



# (12) 发明专利申请

(10) 申请公布号 CN 115516528 A

(43) 申请公布日 2022. 12. 23

(21) 申请号 202180033044.2

M·R·加布雷 P·马赫什瓦里

(22) 申请日 2021.05.04

(74) 专利代理机构 北京泛华伟业知识产权代理有限公司 11280

(30) 优先权数据

63/021087 2020.05.06 US

17/013446 2020.09.04 US

专利代理师 王勇

(85) PCT国际申请进入国家阶段日

2022.11.03

(51) Int.Cl.

G06V 20/56 (2022.01)

(86) PCT国际申请的申请数据

PCT/US2021/030580 2021.05.04

(87) PCT国际申请的公布数据

W02021/226027 EN 2021.11.11

(71) 申请人 卢米诺有限责任公司

地址 美国佛罗里达州

(72) 发明人 V·R·拉梅扎尼 A·兰格什

B·恩格拉德 S·S·马特拉

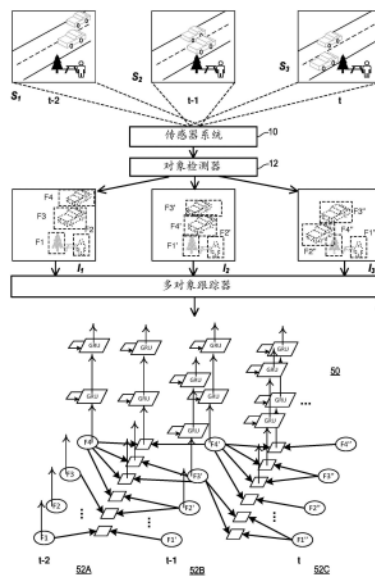
权利要求书2页 说明书19页 附图14页

(54) 发明名称

用于多对象检测与跟踪的图神经网络

(57) 摘要

一种用于多对象跟踪的方法包括:接收由一个或多个传感器在相应时间生成的图像序列,该一个或多个传感器被配置为感测对象相对于一个或多个传感器移动通过的环境;以及构造消息传递图,其中多层中的每一层对应于图像序列中相应的一个图像。该方法还包括通过图像序列跟踪多个特征,包括通过消息传递图在向前方向和向后方向中传递消息以跨时间共享信息。



1. 一种多对象跟踪方法,所述方法包括:

由处理硬件接收由一个或多个传感器在相应时间生成的图像序列,所述一个或多个传感器被配置为感测对象相对于所述一个或多个传感器移动通过的环境;

由所述处理硬件构造消息传递图,其中多层中的每一层对应于所述图像序列中相应的一个图像,所述构造包括:

针对所述层中的每一层生成多个特征节点,以表示在对应图像中检测到的特征,以及生成跨图神经网络的相邻层将所述特征节点中的至少一些特征节点进行互连的边,以表示所述特征之间的关联;以及

由所述处理硬件通过所述图像序列跟踪多个特征,包括通过所述消息传递图在向前方向和向后方向中传递消息以跨时间共享信息。

2. 根据权利要求1所述的方法,其中,构造所述消息传递图进一步包括:

生成与相应边相关联的边节点,每个节点恰好连接到第一层中的一个特征节点和第二层中的一个特征节点,所述第一层紧接在所述第二层前面。

3. 根据权利要求2所述的方法,包括将所述消息从特征节点传递到边节点。

4. 根据权利要求2所述的方法,其中,所述跟踪包括:

通过所述消息传递图的所述多层识别多个轨迹,其中,每个轨迹是边节点和表示同一对象的相关联特征的特征节点之间的连接序列。

5. 根据权利要求2所述的方法,其中,构造所述消息传递图进一步包括:

针对所述特征节点中的每一个特征节点生成至少一个相应的记忆单元,所述记忆单元被配置为输出对应特征的检测正确的概率;以及

针对所述边节点中的每一个边节点生成至少一个相应的记忆单元,所述记忆单元被配置为输出两个对应特征节点之间的连接正确的概率。

6. 根据权利要求5所述的方法,其中,所述记忆单元被实现为长短期记忆单元(LSTM)。

7. 根据权利要求5所述的方法,其中,所述记忆单元被实现为门控递归单元(GRU)。

8. 根据权利要求1所述的方法,其中,所述跟踪包括:

将所述消息的所述传递仅限制在当前处于有限大小的滚动窗口内的那些层。

9. 根据权利要求8所述的方法,其中,所述跟踪进一步包括:

响应于基于新的图像生成所述消息传递图的新的层,在所述向前方向中推进所述滚动窗口。

10. 根据权利要求9所述的方法,其中,所述跟踪进一步包括:

响应于推进所述滚动窗口经过一个层:(i) 固定该层的参数,以及(ii) 排除对该层的任何进一步更改。

11. 根据权利要求8所述的方法,其中,所述滚动窗口的大小以层数测量在3和10之间。

12. 根据权利要求1所述的方法,进一步包括:

训练由所述消息传递图支持的神经网络,所述训练包括:

在所述向前方向中通过所述消息传递图执行传递,包括计算总交叉熵损失;以及在所述向后方向中通过所述消息传递图执行传递,包括基于计算得到的总交叉熵损失更新所述边节点中的至少一些边节点。

13. 根据权利要求12所述的方法,其中,在所述向前方向中执行传递包括基于检测添加

新的特征节点和新的边节点。

14. 根据权利要求12所述的方法,其中,在所述向前方向中执行传递包括修剪低概率特征节点和边节点。

15. 根据权利要求1所述的方法,进一步包括:

使用由所述消息传递图支持的神经网络生成推断,所述生成包括:

在所述向前方向中通过所述消息传递图执行传递以生成概率,以及

使用已生成的概率通过所述消息传递图产生一个或多个轨迹。

16. 根据权利要求1所述的方法,其中,所述构造进一步包括针对每一个特征节点:

使用对象检测器,生成用于对应特征的检测的特征向量;

使用所述特征向量初始化所述特征节点的隐藏状态;以及

执行由所述消息传递图支持的神经网络的端到端训练,以联合优化对象检测和对象跟踪。

17. 根据权利要求1所述的方法,其中,接收所述图像序列包括从LIDAR或相机中的至少一个接收传感器数据。

18. 根据权利要求17所述的方法,其中,所述多个特征中的每一个特征表示像素簇。

19. 一种非暂态计算机可读介质,在其上存储有能够由一个或多个处理器执行的指令,以实现用于多对象跟踪的多对象跟踪架构,所述多对象跟踪架构被配置为:

接收由一个或多个传感器在相应时间生成的图像序列,所述一个或多个传感器被配置为感测对象相对于所述一个或多个传感器移动通过的环境;

构造消息传递图,其中多层中的每一层对应于所述图像序列中相应的一个图像,所述构造包括:

针对所述层中的每一层生成多个特征节点,以表示在对应图像中检测到的特征,以及

生成跨图神经网络的相邻层将所述特征节点中的至少一些特征节点进行互连的边,以表示所述特征之间的关联;以及

通过所述图像序列跟踪多个特征,包括通过所述消息传递图在向前方向和向后方向中传递消息以跨时间共享信息。

20. 根据权利要求19所述的非暂态计算机可读介质,其中,所述多跟踪架构被配置为通过以下方式构造所述消息传递图:

生成与相应边相关联的边节点,每个节点恰好连接到第一层中的一个特征节点和第二层中的一个特征节点,所述第一层紧接在所述第二层前面。

## 用于多对象检测与跟踪的图神经网络

[0001] 相关申请的交叉引用

[0002] 本申请要求于2020年5月6日提交的美国临时专利申请No.63/021,087和于2020年9月4日提交的美国专利申请No.17/013,446的优先权,出于所有目的,这些申请的公开内容将通过引用全部并入本文。

### 技术领域

[0003] 本公开总体上涉及多对象跟踪,并且更特别地,涉及使用消息传递图在图像序列中执行多对象跟踪。

### 背景技术

[0004] 本文提供的背景描述是为了总体上介绍本公开的上下文。目前列出姓名的发明者在本背景部分中描述的范围内的工作,以及在提交申请时可能不构成现有技术描述的方面,既不明确也不隐含地被承认为相对于本公开的现有技术。

[0005] 自动驾驶或“自主”车辆通常采用传感器,诸如光检测和测距(激光雷达)设备和/或传统相机,以在车辆向其目的地移动时检测或“查看”周围环境。此类车辆包括控制系统,该控制系统处理传感器数据,并基于感测到的环境和期望的目的地,确定哪些操纵和操作参数(例如,速度、制动力、转向方向)在整个行程中在或多或少的连续情况下最合适。自主车辆不仅要到达期望的目的地,还要保证自主车辆乘客和可能在自主车辆附近的任何个人的安全。

[0006] 自主车辆的控制架构可以实现感知功能,以识别各种对象,诸如其它车辆、行人、交通标志和信号、车道和道路边界等,以及预测功能,以确定对象相对于自主车辆可能的轨迹。为了有效地实现感知和预测,控制架构必须跨激光雷达数据的不同帧或者更一般地跨传感器在不同时间收集的图像来跟踪多个对象。然而,高效和准确的多跟踪仍然是难以解决的问题。

### 发明内容

[0007] 本公开的系统基于在不同时间收集的传感器数据(诸如照片或激光雷达影像帧)跟踪多个对象。为此,系统使用消息传递图,其中信息可以在向前方向和向后方向二者中行进。

[0008] 更具体地,在一个实施例中,用于多对象跟踪的方法包括:由处理硬件接收由一个或多个传感器在相应时间生成的图像序列,该一个或多个传感器被配置为感测对象相对于一个或多个传感器移动通过的环境。该方法还包括由处理硬件构造消息传递图,其中多层中的每一层对应于图像序列中相应的一个图像,该构造包括:针对层中的每一层生成多个特征节点,以表示在对应图像中检测到的特征,以及生成跨图神经网络的相邻层将特征节点中的至少一些特征节点进行互连的边,以表示特征之间的关联。该方法还包括由处理硬件通过图像序列跟踪多个特征,包括通过消息传递图在向前方向和向后方向中传递消息以

跨时间共享信息。

[0009] 在另一个实施例中,非暂态计算机可读介质在其上存储由一个或多个处理器可执行的指令,以实现多对象跟踪架构。多对象跟踪架构被配置为接收由一个或多个传感器在相应时间生成的图像序列,该一个或多个传感器被配置为感测对象相对于一个或多个传感器移动通过的环境。该多对象跟踪架构还被配置为通过如下方式构造消息传递图,其中多层中的每一层对应于图像序列中相应的一个图像:针对层中的每一层生成多个特征节点,以表示在对应图像中检测到的特征,以及生成跨图神经网络的相邻层将特征节点中的至少一些特征节点进行互连的边,以表示特征之间的关联。该多对象跟踪架构被配置为通过图像序列跟踪多个特征,包括通过消息传递图在向前方向和向后方向中传递消息以跨时间共享信息。

## 附图说明

[0010] 图1A示意性示出根据本公开的技术应用于图像序列的用于多对象跟踪的示例消息传递图的构造;

[0011] 图1B示意性示出基于图1A的消息传递图的多对象跟踪器产生的输出;

[0012] 图2A示出滚动窗口对消息传递图的应用;

[0013] 图2B示出可用于图1A和1B的消息传递图的示例节点和相邻边;

[0014] 图2C示出可用于图1A和1B的消息传递图的一条边和两个相邻节点的示例;

[0015] 图3是可在其中实现本公开的技术的自主车辆的示例软件架构的框图;

[0016] 图4A示出可以在其中实现本公开的技术的示例自主车辆;

[0017] 图4B示出可以在其中实现本公开的技术的另一个示例车辆;

[0018] 图5A示出自主车辆行进方向中的示例环境;

[0019] 图5B示出在自主车辆中操作的传感器系统可针对图5A的环境生成的示例点云;

[0020] 图6示出本公开的系统可以实现以跟踪多个对象的示例方法的流程图;

[0021] 图7示出用于构造本公开的消息传递图的示例方法的流程图;

[0022] 图8示出使用本公开的技术通过图像序列跟踪特征的示例方法的流程图;

[0023] 图9示出用于训练神经网络的示例方法的流程图,该神经网络包括本公开的消息传递图;以及

[0024] 图10示出用于使用本公开的消息传递图从图像序列生成推断的示例方法的流程图。

## 具体实施方式

### [0025] 概述

[0026] 一般来说,本公开的技术允许系统跨不同时间收集的传感器数据来有效地跟踪多个对象。激光雷达系统可以以特定的周期率扫描能视域,以产生例如点云形式的环境的图像序列,并且计算系统可以实现这些技术来跟踪在图像序列中观察到的对象。

[0027] 为此,多对象跟踪器可以构造一个图,其中对象的特征定义节点。特征可以是计算系统例如通过分割过程生成的特定大小(例如, $N \times M$ 像素)的点簇。作为另一个示例,特征可以通过进一步处理点云获得的降维值的向量。特征不需要具有相同的大小,并且通常可

以对应于图像的任何部分。因为如下文所讨论的，图也可以包括与边相关联的节点，所以表示特征的节点被称为“特征节点”。系统将在时间 $t$ 从传感器数据或图像中提取的特征进行关联，以定义层 $L$ 。

[0028] 多对象跟踪器可以经由边将设置在不同层中的特征进行互连，以定义关于特征跨图像序列的可能轨迹的假设。例如，计算机系统可以将层 $L_1$ 中的特征 $F_1$ 与相邻的时间较晚的层 $L_2$ 中的特征 $F_1'$ 以及层 $L_2$ 中的特征 $F_2'$ 、层 $L_2$ 中的特征 $F_3'$ 等进行互连。在系统使用本公开的技术训练由图支持的神经网络并生成推断后，系统为各种特征生成轨迹，以便确定例如层 $L_1$ 中的特征 $F_1$ 可能与层 $L_2$ 中的特征 $F_2'$ 相关联，而层 $L_2$ 中的特征 $F_2'$ 进而可能与层 $L_3$ 中的特征 $F_5'$ 相关联等。

[0029] 此外，除了特征节点之外，系统还可以生成边节点。一个边节点与一条边相关联，并恰好连接到一层中的一个特征和相邻层中的一个特征。因此，图包括经由边与边节点互连的特征节点的层。然后，系统可以根据图神经网络消息传播技术实现消息传递，使得图作为消息传递图（具有两类节点，即特征节点和边节点）而进行操作。更具体地，与特征节点相关联的函数可以在变量子集上进行操作来计算消息（例如，部分和），并将这些消息传递给边节点。类似地，特征节点可以从边节点接收消息。

