

(12) 发明专利申请

(10) 申请公布号 CN 102693007 A

(43) 申请公布日 2012.09.26

(21) 申请号 201210052880.8

(22) 申请日 2012.03.02

(30) 优先权数据

13/040,487 2011.03.04 US

(71) 申请人 微软公司

地址 美国华盛顿州

(72) 发明人 S·诺沃辛 P·科利

J·D·J·肖顿

(74) 专利代理机构 上海专利商标事务所有限公司 31100

代理人 陈斌

(51) Int. Cl.

G06F 3/01 (2006.01)

G06K 9/00 (2006.01)

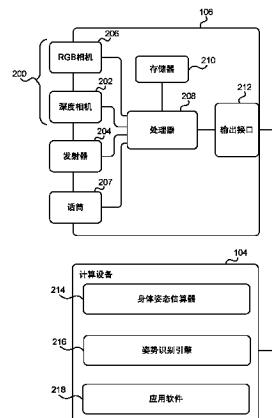
权利要求书 2 页 说明书 16 页 附图 11 页

(54) 发明名称

姿势检测和识别

(57) 摘要

本发明涉及姿势检测和识别。描述了一种姿势检测和识别技术。在一个示例中，接收与执行姿势的用户的运动有关的数据项的序列。对照预先学习的门限值测试从该序列中选出的数据项集合，以确定该序列表示某个姿势的概率。如果该概率大于预定值，则检测到该姿势，并执行动作。在示例中，这些测试是由经训练的决策树分类器执行的。在另一示例中，可将数据项的序列与预先学习的模板进行比较，并确定它们之间的相似度。如果模板的相似度超过门限，则更新与该模板相关联的姿势的未来时间相关联的可能性值。然后，当达到该未来时间时，如果该可能性值大于预定值，则检测到该姿势。



1. 一种计算机实现的姿势检测方法,包括:

在处理器(1102)处,接收与执行姿势的用户(102)的运动有关的数据项的序列(600);

将来自所述序列(600)的多个所选择的数据项对照预定义门限值进行测试以确定所述序列(600)表示姿势的概率;以及

如果所述概率大于预定值则检测到所述姿势。

2. 如权利要求1所述的方法,其特征在于,还包括响应于检测到所述姿势执行命令的步骤。

3. 如权利要求1或2所述的方法,其特征在于,所述序列(600)中的数据项表示以下各项中的至少一个:

- i) 所述用户(102)的至少两个身体部位之间的角度;
- ii) 所述用户(102)的至少两个身体部位之间的角度的变化率;
- iii) 所述用户(102)的至少一个身体部位的运动速度;
- iv) 从所述用户的深度图像推导出的特征;
- v) 从所述用户的RGB图像推导出的特征;
- vi) 来自所述用户的语音信号;
- vii) 移动设备(300)的惯性;
- viii) 移动设备(300)的速度;
- ix) 移动设备(300)的定向;以及
- x) 移动设备(300)的位置。

4. 如权利要求1、2或3所述的方法,其特征在于,测试的步骤包括将所述序列应用到经训练的决策树(500)。

5. 如权利要求4所述的方法,其特征在于,将所述序列应用到经训练的决策树(500)的步骤包括使所述序列通过所述树中的多个节点,直到到达所述树中的叶节点(510),且其中所述序列表示姿势的概率取决于所述决策树中到达的所述叶节点(510)而确定。

6. 如权利要求5所述的方法,其特征在于,所述决策树(500)的每个节点与索引值以及所述预定义门限值中的至少一个相关联,并且将所述序列(600)应用到经训练的决策树(500)的步骤包括在每个节点处,将所述序列(600)中位于所述索引值处的数据项与所述预定义门限值中的所述至少一个相比较以确定要将所述序列发送到的后续节点。

7. 如权利要求5或6所述的方法,其特征在于,测试的步骤还包括将所述序列(600)应用到至少一个进一步的、有区分的经训练的决策树(502),以及通过对来自所述经训练的决策树中的每一个的概率求平均来确定所述序列表示所述姿势的概率。

8. 如权利要求4到7中的任一项所述的方法,其特征在于,还包括在接收所述序列(600)之前训练所述决策树(500)的步骤,其中训练所述决策树的步骤包括:

选择所述决策树的节点;

选择至少一个经注释的训练序列;

生成多个随机索引值和随机门限值;

将来自位于所述随机索引值中的每一个处的所述经注释的序列的数据项与所述随机门限值中的每一个相比较以获得多个结果;

依据所述多个结果,为所述节点选择选中的索引值和至少一个选中的门限值;以及将与所述节点相关联的所述选中的索引值和所述至少一个选中的门限值存储在存储设备处。

9. 如权利要求 8 所述的方法,其特征在于,选择选中的索引值和至少一个选中的门限值的步骤包括确定所述多个结果中的每一个的信息增益,并且选择与具有最大信息增益的结果相关联的选中的索引值和至少一个选中的门限值。

10. 一种姿势识别系统,包括:

被安排为接收用户(102)的深度图像的序列的用户接口(1104);

被安排为存储包括多个有区分的经训练的决策树的随机决策树林的存储器(1110);以及

处理器(1102),所述处理器被安排为:从所述深度图像生成与所述用户(102)的运动有关的数据项的序列(600);将所述数据项的序列(600)应用到所述经训练的决策树中的每一个以获得所述序列表示多个预定义姿势中的一个的多个概率;聚集来自所述经训练的决策树中的每一个的概率;以及如果所述姿势的聚集的概率大于预定值,则执行与所检测到的姿势相关联的命令。

姿势检测和识别

技术领域

[0001] 本发明涉及姿势检测和识别。

背景技术

[0002] 基于姿势的交互技术提供了用户与计算设备交互的直观而自然的方式。许多设备和系统为用户提供使用简单的、易检测的姿势（诸如在触敏屏上的捏合或滑动）来进行交互的能力。这种基于姿势的交互能够极大地提升用户体验。

[0003] 然而，为了支持更加丰富或更加不同的姿势，精确检测和识别姿势的计算复杂度能够显著增加。例如，随着姿势的数量和 / 或复杂度的增加，检测姿势所涉及的计算复杂度可能导致所执行的姿势和计算设备进行动作之间的可注意到的滞后。在一些应用（诸如游戏）的情况下，这种滞后可能负面影响用户体验。

[0004] 此外，随着基于姿势的用户交互的使用变得越来越常见，越来越多样化的用户用这种方式进行交互。例如，执行姿势的用户来自更广的年龄范围，并具有不同的经验水平。这意味着，不同用户可能非常不同地执行相同的姿势，从而对姿势识别技术要产生一致而精确的检测带来挑战。

[0005] 此外，对自然用户界面的使用变得越来越广泛，其中用户使用例如基于相机的输入或用于跟踪用户身体部位的移动的设备来更直观地与计算设备交互。这种自然用户界面使得姿势的输入比例如在触敏屏上执行的姿势更加“自由”（即，约束更少）。这带来了姿势的更多的自由度和变化度，从而进一步增加了姿势识别技术的需要。

[0006] 以下描述的各实施例不限于解决已知姿势识别技术的缺点中的任一个或全部的实现。

发明内容

[0007] 下面呈现了本发明的简要概述，以便向读者提供基本理解。本发明内容不是本发明的详尽概述，并且不标识本发明的关键 / 重要元素，也不描述本发明的范围。其唯一的目的以简化形式呈现此处所公开的精选概念，作为稍后呈现的更详细的描述的序言。

[0008] 描述了一种姿势检测和识别技术。在一个示例中，接收与执行姿势的用户的运动有关的数据项的序列。对照预先学习的门限值测试从该序列中选出的数据项集合，以确定该序列表示某个姿势的概率。如果该概率大于预定值，则检测到该姿势，并执行动作。在示例中，这些测试是由经训练的决策树分类器执行的。在另一示例中，可将数据项的序列与预先学习的模板进行比较，并确定它们之间的相似度。如果模板的相似度超过门限，则更新与该模板相关联的姿势的未来时间相关联的可能性值。然后，当达到该未来时间时，如果该可能性值大于预定义值，则检测到该姿势。

[0009] 通过结合附图参考以下具体实施方式，可更易于领会并更好地理解许多附带特征。

附图说明

- [0010] 根据附图阅读以下具体实施方式,将更好地理解本发明,在附图中:
- [0011] 图 1 示出了用于控制计算机游戏的示例性的基于相机的控制系统;
- [0012] 图 2 示出了图像捕捉设备的示意图;
- [0013] 图 3 示出了移动设备的示意图;
- [0014] 图 4 示出了用于训练决策树林以识别姿势的过程的流程图;
- [0015] 图 5 示出了随机决策树林的示例部分;
- [0016] 图 6 示出了对数据序列的示例门限测试;
- [0017] 图 7 示出了使用包括图 6 的测试的决策树的示例姿势识别计算;
- [0018] 图 8 示出了用于使用经训练的决策树林来识别姿势的过程的流程图;
- [0019] 图 9 示出了用于使用经训练的逻辑斯蒂 (logistic) 模型来识别姿势的过程的流程图;
- [0020] 图 10 示出了使用经训练的逻辑斯蒂模型的示例姿势识别计算;以及
- [0021] 图 11 示出可在其中实现姿势识别技术的各实施例的示例性的基于计算的设备。
- [0022] 在各个附图中使用相同的附图标记来指代相同的部件。

具体实施方式

[0023] 下面结合附图提供的具体实施方式旨在作为本发明示例的描述,并不旨在表示可以构建或使用本发明示例的唯一形式。本描述阐述了本发明示例的功能,以及用于构建和操作本发明示例的步骤的序列。然而,可以通过不同的示例来实现相同或等效功能和序列。

[0024] 虽然在本文中将本发明的示例描述并示出为在游戏系统或移动通信设备中实现,但是它们只是作为示例而非限制来提供的。本领域的技术人员将理解,本示例适于在各种不同类型的计算系统中应用。

[0025] 本文描述了一种姿势识别技术,该技术允许用在计算上高效的方式、以低等待时间检测并识别用户姿势。这些姿势识别技术可被应用于自然用户接口。例如,图 1 和图 2 示出了其中能够用由相机捕捉的用户姿势来控制计算设备(诸如游戏设备)的示例,而图 3 示出了其中能用运动和 / 或定向传感器检测的用户姿势来控制手持设备的示例。

[0026] 首先参考图 1,图 1 示出了用于控制计算机游戏的示例性的基于相机的控制系统 100。图 1 示出了在该所示示例中用户 102 正在玩拳击游戏。在一些示例中,基于相机的控制系统 100 可被用于确定身体姿态、绑定、识别、分析、跟踪、关联至人类目标、提供反馈、和 / 或适应于诸如用户 102 这样的人类目标的各方面,等等。

[0027] 基于相机的控制系统 100 包括计算设备 104。计算设备 104 可以是通用计算机、游戏系统或控制台、或专用图像处理设备。计算设备 104 可包括硬件组件和 / 或软件组件,以使计算设备 104 可用于执行诸如游戏应用和 / 或非游戏应用等应用。后文参考图 11 讨论了计算设备 104 的结构。

[0028] 基于相机的控制系统 100 进一步包括捕捉设备 106。捕捉设备 106 可以是例如图像传感器或检测器,该图像传感器或检测器可用于在视觉上监视一个或多个用户(诸如用户 102),以使得可以捕捉、分析、处理、及跟踪一个或多个用户所执行的姿势以执行游戏或应用内的一个或多个控制或动作,如将在下面更详细地描述的。

[0029] 基于相机的控制系统 100 可进一步包括连接到计算设备 104 的显示设备 108。该计算设备可以是可向用户 102 提供游戏或应用视觉（和可任选的音频）的电视机、监视器、高清电视机（HDTV）等。

[0030] 在操作中，可以使用捕捉设备 106 跟踪用户 102 以使得用户 102 的位置、移动和大小可以被计算设备 104（和 / 或捕捉设备 106）解释为可用于影响由计算设备 104 执行的应用的控制。结果，用户 102 能够移动他或她的身体以控制所执行的游戏或应用。

[0031] 在图 1 的说明性示例中，计算设备 104 上执行的应用是用户 102 正在玩的拳击游戏。在本示例中，计算设备 104 控制显示设备 108 来向用户 102 提供拳击对手的视觉表示。计算设备 104 还控制显示设备 108 来提供用户 102 可用他或她的移动来控制的用户化身的视觉表示。例如，计算设备 104 可包括身体姿态估算器，该估算器被安排为识别并跟踪用户的不同身体部位，并将这些部位映射到化身上。用这种方式，化身复制用户 102 的移动，以使得如果用户 102 例如在物理空间中挥拳，则这会致使该用户化身在游戏空间中挥拳。

