(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 104583902 A
(43)申请公布日 2015.04.29

(21)申请号 201380040370.1

(22)申请日 2013.07.08

(30)优先权数据
1250911-3 2012.08.03 SE
61/679,529 2012.08.03 US

(85)PCT国际申请进入国家阶段日
2015.01.29

(86)PCT国际申请的申请数据
PCT/SE2013/050878 2013.07.08

(87)PCT国际申请的公布数据
WO2014/021760 EN 2014.02.06

(71)申请人 科智库公司
地址 瑞典马尔默

(72)发明人 P·克隆霍尔姆

(74)专利代理机构 北京同恒源知识产权代理有限公司 11275
代理人 王维绮

(51)Int.Cl.
G06F 3/01(2006.01)
G06F 9/445(2006.01)
G06F 17/30(2006.01)
G06K 9/00(2006.01)

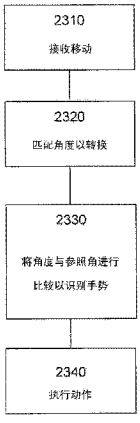
权利要求书2页 说明书19页 附图13页

(54)发明名称

改进的手势的识别

(57)摘要

本发明披露了一种用于识别由影像流中被追踪的对象所作的手势的计算装置，所述计算装置包括存储器和控制器，其中所述控制器被设置成：确定被追踪对象的移动向量；将移动向量与手势的参考库进行比较；并由此识别匹配移动向量的手势；其中所述移动向量包括至少一部分具有由相对角识别的方向的移动，并且其中所述比较是基于所述相对角的。摘要附图为图23。



CN 104583902 A

1. 一种用于识别由影像流中被追踪的对象 (H,H',01,02) 所作的手势的计算装置 (100,200),所述计算装置 (100,200) 包括存储器 (240) 和控制器 (210),其中所述控制器 (210) 被设置成 :

确定被追踪对象 (H,H',01,02) 的移动向量 ;

将移动向量与手势的参考库进行比较 ;并由此识别匹配移动向量的手势 ;

其中所述移动向量包括具有由相对角 (α) 所识别的方向的至少一部分移动,并且其中所述比较基于所述相对角 (α)。

2. 根据权利要求 1 所述的计算装置 (100,200),其中所述控制器 (210) 还被设置成确定基线方向,用于适应所述相对角或用于适应待比较的参照角 (β)。

3. 根据权利要求 1 所述的计算装置 (100,200),其中所述控制器 (210) 还被设置成遍及状态图例如杜林机 (1800,1900,2000,2200),其中转换对应于所述移动向量的元素并且至少一个状态对应一动作 ;其中控制器根据移动向量的元素执行从一状态到另一状态的转换,直到通过到达对应的状态来识别手势。

4. 根据权利要求 3 所述的计算装置 (100,200),其中所述状态图包括至少一个稳定状态,其中稳定状态是对应一动作的状态或初始状态,和

其中所述控制器 (210) 还被设置成当状态图中对于移动向量的元素不存在其它转换时返回之前的稳定状态。

5. 根据权利要求 4 所述的计算装置 (100,200),其中所述控制器 (210) 还被设置成若没有其它部分移动在超时阈值 (T1) 内被检测到。

6. 根据权利要求 1-5 中任一权利要求所述的计算装置 (100,200),其中所述控制器 (210) 还被设置成当检测到手势时预先执行对应假设手势的动作,其中控制器 (210) 被设置成假设被检测的手势是假设的手势,如果目前可见的状态图的状态在待检测的手势的转换途中。

7. 根据权利要求 6 所述的计算装置 (100,200),其中所述控制器 (210) 还被设置成当达到待检测的手势的状态时呈现或产生对应于所假设手势的动作的结果。

8. 根据权利要求 1-7 中任一权利要求所述的计算装置 (100,200),其中所述状态转换图包括至少一个稳定状态在通往另一个稳定状态的途中,使得第一手势能够由用于第二手势的移动的子序列限定。

9. 根据权利要求 3-8 中任一权利要求所述的计算装置 (100,200),其中所述控制器 (210) 还被设置成通过遍及各自与多个对象的每一个有关联的多个状态图来识别多个对象的手势,其中所述控制器 (210) 还被设置成通过遍及所关联的状态图来识别所述多个被追踪对象的手势,其中,所关联的状态图中的转换对应所关联的待追踪对象的移动向量的元素。

10. 根据权利要求 3-8 中任一权利要求所述的计算装置 (100,200),其中所述控制器 (210) 还被设置成通过遍及状态图来识别多个对象的手势,其中每个转换对应于多个对象的移动。

11. 根据权利要求 1-10 中任一权利要求所述的计算装置 (100,200),其中所述控制器 (210) 还被设置成确定匹配被比较的参照角 (β) 的角 (α),如果被比较的角 (α) 在所述

参照角（β）的容许误差(φ)内。

12. 根据权利要求 1-11 中任一权利要求所述的计算装置 (100,200)，其中所述控制器 (210) 还被设置成：

确定移动的距离 (dist)；

确定移动的距离是否在第一距离阈值 (V1) 之上；如果是，

将移动的角（α）与参照角（β）进行比较。

13. 根据权利要求 1 到 12 中任一权利要求所述的计算装置 (100,200)，其中所述移动向量还包括取自由下述构成的组的至少另一量度：深度、旋转、转动角度、移动的距离和移动的速度。

14. 根据权利要求 12 所述的计算装置 (100,200)，其中所述控制器被设置成区分快速移移动和慢速移动。

15. 根据权利要求 12 或 13 所述的计算装置 (100,200)，其中所述控制器被设置成区分短移动和长移动。

16. 根据权利要求 1-15 中任一权利要求所述的计算装置 (100,200)，其中所述控制器被设置成根据待追踪对象的大小、与待追踪对象的距离、显示屏的大小和摄像机的视野确定距离。

17. 一种用在计算装置中用于识别由影像流中被追踪的对象 (H, H', O1, O2) 所作的手势的方法，所述方法包括：

确定被追踪的对象 (H, H', O1, O2) 的移动向量；

将移动向量与手势的参考库进行比较；并借此识别匹配移动向量的手势，

其中所述移动向量包括具有由相对角（α）识别的方向的至少一部分移动，并且其中所述比较基于所述相对角（α）。

18. 一种编码有指令 (31) 的计算机可读存储介质 (30)，其中当被加载到处理器上或在处理器上执行时，执行权利要求 17 所述的方法。

改进的手势的识别

技术领域

[0001] 本申请涉及用于识别对象所做手势的方法、计算机可读介质和计算装置，特别涉及用于改进识别视频流中由对象所做手势的方法、计算机可读介质和计算装置。本申请还涉及用于识别由多个对象所做的手势的方法、计算机可读介质和计算装置。

技术背景

[0002] 开发人员一直试图在诸如游戏机、计算机和智能手机的当代设备成功地实现基于视觉的控制。大多数的尝试在提供充分有效的实际操作于所有现实生活场景的控制系统上并不成功。下文给出了所述系统的一些示例。

[0003] 美国公开号 US2011299737 的美国专利申请披露了基于视觉的手部动作识别系统及其方法。在一实施例中，首先根据连续的手部影像识别手部姿势。如果手部姿势匹配起始姿势，系统随后将连续手部影像分成多个影像群组并计算这些影像群组的运动向量。这些运动向量的分布与多个三维运动向量的直方图等化进行比较，以确定每个影像群组的对应运动。例如，对应的运动可以是左移的动作、右移的动作、上移的动作或下移的动作。最后，这些对应运动的组合被定义为姿势，并且然后执行映射该姿势的指令。

[0004] 公开号 WO09128064 的国际专利申请披露了与电子显示相关的与电子装置进行人机互动的方法，包括捕捉定位于输入装置上的至少一手的影像，从影像追踪手的位置或姿势；由基于与输入装置互动的互动切换至指向装置仿真，来响应检测到用手进行的手势，及基于追踪而仿真指向装置，而手不再进行手势。

[0005] 美国专利号 US7,970,176 的美国专利披露了一种识别使用者姿势以便用于交互式游戏应用的方法。获得了使用者的视频摄像机影像，并且在影像中识别出使用者身体的特征点位置。相似性测度用于将影像中的特征点位置和手势库进行比较。库中的手势对应最大计算的相似性测度，其大于被识别为使用者的手势的手势的阈值。所识别的手势可与虚拟游戏环境中使用者的运动成一体，并且向使用者提供视觉反馈。

[0006] 公开号 GB2474536 的英国专利申请披露了使用者如何基于电脑视觉来控制电子装置（电视、DVD 机、PC、手机、相机、STB）。影像传感器捕捉一序列的视野影像。处理器接收到影像序列；检测在影像中至少一对象的运动；将形状识别算法（如轮廓检测）应用于至少一运动对象上，通过结合来自于对象的至少两个影像的信息确认该对象是使用者的手；和，追踪对象以检测用于控制装置的控制手势。形状识别可与运动检测一起施用或是在移动检测之前施用。在第一阶段，可能检测到诸如像挥手的初始手势。在较差的照明条件下，可能主要基于运动检测来识别使用者的手。使用者手势可控制光标移动和操作，选择并操作对象（如图标），或提供按钮点击仿真，例如鼠标点击命令。影像传感器可以是诸如网络摄像机的 2D 摄影机或 3D 摄影机，并可以与装置成一体或在装置外部或是 IR 敏感的。

[0007] 由所述系统提供的手势识别太慢而没有效果。

[0008] 此外，现有技术没有考虑摄影机尤其是在移动装置中可能不会与使用者完美地对准，这可导致手势的错误解释。

[0009] 另一主要缺点是现有技术系统中所涉及的计算的复杂性，需要大量的计算资源。

[0010] 因此，需要一种识别由视频流中的对象所作手势的方式，能够适应摄像机和使用者之间的失调。

[0011] 此外，极需要追踪在影像串流中的对象而并不需要庞大的计算资源的方式。

发明内容

[0012] 本申请的一个目的是通过提供用于识别由影像串流中被追踪的对象所作的手势的计算装置来克服上述问题，所述计算装置包括存储器和控制器，其中控制器被设置成：确定被追踪对象的移动向量，将移动向量与手势的参考库进行比较，并借此识别匹配移动向量的手势，其中所述移动向量包括至少一部分具有由角度识别的方向的移动，并且其中所述比较是基于所述角度的。应当指出，在一个实施例中，角度或方向是相对于其它向量的且并非是绝对角度或方向。这解决了必须将摄影机与待追踪的对象对准或者根据摄影机和待追踪的对象的当前对准调整手势的问题。应当了解，存在该问题并且识别该问题也需要创造性的推理。

[0013] 所述计算装置不需要大量的计算资源，这是因为通过使用（部分）移动的方向的角度来识别和存储手势这一巧妙的方式。

[0014] 在一个实施例中，控制器还被设置成穿过状态图，例如杜林机，其中转换对应于所述移动向量的元素，并且至少一个状态对应于一动作；借此控制器根据移动向量的元素执行从一状态到另一状态的转换，直到识别到手势到达对应的状态。

[0015] 在一个实施例中，计算装置是移动通讯终端。

[0016] 在一个实施例中，计算装置是因特网平板电脑或便携式电脑。

[0017] 在一个实施例中，计算装置是游戏机。

[0018] 在其它实施例中，计算装置是具有用户界面的打印机、具有非接触式用于界面的摄影机，例如水下摄影机，其中触摸显示或其他用户界面不会工作（由于防水壳或由于水的物理特性），具有非接触式用户界面的（智能）表。

[0019] 视频流也可通过下载的或串流的视频流提供，可能不源于装置的摄影机。

[0020] 本申请的另一目的是通过提供一种用于计算装置用于识别影像串流中由被追踪的对象所作的手势的方法来克服上述问题，所述方法包括：确定被追踪对象的移动向量，将移动向量与手势的参考库进行比较，并借此识别匹配移动向量的手势，其中所述移动向量包括至少一部分具有由角度识别的方向的移动，并且其中所述比较是基于所述角度的。

[0021] 本申请的另一目的是通过提供一种计算机可读介质来克服上述问题，所述计算机可读介质包括指令，所述指令当被加载到控制器（例如处理器）和由控制器（例如处理器）执行时执行本文所述的方法。

[0022] 本发明的发明人在创造性和仔细推理后认识到，透过巧妙的主要聚焦于（部分）移动的方向和状态图的使用，可以实时快速地识别手势而不需要大量的计算资源。巧妙地聚焦于（部分）移动的方向是主要比较方向的相对角度，不像现有技术的系统比较绝对角度。这使得能够独立于摄像机和待追踪的对象的当前对准来识别手势。

[0023] 本发明可用于具有用户界面的计算装置的控制系统，例如智能手机、因特网平板电脑、电脑和游戏机。

[0024] 所披露实施例的其它特征和优点会由下文的详细描述、由所附权利要求以及附图揭示。一般而言，权利要求中使用的所有术语将根据其技术领域中的普通含义来解释，除非本文另有明确定义。

[0025] 所有提及“元件、装置 / 设备、部件、方法、步骤等”之处被公开解释为是指元件、装置 / 设备、部件、方法、步骤等的至少一实例，除非另有明确描述。本文所披露的任何方法的步骤不必然按所披露的确切顺序执行，除非有明确说明。