[0030] 此外，该系统还可以将特征节点和边节点与相应的神经网络单元相关联，该神经网络单元被配置为针对特征节点产生对应特征的检测正确的概率；并且针对边节点，边表示的两个特征之间的连接正确的概率（如上面讨论的，每一个边节点恰好连接到两个特征节点：一个来自时间 $t$ ，并且另一个来自时间 $t+1$ ）。放置在每个特征和边节点处的这些神经网络单元例如可以是长短期记忆（LSTM）单元。如本领域公知的，LSTM单元由如下部分组成：单元格，其在特定的时间段内保留或“记住”值；以及输入门、输出门和遗忘门，以调节单元格的信息流进出。该系统可以将LSTM或其它合适的记忆单元进行链接，在“水平”消息传递图的顶部生成“竖直”堆栈。当边节点接收到来自相邻特征节点的消息时，边节点因此通过其对应的LSTM单元的竖直堆栈来传播该信息。从边节点接收消息的特征节点可以以类似的方式进行操作。

[0031] 在训练和推断操作期间，系统在向前方向中（从时间较早的层到时间较晚的层）和向后方向中（从时间较晚的层到时间较早的层）通过消息传递图传递消息。

[0032] 系统可以通过消息传递图来推进滚动窗口，从而限制消息传递的范围并考虑新的观察结果。例如，滚动窗口的大小可以为3，在这种情况下，系统在层 $L_i$ 、 $L_{i+1}$ 和 $L_{i+2}$ 上操作。在使用新的观察结果创建新的层 $L_i$ （并因此在层 $L_i$ 中创建一组新的特征）之后，系统推进窗口以在 $L_{i+1}$ 、 $L_{i+2}$ 和 $L_{i+3}$ 层上操作。该系统然后可以固定层 $L_i$ 的层神经网络参数，并排除其任何进一步的变化，因为滚动窗口不再包括该层。由于系统在之前的迭代中使用了现在固定的层 $L_i$ ，所以来自层 $L_i$ 的信息已经通过消息传递图向前传播。

[0033] 在一些实施方式中，系统使用上述用于构造消息传递图的技术来训练对象检测和跟踪这二者。为此，系统可以配置对象检测器，以针对每次检测输出特征向量，并使用这些向量来初始化特征节点。

[0034] 多对象跟踪场景中的示例消息传递图

[0035] 参考图1A，传感器系统10可以分别在时间 $t_1$ 、 $t_2$ 和 $t_3$ 捕获环境12的场景 $S_1$ 、 $S_2$ 、 $S_3$ ，并生成图像 $I_1$ 、 $I_2$ 、 $I_3$ 等的序列。传感器系统10可以包括能够感测环境的任何适当的设备组合，

诸如一个或多个激光雷达传感器(诸如图4A和4B中所示的那些激光雷达传感器)、一个或多个相机、一个或多个雷达、一个或多个热成像设备等。此外,传感器系统10可实现为图3中的传感器102。在一些实施方式中,传感器系统10生成图像 $I_1$ 、 $I_2$ 、 $I_3$ 等作为点云,其中每个点在3D空间中具有特定的坐标,并且在一些实施方式中,例如具有附加参数,诸如强度。在其它实施方式中,传感器系统10生成图像 $I_1$ 、 $I_2$ 、 $I_3$ 等,作为二维图像。

[0036] 在传感器系统10生成图像 $I_1$ 、 $I_2$ 、 $I_3$ 等后,对象检测器12可以检测到图像 $I_1$ 中的特征 $F_1$ 、 $F_2$ 、 $F_3$ 和 $F_4$ ,图像 $I_2$ 中的特征 $F'_1$ 、 $F'_2$ 、 $F'_3$ 和 $F'_4$ ,以及图像 $I_3$ 中的特征 $F''_1$ 、 $F''_2$ 、 $F''_3$ 和 $F''_4$ 。为了简单起见,图1A将这些特征示为包括整个对象,诸如小汽车、人或树,但一般来说,特征可以表示任何适当大小的对象的一部分。由于特征 $F_1$ 可能不一定对应于特征 $F_2$ ,因此多对象跟踪器14使用消息传递图50通过后续图像 $I_2$ 、 $I_3$ 等估计特征 $F_1$ 、 $F_2$ 、 $F_3$ 和 $F_4$ 的轨迹。对象检测器12和多对象跟踪器14可以实现为合适的自动驾驶软件架构的组件,诸如图3中描绘的示例架构100(特别地,对象检测器12和多对象跟踪器14可以作为跟踪模块114的组件进行操作)。

[0037] 如图1A中所示,消息传递图50针对在时间 $t_1$ 生成的图像 $I_1$ 生成层52A,针对在时间 $t_2$ 生成的图像 $I_2$ 生成层52B,以及针对在时间 $t_3$ 生成的图像 $I_3$ 生成层52C。通常,消息传递图50可以包括任何适当数量的层,每个层具有任何适当数量的特征(然而,如下面所讨论的,多对象跟踪器14可以对消息传递图50应用滚动窗口,以便限制在任一时间使用的层数)。层52A-C中的每一层都包括与对象检测器12在对应图像中识别的特征相对应的特征节点(由椭圆形表示)。特征节点经由边互连,在该示例实施方式中,该边还包括边节点(由矩形表示)。边节点恰好连接到两个特征节点:一个在时间较早的层 $L_N$ 中,并且另一个在时间较晚的层 $L_{N+1}$ 中。此外,特征节点和边节点中的每一个都包括可以作为LSTM实现的GRU的“堆栈”。

[0038] 一般来说,诸如LSTM和GRU的记忆单元“学习”和“遗忘”信息以便做出预测。特别地,记忆单元学习和保留重要的信息,同时丢弃(例如“遗忘”)不重要的信息。

[0039] 基于图50(或下面讨论的图70)的神经网络随着观察到新的检测和生成连接该检测的新的边节点而不断生长。多对象跟踪器14控制神经网络的增长是由(参考图2A讨论的)“滚动”或“滑动时间”窗口控制的,该窗口修剪旧的边和特征节点。窗口的大小是超参数(例如,其值在学习过程开始之前设定的参数)。在一些情况下,滑动时间窗口的大小在3到5个时间单位之间。

[0040] 在操作中,多对象跟踪器14可以通过图50传播消息,使得最上层的GRU产生指示例如图像 $I_1$ 的特征 $F_4$ 和图像 $I_2$ 的特征 $F'_3$ 之间以及图像 $I_2$ 的特征 $F'_3$ 和图像 $I_3$ 的特征 $F''_3$ 之间的最强(即最可能的)连接的值。换句话说,多对象跟踪器14基于图50训练神经网络,使得神经网络产生推断,根据该推断,特征 $F_4$ 、 $F'_3$ 和 $F''_3$ 可能表示同一物理对象或同一物理对象的一部分。

[0041] 参考图1B,多对象跟踪器14可在图50上进行操作以产生输出,根据该输出,层52A的特征 $F_4$ 可能对应于层52B的 $F'_3$ 和层52C的特征 $F''_3$ 。特别地,多对象跟踪器14通过图50在向前方向和向后方向中传播消息(分别对应于在时间上向前和向后),这使每个节点处的函数更新对应的值。本示例中的GRU 60-66生成高值,其对应于特征 $F_4$ 是与 $F'_3$ 相同的特征、而 $F'_3$ 进而是与 $F''_3$ 相同的特征的高概率。以类似的方式,并且在GRU 60-66生成输出(在图1B中示意性示出)的同时,多对象跟踪器14可以在不同层中的特征之间建立其它可能的关系,例如

$F_1 \rightarrow F'_1 \rightarrow F''_4$ 。

[0042] 如图2A中所示,多对象跟踪器14可以应用滚动窗口70,从而一次只考虑图70中的M层(例如,3、4、5层)。滚动窗口70跨越持续时间为T的间隔,它可以是任何适当固定的、可配置的或动态的值(例如,10ms、50ms、200ms、500ms)。在滚动窗口70推进经过某些层72之后,对象跟踪器14可以在这些时间较早的层72中确定特定的边(即,选择为可能正确的),识别假阳性(即,在特定图像中可能被错误识别并且不应该连接到其它层中的任何特征的特征)。为了简单起见,图70示出了不具有对应边节点的边。

[0043] 特别地,特征节点和 $d_1^0$ 、 $d_2^0$ 和 $d_3^0$ 可以被认为是连接到时间较晚的层中的其它预测的真阳性检测节点(例如, $d_3^0$ 到 $d_1^1$ )的预测的真阳性检测节点。预测的真阳性检测节点经由数据关联后生成的最终边80以成对的方式跨层连接。因此,多对象跟踪器14确定同一对象在时间t收集的影像中和在时间t+1收集的影像中的位置。

[0044] 另一方面,特征 $d_2^1$ 可被认为是预测的假阳性检测节点,因为多对象跟踪器14不能将该节点链接到时间较晚的层中的另一个节点,并且因此不能以足够的置信度识别时间较晚的影像中的特征。

[0045] 滚动窗口72内的特征 $d_1^t, d_2^t \dots d_N^{t+T}$ 可被认为是活动检测节点,其中活动检测节点中的至少一些活动检测节点由活动边82互连,该活动边是使用已学习的特征表示来生成的。与时间较早的层74中的节点不同,节点和边的值(例如,GRU输出)在这些节点处于滚动窗口72内时继续更改。因此,活动检测节点 $d_1^t, d_2^t \dots d_N^{t+T}$ 由多个活动边互连,直到多对象跟踪器14可消除大部分连接并选择最优连接作为最终边。此外,需要注意的是,如图2A中所示,在一些情况下,最终边80和检测边82可以将非相邻层中的节点进行互连。因此,在一些情况下,消息可以“跳过”一层或多层。

[0046] 如上面讨论的,多对象跟踪器14的一些实施方式使用(至少)两类节点:(i)边节点和(ii)特征节点(其有时可称为检测节点)。特征节点表示检测,而边节点表示特征的关联。在一些情况下,并且由于下面讨论的原因,每一个边节点连接到两个且仅两个特征节点:一个在时间上向后,而另一个在时间上向前。在另一种情况下,特征节点连接到所有的边节点。

[0047] 轨迹是沿相关联特征的唯一路径进行跟踪的边节点连接序列。边节点和特征节点可以遵循消息传递架构(例如,以“水平”方式),这允许邻居节点以排列不变的方式进行操作。多对象跟踪器14可以允许两组不同的排列不变操作:一组用于边,一组用于节点。

[0048] 回到图1A和1B的示例,在一些实施例中,除了“水平”维度外,还存在“竖直”维度。在这些实施例中的一些实施例中,存在附接到每一个边节点或特征节点的一组“竖直”长短期记忆单元(LSTM)(或其它类型的记忆单元,诸如门控递归单元(GRU)等),其附接到产生输出的全连接层:对于特征节点,输出是具有其特征的检测为假阳性或真阳性的概率;以及对于边节点,输出是表示附接到特定边节点的特征节点是否相连的连接性度量。因此,存在边和特征节点的“水平”网络和一组“竖直”记忆单元。

[0049] 如上面所指出的,图50或图70的每一层表示一个时间步,即对应于时间上的不同实例。数据关联可以被理解为图问题,其中时间上的观察结果之间的连接是一条边。多对象跟踪器14实现了神经网络,该神经网络从示例中学习如何建立这些连接,并进一步从示例中学习观察结果是真(例如,节点 $d_3^0$ 和 $d_1^1$ 之间的连接)还是假(例如,导致节点 $d_2^1$ 没有连接

到其它时间的任何其它节点,并且因此成为假阳性特征节点)。

[0050] 继续参考图2A,多对象跟踪器14通过移动滑动窗口72在多分图70上操作(例如,从左到右,如图2A中所示,从过去朝向未来的方向中)。多对象跟踪器14可以通过实现由以下伪代码(算法1)所示的算法来执行训练。在这些示例中,首字母缩写词“MPNN”代表消息传递神经网络(Message Passing Neural Networks),下面将详细讨论。

```

net ← TrackMPNN()                                ▷ 初始化 TrackMPNN 网络

G ← initialize_graph()                            ▷ 使用来自前两个时间步的检测来初始化图

total_loss ← 0                                    ▷ 将损失初始化为 0

(probabilities, loss) ← net.forward(G, labels)     ▷ 向前传递

total_loss ← total_loss + loss                    ▷ 加入总损失

for t ← 2 to tend do

    G ← update_graph(G, t)                        ▷ 将新的检测和边添加到图; 移除一些其他的检测和边

    (probabilities, loss) ← net.forward(G, labels) ▷ 向前传递

    total_loss ← total_loss + loss                ▷ 加入总损失

    if condition then

        G ← prune_graph(G, probabilities)         ▷ 移除低概率边和节点以限制存储器占用

    end if                                        ▷

end for                                          ▷

net ← net.backward(total_loss)                    ▷ 向后传递以训练网络

return net

```

[0052] (算法1:通过序列进行一次训练)

[0053] 为了执行推断,多对象跟踪器14可以实现由以下伪代码(算法2)所示的算法:

```

tracks ← {}                                       ▷ 将轨迹初始化为空

net ← TrackMPNN(trained_weights)                ▷ 使用已训练的权重来初始化 TrackMPNN 网络

[0054] G ← initialize_graph()                    ▷ 使用来自前两个时间步的检测来初始化图

probabilities ← net(G)                           ▷ 向前传递以获得概率

tracks ← tracks ∪ decode_graph(G, probabilities)

```

▷ 针对期望窗口解码网络概率以产生轨迹

**for**  $t \leftarrow 2$  **to**  $t_{end}$  **do**

$G \leftarrow \text{update\_graph}(G, t)$  ▷ 将新的检测和边添加到图；移除一些其他的检测和边

$\text{probabilities} \leftarrow \text{net}(G)$  ▷ 向前传递以获得概率

$\text{tracks} \leftarrow \text{tracks} \cup \text{decode\_graph}(G, \text{probabilities})$

[0055]

▷ 针对期望窗口解码网络概率以产生轨迹

**if** condition **then**

$G \leftarrow \text{prune\_graph}(G, \text{probabilities})$

▷ 移除低概率边和节点以限制存储器占用 **end if** ▷

**end for** ▷

**return**  $\text{tracks}$

[0056] (算法2:序列推断)

[0057] 算法利用了以下高级函数,解释如下:

