



(12) 发明专利申请

(10) 申请公布号 CN 113039563 A

(43) 申请公布日 2021.06.25

(21) 申请号 201980074940.6

A·T·巴留索 S·菲德勒

(22) 申请日 2019.11.15

(74) 专利代理机构 北京市磐华律师事务所
11336

(30) 优先权数据

62/768,846 2018.11.16 US

62/827,313 2019.04.01 US

代理人 高伟

(85) PCT国际申请进入国家阶段日

2021.05.13

(51) Int.Cl.

G06N 3/08 (2006.01)

G06N 3/04 (2006.01)

G06T 11/60 (2006.01)

G06N 20/20 (2019.01)

G06F 16/901 (2019.01)

G06T 7/00 (2017.01)

(86) PCT国际申请的申请数据

PCT/US2019/061820 2019.11.15

(87) PCT国际申请的公布数据

W02020/102733 EN 2020.05.22

(71) 申请人 辉达公司

地址 美国加利福尼亚州

(72) 发明人 A·卡尔 A·普拉卡什 刘洛培

D·J·阿库纳·马雷罗

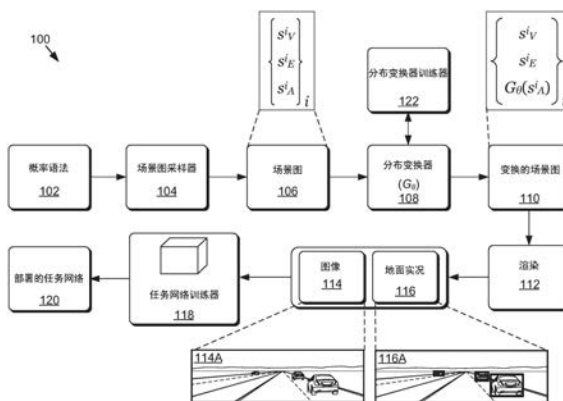
权利要求书2页 说明书14页 附图13页

(54) 发明名称

学习生成用于训练神经网络的合成数据集

(57) 摘要

在各种示例中,生成模型用于合成数据集,以用于训练下游机器学习模型来执行相关任务。可以通过从场景语法(诸如概率语法)中采样场景图并将所述场景图应用于所述生成模型来计算更代表真实世界数据集的对象属性分布的更新场景图以生成所述合成数据集。所述下游机器学习模型可以针对真实世界的验证数据集进行验证,并且所述模型在所述真实世界验证数据集上的性能可以用作进一步训练或微调所述生成模型的附加因素,用于生成特定于所述下游机器学习模型所述任务的所述合成数据集。



1. 一种方法,包括:对使用场景语法生成的场景图进行采样;将表示所述场景图的第一数据应用于第一机器学习模型,所述第一机器学习模型被训练以预测更新场景图,所述更新场景图具有在真实世界属性分布之后建模的合成属性分布;使用所述第一机器学习模型并且至少部分地基于所述第一数据,来计算表示变换的场景图的第二数据,所述变换的场景图包括由所述场景图内的初始属性表示的至少一个对象的更新属性;至少部分地基于所述第二数据来渲染表示图像的图像数据;至少部分地基于所述第二数据来生成表示相应地面实况的地面实况数据;以及使用所述图像数据和所述地面实况数据来训练第二机器学习模型。

2. 根据权利要求1所述的方法,其中,所述场景语法是概率语法。

3. 根据权利要求1所述的方法,其中,所述初始属性和相应的所述更新属性包括位置、姿势、维度、颜色、类别或标识值中的至少一个。

4. 根据权利要求1所述的方法,其中,所述第一机器学习模型包括图卷积网络。

5. 根据权利要求1所述的方法,其中,所述第一机器学习模型被进一步训练以预测所述变换的场景图,所述变换的场景图具有所述合成属性分布,所述合成属性分布是针对所述第二机器学习模型的任务量身定制的。

6. 根据权利要求5所述的方法,其中使用第一损失函数来训练所述第一机器学习模型用于预测所述变换的场景图,所述变换的场景图具有在所述真实世界属性分布之后建模的所述合成属性分布,并且使用与所述第一损失函数不同的第二损失函数来训练所述第一机器学习模型用于预测所述变换的场景图,所述变换的场景图具有所述合成属性分布,所述合成属性分布是针对所述第二机器学习模型的所述任务量身定制的。

7. 根据权利要求1所述的方法,其中,所述初始属性的第一子集是可变的,并且所述初始属性的第二子集是固定的,进一步地,其中在所述变换的场景图内仅所述第一子集考虑所述更新属性。

8. 根据权利要求1所述的方法,其中,所述地面实况数据是使用所述图像数据自动生成的,以对应于所述第二机器学习模型的任务。

9. 根据权利要求1所述的方法,其中,在部署之前,在真实世界验证数据集上测试所述第二机器学习模型。

10. 一种方法,包括:使用机器学习模型并且至少部分地基于从场景语法采样的表示初始场景图的第一数据来计算表示变换的场景图的第二数据,每个变换的场景图包括变换的属性,所述变换的属性与对应于所述初始场景图的初始属性不同;至少部分地基于所述第二数据渲染表示合成图像的第三数据;通过将对应于所述合成图像的合成属性分布与对应于真实世界图像的真实世界属性分布进行比较来确定差异;至少部分地基于所述差异,生成表示网络更新信息的第四数据;以及使用所述第四数据来训练所述机器学习模型。