[0032] 然而，仅在游戏空间中复制用户移动限制了用户和游戏之间的交互的类型和复杂度。例如，许多游戏内控制是瞬间的动作或命令，在传统游戏系统中，这些动作或命令可以使用按钮按下来触发。这些动作或命令的示例包括诸如挥拳、射击、换武器、投掷、踢、跳、和 / 或下蹲。通过识别出用户正在执行这些动作之一并且触发相应游戏内动作，而不是仅复制用户的移动，在许多应用中可以更可靠地控制这些动作或命令。

[0033] 计算设备（诸如游戏系统）的控制还包括控制化身以外的许多动作的输入。例如，命令被用于控制菜单的选择、后退 / 退出、打开或关闭系统、暂停、保存游戏、与朋友通信等。此外，控制被用于与游戏之外的应用进行交互，例如以输入文本、选择图标或菜单项、控制媒体回放、浏览网站或操作操作系统或应用的任何其他可控方面。

[0034] 这些命令和动作不能仅通过反映用户的移动来控制。而是，更高层次的处理分析这些移动以检测用户是否正在执行与这些命令或动作之一相对应的姿势。如果姿势被识别，则可执行相应的动作或命令。然而，在姿势检测的速度和准确性方面，对姿势识别系统施加了要求。如果存在滞后或等待时间，则这影响计算设备的可用性。类似地，如果姿势被不准确地检测，则这也会影响用户体验带来负面影响。下面描述一种允许对姿势进行快速、准确的检测和识别的姿势识别技术。

[0035] 现在参考图 2，图 2 示出了可在图 1 的基于相机的控制系统 100 中使用的捕捉设备 106 的示意图。在图 2 的示例中，捕捉设备 106 被配置为捕捉具有深度信息的视频图像。这种捕捉设备可被称为深度相机。深度信息可以是包括深度值的深度图像的形式，即深度值是与深度图像的每个图像元素相关联的值，该值和该深度相机和位于该图像元素处的项目或物体之间的距离有关。注意，术语“图像元素”用于指代像素、像素组、体素（voxel）、或图像的其他更高层次的分量。

[0036] 深度信息可以使用任何合适的技术获得，包括例如飞行时间、结构化光、立体图像等。在一些示例中，捕捉设备 106 可将深度信息组织为“Z 层”，或可垂直于从深度相机沿其视线延伸的 Z 轴的层。

[0037] 如图 2 所示，捕捉设备 106 包括至少一个成像传感器 200。在图 2 所示的示例中，成像传感器 200 包括被安排为捕捉场景的深度图像的深度相机 202。所捕捉的深度图像可包括所捕捉的场景的两维（2-D）区域，其中该 2-D 区域中的每个图像元素表示一深度值，诸

如所捕捉的场景中的物体离深度相机 202 的长度或距离。

[0038] 该捕捉设备还可包括被安排为以可由深度相机 202 查明深度信息的方式来照亮该场景的发射器 204。例如,在深度相机 202 是红外 (IR) 飞行时间相机的情况下,发射器 204 将 IR 光发射到该场景上,并且深度相机 202 被安排为检测从该场景中的一个或多个目标和物体的表面反向散射的光。在一些示例中,可以从发射器 204 发出脉冲红外光,使得外出光脉冲与对应的传入光脉冲之间的时间可由深度相机来检测和测量,并被用来确定从捕捉设备 106 到该场景中的目标或物体上的位置的物理距离。另外,在一些示例中,可将来自发射器 204 的出射光波的相位与深度相机 202 处的入射光波的相位进行比较来确定相移。该相移然后可以用于确定从捕获设备 106 到目标或物体上的一位置的物理距离。在未来示例中,可使用飞行时间分析,通过经由包括例如快门式光脉冲成像的各种技术来分析反射光束随时间的强度以间接地确定从捕捉设备 106 到目标或物体上的一位置的物理距离。

[0039] 在另一示例中,捕捉设备 106 可使用结构化光来捕捉深度信息。在这种技术中,可使用发射器 204 将图案化光(例如,显示为诸如斑点、网格、或条形图案等已知图案的光,它也可随时间变化)投影到场景上的一个或多个目标或物体的表面上以后,图案变形。这种图案变形可由深度相机 202 捕捉然后被分析以确定从捕捉设备 106 到该场景中的目标或物体上的一位置的物理距离。

[0040] 在另一示例中,深度相机 202 可以是从不同角度查看场景的两个或更多个物理上分开的相机的形式,以便获得能被解析以生成深度信息的视觉立体数据。在此情况下,发射器 204 可被用来照亮场景,或可被省略。

[0041] 在一些示例中,除了深度相机 202 之外或作为对深度相机 202 替换,捕捉设备 106 可包括被称为 RGB 相机 206 的常规视频相机。RGB 相机 206 被安排为在可见光频率捕捉场景的图像序列,并能因此提供可用于扩充深度图像的图像。在一些示例中,可以使用 RGB 相机 206 而不是深度相机 202。捕捉设备 106 还可以可任选地包括话筒 207 或话筒阵列(其可以是有向的和 / 或可转向的),该话筒或话筒阵列被安排为捕捉声音信息(诸如来自该用户的语音输入)并且可以用于语音识别。

[0042] 图 2 中示出的捕捉设备 106 进一步包括至少一个处理器 208,处理器 208 与图像传感器 200(即,图 2 的示例中的深度相机 202 和 RGB 相机 206)、发射器 204、和话筒 207 通信。处理器 208 可以是通用微处理器、或专用信号 / 图像处理器。处理器 208 被安排为执行指令以控制成像传感器 200、发射器 204 和话筒 207 以捕捉深度图像、RGB 图像和 / 或语音信号。处理器 208 也可以可任选地被安排为对这些图像和信号执行处理,如后文更详细地概述的。

[0043] 图 2 中示出的捕捉设备 106 进一步包括存储器 210,存储器 210 被安排为存储由处理器 208 执行的指令、深度相机 202 或 RGB 相机 206 捕捉到的图像或图像的帧、或任何其他合适的信息、图像等。在一些示例中,存储器 210 可包括随机存取存储器 (RAM)、只读存储器 (ROM)、高速缓存、闪存、硬盘、或任何其他合适的存储组件。存储器 210 可以是与处理器 208 通信的分开的组件,或者可被集成到处理器 208 中。

[0044] 捕捉设备 106 还包括与处理器 208 通信并被安排为经由通信链路向计算设备 104 提供数据的输出接口 212。该通信链路可以是例如有线连接(诸如 USB、火线、以太网或类似连接)和 / 或无线连接(诸如 WiFi、蓝牙或类似连接)。在其他示例中,输出接口 212 可

与一个或多个通信网络（诸如因特网）界接并经由这些网络向计算设备 104 提供数据。

[0045] 计算设备 104 执行与基于相机的姿势识别有关的多个功能，诸如可任选的身体姿态估算器 214 和姿势识别引擎 216，如下面更详细地描述的。身体姿态估算器 214 被安排为使用计算机视觉技术来检测和跟踪该用户的不同身体部位。身体姿态估算器 214 可向姿势识别引擎提供与用户的身体姿态有关的数据的时间系列形式的输出。这可以是该用户的完全跟踪的骨架模型的形式，或者是该用户的可见身体部位的更粗略标识。例如，这些时间系列序列可包括与用户的至少两个身体部位之间的随时间变化的角度、用户的至少两个身体部位之间的角度的变化率、用户的至少一个身体部位的运动速度、或其组合有关的数据。不同类型的数据（某些身体部位之间的角度、速度等）被称为“特征”。在其他示例中，身体姿态估算器 214 可从随时间改变的用户姿态推导出其他用户序列（即，其他特征）。在进一步示例中，姿势识别引擎 216 可利用除身体姿态估算器以外的不同来源推导输入（即，特征）。还可在计算设备 104 上执行应用软件 218 并使用所述姿势来控制该应用软件。

[0046] 现在参考图 3，图 3 示出了可由姿势控制的示例手持或移动设备 300。在一个示例中，图 3 的移动设备可以是移动电话或其他移动计算或通信设备。与这种移动设备的交互涉及命令的使用，诸如，例如，导航到实体的联系人细节、启动应用、呼叫某人、将设备置于不同的工作模式（静音、振动、户外等）、回答呼叫和许多其他命令。本文描述的姿势识别技术允许用户通过基于运动的姿势做出这些命令。换言之，用户能够通过以某种方式移动移动设备 300 来控制该移动设备。

[0047] 移动设备 300 包括提供与移动设备 300 的运动、定向和 / 或位置有关的信息的一个或多个传感器。在图 3 的示例中，移动设备 300 包括加速度计 302、陀螺仪 304、惯性测量单元 (IMU) 306、和罗盘 308，其中加速度计 302 测量该设备在一个或多个轴上的适当的加速度，陀螺仪 304 能确定该移动设备的定向，IMU 306 能测量加速度和定向两者，罗盘 308 能测量该移动设备的方向。在其他示例中，移动设备 300 能包括这些传感器中的一个或多个的任何组合。

[0048] 所述传感器向计算设备 104 提供与该移动设备随时间的运动或定向有关的数据项的序列形式的信息。注意，计算设备 104 可被集成到移动设备 300 中，或在其他示例中，可位于远程位置处。计算设备 104 执行姿势识别引擎 216，引擎 216 被安排为解释关于该移动设备的运动和 / 或定向（即，特征）的信息并识别用户做出的姿势。来自姿势识别引擎 216 的命令控制在计算设备 104 上执行的应用软件 218，诸如上面提到的那些。移动设备 300 还可以包括显示设备 310，诸如用于向用户显示信息的屏幕，而且还包括一个或多个输入设备 312，诸如触摸传感器或按钮。

[0049] 下面描述了用于在诸如上面描述的那些场景中检测和识别可被应用给自然用户界面的姿势的两种技术。注意，除了上面描述的基于相机的和基于运动的示例外，这些姿势识别技术还可许多其他场景中应用。参考图 4 到图 8 描述的第一种技术是基于机器学习分类器的使用的。参考图 9 和图 10 描述的第二种技术是基于经训练的逻辑斯蒂模型的。

[0050] 如同提到过的，本文描述的第一种姿势识别技术利用机器学习分类器来对姿势进行分类并相应地对其动作。本文使用的机器学习分类器是随机决策树林。然而，在其他示例中，还可以使用另选的分类器（诸如支持向量机、增强 (boosting)）。在进一步示例中，可以使用单个经训练的决策树（这等同于下面的解释中的仅具有一棵树的树林），而不是使

用决策树林。在下面的描述中,首先参考图 4 到图 6 讨论了训练用于机器学习算法的决策树林的过程,其次参考图 7 和图 8 讨论了使用经训练的决策树来使用经训练的算法来对姿势进行分类(识别)的过程。

[0051] 该决策树是使用一组经注释的训练序列来训练的。经注释的训练序列包括与在姿势识别技术的操作期间看到的那些相对应的数据项的序列。然而,对训练序列进行注释以对每个数据项进行分类。

[0052] 数据项的序列可以描述可由姿势识别技术解释的各个不同特征。例如,这些特征包括但不限于:

- [0053] ●从身体姿态估算器推导出的两个或更多个身体部位之间的角度;
- [0054] ●从身体姿态估算器推导出的两个或更多个身体部位之间的角度的变化率;
- [0055] ●使用身体姿态估算器跟踪的一个或多个身体部位的速度;
- [0056] ●移动设备的惯性、加速度或定向;
- [0057] ●来自话筒或语音识别器的语音信号;
- [0058] ●原始深度图像特征(即,不是来自身体姿态估算器的),诸如深度上的光学流和/或所跟踪的特征点的速度;
- [0059] ●原始 RGB 图像特征,诸如该 RGB 图像中的光学流的统计数据;
- [0060] ●基于与原始深度图像结合的身体姿态估算器输出的特征,诸如身体部位概率的时间导数;或
- [0061] ●这些特征的任何组合。

[0062] 每个姿势具有一个或多个时间点,在该时间点处触发该姿势与其有关的命令或动作。这被称为该姿势的“动作点”,且表示该姿势运动的末端(或姿势运动期间的可标识点)(例如,挥拳姿势的顶点)。在姿势的动作点处的数据项及其时间历史被分类为属于该姿势,并且所有其他数据项被分类为“背景”。该组经训练的序列可包括与多个不同姿势有关的序列,并且可包括与不同的测量或测量组合(例如,角度、速度、加速度等)有关的数据序列。在一些示例中,训练序列可能被随机地时间翘曲或适应该特征(例如,将来自身体姿态估算器的基于骨架的特征重新目标化到不同大小的骨架)扰乱。