[0058] `initialize graph()`:该函数使用来自两个连续的时间步的检测来创建初始二分图,来自两个连续的时间步的检测之间具有完全连接的边。

[0059] `update graph()`:该函数在每个时间步之后被调用,以将新的节点(检测)和对应的边添加到图的当前活动部分(例如,如下面讨论的,在滑动时间窗口内的图的部分)的末尾,并从图的当前活动部分固定最古老的一组节点和边的参数并且排除对该最古老的一组节点和边的进一步更改,因为滑动时间窗口不再包括它们的层。这实际上将滑动时间窗口向前移动了一步。

[0060] `prune graph()`:该函数使用用户指定的阈值从图的当前活动部分移除低概率的边和节点。每当存储器/计算需求超过允许(例如,超过存储器或处理资源的预定值)时可以调用该函数。

[0061] `decode graph()`:调用该函数将每个节点/边的输出概率解码为对应的轨迹。这既可以用贪婪的方式(例如,通过从左到右遵循最高概率路径)也可以通过使用匈牙利算法(在从左到右的连续的时间步上)来完成。

[0062] `TrackMPNN()`:该函数初始化建议的神经网络模型的实例。

[0063] `TrackMPNN.forward()`:该函数通过网络执行一次数据的向前传递。

[0064] `TrackMPNN.backward()`:该函数通过模型执行一次向后传递,以产生相对于输出的梯度。

[0065] 多对象跟踪器14的该示例方法的一些优点可以总结如下:(1)当关联检测对时考虑所有检测;(2)通过消息传递操作跨时间共享信息;(3)过去的错误可以被纠正,因为存在多个假设;(4)不需要手工制作成本函数,而是网络通过直接优化任务从数据中学习这一点;(5)该方法是可微的,并且因此可以端到端训练;(6)可以使用来自图神经网络(GNN)的

梯度来监督对象检测器,以生成更好的关联特征;以及(7)通过向网络添加新的输出头,可以潜在地合并跟踪器和预测器功能。

[0066] 下面的示例架构部分将进一步讨论多对象跟踪器14可以应用这些函数的示例模型。

[0067] 如上面所指出的,在一些情况下,多对象跟踪器14可以利用称为消息传递神经网络(MPNN)的一类GNN。MPNN可以实现消息函数,该消息函数根据示例公式进行操作:

$$[0068] \quad m_v^{t+1} = \sum_{w \in \mathcal{N}(v)} M_t(h_v^t, h_w^t, e_{vw}) \quad (\text{方程 1})$$

[0069] 根据如下公式进行操作的顶点更新函数:

$$[0070] \quad h_v^{t+1} = U_t(h_v^t, m_v^{t+1}), \quad \text{以及} \quad (\text{方程 2})$$

[0071] 根据如下公式进行操作的读出/输出函数(方程3)

[0072] 其中,在该上下文中(例如因子图),顶点可以表示节点(检测)或边(关联)。

[0073] 为了本目的,一些实施例针对节点和边使用单独的MPNN函数/权重。如上面所讨论的,节点表示检测特征,并且用其对应的特征向量进行初始化。边表示成对关联,并被初始化为与检测特征具有相同长度的零向量。网络的一种版本具有三层深度(分别是一个输入层、一个隐藏层和一个输出层);然而,该方法可以很容易地扩展到网络的多层版本。每一层具有其自己的消息和顶点更新函数,而输出层也具有读出函数。除了输出的维度,消息和顶点更新函数的函数形式对于所有层都是相同的。

[0074] 为了更详细地示出节点操作,在图2B中描绘了示例节点 $d_n$ (标签为90)及其相邻边92。节点90可以包括在消息传递图中,诸如上面讨论的图50或70。多对象检测器14可以针对节点 $d_n$ 实现几种操作,下面将考虑其中几种操作。

[0075] 多对象检测器14可以根据以下公式执行节点的初始化以表示检测:

$$[0076] \quad h_n^0 = \text{relu}(W_{node}^f f_n + b_{node}^f) \quad (\text{方程 4})$$

[0077] 因此,节点的隐藏状态由其输入特征表示 $f_n$ 的线性转换(随后是非线性)来初始化。

[0078] 此外,多对象检测器14可根据以下公式实现节点(检测)的消息和顶点更新函数:

$$[0079] \quad h_n^t = \text{GRU}(h_n^{t-1}, \sum_{e \in \mathcal{N}(n)} W_{node}^h h_e^{t-1} + b_{node}^h) \quad (\text{方程 5})$$

[0080] 因此,节点的隐藏状态是基于其之前的隐藏状态和其相邻边的之前的隐藏状态来更新的。

[0081] 此外,多对象检测器14可以根据以下公式实现节点(检测)的读出函数:

$$[0082] \quad o_n^t = \text{sigmoid}(W_{node}^o h_n^t + b_{node}^o) \quad (\text{方程 6})$$

[0083] 这里,节点输出 $o_n^t$ 表示标量概率值。每个节点输出都与二元交叉熵损失相关联。如果节点是真检测,则该节点具有阳性类(参见图2A和对应的讨论)。

[0084] 接下来,图2C示出示例边98将检测节点96A和96B进行互连的边操作。节点96A和96B以及边98可以在上面讨论的图50或70或任何其它合适的消息传递图中实现。多对象检测器14可以实现几种边操作,包括下面讨论的那些。

[0085] 首先,多对象检测器14可以根据以下公式实现表示关联的边的初始化:

[0086]  $h_e^0 = \mathbf{0}$  (方程 7)

[0087] 因此,边的隐藏状态用向量0来初始化。

[0088] 多对象检测器14可以根据以下公式针对表示关联的边实现消息和顶点更新函数:

[0089]  $h_e^t = GRU(h_e^{t-1}, W_{edge}^h (h_{n_1}^{t-1} \oplus h_{n_2}^{t-1}) + b_{edge}^h)$  (方程 8)

[0090] 因此,边的隐藏状态基于其之前的隐藏状态和其两个相邻节点的之前的隐藏状态来更新。

[0091] 为了实现表示关联的边的读出函数,多对象检测器14可以利用以下公式:

[0092]  $o_e^t = \text{sigmoid}(W_{edge}^o h_e^t + b_{edge}^o)$  (方程 9)

[0093] 节点输出  $o_e^t$  表示标量概率值。每个边输出都与二元交叉熵损失相关联。如果边连接了属于同一轨迹的节点,则该边具有阳性类。

[0094] 使用上面讨论的框架,多对象跟踪器14可以基于消息图50或70端到端训练神经网络。该方法由于联合优化,既可以改进检测功能,又可以改进跟踪功能。更具体地,多对象跟踪器14可以通过修改对象检测器针对每次检测输出(待学习的)特征向量来实现该改进,该特征向量用于初始化图中节点的隐藏状态,如上面提到的节点初始化方程  $h_n^0 = \text{relu}(W_{node}^f f_n + b_{node}^f)$  所示。这确保了训练期间从跟踪器到检测器的梯度流,其可用于学习更好的用于跟踪的特征。

[0095] 参考图3-5B,接下来讨论可以实现这些技术的示例感知系统和自主车辆。

[0096] 自动驾驶中用于感知和预测的示例架构

[0097] 图3示出示例的基于软件的传感器控制架构100,其用于动态控制一个或多个传感器102的一个或多个参数。传感器102可由自主车辆(例如,基于车辆的当前环境做出智能驾驶决策)或由用于其它目的(例如,收集与特定驾驶行程有关的数据)的非自主车辆利用。如本文所使用的术语,“自主”或“自动驾驶”车辆是一种被配置为感测其环境并在没有人类输入、很少人类输入、可选的人类输入和/或特定于环境的人类输入的情况下导航或驾驶的车辆。例如,自主车辆可被配置为在整个行程中驾驶到任何合适的位置并控制或执行所有安全关键功能(例如,驾驶、转向、制动、停车),而驾驶员不需要(或甚至不能)在任何时候控制车辆。作为另一个示例,自主车辆可以允许驾驶员在特定环境(例如高速公路上)和/或特定驾驶模式下,安全地将他或她的注意力从驾驶任务中转移开。

[0098] 自主车辆可被配置为在车辆中有驾驶员在场的情况下驾驶,或者被配置为在没有人类驾驶员在场的情况下驾驶。作为一个示例,自主车辆可包括具有相关联的控制装置(例如,方向盘、加速踏板和制动踏板)的驾驶员座位,并且车辆可被配置为在没有人坐在驾驶员座位上的情况下驾驶,或者在有限制的、有条件的或没有从坐在驾驶员座位上的人输入的情况下驾驶。作为另一个示例,自主车辆可能不包括任何驾驶员座位或相关联的驾驶员控制装置,车辆始终执行基本上所有的驾驶功能(例如,驾驶、转向、制动、停车和导航),而不需要人类输入(例如,车辆可被配置为在车辆中没有驾驶员在场的情况下运输人类乘客或货物)。作为另一个示例,自主车辆可被配置为在没有任何人类乘客的情况下进行操作(例如,车辆可被配置用于在没有车载的任何人类乘客的情况下运输货物)。

[0099] 如本文所使用的术语,“车辆”可以指被配置为运输人或货物的移动机器。例如,车

辆可包括,可采取如下形式,或可称为:小汽车、汽车、机动车辆、卡车、公共汽车、货车、拖车、越野车辆、农用车辆、割草机、建筑设备、高尔夫球车、房车、出租车、摩托车、踏板车、自行车、滑板、火车、雪地摩托、船只(例如,轮船或小船)、飞机(例如,固定翼飞机、直升机或飞船)或航天器。在特定实施例中,车辆可以包括为车辆提供推进力的内燃机或电动机。

[0100] 如图3中所示,车辆包括N个不同的传感器102,N是任何合适的整数(例如,1、2、3、5、10、20等)。传感器102中的至少“传感器1”被配置为通过以某种方式与环境进行物理交互来感测自主车辆的环境,该方式诸如发射和接收从环境中的对象反射的激光(例如,如果传感器是激光雷达设备),发射和接收从环境中的对象反射的声学信号(例如,如果传感器是无线电检测和测距(雷达)设备),简单地接收从环境的不同区域生成或反射的光波(例如,如果传感器是相机),等等。取决于实施例,所有传感器102可被配置为感测环境的部分,或者传感器102中的一个或多个传感器可以不与外部环境进行物理交互(例如,如果传感器102之一是惯性测量单元(IMU))。传感器102可以都是同一类型,或可包括多个不同的传感器类型(例如,具有不同观看视角的多个激光雷达设备,和/或激光雷达、相机、雷达和热成像设备的组合,等等)。

[0101] 由传感器102生成的数据被输入到传感器控制架构100的感知组件104,并由感知组件104处理以生成描述车辆环境当前状态的感知信号106。应理解,术语“当前”实际上可能是指生成任何给定感知信号106之前的一段很短的时间,例如由于感知组件104和其它因素引入的短处理延迟。为了生成感知信号106,感知组件104可包括分割模块110、分类模块112和跟踪模块114。

[0102] 分割模块110通常被配置为识别环境内的可区别对象,如由传感器数据(或传感器数据的一部分)所表示。取决于实施例和/或场景,可以针对多种不同类型的传感器数据中的每一种来单独执行分割任务(例如,分割模块110可包括并行操作的多个模块),或可以对多种类型的传感器数据的融合联合执行分割任务。在使用激光雷达设备的一些实施例中,分割模块110分析点云帧以识别每个帧内与环境中的可能的物理对象对应的点子集。在其它实施例中,分割模块110结合相机(和/或其它)图像帧来联合分析激光雷达点云帧,以识别环境中的对象。下面将进一步详细讨论激光雷达设备/系统和点云的示例。其它合适的技术和/或来自其它合适的传感器类型的数据也可用于识别对象。如在本文使用的,对不同或可区别“对象”的引用可以包括完全不相连的物理事物(例如,两个车辆是两个不同的“对象”),以及连接或部分连接的物理事物(例如,车辆是第一“对象”,而车辆的挂车是第二“对象”)。

[0103] 分割模块110可以使用预定的规则或算法来识别对象。例如,分割模块110可以在点云内将满足特定标准的任何点簇识别为可区别对象(例如,簇中所有点之间的距离不超过特定的最大距离,等等)。可替代地,分割模块110可以利用已训练的神经网络来识别环境内的可区别对象(例如,对测试数据点云内的不同对象使用监督学习,该监督学习使用手动生成的标签,等等),或利用另一种合适类型的基于机器学习的模型。对于感知组件104处理点云数据的实施例,在图5B中并在下面更详细地讨论了分割模块110的示例操作。

[0104] 分类模块112通常被配置为确定已由分割模块110识别的不同对象的类(标签、类别等)。与分割模块110一样,分类模块112可以对不同组的传感器数据分别执行分类(例如,分类模块112可包括并行操作的多个模块),或者可以基于来自多个传感器的数据的融合对

对象进行分类,等等。此外,并且同样类似于分割模块110,分类模块112可以执行预定的规则或算法对对象进行分类,使用已训练的神经网络对环境内已识别的对象进行分类(例如,对可区别对象的不同点云表示使用监督学习,该监督学习使用手动生成的标签,等等),或使用另一个合适的基于机器学习的模型对对象进行分类。对于感知组件104处理点云数据的实施例,在图5B中并在下面更详细地讨论了分类模块112的示例操作。

[0105] 跟踪模块114通常被配置为随时间推移跟踪可区别对象(例如,跨多个激光雷达点云或相机图像帧)。跟踪的对象通常是已经被分割模块110识别的对象,但可能是也可能不是被分类模块112分类的对象,这取决于实施例和/或场景。分割模块110可将标识符分配给已识别的对象,并且跟踪模块114可在适当的情况下将现有标识符与特定对象关联(例如,对于激光雷达数据,通过将相同标识符与连续点云帧中不同位置处的不同点簇关联)。与分割模块110和分类模块112一样,跟踪模块114可以基于不同组的传感器数据执行单独的对象跟踪(例如,跟踪模块114可以包括并行操作的多个模块),或者可以基于来自多个传感器的数据的融合来跟踪对象。此外,并且同样类似于分割模块110和分类模块112,跟踪模块114可以执行预定的规则或算法来跟踪对象,可以使用已训练的神经网络来跟踪环境内已识别(并且可能已分类)的对象(例如,针对不同的对或组的点云帧使用监督学习,该监督学习使用手动生成的标签,等等),或者使用另一个合适的机器学习模型来跟踪对象。

