



(12) 发明专利

(10) 授权公告号 CN 101911048 B

(45) 授权公告日 2014. 07. 16

(21) 申请号 200980102343. 6

(51) Int. Cl.

(22) 申请日 2009. 01. 15

G06F 15/76(2006. 01)

(30) 优先权数据

(56) 对比文件

61/022, 143 2008. 01. 18 US

US 2007/0146325 A1, 2007. 06. 28, 全文.

12/106, 921 2008. 04. 21 US

US 2008/0009348 A1, 2008. 01. 10, 摘

(85) PCT国际申请进入国家阶段日

要、说明书第 6-9, 36-37, 41-43, 47-48, 54-

2010. 07. 16

55, 61-62, 76-81, 89-95, 102-125 段、附图

(86) PCT国际申请的申请数据

2-3, 6-7, 12-13, 15, 20A-20B, 22.

PCT/US2009/031075 2009. 01. 15

审查员 崔志鹏

(87) PCT国际申请的公布数据

W02009/091873 EN 2009. 07. 23

(73) 专利权人 因文森斯公司

地址 美国加利福尼亚州

(72) 发明人 史蒂文·S·纳西里 约瑟夫·姜

大卫·萨克斯

(74) 专利代理机构 北京安信方达知识产权代理

有限公司 11262

代理人 汤慧华 郑霞

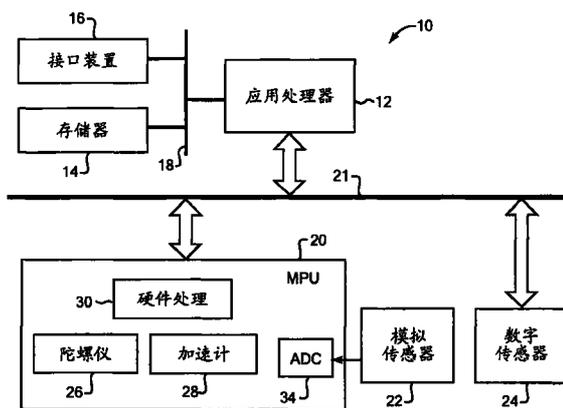
权利要求书2页 说明书17页 附图9页

(54) 发明名称

装置的运动传感器和接口应用程序

(57) 摘要

装置的运动传感器和接口应用程序。在一方面, 从在运动传感装置上运行的应用程序接收高级命令, 其中所述应用程序实现可用于在所述装置上使用的多种不同类型的应用之一。所述高级命令请求来源于所述装置的运动传感器的输出的高级信息, 所述运动传感器包括旋转运动传感器和线性运动传感器。转变命令以引起由所述运动传感器输出的运动传感器数据的低级处理, 所述低级处理遵循应用的类型的要求并响应于所述命令确定所述高级信息。所述应用程序不知道所述低级处理, 以及将所述高级信息提供给所述应用程序。



1. 一种将来自在运动传感装置上的运动传感器的运动传感器数据提供给在所述运动传感装置上运行的应用程序的方法,所述方法包括:

以第一采样率采样由所述运动传感器输出的运动传感器数据,其中所述运动传感器包括多个陀螺仪和多个线性运动传感器;其中所述多个陀螺仪探测所述运动传感装置自身的运动和方向;

在缓冲器中存储所述运动传感器数据;以及

提供与所述应用程序通信的应用程序接口 API 层,其中所述应用程序开始发送命令到所述 API 层以提供设置,所述设置设定在后台运行的一个或多个算法,其中所述设置在系统操作期间指定所述应用程序想要的与所缓冲的运动传感器数据相关的哪种类型的信息,其中所述设置指示给所述 API 层所述一个或多个算法中的哪些应运行、所述一个或多个算法中的哪些提供关于所述装置的信息回到所述应用程序。

2. 如权利要求 1 所述的方法,其中所述运动传感器数据的处理包括在所述运动传感装置的操作期间在后台运行一个或多个运动算法,所述运动算法为了满足一个或多个条件而检查所述运动传感器数据,这引起所述应用程序被通知。

3. 如权利要求 1 所述的方法,还包括:缓冲以较高采样率和精确定时要求获得的所述运动传感器数据,以及处理所缓冲的运动传感器数据并以较低采样率和放宽的定时要求将所缓冲的运动传感器数据提供给所述应用程序。

4. 如权利要求 1 所述的方法,其中所述多个陀螺仪包括三个陀螺仪,以及其中所述多个线性运动传感器包括三个加速计,所述三个陀螺仪和所述三个加速计提供 6- 坐标轴感测。

5. 如权利要求 1 所述的方法,还包括:检查所述运动传感装置的全部所述运动传感器的全部输出,且在检查所述运动传感器的所述全部输出之后,确定对于所述应用程序实现的应用的类型选择性地仅使用所述运动传感器的坐标轴的子集。

6. 如权利要求 5 所述的方法,还包括:关闭不用于所述应用的类型的运动传感器的操作,以节省能量消耗或处理带宽。

7. 如权利要求 1 所述的方法,其中所述缓冲器是在与应用处理器分离的运动处理单元中实现的硬件缓冲器。

8. 如权利要求 1 所述的方法,其中所述缓冲器是在存储器中实现的软件缓冲器,以及其中中断程序采样所述运动传感器数据并中断运行所述应用程序的应用处理器,以取回至少一部分所述缓冲的运动传感器数据。

9. 如权利要求 1 所述的方法,其中与应用处理器分离的运动处理单元扫描所存储的运动传感器数据以检测与所述应用程序有关的特性。

10. 如权利要求 1 所述的方法,其中与应用处理器分离的运动处理单元将所述缓冲器中的所述运动传感器数据减少到与所述应用程序有关的特征集,以及其中所述缓冲的运动传感器数据的所取回的部分是所述特征集。

11. 如权利要求 1 所述的方法,其中回到所述应用程序的所述信息包括所述装置的位置、方向和通知中的任一项或其任意组合。

12. 如权利要求 1 所述的方法,其中至少一个其它模拟传感器提供传感器数据给所述应用程序。

13. 如权利要求 1 所述的方法,其中至少一个其它传感器提供传感器数据,所述至少一个其它传感器包括罗盘传感器和气压计传感器中任一个或其任意组合。

14. 一种运动传感装置,该运动传感装置包括用于在运动传感装置上提供的一个或多个应用程序的接口,所述运动传感装置包括:

运动处理单元,该运动处理单元包括用于感测所述运动传感装置的运动的多个运动传感器,所述运动传感器包括至少一个陀螺仪和至少一个线性运动传感器;其中所述至少一个陀螺仪探测所述运动传感装置自身的运动和方向;以及

应用处理器,该应用处理器耦合到所述运动处理单元,所述应用处理器提供应用程序,该应用程序实现可用于在所述运动传感装置上使用的多种不同类型的应用之一,以及

与所述应用程序通信的应用程序接口 API 层,其中所述应用程序开始发送命令到所述 API 层以提供设置,所述设置设定在后台运行的一个或多个运动算法,其中所述设置在系统操作期间指定所述应用程序想要的与所缓冲的运动传感器数据相关的哪种类型的信息,其中所述设置指示给所述 API 层所述一个或多个运动算法中的哪些应运行、所述一个或多个运动算法中的哪些提供关于所述装置的信息回到所述应用程序。

15. 如权利要求 14 所述的运动传感装置,其中回到所述应用程序的所述信息包括所述装置的位置、方向和通知中的任一项或其任意组合。

16. 如权利要求 14 所述的运动传感装置,其中至少一个其它模拟传感器提供传感器数据给所述应用程序。

17. 如权利要求 14 所述的运动传感装置,其中至少一个其它传感器提供传感器数据,所述至少一个其它传感器包括罗盘传感器和气压计传感器中任一个或其任意组合。

装置的运动传感器和接口应用程序

[0001] 相关申请的交叉引用

[0002] 本申请要求于 2008 年 1 月 18 日提交的题目为“Motion Sensing Application Interface”的第 61/022,143 号美国临时申请的权益,并且本申请与于 2007 年 7 月 6 日提交的题目为“Integrated Motion Processing Unit (MPU)With MEMS Inertial Sensing And Embedded Digital Electronics”的第 11/774,488 号共同未决的美国实用申请有关,其所有内容在此全文并入作为参考。

发明领域

[0003] 本发明通常涉及运动传感装置,并更特别地涉及装置的运动传感器的接口应用程序。

[0004] 发明背景

[0005] 运动传感器如加速计或陀螺仪经常用在电子装置中。加速计可以用于测量线性加速度,并且陀螺仪可以用于测量角速度。运动传感器的市场包括移动电话、视频游戏控制器、PDA、移动互联网装置(MID)、个人导航装置(PND)、数码照相机、数码摄像机及更多。例如,蜂窝电话可使用加速计以在空间检测装置的倾斜,其允许在对应于倾斜的方位上显示视频图片。视频游戏控制台控制器可使用加速计和/或陀螺仪以检测用于为游戏提供输入的人工控制器的运动。即使在低档或中档的照相机中,图片和视频的稳定性也是重要的特征,其中镜头或图像传感器被移动以补偿由陀螺仪测量的手抖。全球定位系统(GPS)和基于位置的服务(LBS)应用依赖于确定装置的正确位置,并且当GPS信号被削弱或难以获得时,或者为了提高GPS位置查找的正确度,经常需要运动传感器。

[0006] 大部分现有的电子装置趋向于仅使用非常基本的运动传感器,如带有“峰值检测”或稳态测量的加速计。例如,当前移动电话使用加速计以确定装置的倾斜,其可以使用稳态重力测量被确定。这种简单的确定不可以用在使用例如陀螺仪的更加复杂的应用或具有精确定时要求的其他应用中。没有在装置中包括陀螺仪,装置的倾斜和加速度不能被可靠地传感。这在无陀螺仪导航装置中经常引错方向;例如,当汽车在倾斜的交通灯处停止时,“无陀螺仪”推算定位(dead reckoning)特征可以判断错误汽车的位置达几个街区。并且因为装置的运动不总是线性或平行于地面的,所以为了较大的正确度,需要使用加速计或陀螺仪来测量运动的几个不同的坐标轴。

[0007] 更加复杂的运动传感器一般不用在电子装置中。例如,对于带有GPS或较高分辨率照相机的移动电话,图像稳定性和推算定位特征两者都是重要特征,但是在当前移动电话市场上没有可用的解决方案。在特定应用中,对于更加复杂的运动传感器已经做出一些尝试,如检测带有确定动作的运动。但是这些努力中的大部分失败了或作为产品不足够健全。这是因为为了得到运动的运动传感器的使用是复杂的。例如,当使用陀螺仪时,识别装置的倾斜或者移动并非不重要。为了图像稳定性、传感位置、或其他复杂的应用而使用运动传感器需要对运动传感器的深入理解,其使得运动传感设计非常困难。

[0008] 运动传感设计的难点还导致在将设计从一个系统移植到另一系统中的难点。来自

运动传感器的大部分原始数据在运行在装置上的应用软件级别被聚集和处理,其不允许其他应用利用相同的传感器输出。此外,运动传感设计是依赖且用于一个具体应用的一般真实系统,这阻止它被移植到其它系统。例如,图像稳定性软件一般非常依赖于在其中使用它的数码照相机的特定硬件,如可用的图片分辨率和缩放范围。

[0009] 因此,在许多应用中将期望一种系统和方法,其提供简单的应用接口(API)以可用于不同应用,允许运动传感器数据收集更加容易地由用户定义和使用,以及允许对于不同硬件要求的运动传感设计的更加容易地移植和维护。

发明内容

[0010] 本申请的发明涉及将应用程序接口到装置的运动传感器。在发明的一方面,向在运动传感装置上设置的一个或多个应用提供接口包括从在所述运动传感装置上运行的应用程序接收高级命令,其中所述应用程序实现可在所述运动传感装置上使用的多种不同类型的应用之一。所述高级命令请求来源于所述运动传感装置的运动传感器的输出的高级信息,所述运动传感装置包括旋转运动传感器和线性运动传感器。转变所述高级命令以引起由所述运动传感器输出的运动传感器数据的低级处理,所述低级处理遵循应用的类型的要求并响应于接收所述高级命令而智能地确定所述高级信息。所述应用程序不知道所述低级处理,并将所述高级信息提供给所述应用程序。