[0063] 图 4 示出了用于训练决策树林以识别姿势的过程的流程图。首先,接收该组经注释的训练序列 402,并且选择要在随机决策树林中使用的决策树的数量 404。随机决策树林是决定性决策树的集合。决策树可以在分类算法中使用,但是可能遭受过拟合,过拟合导致不良的概括(generalization)。然而,许多经随机训练的决策树的全体(随机树林)产生改进的概括。在训练过程期间,树的数量是固定的。在一个示例中,树的数量是十个,然而也可以使用其他值。

[0064] 下面的注释被用于描述用于姿势识别的训练过程。树林由被表示为 $\Psi_1, \dots, \Psi_t, \dots, \Psi_T$ 的 T 个树构成,其中 t 索引每个树。图 5 示出了示例随机决策树林。图 5 的说明性决策树林包括三个决策树:第一树 500(表示为树 Ψ_1);第二树 502(表示为树 Ψ_2);以及第三树 504(表示为树 Ψ_3)。每个决策树包括根节点(例如,第一决策树 500 的根节点 506)、多个被称为分割节点(例如,第一决策树 500 的分割节点 508)的内部节点、以及多个叶节点(例如,第一决策树 500 的叶节点 510)。

[0065] 在操作中,每个树的每个根和分割节点对输入数据执行二进制测试并基于其结果

来将数据导向左侧或右侧的子节点。叶节点不执行任何动作；它们仅存储概率分布（例如，图 5 的第一决策树 500 的一叶节点的示例概率分布 512），如后文所述的。

[0066] 现在参考图 4 的剩余部分来描述选择每个分割节点所使用的参数的方式以及如何计算叶节点概率。从该决策树林中选择决策树（例如，第一决策树 500）406，并且根节点 506 被选择 408。然后选择来自该训练集合的所有经注释的序列。该训练集合中的序列中的每个数据项（及其相关联的时间历史） x 与已知类标记（用 $Y(x)$ 表示）相关联。因此，例如， $Y(x)$ 指示数据项 x 是否与挥拳、踢、跳、射击、呼叫、选择、回答、退出或背景的姿势类的动作点有关，其中背景类标记指示数据项 x 不与已定义的姿势动作点有关。

[0067] 然后生成测试参数的随机集合以供在根节点 506 处执行的二进制测试使用 410。在一个示例中，该二进制测试的形式是： $\xi > f(x; \theta) > \tau$ ，以使得 $f(x; \theta)$ 是被应用到该序列中的数据项 x 的函数，该函数具有参数 θ ，并且将该函数的输出与门限值 ξ 和 τ 进行比较。如果 $f(x; \theta)$ 的结果在 ξ 和 τ 之间的范围内，则该二进制测试的结果为真。否则，该二进制测试的结果为假。在其他示例中，可以只使用门限值 ξ 和 τ 中的一个，以使得如果 $f(x; \theta)$ 的结果大于（或可另选地，小于）门限值则该二进制测试的结果为真。在此所述示例中，参数 θ 定义该序列中的时间点从当前数据项的偏移，以及可任选地在该偏移处从该序列读取的特征的类型。后文参考图 6 描述了示例函数 $f(x; \theta)$ 。

[0068] 在根节点或分割节点处执行的二进制测试的结果确定数据项被传递到哪个子节点。例如，如果二进制测试的结果为真，则该数据项被传递到第一子节点，而如果结果为假，则该数据项被传递到第二子节点。

[0069] 所生成的测试参数的随机集合包括函数参数 θ 的多个随机值以及门限值 ξ 和 τ 。为了将随机性注入到决策树中并且减少计算，仅相对所有可能的参数的随机采样的子集 Θ 来优化每个分割节点的函数参数 θ 。例如，子集 Θ 的大小可以是五百。这是将随机性注入到树中，并且增加概括，同时避免所有可能测试上的计算密集搜索的有效而简单的方式。

[0070] 然后，将测试参数的每种组合应用到训练集合中的每个数据项。换言之，对于每个训练序列中的每个经注释的数据项，与 ξ 和 τ 的所有可用值相组合，一个接一个地尝试 θ 的所有可用值（即， $\theta_i \in \Theta$ ）。对于每个组合，计算信息增益（也被称为相对熵）。选择使信息增益最大化的参数的组合（被表示为 θ^* 、 ξ^* 和 τ^* ）414，并将其与当前节点相关联地存储以供以后使用。作为对信息增益的另选，可以使用其他准则，诸如 Gini 熵，或“双 ing(two-ing)”准则。

[0071] 然后确定 416 用于最大化的信息增益的值是否小于门限。如果用于该信息增益的值小于门限，则这指示该树的进一步扩展不提供明显的好处。这引起了异步树，异步树在没有更多节点有用时自然停止生长。在此情况下，将当前节点设定 418 为叶节点。类似地，确定 416 该树的当前深度（即，在根节点和当前节点之间有多少层节点）。如果此值大于预定义的最大值，则将当前节点设定 418 为叶节点。在一个示例中，可以将最大树深度设定为 15 层，然而也可以使用其他值。

[0072] 如果用于最大化的信息增益的值大于或等于该门限而树的深度小于最大值，则将当前节点设定 420 为分割节点。在当前节点是分割节点时，它具有子节点，并且该过程随后移至训练这些子节点。在当前节点处使用所述数据项的子集来训练每个子节点。使用使信

息增益最大化的参数 θ^* 、 ξ^* 和 τ^* 来确定被发送至子节点的数据项的子集。在二进制测试中使用这些参数并且在当前节点处对所有数据项执行 422 该二进制测试。通过该二进制测试的数据项形成被发送至第一子节点的第一子集，而没有通过该二进制测试的数据项形成被发送至第二子节点的第二子集。

[0073] 对于子节点中的每一个，对于针对相应子节点的训练数据项的子集递归执行 424 图 4 的框 410 到 422 中所概括的过程。换言之，对于每个子节点，生成 410 新的随机测试参数并将其应用 412 到数据项的相应子集，选择 414 最大化信息增益的参数，并确定 416 节点的类型（分割节点还是叶节点）。如果它是叶节点，则当前递归分支停止。如果它是分割节点，则执行 422 二进制测试以确定数据项的更多子集并且另一递归分支开始。因此，这个过程以递归方式遍历树，从而训练每个节点，直到在每个分支到达叶节点。当到达叶节点时，该过程等待 426，直到所有分支中的节点已被训练。注意，在其他示例中，可使用可另选的递归技术来获得相同的功能。例如，一种可另选技术是训练“宽度优先”，其中树的整个层次被一次训练，即，在每次迭代中树的大小加倍。

[0074] 一旦该树中的所有节点被训练以确定使在每个分割节点处的信息增益最大化的二进制测试的参数，并且叶节点已被选择以终止每个分支，则该树的所有叶节点的概率分布可被确定。这是通过对达到所述叶节点中的每一个的训练数据项的类标记进行计数 428 来实现的。所有训练数据项在该树的叶节点处停止。当每个训练数据项具有与其相关联的类标记时，每个类中的训练数据项的总数量可以在每个叶节点处被计数。从叶节点处的每个类中的训练数据项的数量和该叶节点处的训练数据项的总数量，可以生成 430 该叶节点处的类的概率分布。为了生成该分布，直方图被归一化。可任选地，可以对所有类增加小的先前计数以使得没有类被分配零概率，这会改进概括。

[0075] 图 5 示出了叶节点 510 的示例概率分布 512。该概率分布显示了数据项的类 c 对照该叶节点处属于该类的数据项的概率，该概率被表示为 $P_{l_t(x)}(Y(x)=c)$ ，其中 l_t 指示第 t 个树的叶节点 1。换言之，叶节点存储被训练的类上的后验概率。因此这种概率分布能被用于确定到达该叶节点的数据项包括给定姿势类的动作点的可能性，如后文更详细地描述的。

[0076] 回到图 4，一旦确定了该树的叶节点的概率分布，则确定 432 该决策树林中是否存在更多树。如果存在，则选择该决策树林中的下一个树，并且该过程重复。如果该树林中的所有树均已被训练，而没有其他树剩下，则该训练过程完成且该过程终止 434。

[0077] 因此，作为该训练过程的结果，使用训练序列训练了多个决策树。每个树包括存储经优化的测试参数的多个分割节点以及存储相关联的概率分布的叶节点。由于参数是从在每个节点处使用的有限子集中随机生成的，树林中的树彼此有区分（即，不同）。

[0078] 参考图 6 示出了具有三个随机测试参数集合的序列的形式 $\xi > f(x; \theta) > \tau$ 的示例测试。图 6 示出了在动作点中完结的姿势的示例序列。图 6 的序列 600 被示出为具有特征值 602（诸如关节角度、速度、惯性、深度 /RGB 图像特征、音频信号等中的一个或多个），对照时间 604，其中在当前时间 608 处出现动作点 606。例如，如果序列 600 与挥拳姿势有关，则动作点 606 处的数据项具有分类“挥拳”而所有其他数据项具有分类“背景”。

[0079] 注意，图 6 的示例为了清楚仅示出了包括单个特征的值的序列，而在其他示例中，该序列可具有并行描述若干不同特征的数据项，每个特征具有相关联的特征值。在其中存在若干特征的示例中，节点处的参数 θ 还可标识要针对给定测试分析哪个特征，但这在图

6 中没有被示出。还要注意，该序列包括（在大多数示例中）底层连续信号的离散样本。此外，这些样本可以不按照统一的时间间隔接收。因此该序列可以使用内插（例如，最近邻居、线性、二次等），其具有对不同时间间隔的补偿。

[0080] 在第一示例测试中，序列索引值 θ_1 610 已被随机生成，该索引值对应于沿该序列的一时间点。这可被表示为从当前时间 608 的偏移（在本示例中是动作点 606）。为了计算该序列的 $f(x; \theta_1)$ ，找到由索引值 θ_1 610 表示的时间实例处的该数据项的特征值。

[0081] 图 6 还示出了一对随机生成的门限 ξ_1 612 和 τ_1 614。门限 ξ_1 612 和 τ_1 614 表示与索引值 θ_1 610 相关联的特征值门限。因此，当索引值 θ_1 610 处的数据项的特征值小于门限 ξ_1 612 且大于门限 τ_1 614 时，通过了测试 $\xi_1 > f(x; \theta_1) > \tau_1$ （如图 6 的示例中的情况）。

[0082] 在第二示例测试中，第二序列索引值 θ_2 616 以及门限 ξ_2 618 和 τ_2 620 已被随机生成。如上，当索引值 θ_2 616 处的数据项的特征值在测试门限 ξ_2 618 和 τ_2 620 之间时，序列 600 通过示例测试 $\xi_2 > f(x; \theta_2) > \tau_2$ 。类似地，在第三示例中，第三序列索引值 θ_3 622 以及门限 ξ_3 624 和 τ_3 626 已被随机生成。再次，当索引值 θ_3 622 处的数据项的特征值在测试门限 ξ_3 624 和 τ_3 626 之间时，序列 600 通过示例测试 $\xi_3 > f(x; \theta_3) > \tau_3$ 。

[0083] 如果这三个随机生成的二进制测试被用于决策树中，则通过由所述参数限定的三个范围的序列满足所有三个二进制测试，并且（在本示例中）可以具有很高的概率是与在动作点 606 处出现的姿势相同的姿势。显然，本示例仅示出了索引值和门限的大量可能组合中的一些，并且仅是说明性的。尽管如此，这示出了可以如何通过考虑序列中的代表性或区别的点是否在门限内而捕捉序列之间的相似度。

[0084] 如果在上面描述的训练过程期间，算法是选择图 6 中示出的三组随机参数以在决策树的三个节点处使用，则这些参数可用于测试如图 7 所示的序列。图 7 示出了具有三个层次的决策树，其使用图 6 的示例测试。训练算法已选择来自图 6 的第一示例的第一组参数 θ_1 、 ξ_1 和 τ_1 作为在图 7 的决策树的根节点 700 处应用的测试。如上所述，该训练算法选择这个测试是因为它具有训练图像的最大信息增益。该序列的当前数据项 x （即，最新近接收的数据项）被应用到根节点 700，并对此数据项执行测试。如果使用图 6 的序列 600 作为示例，则可以看出，在 θ_1 处此项在门限 ξ_1 和 τ_1 之间，因此测试的结果为真。如果对在 θ_1 处在门限 ξ_1 和 τ_1 之外的序列执行该测试，则结果将为假。

[0085] 因此，当与姿势有关的数据项的序列被应用到图 7 的经训练决策树时，具有在 θ_1 处具有在门限 ξ_1 和 τ_1 之间的值的（即，通过该二进制测试的）数据项的序列被传递到子分割节点 702，而没有通过该二进制测试的序列被传递到其他子节点。