附图说明

[0026] 参考附图，将进一步详细描述本发明，其中：

[0027] 图 1A 和 1B 是根据本发明的每一计算装置的示意图；

[0028] 图 2 是根据本发明的计算装置的部件的示意图；

[0029] 图 3 是根据本发明的计算机可读存储器的示意图；

[0030] 图 4A 示出了待被追踪对象的示例影像，图 4B、4C 和 4D 各自示出了根据本发明的经处理对象版本的表示；

[0031] 图 5 示出了根据本发明的实施例的如何可确定描述符的示意图；

[0032] 图 6 示出了根据本发明的实施例已被分配描述符的待被追踪对象的示图；

[0033] 图 7 示出了根据本发明实施例的已被分配描述符的待被追踪移动对象的示图；

[0034] 图 8 示出了根据本发明实施例确定相对距离的直方图；

[0035] 图 9 示出了根据本发明的示例实施例在实验中确定相对距离的直方图；

[0036] 图 10 示出了根据本发明实施例的示出一般方法的流程图；

[0037] 图 11 示出了根据本发明实施例已被分配描述符的待被追踪的两移动对象的示图；

[0038] 图 12 示出了根据本发明实施例确定两对象相对距离的直方图；

[0039] 图 13 示出了根据本发明实施例已被分配描述符的待被追踪的移动对象沿着与视线平行的方向的示图；

[0040] 图 14 示出了根据本发明实施例确定平均位置和相对距离的线性拟合的示图；

[0041] 图 15A 和 15B 分别示出了根据本发明实施例基于直角坐标确定平均位置和相对距离的线性拟合的示图；

[0042] 图 16 示出了根据本发明实施例用于追踪对象的一般方法的流程图；

[0043] 图 17 示出了对象作手势的示意性示图；

[0044] 图 18 示出了根据本发明实施例用于识别手势的状态图或杜林机的示意性示图；

[0045] 图 19 示出了根据本发明实施例用于识别手势的状态图或杜林机的示意性示图；

[0046] 图 20 示出了根据本发明实施例用于识别手势的状态图或杜林机的示意性示图；

[0047] 图 21 示出了两对象作手势的示意性示图；

[0048] 图 22 示出了根据本发明实施例用于识别手势的状态图或杜林机的示意性示图；

[0049] 图 23 示出了根据本发明实施例用于识别手势的一般方法的流程图。

具体实施方式

[0050] 现在将参照附图在下文更充分地描述本发明揭示的实施例，其中示出了本发明的

某些实施例。然而，本发明可以许多不同的形式来实施，并且不应该被解释为限于这里所阐述的实施例，而是通过示例的方式提供这些实施例，以使本本发明将是彻底和完整的，并且将充分地传达本发明范围给本领域技术人员。相同的数字代表相似的组件。

[0051] 图 1 总体示出了根据本文实施例的计算装置 100。在一实施例中，计算装置 100 被配置用于网络通讯，无线或有线。这样的计算装置 100 的例子是 :个人计算机、桌上型或笔记型计算机、因特网平板计算机、移动通讯终端诸如移动电话、智能电话、个人数字助理、游戏控制台。两个实施例将被作为范例和描述为图 1A 中的智能手机和图 1B 中的便携式电脑 100。

[0052] 参照图 1A，形式为智能手机 100 的移动通讯终端包括壳体 110，其中配置有显示器 120。在一个实施例中，显示器 120 是触控显示屏。在其它实施例中，显示器 120 是非触控式显示器。此外，智能型手机 100 包括两个键 130a、130b。在本实施例中，有两个键 130，但是任何数量的键是可能且取决于智能型手机 100 的设计。在一实施例中，智能型手机 100 被设置成显示和操作触控显示器 120 上的虚拟键 135。应当注意，虚拟键 135 的数量依赖于智能型手机 100 的设计和智能型手机 100 上执行的应用程序。智能型手机 100 还配备了摄影机 160。摄影机 160 是数字摄影机，被设置成通过将影像记录于电子影像传感器（未绘示）来取得视频或静止照片。在一个实施例中，摄影机 160 是外部摄影机。在一个实施例中，摄影机可替换为提供影像串流的来源。

[0053] 参照图 1B，笔记型计算机 100 包括显示器 120 和壳体 110。壳体包括控制器或 CPU（未绘示）和一或更多个计算机可读取储存媒体（未绘示），如储存单元和内部存储器。储存单元的范例是磁盘驱动器或硬盘驱动器。计算装置 100 还包括至少一数据端口。数据端口可以是有线的和 / 或无线的。数据端口的范例是 USB（通用串行总线）端口、以太网络端口或 WiFi（根据 IEEE 802.11 标准）端口。数据端口被配置成使得计算装置 100 能够与其它计算装置或服务器连接。

[0054] 笔记型计算机 100 还包括至少一个输入单元，如键盘 130。输入单元的其它例子是计算机鼠标、触控板、触控屏幕或操纵杆等。

[0055] 笔记型计算机 100 还配备有摄影机 160。摄影机 160 是数字摄影机，配置成通过将影像记录在电子影像传感器上来取得（图中未绘示）视频或静止影像。在一个实施例中，摄影机 160 是外部摄影机。在一个实施例中，摄影机可替换为提供影像串流的来源。

[0056] 图 2 绘示了根据图 1 的计算装置的总体结构示意图。计算装置 100 包括控制器 210，其负责计算装置 200 的整体运作且优选通过任何商业上可用的 CPU（“中央处理单元”）、DSP（“数字讯号处理器”）或任何其它的电子可程序化逻辑装置实施。控制器 210 可使用赋能硬件功能的指令来实施，例如通过使用在通用或专用处理器中的可执行计算机程序指令，其可被储存在计算机可读取储存媒体（磁盘、存储器等）240 以由此种处理器执行。控制器 210 被设置成自存储器 240 读取指令并执行这些指令来控制计算装置 100 的运作。存储器 240 可使用用于计算机可读取存储器例如 ROM、RAM、SRAM、DRAM、CMOS、FLASH、DDR、SDRAM 的任何通常已知技术或一些其它存储器技术来实现。存储器 240 通过控制器 210 被用于不同目的，其中之一是用于为计算装置 200 中的各种软件模块储存应用程序数据和程序指令 250。这些软件模块包括实时操作系统、用户界面 220 的驱动程序、应用程序处理器以及各种应用程序 250。应用程序是一组指令，当由控制器 210 执行时，控制计算装置 200

的运作。应用程序 250 可以包括讯息应用程序，如电子邮件、浏览器应用程序、媒体播放器应用程序，以及其它各种应用程序 250，如语音通话、视频通话、文件阅读和 / 或文件编辑的应用程序、实时讯息应用程序、日历应用程序、控制面板应用程序、一个或更多的视频游戏、记事本应用程序、语音通话应用程序、短讯服务应用程序、定位应用程序。

[0057] 计算装置 200 还包括用户界面 220，其在图 1A 和图 1B 中的计算装置中由显示器 120 和键 130、135 组成。

[0058] 计算装置 200 还包括射频接口 230，其适于使得计算装置通过使用不同的无线电频率技术通过无线电频带而与其它装置通讯。这种技术的例子是 IEEE 802.11、IEEE 802.11s 和 Bluetooth®。可在计算装置 100 实现的无线电技术的其它示例有 W-CDMA、GSM、UTRAN、LTE、和 NMT 等。

[0059] 计算装置 200 还配备有摄影机 260。摄影机 260 是数字摄影机，被配置成通过将影像记录于电子影像传感器来取得（未绘示）视频或静止影像。

[0060] 数字摄影机 260 的分辨率通常会被将光线转换成离散讯号的影像传感器（通常是 CCD 或 CMOS 传感器芯片）所限制。传感器是由基本上计数撞击传感器的光子数量的大量受体所组成。在给定点时传感器上的影像越亮，该像素读出的值越大。影像中得到的像素数量决定了它的“像素数”。例如，640×480 的影像具有 307,200 像素或约 307 千像素 (kilopixels)；3872x2592 的影像具有 10,036,224 像素或者为约 10 兆像素 (megapixel)。

[0061] 摄影机 260 是配置成拍摄数字影像或图片。有许多不同的方式可做到这点。在称为单拍的方式中，摄影机 260 的传感器（未绘示）被暴露于通过摄影机镜头的光线（图中未明确绘示）。单拍捕捉系统使用一个 CCD（电荷耦合装置）或通过分束器曝光于同一影像的三个独立的影像传感器（分别用于红色、绿色和蓝色三个主要添加色）。

[0062] 摄影机 260 最好是能够作为数字视频摄影机。数字视频摄影机通常采用两种不同的影像捕捉格式之一：隔行扫描和非隔行扫描 / 连续扫描。隔行扫描摄影机以交替的线组记录影像：扫描奇数行，然后扫描偶数行，然后再次扫描奇数行等等。一组奇数或偶数行称为“场”，两场相反奇偶性的连续配对称为帧。非隔行扫描的摄影机将每个帧记录为独特的，在同一时刻及时捕捉所有的扫描线。因此，每秒钟对于相同数目的帧，隔行扫描的视频捕捉以连续扫描视频的两倍频率取样场景动作。连续扫描摄影机通常会产生稍微较清晰的影像。然而，动作可能不如使用每秒 50 或 59.94 场的隔行视频顺畅，尤其是如果它们采用每秒 24 帧的影像标准。

[0063] 数字视频摄影机 260 提供数字视频流 265，也称为作为数字视频。数字视频包括以固定速率快速连续显示的正交位图数字影像的序列或流。在视频的情境中，这些影像被称为帧。我们以每秒帧数 (FPS) 来测量帧所显示的速率。由于每个帧是正交的位图数字影像，因此它包括像素栅格。如果它的宽为 W 像素而高为 H 像素，我们称帧的大小为 WxH。像素只有一种属性，即它们的颜色。像素的颜色是由固定的位数表示。位数越多可以再现越多的颜色的微妙变化。这就是所谓的视频的色彩深度 (CD)。范例视频的持续时间 (T) 为 1 小时 (3600 秒)，帧大小为 640×480(WxH)，色彩深度为 24 位，帧速率为 25fps。这个范例视频具有以下属性：每帧像素 = 640*480 = 307,200；每帧位数 = 307,200*24 = 7,372,800 = 7.37Mbits；位速率 (BR) = 7.37*25 = 184.25Mbits/sec；视频大小 (VS) = 184Mbits/sec*3600 秒 = 662,400Mbits = 82,800Mbytes = 82.8Gbytes。在一个实施例中，摄影机

260 被设置成在每秒 8 帧的帧速率或 16、24、32 或更高的帧速率的任何一者运作。

[0064] 摄影机 260 被可操作地连接到控制器 210，以提供视频流 265（即捕捉的影像序列）给控制器，用于进一步处理，可能用于及 / 或根据应用程序 250 中的一或几个。

[0065] 在一个实施例中，摄影机 260 是外部摄影机或影像串流来源。

[0066] 提及“计算机可读取储存媒体”、“计算机程序产品”、“有形实施的计算机程序”等或“控制器”、“计算机”、“处理器”等应被理解为不仅包括具有不同体系结构的计算机，如单 / 多处理器架构和依序 (Von Neumann) / 并行架构，亦包括专门电路，如场可程序化门阵列 (FPGA)、应用程序专门电路 (ASIC)、讯号处理装置和其它装置。提及计算机程序、指令、代码等应被理解为包括可程序化处理器的软件或固件，例如硬件装置的可程序化内容，不论是处理器的指令或是固定功能装置、门阵列或可程序化逻辑装置等的配置设定。

[0067] 图 3 绘示如上所述的计算机可读取媒体的示意图。在本实施例中，计算机可读取媒体 30 是记忆卡，如通用串行总线 (USB) 的卡。USB 卡 30 包括具有接口的壳体 33，例如连接器 34 和记忆芯片 32。记忆芯片 32 是闪存，也就是可电性抹除和重新程序化的非挥发性数据储存装置。记忆芯片 32 是以指令 31 程序化，当加载到诸如处理器之控制器时（可能通过接口 34），执行根据上述揭示的实施例的方法或程序。USB 卡被配置成连接至诸如根据图 1 之终端的读取装置且由该读取装置读取，以便将指令加载到控制器。应当指出的是，计算机可读取媒体也可以是其它的媒体，如光盘、数字视频光盘、硬盘或其它常用的储存技术。指令也可经由无线接口自计算机可读取媒体下载以加载到控制器。

[0068] 下面将参照附图揭示一种改善的追踪对象的方式。将聚焦于影像数据进行范例说明，但应该清楚的是，该处理是部分或全部执行于包括控制器的计算装置，如参考图 1 和 2 于上文所揭示，或通过执行存储在计算机可读取媒体上的指令执行，如参考图 3 之揭示。

[0069] 图 4A 绘示被追踪对象的影像，在本例中是手。影像是由摄影机（图 4 中未绘示，但在图 2 中标号为 260）提供给计算装置（在图 4 中未绘示，但在图 2 中标号为 200）的控制器（图 4 中未绘示，但在图 2 中标号为 210）的影像串流（在图 4 中未绘示，但在图 2 中标号为 265）的一部分。影像串流可由外部摄影机或诸如因特网上之串流视频的另一影像串流来源提供给控制器 (210)。

[0070] 为了能够进行有效地对象追踪，重要的是该影像包含足够的细节。因此，相机的分辨率是重要的。对于远离相机的对象，分辨率就变得更加重要，对于长距离追踪，具有更高分辨率的摄影机是优选的。对于短距离追踪，低分辨率的相机就足够了。使用本文所揭示的方式，160X120 的影像分辨率已被证明是足够能够成功地追踪对象。

[0071] 此外，为了准确和精确追踪，高帧速率是优选的，因为这允许详细的对象追踪。这种高帧速率对于追踪方法的性能提出很高的要求，因为每个追踪操作必须在收到下一帧之前完成。本文所揭示的方式已被证明使用高于 30fps 和低至 8fps 的帧速率能够成功地追踪对象。

[0072] 常见的影像尺寸是视频图形阵列 (VGA) 格式，像素为 480×640，即总共为 307200 像素。这样的数据集对执行大量的计算是不切实际的，因此控制器被设置成通过确定影像中的轮廓线来减少影像的数据集，藉此过滤对象内的所有点或像素。这样的轮廓线可能会利用边缘检测确定。

[0073] 边缘检测是影像处理、机器视觉和计算机视觉的知名工具，旨在识别数字影像中

影像亮度急剧变化的点，或者更正式的说法是具有不连续性。一种用于边缘检测的常用方法是 Canny 边缘检测器，其是使用多级算法来检测影像中的广泛的边缘范围的边缘检测运算子。

[0074] 在这个例子中，控制器被配置成使用 Canny 运算子确定轮廓线，藉此产生影像或第一画面（第一是指当前正被处理）的 Canny 线。图 4B 绘示图 4A 中的影像产生的轮廓线 CL 的视图。正如可以看到的，影像的数据集已经大大降低了。

[0075] 为了提高轮廓线确定的速度，控制器可以配置以使用 Canny 运算子来利用速度的增加。这将导致较低的精确度，但提高了处理速度。降低的精确度已被证明有较小的后果。