[0011] 在发明的另一方面,将来自运动传感器的运动传感器数据提供给在运动传感装置上运行的应用程序的方法包括,以第一采样率采样由所述运动传感器输出的运动传感器数据,其中所述运动传感器包括旋转运动传感器和线性运动传感器。所述运动传感器数据被存储在缓冲器中,以及以由所述应用程序实现的应用所需要的第二采样率,取回用于所述应用程序的至少一部分缓冲的运动传感器数据,所述第二采样率比所述第一采样率低。

[0012] 在发明的另一方面,处理来自在运动传感器装置上的运动传感器的运动传感器数据的方法包括,以第一采样率采样由所述运动传感器输出的运动传感器数据,以使所述运动传感器数据用于将信息提供给在所述运动传感器装置上实现第一应用的第一应用程序。所述运动传感器包括旋转运动传感器和线性运动传感器。将所述运动传感器数据低通滤波到比所述第一采样率低的第二采样率,其中被滤波的运动传感器数据用于将信息提供给在所述运动传感器装置上实现不同于所述第一应用的第二应用的第二应用程序。所述第一和第二应用程序同时在所述运动传感器装置上运行,以及所述第一应用需要运动传感器数据以比所述第二采样率大的采样率被采样。

[0013] 本发明的方面提供可用于在运动传感装置上运行的应用的应用程序接口。接口允许在具有运动传感器的装置中使用复杂的运动传感器数据容易地开发应用程序,允许应用程序移植到具有不同硬件和操作系统的不同运动传感装置或从具有不同硬件和操作系统的不同运动传感装置移植,以及提供系统的较容易的维护。发明的其他方面放宽采样率要求并将应用程序的处理减少到与传感器数据有关的采样信息,以及允许具有不同传感器数据采样要求的应用程序同时在运动传感装置上运行。

[0014] 附图简述

[0015] 图 1 是适合用于本发明的运动传感装置的方块图;

[0016] 图 2 是适合用于本发明的运动处理单元的一种实施方式的方块图;

[0017] 图 3 是说明本发明的软件层次的例子的方块图；

[0018] 图 4 是说明在应用处理器上运行的图 3 的软件层的例子的方块图；

[0019] 图 5 是说明访问在运动传感装置上实现的 API 的应用程序的本发明的方法的流程图；

[0020] 图 6 是说明与在运动传感装置上实现的其他软件层接口的 API 的本发明的方法的流程图；

[0021] 图 7 是说明用于同时处理在移动传感器装置上并发运行的不同应用的数据的本发明的方法的流程图；

[0022] 图 8A-8D 是说明用于提供缓冲以减少应用处理器在接收运动传感器输出中的处理的本发明的方法的流程图。

[0023] 详细描述

[0024] 本发明通常涉及运动传感装置,并更特别地涉及装置的运动传感器的接口应用程序。提出以下描述以使本领域普通技术人员能够进行并使用本发明,并且在专利申请和它的要求的背景下提供以下描述。对优选的实施方式和在此描述的一般原则和特征的多种修改对于那些本领域技术人员将容易是明显的。从而,本发明不是用来被限制到所示的实施方式,而是要符合与在此描述的原则和特征一致的最宽范围。

[0025] 为了更加详细地描述本发明的特征,请参考附图 1-8D 连同下面的讨论。

[0026] 图 1 是适合用于本发明的运动传感系统或装置 10 的方块图。装置 10 可以作为装置或设备被提供,如可以由用户在空间移动并且它在空间的运动和 / 或方位因此被感测的便携式装置。这种便携式装置可以是移动电话、个人数字助理 (PDA)、视频游戏机、视频游戏控制器、导航装置、移动互联网装置 (MID)、个人导航装置 (PND)、数码照相机、数码摄像机、双目望远镜、望远镜头、或其他便携式装置,或者这些装置中的一个或多个的结合。在其他实施方式中,装置 10 的一些部分可以在便携式或可移动装置中被提供,而其他部分可以在非便携式装置如台式计算机、电子桌面装置、服务器计算机等中被提供,非便携式装置可以例如通过网络连接与可移动或便携式部分通信。

[0027] 装置 10 包括应用处理器 12、存储器 14、接口装置 16、运动处理单元 20、模拟传感器 22、以及数字传感器 24。应用处理器 12 可以是微处理器、中央处理单元 (CPU)、或运行用于装置 10 的不同软件程序的其他处理器中的一个或多个。例如,可以提供不同软件应用程序如菜单导航软件、游戏、照相机功能控制、导航软件、以及电话或其他功能接口。在一些实施方式中,多种不同的应用可以在单个装置 10 上被提供,并且在那些实施方式的一些中,多种应用可以在装置 10 上同时运行。在应用处理器上运行的软件的例子关于图 3 和 4 在下面被更加详细地描述。

[0028] 装置 10 还包括辅助应用处理器 12 的组件,如存储器 14 (RAM、ROM、闪存等) 和接口装置 16。接口装置 16 可以是将输入和 / 或输出提供给用户的各种不同的装置中的任一种,如显示屏、音频扬声器、按钮、触摸屏、操纵杆、滑动器 (slider)、打印机、扫描仪、照相机等。存储器 14 和接口装置 16 可以通过总线 18 被耦合到应用处理器 12。

[0029] 装置 10 还可以包括运动处理单元 (MPU™) 20。MPU 是包括可以测量装置 10 (或它的一部分) 在空间中的运动的运动传感器的装置。例如,MPU 可以测量装置的一个或多个旋转坐标轴和一个或多个加速度坐标轴。在一些实施方式中,执行这些功能的组件被集成

在单个封装中。MPU20 可以将运动传感器数据传递到接口总线 21,应用处理器 12 也连接到该接口总线 21。在一种实施方式中,处理器 12 是总线 21 的控制器或母版。例如,在一些实施方式中接口总线 21 可以是根据像 I2C 或串行外围接口 (SPI) 总线这样的标准实现的数字串行接口总线。一些实施方式可以提供总线 18 作为与接口总线 21 相同的总线。

[0030] MPU 20 包括运动传感器,其包括一个或多个旋转运动传感器 26 和一个或多个线性运动传感器 28。例如,在一些实施方式中,旋转运动传感器是陀螺仪,以及线性运动传感器是加速计。陀螺仪 26 可以测量包含陀螺仪 26 的装置 10(或它的一部分)的角速度。一般可以提供从一个到三个陀螺仪,其取决于在特定实施方式中期望被感测的运动。加速计 28 可以测量包含加速计 28 的装置 10(或它的一部分)的线性加速度。一般可以提供从一个到三个加速计,其取决于在特定实施方式中期望被感测的运动。例如,如果使用三个陀螺仪 26 和三个加速计 28,那么提供 6- 坐标轴感测装置,用于提供全部六个自由度的感测。

[0031] 在一些实施方式中,陀螺仪 26 和 / 或加速计 28 可以作为微机电系统 (MEMS) 来实现。还可以提供支持硬件,如对于来自运动传感器 26 和 28 的数据的存储寄存器。

[0032] 在一些实施方式中,MPU 20 还可以包括硬件处理块 30。硬件处理块 30 可以包括逻辑或控制器以提供以硬件方式的运动传感器数据的处理。例如,在一些实施方式中,运动算法或算法的部分可由块 30 来实现。一些实施方式可以包括在块 30 中的硬件缓冲器,以存储从运动传感器 26 和 28 接收的传感器数据。例如,在此处描述的一些实施方式中,硬件缓冲器存储由运动算法采样的传感器数据,如在下面关于图 8A-8D 更加详细地描述的。

[0033] 参考图 2,下面描述了 MPU 20 的一个例子。适合用于本发明的 MPU 的其他例子在 2007 年 7 月 6 日提交的题目为“Integrated Motion Processing Unit(MPU)With MEMS Inertial Sensing And Embedded Digital Electronics(带有 MEMS 惯性传感和嵌入式数字电子装置的集成运动处理单元 (MPU))”的第 11/774,488 号共同未决的美国专利申请中被描述,并在此全文并入作为参考。从 Invensense, Inc. of Santa Clara, CA 可得到用于装置 10 中的 MPU 20 的适当的实现。

[0034] 装置 10 还可以包括其他类型的传感器。模拟传感器 22 和数字传感器 24 可以用于提供关于装置 10 所处于的环境附加的传感器数据。例如,可以提供传感器如气压计、罗盘 (compass)、温度传感器、光传感器(如照相机传感器、红外传感器等)、超声波传感器、射频传感器、或其他类型传感器。在所示的代表性实现中,数字传感器 24 可以提供直接到接口总线 21 的传感器数据,而模拟传感器可以提供到模数转换器 (ADC) 34 的传感器数据,其中 ADC 34 将数字形式的传感器数据提供给接口总线 21。在图 1 的例子中,ADC 34 在 MPU 20 中被提供,以便使 ADC 34 可以提供转换的数字数据到 MPU 的硬件处理 30 或到总线 21。在其他实施方式中,ADC 34 可以在装置 10 的别处实现。

[0035] 图 2 显示适合用于在此描述的发明的运动处理单元 20 的实施方式的一个例子。图 2 的 MPU 20 包括算术逻辑单元 (ALU) 36,其执行对传感器数据的处理。ALU36 可以由存储在程序 RAM(随机存取存储器)37 中并从程序 RAM37 取回的一个或多个程序智能地控制。ALU 36 可以控制直接存储器存取 (DMA) 块 38,其可以从运动传感器如陀螺仪 26 和加速计 28、以及其他传感器如温度传感器 39,独立地读取 ALU 36 或其他处理单元的传感器数据。一些或全部传感器可以在 MPU 20 上被提供或者是在 MPU 20 外部被提供;举例来说,在图 2 中所示的加速计 28 在 MPU 20 外部。DMA 38 还可以关于读或写操作的状态而将中断信号提

供给 ALU。DMA 38 可以将从传感器读取的传感器数据提供给数据 RAM 40 用于存储。数据 RAM 40 将数据提供给 ALU 36 用于处理,以及 ALU 36 将包括所处理的数据的输出提供给数据 RAM 40 用于存储。总线 21(也在图 1 中示出)可以被耦合到数据 RAM 40 和 / 或 FIFO 缓冲器 42 的输出,以便使应用处理器 12 可以读取由 MPU 20 读取和 / 或处理的数据。

[0036] FIFO(先入先出)缓冲器 42 可以用作为硬件缓冲器,用于存储可由应用处理器 12 通过总线 21 访问的传感器数据。在下面的几个实施方式中,描述了硬件缓冲器如缓冲器 42 的使用。例如,多路复用器 44 可以用于选择 DMA 38 将原始传感器数据写到 FIFO 缓冲器 42,或是数据 RAM 40 将所处理的数据写到 FIFO 缓冲器 42(举例来说,由 ALU 36 处理的数据)。

[0037] 如在图 2 中所示的 MPU 20 从而可以支持处理运动传感器数据的一个或多个实现。例如,MPU 20 可以充分地处理原始传感器数据,其中在程序 RAM 37 中的程序可以控制 ALU 36 智能地处理传感器数据并将高级数据提供给应用处理器 12,并且应用程序在其上运行。或者,原始传感器数据可以使用 ALU 36 由 MPU 20 预处理或部分地处理,其中在将作为结果的高级信息提供给应用程序之前,所处理的数据然后可以由应用处理器 12 取回,用于在应用处理器 12 上附加的低级处理。或者,原始传感器数据仅仅可以由 MPU 20 缓冲,其中原始传感器数据由应用处理器 12 取回,用于转为应用程序的高级数据的所需要的低级处理。在一些实施方式中,在相同的装置 10 上运行的不同应用或应用程序可以使用最适合于应用或程序的这些处理方法中的不同的几种。

[0038] 图 3 是说明本发明的软件层次 50 的例子的方块图,其可以在运动传感装置 10 的应用处理器 12 上实现 / 执行。可以在计算机可读介质如电子存储器或其他存储介质如硬盘、光盘等上提供软件的层。

[0039] 应用软件层 52 包括一般由一个或多个应用开发人员提供的并在应用处理器 12 上运行的一个或多个应用程序。应用程序在装置 10 上实现特定应用(或它的部分)(其还可以被称为“模式”),举例来说,每个应用由一个或多个应用程序来实现。应用程序可以提供处理和输入 / 输出功能以及对其应用特有的适当的用户接口功能,如在照相机应用中控制照相机镜头和显示图像、在游戏应用中实现游戏、在导航应用中输出装置 10 的方向和当前位置等。

[0040] 应用软件层 52 与系统软件 54 通信,其管理装置的资源,包括在硬件和软件组件之间的通信。在嵌入式系统应用中用于运动感测的软件结构可以在如图 3 中所示的单独的层中被定义。从而,系统软件 54 包括应用程序接口(API)层 56、实时操作系统(RTOS)层 58、运动算法层 60、以及传感器装置驱动器层 62。适当地,还可以包括附加层。