[0106] 因为图3的框图(以及本文描述的各种其它图)描绘了软件架构而不是物理组件,所以可以理解,当本文引用特定的神经网络或被“训练”的其它软件架构组件时,或者引用任何软件架构组件(例如,传感器102)在进行此类训练中的作用时,所描述的操作或过程可能发生在不同的计算系统上(例如,使用专门的开发软件)。因此,例如,分割模块110、分类模块112和/或跟踪模块114的神经网络在被实现到任何车辆内之前,可能已经在不同的计算机系统上进行了训练。换句话说,传感器控制架构100的组件可包括在特定车辆内的“最终”产品中,而该车辆或其物理组件(传感器102等)不必用于任何训练过程。

[0107] 传感器控制架构100还包括预测组件120,该预测组件120处理感知信号106以生成预测信号122,该预测信号122描述车辆环境的一种或多种预测的未来状态。例如,对于给定的对象,预测组件120可以分析对象的类型/类(如由分类模块112确定)以及该对象的最近跟踪运动(如由跟踪模块114确定),以预测该对象的一个或多个未来位置。作为相对简单的示例,预测组件120可以假设任何移动的对象将继续行进,而不改变其当前的方向和速度,可能考虑到一阶或高阶导数,以更好地跟踪持续改变方向的对象、正在加速的对象等等。在一些实施例中,预测组件120还基于更复杂的行为来预测对象的运动。例如,预测组件120可假设已被分类为另一车辆的对象将遵循道路规则(例如,在接近红灯时停止),并将以特定的方式对其它动态对象做出反应(例如,尝试与其它车辆保持一定的安全距离)。例如,通过利用神经网络或其它合适的机器学习模型,预测组件120可以固有地解释此类行为。在一些实施例中,可以从传感器控制架构100中省略预测组件120(例如,如果车辆不执行对未来环境状态的任何预测,或者如果车辆执行预测但预测的环境状态未用于控制任何传感器)。

[0108] 在一些实施例中,感知信号106包括表示“占用网格”的数据(例如,每T毫秒一个网格),每个占用网格指示自主车辆环境俯视图内的对象位置(以及可能的对象边界、定向等)。在占用网格内,每个“单元格”(例如,像素)可能与由分类模块114确定的特定类相关联,对于未成功分类的某些像素,可能与“未知”类相关联。同样,对于由感知组件104生成的

每个此类网格,预测信号122可包括一个或多个“未来占用网格”,其指示在一个或多个未来时间的(例如,提前1、2和5秒)预测的对象位置、边界和/或定向。在其它实施例中,传感器控制架构100不生成或利用占用网格。

[0109] 感知信号106和(在一些实施例中的)预测信号122输入到传感器控制组件130,该传感器控制组件130对信号106、122进行处理以生成传感器控制信号132,该传感器控制信号控制传感器102中的至少一个传感器的一个或多个参数(包括“传感器1”的至少一个参数)。特别地,传感器控制组件130尝试基于车辆环境内“动态”对象的存在、位置和/或类型,来引导传感器102中的一个或多个传感器的聚焦。为此,传感器控制组件130包括与参数调节模块136通信的动态对象检测器134。动态对象检测器134可以基于分类模块112对这些对象的分类来检测动态对象的存在。例如,动态对象检测器134可以访问本地存储的与动态对象对应的类列表(例如,“车辆”、“骑车人”、“行人”、“鹿”等),并且如果分类模块112根据动态对象类之一标记了特定的对象,则可以将该对象标记为“动态”。

[0110] 可替代地或者附加地,动态对象检测器134可以基于观察到的对象的行为来检测动态对象的存在。例如,动态对象检测器134可以访问本地存储的一组规则或算法,该规则或算法确定由跟踪模块114正在跟踪的对象是否被标记为“动态”对象。作为更具体的示例,如果来自跟踪模块114(在感知信号106内)的数据指示,对象在任何方向中(或在任何横向中等等)比阈值速度(例如,每秒0.25米、每秒0.5米等)更快地移动,则动态对象检测器134可以将对象标记为“动态”。在一些实施例中,动态对象检测器134包括被训练(例如,使用手动标记的训练数据集)的神经网络,以基于感知信号106和/或预测信号122识别动态对象。

[0111] 在其它实施例中,在由感知组件104处理传感器数据(例如,基于来自传感器102中的一个或多个传感器的原始传感器数据)之前,动态对象检测器134基于来自传感器102中的一个或多个传感器的传感器数据来检测动态对象的存在。例如,动态对象检测器134可以应用一个或多个规则或算法,或使用机器学习模型,以直接识别来自传感器102中的一个传感器的点云帧内的动态对象102。实际上,在一些实施例中,这可被视为动态对象检测器134独立于感知组件104的操作实现了分割模块110、分类模块112和/或跟踪模块114的功能的更简单版本。例如,动态对象检测器134可以使用简单的启发法来识别可能大致对应于单个对象的点“群集”或“簇”(例如,当在三维空间中表示点时,在阈值距离内具有超过阈值数量的点),并确定群集或簇正在随时间推移而移动(例如,通过计算不同群集或簇的质心跨多帧的运动,并假设相邻帧中的簇的质心在满足特定标准时属于同一对象)。因此,如本文所使用的(除非上下文明确表示不同的含义),对“动态对象”的识别或检测的引用并不一定意味着感知组件104已经识别、分类或跟踪了该对象。

[0112] 参数调节模块136基于由动态对象检测器134检测到的动态对象,确定(传感器102中的)受控传感器的参数设置。特别地,参数调节模块136确定一个或多个参数的值,该参数设定受控传感器的聚焦区域。通常,受控参数是影响由特定传感器感知车辆环境的哪个区域/部分的参数。例如,参数调节模块136可以确定设定受控传感器的水平和/或竖直能视域(例如,能视域覆盖的方位角和/或仰角的范围)、能视域的中心(例如,通过机械移动整个传感器,或调节移动能视域的中心镜子)和/或由传感器产生的扫描线的空间分布的值。在一些实施例中,受控传感器参数不仅影响传感器的聚焦区域,而且还影响感测车辆环境的给定区域的方式。例如,参数调节模块136可以控制传感器的帧/刷新率、传感器的分辨率

(例如,每个点云帧的点数),等等。

[0113] 参数调节模块136可以基于一个或多个动态对象的当前位置(例如,使用感知信号106)和/或基于动态对象的预测/预期的位置(例如,使用预测信号122)来确定受控传感器的期望的聚焦区域。例如,参数调节模块136可以设定激光雷达设备参数,使得激光雷达设备的能视域以动态对象的当前位置为中心,并可能还对该对象进行“放大”(例如,通过减少水平和垂直能视域,而不必减少每个点云帧中的点数)。可替代地,参数调节模块136可以设定激光雷达设备参数,使得激光雷达设备的能视域以动态对象在不久的将来某个时间(例如,未来0.5秒,未来1秒等)的预期/预测位置为中心。

[0114] 参数调节模块136可以基于单个动态对象、基于动态对象的类和/或基于多个动态对象或动态对象类的高密度区域来确定期望的聚焦区域。例如,参数调节模块136可以将聚焦区域识别为包含行人或车辆的区域,包括至少两个行人或车辆的区域,包括任何种类的动态对象的最高密度的区域(与感测到的环境的其余部分相比),等等。参数调节模块136还可以设定聚焦区域(例如,水平和/或垂直能视域),使得检测到的动态对象完全包括在该区域内(例如,不排除或“切断”检测到的车辆的一部分,或检测到的行人身体的一部分,等等)。

[0115] 如上面提到的,在一些实施例中,通过改变传感器扫描线的空间分布而不是(或除此以外)改变传感器的能视域的中心或大小来设定传感器的聚焦区域。例如,激光雷达或雷达设备可以聚焦于能视域内的特定仰角范围——而不必改变能视域的大小——通过相对于覆盖能视域内其它仰角的扫描线密度,增加覆盖该仰角范围的扫描线密度。

[0116] 动态对象检测器134可以使用基于来自正被控制的传感器的数据生成的传感器数据,和/或使用基于来自传感器102中的一个或多个其它传感器的数据生成的传感器数据来检测动态对象的存在。例如,如果传感器控制组件130仅控制“传感器1”,则动态对象检测器134可以使用仅基于来自“传感器1”的数据生成的感知信号106,使用仅基于来自“传感器2”至“传感器N”中的任何一个或多个传感器的数据的感知信号106,或使用基于来自“传感器1”的数据和来自“传感器2”至“传感器N”中的任何一个或多个传感器的数据二者的感知信号106,来识别动态对象。因此,例如,具有环境的广角视角的相机可以用于确定激光雷达设备的较窄聚焦区域,或者激光雷达设备可以最初设定为具有相对大的能视域,并且然后被设定为聚焦(例如,将较小的关注场集中在)在较大能视域的特定部分中检测到的动态对象上,等等。

[0117] 具有激光雷达系统的示例车辆

[0118] 图4A示出具有激光雷达系统302的示例车辆300。激光雷达系统302包括多个传感器头312A-312D,每一个传感器头都配备了相应的激光器。可替代地,传感器头312A-D可以经由合适的激光传感器链路耦合到单个激光器。通常,激光传感器链路314中的每一个激光传感器链路可包括一个或多个光学链路和/或一个或多个电气链路。

[0119] 图4A中的传感器头312被定位或定向为提供大于30度的车辆周围环境的视角。更普遍地,具有多个传感器头的激光雷达系统可在车辆周围提供约30°、45°、60°、90°、120°、180°、270°或360°的水平能视域。传感器头312中的每一个可以附接到保险杠、挡泥板、格栅、侧板、扰流板、车顶、前照灯组件、尾灯组件、后视镜组件、引擎盖、后备箱、车窗或车辆的任何其它合适的部分,或者并入其中。

[0120] 在图4A的示例中,四个传感器头312定位于车辆的四个角处或附近(例如,传感器头312中的每一个传感器头可并入灯组件、侧板、保险杠或挡泥板),并且激光器310可位于车辆300内(例如,在后备箱内或附近)。四个传感器头312可以各自提供90°到120°的水平视域(FOR),并且四个传感器头312可以定向成使得它们一起提供车辆周围的完整360度视角。作为另一个示例,激光雷达系统302可包括定位于300车辆上或周围的六个传感器头312,其中传感器头312中的每个传感器头提供60°至90°的水平FOR。作为另一个示例,激光雷达系统302可包括八个传感器头312,并且传感器头312中的每个传感器头可提供45°至60°的水平FOR。作为又一个示例,激光雷达系统302可包括六个传感器头312,其中传感器头312中的每个传感器头提供70°的水平FOR,相邻FOR之间重叠约10°。作为另一个示例,激光雷达系统302可包括两个传感器头312,它们一起提供大于或等于30°的前向的水平FOR。

[0121] 来自传感器头312中每一个传感器头的的数据可以组合或缝合在一起,以生成覆盖车辆周围大于或等于30度的水平视角的点云。例如,激光器310可包括控制器或处理器,该控制器或处理器(例如,经由对应的电气链路320)从传感器头312中的每一个传感器头接收数据,并处理接收到的数据以构造覆盖车辆周围360度水平视角的点云,或确定与一个或多个目标的距离。可以经由对应的电气、光学或无线电链路320将点云或来自点云的信息提供给车辆控制器322。车辆控制器322可包括一个或多个CPU、GPU、以及具有持久性组件(例如闪存、光盘)和/或非持久性组件(例如RAM)的非暂态存储器。

[0122] 在一些实施方式中,点云是通过在激光器310内包括的控制器处组合来自多个传感器头312中的每一个传感器头的的数据来生成的,并提供给车辆控制器322。在其它实施方式中,传感器头312中的每一个传感器头包括控制器或处理器,该控制器或处理器构造车辆周围360度水平视角的一部分的点云,并将相应的点云提供给车辆控制器322。车辆控制器322然后将来自相应传感器头312的点云组合或缝合在一起,以构造覆盖360度水平视角的组合点云。此外,在一些实施方式中,车辆控制器304与远程服务器通信以处理点云数据。

[0123] 在任何情况下,车辆300可以是自主车辆,其中车辆控制器322向车辆300内的各个组件330提供控制信号,以操纵或以其它方式控制车辆350的操作。组件330在图4A中的展开视图中描绘,仅为便于说明。组件330可包括加速器340、制动器342、车辆发动机344、转向机构346、灯348(诸如刹车灯、前照灯、倒车灯、应急灯等)、挡位选择器350、IMU343、附加传感器345(例如相机、雷达、声学传感器、大气压力传感器、湿度传感器、环境光传感器,如下所示)和/或实现和控制车辆300的运动的其它合适组件。挡位选择器350可包括驻挡、倒车、空挡、驱动挡位等。组件330中的每一个组件可包括接口,该组件经由该接口接收来自车辆控制器322的命令,诸如“加速”、“减速”、“左转5度”、“激活左转信号”等,并在一些情况下向车辆控制器322提供反馈。

[0124] 车辆控制器322可以包括感知模块352,该感知模块352接收来自组件300的输入,并使用感知机器学习(ML)模型354向运动规划器356提供对检测到的对象、道路标记等的指示,该运动规划器356为组件330生成操纵车辆300的命令。返回参考图1,在一个实施例中的组件352-356在任何合适的配置中实现组件102-107。

[0125] 在一些实施方式中,车辆控制器322经由链路320从传感器头312接收点云数据,并使用本文描述的任何一种或多种技术分析接收到的点云数据。然后,车辆控制器322经由另一个链路320向组件330提供控制信号,以基于分析得到的信息来控制车辆的操作。

[0126] 除了激光雷达系统302之外,车辆300也可能配备其它传感器345,诸如相机、热成像仪、传统雷达(没有示出以避免混乱)等。传感器可以经由有线或无线通信链路向车辆控制器322提供附加数据。此外,在示例实施方式中的车辆300包括麦克风阵列,其作为被配置为确定声源的声源定位系统的一部分进行操作。

[0127] 作为另一个示例,图4B示出车辆360,其中有几个头372,每个头可以类似于图4A的传感器头312中的一个,并且位于后备箱上。特别地,传感器头372B和C相对于车辆360的定向朝后方,并且传感器头372A和D相对于车辆360的定向朝前方。例如,在另一个实施方式中,附加传感器被放置在侧视镜处。

[0128] 图5A描绘示例的真实世界驾驶环境380,并且图5B描绘示例的点云390,该点云是由扫描环境380的激光雷达系统生成的(例如,图2和图3的激光雷达系统200或图4A的激光雷达系统302)。如图5A中所示,环境380包括具有中间墙壁的高速公路,该中间墙壁将两个方向的交通分隔开,每个方向中都有多条车道。图5B中的点云390对应于一个示例实施例,其中两个激光雷达设备各自捕获大约60度的水平能视域,并且其中两个能视域具有小的重叠392(例如,2或3度的重叠)。例如,点云390可能是使用图4A中的传感器头312A和312D或者图4B中的传感器头372A和372G生成的。虽然在图5B中被描绘为视觉图像,但可以理解,在一些实施例中,点云390在任何时候实际上都不会被渲染或显示。