[0076] 应该指出的是，存在有检测用于产生轮廓线的边缘的其它方式且为本发明范围的一部分。

[0077] 为了进一步减少数据集，控制器被配置以过滤掉静止的所有轮廓线。（轮廓）第一影像的轮廓线 CL 与影像串流中的前一影像的轮廓线进行比较，以确定是否有轮廓线的覆盖或匹配。此类匹配轮廓线代表静止的对象且将被过滤掉，藉此进一步减少影像的数据集。图 4C 绘示所有静止的轮廓线被过滤掉后的手的轮廓线 CL。

[0078] 为了进一步减少数据集，控制器被配置成过滤出不太可能成为被追踪对象的一部分的轮廓线。

[0079] 为了确定轮廓线不是对象的一部分，将轮廓线的属性与对象的假定或已知属性进行比较。

[0080] 例如，如果是要追踪矩形对象，所有环状或圆形的轮廓线会被过滤，因为它们不太可能是矩形对象的一部分。类似地，如果是要追踪矩形的对象，所有圆形或弯曲的线会被过滤掉。

[0081] 为了能够过滤某些轮廓线，控制器可被设置成根据该轮廓线的曲率过滤掉该轮廓线。根据待过滤的对象，曲率可由设计者设定。如果在过滤圆的追踪对象的情况下，所有曲率低于曲率阈值以下的轮廓线会被过滤掉，例如 1 或 5 或 1 至 5 度。

[0082] 在过滤矩形的追踪对象的情况下，所有曲率高于一曲率阈值的轮廓线会被过滤掉，例如 1 或 5 或 1 至 5 度。

[0083] 曲率可以根据下文揭示的描述符而确定或通过比较轮廓在线的两个或更多个点的曲率。

[0084] 为了过滤出矩形对象，轮廓线没有角度的所有对象可能会被过滤掉。

[0085] 要过滤不是手的一部分的对象，透过比较轮廓线的两侧上的颜色，控制器可配置以确定轮廓线不是手的一部分。如果轮廓线两侧的颜色都不是皮肤的颜色，轮廓线不是手的一部分。

[0086] 待比较的皮肤颜色可自视频流中的第一影像确定，其中对象被识别且其颜色被决定。这使得本文的方式可以用于所有的皮肤颜色且可用于手套或其它硬件或者纹身（指甲花或其它方式）及珠宝。

[0087] 通过确定追踪的前几个帧中的对象颜色，允许过滤轮廓线，也允许追踪其它的对象。

[0088] 应当注意到，基于如上面列出的准则组合，控制器可被配置成过滤出不太可能是要追踪对象的一部分的轮廓线。例如，手不太可能包含圆形的轮廓线和具有尖角的轮廓线。

[0089] 为了进一步减小数据集，控制器被配置成确定在影像中 / 在轮廓线上的哪些点是感兴趣要追踪的。为了识别感兴趣的点，控制器被设置成指派描述符至每个点。描述符是点的特征，可用于识别该点。

[0090] 在一个实施例中，控制器被设置成将三个描述符指派到每个点。图 4D 绘示轮廓线的放大图，可以看出指尖的轮廓线 CL。三个描述符 D1、D2 和 D3 是以图 4D 中的白色箭头表示。这三个描述符是：

[0091] D1 ＝点的正距的角；

[0092] D2 ＝点右侧的曲率；及

[0093] D3 ＝点左侧的曲率。

[0094] 可选地，控制器可配置以附加地或替代地指派描述符，基于点的颜色或该点的轮廓线的强度。在一个实施例中，控制器被设置成根据下述而仅指派两个描述符 D1 和 D2：

[0095] D1 ＝点的正距的角及

[0096] D2 ＝点附近的曲率。

[0097] 指派较少的（二个）描述符加快处理速度，但降低了精确度，因为这些描述符没有如使用更多的描述符时那样清楚地识别点。增加的精确度能够有效的过滤，这减少了要处理的数据集。

[0098] 在一个实施例中，控制器被设置成使用 Sobel 演算子确定正距的角。Sobel 演算子被用在影像处理中，特别是在边缘检测算法。从技术上讲，它是离散的微分演算子，计算影像强度函数的梯度的近似值。在影像中的每个点，Sobel 演算子的结果是对应的梯度向量或此向量的正距 (normal)。Sobel 演算子是以小的、可分离、和整数值的过滤器在水平和垂直方向上基于卷积影像，因此在计算上相对便宜。Sobel 演算子的操作包括从直角到极坐标的转换。应当指出的是，有确定正距的角度的其它方式存在，并且是本发明范围内的一部分。在一个实施例中，正距的角是取自在执行边缘检测时已经执行的计算。这是有益的，因为它减少了需要执行的计算数量。

[0099] 图 5 绘示如何可在轮廓线 CL 上的点之间确定角度 α1、α2、α3。角度 α1、α2、α3 对应点 P1 和最接近的两个相邻点 P12 和 P3 的正距，且在一个实施例中是使用 Sobel 演算子确定。两点之间的距离被表示为△ step。根据图 5 中的术语，描述符可以根据以下确定：

[0100] D 1 ＝ α1[＝ Sobel(P 1)]；

[0101] D2 ＝ (α1 - α2)/ △ step[＝ (Sobel(P1)-(Sobel(P2))/ △ step]；和

[0102] D3 ＝ (α1 - α3)/ △ step[＝ (Sobel(P1)-(Sobel(P3))/ △ step]。

[0103] 这提供了一种计算或确定描述符和曲率的简单方式，该方式不要求复杂的操作，特别是对于确定曲率而言，因为已经确定了相邻点的正距的角度。

[0104] 使用两个描述符，描述符 D1、D2 可如下确定：

[0105] D1 ＝ α1[＝ Sobel(P1)]；和

[0106] D2 ＝ (α2 - α3)/2 △ step[＝ (Sobel(P2)-(Sobel(P3))/2 △ step]。

[0107] 为了进一步减小数据集，控制器被设置成过滤掉难以追踪的点。图 6 绘示手的影像，其中 4 个点 P1、P2、P3 和 P4 已被标记为描述符（以箭头表示）。两个点 P3、P4 位于食指一侧，两点 P1、P2 位于食指的尖端上。如果根据上述确定，两个侧点 P3、P4 的描述符非

常相似。这一点在图 6 中通过具有相同角度的对应描述符的箭头绘示。然而，指尖上两点 P1、P2 的描述符彼此不同，且不同于侧点 P3、P4 的描述符。这在图 6 中是以具有不同的角度的对应描述符箭头绘示。这种不同于影像中的大部分点的点较容易追踪，控制器被设置成过滤掉其描述符不异于其它点的描述符的所有点。

[0108] 在一个实施例中，通过计算与该点相比之前一图片中具有类似或相同描述符的其它点的数目，控制器被设置成过滤掉非描述点。如果比较的点的数量超过一阈值的计数值，被比较的点会被丢弃或过滤。阈值计数可能代表绝对计数，例如 100 或 500，或者相对计数（进行比较的点的数量和点的总数量之间的配额），例如 1/5、1/10、1/20 或 1/40。

[0109] 为了分出噪音并进一步减小数据集，控制器被设置成过滤掉移动太多的点。过滤是通过控制器将影像中轮廓线上的所有点与前一影像中轮廓线上的所有点比较，显现太多不同的点对会被丢弃而获得。在一个实施例中，如果在描述符△ D1、△ D2、△ D3 中的变化根据下述式子是低于或在阈值内，则控制器被设置成保留一点对：

[0110] △ D1< ＝ T1，例如 T1 ＝ 11.25 度；

[0111] T2L< ＝△ D2< ＝ T2H，例如描述符值 D2 的 T2L ＝ -50％和 T2H ＝ +100％；以及

[0112] T3L< ＝△ D3< ＝ T3H，例如描述符值 D3 的 T3L ＝ -50％和 T3H ＝ +100％，

[0113] 其中 T1 是描述符 D1 中所允许变化的阈值，T2L 是描述符 D2 中所允许变化的较低阈值，T2H 是描述符 D2 中所允许变化的较高阈值，T3L 是描述符 D3 中所允许变化的较低阈值，T3H 是描述符 D3 中所允许变化的较高阈值。

[0114] 应当注意的是，阈值 T1、T2H、T2L、T3H 和 T3L 愈小或愈窄，如上面所讨论的，用于比较非描述点的阈值计数愈高。

[0115] 在本实施例中，允许两个曲率描述符 D2、D3 的变化因子为 2。应该指出的是，其它因子也是可能的，例如 3 或 1.5。

[0116] 保留的数据集现在已经减少至包括感兴趣的点。

[0117] 要确定被追踪对象的水平和垂直移动，控制器被设置成将数据集中的所有点与数据集中的所有点比较，以确定移动。在一个实施例中，控制器是经比较以根据描述符中的变化与保留的像素对比较。

[0118] 图 7 绘示如何执行移动比较。该图显示了两只手，每只手具有两个点，左手 H 是 P1、P2，右手 H’ 是 P1’、P2’。在图 7 的例子中，右手 H’ 是当前影像的手，左手 H 是先前影像的手。在图 7 中，手 H 向右（手 H’）移动了一定的距离，这是以手 H 绘示，H’ 被绘制为彼此间有一距离。应该指出的是，虽然图 7 中每一手只显示了两个点 P1、P2，但在真实的例子中是处理更多的一些点。

[0119] 通过计算每个点对之间的距离，控制器将前一影像的数据集 P1、P2 中的所有点与当前影像中的数据集 P1’、P2’ 中的所有点进行比较。这样的距离是绘示为一点自前一位置 P1 至当前位置 P1’，且该距离表示为 P1-P1’。

[0120] 距离 P1-P1’ 是储存在每个距离的矩阵中。该距离是一两维实体，具有 X（水平）和 Y（垂直）分量，且该矩阵被产生以使每个单元格对应于由对应的 X 和 Y 坐标或分量所给定的距离。对于每一发生的对应距离之经计算距离，矩阵中的每个单元格增加一。图 8 绘示了图 7 中所绘示情况的一维范例，其中对于每一距离，绘制每个距离的发生情形。

[0121] 可以看出，有四个距离，其中两者相同，代表真正的移动。这两者是 P1 和 P1’ 及

P2 和 P2’之间的距离。另外两个是 P1 和 P2’及 P2 和 P1’之间的距离。真正的移动是由找出距离矩阵中的最大发生值而确定。在图 8 中，最大值是在最高棒，在本例中为两个距离 P1-P1’和 P2-P2’的栏。

[0122] 图 9 绘示了实验的二维表示，其中一对象已被追踪。可以看到在 X = -12，Y = 0 附近是峰顶中心。最大值表示该对象在负 X 轴方向已移动了 12 点或像素。

[0123] 在一个实施例中，控制器被设置成确定或计算在同一时间的相对距离，因为如果点对表现出过多的移动，该相对距离是确定的，但在此之后，可以确定的是点对不会表现出过多的移动。这减少了控制器需要遍历数据集的次数，藉此提高计算效率。

[0124] 使用上述揭示的方式减少数据集，然后确定最大可能的移动，对象的真实的水平和垂直移动可以只需要极少的计算资源的方式确定。

[0125] 实验显示适合于实现上述揭示方式的智能型手机能够利用低分辨率的摄影机及在光线不好的条件下（低到 20 勒克司）满意地追踪对象，藉此使这里的教导适合用于移动装置，如智能型手机、笔记型计算机、因特网平板等，以及在低于最佳环境中使用的固定式装置，如游戏机（可携式和固定式）。

[0126] 本文所教示的方法适合追踪用于实现装置之手势控制的对象，例如手。本发明的一范例实施方式是浏览应用程序，其中一只手从一侧移动到另一侧的手势表示与追踪的移动方向相同的“翻页”操作。在图 9 的例子中，对应的操作是翻至下一页面（假设 X 方向是往右，由此所确定的移动是向左移动）。

[0127] 本文所教示的方法亦适合追踪用于实现视线追踪的对象，例如眼睛。视线追踪技术可以用来收集有关阅读行为的数据，确定显示内容上的当前感兴趣的点，也用于以特定的眼球移动来控制装置（提供残障支持特别有用）。本发明的一范例实施方式是广告应用程序，其中兴趣点（即是使用者聚焦超过阈值的时点的一点或使用者重复聚焦的点）与特定主题相关联，且应用程序被设置成提出并显示有关特定主题的广告。在图 9 的例子中，应用程序将向左追踪眼睛移动（假设 X 方向是向右，由此所确定的移动是向左移动）。如果有一阵子没有检测到进一步的动作，且所检视内容包含左栏上的潜水文章，该应用程序可以下载和显示潜水装置的广告。

[0128] 图 10 绘示了根据本文教示的一种概略方法。在 1010，于计算装置接收影像串流。为了减少要处理的数据，计算装置在 1020 确定轮廓线并在 1030 过滤掉静止线。可选择地（如虚线表示），在 1035，计算装置过滤掉不可能属于被追踪对象的点。

[0129] 在 1040，计算装置确定并指派描述符到每个点。该描述符可以用于识别点。在 1044，为了减少要处理的数据集，计算装置过滤掉其描述符指示该点是非描述的点（很难与其它点区分）。在 1048，为了移除假动作和减少要处理的数据集，计算装置过滤出描述符指示已经有过大移动的点对。在 1050，计算装置确定数据集中各点对之间的相对距离，并在 1060 找到相对距离的最大发生值。最大相对距离对应于要被追踪的对象的移动。

[0130] 图 11 绘示了如何使用本文所揭示的方式追踪多个对象。在图 11 的范例绘示中，对象 01 和 02 每者是以三点识别，第一对象 01 是以 P1、P2 和 P3 识别，第二对象 02 是以 P4、P5 和 P6 识别。如上所揭示，参照图 4 至图 7，对应于影像的数据集已经减小。

[0131] 在图 11 中，当控制器确定第一和第二对象 01 和 02 的所有点对的相对距离时，它产生如图 12 所示的直方图（一维），显示图 11 中 6 个点 P1-P6 的相对距离的范例直方图。

为了清楚起见，假移动的相对距离尚未于图 12 的描点中识别。此外，图 11 和图 12 的相对距离是仅供说明之用，图 11 的距离和图 12 中的描点之间的相关性可能不准确。

[0132] 在图 12 中可以识别两个极大值，一个用于第一对象 01，一个用于第二对象 02。由此可以确定，01 已经移动一定的距离，02 已经移动另一距离，其中第一对象 01 较第二对象 02 移动得更远。