[0041] API 层 56 提供提取的且高级的功能集用于应用程序使用,其使在应用程序和装置 10 的较低级组件如运动传感器之间的接口和通信大大简单化。通过提供与运动传感器数据有关的使用的不同低级功能或算法的“运动库”,API 层促进到装置 10 的运动感测引擎的接口。在 API 层 56 内的特定 API 可以被定义,以对应于一个或多个运动算法,其中那些对应的算法可以通过应用访问该 API 而被使用。

[0042] 基于从装置 10 的运动传感器采样的运动传感器数据,API 层 56 可以提供预先确定的、高级的、提取的命令以及关联的控制参数或设置以由应用程序使用来接收信息,包括高级信息。API 层 56 将高级命令从应用程序转变为实现命令所需要的必要的运动算法,并

且从而可以将对与特定应用有关的信息的请求转变为该特定应用必要的处理。在装置 10 上运行的每种类型的应用需要基于该应用类型的功能的基于运动的高级数据特定集合。例如,用户运动 / 光标控制类型的应用基于装置 10 的运动可需要 2-D 或 3-D 坐标,而图像稳定性类型的应用基于装置 10 的运动可需要模糊指示或有关信息。API 访问的运动算法知道与运动传感器数据的低级处理关联的精确要求,以获得由应用程序命令的特定类型的高级信息,其与关联于高级命令的应用有关。例如,运动算法在已接收到指示它将用于图像稳定性应用或功能的高级命令后,知道以较高的采样率如 500Hz 采样来自运动传感器的运动传感器数据,用于图像稳定性应用。

[0043] 从而低级处理可以智能地确定被应用程序所需要的高级信息,并将它提供给应用程序,而应用程序可以不知道运动传感器数据的任何所需的低级处理和在该处理中使用的、被需要以用来获得高级信息的处理和定时要求。例如,应用程序开发人员仅需要用 API 命令构成调用,例如,以接收适合于开发人员的应用的所处理的运动传感器数据,没有必要知道如何收集和处理原始传感器数据。API 层 56 可以被定义为尽可能独立的操作系统,以使应用软件尽可能可移植。

[0044] API 层 56 可以提供多种特定类型的 API,其对于可以在装置 10 上实现和使用的一种或多种类型的应用是有用的。例如,特定的运动传感类型的 API 可以被提供,其可由确定的应用如蜂窝电话或游戏应用来使用。多种不同类型的应用可以优选地在装置 10 上实现,并且提供对于这些应用的多个不同的 API。一些类型的应用可不需要特定类型的 API。例如,导航应用可不需要使用图像稳定性 API。

[0045] 在一些实施方式中,API 可以被分解为低级 API (举例来说,原始数据获得、校准数据获得、初始化传感器、设置参数、执行数学功能),中级 API (举例来说,获得所处理的数据如滚动、倾斜、以及偏航、航向、位置、或模糊数量),以及高级 API (举例来说,在一行代码中运行 DRKalman 滤波)。中级 API 可由希望由更加基本的算法组合块构造它们自己的应用的应用开发人员使用。其他实施方式可以仅提供两个 API 层级。在系统操作的期间,API 软件有效地在后台运行,并且在层 52 中的应用程序可以以几种不同的计算机语言或格式中的任一种被提供。从而,可以为允许其用于其他程序语言的 API 层 56 创建“组合(binding)”。

[0046] RTOS 层 58 可以作为装置 10 的操作系统被提供,以实时控制和管理系统资源并启用应用软件 52 以及其他层如 API 层 56、运动算法层 60 以及运动传感器装置驱动器 62 的功能。操作系统可以将应用程序与装置 10 的其他软件和功能接口。RTOS 层 58 可以是适于具有实时操作的嵌入式系统的多种可用的操作系统中的任一种。RTOS 层 58 可以与其他层中的任一层通信,其包括使用由 RTOS 层 58 提供的功能的应用软件 52。RTOS 层 58 可以作为装置 10 的具体操作系统或作为适用于装置 10 的更加普通的操作系统来实现。

[0047] 运动算法层 60 提供运动算法的“运动库”,其向由运动传感器和其他传感器提供的原始传感器数据提供智能、低级处理。在层 60 中的这些算法可以获得原始传感器数据并将数据结合和形成为高级信息、和 / 或具体应用信息。

[0048] 例如,运动算法可以从几个运动传感器获得运动传感器数据,并将数据处理和 / 或结合为对于特定应用有用的高级信息,特定应用如装置的滚动 / 倾斜 / 偏航,或用于在二维屏幕上定位光标的 X 和 Y 坐标。算法一般具有精确的定时要求,用于采样来自运动传感器的运动传感器数据。例如,算法可以积累运动传感器数据并合并数据点,以获得装置 10

的运动角度或其他运动特征。另外,除运动传感器数据之外,算法还可以使用来自一个或多个其他传感器的数据以确定高级信息,来自其他传感器如来自压力、温度、和 / 或方向传感器。算法的输出可以被上报到 API 层 56 或由 API 层 56 取回,API 层 56 将高级信息提供给应用程序,并因此由运动算法执行的处理不需要被任何应用开发人员知道。开发人员仅需要使用在 API 层 56 提供的 API 功能,以请求期望的基于运动传感器数据的高级信息。

[0049] 运动算法可以与一个或多个特定应用和 / 或 API 关联。例如,“运动感测”API 可以使用手势识别运动算法,其检查传感器数据序列并当确定的“特征”如手势被检测时输出这样的特征。在 API 层 56 中依次定义该手势检测为简单命令、应用程序的输入或事件。运动算法可以被设计为尽可能独立的操作系统,其减少在不同系统之间移植算法的阻碍。

[0050] 传感器装置驱动器层 62 将软件接口提供给装置 10 的硬件传感器。驱动器层 62 提供与装置 10 的多种类型传感器的接口,传感器包括陀螺仪 26、加速计 28、以及如参考图 1 描述的其他传感器 22 和 24。例如,来自运动传感器 26 和 28 的所感测的结果可以被硬件写入在 MPU 中的寄存器内,并且那些寄存器然后由装置驱动器层 62 访问。驱动器层 62 然后将所访问的传感器数据提供给在运动算法层 60 中的一个或多个运动算法用于处理。在一些实施方式中,装置驱动器层 62 与 MPU 的硬件处理 30 接口,其中硬件处理可已经使用运动算法处理一些传感器数据。

[0051] 图 4 是说明在应用处理器 12 上运行的图 3 的软件层的例子方块图。应用软件层 52 可以包括多个不同的应用程序以实现多种不同类型的应用。例如,移动电话应用可以通过一个或多个应用程序 60 来实现,用于实现电话功能(呼叫其他电话、接收呼叫并将它们提供给用户等)以及菜单导航。一个或多个游戏应用可以通过应用程序 62 来实现,其提供不同的视频游戏,如在装置 10 的二维显示屏上显示图形图像并从装置 10 的音频扬声器输出声音,以及基于通过按钮、旋钮、操纵杆、触针、触摸垫或触摸屏、麦克风、或其他输入装置的用户输入改变输出。健康应用可以通过健康和计步器应用程序 64 来实现,用于控制和显示测量用户的步行距离并测量其他身体特征的计步器。数码照相机应用可以通过应用程序 66 来实现,用于控制数码照相机或数码摄像机,包括图像稳定性、缩放和分辨率功能等。导航应用可以通过应用程序 68 来实现,用于个人导航装置(PND)、步行推算定位(PDR)、或其他导航功能,并可以包括应用程序以控制这些装置和输出导航结果到用户,如当前位置。

[0052] RTOS 层 58 可以监视包括 API 层 56、运动算法层 60 以及传感器装置驱动器层 62 的其他软件层。API 层 56 可以包括不同类型的 API,每个 API 表现为具体类型的高级信息,其可用于可以由装置 10 执行的一个或多个应用程序。在图 4 的例子中,API 层 56 包括运动感测 API 70、图像稳定 API 72 以及导航 API 74。

[0053] 运动传感 API 70 可以包括高级命令和功能,用于请求描述装置 10 的运动(或它的一部分)或描述由装置的运动确定的特征的高级信息。这种高级功能可以包括手势识别,其识别由用户输入的特定运动以调用特定的命令。例如,手势可以包括在特定的方向 / 方位或在方向 / 方位的序列中、和 / 或在指定的持续时间段移动装置 10,以启动与该手势关联的功能,如“删除”、“随动拍摄”、“缩放”等。API 70 可以允许应用程序开发人员简单地请求特定的预先确定的手势中的任一个是否已经被检测,开发人员没有必要知道如何使用原始运动传感器数据检查那些手势或如何使用适当的算法处理该传感器数据。其他运动传感器应用包括请求有关于用户接口功能如滚动图像或光标控制的运动数据,以及请求指示输

入已经提供给游戏的数据,如运动 / 方位数据或在屏幕上移动图形对象或激活在游戏中的功能的手势,如使武器开火或打开门。

[0054] 在 API 层 56 中的不同 API 的另一例子是图像稳定 API 72。该 API 允许应用程序请求有关于高级图像稳定功能的装置 10 的状态,举例来说,用于数码照相机或摄像机中。例如,这种高级图像稳定功能可以包括估计在图像中的模糊的数量或程度,其中图像基于在图像捕获期间检测的特定运动的数量由照相机镜头捕获。API 72 可以允许应用程序简单地请求在图像中模糊的数量,其中应用程序不必须使用原始传感器数据计算该模糊或用算法处理原始传感器数据。其他高级图像稳定功能可以包括图像重组,其中应用程序可以请求在捕获的图像上进行数字处理,以纠正或重组原始色彩或模糊图像的其他特征,没有必要执行该数字处理本身。其他图像稳定功能可以包括对于有关于运动感测的特定特征(如模糊)产生点扩展函数(PSF)、混合图像、或评估图像。

[0055] 在 API 层 56 中的不同的 API 的另一例子是导航 API 74。该 API 允许应用程序请求有关于高级导航功能的装置 10 的状态,举例来说,用于导航装置。例如,这种高级导航功能可以包括提供装置 10 的推算定位位置,以及在这种推算定位估计中补偿重力。API 74 可以允许应用程序简单地请求在数据中补偿该重力,而不必要执行该补偿本身。其他高级导航功能可以包括将导航 Kalman 滤波应用到传感器数据,以当提供关于装置 10 的位置和 / 或速度的连续更新信息时,补偿在记录的位置中的误差,以及设置噪声矩阵或处理矩阵。

[0056] 在 API 层 56 中的不同 API 可以全部在单个装置 10 上被提供,其现在在应用层 52 中的不同的多个应用。使用 API 的方便允许大量的完全不同种类的应用在单个装置 10 上被提供,而不需要每个应用开发人员在对于特定应用的运动传感器数据的收集和处理中具有专业知识。在一些实施方式中,多个应用可以在装置 10 上被同时执行。这在下面关于图 7 被更加详细地描述。

[0057] 运动算法层 60 可以包括算法,如滤波,其被在 API 层 56 中的 API 调用。运动算法中的一些可以是基本功能算法,如提供装置 30 在空间中的当前方位,或指示在空间中装置当前是否被移动。算法可以结合不同传感器的输出并处理结果(举例来说,使用积分、导数等),以传送较高精确性的感测结果,以及提供来源于原始传感器数据的较高级信息。

[0058] 还可以对于特定的应用包括更加具体的算法。例如,运动算法程序可以包括手势识别算法 80,用于和运动传感 API 70 一起使用,其确定是否由用户使用输入装置已经做出特定的手势。这种算法可以基于传感器数据等确定用户是否输入“轻敲”手势或其他手势,或对于更加“中级”的 API 提供装置的滚动、倾斜、以及偏航或装置的 z- 旋转。同样地,基于来自原始运动传感器数据的由算法确定的滚动、倾斜和偏航数据,光标控制算法 82 可以与运动传感 API 70 一起使用以确定用户输入装置的 X-Y 坐标,在其位置处光标被显示在显示屏上。其他用户接口和连续的控制算法还可以用于,如确定在显示屏上的图标是否已经基于传感器输入被用户选择、在光标控制期间手抖的补偿、以及基于运动传感器数据移动一个或多个所显示的图像。游戏算法 84 可以用于运动传感 API 70 以提供有关于游戏应用的数据,如特定手势或运动是否被接收,应用将其用作输入以控制游戏功能。

