 614 和 622 两者。在 622,方法 600 包括向计算设备上运行的程序发送包括控制器参数的控制器输入消息。在一些实施例中,步骤 602-622 可由 API 实现,而步骤 624 可以由计算设备上执行的程序实现。从而,从 622 到 624 的处理流程可退出操作系统 /API 空间,并进入程序空间。在 624,该方法可包括在程序中基于接收到的控制器输入消息调整 GUI 元素。从 624,方法 600 可结束,或者可替代地循环到步骤 606 以继续检测输入。

[0041] 返回到步骤 612 的结束,该方法也可分支到 614,在 614 处,该方法包括例如基于检测到在预定的时间段内没有与显示器的接触来判断是否发生手指抬起事件。如果对 614 的回答是“否”,则方法 600 继续至 606 并重新开始检测输入。如果对 614 的回答是“是”,则方法 600 继续至 616 且包括检测后续手指落下位置。

[0042] 然后方法 600 继续至 618 并包括确定是否已在辅助控制区中检测到后续手指落下位置。如果对 618 的答案是“否”,则方法 600 继续至 604,以重新建立空档控制器位置。也

就是说,604 可包括如上所述的在后续手指落下位置重新定位虚拟控制器的空档位置。

[0043] 如果对 618 的回答是“是”,则不重新定位空档位置。相反,方法 600 继续至 620,并包括基于辅助控制区中的后续手指落下位置确定辅助控制输入参数。例如,可以将辅助控制图标显示在辅助控制区中,辅助控制图标可被配置为 D-pad 控制器,示例辅助控制输入参数可以是 D-pad 控制器按下事件。上文还讨论了其他示例。从 620,方法 600 返回至 612,在 612,生成控制器输入消息可包括关于辅助控制输入参数的指示。

[0044] 如上文所讨论的,在一些实施例中,方法 600 的一些部分可由计算设备中执行的 API 执行。例如,API 可执行检测初始手指落下位置、建立空档位置、检测后续移动、确定控制器输入参数和 / 或生成控制器输入消息。

[0045] 如上文所讨论的,在一些实施例中,方法 600 的一些部分可进一步包括在诸如手指落下位置和 / 或后续手指落下位置之类的手指触摸出现在靠近触敏显示器的边界的区域中(这会导致虚拟控制器的可操纵区域被边界妨碍)的情况下,确定最近可接受虚拟控制器位置。该最近可接受虚拟控制器位置可以如上所述地被确定。在该最近可接受虚拟控制器位置而不是在手指触摸所检测到的空档位置处,虚拟控制器可被定位且一控制图标可被显示。在其他示例中,可以拒绝所检测到的空档位置而不显示最近可接受虚拟控制器位置,或者虚拟控制器可部分地可见以及可被操纵,如上所讨论的。

[0046] 将理解,上述系统和方法可被实现来使得用户能够使用显示在被配置为接受相对输入且还可具有相关联的辅助控制的触敏显示器上的虚拟控制器来与程序交互。另外,虚拟控制器可在玩家同时紧握设备并操纵控制时,方便地重新定位其空档位置以跟踪用户的手位置的任何变化。结果,可避免控制器错误点击,且可增强用户体验。

[0047] 本文使用术语“模块”、“程序”和“引擎”来指代在计算设备的处理器执行时执行一个或多个特定功能的软件。这些术语意图包含例如个体的或成组的可执行文件、数据文件、库、驱动程序、脚本、数据库记录。本文所述的实施例示出了这些模块、程序和引擎等的一个示例组织,但应理解的是,本文所描述的功能可由被不同地组织的软件组件来实现。此外,将理解的是,多点触摸显示器是一种能够在显示器上同时感测多个触摸的显示器。

[0048] 要理解的是,本文所述的配置和 / 或方法在本质上是示例性的,且不应在限制性的意义上考虑这些特定的实施例或示例,因为许多变型是可能的。本文所述的特定例程和方法可代表任何数量的处理策略中的一个或多个。如此,所示各种动作可以按所示顺序执行,以其他顺序执行,并行执行,或在某些情况下被省略。类似地,上述的过程的顺序可被改变。