[0133] 因此，使用此处的教示，可能追踪多于一个的对象，而且除了找第二最大值的简单运算之外，不需要大量的额外计算。追踪多个对象可利于用来实现用于装置的多手势控制系统。它也可以被用来向视频游戏提供多个播放器的功能或为控制系统提供多个运算输入。

[0134] 例如，要在手势控制的操作系统中能够点击或选择动作或变焦操作，知道对象在影像深度上的移动或对象在影像串流中与视线平行的维度上的移动是有利的。在现有技术系统中，已经通过使用例如立体声或双摄影机或使用也称为测距摄影机的深度摄影机来确定深度。这对所使用的摄影机提出了特殊的要求，造成计算装置的成本增加，也限制配备有此种摄影机之装置或适合于处理此种摄影机所提供之延伸数据之装置的可用性。

[0135] 本文之教示揭示了适用于使用任何摄影机的方式，并可能因此利于使用于智能型手机、笔记型计算机、因特网平板、游戏机和其它计算装置。

[0136] 图 13 示意性地绘示对应对象（未具体绘示）的点 P1、P2 的范例移动。可以看出，这些点已经从第一位置 P1、P2 移到之后的第二位置 P1’、P2’。每个点 P1、P2 的移动长度 △ r1，△ r2 是与自焦点 F 的对象距离 R1、R2 成正比。如果距离 R1、R2 增加，它表示朝向摄影机的移动，如果距离 R1、R2 减小，它表示远离摄影机的移动。由于距离 R 也可以通过 $R^2 = X^2+Y^2$ 表示，其中 X 和 Y 是直角坐标或每个点的元素，本文教示之作者在深度推理后已经理解到，即使坐标系统的原点不与焦点 F 配合，可以使用 X 和 Y 元素而非半径 R。这简化了将要执行的大量计算。

[0137] 在一个实施例中，控制器被设置成确定属于已根据上述而减少之数据集的点对 P1、P1’和 P2、P2’的平均位置。这将产生用于点对的平均位置阵列。

[0138] 可选地，控制器可被设置成过滤掉与根据上述所揭示方式确定的检测的总移动相差太大的那些点。图 9 所示的例子中，总移动为 X = -12，Y = 0。

[0139] $P1_M$ 的平均位置通过 $P1_M = (P1+P1')/2$ 确定。

[0140] 控制器进一步被设置成产生用于点对 P1、P1’和 P2、P2’的相对距离△ r1、△ r2 的对应阵列，并将相对距离△ r1、△ r2 与平均位置 P1m、P2m 比较。图 14 绘示了相对距离△ r 与距离 R 至焦点 F 的绘图，图 15A 和图 15B 显示 X 和 Y 分量或坐标的对应图。

[0141] 控制器被设置成通过执行阵列中数据的线拟合，确定对应线 LR、LX 和 LY 的斜率。对于图 14 中的描点，产生的线 LR 可表示为：△ r = a+bR，其中 a 是截距，b 是线 LR 的斜率。

[0142] 斜率 b 是与影像的深度方向的移动成比例，并可以用来追踪在深度方向移动的对象，深度方向也称为 Z 方向。正斜率 (b>0) 表示向摄影机移动，负斜率 (b< =) 表示背离摄影机移动。

[0143] 在一个实施例中，控制器被设置成执行这种线拟合，以基于点对 P1；P1’，P2；P2’的 X 和 Y 分量决定斜率。图 15A 和 15B 绘示了得到的 X 和 Y 分量或坐标的线 LX 和 LY，经确定的用于线 LX 和 LY 的斜率可以由控制器使用以确定在 Z 方向的移动。对于仅在 Z 方向上

的移动，在 X 和 Y 分量的线性拟合 LX 和 LY 的斜率是相同的。

[0144] 可以由 X 和 Y 分量的线拟合确定焦点 F。在图 15A 和 15B 中，焦点 F 的坐标在相对距离为零时可以被确定，即当△ X ＝ 0 得到 PX，△ Y ＝ 0 得到 PY，其中 F ＝ (PX ;PY)。

[0145] 焦点 F 可用于识别对象并与其它的对象区别，因为 F 表示对象的中心点。因此，追踪多个对象时，焦点 F 可以用于识别对象，如参照图 11 和图 12 于上之揭示。

[0146] 应当注意到，通过从直角坐标到极坐标的有效转换，距离 R 和距离变化△ r 可以用来计算斜率及在 Z 方向的移动。

[0147] 在追踪多个对象时，R、X 和 Y 分量的点会显示几个趋势，一趋势将与每个对象关联，并将由控制器执行多个线拟合。每个线拟合的斜率对应相关联对象的移动。

[0148] 图 16 绘示了根据本文教示之追踪对象的概略方法。在 1610，接收包括点对的数据集。数据集优选根据上文教示而已经减小，但本文揭示之用来决定 z 方向移动的方法和方式也可用于以其它方式产生和减小的数据集。

[0149] 在 1620 确定点对的平均位置，以及在 1630 确定点对的相对距离，在 1640 确定斜率，例如通过执行相对距离与平均位置的线拟合。

[0150] 在 1650，根据斜率来确定 Z 方向上的移动。

[0151] 根据本文而追踪对象在 z 方向的移动方式是有益的，因为它需要很少的计算资源。再一主要的好处是它不要求特殊的摄影机或多个摄影机，并可用于被配置成接收影像的数据集或例如来自摄影机、内部或外部的影像串流的任何计算装置。

[0152] 通过结合上述揭示用于在垂直和水平方向上追踪对象的方法和方式及如何在与视线平行之方向追踪对象之教示，实现在三维空间中追踪对象的方式。这种结合可通过结合图 10 和图 16 的方法而达成，使得在 1610 收到的点对是过滤后所余的点对，例如在 1048 过滤表现出过大移动的点对。

[0153] 为了确定和识别由诸如手之追踪对象所执行的手势，控制器被设置成确定追踪对象的移动向量并将移动向量与向量库进行比较。

[0154] 图 17 绘示了手 H 的示意图，手 H 通过在三个方向上、右、下移动而执行手势。在图 17 中，手势由三个部分动作代表，1710 上、1720 右、1730 下。对于每个移动，部分移动的方向 1710-1730 在图 17 中是通过在括号内的方向（度）所示。移动向量因此包括每者具有一方向的至少一部分移动。在图 17 中，手势的移动向量有三个分量，可通过部分移动方向来表示，例如 {90,0,270}，该方向是以该方向的角度识别。将手势描述为方向序列是有存储器空间效率的，同时还允许执行简单而快速的比较以识别手势。

[0155] 图 17 的范例是两维，但应指出的是，参照图 17-23，本文所教示的概念可以很容易地扩展到三维或更多维度，将讨论如下。

[0156] 此外，应当指出的是，为了使装置即使在摄像机和待被追踪的对象对于它们相应的上下方向不对准时仍能工作，本文所披露的方式披露了通过相对的方向和角度工作。

[0157] 例如，如果用户正在媒体站例如电视机上看电影，他可能不会坐直而是例如躺在沙发中。如果用户躺着但作了一个姿势如向上和向左（利用绝对值由 0 和 270 度方向组成），以便他的手相对于摄像机的垂直线成一角度移动。如果角度大于容许误差（容许误差的进一步细节见下文），比如说 30 度，则检测的角度或方向将分别是 30 和 300 度而不是 0 和 270 度，因此控制器不能够正确地确定事实上使用者所做的手势。这对于使用者来说是

烦人的，因为在他认为他正在做一正确的手势并且不愿意太多地改变他的位置。

[0158] 当如果装置未被正确固定、摆放在不平的表面上或其它这样的情况而出现摄像机稍微未对准时，同样的问题会产生。

[0159] 因此，下文中所有涉及检测的方向涉及相对方向。

[0160] 为了利用相对方向来扩大可用手势的数量，控制器可被设置成确定基线或基线方向。可以在开始时、启动新程序时或定时或间歇地确定基线或基线方向以适应使用者移动。

[0161] 可以为用户或摄像机或者两者确定基线方向。

[0162] 在一个实施例中，控制器被设置成为待被追踪的对象确定基线方向。在一个实施例中，这可以在检测待被追踪对象的每次进行。对象的当前定向随后被确定并与一般定向进行比较，基此可确定偏移值。通过将待追踪对象的结构与模型结构进行比较可以确定当前的定向。在追踪手的示例中，手指的方向可用于确定当前的定向。

[0163] 因此，控制器被设置成根据下式适应所有检测到的角度：

$$\alpha 1 = \alpha + \text{偏移量}$$

其中，α 为检测到的角度，而 α 1 为待比较的经调整的角度。在一个实施例中，控制器被设置成确定摄像机的基线方向。在一个实施例中，这可以在开始时进行。摄像机的当前定向例如通过识别应当是垂直的或水平的直线确定并相应地调整视角。随后可相应调整参照角。如果许多线几乎是垂直的（水平的），则控制器可假设它们应该是垂直的（水平的）并设定一偏移量使得线是垂直的（平行的）。应当意识到，在现代社会中，许多直线自然出现在通常的住处或其它现代环境中。例如，拐角几乎总是直的，在两壁相交或者一壁与地面和 / 或天花板交汇处提供直线。像框和窗框也几乎总是垂直的 / 水平的。应当清楚，存在许多其它的示例。因此，通过设置控制器以在待被追踪对象的影像的背景中检测直线，可检测摄像机的定向。

[0164] 因此，控制器被设置成根据下式调整所有检测到的角度：

$$\beta 1 = \beta + \text{偏移量}$$

其中 β 是待比较的参照角；而 β 1 是待比较的经调整的参照角。

[0165] 可替换地，如上文所述调整检测到的角度。

[0166] 为了允许使用者移动不准确，控制器被设置成确定，如果比较的角度是在被比较角度的范围或容许误差内，则有匹配。被比较的角度在下文中将称为参考角或参照角。换句话说，控制器被设置成确定如果$\beta-\varphi<\alpha<\beta+\varphi$，则存在部分移动的匹配。其中 β 是参照角，α 是比较角，而φ是容许误差。

[0167] 为了过滤出非主动移动，控制器可能设置成，除了移动的角度之外，还确定移动的距离，并且只有当距离 (dist) 高于阈值 (V1) 时，才将移动角度与参考角比较。换言之，如果 dist> = V1，则部分移动会被检测到并进行比较。

[0168] 通过结合距离的测量和角度的容许误差，控制器被设置成通过区分手势（或手势的一部分）与被追踪对象的随机移动而检测到手势（或手势的一部分）。距离和角度容许误差的结合限定一锥形（其顶部被截掉），在此锥形内，任何产生的移动将被检测为沿该角度方向的移动。

[0169] 在一个实施例中，载有摄像机的装置的定向可利用陀螺仪或其它水平测量装置确定。如果摄像机和装置的定向是已知的，则定向可被添加至检测到的角度或从检测到的角

度减去。不过，在所述实施例中，本文所揭示的方式仍可被用为未知的用户和 / 或用户运动的定向。知道摄像机并非水平对准，可增加容许误差以适应任何变化。

[0170] 对于图 17 的范例移动，三个部分移动构成追踪对象 H 的手势。第一移动代表从位置 H 到 H' 的移动，角度为 90 度且距离为 dist1，第二移动代表从位置 H' 到 H" 的移动，角度为 0 度且距离为 dist2，第三移动代表从位置 H" 到 H"' 的移动，角度为 270 度且距离为 dist3。仅当所有的距离 dist1、dist2、dist3 大于阈值 V1 且各个角度在容许误差内时，将识别出手势。可替换地，如果距离之和大于阈值也将识别出手势，也就是说，如果 dist1+dist2+dist3> = V1，将识别或辨认出手势。

[0171] 本文所揭示的对象追踪提供了用于对象追踪的实时坐标，特别是参照图 4 至图 16。这允许适合用于实时系统的快速手势识别，因为坐标是实时提供且手势识别是在没有广泛或高资源要求的计算下执行的。

[0172] 在一个实施例中，控制器被设置成通过使用诸如杜林机的状态图来识别手势。图 18 示出了这样的一个杜林机示例，用于识别列于表 1 的范例手势。杜林机 1800（如图 19、图 20 和 22 的杜林机 1900、2000、2200）是由计算装置实现和执行的，如参考图 1 和 2 之揭示。

[0173] 控制器（未绘示在图 18 中，但在图 2 中为 210）被确定以图 18 中粗线和 0 所示的初始或启动状态开始。如上所讨论的，当检测到移动时，可能是通过确定移动的距离是高于阈值 V1，移动的角度启动状态的转变。在图 18 中的范例中，只有 4 个角度是用于移动，四个角度为 0 度（向右移动）、90 度（向上移动）、180 度（向左移动）和 270 度（向下移动）。应该清楚，根据本申请的教示，可应用和使用进一步的指示来定义和识别手势。

[0174] 如果检测到的移动具有不同于识别一转变的小于容许误差的角度，则执行该转变。例如，如果向左移动被侦测到（在 170 度的角度），假设容许误差为 10 度，控制器将启动转换，从初始状态 0 到图 18 中以 {180} 指示的另一种状态。状态 {180} 对应一动作，其通过实线绘示的状态表示。从表 1 中可以识别该动作是一还原动作。

[0175] 在这个例子中，手势及其对应的动作可以储存在数据库中或记录在计算装置的存储器中（参考图 2 中 240）。

[0176] 通过图 18 的杜林机 1800 中的状态转换，可以检测和识别一系列的移动。例如图 17 中所示的一系列的移动会导致状态指示为 {90 ;0 ;270}，其中根据表 1 是对应于放大的动作。

[0177] 如果控制器追踪一角度的移动，例如 120 度，该角度超出任何角度的容许误差锥（假设是 10 度的容许误差），控制器将返回到初始状态 0。在另一实施例中（将讨论于上），控制器将返回到先前的稳定状态，稳定状态是对应一动作的状态或初始状态。在图 17 的例子中，先前的稳定状态是所有其它（不稳定）状态的初始状态。

[0178] 如果控制器无法追踪距离大于阈值的移动，可能是在超时数值 T1 内，及没有达到对应动作的（稳定）状态，控制器将返回到初始状态 0。在图 18 中，对应动作的状态是以连续线标示，非对应动作的状态是以虚线标示。超时阈值 T1 的范例数值是 0.5 秒。另一范例是 0.3 至 1 秒。