[0059] 图像稳定算法可以包括模糊估计和图像重组算法。模糊算法 86 可以用于在图像捕获时,例如通过照相机镜头捕获图像的时候,基于运动传感器数据确定在图像中模糊是否发生。例如,点扩展函数(PSF)可以被算法使用以确定模糊。图像重组算法 88 可以

用于纠正或重组图像的特征,其例如通过模糊已失真或被移除。其他图像稳定算法在第 11/774,488 号共同未决专利申请的美国专利申请中被描述,并在此全文并入作为参考。导航算法可以包括可以用于当确定推算定位时补偿重力的推算定位重力补偿算法 90 和可以用于在感测位置补偿误差的导航 Kalman 滤波 92。例如,推算定位位置算法可以使用来自一个或多个加速计的数据以确定装置 10 的位置,以及来自一个或多个陀螺仪 26 的数据以确定装置 10 的运动航向或方向。

[0060] 装置驱动器层 62 将软件接口提供给装置 10 的硬件运动传感器 26 和 28 以及其他传感器 22 和 24,如上关于图 3 所描述的。层 62 可以包括用于陀螺仪 26、加速计 28、气压计、罗盘传感器、以及如参考图 1 所描述的其他传感器的驱动器。

[0061] 图 5 是说明访问在运动传感装置 10 上实现的 API 的应用程序的本发明的方法 200 的流程图。方法 200 和在此描述的其他方法,如果没有具体描述,可以在硬件、软件、或硬件和软件两者的组合中实现。在此的方法可以使用程序指令或程序代码来实现,程序指令或程序代码可以由处理器或计算机执行并可以在计算机可读介质如电子存储器、磁带、磁盘、光盘等上被提供。

[0062] 在方法 200,应用程序与提供对于应用程序有用的运动传感功能的 API 层 56 的特定 API 接口。在一些实施方式中,多个应用可以与单个 API 接口,而其他应用程序可具有对于他们唯一使用的专用 API。当需要时应用程序还可以访问多种类型 API。与 API 的接口是在高级别上,允许应用程序忽视它请求的信息的实现细节。从而,例如,到 API 的命令可以相当于“提供装置的当前位置”或“提供最近的图像的模糊核心”或“提供装置当前正呈现的运动的类型”。

[0063] 本发明的 API 还可以使用“事件驱动程序设计”。例如,当使用运动传感 API 时,应用程序一般不需要每 10ms 检查运动传感器的输入。代替地,如果应用程序正在寻找包括特定运动传感器数据的手势,当该手势被检测到(这是“事件”)时它请求 API 中断应用程序。下面在列表 1 和 2 中显示应用程序使用本发明的 API 的例子,其中“ml”代表“运动库”。

[0064] mlInitializeSensors();

[0065] mlRegisterGestures(ML_DOUBLE_TAP | ML_ROLL);

[0066] mlGestureCallback(got Gesture);

[0067] void gotGesture(gesture) {

[0068] if(gesture == ML_DOUBLE_TAP) {

[0069] [... 如果检测到双击手势,则执行这里列出的功能或操作。]

[0070] }

[0071] else if(gesture == ML_ROLL) {

[0072] [... 如果检测到滚动手势,则执行这里列出的功能或操作。]

[0073] }

[0074] }

[0075] 列表 1

[0076] ml.Bind(ml.EVT_DOUBLE_TAP, OnDoubleTap) // 当检测到双击手势

[0077] 时,运行功能“OnDoubleTap”

[0078] ml.Bind(ml.EVT_ORIENTATION_CHANGE, OnOrientationChange) //

```
[0079] 当改变方向时,运行功能“OnOrientationChange”
[0080] ml.Bind(ml.EVT_MOTION, OnMotion)// 当便携式装置移动时,运行功
[0081] 能“OnMotion”
[0082] ml.MainLoop()// 开始无限循环,运行在后台的引擎。除非事件需要,
[0083] 不再运行程序。
[0084] def OnDoubleTap():
[0085]     [... 如果检测到双击手势,则执行这里列出的功能或操作。]
[0086] def OnOrientationChange():
[0087]     [... 如果检测到方向改变,则执行这里列出的功能或操作。]
[0088] def OnMotion():
[0089]     [... 如果检测到移动,则执行这里列出的功能或操作。]
[0090] 列表 2
```

[0091] 图 5 的方法 200 在 202 开始,并在步骤 204,应用程序开始发送命令到 API 以提供设置,其设置或“记录”在装置 10 的操作期间在后台作为“回调函数”运行的一个或多个算法(如滤波)。设置可以在系统操作期间指定应用程序想要的哪种类型的信息。这种设置指示给 API 哪些算法应运行,并提供信息(如位置、方位、通知等)回到应用程序。例如,当应用程序被用户在空间中四处移动时,应用程序可希望通过装置 10 的位置和方位被连续地更新。应用程序可以发送设置到将设置算法的 API,以将期望的信息连续提供给应用程序。在一些实施方式中,应用程序可以可选地指定参数以定制它将接收的高级信息,如设置应用程序将从 API 接收信息的速率、在其期间这种信息将被发送到应用程序的持续时间的总量、其应被通知的次数等。另外,应用程序可以发送指示一个或多个条件的设置,当基于传感器数据确定那些条件被满足时,一个或多个条件引起信息被发送到应用程序。例如,设置可以指定哪些特定运动或手势在被检测时将触发 API 以通知应用程序其事件,或指定为了将它认为是“双击”手势的两次所检测的敲击之间所需要的时间总量。应用程序还可指定数据被处理要用的算法或滤波,或指定详细资料,如指定使装置旋转所通过的连续运动和/或角度的积分,以引起触发。

[0092] 在步骤 206,应用程序基于被请求在后台运行的调回函数算法而从 API 接收信息,包括高级信息。API 基于适当的算法的操作,将期望的信息提供给应用程序,适当的算法基于步骤 204 的信息开始并在后台运行。根据由应用程序指定的或对于由 API 和较低层实现的请求所固有的任何要求,步骤 206 可以连续地被执行。例如,在照相机应用中,应用程序可需要连续地知道在照相机上是否有过多的手抖,其中如果确实如此,应用程序可以发布警报信号给用户。步骤 206 还可在不规则的间隔中被执行,如当符合条件(如由应用程序在前指定的)时。例如,如果装置 10 在某方向上倾斜得太多,应用程序可能需要发布输出到用户,并需要被通知这一倾斜。步骤 206 可在方法 200 的其他步骤的执行期间被执行。

[0093] 在步骤 208,应用程序检查它是否需要信息以完成它的执行,这样的信息是(至少部分地)基于由装置 10 的运动传感器读出的传感器数据。例如,应用程序可需要知道装置的当前位置以响应于来自用户的对于装置位置的请求。如果在步骤 208 没有信息被需要,那么处理返回到步骤 206 以接收信息(如果合适)。如果需要信息,那么在步骤 210 应用程序使用由 API 定义的高级命令从 API 请求所需的高级信息。在步骤 212,应用程序从 API

接收所请求的信息。处理然后返回到步骤 206。

[0094] 图 6 是说明 API 与在运动传感装置 10 上实现的其他软件层接口的本发明的方法 250 的流程图。过程在 252 开始,并在步骤 254, API 从应用程序接收高级命令以设置运行的算法。高级命令提供设置,设置指示应用程序(至少部分地)基于由装置 10 的传感器提供的运动传感器数据期望从运行的算法接收的哪项高级信息。参考图 5 如上说明的,设置可指定哪种类型的信息、信息要被发送的速率、信息要被发送的条件等。API 将来自应用程序的命令和设置转变为特定运动算法,其将遵循关联的应用的精确要求提供所请求的信息,举例来说,通过处理以该应用所需的采样率从适当的传感器采样的运动传感器数据。

[0095] 在步骤 256, API 基于接收到的命令和设置启动所请求的运动算法,并且这些算法在装置 10 的操作期间在后台运行。API 可以运行满足应用程序的请求和设置的特定算法。在一些实施方式中,来自应用程序的命令是足够高级的,它没有指定在后台运行的特定算法,而 API 可以将正确的算法关联到接收到的命令。在可选择的实施方式中,接收到的命令可以指定特定的算法。算法可以用应用所需要的精确定时采样并处理传感器数据,并以应用程序的所需定时向应用程序上报所得到的信息。算法还可以检查由应用程序指定的任何条件。在一些实施方式中,较高软件层,如在 API 中,可以检查一些或全部这样的指定条件。

[0096] 在步骤 258,如果请求任何这样的信息,其包括高级信息,则 API 将连续的信息提供给应用程序。例如,API 可以从运动算法层 60 中的一个或多个算法接收所处理的信息,并将该数据提供给应用程序。API 或算法以适合于由应用程序实现的应用的特定速率,优选地提供信息,或允许应用程序取回和采样信息。

[0097] 在步骤 260, API 检查以查看它是否对于特定信息从应用程序接收请求。如果没有,则启动步骤 264,在下面描述。如果接收到,则在步骤 262, API 将所请求的信息提供给应用程序。通过查询适当的算法和从算法接收信息,获得该信息。过程然后返回到步骤 258。在步骤 264, API 检查以查看触发了信息到应用程序的发送的事件是否发生。例如,事件可以由应用程序指定的或符合 API 的一个或多个条件。如果没有事件发生,则过程返回到步骤 258。如果事件发生,在步骤 266, API 将所触发的信息(一般地,高级信息)提供给应用程序。从适当的算法获得该信息。过程然后返回到步骤 258。

[0098] 在一些实施方式中,运动算法的低级处理包括智能处理,其对于由接收到的高级命令所指示的应用的类型和由 API 发送到运动算法的参数或指令是所特有的。“内容感知”API 从而可以被提供,其足够地智能以知道装置 10 的什么应用当前正在运行并相应地启用与那些应用关联的关键特征。例如,如果当前在使用中的运行的应用是游戏应用或导航应用,则 API 分别启用有关于游戏或导航的适当的特征,包括读取的适当传感器和用于读取这些传感器的定时要求,以及启用适当的高级信息和功能,以将包括接口功能和输出的信息和功能提供给用户。

[0099] 例如,低级处理可以包括检查运动传感装置的全部运动传感器的全部输出,以确定由应用程序所实现的应用类型所需要的高级信息。依赖于应用的类型,低级处理在检查全部的运动传感器的输出后,可以选择性地仅使用应用的类型所需要的运动传感器的坐标轴的子集。例如,在运行在 6-坐标轴装置 10 上的特定的应用中,算法可以检查全部三个陀螺仪和全部三个加速计以确定装置 10 或者是在空间保持垂直,而在这样情况下仅来自加速计的感测的数据被处理以产生该应用的高级信息;或是装置在空间保持水平,而在这样

情况下仅来自陀螺仪的感测的数据对于该应用被处理。在其他实施方式中,基于应用的类型,由 API 接收的高级命令可以确定运动传感器的哪些坐标轴和输出选择性地用于确定高级信息。在一些实施方式中,高级命令不需要指定使用的特定坐标轴和输出;相反地,基于特定的高级命令是什么和 API 与该高级命令关联的处理,低级处理可以知道使用哪些传感器输出。

[0100] 在一些实施方式中,运动传感器的仅一些坐标轴或输出的选择使用可以进一步地通过关闭没有用于任何当前运行应用的运动传感器(或它的特定的坐标轴)的操作或感测功能来使用。这可以节省装置 10 的能量消耗和/或处理带宽。

[0101] 进一步地电源管理特征还可以被包括在 API 层中,包括睡眠模式和唤醒模式,以及运动传感器资源的全部功率管理。例如,API 层可以实现一个或(多个)在应用处理器 12 或 MPU 20 上运行的程序,应用处理器 12 或 MPU 20 在任何陀螺仪 26 没有感测任何数据的时候或如果它们感测的数据没在使用时,关闭陀螺仪 26 的电源,其中陀螺仪 26 可消耗被其他传感器更多的能量。在一种实施方式中,一旦由其他运动传感器感测装置 10 的某一运动,陀螺仪就可以被“唤醒”。