11. 根据权利要求10所述的方法,其中所述合成属性分布或所述真实世界属性分布中的至少一个通过以下步骤确定:将表示所述真实世界图像或所述合成图像中的至少一个的第五数据应用于被训练用于特征提取的另一机器学习模型;以及至少部分地基于所述另一机器学习模型的输出来确定所述合成属性分布或所述真实世界属性分布中的至少一个。

12. 根据权利要求10所述的方法,其中,所述场景语法是概率场景语法。

13. 根据权利要求10所述的方法,其中,所述机器学习模型是图卷积网络。

14. 根据权利要求10所述的方法,其中,所述差异是最大平均差异(MMD)。

15. 根据权利要求10所述的方法,还包括:使用所述合成图像和相应的地面实况来训练另一机器学习模型;在真实世界验证集上测试所述另一机器模型;至少部分地基于所述测试来确定所述另一机器学习模型的性能得分;以及至少部分地基于所述性能得分来进一步训练所述机器学习模型。

16. 根据权利要求15所述的方法,其中第一损失函数用于使用所述第四数据来训练所述机器学习模型,并且第二损失函数用于至少部分地基于所述性能得分来训练所述机器学习模型,所述第一损失函数不同于所述第二损失函数。

17. 根据权利要求15所述的方法,其中,至少部分地基于所述第三数据自动生成所述地面实况。

18. 根据权利要求10所述的方法,其中所述机器学习模型的参数通过训练所述机器学习模型来初始化,以计算更新场景图,以使得所述更新场景图在与所述初始场景图的阈值相似性内。

19. 根据权利要求10所述的方法,其中,使用自动编码器来执行初始化所述机器学习模型的所述参数。

20. 一种系统,包括:计算设备,所述计算设备包括一个或多个处理设备和一个或多个存储设备,所述一个或多个存储设备可通信地耦合至所述一个或多个处理设备,所述一个或多个存储设备在其上存储编程指令,当由所述处理器执行所述编程指令时,会导致以下项的实例化:分布变换器,用于使用机器学习模型并且至少部分地基于从场景语法采样的表示初始场景图的第一数据来计算表示变换的场景图的第二数据,每个所述变换的场景图包括变换的属性,所述变换的属性与对应于所述初始场景图的初始属性不同;渲染器,用于至少部分地基于所述第二数据来渲染表示合成图像的第三数据;差异确定器,用于确定对应于所述合成图像的合成属性分布与对应于真实世界图像的真实世界属性分布之间的差异;以及模型训练器,用于生成表示网络更新信息的第四数据,所述网络更新信息用于训练所述机器学习模型,所述网络更新信息至少部分地基于所述第三数据生成。

21. 根据权利要求20所述的系统,其中所述一个或多个存储设备在其上存储编程指令,当由所述处理器执行所述编程指令时,会导致以下项的实例化:任务网络训练器,用于使用所述第三数据和所述第四数据来训练任务网络;以及任务网络验证器,用于在处理真实世界验证集时确定所述任务网络的性能得分,其中,所述模型训练器至少部分地基于所述性能得分来进一步训练所述机器学习模型。

22. 根据权利要求20所述的系统,其中,所述分布变换器包括图卷积网络。

23. 根据权利要求20所述的系统,其中,所述模型训练器使用第一损失函数来使用所述第四数据训练所述机器学习模型,以及使用与所述第一损失函数不同的第二损失函数来至少部分地基于所述性能得分训练所述机器学习模型。

学习生成用于训练神经网络的合成数据集

背景技术

[0001] 数据收集和标记是一项费力、昂贵且耗时的任务,需要无数的人力和计算资源。然而,机器学习模型(诸如神经网络)需要大量数据和相应的地面实况信息,以便在部署之前进行有效的训练。结果,模型训练的数据收集和标记部分成为大多数机器学习管线中的重大瓶颈。

[0002] 为了解决这个问题,合成数据生成已经显现为生成更大数量的地面实况信息的解决方案——例如,使用图形引擎。用于合成数据创建的一些常规方法需要有资格的专家来创建虚拟世界,并从该虚拟世界中采样合成数据。然而,手动创建虚拟世界的过程与手动标记真实世界数据一样是一项艰巨的任务。代替以这种方式创建虚拟世界,一些常规方法使用域随机化(DR)作为照片真实感环境模拟的廉价替代方案。DR技术通过将对象插入随机位置和姿势的环境中,来生成大量多样的场景——然而,这种随机性通常会导致创建与真实世界场景的代理非常不同并且不是最佳的环境。

[0003] 其他常规方法旨在解决合成环境与真实世界环境之间的领域差距,诸如生成比现有解决方案更接近真实世界外观的合成数据。此类方法使用生成式对抗网络(GAN)以将合成数据的外观变换为更类似于真实世界的数据——但是这些方法仍无法解决与真实世界环境匹配的合成环境的不足。例如,模拟环境中对象的属性(诸如位置、姿势、颜色、纹理、形状、大小等)不能反映真实环境中相似对象的属性。

发明内容

[0004] 本公开的实施例涉及生成用于训练神经网络的合成数据集。公开了使用生成模型(诸如图形卷积网络(GCN))将从场景语法中采样的初始场景图变换成变换的场景图的系统和方法,所述变换的场景图具有相对于初始场景图的属性更新的属性。

[0005] 可以训练生成模型,以计算变换的场景图,使得对应属性的分布更接近地反映真实世界环境或场景的分布。此外,使用变换的场景图生成的合成数据集和相应的地面实况,可用于训练下游任务网络,并且可利用在真实世界验证数据集上的下游任务网络的性能,来进一步训练和微调生成模型。结果,所述生成模型不仅可以预测可用于渲染更多合成数据集的变换的场景图,而且还可以对合成数据集进行调整,以更有效地训练下游任务网络用于其相应任务。