[0086] 该训练算法已选择来自图 6 的第二示例的第二组测试参数 θ_2 、 ξ_2 和 τ_2 作为在分割节点 702 处应用的测试。如图 6 所示，通过此测试的序列是在索引值 θ_2 处在门限 ξ_2 和 τ_2 之间通过的那些序列。因此，在只有通过与根节点 700 相关联的二进制测试的序列到达分割节点 702 的情况下，通过此测试的序列是既在 θ_1 处在门限 ξ_1 和 τ_1 之间又在 θ_2 处在门限 ξ_2 和 τ_2 之间通过的那些序列。通过该测试的序列被提供到分割节点 704。

[0087] 该训练算法已选择来自图 6 的第三示例的第三组测试参数 θ_3 、 ξ_3 和 τ_3 作为在分割节点 704 处应用的测试。图 6 示出了只有具有在 θ_3 处在 ξ_3 和 τ_3 之间的特征值的

那些序列通过此测试。只有通过根节点 700 和分割节点 702 处的测试的序列到达分割节点 704，所以通过分割节点 704 处的测试的序列是落在图 6 中示出的门限的每一个之间的那些序列（诸如图 6 的示例序列 600）。通过分割节点 704 处的测试的序列被提供给叶节点 706。

[0088] 叶节点 706 存储不同姿势类的概率分布 708。在本示例中，该概率分布指示到达此叶节点 706 的数据项是与挥拳姿势的动作点相对应的数据项的高概率 710。要理解，学习算法可以任意地对这些测试进行排序，而所评估的特征不必是按照时间次序的。

[0089] 在上面描述的图 6 和图 7 的示例中，当被测试的序列包含足够多的数据项以进行所讨论的测试时，这些测试中的每一个均能够被执行。然而，在一些情况下，树可以被训练以使得在不能被应用到某个序列的节点中使用测试。例如，如果被测试的序列具有数量少的数据项，则使用与从当前数据项回溯到过去的时间相对应的索引值 0 并且所述索引值大于数据项的数目的测试不能被执行。在这种情况下，不执行测试，并且向两个子节点均发送当前数据项，以使得该树中较低处的更多测试还能够用于获得结果。该结果可以通过利用所到达的所有叶节点来获得。在一可另选的示例中，为了避免对短序列的测试，可定义最大特征时间窗口（例如，1 秒的时间窗口），并且直到获得足够的读数才执行分类（例如，该序列的第一秒被忽略）。

[0090] 显然，图 6 和图 7 提供了简化示例，并且在实践中经训练的决策树可具有多得多的层次（并因此沿该序列计入多得多的采样点）。此外，在实践中，在树林中使用许多决策树，并且将结果相组合以增加准确度，如下面参考图 8 概述的。

[0091] 图 8 示出了用于使用已如上文所述训练过的决策树林来标识先前未见的序列中的特征的过程的流程图。首先，在姿势识别算法处接收 800 新数据项。该算法的目标是将该新数据项分类为指定姿势的动作点或背景。通过从该新数据项和存储设备处已存储的多个先前接收的数据项形成时间系列，来生成 802 序列。要生成的序列的长度可被预定义。在一个示例中，该算法可被安排为生成 30 个数据项的序列，然而可以使用任何适当的值。在一些示例中，当没有足够的先前数据项已被接收时，该序列可以更短。该序列可以被称为“未见”的，以将其与具有已被手工分类的数据项的训练图像相区分。

[0092] 注意，如同上面提到的，一些示例可以利用基于并发地形成的多个特征的序列。例如，序列可包括描述用户的关节间的角度以及用户的身体部位的速度两者的数据项。在这样的示例中，在该树的每个节点处的测试参数 0 指定要对照门限测试哪个特征。

[0093] 选择 804 来自该决策树林的经训练的决策树。推动 806 该新数据项及其相关联序列通过所选择的决策树（按照类似于上面参考图 6 所描述的方式），以使得在一节点处对照该经训练的参数进行测试，然后取决于该测试的结果而将其传递到适当的子，并且该过程重复直到该新数据项到达叶节点。一旦该新数据项到达叶节点，则为该新数据项存储 808 与此叶节点相关联的概率分布。

[0094] 如果确定 810 该树林中存在更多决策树，则选择 804 新决策树，推动 806 该新数据项通过该树并存储 808 概率分布。重复此过程，直到对于该树林中的所有决策树都执行了该过程。注意，用于推动新数据项通过决策树林中的多个树的过程还可以并行执行，而不是如图 8 所示按序列执行。

[0095] 一旦已推动该新数据项及其序列通过该决策树林中的所有树，则为该新数据项存储了多个姿势分类概率分布（每个树至少一个）。然后聚集 812 这些概率分布以形成该新

数据项的总概率分布。在一个示例中,总概率分布是来自该 T 个不同决策树的所有个体概率分布的平均值。这由下式给出 :

$$[0096] \quad P(Y(\mathbf{x})=c) = \frac{1}{T} \sum_{t=1}^T P_{l_t(\mathbf{x})}(Y(\mathbf{x})=c)$$

[0097] 注意,也可以使用除了求平均以外的组合树后验概率的可另选的方法,诸如将这些概率相乘。可任选地,可以执行对个体概率分布之间的可变性的分析(图 8 未示出)。这样的分析能够提供与总概率分布的不确定度有关的信息。在一个示例中,可确定标准偏差,作为可变性的度量。

[0098] 一旦找到了新数据项的总概率分布,则将每个分类的概率与和与每个分类相关联的门限值进行比较 814。在一个示例中,可为每个姿势分类设定不同的门限。

[0099] 如果确定 816 每个分类的概率不大于其相关联门限值,或指示该新数据项具有“背景”分类,则该过程等待接收下一个新数据项,并重复。然而,如果确定 816 一分类的概率大于其相关联的门限值并指示该新数据项不是“背景”,则这指示该新数据项与一姿势的动作点相对应。检测 818 与分类 c 相对应的姿势,并且执行 820 与此姿势相关联的功能、命令或动作,如同上面描述的。

[0100] 因此,上述姿势识别算法允许通过利用该序列中的先前接收的数据项的信息而将与特征有关的新接收的、未见的数据项分类为姿势的动作点。随机决策树林提供用于训练分类器并将测试应用到先前接收的数据项的有用技术。尽管用于训练该决策树的处理相对复杂,然而为了分类新数据项而执行的评估在处理方面相对轻量,并且可以被快速执行以最小化识别滞后。

[0101] 如果需要,通过训练决策树然后将经优化的测试硬编码到一组指令中,可以进一步减小识别滞后。例如,经训练的决策树所执行的测试可以被写出为 C 程序,然后由处理器编译。该编译针对所使用的处理器优化该 C 程序,并且使得决策树分类执行起来很快。

[0102] 现在参考图 9 和图 10,它们示出了用于数据项(诸如上面描述的那些)的序列识别特征的可另选的技术。此技术是基于经训练的逻辑斯蒂模型的。首先描述该技术的操作,然后描述该模型的训练。

[0103] 在一个示例中,所使用的模型是下面的对数 - 线性(log-linear)逻辑斯蒂模型:

$$[0104] \quad p_g(t) = \frac{1}{1 + \exp(-w_g^\top \phi(x_{(t-W+1):t}))}$$

[0105] 其中 $x_{(t-W+1):t}$ 是具有从当前时间 t 向后延伸长度 W 的数据项的序列(类似于上面参考决策树林示例所描述的那个), $\phi(x_{(t-W+1):t})$ 是用于测试该序列的特征函数(在下面更详细地描述),而 $p_g(t)$ 是来自时间 t 的数据项是姿势 g 的动作点的概率。通过学习适当的权重 w_g^\top ,这些特征可被加权并组合以产生总的姿势概率。

[0106] 参考图 9 描述在使用此模型时本技术的操作。首先,在姿势识别算法处接收 900 新数据项。通过从该新数据项和存储设备处已存储的多个先前接收的数据项形成时间系列,来生成 902 序列。该序列的长度被表示为 W,如上所述。

[0107] 例如,参考图 10,示出了说明性序列 1000,其包括在当前时间 1002 处接收的新数据项和多个先前接收的数据项,每个数据项具有特征值。注意,图 10 的示例仅示出了该序

列中的单个类型的特征，然而在其他示例中，该序列可包括多个不同特征。然后将测试特征函数 $\phi(x_{(t-w+1):t})$ 应用到该序列。在本示例中，此测试包括将所生成的序列与一组预定义的已存储模板 906 进行比较 904，该模板先前已在训练过程中被选择以提供姿势的指示性示例（也在图 10 中示出）。确定该序列和所存储的模板中的每一个之间的类似度的测量。在一个示例中，所使用的类似度的测量是该模板和该序列之间的欧几里德距离。然后通过确定 908 该类似度是否大于预定义门限值来找到该测试的结果。

[0108] 更正式而言，这些模板中的每一个具有与其相关的一组参数。例如，每个模板可包括参数 ω 、 A 、 M 、 r 、 v 和 w_f ，其中 ω 定义该模板序列的长度， A 定义要测试的特征， M 是模板模式本身， r 是用于确定 908 该模板是否与该序列足够类似的门限值， v 是到权重或投票要投射的未来的距离（下面更详细地描述），而 w_f 是与该模板相关联的权重。然后给定模板 f 的测试特征函数 $\phi(x_{(t-w+1):t})$ 的结果由下式给出：

[0109]

$$\phi_f(x_{(t-w+1):t}) = \begin{cases} 1 & D(x_{(t-v-\omega+1):(t-v)}|_A, M) \leq r, \\ 0 & \text{否则} \end{cases}$$

[0110] 其中符号 $|_A$ 表示在特征的集合 A 上的投影，而 $D(\cdot, \cdot)$ 是该序列和该模板之间的距离函数（诸如欧几里德距离）。

[0111] 如果确定 908 该序列和该模板之间的类似度不大于该门限（即， $\phi_f(x_{(t-w+1):t})$ 的结果为 0），则该过程等待要接收的下一个数据项，并重复。然而，如果确定 908 该序列和该模板之间的类似度大于该门限（即， $\phi_f(x_{(t-w+1):t})$ 的结果为 1），则计算 910 与该类似模板相关联的未来时间。这是使用将上面提到的参数 v 加到当前时间来计算的。例如，参考图 10，从类似模板的参数计算未来时间 1004。

[0112] 该姿势识别算法维护姿势可能性列表 1006，该列表是存储与给定姿势动作点在给定时间发生的所计算的可能性相对应的值的时间系列列表。将该模板的权重 w_f 添加 912 到在该未来时间 1004 处的该姿势可能性列表。换言之，如图 10 所示，权重 w_f 被投射 1008 到该未来，并在未来时间 1004 处被与任何现有可能性值相聚集以给出与该模板相关联的特征的经更新的可能性值 1010。

[0113] 然后读取 914 姿势可能性列表 1006 以确定当前时间 t 的姿势可能性值（诸如可能性值 1012）。如果确定 916 这些姿势可能性值中的一个大于预定义值，则算法认为当前时间 t 是此姿势的动作点，而该姿势被检测到并可以对其进行动作，如同前面概述的。另一方面，如果确定 916 这些姿势可能性值均不大于该预定义值，则该算法没有检测到姿势，而等待新数据项并重复。

[0114] 因此，通过对未来时间投射权重，本算法确保当姿势被执行时该姿势的可能性被开发，以使得当出现动作点时，用于姿势探测的处理的主要部分已经被完成，并因此可以低滞后地检测到该姿势。例如，看图 10，如果用于检测姿势的预定义值 1014 如所示一样设定，则投射 1008 到未来时间 1004 的权重 w_f 足以使得经更新的可能性值 1010 大于预定义值 1014。因此，当到达未来时间 1004 时，已经确定看到了足够的证据来指示未来时间 1004 是姿势的动作点，并且使得此姿势能被快速检测到。

[0115] 将权重投射到未来中的方案使得姿势识别系统能够向应用提供中间结果。例如，

在计算机游戏的情况下,这允许游戏设计者提前知道是否有可能的姿势就要到来,并因此向用户提供某种反馈。例如,如果用户开始挥拳,则计算机游戏图形可以在游戏中显示虚拟的挥拳线路以鼓励你完成该挥拳。这可以被安排为当该姿势变得更可信(即,对于该姿势,更多权重被投射)时淡入(fade in)。这使得用户能够在背景移动期间姿势暗淡地显示在屏幕上时更容易发现这些姿势。