[0102] 下面提供了根据本发明可用于 API 的 API 命令的例子。

[0103] mlInitializeSensors()
[0104] mlGetRawData()
[0105] mlGetCalibratedData()
[0106] mlGetAugmentedData()
[0107] mlGetHumanMotion()
[0108] mlGetRotationMatrix()
[0109] mlGetQuaternion()
[0110] mlGetRoll
[0111] mlGetPitch()
[0112] mlGetYaw()
[0113] mlHeading()
[0114] mlGetLocation()
[0115] mlGetPSF()
[0116] mlGetBlurAmount()
[0117] mlCalibrate()
[0118] mlUpdateState()
[0119] mlUpdateRotationMatrix()
[0120] mlUpdateHumanMotion()
[0121] mlUpdateGesture()
[0122] mlRegisterGestures()
[0123] mlGestureCallback()
[0124] mlUpdateKalmanFilter()
[0125] mlUpdateSensorCalibration()
[0126] mlSetNoiseMatrix()

[0127] mlSetProcessMatrix()

[0128] mlGeneratePSF()

[0129] mlBlendImages()

[0130] mlEvaluateImage()

[0131] 同时的多个应用的处理

[0132] 图 7 是说明通过一个或多个并发运行的应用程序,用于同时处理同时在移动传感器装置 10 上运行(即,同时被实现)的不同应用的数据的本发明的方法 300 的流程图。方法 300 假定多个应用正并发运行且需要运动传感器数据或来源于运动传感器数据的数据。在一些实施方式中,方法 300 可以由运动算法层 60 来实现。

[0133] 至少两个并发运行的应用需要运动传感器数据以不同的最小采样率被采样。例如,图像稳定性应用需要运动传感器数据以高速率如 500Hz 被采样,而用户接口应用可需要运动传感器数据以较低速率如 50Hz 被采样。

[0134] 处理在 302 开始,并且在步骤 304,传感器数据在第一采样率被接收。第一采样率被确定为在被在装置 10 上当前运行的任何应用所需要的的最高的最小采样率(还可以使用比所需的最小值大的采样率)。在系统 10 上可得到的但非当前运行的其他应用程序的采样率与该第一采样率的确定无关。在步骤 306,以第一(最高)采样率采样的传感器数据用于在装置 10 上运行的第一应用(或如果多于一个应用需要数据以第一采样率被最低限度地采样,则用于多个第一应用)。如上所描述,在层 60 中的运动算法可以处理传感器数据并将所处理的高级信息提供给 API 和实现第一应用的第一应用程序。

[0135] 在步骤 308,对在步骤 304 以第一采样率接收的传感器数据使用低通滤波,以实现适合于在装置 10 上并发运行的不同的第二应用的较低的第二采样率,但是其是过低的速率而不能用于第一应用。例如,第二应用可仅需要以比第一应用低的速率采样的传感器数据,以便可以使用滤波来减少第一采样率。步骤 308 的低通滤波可以是多种不同类型中的任一种,如点运行平均(point running average)、无限冲激响应(IIR)滤波、有限冲激响应(FIR)滤波等。在步骤 310,在第二采样率采样的传感器数据用于第二应用(或如果多于一个应用可以在第二采样率使用传感器数据,则用于多种应用)。对于方法 300 的其他步骤,步骤 310 和 306 可以同时执行。

[0136] 如果附加的应用当前正运行且可以用不同的采样率使用数据,那么在步骤 312,低通滤波类似地用在被滤波的传感器数据上,以实现对于当前运行的附加的应用更加较低的采样率,如果合适,并且采样数据用于那些附加的应用。如果这种应用正在运行,步骤 308 的第二采样率是被当前正运行的任何应用程序所需要的第二最高采样率,并且步骤 312 的采样率是比步骤 308 的第二采样率更低的采样率。例如,如果两个附加的应用可以使用不同的较低采样率,那么以第二采样率提供的数据被滤波,以为应用中的适当的一个提供第三采样率,并且以第三采样率的数据被滤波,以为其他应用提供第四较低采样率。然后在 314 过程完成。

[0137] 方法 300 从而允许带有不同采样率要求的不同应用在装置 10 上通过以树的形式低通滤波数据而同时被执行,其通常是困难的。例如,模糊计算可需要传感器数据以 500Hz 被采样,用户接口应用可需要 100Hz 的采样率,且导航应用可需要 20Hz 的采样率。在应用程序级别,用户接口显示可以 10Hz 被更新,而导航数据以 1Hz 与 GPS 数据结合。为了

同时处理多个应用的传感器数据,同时运行这些应用的运动传感引擎必须能够以不同采样率同时接收运动传感器数据。例如,当用运动手势控制照相机用户接口时,用户可想要获得没有模糊的图片。同时,装置可以是推算定位,以使在照相机应用期间不失去装置的位置。对于启用了 GPS 的照相机-电话,每当获得图片时,用户可能不想失去他的或她的位置。

[0138] 在方法 300 中描述的以树的形式滤波允许不同的采样率从单个最高采样率被提取。例如,为了电子图像稳定性,原始传感器数据可以用 250Hz 抗混叠滤波器以 500Hz 被采样。步骤 308 的 10 点运行平均(举例来说,可以使用的一种类型的低通滤波)提供更多抗混叠,将带宽从 500Hz 减少到 50Hz,对于用户接口应用允许 100Hz 采样率(应用可以被允许采样率等于两倍的传感器采样率)。可以在 100Hz 使用另一 10 点运行平均,提供适合于 20Hz 导航滤波的 10Hz 带宽。全部这些滤波的输出可以它们各自的采样率被更新,而用户并不知道。由于仅图像稳定滤波需要以最高采样率运行,这最小化了应用处理器时间。

[0139] 在另一例子中,图像稳定性应用可以与用户接口应用结合。原始加速计数据或增加的数据(在该例子中,来源于来自加速计 28 和陀螺仪 26 两者的传感器数据的重力加速度)可以用于跟踪照相机的方位,用于图像旋转(人像或风景)和校平(leveling)信息。同时,手势还可以由用户输入以控制照相机模式,例如从照相机模式转换成摄像机模式。使用本发明,与用于手势识别数据的低采样率陀螺仪传感器数据同时,捕获用于模糊数据的高采样率陀螺仪传感器数据。

[0140] 而在另一例子中,导航可以与用户接口应用结合。除为了导航应用运行推算定位算法之外,还可需要手提装置运行地图旋转算法和对于用户接口应用的用户输入算法。使用本发明,由于地图旋转需要低等待时间以便成为有效用户接口,因此这些算法可以不同的采样率同时运行,但是导航算法可仅需要以 20Hz 运行。

[0141] 由本发明提供的不同的采样率还可以用于装置 10 的功率管理。在一些实施方式中,可以减少装置 10 的功率消耗,其对于手持和便携式应用如步行导航可以是重要的。例如,在导航应用中,GPS 单元(未示出)可以是连接到装置 10 的附加单元,以确定装置的位置。如果与 GPS 接收机单独用于确定位置相比,与 GPS 接收机结合使用的在层 60 中的推算定位算法可以更加准确地确定当前位置。推算定位滤波具有附加的益处,其可以减少 GPS 单元所需要的处理,从而减低功率。如果和 GPS 位置确定一起使用推算定位,1Hz 的 GPS 采样率可下降到 0.1Hz,或卫星跟踪器可跟踪仅 3 个卫星而不是 12 个卫星。在 GPS 算法中的任何错误可以由通过推算定位算法实现的附加的信息来补偿。

[0142] 硬件和软件缓冲

[0143] 硬件如 MPU 20 用在装置 10 中的一个原因是减少在装置 10 的应用处理器 12 和/或其他处理器上的处理负荷。由于处理器/CPU 的系统开销、定时约束、以及编码或用户化和可移植性的难点,仅使用软件来捕获运动传感器数据可以是效率低的。运动传感驱动器理想地应使用 CPU 时间的仅仅 1%,且保证正确的定时在移动装置操作系统内可能是困难的。然而,由于几个原因,包括增加的成本,以及不准确且有限的特征,硬件不用于执行全部传感器功能。例如,将附加的硬件处理增加到传感器集成电路增加了芯片区域,如果该相同的处理可以在应用处理器 12 上以软件执行,则其可能是不必要的。

[0144] 对算法的数据的输入而非对用户的输出经常需要运动传感所必需的精确定时,即,定时要求对于用户的应用程序的输出更加放宽。例如,在用户接口应用中,运动传感经

常在 100Hz 被执行,带有精确地定时以捕获全部必要运动。然而,通过应用程序的对用户的输出出可仅在 10Hz,带有不精确的定时。

[0145] 本发明的实施方式可以使用硬件和 / 或软件以获得对于应用程序输出放宽的定时约束的优点,并从而减少应用处理器的处理时间。一些实施方式可以为那些具有快速处理器的系统提供更多使用软件的处理,而其他实施方式可以为带有较慢处理器或其他处理负荷的系统提供更多使用硬件的处理。在任何实施方式中,相同的 API 层 56 可以用于将作为结果的高级信息提供给应用程序,允许到任何系统的最大化可移植性。

[0146] 图 8A 是说明用于提供缓冲以减少应用处理器在接收运动传感器输出中的处理的本发明的第一方法 400 的流程图。方法 400 使用硬件缓冲以允许应用处理器 12 放宽在获取传感器数据中它的定时约束。

[0147] 方法在 402 开始,并且在步骤 404,MPU 20 以较高速率采样运动传感器数据并将传感器数据记录到硬件缓冲器。例如,硬件缓冲器可以在如在图 1 中所示的 MPU 20 的硬件处理块 30 中,或在装置 10 的不同的所连接的硬件中。MPU 以高速率和捕获运动数据用于特定应用通常所需的精确定时来采样传感器数据。

[0148] 在步骤 406,应用处理器 12 读取自从它最后一次读取缓冲器后在硬件缓冲器中积累的全部传感器数据,其中处理器 12 以比用于记录传感器数据的 MPU 低的速率采样硬件缓冲器(和传感器数据)。处理器在硬件缓冲器中一次读取全部这些传感器数据点。

[0149] 在步骤 408,基于取回的多个传感器数据点快速连续地更新运动算法,并且以较低的采样率将算法的最终输出提供给应用程序。这以较低的所需采样率将信息提供给应用程序。处理然后返回到步骤 404。

[0150] 在缓冲器中的传感器数据的积累允许应用处理器 12 的应用程序放宽用于从 MPU 采样数据的定时约束而没有遗漏任何运动信息。例如,MPU 硬件可在 100Hz 的较高速率将传感器数据记录在缓冲器中。在应用处理器 12 上的应用程序可需要以仅 10Hz 的采样率基于运动更新显示屏。使用方法 400,运动算法可以在 10Hz 的速率一次读取存储在 MPU 缓冲器中的全部数据。从而,如果 10 个数据点在缓冲器中,运动算法快速连续地更新 10 次,并且有效地以 10Hz 将通过运动算法输出的显示信息提供给应用程序。这允许仅当应用处理器 12 需要时,应用处理器 12 例如以 10Hz 的速率(显示刷新率)读取缓冲器,这减少处理器 12 的总处理时间。因为运动算法使用接收到的运动数据的全部数据点而平均起来仍将以较高的速率运行,因此减少可能并非大量的。

[0151] 图 8B 是说明用于提供缓冲以减少应用处理器在接收运动传感器输出中的处理的本发明的第二方法 420 的流程图。方法 420 使用软件缓冲以允许应用处理器 12 放宽在获取传感器数据中它的定时约束。

[0152] 方法在 422 开始,并且在步骤 424,中断程序以较高速率采样运动传感器数据。例如,中断程序是在应用处理器 12 上运行的软件程序,并可以在算法层 60 中提供。中断程序以高速率以及捕获适用于应用的运动数据通常所需的正确和精确的定时来采样来自 MPU 20 的传感器数据。在步骤 426,中断程序存储在步骤 424 在软件缓冲器中读取的数据,没有处理数据。例如,软件缓冲器是应用处理器可访问的存储器,并可以在应用处理器 12(寄存器、RAM 等)中、在装置 10 的存储器 14 中、或在其他可用的存储器中被提供。

[0153] 在步骤 428,中断程序中断处理器 12 以指示积累了足够的传感器数据以被读取。

在步骤 430, 应用处理器 12(举例来说, 运动算法) 读取自从它最后一次读取缓冲器后在软件缓冲器中积累的全部传感器数据。从而, 处理器 12 以比中断程序用于采样和存储传感器数据低的速率读取传感器数据。在步骤 432, 基于取回的多个传感器数据点, 快速连续地更新运动算法, 并且将算法的最终输出以较低的采样率提供给应用程序。这将信息以较低的所需采样率提供给应用程序。过程然后返回到步骤 424。