[0006] 因此,生成模型有助于弥合合成数据和真实世界数据之间先前存在的内容差距。结果,作为非限制性示例,其中常规方法可生成合成场景,所述合成场景包括照片般逼真的车辆、行人、建筑物和/或以不真实的方式分布的其他对象(例如,垂直于道路行驶方向的车辆、比实际环境中的人行道宽的人行道、靠得太近的行人等),本公开的技术允许这些合成场景不仅包括照片般逼真的渲染,而且还包括以更真实、真实世界的类似方式分布的对象。因此,除了弥合先前方法所关注的领域差距之外,当前方法还弥合了内容差距,因此产生了照片般逼真的环境,所述环境可以模拟真实世界的场景,其具有使得下游任务网络能够在这些合成数据集上进行训练的准确性,同时,将其部署为使用真实世界数据在真实世界应

用中使用。

附图说明

[0007] 下面参考附图详细描述用于学习生成用于训练神经网络的合成数据集的本系统和方法,其中:

[0008] 图1是示出了根据本公开的一些实施例的用于训练分布变换器模型以生成用于训练下游任务网络的合成数据集的过程的数据流程图;

[0009] 图2A是示出了根据本公开的一些实施例的用于预训练分布变换器模型的过程的示例说明;

[0010] 图2B是示出了根据本公开的一些实施例的用于使用真实世界数据的分布与合成数据的分布之间的差异来训练分布变换器模型的过程的示例说明;

[0011] 图2C是示出了根据本公开的一些实施例的用于训练分布变换器模型以计算针对下游任务网络的任务而微调变换的场景图的过程的示例说明;

[0012] 图3A-3B是示出了根据本公开的一些实施例的使用真实世界分布和合成分布来训练分布变换器模型以计算变换的场景图的示例说明;

[0013] 图4A-4B是示出了根据本公开的一些实施例的使用真实世界分布和合成分布来训练分布变换器模型以计算变换的场景图的示例说明;

[0014] 图5A-5B是示出了根据本公开的一些实施例的当使用与基线模型相比由分布变换器模型生成的合成数据集时下游任务网络性能的示例说明;

[0015] 图6是示出了根据本公开的一些实施例的用于训练分布变换器模型以计算变换的场景图方法的流程图;

[0016] 图7是示出了根据本公开的一些实施例的用于使用分布变换器模型生成的合成数据集来训练下游任务网络方法的流程图;以及

[0017] 图8为适合用于实现本公开的一些实施例的示例计算设备的框图。

具体实施方式

[0018] 公开了与用于训练神经网络的生成合成数据集相关的系统和方法。这些系统和方法可用于多种实施方式,实施方式包括但不限于自动驾驶车辆控制和导航(例如,关于自主或半自主车辆、水上船只或水上飞行器、无人机、飞机、应急车辆、施工设备和/或其他自主或半自主车辆类型)、用于测试的模拟环境的生成(例如,测试机器人、无人机、自主车辆等)、游戏和/或室内导航、光流估计、无人机或四轴飞行器的控制和导航等等。例如,本文描述的系统和方法可用于生成用于训练神经网络的合成数据集,所述神经网络包括在自动驾驶软件栈、机器人软件栈、无人机或四轴飞行器软件栈等中。作为另一个示例,本文描述的系统和方法可用于生成更类似于真实世界环境的模拟环境,以便测试或训练车辆(例如,人为控制的、自主的、半自主的等)、机器人、无人机、人类(例如,用于真实世界场景,诸如战斗、操作机器等)等。在一些非限制性实施例中,本文关于图8的示例计算设备800描述的一些或全部组件、特征和功能可用于执行过程100(图1)和过程200(图2A-2C)的一些或全部。

[0019] 现在参考图1,图1是示出了根据本公开的一些实施例的用于训练分布变换器模型以生成用于训练下游任务网络的合成数据集的过程100的数据流程图。例如,给定真实图像

X_R 和任务T(例如,对象检测、车道分类等)的数据集,过程100可用于合成训练数据集, $D_T=(X_T, Y_T)$,包含合成图像 X_T 和相应的地面实况 Y_T 。此外,在为特定任务T标记真实世界数据的验证集 V_T 的情况下,过程100可以针对元目标(例如,特定任务T的下游性能)另外优化或调整分布变换器108。结果,过程100可以不仅用于训练分布变换器108以生成与真实世界的场景更相似场景的合成数据集(例如,图像114和对应的地面实况116),而且过程100还可以用于微调合成数据集的生成,所述合成数据集用于特定下游任务(例如,部署的任务网络120的任务)。

[0020] 过程100可以包括场景语法(例如概率语法102),所述场景语法可以定义规则,诸如对应于场景的对象和特征的拓扑、概率参数和/或分布(例如,属性分布),用于生成定义场景实例的场景图。例如,概率语法102可以定义场景集和场景上的概率分布。可以使用一组构造块和一组规则来定义每个场景,这些规则定义一组对象的共现。然后,可以将对象的构造块和共现组合在一起,以形成可以由规则支配的场景的可能结构和/或构图的更大集合。由场景语法表示的场景可以捕获环境中不同对象和特征的存在,以及不同对象和特征之间的关系。例如,概率语法102可以定义一组可能的场景以及场景上的概率分布,每个场景由使用一组规则连接在一起的一组块定义。