[0116] 为了允许上述姿势识别算法,使用机器学习技术,基于与上面参考决策树林示例描述的数据相类似的经注释的训练数据,学习模板的参数。在一示例中,权重的学习可以作为逻辑斯蒂回归问题而被投射。如此,在一个示例中,模板的参数可以通过采样随机集合来学习。在一可另选示例中,为了增加选择出良好集合的机会,可以用类似增强(boosting-like)的方式,按需生成良好特征。此外,为了确保仅有几个特征接收非零权重 w_f ,则可以使用 w 上的稀疏性引起范数(sparsity-inducing norm)。这引起了以下学习问题:

$$[0117] \min_w \|w\|_1 + \sum_{i=1}^{\ell} \sum_{t=1}^{T_i} C_{i,t} \log \left(1 + \exp \left(-y_{i,t} w^\top \phi(x_{i,(t-W+1):t}) \right) \right) \quad (1)$$

[0118] 其中 T_i 是第*i*个训练序列的长度,而 $C_{i,t} > 0$ 确定正确预测序列*i*中的第*t*项的相对重要性。在解决上述学习问题以后,逐数据项权重(per-data item weight)可以被推导为:

$$[0119] \lambda_{i,t} = C_{i,t} y_{i,t} \frac{\exp \left(-y_{i,t} w^\top \phi(x_{i,(t-W+1):t}) \right)}{1 + \exp \left(-y_{i,t} w^\top \phi(x_{i,(t-W+1):t}) \right)}$$

[0120] 这允许对所谓“定价问题”的定义,以找到保证改进分类器的要添加的特征,如在方程(1)中的目标值上测量的:

$$[0121] \min_{f \in F} - \sum_{i=1}^{\ell} \sum_{t=1}^{T_i} \lambda_{i,t} \phi_f \left(x_{i,(t-W+1):t} \right) \quad (2)$$

[0122] 这是对模板参数的所有可能值的假设空间F的优化问题。对此定价问题求解使得当被添加到 F' (F的子集)时使方程(1)的目标减小最多的特征能被找到。例如,如果 f^* 是方程(2)的最小化者(minimizer),则若有下式则目标减小:

$$[0123] \left| \sum_{i=1}^{\ell} \sum_{t=1}^{T_i} \lambda_{i,t} \phi_{f^*} \left(x_{i,(t-W+1):t} \right) \right| > 1$$

[0124] 在一个示例中,通过随机选择参数 ω 、A和v,可以优化方程(2)。模板模式M是从训练序列中的姿勢动作点之前取的,而r是对其明确优化的。这提供了对方程2的解的逼近。所选择的数量个最前面的特征被维持(诸如前几百个)并被用于求解方程(1)。

[0125] 因此上面的两种技术(基于决策树和逻辑斯蒂模型)均提供了允许以在计算上高效的方式、低等待时间地检测和识别用户特征的姿势识别技术。这些姿势识别技术已在能够用由相机捕捉的用户姿势控制的静态设备(诸如游戏设备)、以及能够用由设备内的运动和/或定向传感器检测的用户姿势控制的移动手持设备(诸如移动电话)的示例上下文中示出。

[0126] 在进一步示例中,这两种上下文可被组合。例如,手持移动设备可以与游戏设备通

信,使得来自手持设备(例如,惯性传感器)的特征能够与来自相机的特征(例如,身体部位特征)相组合。这可以用于允许从手持设备传感器捕捉细微的运动细节并将其合并到姿势识别(其可在静态设备中执行)中。作为说明性示例,这可以在其中用户在玩保龄球时将移动设备持在手中的保龄游戏中使用,以使用相机来捕捉用户的更广的、粗略的移动,而使用手持设备传感器来捕捉表示例如球上的旋转的细微移动。这些分开的信号共同用于姿势识别以控制保龄游戏的操作。

[0127] 这可以通过移动设备向静态设备报告连续的传感器数据流,而静态设备使用这两种类型的特征来执行姿势识别来执行。在一可另选的示例中,可以在移动设备和静态设备两者上分开地运行识别算法,而移动设备可被安排为一旦识别出姿势就将离散的所识别的姿势信号发送回静态设备,从而减少带宽使用。

[0128] 在上述示例中,姿势仅在该姿势的完成处具有动作点。然而,在其他示例中,姿势识别技术可用于检测包括若干动作点的姿势。例如,如果姿势包括环形转动手持设备,则这可以被划分为四个分开的子姿势,每个子姿势对应于通过罗盘上的四个点中的一个的移动。因此,“环”姿势可以被认为具有四个动作点。这些子姿势中的每一个按照序列被检测,以触发总的“环”姿势。这些子姿势中的每一个可以在上述分类器中被训练为类,并被识别。来自分类器的分类输出可以被提供给状态机,该状态机可以通过按照已定义的方式组合所检测到的子姿势以在满足状态机条件时触发整个姿势。

[0129] 图11示出了示例性计算设备104的各种组件,该计算设备可以实现为任何形式的计算设备和/或电子设备,且在该计算设备中可以实现上述姿势识别技术的实施例。

[0130] 计算设备104包括一个或多个处理器1102,这些处理器可以是微处理器、控制器或用于处理计算可执行指令以控制设备操作以便执行姿势识别的任何其他合适类型的处理器。在一些示例中,例如在使用片上系统架构的示例中,处理器1102可以包括一个或多个固定功能块(亦称为加速器),这些块以硬件(而非软件或固件)来实现姿势识别方法的一部分。

[0131] 基于计算的设备104还包括输入接口1104,该输入接口被安排为从一个或多个设备(诸如图2的捕捉设备106或图3的传感器)接收输入。输出接口1106也被提供并被安排为向例如与基于计算的设备集成或通信的显示系统(诸如显示设备108或310)提供输出。显示系统可提供图形用户界面,或任何合适类型的其他用户界面,但这不是必需的。可以可任选地提供通信接口1108,该通信接口可被安排为与一个或多个通信网络(例如,因特网)进行通信。

[0132] 可以使用可由基于计算的设备104访问的任何计算机可读介质来提供计算机可执行指令。计算机可读介质可以包括例如诸如存储器1110等计算机存储介质和通信介质。诸如存储器1110等计算机存储介质包括以用于存储如计算机可读指令、数据结构、程序模块或其他数据等信息的任何方法或技术实现的易失性和非易失性、可移动和不可移动介质。计算机存储介质包括但不限于, RAM、ROM、EPROM、EEPROM、闪存或其他存储器技术、CD-ROM、数字多功能盘(DVD)或其他光存储、磁带盒、磁带、磁盘存储或其他磁存储设备,或者可用于存储信息以供计算设备访问的任何其他非传输介质。相反,通信介质可以以诸如载波或其他传输机制等已调制数据信号来体现计算机可读指令、数据结构、程序模块或者其他数据。如本文所定义的,计算机存储介质不包括通信介质。虽然在基于计算的设备104

中示出了计算机存储介质（存储器 1110），然而应当理解，该存储可以是分布式的或位于远处并经由网络或其他通信链路（例如，使用通信接口 1108）来访问。

[0133] 可以在基于计算的设备处提供包括操作系统 1112 或任何其他合适的平台软件的平台软件以使得能够在该设备上执行应用软件 218。存储器 1110 可存储可执行指令以实现（上面参考图 2 描述的）身体姿态估算器 214 和（例如，使用如上所述的经训练的决策树林或回归模型的）姿势识别引擎 216 的功能。存储器 1110 还可提供数据存储 1114，该数据存储可被用来在执行姿势识别技术时提供对处理器 1102 所使用的数据的存储，如用于先前接收的数据项、经训练的树的参数和 / 或模板。

[0134] 此处所使用的术语‘计算机’是指带有处理能力使得它可以执行指令的任何设备。本领域的技术人员将认识到，这样的处理能力被集成到许多不同的设备中，因此，术语‘计算机’包括 PC、服务器、移动电话、个人数字助理和许多其他设备。

[0135] 本文描述的方法可由有形存储介质上的机器可读形式的软件来执行，例如计算机程序的形式，该计算机程序包括在该程序在计算机上运行时适用于执行本文描述的任何方法的所有步骤的计算机程序代码装置并且其中该计算机程序可被包括在计算机可读介质上。有形（或非瞬态）存储介质的示例可包括盘（disk）、拇指型驱动器、存储器等，而不包括所传播的信号。软件可以适合于在并行处理器或串行处理器上执行，使得各方法步骤可以以任何适当的顺序实现，或同时实现。

[0136] 这确认了软件可以是有价值的、可单独交易的商品。它旨在包含运行于或者控制“哑”或标准硬件以实现所需功能的软件。它还旨在包含例如用于设计硅芯片，或者用于配置通用可编程芯片的 HDL（硬件描述语言）软件等描述或者定义硬件配置以实现期望功能的软件。

[0137] 本领域的技术人员将认识到，用来存储程序指令的存储设备可以分布在网络上。例如，远程计算机可以存储被描述为软件的进程的示例。本地或终端计算机可以访问远程计算机并下载软件的一部分或全部以运行程序。可另选地，本地计算机可以根据需要下载软件的片段，或在本地终端上执行一些软件指令，并在远程计算机（或计算机网络）上执行另一些软件指令。本领域的技术人员还将认识到，通过利用本领域的技术人员已知的传统技术，软件指令的全部或一部分可以通过诸如 DSP、可编程逻辑阵列等等之类的专用电路来实现。

[0138] 如本领域技术人员将清楚的，此处给出的任何范围或者设备值都可以被扩展或者改变而不失去所寻求的效果。

[0139] 尽管用结构特征和 / 或方法动作专用的语言描述了本主题，但可以理解，所附权利要求书中定义的主题不必限于上述具体特征或动作。相反，上述具体特征和动作是作为实现权利要求的示例形式公开的。

[0140] 可以理解，上文所描述的好处和优点可以涉及一个实施例或可以涉及多个实施例。各实施例不限于解决所述问题中的任一个或全部的实施例或具有所述好处和优点中的任一个或全部的实施例。进一步可以理解，对‘一’项目的引用是指那些项目中的一个或多个。

[0141] 此处所描述的方法的步骤可以在适当的情况下以任何合适的顺序，或同时实现。另外，在不偏离此处所描述的主题的精神和范围的情况下，可以从任何一个方法中删除各

单独的框。上文所描述的任何示例的各方面可以与所描述的其他示例中的任何示例的各方面相结合,以构成进一步的示例,而不会丢失寻求的效果。

[0142] 此处使用了术语“包括”旨在包括已标识的方法的框或元素,但是这样的框或元素不构成排它性的列表,方法或设备可以包含额外的框或元素。

[0143] 可以理解,上面对一较佳实施例的描述只是作为示例给出并且本领域的技术人员可以做出各种修改。以上说明、示例和数据提供了对本发明的各示例性实施例的结构和使用的全面描述。虽然上文以一定的详细度或参考一个或多个单个实施例描述了本发明的各实施例,但是,在不偏离本发明的精神或范围的情况下,本领域的技术人员可以对所公开的实施例作出很多更改。