[0129] 如图5B中所示,点云390将地平面394(这里为道路表面)描绘为多条基本上连续的扫描线,并且还在地平面394之上描绘了多个对象396。为了清晰起见,只有图5B中所示的少数对象标记有附图标记。返回参考图3,可以通过分割模块110识别出点云390内对象396中的可区别对象。例如,分割模块110可以在地平面394的扫描线中检测到大量的间隙和/或其它不连续点,并将这些不连续点附近的点组识别为离散对象。分割模块110可以使用任何合适的规则、算法或模型来确定哪些点属于同一对象。一旦对象396被识别,图3的分类模块112就可以尝试对对象进行分类,并且图3的跟踪模块114可以尝试跨类似于点云390的未来点云(即跨多个点云帧)跟踪已分类的对象(以及,在一些实施例/场景中,未分类的对象)。

[0130] 由于各种原因,在点云390内,分割模块110识别特定的对象396和/或分类模块112对特定的对象396进行分类可能更加困难。例如,如图5B中所示,中间墙壁396A可能相对容易识别和分类,这是因为高的点密度以及墙壁396A造成的“阴影”(即点的缺失或相对稀缺)。由于高的点密度(可能还有其阴影的形状),卡车396B可能也相对容易被识别为对象,但由于卡车396B的大部分隐藏在激光雷达阴影内,可能不容易分类。车辆396C可能相对容易被识别为对象,但可能更难分类,由于中间墙壁396A(即沿着车辆396C的下部)造成的激光雷达阴影内缺乏点。车辆396D可能更难被识别为可区别对象,因为距离自主车辆较远的地方缺少点,以及与车辆396D对应的点与附近其它对象的点之间的距离很近。还有一些对象由于体积小和/或不显眼,可能难以识别、分类和/或跟踪。例如,虽然图5B中没有示出,但分割模块110可识别(并且分类模块112可以分类)点云390内的车道标记。车道标记可显现为扫描线路径中的微小但突然的偏差,例如,这些偏差共同形成了与车辆行进方向一致的线形(例如,近似垂直于扫描线的曲线)。

[0131] 尽管存在此类困难,分割模块110、分类模块112和/或跟踪模块114可以使用如下技术,该技术使对象识别、分类和/或跟踪在非常广泛的场景中高度准确,而该场景中的对象的点云表示很少或不太理想。例如,分割模块110、分类模块112和/或跟踪模块114可包括

使用与大量不同的环境/场景(例如,具有不同距离、不同定向、不同隐藏程度、不同天气和/或照明条件等等的各种类型的对象)对应的数据(例如,手动标记的场景)进行训练的神经网络。

[0132] 使用消息传递图的多对象跟踪示例方法

[0133] 为了更清晰,图6示出用于多对象跟踪的示例方法600。方法600可以实现为存储在计算机可读介质上并由一个或多个处理器或以另一种适当的方式执行的一组指令。方法600可以在上面讨论的多对象跟踪器14中实现,也可以在另一个合适的模块或多个模块中实现。为方便起见,图6参考多对象跟踪器14进行讨论。

[0134] 在框602处,多对象跟踪器14接收由一个或多个传感器生成的图像序列。各种实施方式中的传感器可以包括一个或多个激光雷达传感器、一个或多个相机(例如,CMOS相机)、一个或多个雷达,或者不同类型的传感器的组合(例如,如果使用传感器融合)。

[0135] 在框604处,多对象跟踪器14构造消息传递图,其中多层中的每一层对应于图像序列中的相应的图像。接下来,在框606处,多对象跟踪器14通过图像序列跟踪多个特征,包括通过消息传递图在向前方向和向后方向中传递消息以跨时间共享信息。

[0136] 接下来,图7示出用于构造消息传递图(诸如上面示例中的图50或图70)的示例方法700。与方法600类似,方法700可以实现为存储在计算机可读介质上并由一个或多个处理器执行并且例如可以在多对象跟踪器14或其它合适的模块中实现的一组指令。

[0137] 在框702处,多对象跟踪器14接收由一个或多个传感器生成的图像序列,类似于上面讨论的框602。

[0138] 在框710处,多对象跟踪器14生成消息传递图的层,使得多层中的每一层都对应于图像序列中的相应的图像。接下来,在框712处,对于消息传递图中的每一层,多对象跟踪器14生成多个特征节点,以表示在对应图像中检测到的特征。然后,在框714处,对于每一个特征节点,多对象跟踪器14:(i)使用对象检测器为对应特征的检测生成特征向量;(ii)使用特征向量初始化特征节点的隐藏状态;以及(iii)对由消息传递图支持的神经网络执行端到端训练,以联合优化对象检测和对象跟踪。

[0139] 在框716处,多对象跟踪器14生成边,该边跨消息传递图相邻层将特征节点中的至少一些特征节点进行互连,以表示特征之间的关联。然后,在框718处,多对象跟踪器14在消息传递图中生成边节点。边节点与相应的边相关联,并且每个节点恰好连接到第一层中的一个特征节点和第二层中的一个特征节点,第一层紧接在第二层前面。

[0140] 在框720处,对于特征节点和/或边节点中的每一个,多对象跟踪器14配置至少一个相应的记忆单元,以输出对应特征的检测正确的概率,记忆单元或者是长短期记忆单元(LSTM)或者是门控递归单元(GRU)。

[0141] 接下来,图8示出示例方法800,用于通过消息传递图(诸如上面示例中的图50或图70)构造轨迹,该方法也可以在多对象跟踪器14中实现。在框802处,多对象跟踪器14接收由一个或多个传感器生成的图像序列。在框804处,多对象跟踪器14构造消息传递图,其中多层中的每一层对应于图像序列中的相应的图像。

[0142] 在框822处,多对象跟踪器14使用有限大小的滚动窗口来限制消息只传递到当前滚动窗口内的那些层。在框824处,多对象跟踪器14响应于基于新的图像生成消息传递图的新层而在向前方向中推进滚动窗口。

[0143] 在框826处,响应于将滚动窗口推进经过一层,多对象跟踪器14(i)固定该层的参数,以及(ii)排除对该层的任何进一步更改。在框828处,多对象跟踪器14识别消息传递图内的多个轨迹,其中每个轨迹是边节点和表示同一特征的特征节点之间的连接序列。

[0144] 图9示出用于训练神经网络的方法900,该神经网络也可在多对象跟踪器14中实现。

[0145] 在框902处,多对象跟踪器14接收由一个或多个传感器生成的图像序列。然后,多对象跟踪器14构造消息传递图,其中在框906处多层中的每一层对应于图像序列中的一个相应的图像。在框930处,多对象跟踪器14在消息传递图中生成特征节点和边节点,类似于上面的示例。

[0146] 在框930处,多对象跟踪器14训练由消息传递图支持的神经网络。为此,多对象跟踪器14可以在向前方向中通过图执行传递,包括:(i)计算总交叉熵损失;(ii)基于检测添加新的特征节点和新的边节点,以及(iii)修剪低概率特征节点和边节点。在框932处,多对象跟踪器14通过在向后方向中执行传递(包括基于计算得到的总交叉熵损失更新边节点中的至少一些边节点)来进一步训练由消息传递图支持的神经网络。

[0147] 最后,图10示出从图像序列生成推断的示例方法1000,该方法也可以在多对象跟踪器14中实现。

[0148] 在框1002处,多对象跟踪器14接收由一个或多个传感器生成的图像序列。在框1004处,多对象跟踪器14构造消息传递图,其中多层中的每一层对应于图像序列中的相应的图像。在框1028处,多对象跟踪器14通过图像序列跟踪多个特征,包括通过消息传递图在向前方向和向后方向中传递消息以跨时间共享信息,如上面所讨论的。

[0149] 在框1040处,多对象跟踪器14使用由消息传递图支持的神经网络通过(i)在向前方向中通过消息传递图执行传递以生成概率,以及(ii)使用已生成的概率通过消息传递图产生一个或多个轨迹来生成推断。

#### [0150] 一般注意事项

[0151] 在一些情况下,计算设备可用于实现本文公开的各种模块、电路、系统、方法或算法步骤。作为示例,本文公开的模块、电路、系统、方法或算法的全部或部分可以由通用单芯片或多芯片处理器、数字信号处理器(DSP)、ASIC、FPGA、任何其它合适的可编程逻辑器件、分立的门或晶体管逻辑、分立的硬件组件、或其任何合适的组合实现或执行。通用处理器可以是微处理器,或者任何传统的处理器、控制器、微控制器或状态机。处理器还可以实现为计算设备的组合(例如DSP和微处理器的组合)、多个微处理器、与DSP核结合的一个或多个微处理器、或者任何其它此类配置。

[0152] 在特定实施例中,本文描述的主题的一个或多个实施方式可以被实现为一个或多个计算机程序(例如,编码或存储在计算机可读非暂态存储介质上的计算机程序指令的一个或多个模块)。作为示例,本文公开的方法或算法的步骤可以在可以驻留在计算机可读非暂态存储介质上的处理器可执行软件模块中实现。在特定实施例中,计算机可读非暂态存储介质可包括可用于存储或传输计算机软件并且可由计算机系统访问的任何合适的存储介质。在此,计算机可读非暂态存储介质或媒介可以包括一个或多个基于半导体的或其它集成电路(IC)(诸如例如,现场可编程门阵列(FPGA)或专用IC(ASIC))、硬盘驱动器(HDD)、混合硬盘驱动器(HHD)、光盘(例如,压缩光盘(CD)、CD-ROM、数字多功能光盘(DVD)、蓝光光

盘或激光光盘)、光盘驱动器(ODD)、磁光盘、磁光驱动器、软盘、软盘驱动器(FDD)、磁带、闪存、固态驱动器(SSD)、RAM、RAM驱动器、ROM、安全数字卡或驱动器、任何其它合适的计算机可读非暂态存储介质、或这些中的两个或更多的任何合适组合(在适当情况下)。在适当的情况下,计算机可读非暂态存储介质可以是易失性、非易失性或易失性和非易失性的组合。

[0153] 在一些情况下,本文在单独实施方式的上下文中描述的某些特征也可以组合并在单个实施方式中实现。相反,在单个实施方式的上下文中描述的各种特征也可以单独地或以任何合适的子组合在多个实施方式中实现。此外,虽然特征可以在上面描述为以某些组合起作用并且本身甚至最初要求保护,但在一些情况下,要求保护的组合中的一个或多个特征可以从组合中删除,并且要求保护的组合可以针对子组合或子组合的变体。

[0154] 虽然在附图中可以将操作描述为以特定顺序发生,但这不应被理解为要求以所示的特定次序或顺序执行此类操作,或者要求执行所有操作。此外,附图可以以流程图或序列图的形式示意性地描绘一个或多个示例过程或方法。然而,未描绘的其它操作可并入示意性示出的示例过程或方法中。例如,一个或多个附加操作可以在任何所示操作之前、之后、同时或之间执行。此外,在适当的情况下,可以重复图中描绘的一个或多个操作。此外,图中描绘的操作可以以任何合适的顺序执行。此外,虽然特定组件、设备或系统在本文中被描述为执行特定操作,但是任何合适的组件、设备或系统的任何合适的组合可用于执行任何合适的操作或操作的组合。在某些情况下,可以执行多任务或并行处理操作。此外,本文描述的实施方式中各个系统部件的分离不应理解为在所有实施方式中都需要这种分离,并且应该理解,所描述的程序部件和系统可以一起集成在单个软件产品中或打包成多个软件产品。

[0155] 已经结合附图描述了各种实现方式。然而,应当理解,这些图不一定按比例绘制。作为示例,图中描绘的距离或角度是说明性的并且可能不一定与所示设备的实际尺寸或布局具有精确关系。

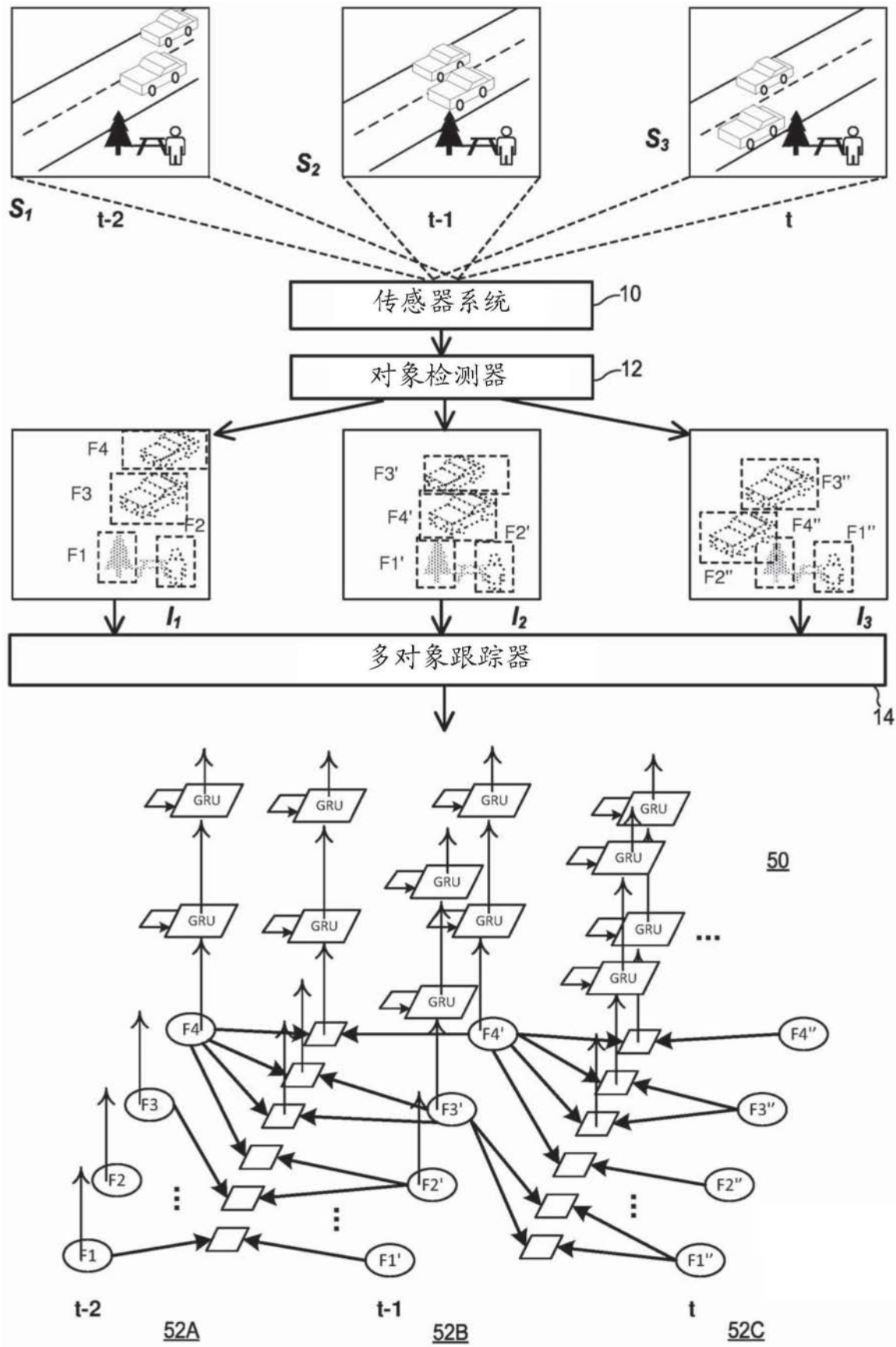
[0156] 本公开的范围包括本领域普通技术人员将理解的对在此描述或示出的示例实施例的所有改变、替换、变化、变更和修改。本公开的范围不限于本文描述或示出的示例实施例。此外,尽管本公开将本文中的各个实施例描述或示出为包括特定组件、元件、功能、操作或步骤,但是这些实施例中的任何一个实施例可包括本领域普通技术人员将理解的本文任何地方描述或示出的任何组件、元件、功能、操作或步骤的任何组合或排列。