[0021] 从场景语法(例如,概率语法102)采样(例如,通过场景图采样器104)场景的每个实例可以表示为场景图106。每个场景图106可以以简洁的层次结构表示场景的元素(例如,对象及其特征或组件),每个元素具有一组属性,例如类别、位置、颜色、姿势、尺寸、纹理、资产标识符等。该层次结构可以定义父子依赖关系,其中子元素的属性可以相对于父元素定义,因此允许以高效和自然的方式来创建和修改场景。通过放置场景图106中所描述的对象或元素,可以轻松渲染相应的合成图像(例如图像114)和像素级别注释(例如地面实况116)。

[0022] 概率语法102,P可以定义用于在由场景图采样器104采样时,生成场景图106,S的生成过程。对于非限制性示例,为了生成交通场景,可以先定义道路的中心线,然后是平行线,然后是车道内的车辆等。场景的结构可以由概率语法102定义,而与每个对象或特征相对应的属性可以从参数分布中进行采样,这需要仔细调整才能准确。因此,使用从概率语法102采样的场景图106来生成图像114和地面实况116,以用于训练下游任务网络,并不像使用本文所述的变换的场景图110那样生成准确的结果。这可能是由于要求对属性分布进行微调的结果,这是一项困难、耗时且通常不准确的尝试,导致从概率语法102中采样的场景与真实世界的场景和环境不太相似。

[0023] 为了弥补这个缺点,过程100可以实施分布变换器108, G_0 ,其可以被训练来更新或变换属性 S_A (对应于场景图106,S)给更新的属性($G_0(S_A)$,对应于变换的场景图110)。在一些实施例中,可以假设场景图106的结构是正确的(例如,驾驶场景具有道路、天空、多个对象等)。这样,变换的场景图110可以包括基本相同的结构(例如,顶点 S_V 和边 S_E 可以保持相同),但是具有变换的属性或更新的属性, $S_A \rightarrow G_0(S_A)$ 。通过维持所述结构但修改属性,变换的场景图110的渲染更接近真实世界的场景。该过程可能等同于学习一个生成模型,所述模型具有可更改对象属性(例如,颜色、姿势、纹理、位置、大小等)的灵活性。本质上,从概率语法102生成的结构得以保持,同时变换属性的分布,从而充当分布变换器108。下面的等式(1)表示当应用分布变换器 G_0 时,场景图106到变换的场景图110的变换:

$$[0024] \quad G_0[S_V, S_E, S_A] = [S_V, S_E, G_0(S_A)] \quad (1)$$

[0025] 在一些非限制性实施例中,由于场景图106的图形结构,分布变换器108 G_0 可以包括卷积神经网络(CNN)——诸如图卷积网络(GCN)。对于非限制性示例,其中分布变换器108是GCN的情况下,GCN可以使用两个不同的权重矩阵来分别捕获自顶向下和自下而上的信息流。在一些实施例中,分布变换器108可以进行每个节点的预测,这意味着分布变换器108可以为 S_V 中的每个节点生成变换的属性 $G_0(S_A)$ 。

[0026] 在一些非限制性实施例中,尽管可以将所有属性 S_A 作为输入提供给分布变换器108,但是可以仅存在要修改的属性的子集。例如,属性的第一子集(例如,可变的或可修改的属性)可以被标识(例如,选择的、学习的等)以用于修改,而属性(例如,不可变的或固定的属性)的第二子集可以被标识(例如,选择的、学习的等)为固定的。在这样的示例中,可以从场景图106信任第二组属性,如从概率语法102采样的那样。作为示例,对于街道场景,当场景用于检测车辆时,可能不需要改变房屋的高度或人行道的宽度。这样,这些属性(例如,高度、宽度)在该任务期间可能是不变的,而与街道或街道上的对象相对应的属性可能是可变的。这样,每个节点属性的第一子集 $v \in S_V$ 可能是可变的,并且可以用 $S_{A,mut}(v)$ 表示。通过减少可变属性 $S_{A,mut}(v)$ 的数量、可以减少分布变换器108要求调整的暴露参数(例如权重、偏差等)的数量——从而改善了训练时间和复杂性,以及减少了推理时的运行时间。

[0027] 尽管本文中相对于使用神经网络,特别是使用GCN作为分布变换器108来描述了示例,但这并不旨在进行限制。本文所述的分布变换器108可以例如但不限于,包括任何类型的机器学习模型,例如使用线性回归、逻辑回归、决策树、支持向量机(SVM)、朴素贝叶斯、k最近邻(Knn)、K均值聚类、随机森林、降维算法、梯度增强算法、神经网络(例如,自动编码器、卷积、循环、感知器、长/短期记忆(LSTM)、Hopfield、玻尔兹曼、深度信念、反卷积、生成式对抗、液态机器等)的一个或多个机器学习模型和/或其他类型的机器学习模型。

[0028] 可以使用分布变换器训练器122来训练分布变换器108,这里将参照图2A-2C更详细地描述分布变换器训练器122。分布变换器训练器122的目标可以是鉴于真实世界的数据来更新分布变换器108的参数,以使得与变换的场景图110相对应的属性分布更接近真实世界数据的属性分布。结果,当由渲染器112渲染变换的场景图110时,渲染的或合成的场景更接近地反映真实世界场景。

[0029] 渲染器112可以渲染包括图像114和地面实况116的合成数据集。然后可以使用合成数据集,经由任务网络训练器118来训练任务网络。例如,假设任务网络的任务是对象检测,则变换的场景图可以表示一个或多个对象,使得渲染图像114(例如,渲染图像114A)可以描绘对象(例如,渲染图像114A中的车辆)。此外,由于在该示例中任务可以是对象检测,可以根据渲染信息(例如,与地面实况116A中车辆相对应的边界形状)(在实施例中自动)确定地面实况116(例如,地面实况116A)。