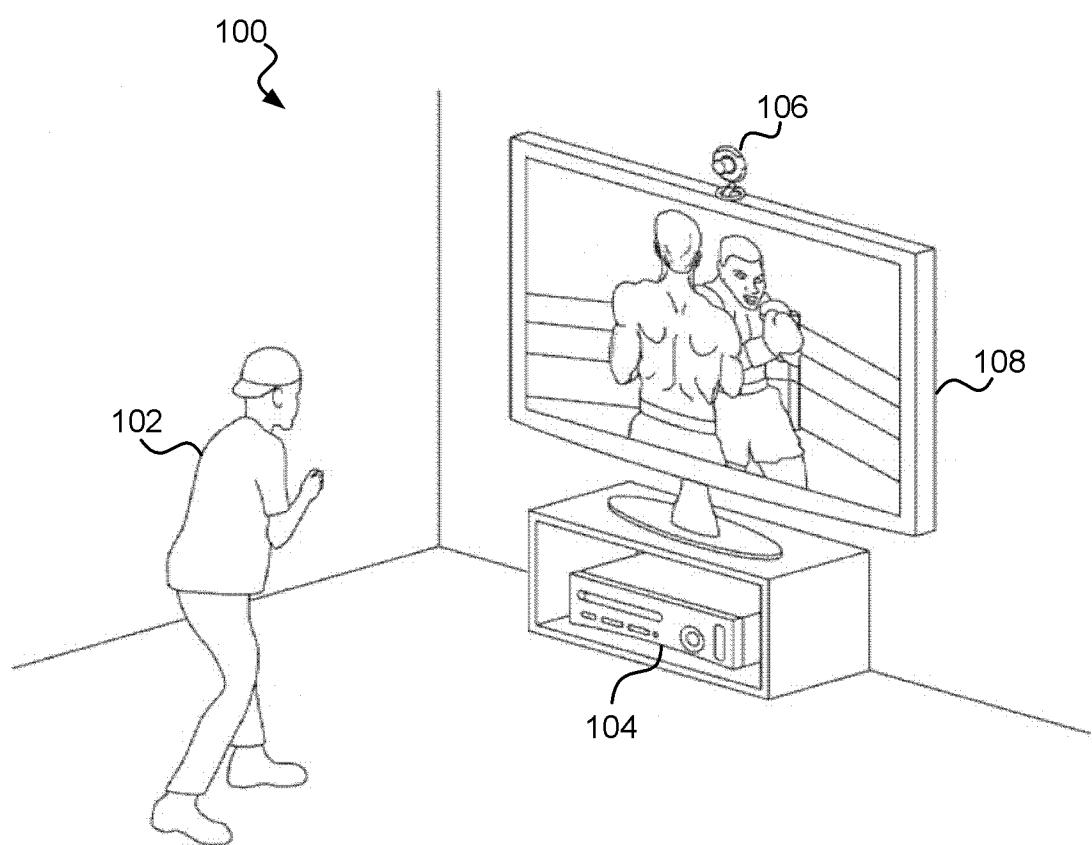


图 1

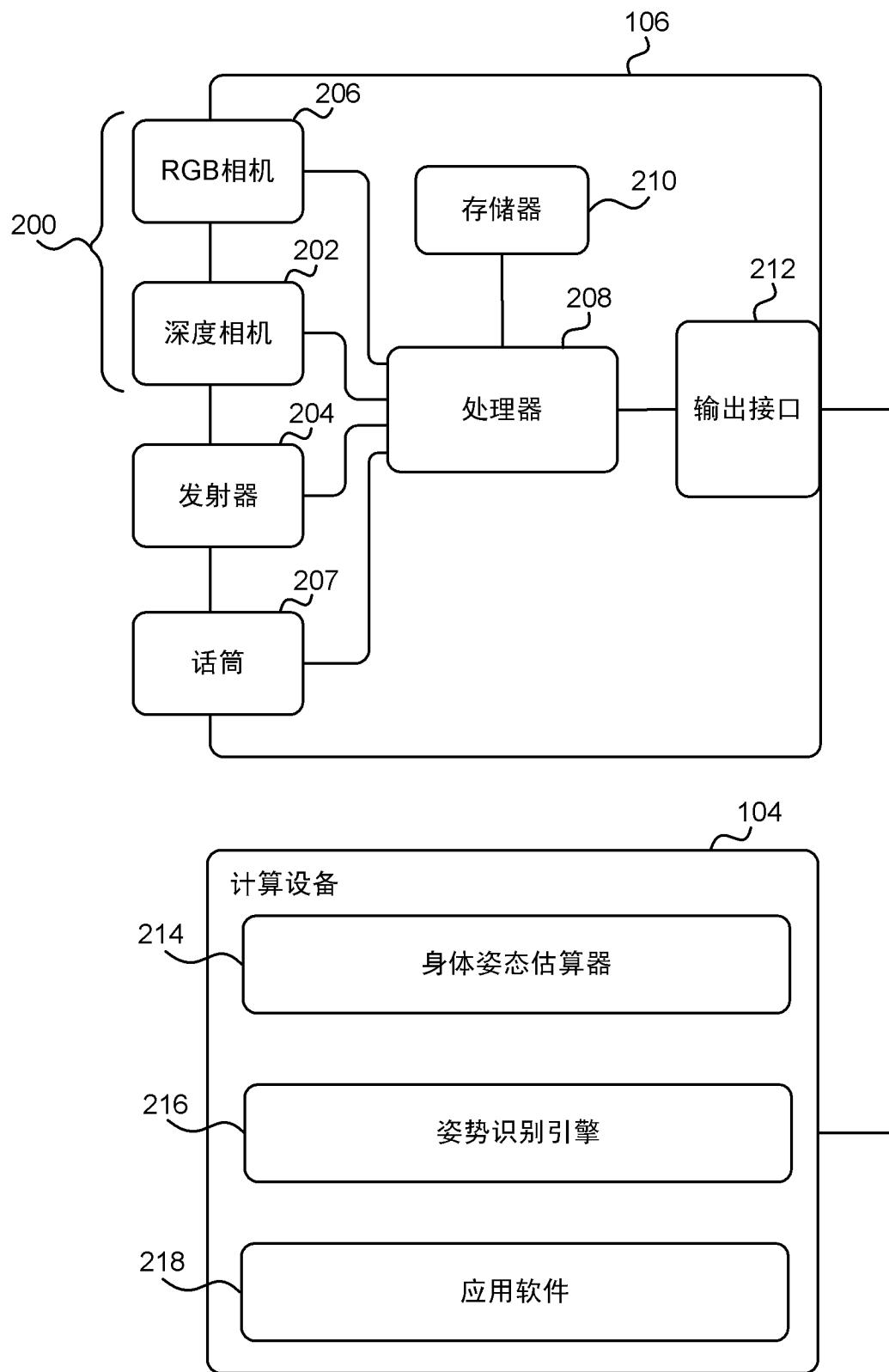


图 2

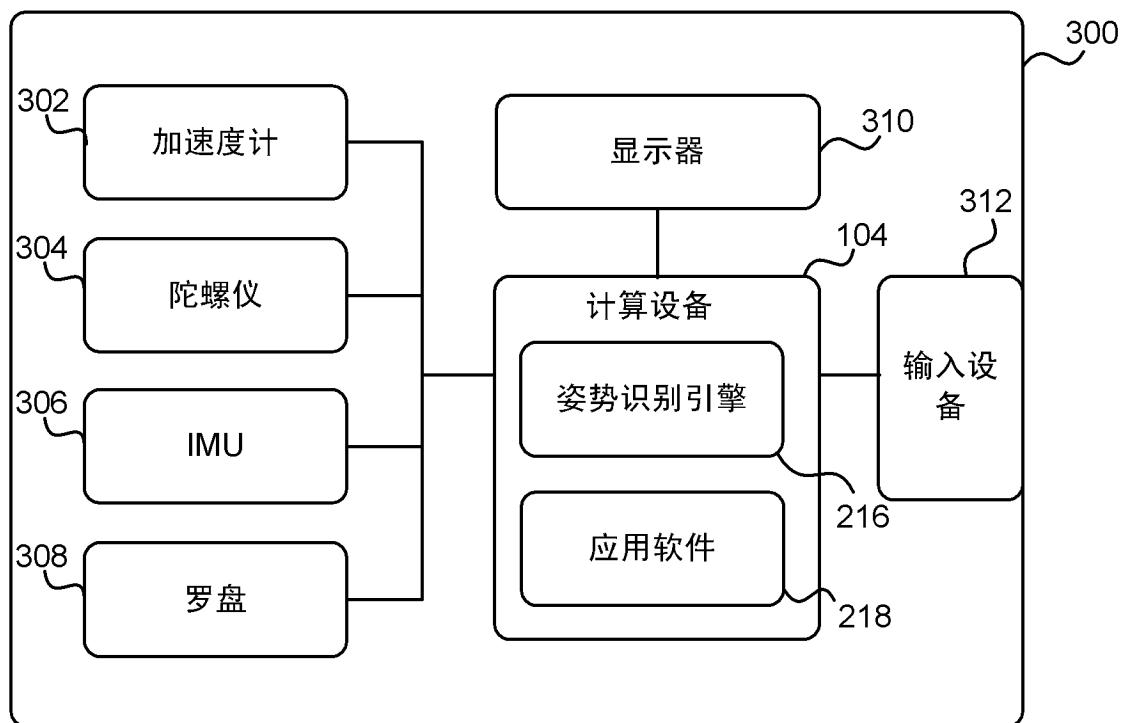


图 3

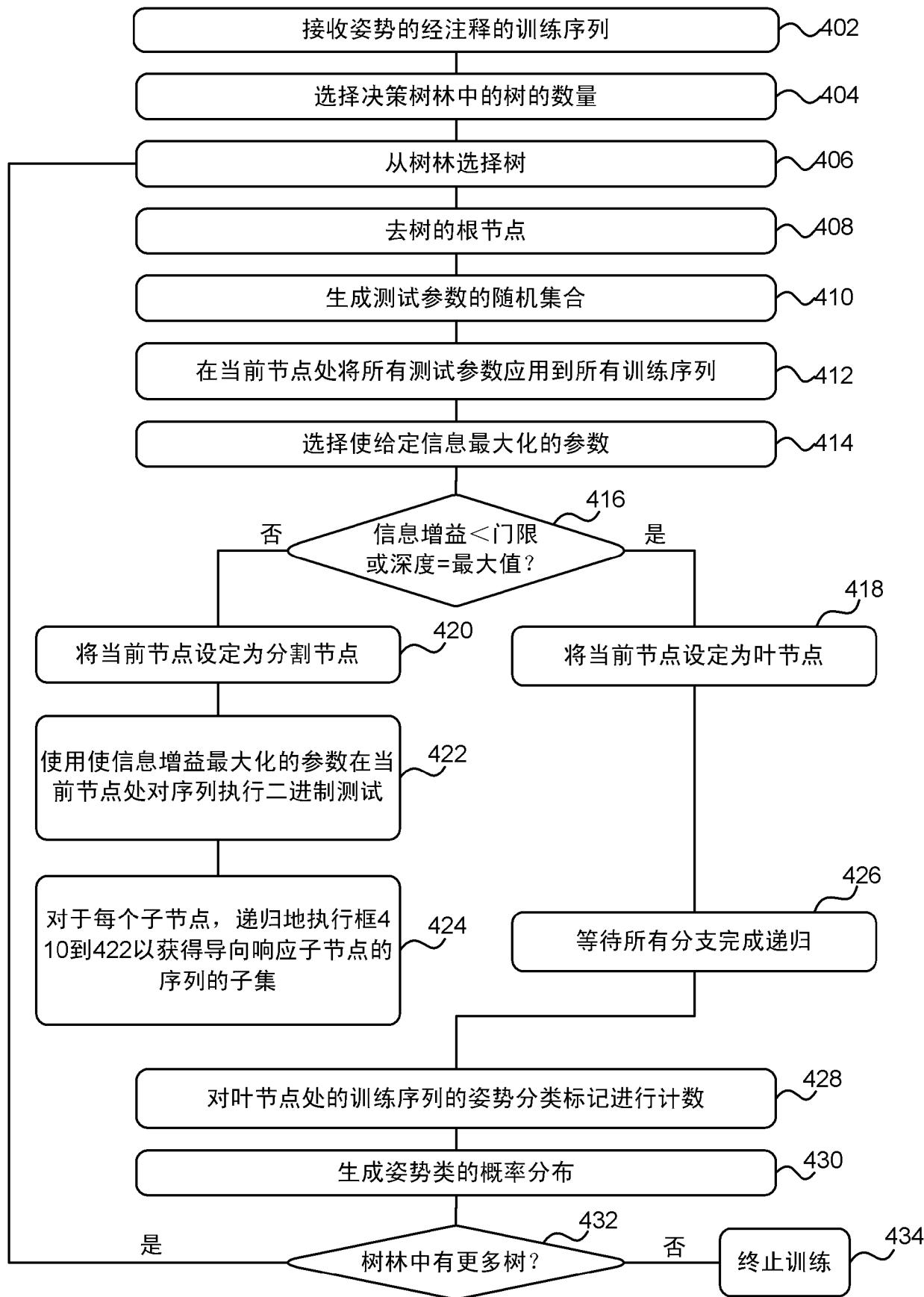