[0179] 在一个实施例中，参照图 4 至 16 揭示之对象追踪可被配置以使得追踪对象的位置只有在对象已经移动大于阈值距离的距离时才提供。这允许诸如中央处理单元 (CPU) 之计

算装置的控制器处理应用数据，而诸如图形处理单元 (GPU) 之另一处理器处理追踪数据。

[0180] 配置控制器以通过使用杜林机 1800 来检测及识别手势能够快速、可靠及实时地进行手势识别，而不需要大量的计算资源。进一步而言，本文揭示之方式是有弹性的，因为手势不需有特定的形式且可通过将状态加入杜林机 1800 中而轻易适应。

[0181] 此方式易于延伸为亦包括其它维度，例如深度、旋转、转动角度、移动距离和移动速度。

[0182] 图 19 绘示以四个维度识别手势的杜林机 1900 的示例，四个维度为两个空间维度、移动速度和移动长度。图 19 的杜林机 1900 是图 18 的杜林机 1800 的部分延伸。应当指出的是，即使图 19 的杜林机 1900 是四个维度，本文之教示适用于任何维度的组合。

[0183] 在图 19 的范例实施例中，杜林机 1900 和实施杜林机的控制器能够区分快速和缓慢的移动，即确定移动的速度。如果移动的速度被确定为速度阈值 S1 以上，则移动被分类为快速移动，如果不是，移动被分类为缓慢的移动。移动的速度可通过将移动距离与达到该距离所用的时间进行比较而确定。这可以在对象追踪期间或移动与对应手势之识别期间确定。在图 19 中，快速移动是以划线表示，如转换 (270) 转换到对应动作打印之状态 {270}。速度阈值的范例是根据距离阈值 V1，使得 S1 = V1/1 秒。速度阈值的另一例子是根据距离阈值 V1 和超时阈值 T1，使得 S1 = V1/T1。

[0184] 控制器也可以被设置成和 / 或可附加地设置成区分长与短的移动。这可以通过比较移动距离 (例如在图 17 的 dist1) 与第二阈值 V2 完成，第二阈值大于第一阈值 V1。如果距离大于第二阈值，移动被分类为长移动，如果不是，移动被分类为短移动。在图 18 中，长移动是以粗体字表示，如转换 (270)，转换到对应动作删除的状态 {270}。

[0185] 因此，通过考虑如移动速度及 / 或移动长度等因素，指派不同的动作至相同的方向是可能的。图 19 和表 1 的例子中，杜林机 1900 提供 {270}、{270}、{270}、{270} 的状态和对应动作。{270} 是对应传送动作的向下短、慢移动，{270} 是对应删除动作的向下长、慢移动，{270} 是对应打印动作的向下短、快移动，及 {270} 是对应删除而没有确认动作的向下长、快移动。这能够将具有严重后果的动作指派给较为鲜明的、本质是决定性或确定性的手势。例如，错误执行长移动较错误执行短移动的可能性低。

[0186] 要根据不同的使用者、不同的计算装置、不同的摄影机及 / 或不同的应用程序和情况来于灵活性，控制器被设置成根据将要追踪的对象大小、与将要追踪对象的距离、显示器大小和摄影机的视野来决定距离阈值 V1 及 / 或 V2。

[0187] 与追踪对象的距离可通过摄影机的聚焦特征或深度方向的追踪来确定，如参照图 13 至图 15 所揭示。使用深度方向的追踪，可根据对象在深度方向 (大致平行于摄影机的视线) 的移动而增加或减少阈值。

[0188] 例如，如果阈值 V1 和 V2 是将被追踪对象的相对大小，V1 和 V2 的一些范例值是 V1 = 3× 对象大小 ;V2 = 6× 对象大小。

[0189] 例如，如果阈值 V1 和 V2 是摄影机视野的相对大小，V1 和 V2 的一些范例值是 V1 = 视野的 20% ;V2 = 视野的 50%。如果视野是 120 度，V1 是 24 度，V2 是 60 度。

[0190] 如上所讨论，本文揭示之手势检测适用于实时执行。要能够更快的整体执行和操作，在侦测到手势时，计算装置的控制器 (在图 2 中分别为 210 和 200) 被设置成在检测手势时预先执行对应于假定手势的动作。返回到图 18，被设置成预先执行对应动作的控制器

之运作将予以揭示。

[0191] 在已经检测到向上移动的情况下（参考图 18 中的转换 (90)），只有两种运作是可能的，即动作放大（对应于 {90 ;0 ;270}）及缩小（对应于 {90 ;180 ;270}）。在此范例中，动作与显示的影像相关。要准备时，控制器可以检索执行变焦操作所需的影像数据，并且还可能确定显示影像的放大和缩小表示。通过准备要采取的行动，动作可以在更短的时间内被执行，藉此导致更快的响应时间。例如，如果需要从外部驱动器加载影像数据，通过在完成及识别手势之前即开始加载影像数据，可省下许多 0.1 秒。已知大于 0.1 秒的响应时间对许多计算装置的使用者而言是感觉到压力和累人的。因此，预先执行动作的计算装置配置是大有裨益的，因为它减少了响应时间和使用压力较小。

[0192] 由于诸如中央处理单元 (CPU) 之现代控制器的计算效率，在执行和侦测手势时，控制器将闲置许多个时钟周期，而且因为相比于处理器，人体移动相对较慢，因此本申请之作者已经理解到，通过利用这些闲置的时钟周期来预先执行动作（至少部分地），可实现更短的动作响应时间。

[0193] 在这个例子中，控制器能够确定对应的手势，即使手势尚未完成。因为自每个（不稳定）的状态只有一转换移出，控制器知道哪个手势已经输入于状态 {90 ;180} 和 {90 ;0}。因此，控制器可以预先执行对应的动作，当达到对应动作的状态时，可以呈现或产生预先执行动作的结果。在这个例子中，显示放大（或缩小）的影像。通过暂缓产生预先执行动作的结果，可让使用者有时间透过中断手势来反悔动作。

[0194] 如果一个控制器执行对象追踪，一处理器执行动作相关的操作，预先执行动作的方式是特别有益的。

[0195] 要赋能更多不同的手势，本申请的作者已经理解到，一手势可以形成另一手势的子序列。

[0196] 图 20 绘示用于控制器的杜林机 2000，赋能以区分第一手势和第二手势，其中第一手势由第二手势的子序列组成。杜林机 2000 是图 18 或 19 的杜林机 1800 或 1900 的一部分的延伸。

[0197] 标示为 {0 ;90} 的状态对应往前翻页的动作（见表 1），标示为 {0 ;270} 的状态对应往后翻页的动作（见表 1），如图 18 的杜林机 1800。杜林机 2000 还包括两个状态，一状态标示为 {0 ;90 ;180}，对应向前滚动的动作，一状态标示为 {0 ;270 ;0} 对应向后滚动的动作。

[0198] 可以看到，所有四个状态都是稳定的状态，标示为 {0 ;90} 的状态是在往状态 {0 ;90 ;180} 的路径上，即 {0 ;90} 的手势是 {0 ;90 ;180} 手势的次手势，标示为 {0 ;270} 的状态是在往状态 {0 ;270 ;0} 的路径上，即 {0 ;270} 的手势是 {0 ;270 ;0} 手势的次手势。

[0199] 为了区分次手势和全手势，如果进一步的移动被检测为具有不对应于任何其它手势的角度，控制器是配置成返回到先前的稳定状态。这允许通过移往未定义方向来完成手势。这也能够定格手势。

[0200] 可替换地和 / 或另外地，如果在时间阈值 T1 内没有检测到进一步的移动，控制器返回到先前的稳定状态。先前的稳定状态可能是目前拜访的状态，如果这样的状态是一稳定的状态。

[0201] 图 20 的范例中，如果使用者执行手势右 - 上 -（下 / 远离 / 右 / 停止），则执行往

前翻页的动作，如果使用者执行手势右－上－左－(下／远离／右／停止)，则执行往前滚动的动作。类似地，如果使用者执行手势右－下－(下／远离／左／停止)，则执行往后翻页的动作，如果使用者执行手势右－下－右－(下／远离／左／停止)，则执行往后滚动的动作。滚动操作可以定义为定格手势，使得当控制器是在滚动状态时，继续保持滚动(即没有检测到进一步的移动时)。

[0202] 为了允许多个对象手势，控制器可被设置成追踪多个对象，如参照图11至12和13至15所揭示。可以通过状态图或杜林机识别每个对象的移动(图11和图12中的01和02)如参照附图18至20所揭示，其中图18、19和20的杜林机1800、1900、2000之每者(或其组合)的杜林机实例系产生并执行用于每个对象。这允许多个对象进行同时或并行的手势。识别用于每个对象的手势可以组合为对应的动作，或者每个手势可有每者为一动作的结果。这允许使用多个并行手势进行同步操作或处理。这种多对象手势的范例使用是一手控制第一应用程序或在游戏应用程序中的第一游戏者，另一手控制第二应用程序或游戏应用程序中的第二游戏者，藉此赋能多执行绪之执行，亦赋能多人游戏。

[0203] 图22绘示了用于识别多个对象执行之复杂手势的杜林机2200。这样的手势可以用于控制一个并且相同应用程序，藉此提供或赋能更丰富的手势库给应用程序的开发人员。图21绘示了两个对象的示意图，在这个例子中，两只手H1和H2执行复杂的、多个对象的手势。第一手(左手)H1向右移动(以(0/-)来表示第一对象移动的方向为0度)到第二位置H1'，而第二手(右手)H2向下移动(以(-/270)来表示第二对象移动的方向为270度)到第二位置H2'。

[0204] 对于复杂的手势，杜林机2200具有三种状态，这取决于哪只手先移动，第一(左)手H1或第二(右)手H2。手势与对应的动作绘示于表2。范例手势和对应的多个动作只是为了说明的目的，应该清楚的是，范例本身不应被解释为限制本文之教示。杜林机2200中的每个转变对应于至少一对象的移动。

[0205] 如果第一(左)手H1先移动，控制器将结束于状态{0/-;-/270}，对应于带出下一页面及删除该页面(或前者)的动作。如果第二(右)手H2先移动，控制器将结束于状态{0/270;0/-}，对应于删除现有页面及带出下一页面的动作。根据所删除页面(之前的或下一页面)，这两个动作可能提供不同的最终结果。另外，此顺序提供使用者在删除前页之前预览下一页面的可能性(状态{0/-;-/270})。

[0206] 如果第一手H1和第二手H2基本上同时移动，第三多个动作会被执行。在这个例子中，同时动作对应缩放操作。

[0207] 在一个实施例中，对象被检测以移进的顺序是没有结果的。这减少了一些可用的手势，但简化了正确手势的输入，因为使用者不需要执行移动的确切时间。

[0208] 具有多个对象手势使得手势库更为丰富，也使得手势库对于应用程序设计者及应用程序的需求而言更加灵活。

[0209] 为了进一步简化使用者的手势输入，控制器可以被设置成在手势识别阶段提供使用者回馈，其通过显示所检测移动的指针或状态(例如通过小图标、箭头、文字或角落里的进度指示，例如显示器的右上角)。每次已成功检测到一移动方向时即更新指示。这使得使用者能够理解手势进展。这特别有利于复杂的手势(含有许多移动方向或移动组合)。

[0210] 图23绘示了根据本文教示之识别手势的概略方法。在2310，计算装置接收数据，

识别被追踪对象之移动，例如通过对象移动方向的角度。在 2330，该角度与参照手势的参考角或参照角比较，以确定是否识别出手势。如果是这样，在 2340，计算装置执行对应的动作。

[0211] 通过遍历状态图，将角度与参照角进行比较，其中该角度是角度序列的一部分，在 2320，序列中的每个角度与状态图中两个状态之间的转换进行匹配。如果到达对应一动作的状态，在 2340 执行动作。

[0212] 识别本文所揭示手势的方式是有利的，因为它是基于方向而不是绝对位置，对于对象和摄影机之间的距离较不敏感。也不需要缩放所追踪对象以适于参照或参考对象。这是特别有用的，因为相同的手势设定可用于不同的装置（例如但不限于电视机、行动电话和固定的计算机）。

[0213] 此外，手势库只需要最少的数据，因而易于储存和快速地操作，因此只需要最少的计算资源。

[0214] 本文所揭示的方式也能够处理缩放变化，因为当一方向的手势与视线平行时，可解释缩放变化或者忽略这种移动。这提供识别手势的强大方式。

[0215] 本文之教示提供的优点是比较角度及可能距离的简单操作不需要任何巨大的计算资源以有效地实施。例如，教导的方式并不需要作出任何缩放、旋转或变形操作，曲率的确定确实是简单和优雅的。因此，本文所教示的方式适合于使用在低或不利的光照条件下，这也使得它非常适合于可携式装置和娱乐系统。

[0216] 另一好处在于，不需要训练追踪系统或手势识别。

[0217] 另外一优点在于，本文所教示的方式很容易整合于现有的装置中，因为需要的输入是影像串流，其可为来自摄影机的视频流（内部或外部），或通过另一视频来源提供，例如影片或串流式视频流。所提供的输出是相对移动的坐标，其与输入结合时，建立与本文所教示方式的简单接口，很容易整合到现有的装置以及到要生产装置的设计中。

[0218] 本发明主要已经参照一些实施例描述于上。然而，如熟习本技术领域者所容易理解，除上述所揭示的其它实施例在本发明范围内的可能性相等，如所附专利权利要求所界定。

手势	动作
{0, 270}	动作 1 :往前翻页
{0, 90}	动作 2 :往后翻页
{90, 0, 270}	动作 3 :放大
{90, 180, 270}	动作 4 :缩小
{180}	动作 5 :复原键入
{270}	动作 6 :打印
{270}	动作 7 :传送

{270}	动作 8 :删除
{270}	动作 9 :删除而没作确认
{0 ;90 ;180}	动作 10 :往后卷动
{0 ;270 ;0}	动作 11 :往前卷动

表 1 :手势和对应动作

手势	动作
{0/- ;-/270}	多个动作 1 :下一页面 + 删除
{-/270 ;0/-}	多个动作 2 :删除 + 下一页面
{0/270}	多个动作 3 :放大