[0030] 用于生成逼真的高分辨率图像的现有解决方案以输入图像为条件,并且变换输入图像的外观(例如,以弥合域差距)。然而,这些方法假设地面实况标签是不变的,因此其内容(例如,结构)的可变性是有限的。因此,如果改变图像的内容(例如,对象的位置、姿势、大小、颜色等),则常规方法可能无法生成准确的地面实况信息。然而,相反地,使用过程100从变换的场景图110生成合成数据集 $D(\theta)$ 解决了这个问题。例如,由于对象包括相关联的语义信息(例如,系统知道对象是汽车或者行人),因此合成或修改合成场景仍将渲染出完美的

地面实况116。这样,在先前方法使用场景图渲染图像,然后修改图像外观的情况下,过程100经由分布变换器108修改场景图106,以生成变换的场景图110,然后从修改的场景图110渲染图像114(例如,使用图形引擎)。这样,使用表示其图像数据和/或表示变换的场景图110(例如,对象类型、位置、姿势、属性等)的数据,可以从图像114自动生成地面实况116。

[0031] 任务网络训练器118可以将表示图像114的图像数据应用于任务网络,并且可以将任务网络的预测与地面实况116进行比较。可以使用一个或多个损失函数来训练任务网络,所述任务网络使用地面实况116,直到任务网络达到特定任务的最佳性能或阈值性能为止。在一些示例中,任务网络可以在使用合成数据集 $D(\theta)$ 进行训练之后,针对真实世界的数据集进行测试,以验证任务网络的性能。此外,在一些非限制性实施例中,可以在模拟或虚拟环境中使用任务网络(在实施例中,也可以使用分布变换器108和变换的场景图110来生成或渲染该环境),以便在部署任务网络以用于真实世界环境之前测试任务网络的性能。例如,任务网络可以是自动驾驶软件栈的部分(例如,该栈的感知层的部分),负责对象检测。这样,虚拟或模拟车辆可以在虚拟环境中实现自动驾驶软件栈,其中模拟车辆可以使用虚拟或模拟图像传感器或照相机(或其他传感器类型)捕获虚拟图像数据(或其他传感器数据类型)。然后将虚拟图像数据应用于任务网络,以测试虚拟环境中自动驾驶栈内任务网络的功能。所述虚拟环境可以类似于在2019年3月27日提交的美国非临时申请No.16/366,875中描述的虚拟或模拟环境,在此通过引用将其整体并入本文。

[0032] 一旦使用任务网络训练器118训练、测试和验证了任务网络,则可以将任务网络部署为部署的任务网络120。例如,可以将部署的任务网络实现为真实世界系统,例如自主或半自主车辆、机器人、光流估计器、游戏系统、无人机或四轴飞行器等。结果,所部署的任务网络120可能已使用更接近真实世界场景的合成数据集进行了训练,从而(至少部分地)消除了捕获真实世界数据、手动标记真实世界数据用于地面实况的生成以及在真实世界数据和地面实况上训练网络的负担。本文所述的大多数机器学习管线的数据收集和标记部分的瓶颈可能会被消除或减少,从而在与仅基于真实世界数据训练的常规任务网络相比时,在不牺牲所部署的任务网络120的准确度或性能的情况下,加快任务网络的训练。

[0033] 现在参考图2A-2C,图2A-2C描述了用于使用分布变换器训练器122训练分布变换器108的过程200。关于图2A-2C,实线箭头可以表示正推法通过过程,而虚线箭头可表示倒推法。例如,关于图2A,如实心箭头所示,变换的场景图110可以由每节点重建损失206使用,并且每节点重建损失206可将信息传回给分布变换器108(例如,经由变换的场景图110),用于更新分布变换器108的参数,如虚线箭头所示。

[0034] 参考图2A,图2A是示出了根据本公开的一些实施例的用于预训练分布变换器模型的过程200A的示例说明。概率语法102可表示关于应该如何生成场景或场景图的先验概率分布。这样,为了预训练分布变换器(例如,初始化诸如权重和偏差的参数),可以使用概率语法102,P。该预训练可以包括初始化分布变换器108, $G(\theta)$,以执行识别功能,使得 $G(\theta)(S) = S$,其中S是场景图106,并且 $G(\theta)(S)$ 是变换的场景图110。每个节点的输入特征可以是其属性集 S_A ,所述属性集 S_A 可以在所有节点上一致地定义。由于 S_A 可以由不同的分类和连续分量组成,因此当使用每节点重建损失206进行训练时,可以使用每特征分量的适当损失。例如但不限于,交叉熵损失可用于分类属性,而L1损失可用于连续属性。最终,使用自动编码器过程200A可以预训练或初始化分布变换器108,以生成与作为输入提供给分布变换器的

场景图106在阈值相似度(例如,基本上相同)内的变换的场景图110。使用过程200A,由于分布变换器108的初始参数可以反映作为概率语法102的先验概率分布,因此可以减少分布变换器的总体训练和收敛时间。

[0035] 现在参考图2B,图2B是示出了根据本公开的一些实施例的用于使用真实世界数据与合成数据的分布之间的差异来训练分布变换器模型的过程200B的示例说明。例如,分布变换器108的第一目标可以是使渲染合成图像 X_r 的分布更接近真实图像 X_r 的分布(例如,使对应于真实场景212的分布216更接近生成的合成场景210的分布214)。过程200B可以包括从概率语法102中采样场景图106,将表示场景图106的数据应用到分布变换器108,通过分布变换器预测变换的场景图110,使用渲染器112并基于变换的场景图110渲染生成的合成场景210,然后对分布214和分布216进行采样以确定差异或不同。结果,如后向传播箭头所示,差异信息随后可用于在多次迭代中更新分布变换器108的参数,直到分布214和分布216在阈值相似度(例如,使得所生成的合成场景210更接近真实场景212)之内。