[0154] 如同用图 8A 的硬件缓冲的实施方式, 在缓冲器中的传感器数据的积累允许应用处理器 12 放宽关于应用程序输出的它的定时约束, 而没有遗漏任何运动信息。例如, 中断程序可在 100Hz 的较高速率在缓冲器中采样和存储传感器数据, 但是应用程序 12 可以在放宽的速率一次读取存储在软件缓冲器中的全部数据。从而, 以 10Hz 的采样率, 应用处理器可以取回在缓冲器中的最近的 10 个数据点, 并快速连续地处理运动算法 10 次, 允许显示应用程序以 10Hz 从算法接收适当的信息, 并以 10Hz 提供它的输出。

[0155] 图 8C 是说明用于提供缓冲以减少应用处理器在接收运动传感器输出中的处理的本发明的第三方法 450 的流程图。方法 450 使用硬件中断, 以允许应用处理器 12 减少在获得传感器数据中它的定时约束。

[0156] 方法在 452 开始, 并且在步骤 454, MPU 20 接收传感器数据且将传感器数据存储在缓冲器中。以高速率以及捕获运动传感器数据通常所需的正确和精确的定时来执行该采样。传感器数据可以被存储在硬件缓冲器或软件缓冲器中。

[0157] 在步骤 456, MPU 扫描缓冲的传感器数据。在步骤 458, MPU 检查在所扫描的传感器数据中是否有任何“感兴趣的”特性已被检测。这样的特性是在数据中的模式或特征, 其可与特定算法或应用有关, 以便它们应被上报到应用处理器 12。例如, 特性可以包括已经被超过的预先确定的阈值, 或指示特定类型的运动的在数据中的模式。例如, 当由用户作出手势时, 可在至少一个运动传感器上超过阈值。或者特定阈值的值可已经被超过, 指示在装置 10 的照相机镜头部分上抖动的级别。在步骤 458 如果没有感兴趣的特性被 MPU 检测到, 那么过程返回到步骤 454。

[0158] 如果一个或多个感兴趣的特性被检测到, 那么在步骤 460, MPU 将中断发送到应用处理器。该中断指示到应用处理器感兴趣的特性已经被检测到并应被应用处理器检查。从而, 在步骤 462, 应用处理器 12 取回在包括所检测的感兴趣的特性的缓冲器中存储的数据, 举例来说, MPU 可以在不同的缓冲器或缓冲器的其他区域中缓冲感兴趣的特性数据, 处理器 12 可以从其中读取, 或者 MPU 可以在主要缓冲器中指定位置以从该指定的位置读取。应用处理器 12 然后分析取回的作为适合于应用的数据, 举例来说, 在算法层 60 中用算法处理数据并以已减少的采样率通过 API 56 将数据提供给应用程序。过程然后返回到步骤 454。

[0159] 使用 MPU 硬件以检查所感测的数据中的特定的特征, 可以减少应用处理器的处理系统开销。即使 MPU 以高速率将中断发送到应用处理器, 如每秒 5 次检测感兴趣的特性, 它仍将必须运行的算法处理和应用程序处理的次数从 100Hz 减少到 5Hz。使用该实施方式, 对于在应用处理器 12 上的全部软件减少了定时约束, 并且也因而减少了处理器 12 的处理时间。

[0160] 图 8D 是说明用于提供缓冲以减少应用处理器在接收运动传感器输出中的处理的本发明的第四方法 470 的流程图。方法 470 使用硬件预处理的更加改进的方法, 以允许应用处理器 12 减少在获得传感器数据中它的定时约束。

[0161] 方法在 472 开始,并且在步骤 474,MPU 20 接收传感器数据且将传感器数据存储在缓冲器中,如 MPU 的硬件缓冲器。以高速率以及捕获运动传感器数据通常所需的精确的定时来采样并存储数据。在步骤 476,MPU 预处理传感器数据以将数据减少到对于一个或多个应用的有关特性集。MPU 可以包括与在应用处理器软件的运动算法层 60 中存在的运动算法类似的运动算法,或者可以包括在应用处理器软件的运动算法层 60 中存在的运动算法的子集。例如,在一些实施方式中,MPU 可以通过处理传感器数据检测特定的手势。该处理能力可以确定哪个特征与在应用处理器 12 上运行的应用有关,和/或为应用程序提供高级信息。例如,有关特征可以是附加的惯性数据导数和积分中计算的参数。

[0162] 在步骤 478,以比用于捕获运动传感器数据的采样率低的采样率,将有关特征集提供给在应用处理器 12 上的应用程序。从而,例如,用于捕获传感器数据的 100Hz 采样率可以减少到仅将有关信息提供给应用处理器的 5Hz 采样率。然后,可以将有关特征以减少的采样率提供给应用程序(或提供给为了进一步处理的适当的运动算法,其以减少的采样率将信息提供给应用程序)。类似于图 8C 的实施方式,使用该实施方式,对于在应用处理器 12 上的全部软件减少了定时约束,并且从而也减少了处理器 12 的处理时间。

[0163] 尽管根据所示的实施方式已经描述了本发明,但是本领域普通技术人员将容易地看出,应存在实施方式的变化并且那些变化在本发明的主旨和范围内。相应地,可由本领域普通技术人员作出许多修改,而不脱离所附权利要求的主旨和范围。