[0157] 如本文所使用的,术语“或”应被解释为包括或意指任何一个或任何组合,除非另有明确说明或上下文另有说明。因此,在本文中,表述“A或B”是指“A、B或A和B二者”。作为另一示例,在本文中,“A、B或C”是指以下中的至少一个:A;B;C;A和B;A和C;B和C;A、B和C。如果元素、设备、步骤或操作的组合在某些方面本质上是相互排斥的,则会出现此定义的例外。

[0158] 如本文所使用的,诸如但不限于“约”、“基本上”或“大约”的近似词是指这样一种条件,当如此修改时被理解为不一定是绝对的或完美的,但将被认为对本领域普通技术人员而言足够接近,以保证将所述条件指定为存在。描述可以变化的程度将取决于可以进行多大的改变,并且本领域的普通技术人员仍然认识到修改的特征具有未修改的特征所需的特性或能力。一般而言,但以前述讨论为前提,本文中通过诸如“约”的近似词修饰的数值可能与所述值相差 $\pm 0.5\%$ 、 $\pm 1\%$ 、 $\pm 2\%$ 、 $\pm 3\%$ 、 $\pm 4\%$ 、 $\pm 5\%$ 、 $\pm 10\%$ 、 $\pm 12\%$ 或 $\pm 15\%$ 。

[0159] 如本文所使用的,术语“第一”、“第二”、“第三”等可用作它们前面的名词的标签,并且这些术语可能不一定暗示特定排序(例如,特定空间、时间或逻辑顺序)。作为示例,系统可以被描述为确定“第一结果”和“第二结果”,并且术语“第一”和“第二”可能不一定暗示在第二结果之前确定第一结果。

[0160] 如本文所使用的,术语“基于”和“至少部分地基于”可用于描述或呈现影响确定的一个或多个因素,并且这些术语可能不排除可能影响确定的附加因素。确定可以仅基于所呈现的那些因素或者可以至少部分地基于那些因素。短语“基于B确定A”表示B是影响A的确定的因素。在一些情况下,其它因素也可能有助于确定A。在其它情况下,A可能仅基于B确定。