[0036] 为了生成分布214和/或分布216,可以将真实场景212和/或生成的合成场景210(例如,表示其的图像)应用于特征提取器。所述特征提取器可以包括计算机视觉算法、训练用于特征提取的神经网络或另一种类型的特征提取器算法。例如但不限于,特征提取器可以是使用线性回归、逻辑回归、决策树、支持向量机(SVM)、朴素贝叶斯、k-最近邻(Knn)、k均值聚类、随机森林、降维算法、梯度提升算法、神经网络(例如,自动编码器、卷积、递归、感知器、长/短期记忆(LSTM)、Hopfield、玻尔兹曼、深层信念、反卷积、生成对抗、液态机等)一个或多个机器学习模型,和/或其他类型的机器学习模型。由特征提取器计算的特征可用于确定分布214和/或216(例如,可以分析该信息以确定在驾驶场景中车辆的位置、姿势、颜色、尺寸等)。在一些实施例中,关于分布214,可以从变换的场景图110确定分布214,而不需要特征提取器(例如,变换的场景图可以包括指示场景210中的对象的属性的语义和/或其他信息)。在其他示例中,可以使用特征提取器、变换的场景图110和/或另一种属性确定方法的组合来确定与所生成的合成场景210和/或真实场景212相对应的分布214。

[0037] 可以使用多种方法中的任何一种来确定分布214和216之间的差异或不同。例如,在一些非限制性实施例中,可以使用最大平均偏差(MMD) 218。MMD 218是可用于训练生成模型(诸如,分布变换器108)的分布之间的相似性的频度量,以将生成的分布214的统计信息与目标分布216的统计信息相匹配。然而,这并不旨在受到限制,并且可以使用其他方法,诸如(不限于)使用鉴别器进行对抗式学习。在一些实例中,使用MMD 218的逐像素生成模型(诸如分布变换器108)已经无法有效地对高频信号进行建模,从而导致模糊数据的生成。但是,由于过程100和200使用渲染器112,完全避免了此问题,因此允许将MMD 218用于训练稳定性。

[0038] 在其中使用MMD 218的示例中,如本文所述,可以使用图像分类模型和/或特征提取器模型在特征空间中计算MMD 218,并用 ϕ 来表示。在一些非限制性示例中,内核起始距离(KID)可用于测量分布214和216之间的相异点。内核技巧可以用于具有高斯内核 $k(x_i, x_j)$ 的计算。

[0039] 关于图2B,可以通过给定从概率语法102和目标真实图像 X_r 采样的给定场景图106、S1、...、SN来执行分布匹配,根据以下等式(2)计算平方MMD 218距离:

$$[0040] \quad \mathcal{L}_{MMD^2} = \frac{1}{N^2} \sum_{i=1}^N \sum_{i'=1}^N k(\phi(X_\theta(S_i)), \phi(X_\theta(S_{i'}))) + \frac{1}{M^2} \sum_{j=1}^M \sum_{j'=1}^M k(\phi(X_R^j), \phi(X_R^{j'})) - \frac{1}{MN} \sum_{i=1}^N \sum_{j=1}^M k(\phi(X_\theta(S_i)), \phi(X_R^j)) \quad (2)$$

[0041] 其中,从S渲染的图像是 $X_\theta(S) = R(G_\theta(S))$ 。

[0042] 为了从 $G_\theta(S)$ 采样,符号 $R(G_\theta(S))$ 可能会过载,因为R实际上可能需要从预测 $G_\theta(S)$ 采样。可以假设跨场景、节点和属性的独立性(由变换的场景图110表示),以便可以对变换的场景图110中的每个节点的每个属性进行独立采样。在使用MMD 218进行训练时,可以将 S_A 中的分类属性标记或标识为不可变的,并且可以将预测的连续属性直接作为样本传递。

[0043] 为了优化MMD损失,在一些实施例中,可以通过不可微分的渲染函数R(例如,使用渲染器112)反向传播该梯度。例如,可以使用有限距离的方法来近似 $R(G_\theta(S))$ 相对于 $G_\theta(S)$ 的梯度。尽管此过程可能会产生噪声梯度,但对于实际训练模型而言已经足够了,并且具有照片般逼真渲染的附加好处。在一些实施例中,可微分渲染可用于减少噪声梯度。

[0044] 现在参考图2C,图2C是示出了根据本公开的一些实施例的用于训练分布变换器模型以计算针对下游任务网络的任务而微调变换的场景图过程200C的示例说明。例如,分布变换器108的第二个目标可以是概率语法102P生成数据,使得当在真实世界数据的目标验证集V上进行测试时,基于该数据训练的模型达到最佳性能。这可以被称为元目标,其中可以优化输入数据,以提高验证集V的准确性。可以将该元目标调整到为特定下游任务进行训练的任务网络。因此,可在以下等式(3)的目标下训练分布变换器108 G_θ :

$$[0045] \quad \max_{\theta} \mathbb{E}_{S' \sim G_\theta(S)} [\text{score}(S')] \quad (3)$$