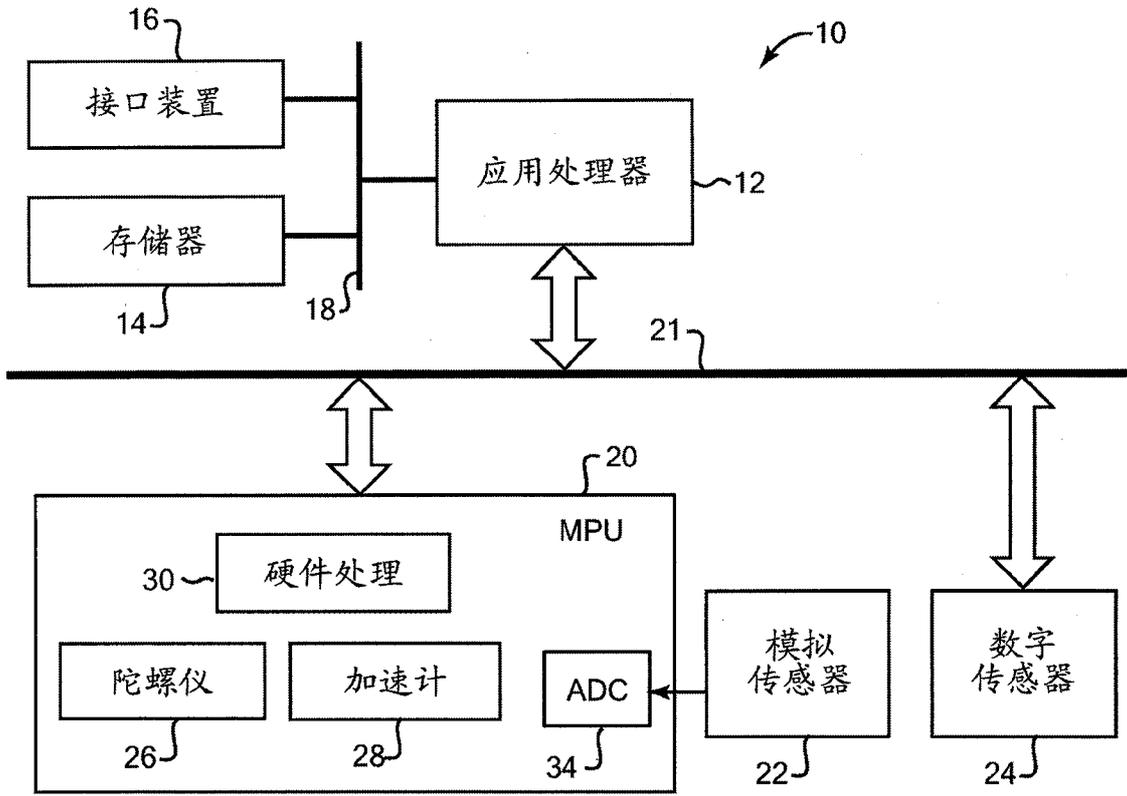


图 1

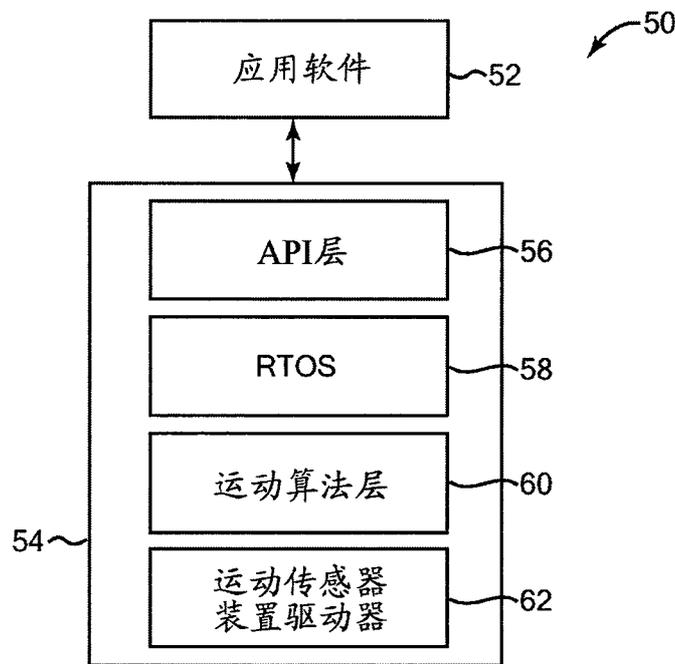


图 3

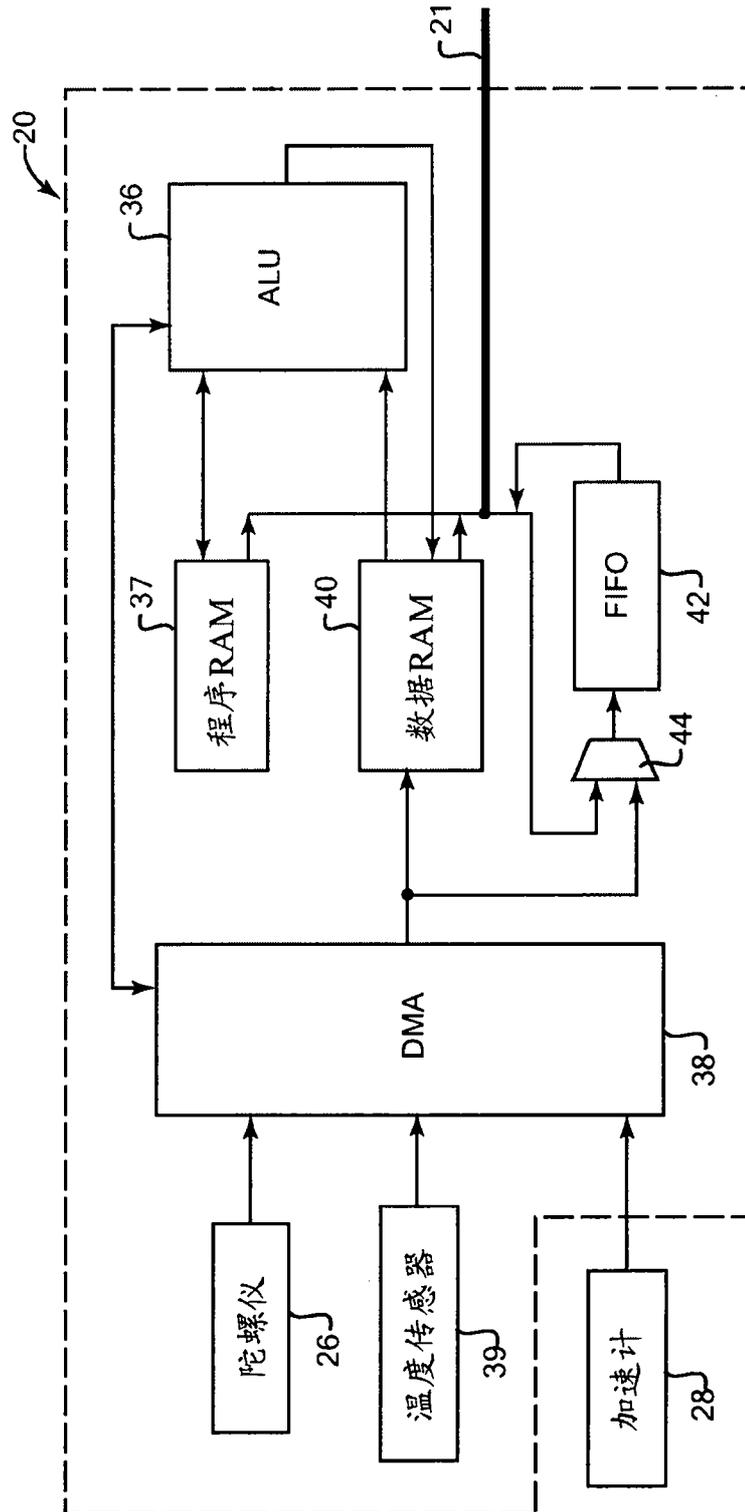


图 2

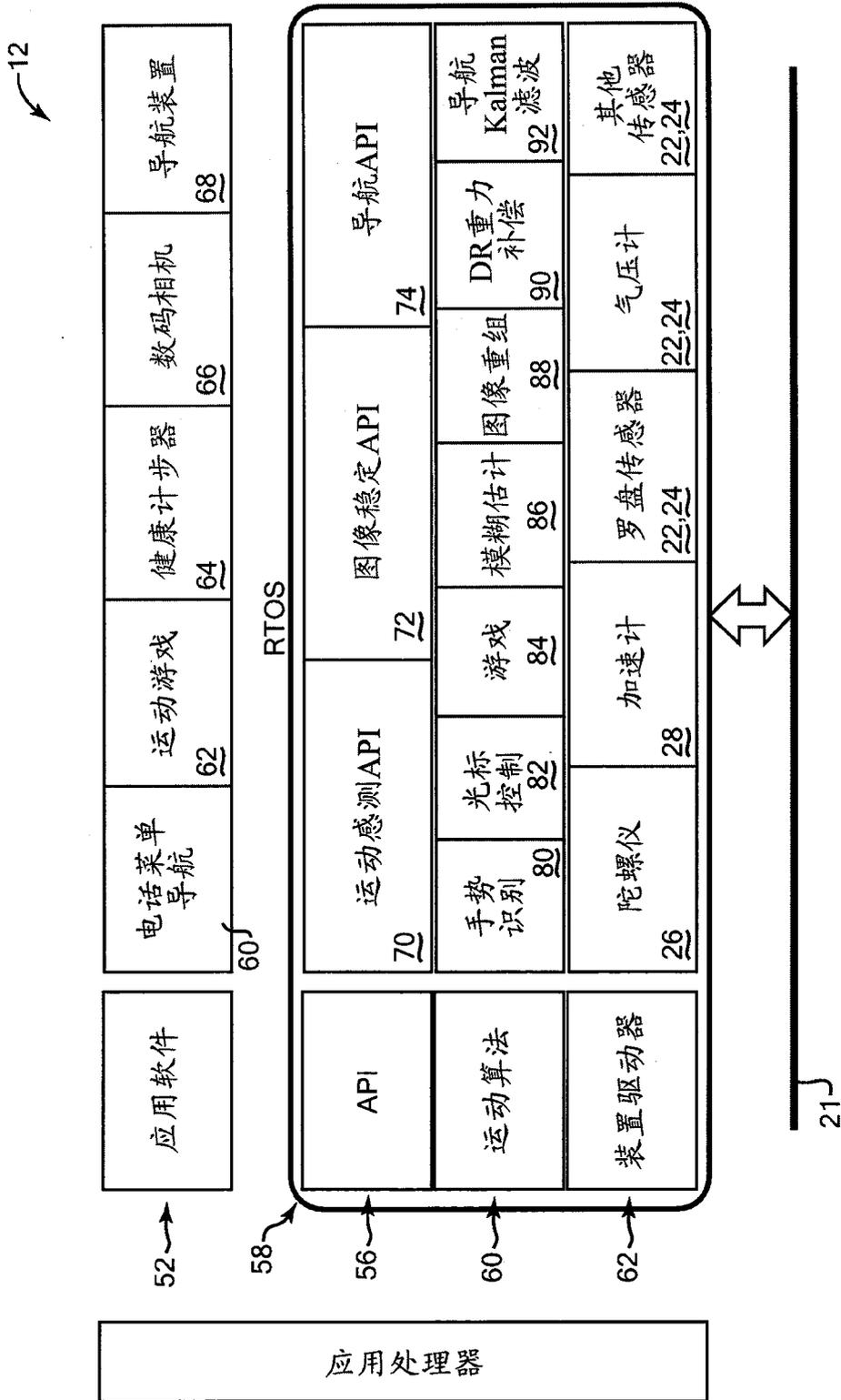


图 4

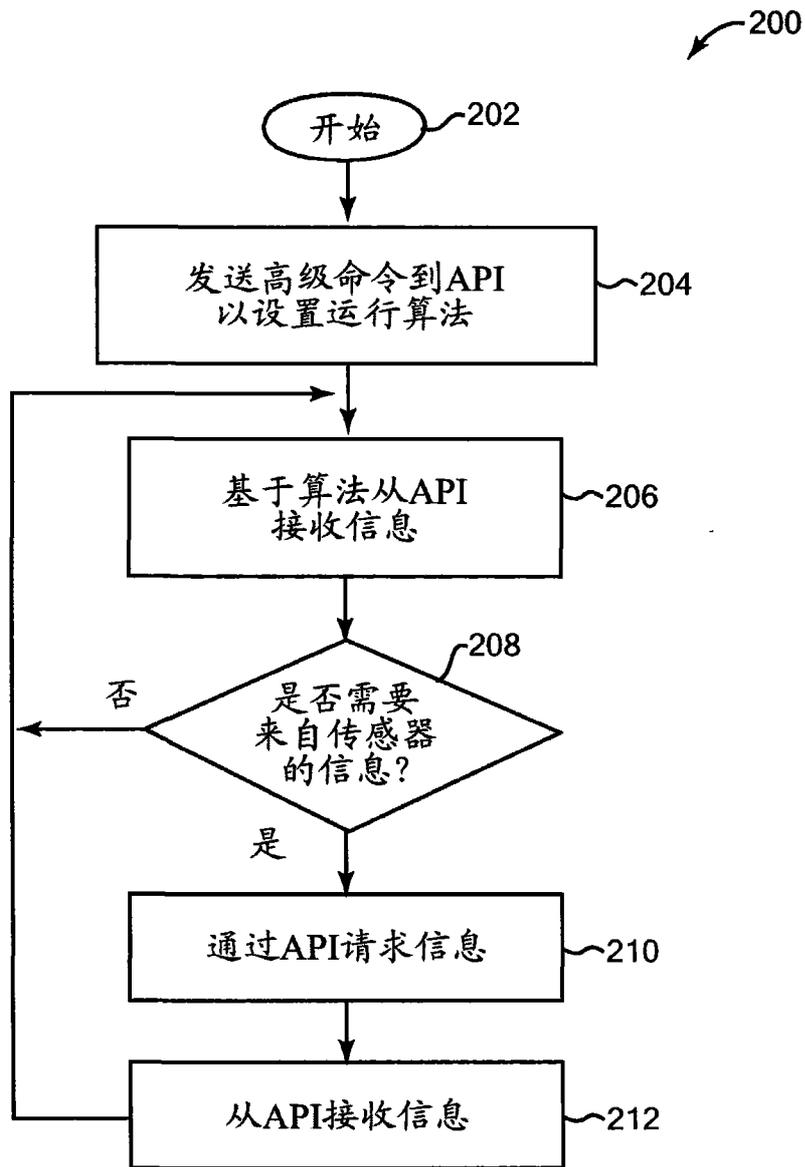


图 5

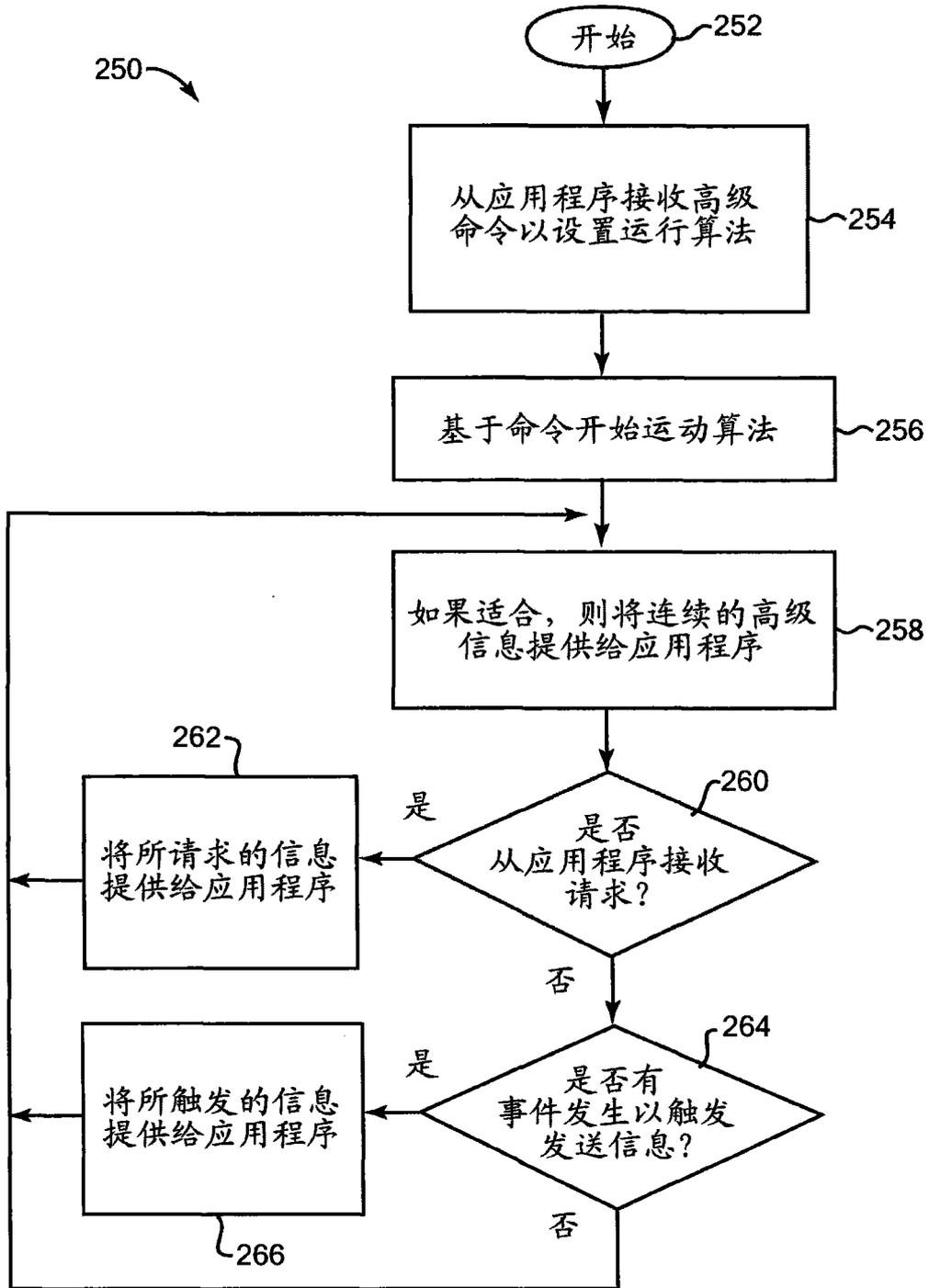