[0049] 本公开的主题包括本文所揭示的各种过程、系统和配置以及其他特征、功能、动作和 / 或特性,以及它们的任何和全部等同物的所有新颖和非显而易见的组合和子组合。

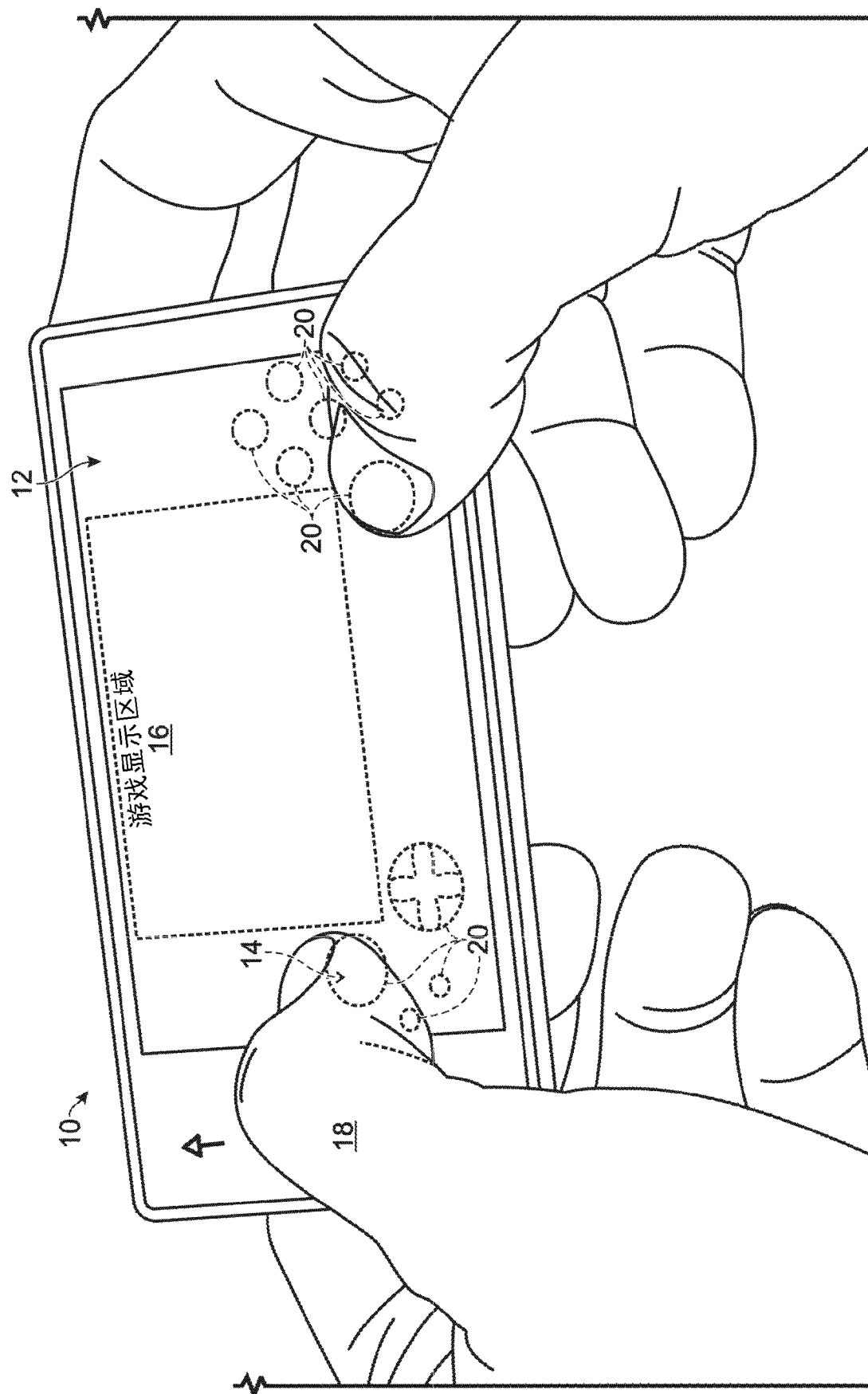


图 1

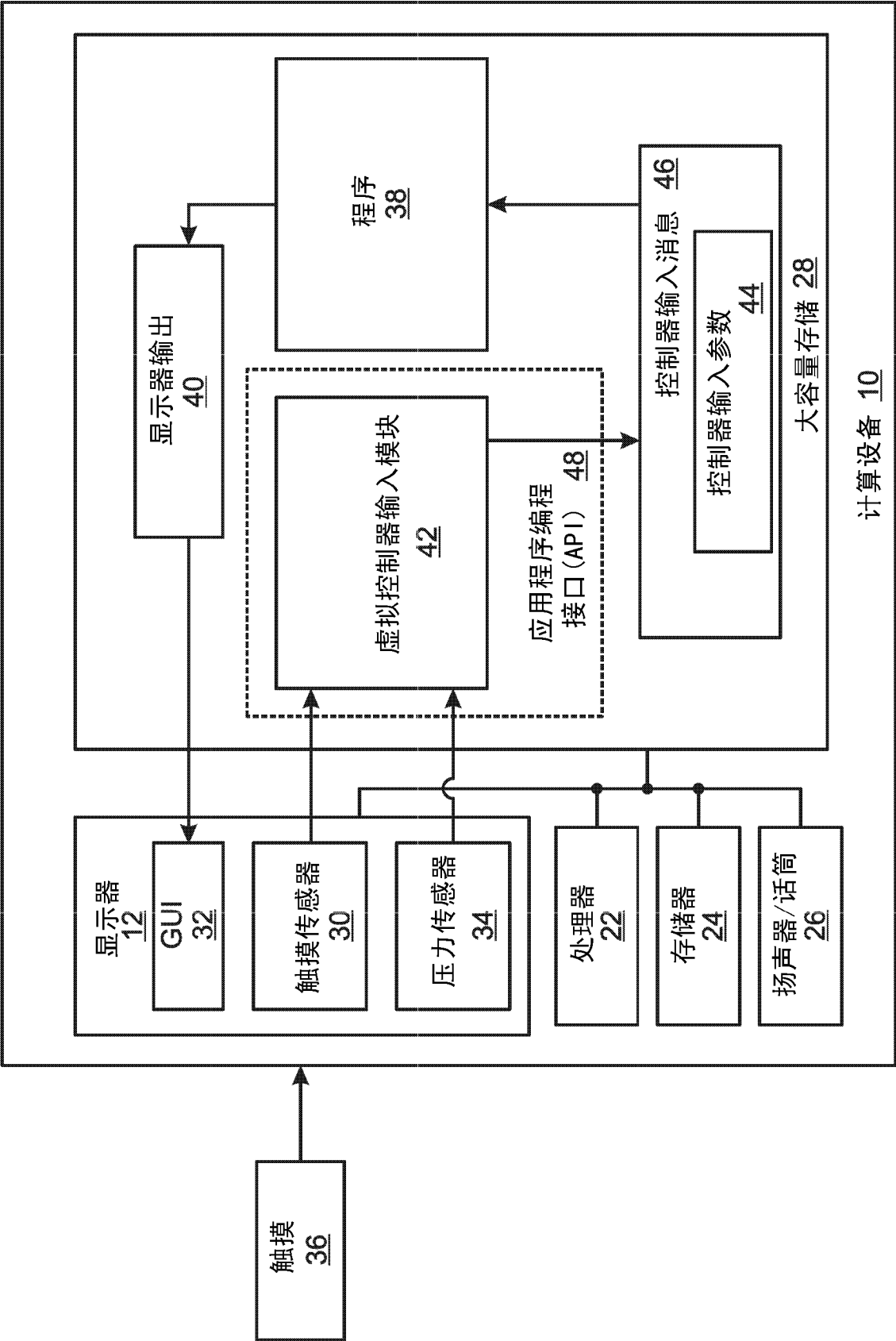