表 2 :多对象手势和对应动作

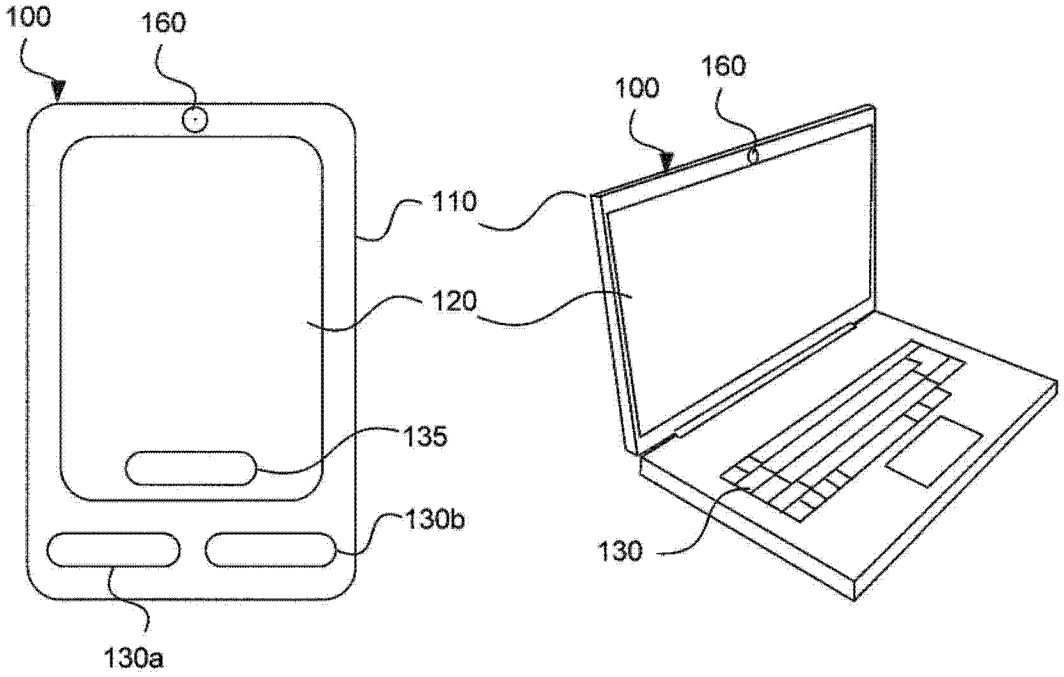

100
160
110
120
135
130b
130a
100
160
130

图 1A 图 1B

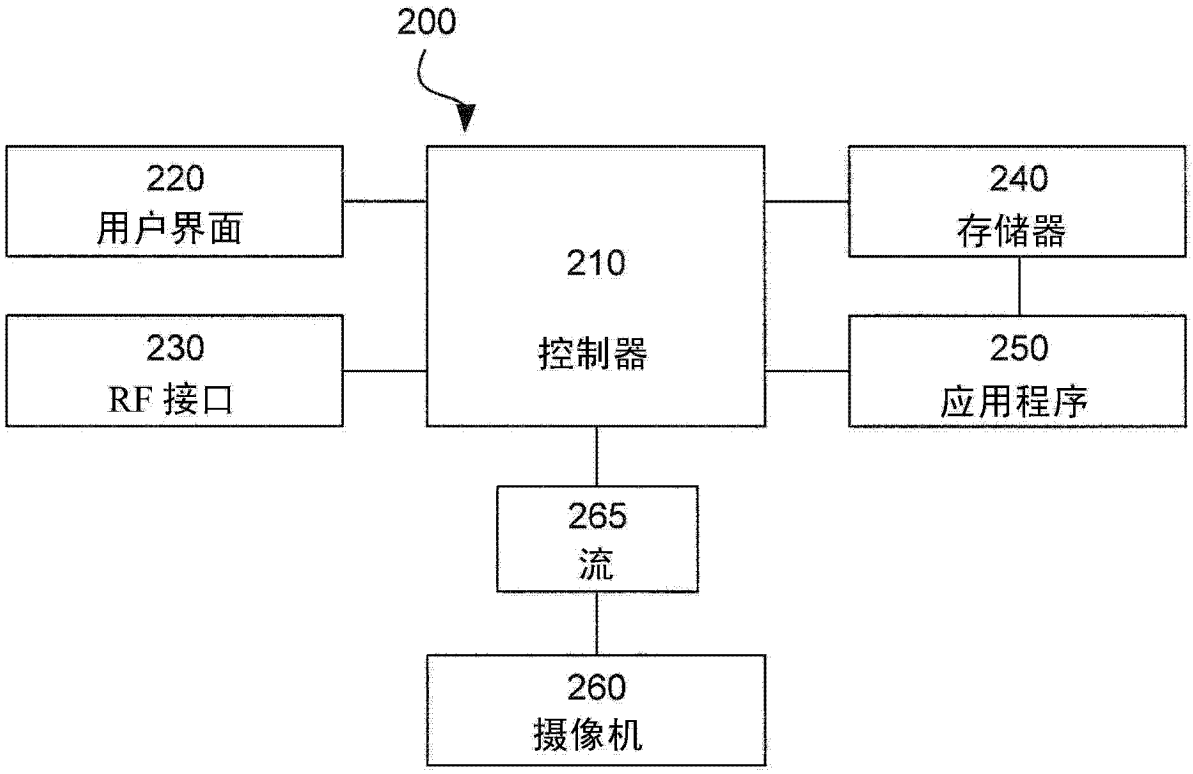

200
220
用户界面
210
控制器
240
存储器
230
RF 接口
250
应用程序
265
流
260
摄像机

图 2

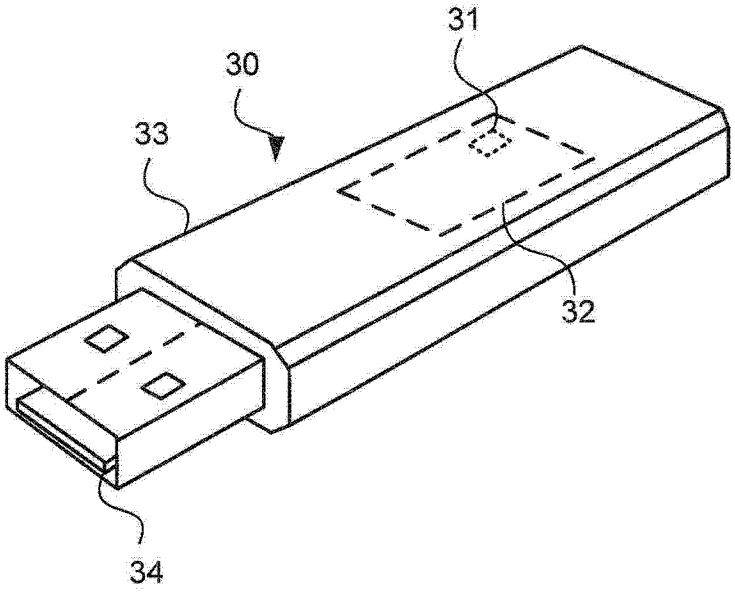


图 3

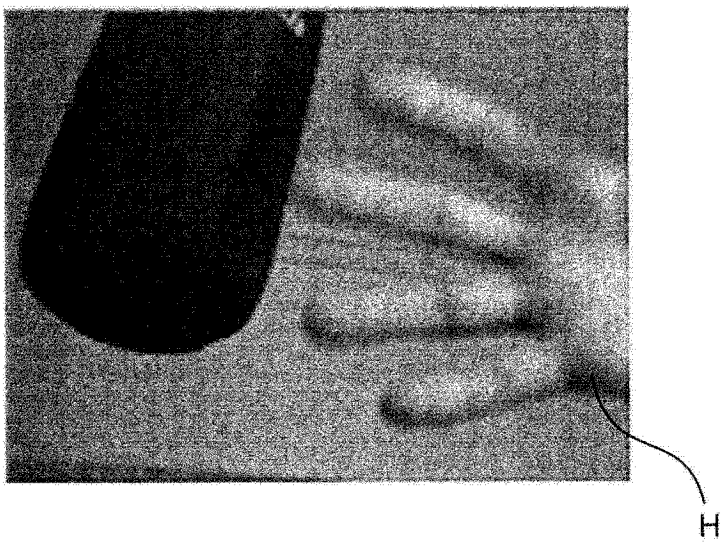


图 4A

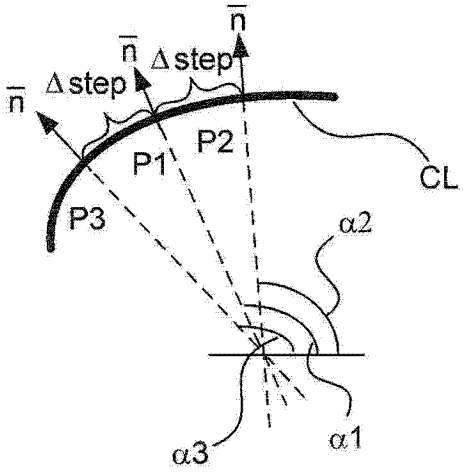


图 5

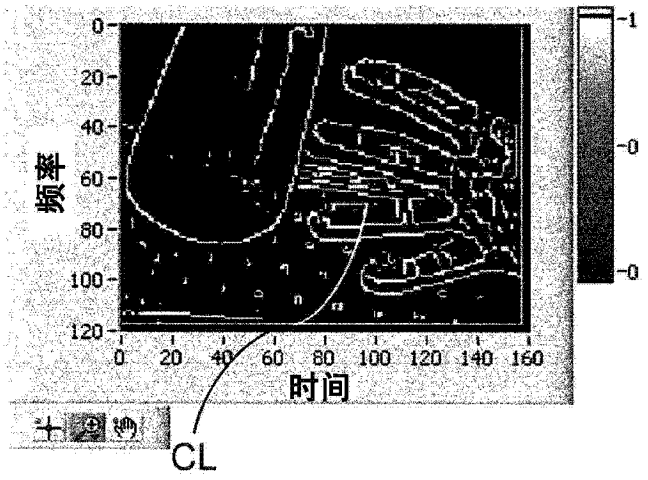

0
20
40
60
80
100
120
频率
0 20 40 60 80 100 120 140 160
时间
-1
-0
-0
CL

图 4B

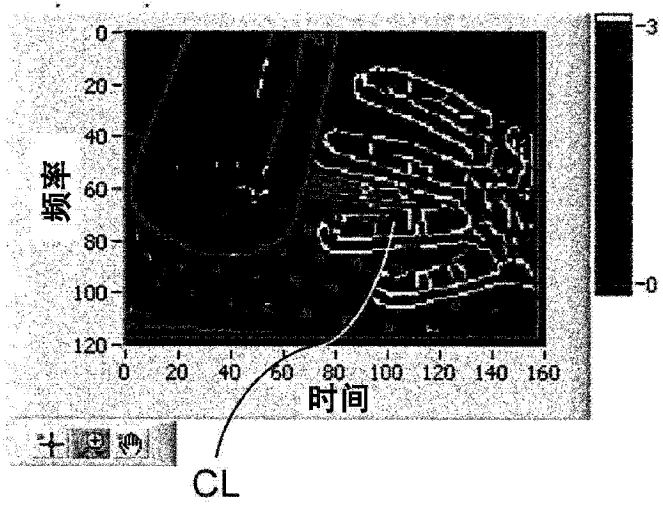

0
20
40
60
80
100
120
频率
0 20 40 60 80 100 120 140 160
时间
-3
-0
CL

图 4C

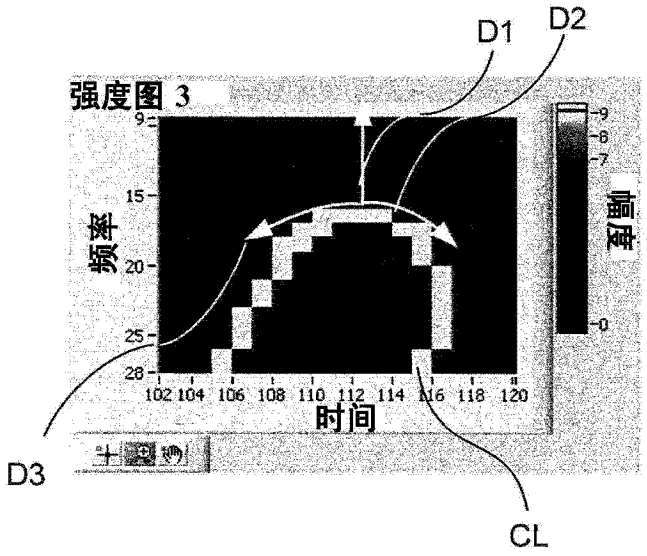

D1
D2
强度图 3
频率
强度
时间
D3
CL

图 4D

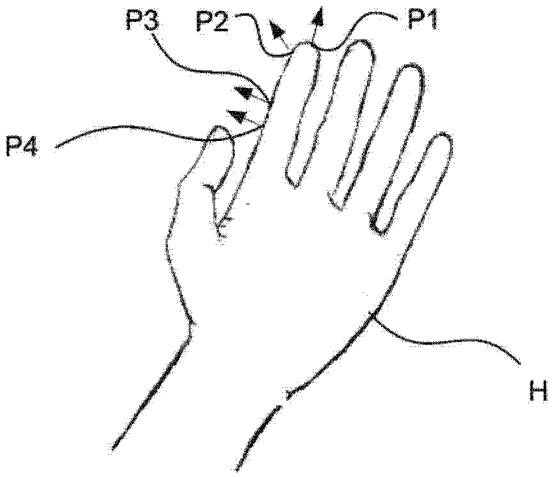

P3
P2
P1
P4
H

图 6

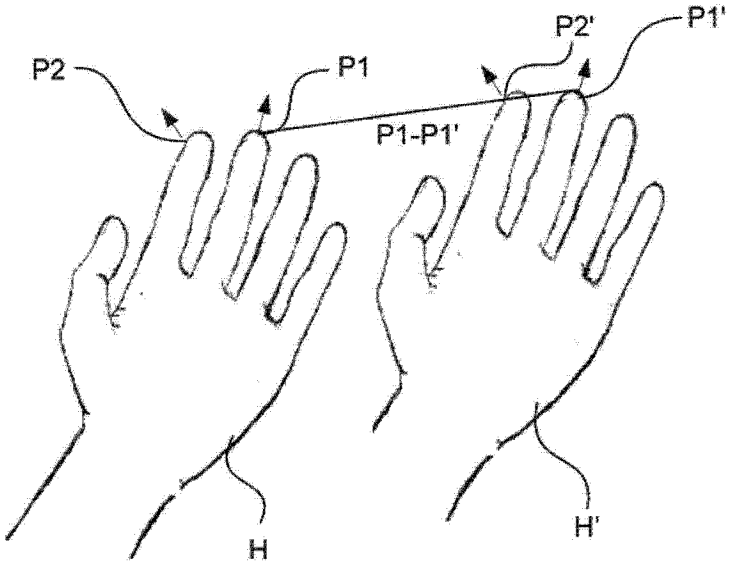

P2
P1
P2'
P1'
P1-P1'
H
H'