[0046] 其中 $\text{score}(S')$ 是在对任务网络进行数据 $R(G_\theta(S))$ 训练后,在验证数据V上获得的性能度量或得分。由于得分是使用验证数据而非 S' 来衡量的,因此等式(3)中的任务损失可能无法相对于参数 θ 进行微分。增强得分函数估计器或梯度的另一无偏估计器,可用于计算等式(3)的梯度。将目标重新定义为损失,并写出以下梯度产出等式(4):

$$[0047] \quad \begin{aligned} \mathcal{L}_{task} &= \mathbb{E}_{S' \sim G_\theta(S)} [\text{score}(S')] \\ \nabla_{\theta} \mathcal{L}_{task} &= \mathbb{E}_{S' \sim G_\theta(S)} [\text{score}(S') \times \nabla_{\theta} \log p_{G_\theta}(S')] \end{aligned} \quad (4)$$

[0048] 为了减少来自估计器梯度的方差,可以跟踪先前得分指数移动平均值,并从当前得分中减去该先前得分指数移动平均值。可以使用来自 $G_\theta(S)$ 的一个样本来近似此期望。

[0049] 上面的等式(4)可能需要能够从分布变换器108采样和/或测量可能性。对于连续属性,可以将分布变换器108解释为预测每个属性的正态分布的均值,并带有预定义的方差。在一些实施例中可以使用从正态分布采样的重新参数化技巧。对于分类属性,可以从每个类别的预测对数概率的多项式分布中采样。如本文所述,在一些非限制性实施例中,分类属性可以是不可变的。

[0050] 为了计算 $\log p_{G_\theta}(S')$,可以假设跨场景、属性和场景中的对象之间的独立性。结果,对于整个场景,等式(4)的可能性可以简单地分解,如下面的等式(5)所示:

$$[0051] \quad \log p_G(S') = \sum_{s' \in S'} \sum_{v \in S'V} \sum_{a \in S'_{A,mut}(v)} \log p_{G_\theta}(s'(v, a)) \quad (5)$$

[0052] 其中, $s'(v, a)$ 表示在批处理 S' 中,单个场景 s' 中,在节点 v 中属性 a 。请注意,总和仅是每个节点的可变属性 $s'_{A,mut}(v)$ 。各个对数概率可以来自在此描述的定义的采样过程。

[0053] 图2C提供了过程200C的概述,如上所述。可以从概率语法102中采样场景图106,并将其应用于分布变换器108。分布变换器108可以计算或预测变换的场景图110,所述变换的场景图110可以由渲染器112来生成合成场景和标签220(例如,图1的图像114和地面实况116)。可以使用合成数据集220(例如,使用图1的任务网络训练器118)来训练任务网络224A(表示未训练的任务网络224)。一旦被训练或至少被部分训练,任务网络224B(表示被部分训练或被完全训练的任务网络224)可以在真实世界标记的验证集222上进行测试。真实世界标记的验证集222可以包括表示真实世界场景或环境的真实世界图像和相应的地面实况或验证标签。如本文所述,可以使用得分(S')来确定和表示真实世界标记的验证集222上的任务网络224B的性能。

[0054] 现在参考图3A-3B,图3A-3B是示出了根据本公开的一些实施例的使用真实世界分布和合成分布来训练分布变换器模型以计算变换的场景图的示例说明。例如,图3A和3B表示来自修改后的美国国家标准技术研究院(MNIST)数据库的数字分类。概率语法102从MNIST数据集(对于任何数字都具有相同的概率)中采样背景纹理、一位数字纹理(图像),然后对数字属性进行采样——例如旋转和位置。渲染器112基于采样变换来变换纹理并将其粘贴到画布上。在此示例中,任务网络是小的2层CNN,后面是三个完全连接层。可以应用完全连接层(具有50、100和10个特征)中的缺失。在过程100和200的测试期间,任务网络在规则MNIST分类任务上实现了大于99%的准确性。

[0055] 在图3A的示例中,概率语法102生成如场景图106所示的输入样本304A,这些输入样本304A是竖立的并且居中的,类似于规则MNIST数字。目标数据V和真实图像XR是图像302A,其中数字居中并始终旋转90度。因此,分布变换器108的目标是学习将输入样本304A中的数字旋转90度,同时保持它们在相同位置居中。从输出样本306A可以看出,分布变换器108成功地将每个输入样本304A旋转了90度,同时保持它们居中。

[0056] 在图3B的示例中,概率语法102生成由场景图106表示的输入样本304B,这些输入样本像规则MNIST数字一样竖立且居中。目标数据V和真实图像XR是图像302B,其中,数字始终旋转90度,并移动到画布各自部分的左下角。因此,分布变换器108的目标是学习将输入样本304A中的数字旋转90度,同时还应将数字变换到画布的各个部分的左下角。如输出样本306A所示,分布变换器108成功地将每个输入样本304A旋转90度,并将样本平移到画布各个部分的左下角。

[0057] 结果,直接对来自概率语法102的输入场景进行训练,得到略高于随机性能的结果。然而,在过程100和200中使用分布变换器108来解决由概率语法102生成的场景与真实世界场景之间的分布差异,下游任务网络的准确性大于99%。更具体地,对于图3A的示例,使用概率语法102的准确度为14.8%,而使用分布变换器108的准确度为99.5%。类似地,对于图3B的示例,使用概率语法102的准确度为13.1%,而使用分布变换器108的准确度为99.3%。

[0058] 现在参考图4A-4B,图4A-4B是示出了根据本公开的一些实施例的使用真实世界分布和合成分布来训练分布变换器模型以计算变换的场景图的示例说明。例如,在MNIST数据集和2D鸟瞰图上的模拟设置中验证过程100和200之后(如在2019年4月1日提交的美国临时申请No.62/827,313号和2018年11月16日提交的美国临时申请No.62/768,846号中所述,其全部内容通过引用合并于此),针对为目标检测的下游任务训练的任务网络评估了过程100