图 6

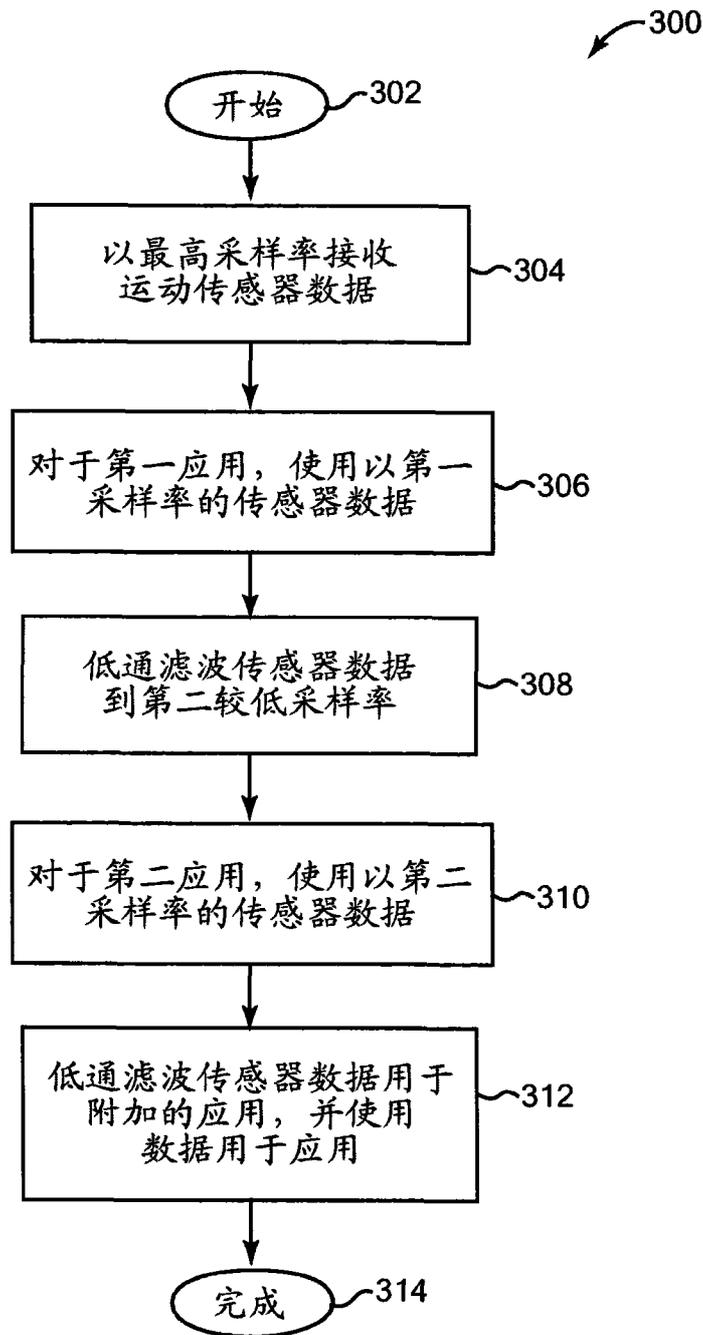


图 7

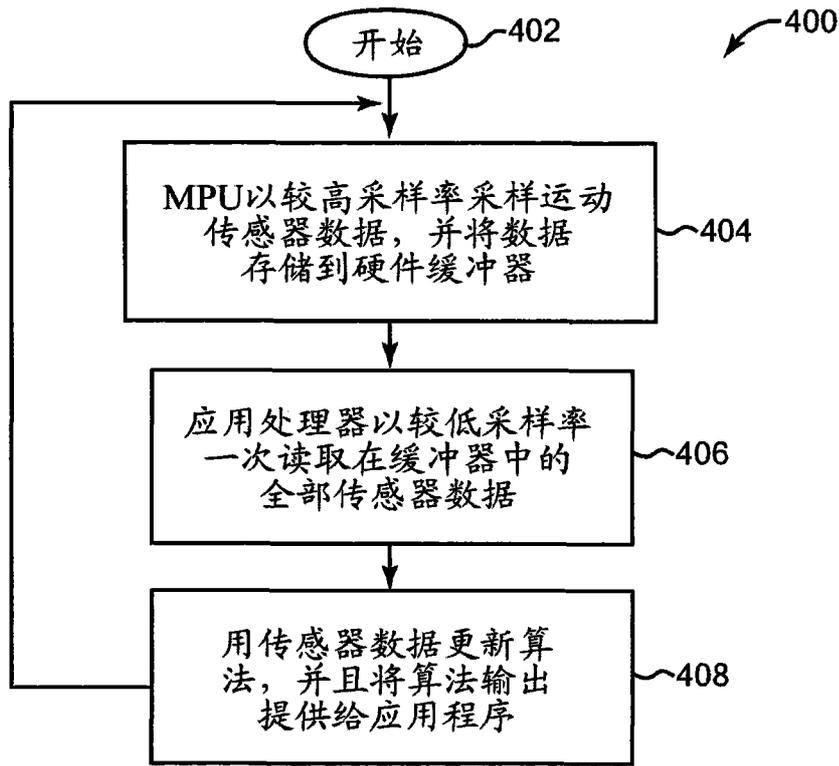


图 8A

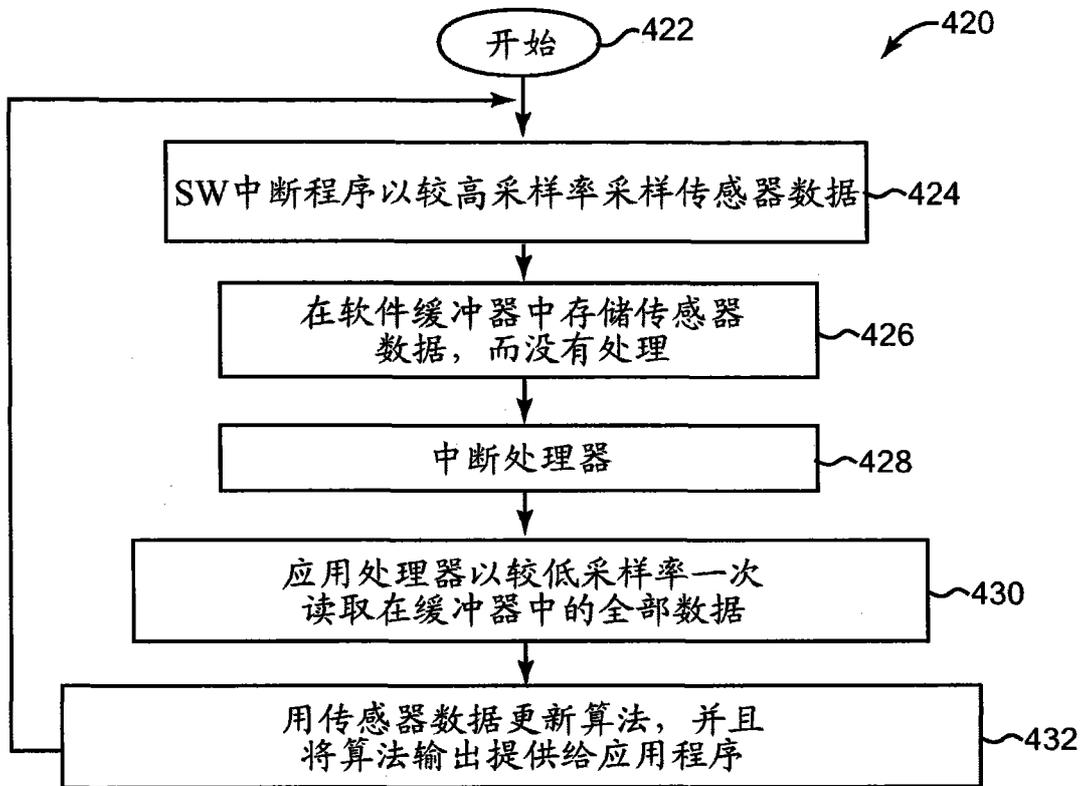


图 8B

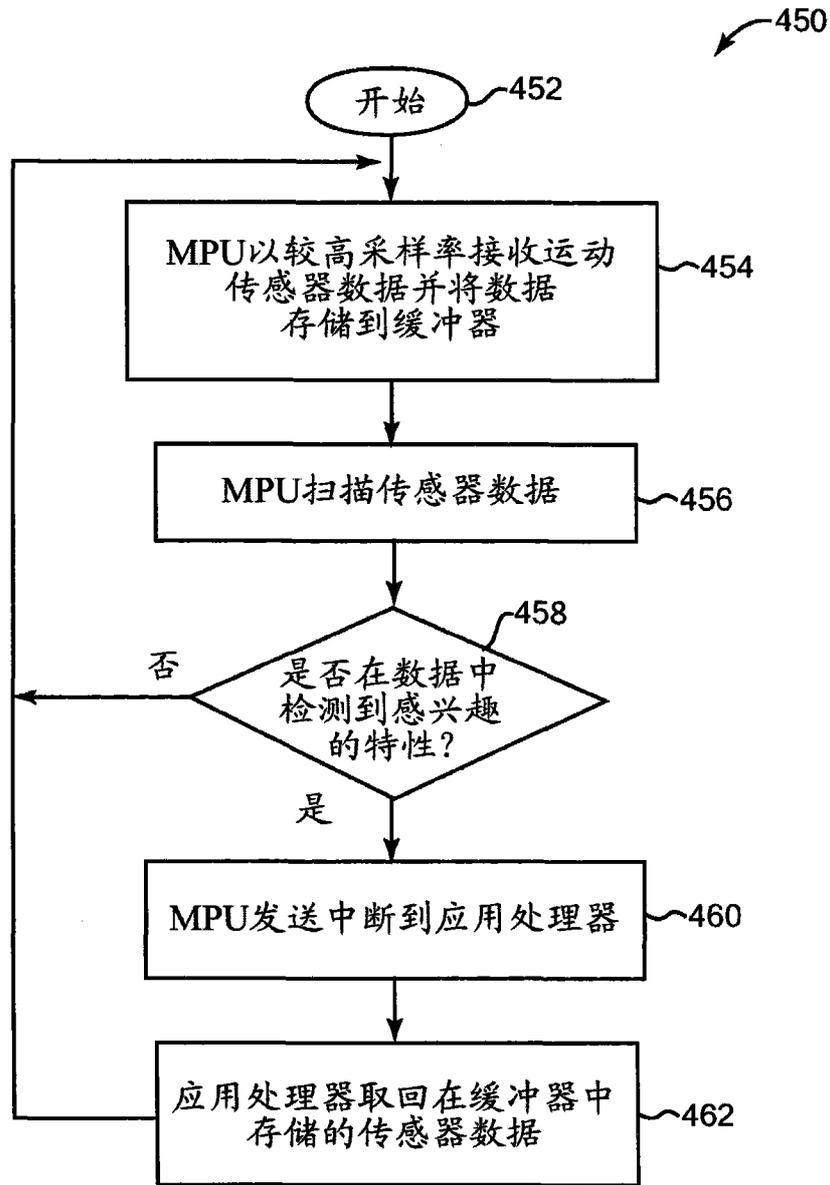


图 8C

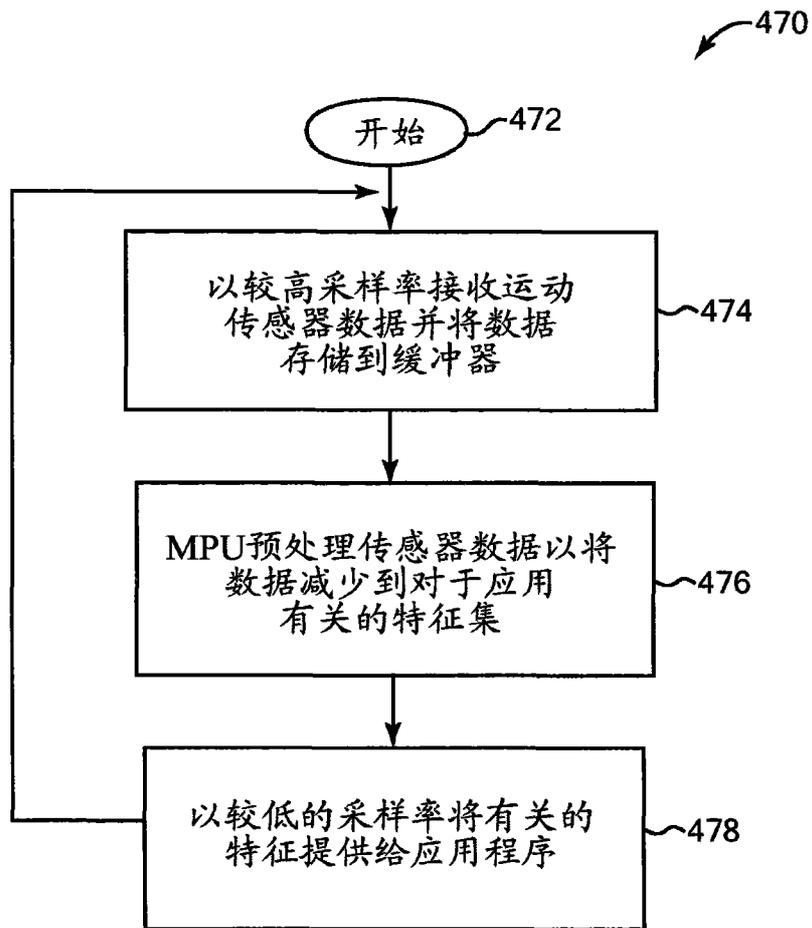


图 8D