图 7

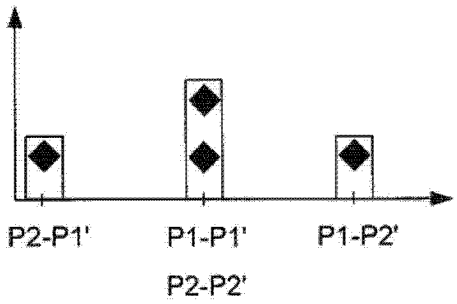

P2-P1'
P1-P1'
P1-P2'
P2-P2'

图 8

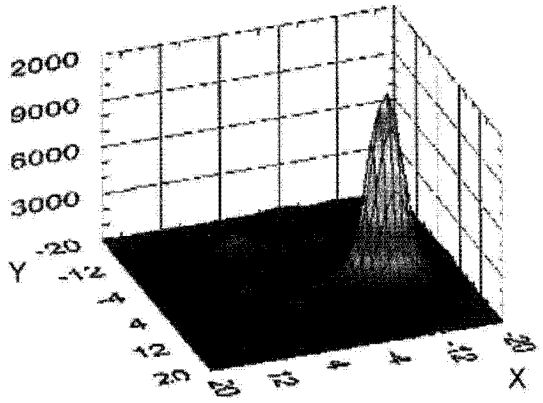

2000
9000
6000
3000
-20
Y
-12
-4
4
12
20
X

图 9

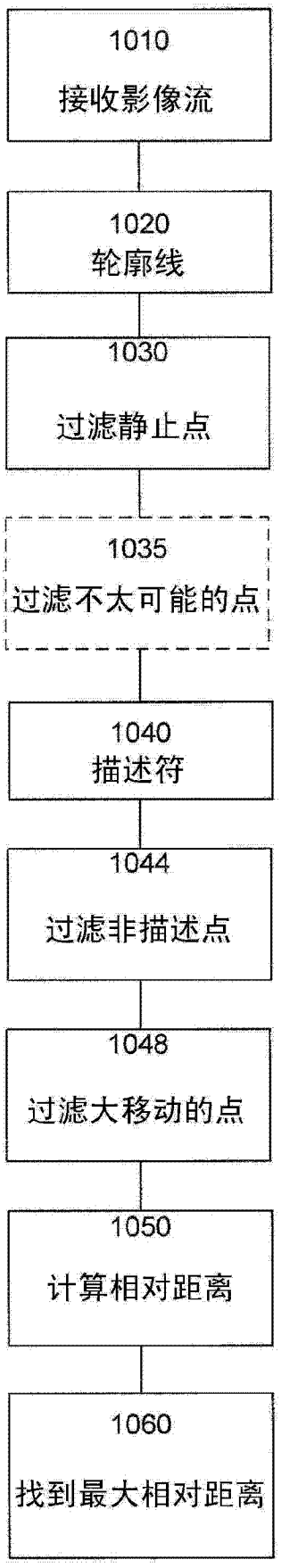

1010
接收影像流
1020
轮廓线
1030
过滤静止点
1035
过滤不太可能的点
1040
描述符
1044
过滤非描述点
1048
过滤大移动的点
1050
计算相对距离
1060
找到最大相对距离

图 10

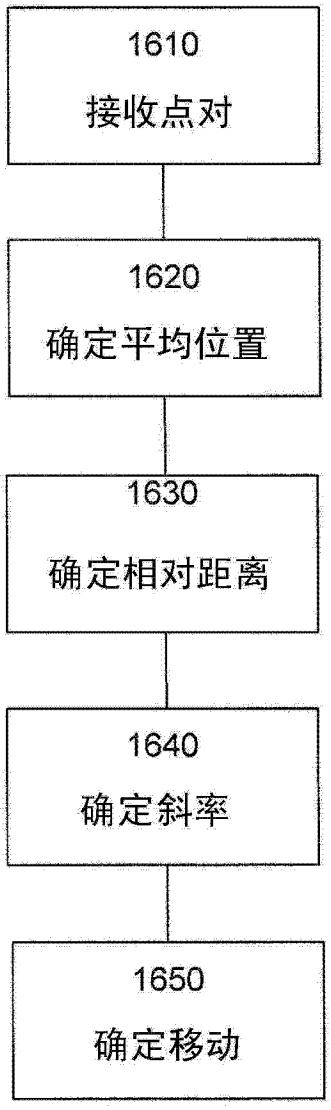


图 16

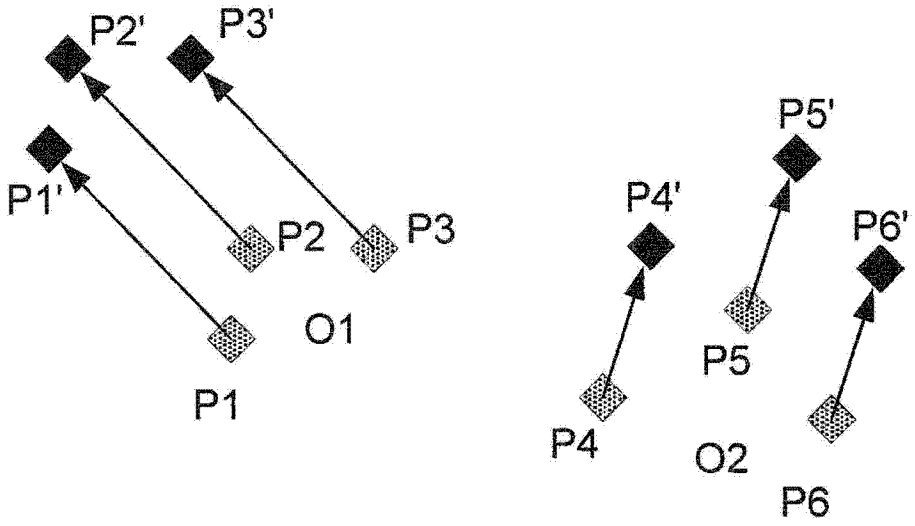

P2'
P3'
P1'
P2
P3
O1
P1
P5'
P4'
P6'
P5
P4
O2
P6

图 11

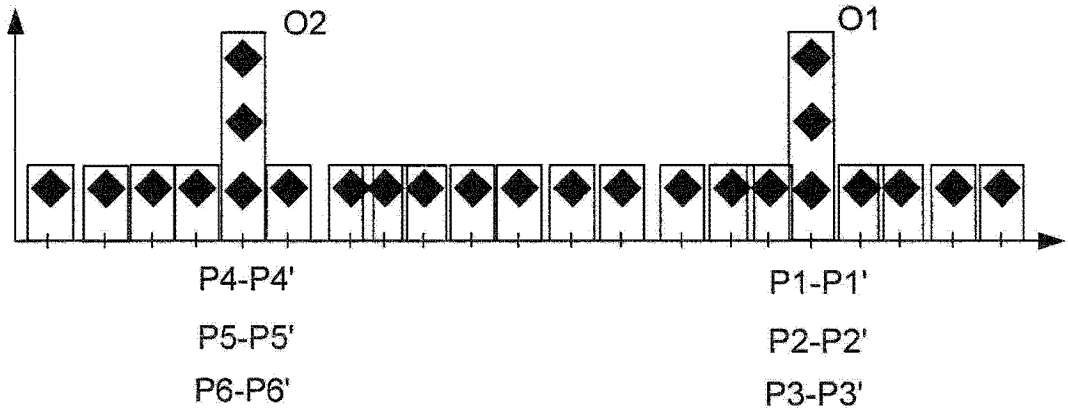

O2
O1
P4-P4'
P5-P5'
P6-P6'
P1-P1'
P2-P2'
P3-P3'