图 4

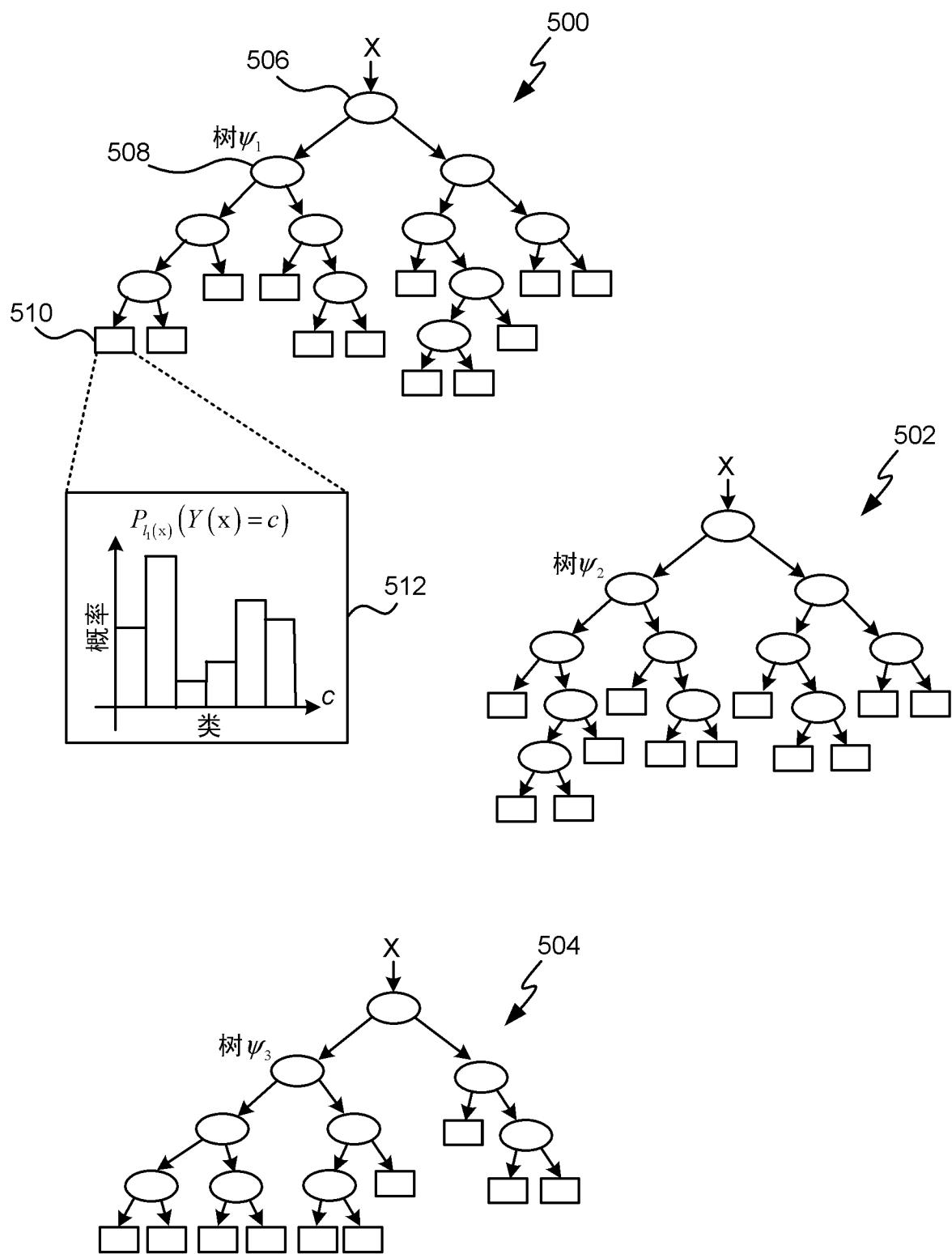


图 5

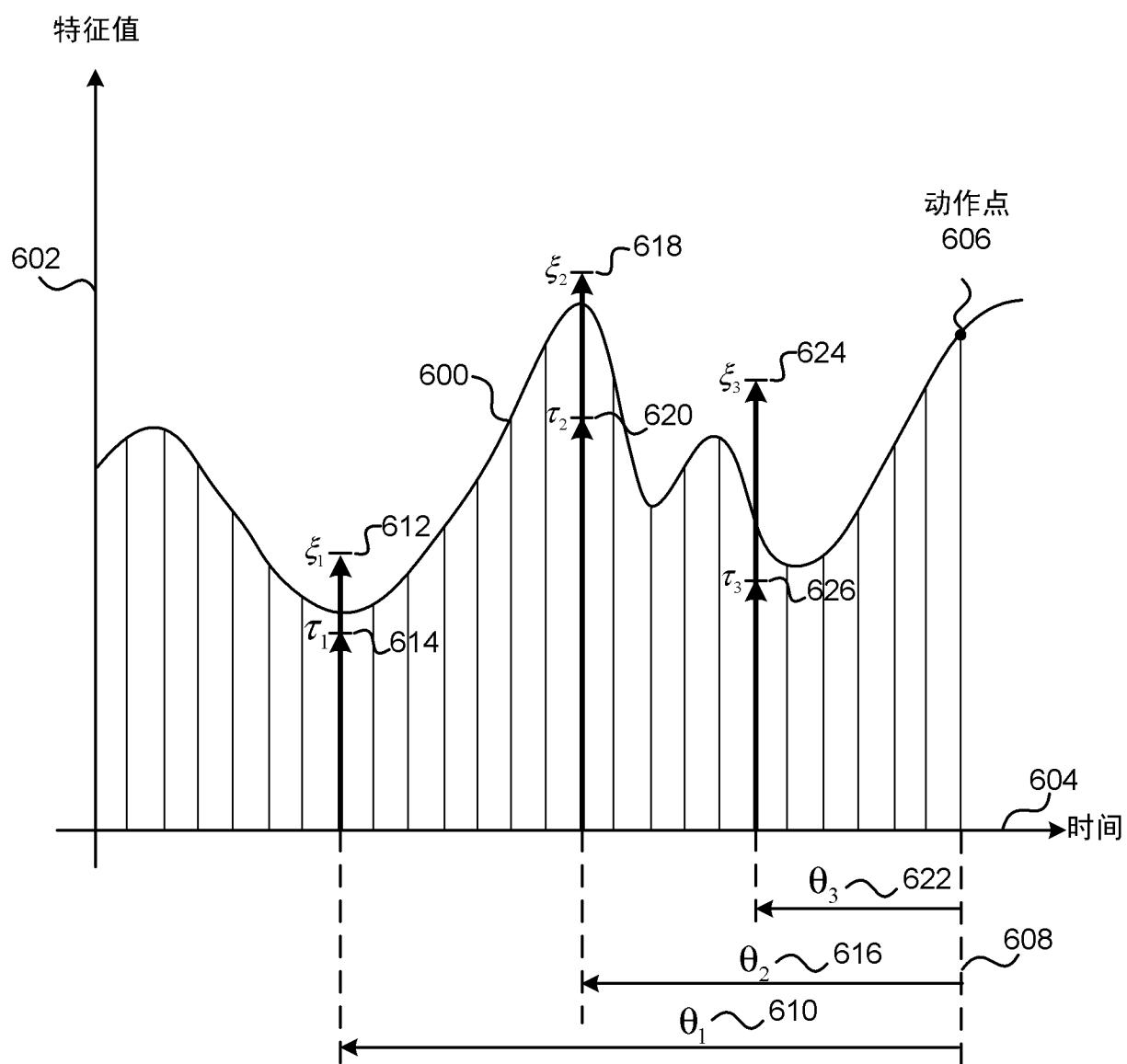


图 6

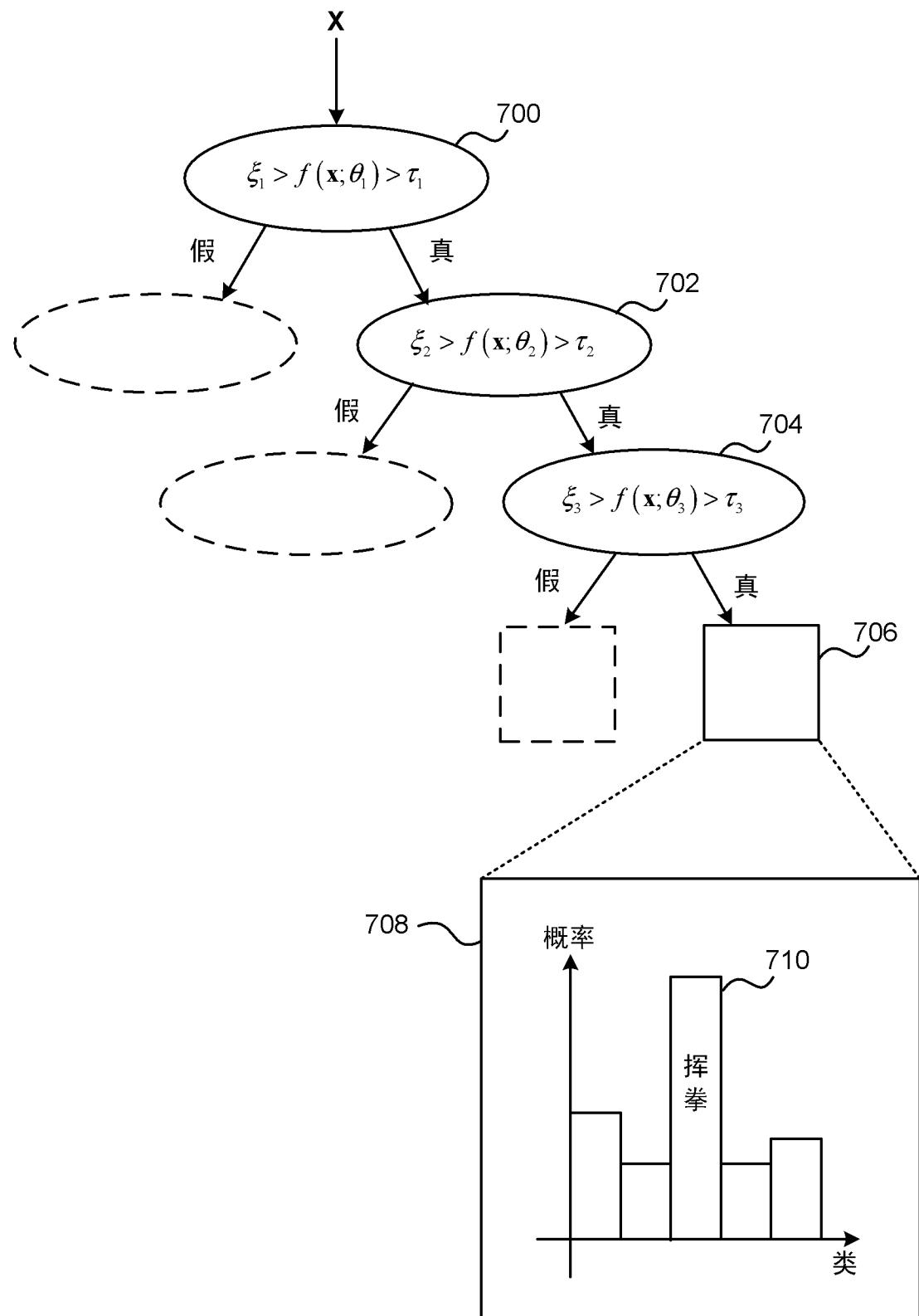


图 7