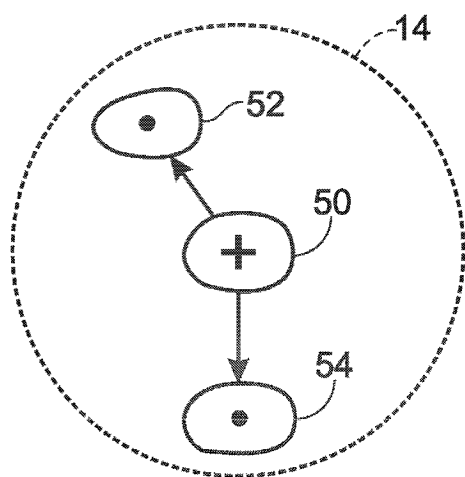
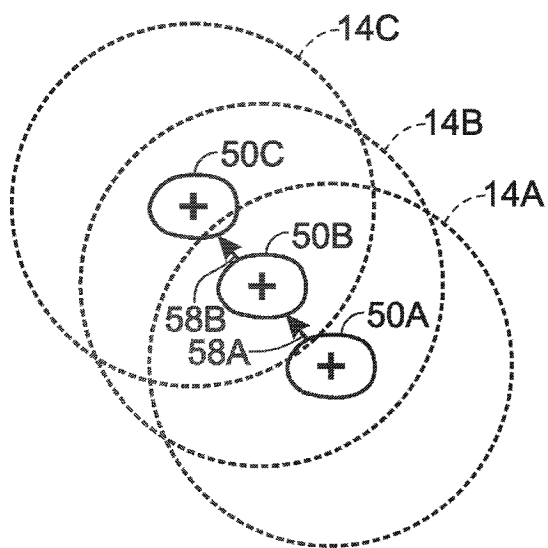