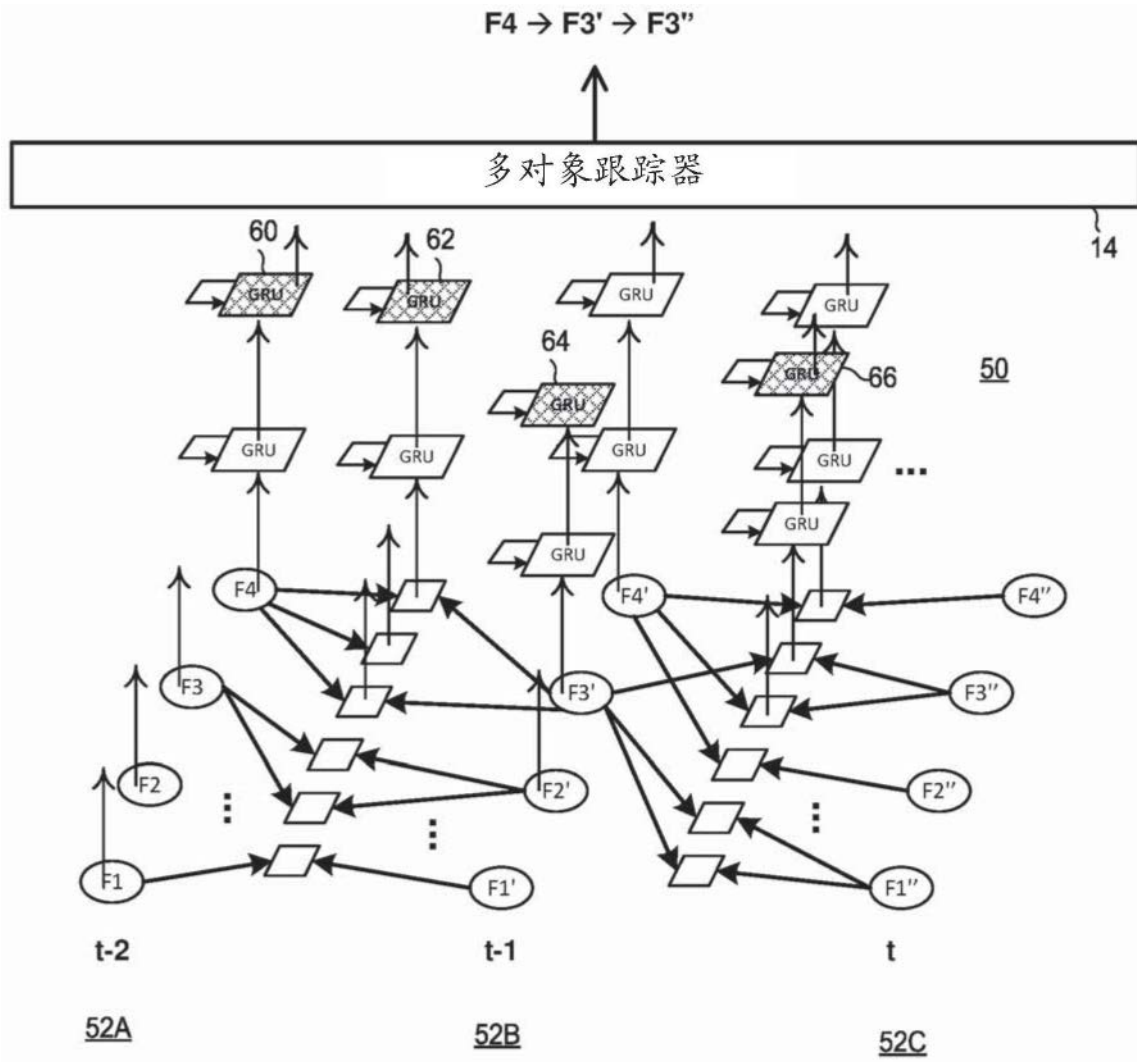


图1B



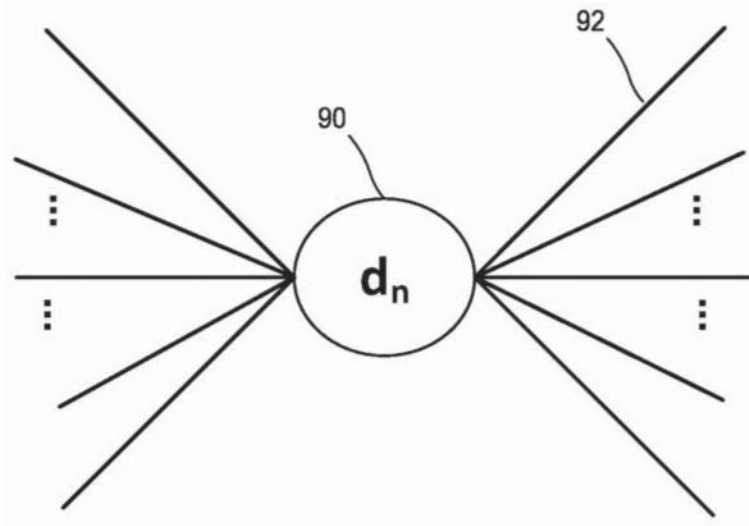


图2B

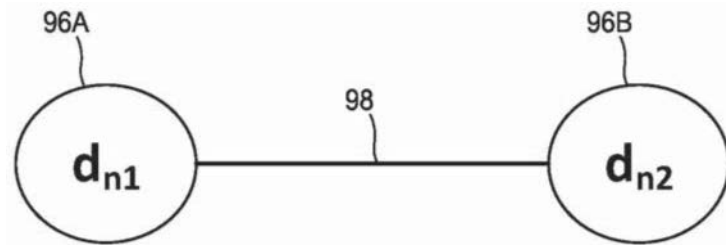


图2C

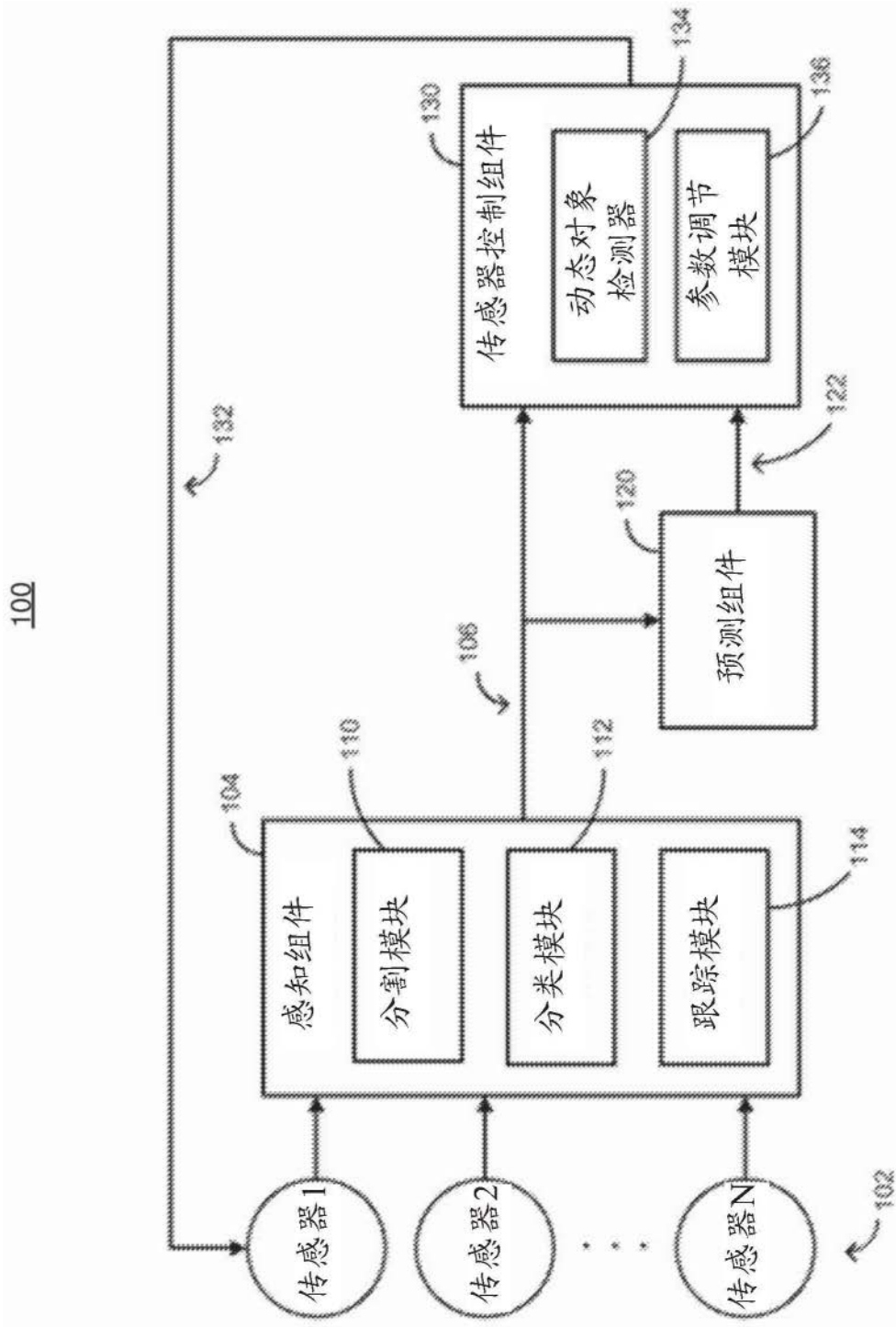


图3

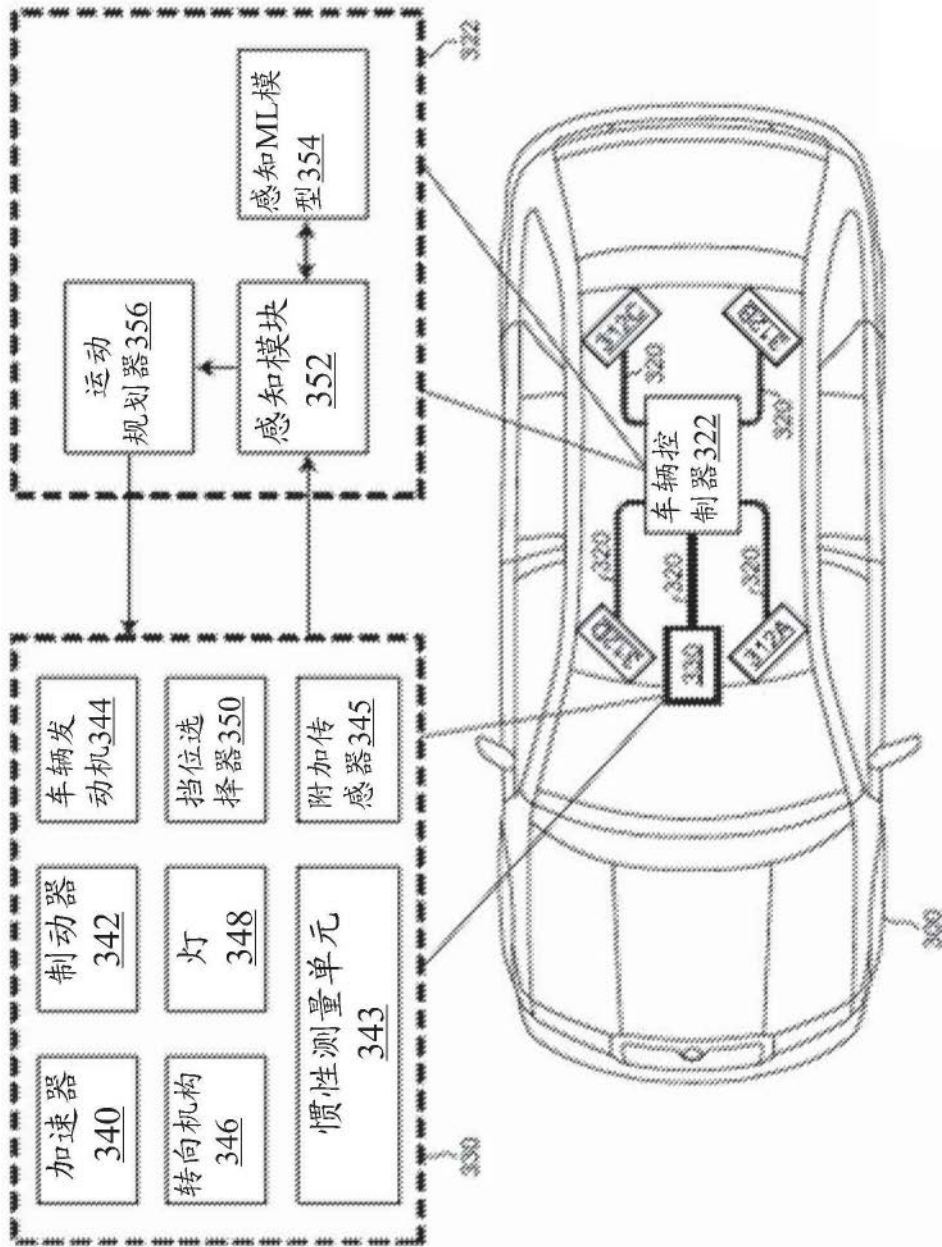


图4A

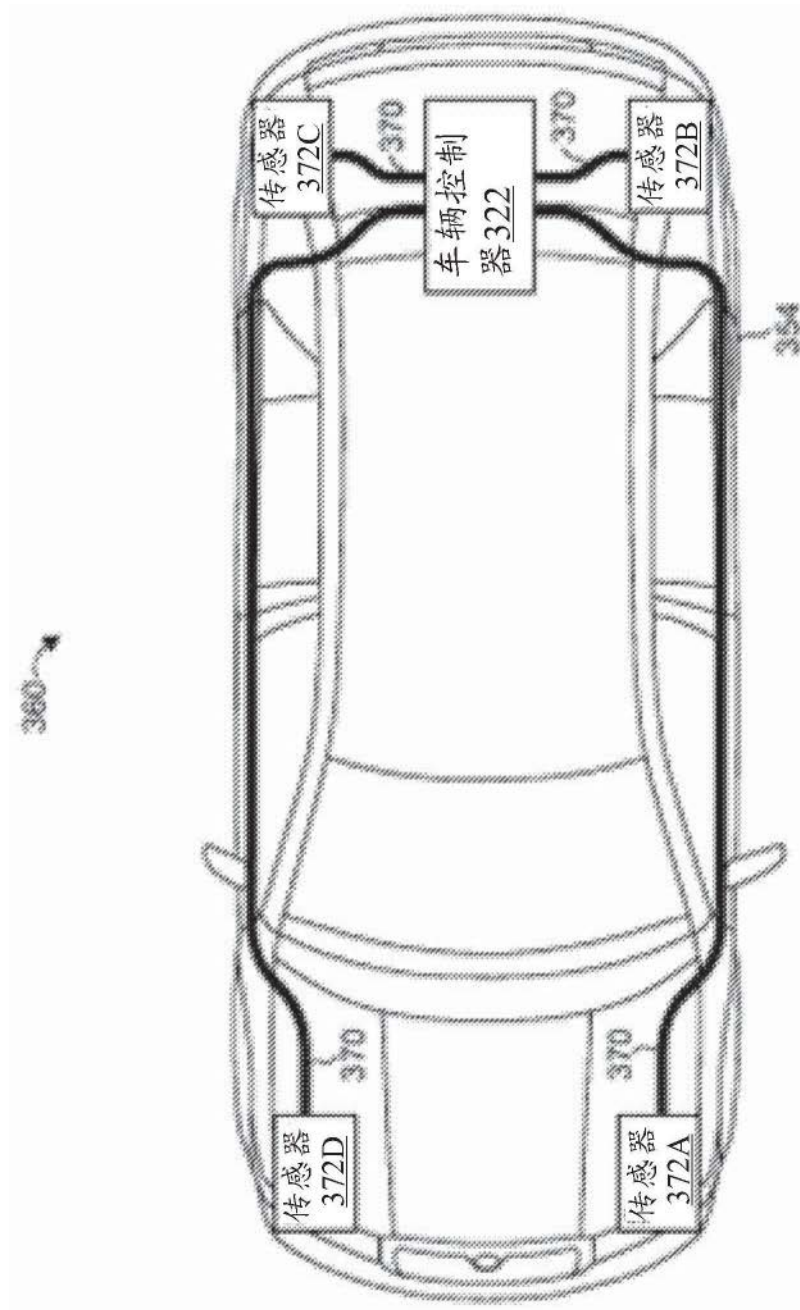


图4B