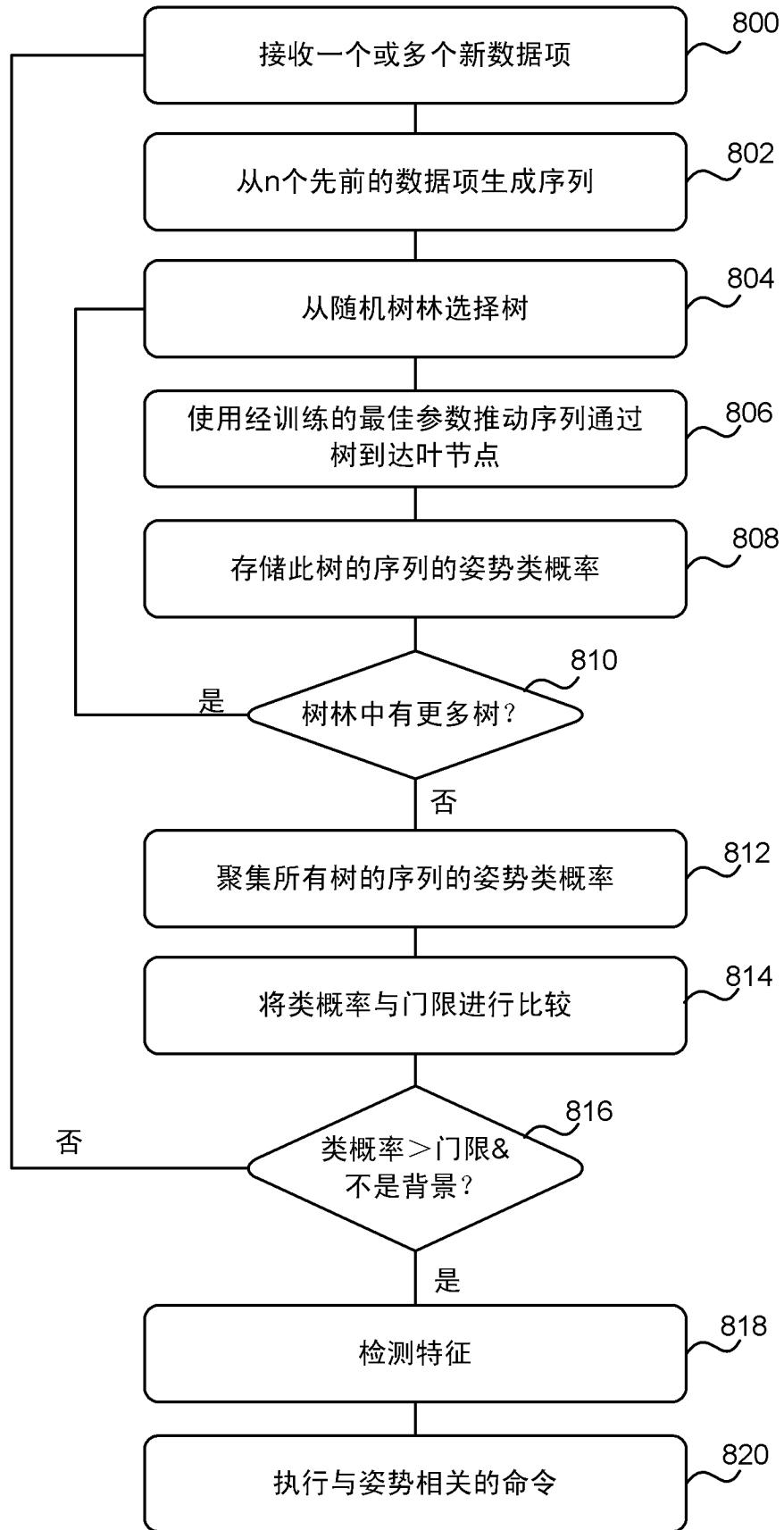


图 8

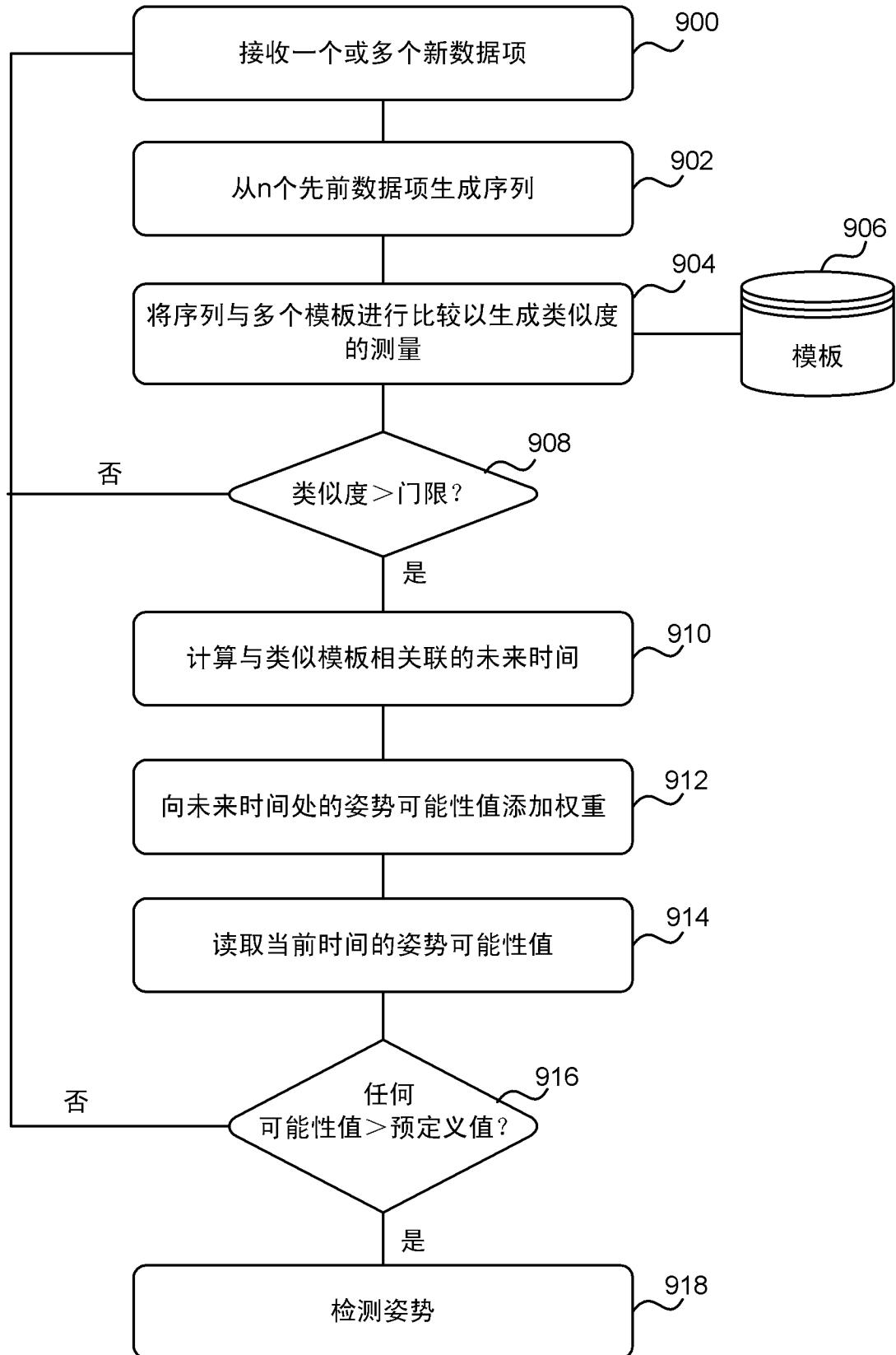


图 9

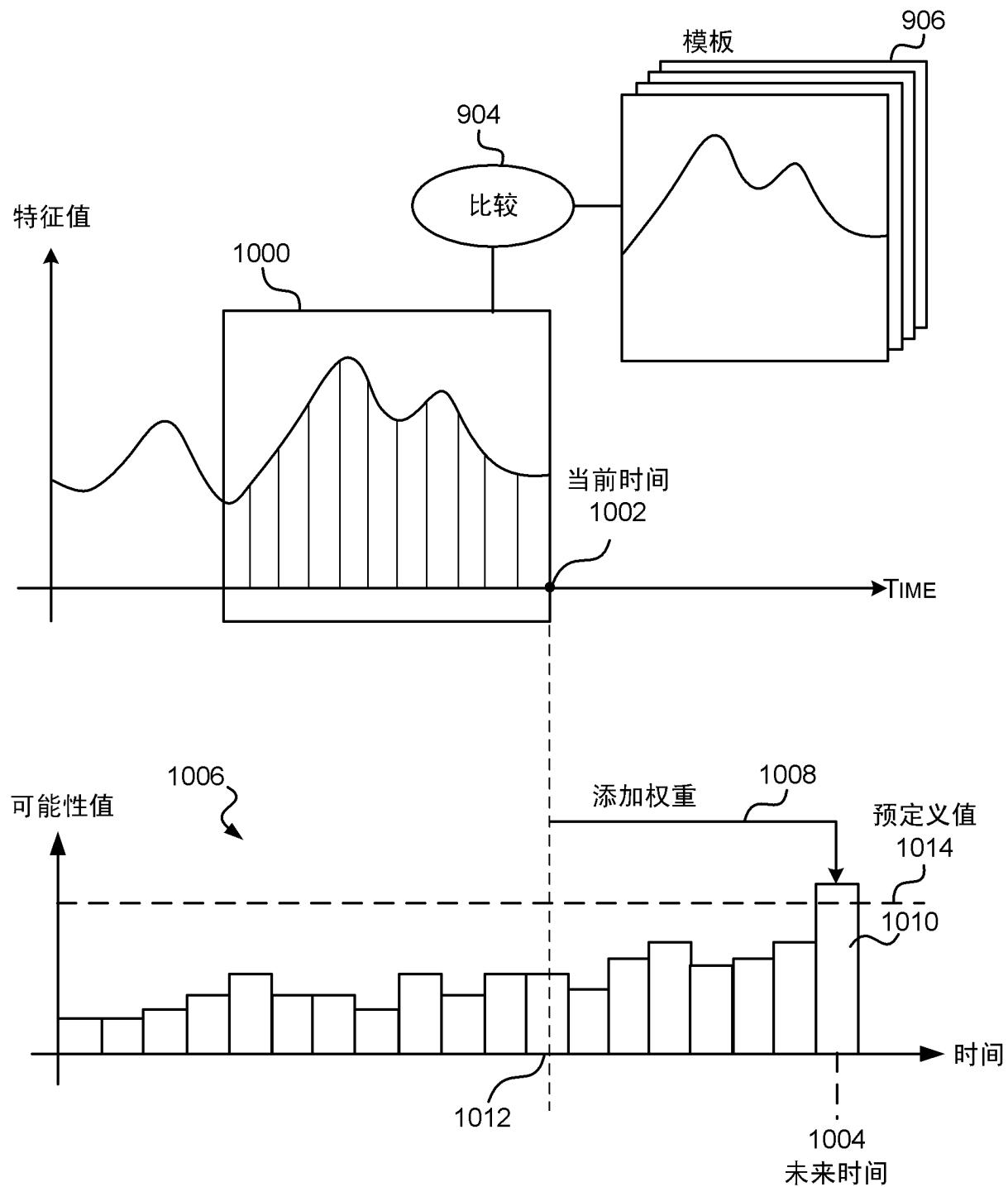


图 10

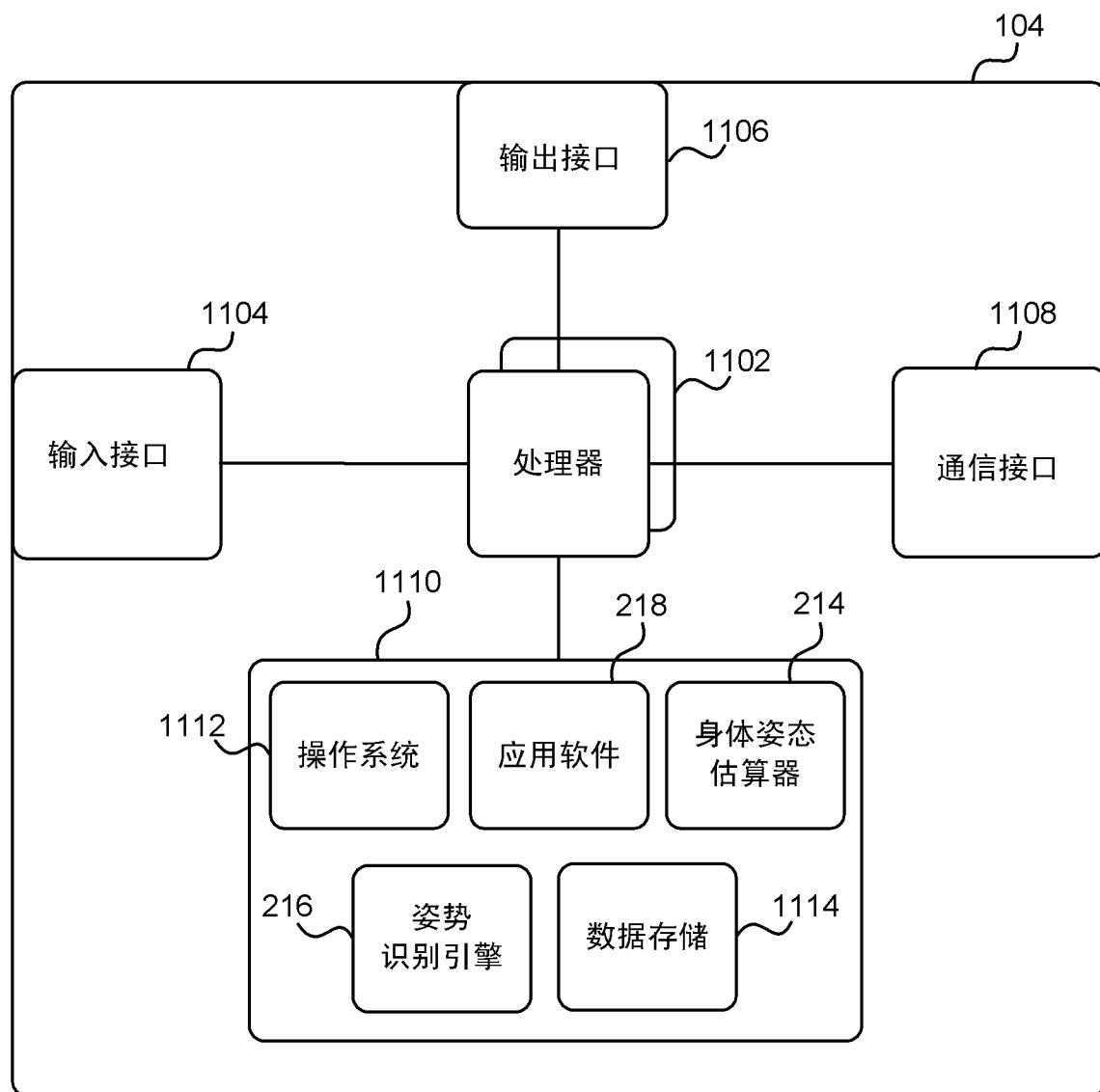


图 11