图 12

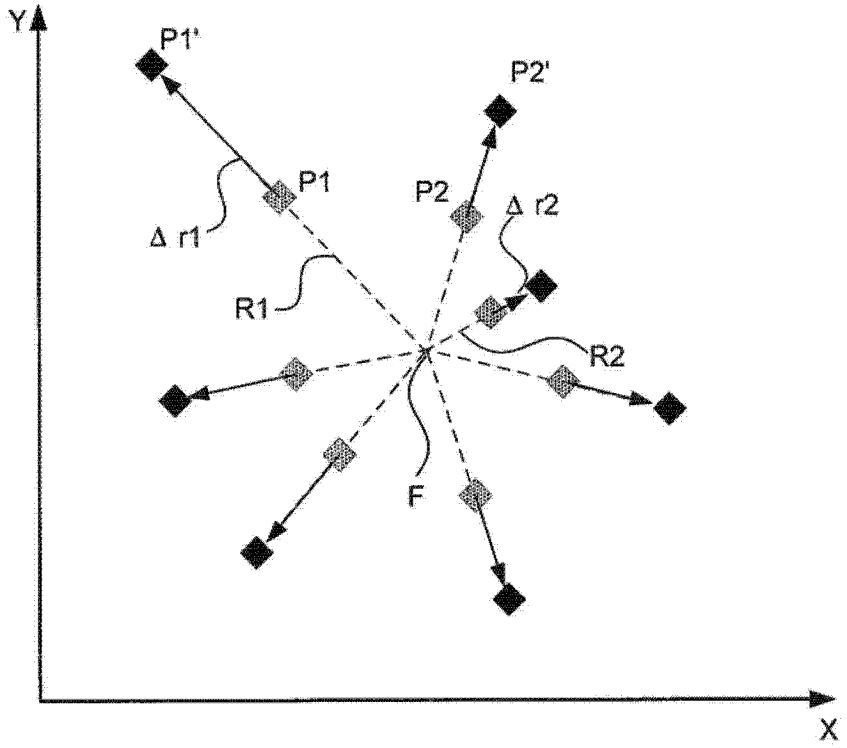


图 13

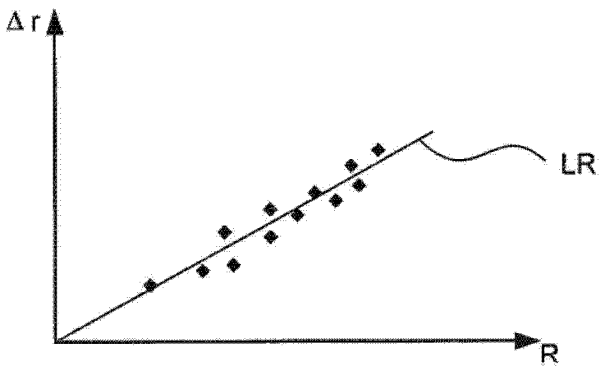


图 14

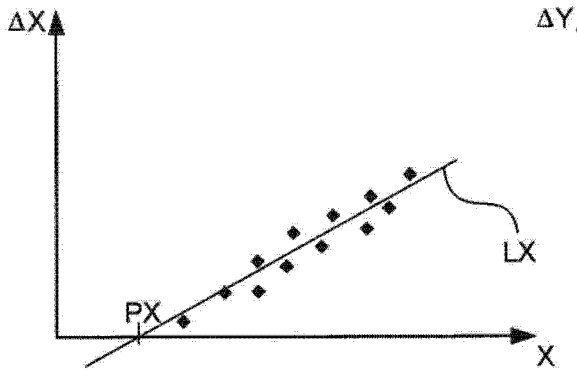


图 15A

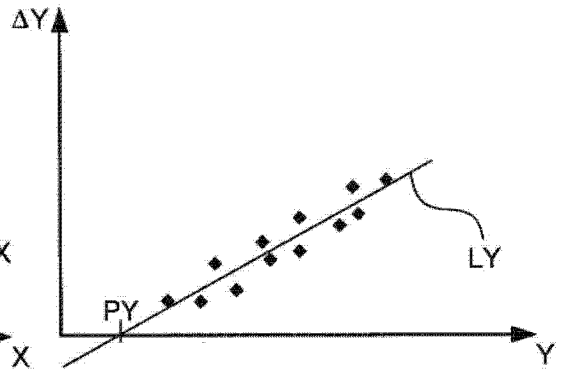


图 15B

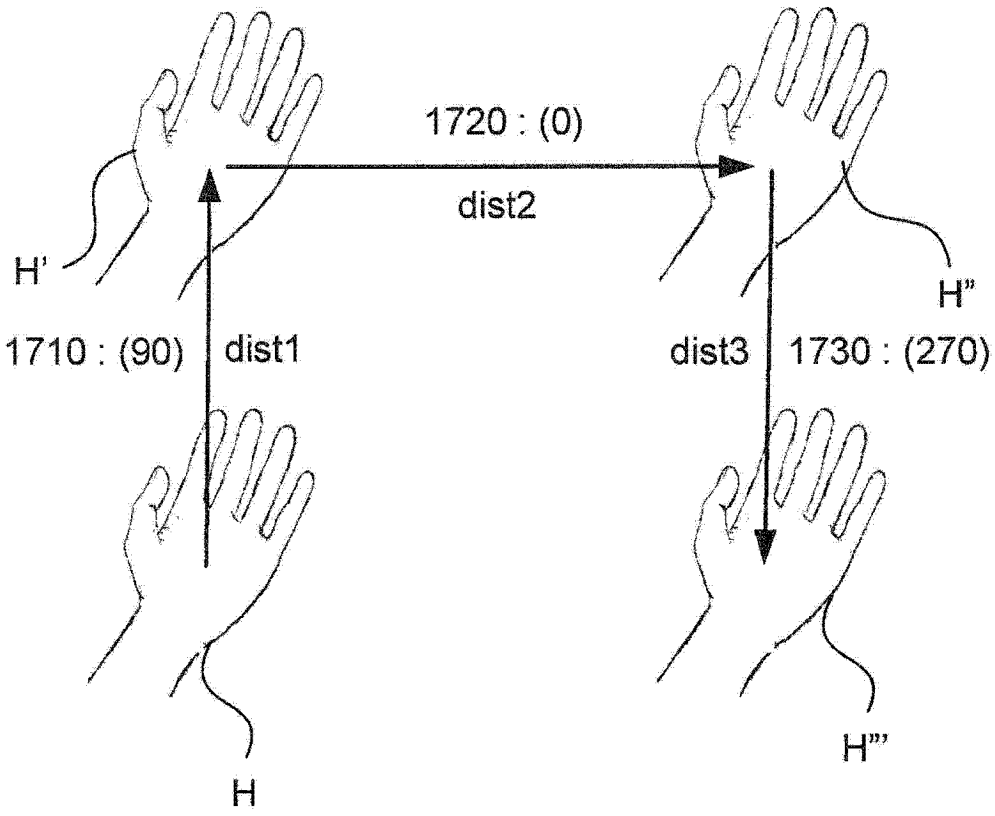


图 17

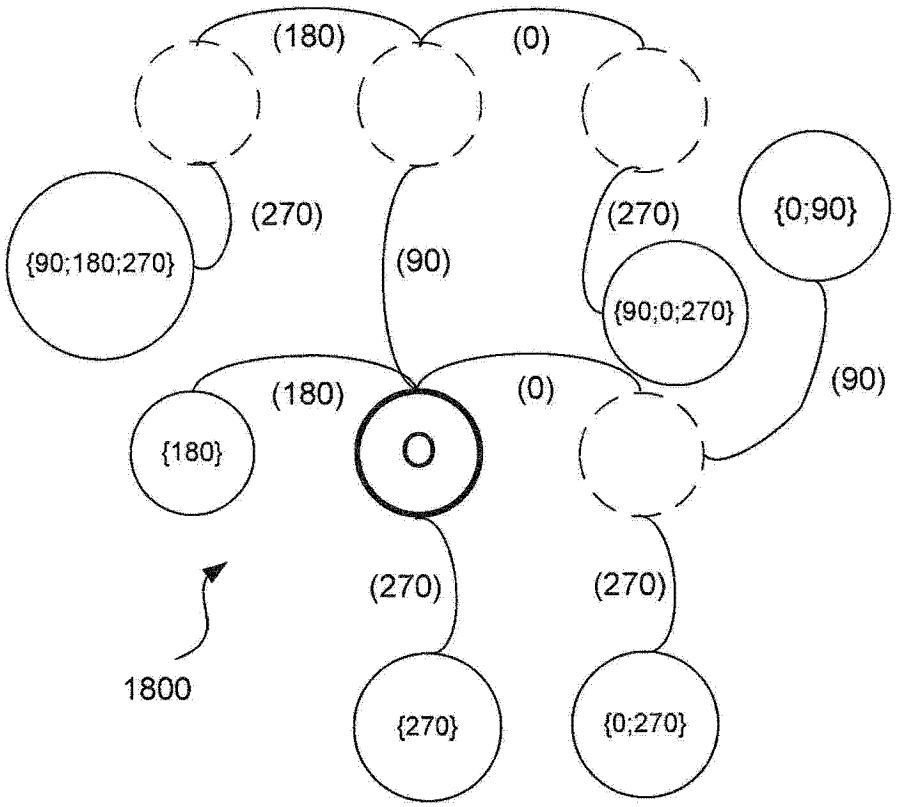

(180)
(0)
(270)
(270)
{0;90}
{90;180;270}
(90)
{90;0;270}
(90)
(180)
(0)
{180}
O
(270)
(270)
1800
{270}
{0;270}

图 18

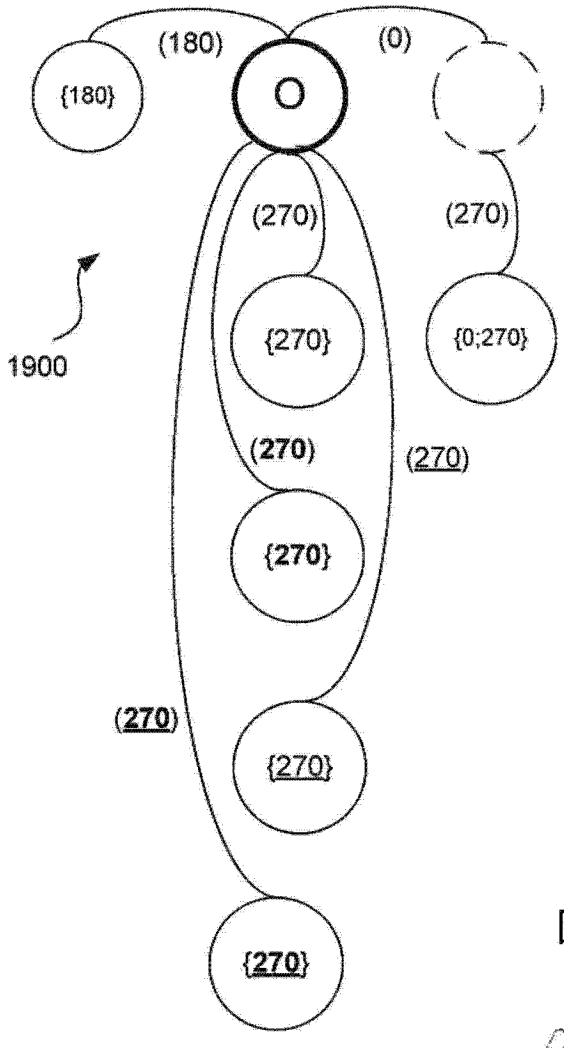

(180)
(0)
{180}
(270)
(270)
1900
{270}
{0;270}
(270)
(270)
{270}
(270)
{270}
{270}

图 19

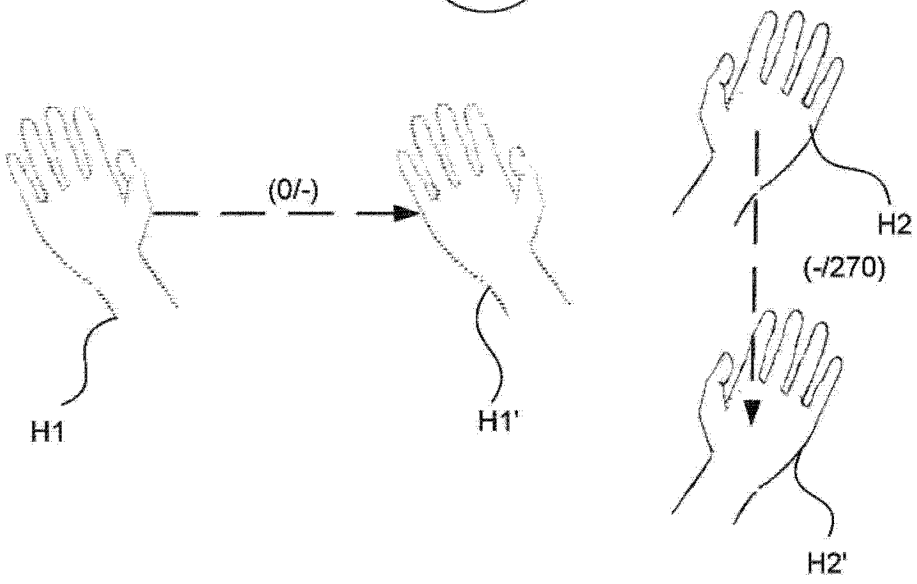

(0/-)
H1
H1'
H2
(-/270)
H2'

图 21

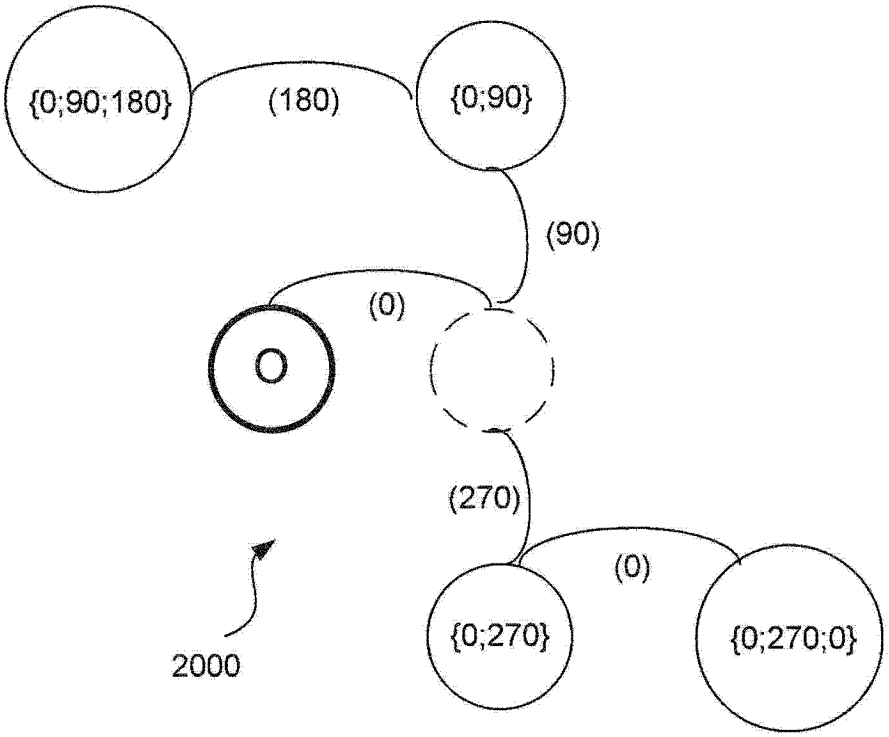

{0;90;180}
(180)
{0;90}
(90)
(0)
(270)
(0)
{0;270}
{0;270;0}
2000

图 20

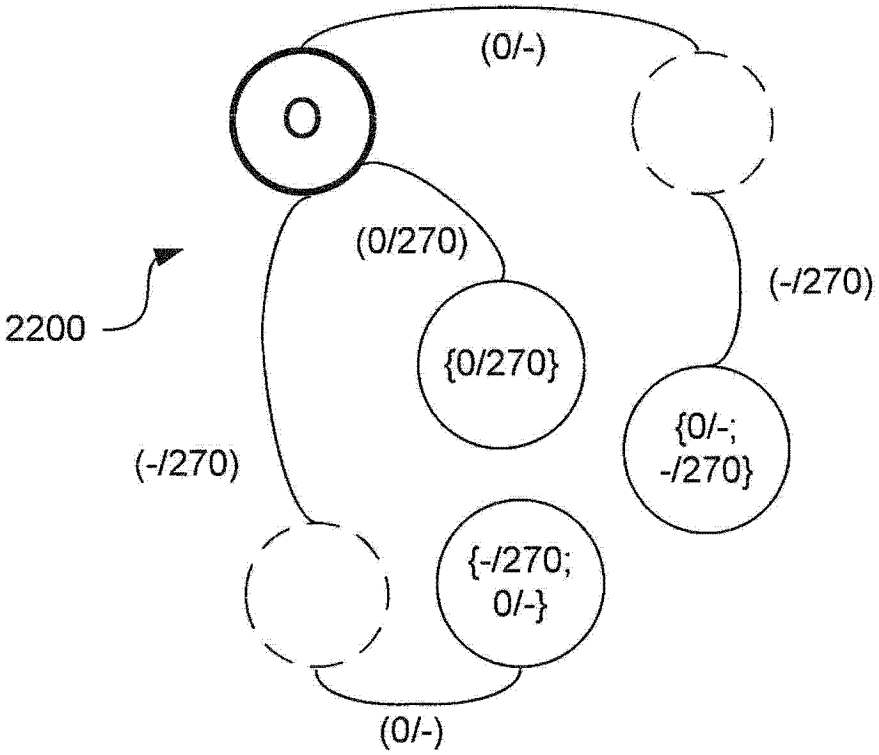

(0/-)
(0/270)
(-/270)
2200
{0/270}
{0/-;
-/270}
(-/270)
{-/270;
0/-}
(0/-)

图 22

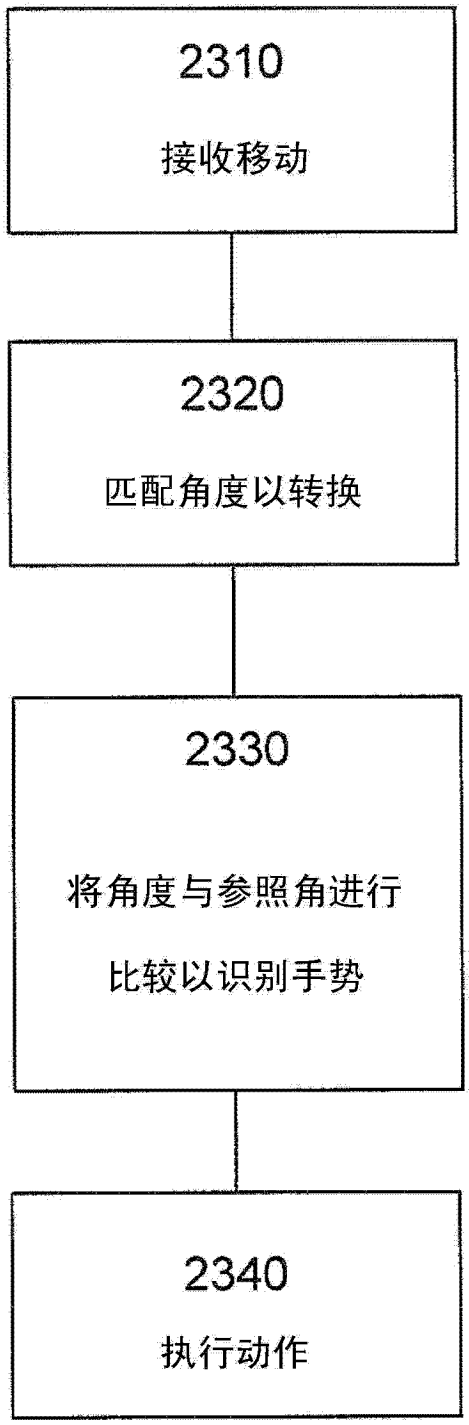


图 23