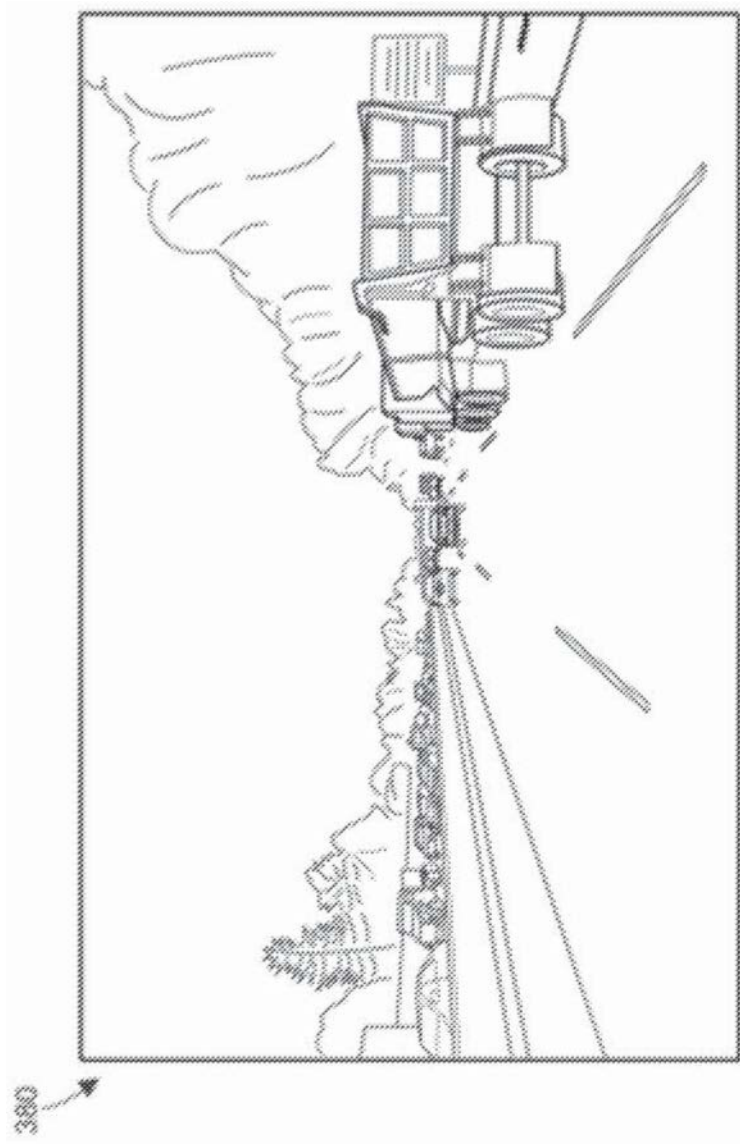


图5A

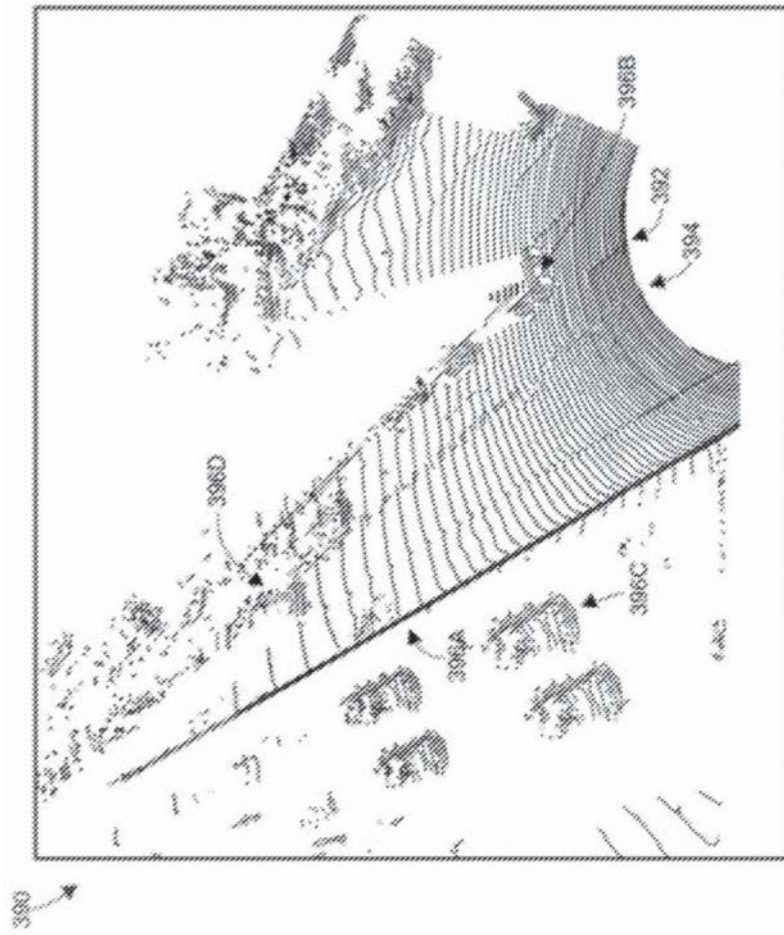


图5B

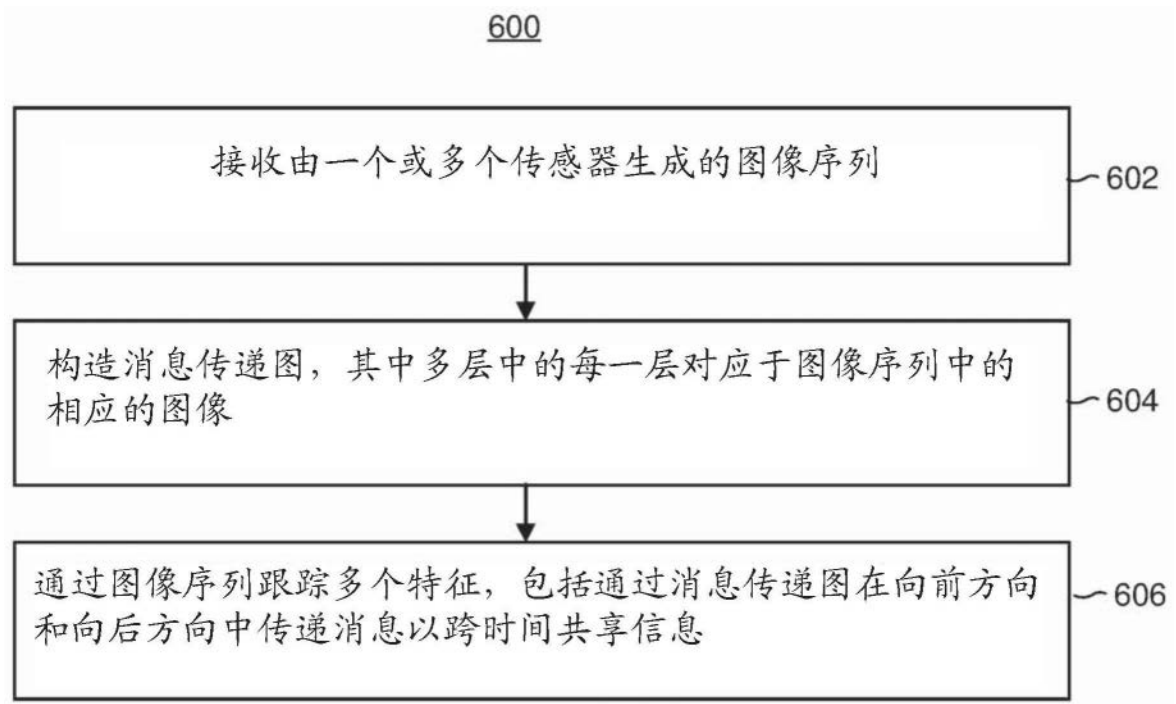


图6

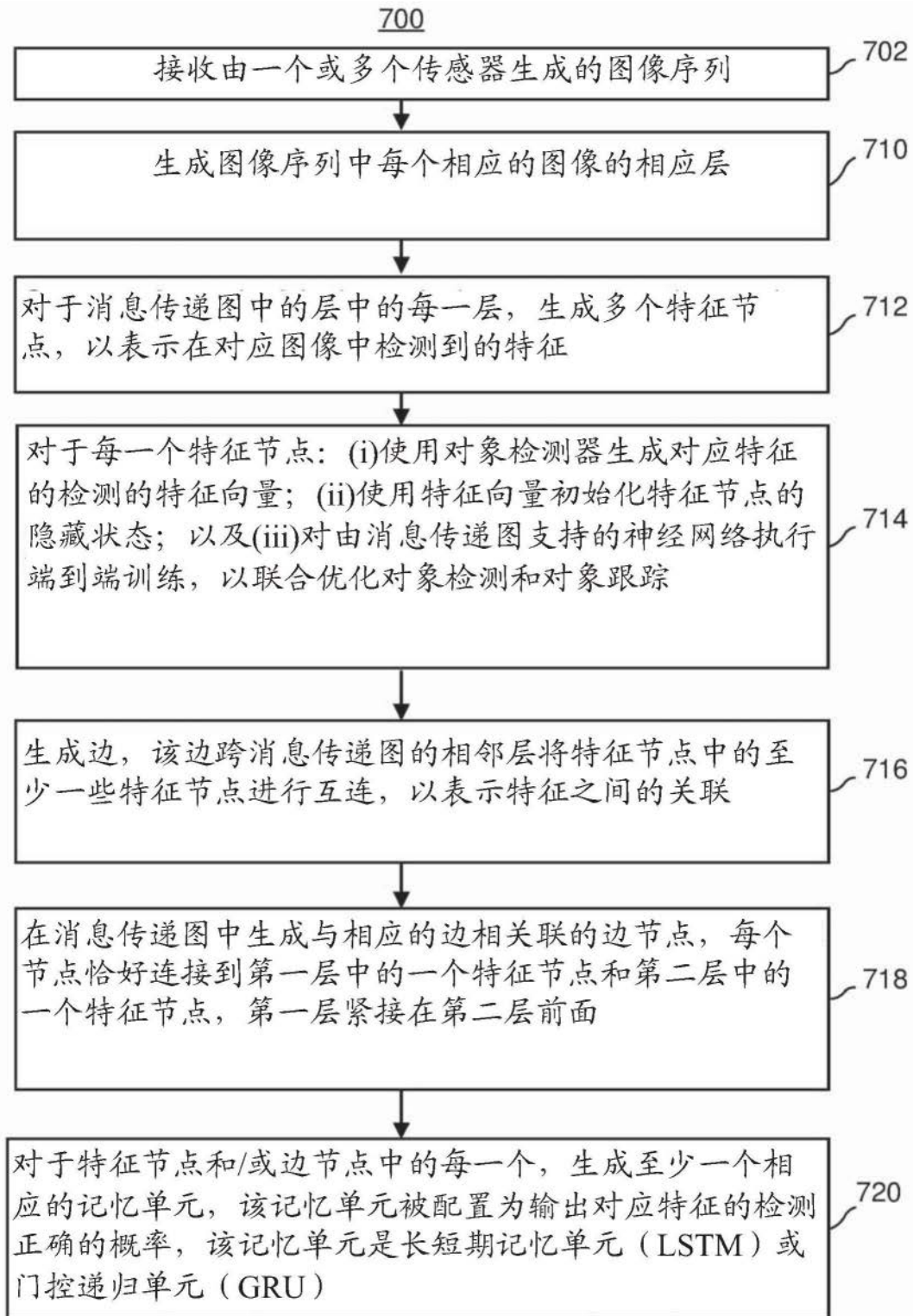


图7

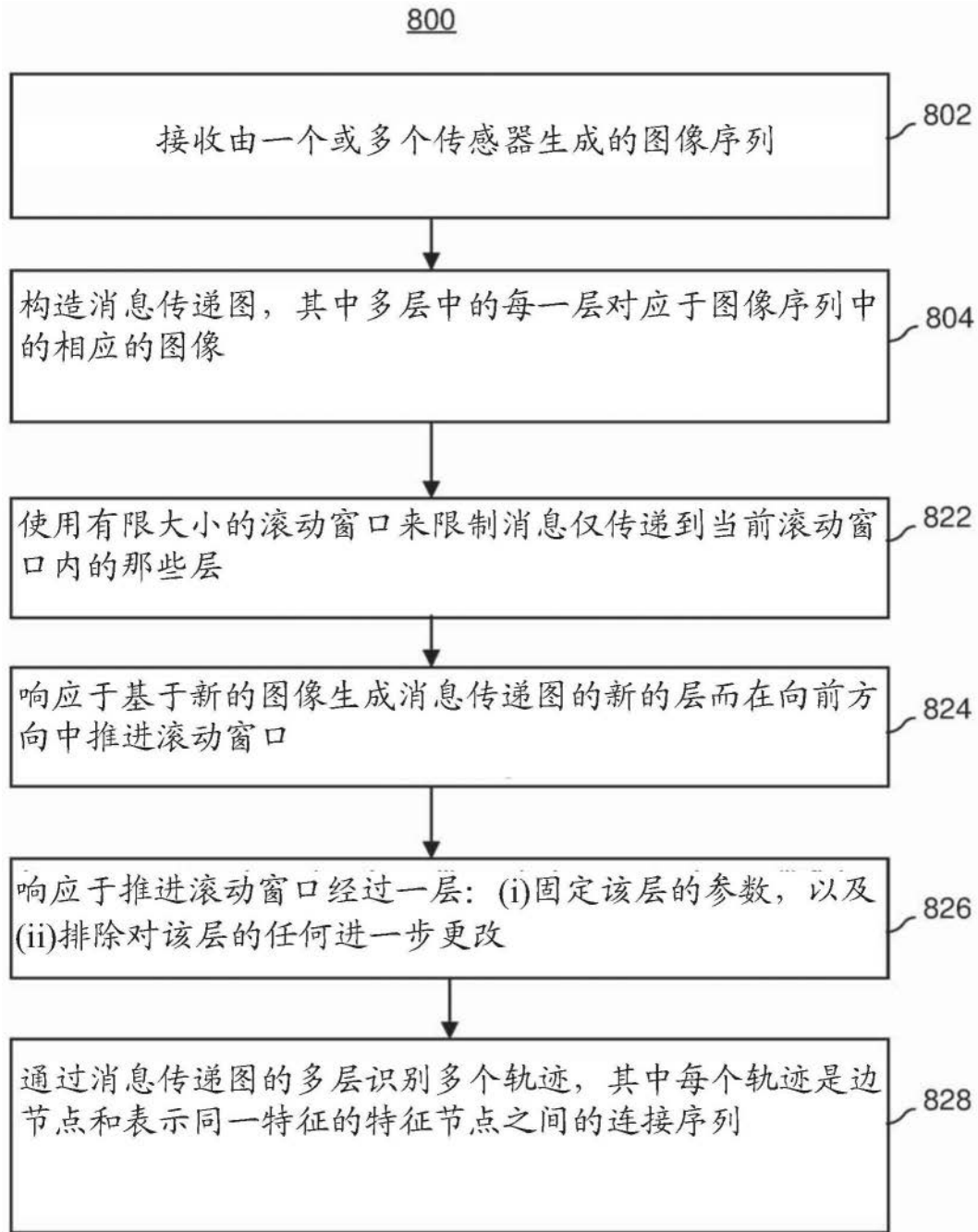


图8

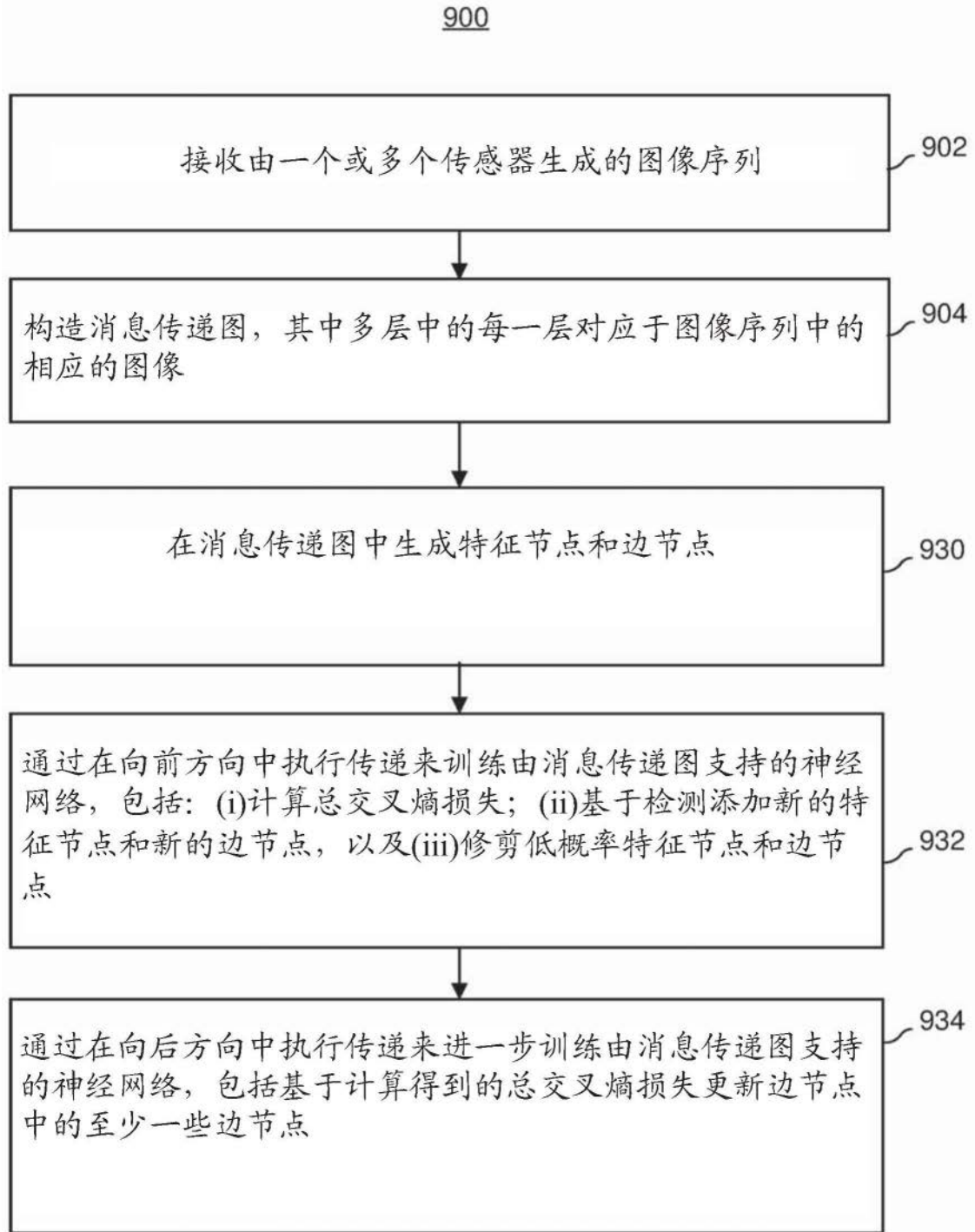


图9

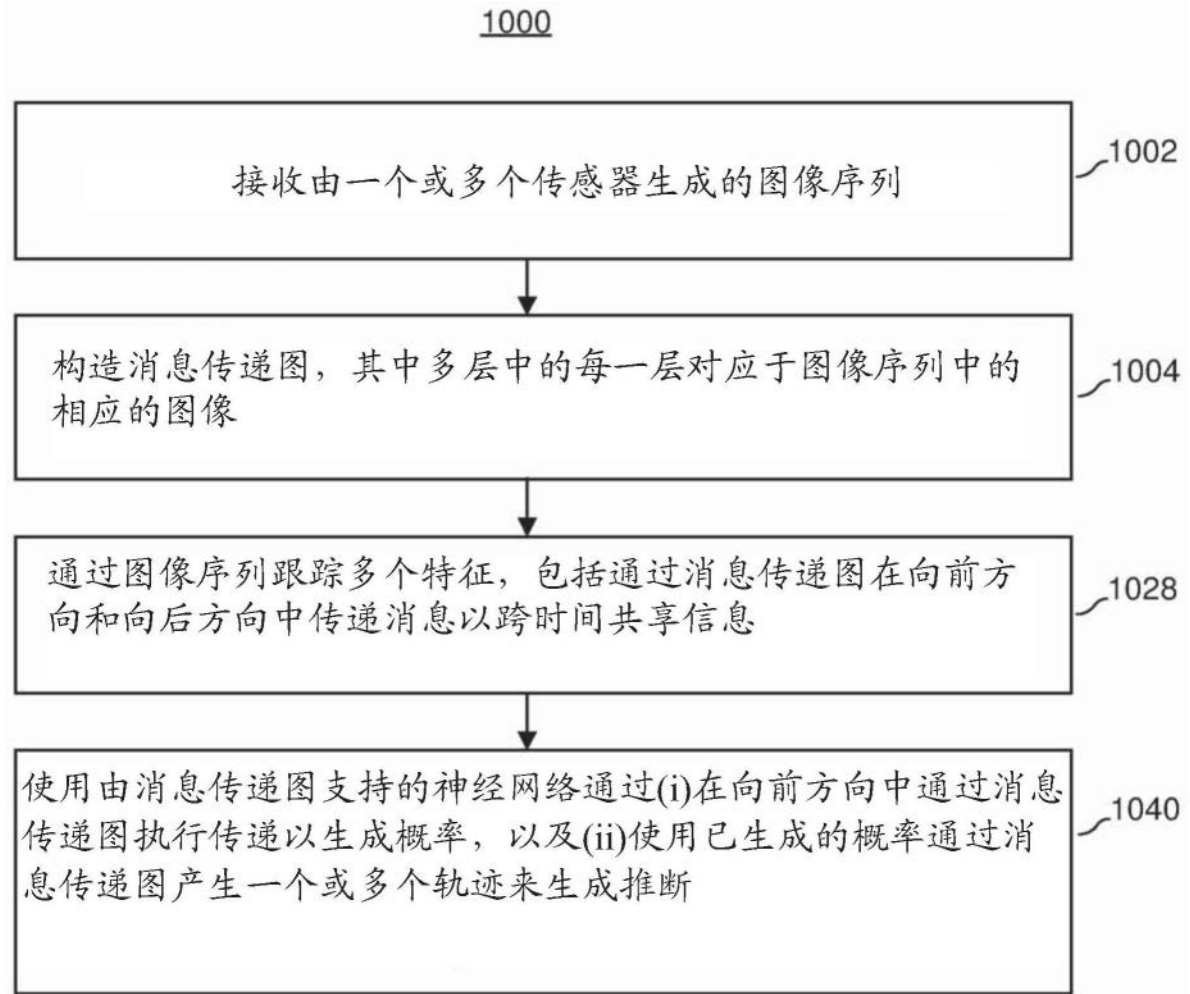


图10