图 2



后续相对移动

图 3



重新定位空档位置

图 4

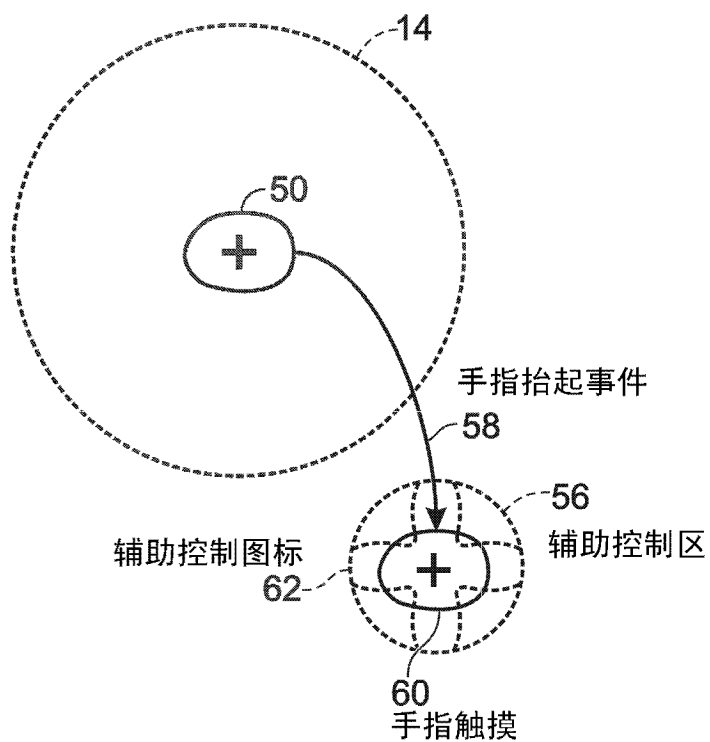


图 5

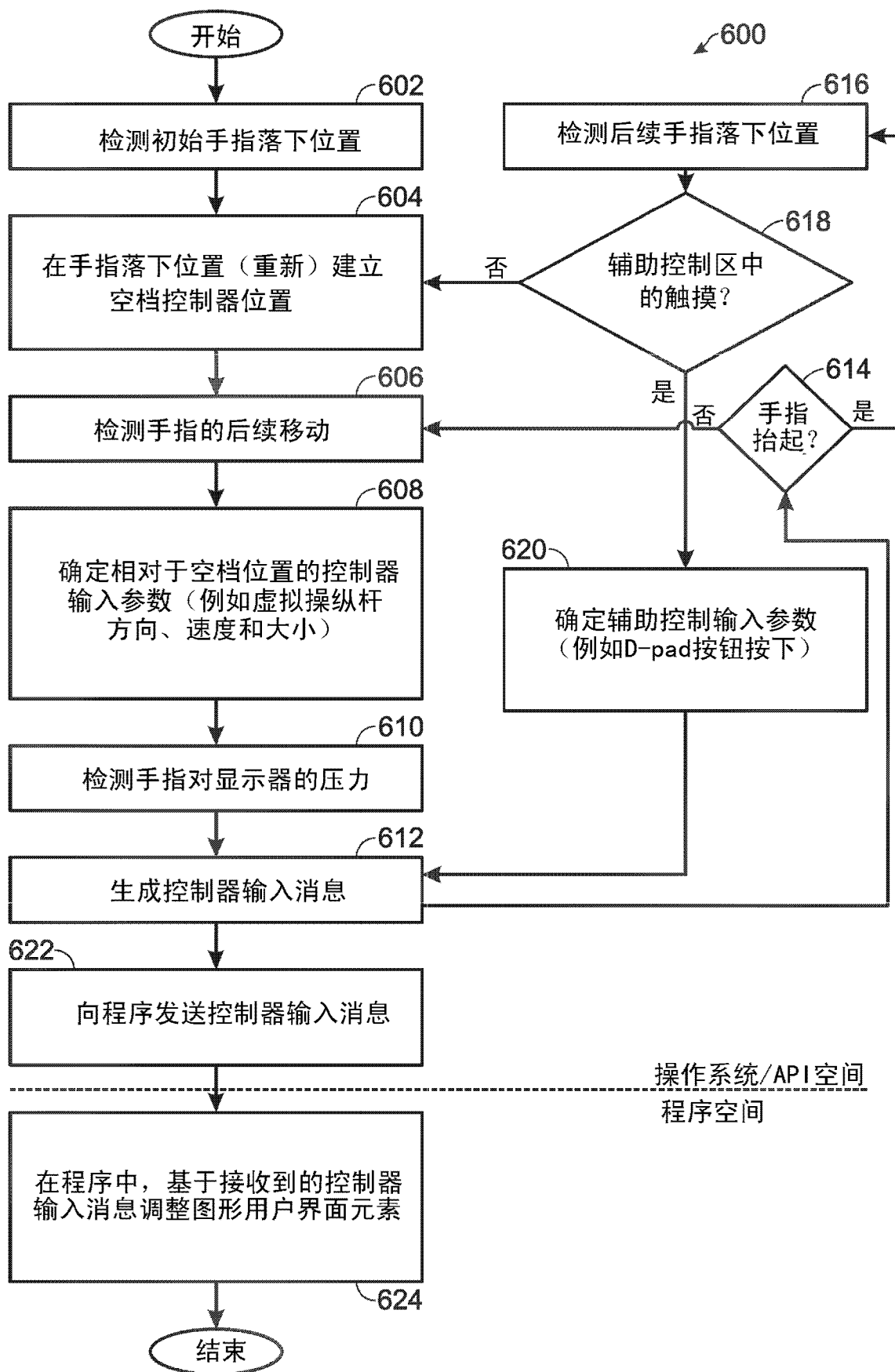


图 6