和200。KITTI数据集用作验证集V和真实世界图像XR。更具体地说,在该实验中,验证集V包括从KITTI训练集中随机抽取的100幅图像,其中包括相应的地面实况标签。其余训练数据(仅图像)形成XR。KITTI数据集包括挑战性的交通场景和从高速公路到城市再到更农村的邻里场景。另外,KITTI数据集包括简单子集、中等子集和困难子集,其范围包括图像数量和场景复杂性。与一些先前的实验相反,真实世界分布(例如,图2B的分布216)和合成分布(例如,图2B的分布214)之间的分布差异在这里自然地出现。

[0059] 概率语法102定义道路场景,例如道路场景语法。在非限制性实施例中,可以由渲染器112以三维(3D)渲染从概率语法102采样场景图106,所述渲染器112可以是游戏引擎。图像402A和402B可以表示从概率语法102采样的场景图106生成的渲染。可以看出,图像402A和402B包括在某些方向、大小、颜色、姿势、位置、纹理等处的车辆、树木、建筑物、人行道和行人。图像402A和402B是2019年4月1日提交的美国临时申请No. 62/827,313和2018年11月16日提交的美国临时申请No. 62/768,846中图12左栏中的两个图像的线图表示,其全部内容通过引用合并于此。图像404A和404B表示从分布变换器108计算出的变换的场景图110生成的渲染。可以看出,图像404A和404B包括在与图像402A和402B不同的某些方向、大小、颜色、姿势、位置、纹理等处的车辆、树木、建筑物、人行道和行人。实际上,图像404A和404B中的表示包括与道路对齐的车辆,并且车辆之间的距离是有意义的。此外,除了树木、行人和建筑物的位置和其他属性的细微变化之外,相机的位置以及视场也可能不同。图像404A和404B是2019年4月1日提交的美国临时申请No. 62/827,313和2018年11月16日提交的美国临时申请No. 62/768,846中图12的中间栏中的两个图像的线图表示,其全部内容通过引用合并于此。

[0060] 为了降低渲染和数值梯度的复杂度,训练分布变换器108顺序地优化场景的特定部分。例如,首先优化与汽车相关的属性,然后优化汽车和相机参数,然后将上下文元素(例如,建筑物、行人、树木等)的参数一起添加到训练中。因此,在本公开的实施例中,可以首先对一个或更多个参数进行优化,然后对附加参数进行优化,或者可以一次对所有参数进行优化。另外,分配和任务训练可以解耦(例如,关于图4A-4B的实验,首先根据过程200B对分布变换器108进行训练以用于分布,然后根据过程200C对分布变换器108进行针对对象检测的元任务的训练)。因此,在一些示例中,可以在重叠的时间执行针对分发和任务的训练,或者可以在不同的时间执行它们。所使用的任务网络是基于区域的卷积神经网络(RCNN),所述卷积神经网络用于目标检测调整。

[0061] 实验结果是使用0.5联合交集(IoU)处的平均精确度进行量化的,所述平均精确度测量两个边界之间的重叠。例如,根据过程100和200训练的任务网络预测可以是边界形状,并且KITTI验证集V可以包括相应的边界形状。这样,可以将任务网络在合成数据训练后预测的边界形状与来自验证集V的边界形状进行比较,以确定IoU。在使用场景图106渲染合成数据并训练任务网络的情况下,简单数据集的平均精确度为63.7,中等数据集的平均精确度为63.7,而困难数据集的平均精确度为62.2。相比之下,在仅使用过程200B的分布训练对汽车进行优化的变换的场景图110用于渲染合成数据和训练任务网络的情况下,简单数据集的平均精确度为66.4,中等数据集的平均精确度为66.5,而困难数据集的平均精确度为65.6。类似地,在仅使用过程200B的分布训练对汽车和相机参数进行优化的变换的场景图110用于渲染合成数据和训练任务网络的情况下,简单数据集的平均精确度为65.9,中等数

据集的平均精确度为66.3,困难数据集的平均精确度为65.9。作为另一个示例,其中仅使用过程200B的分布训练对汽车、相机参数和上下文特征进行优化的变换的场景图110用于渲染合成数据并训练任务网络的情况下,简单数据集的平均精确度为65.9,中等数据集的平均精确度为66.3,困难数据集的平均精确度是66.0。在又一个示例中,其中使用过程200B的分布训练和过程200C的任务训练对汽车、相机参数和上下文特征进行优化的变换的场景图110,用于渲染合成数据并训练任务网络,简单数据集的平均精确度为66.7,中等数据集的平均精确度为66.3,困难数据集的平均精确度为66.2。因此,对于每个不同的训练类型和参数优化类型,当使用分布变换器108来生成变换的场景图110时的结果优于仅使用来自概率语法102的场景图106时的结果。

[0062] 此外,由于验证集V包括标记的训练数据,因此有效的基线是在V上训练模型(例如,来自KITTI训练数据的一百张图像)。根据过程200B,通过仅使用V训练并使用V进行微调,使用从概率语法102采样的场景图106的平均精确度在简单数据集上得到71.3,在中等数据集上得到72.7,在困难数据集上得到72.7。相反,根据过程200B,通过仅使用V训练并使用V进行微调,使用由分布变换器108变换的变换的场景图110的平均精确度在简单数据集上得到72.4,在中等数据集上得到73.9,在困难数据集上得到73.9。因此,这些数量训练结果表明,即使对于根据过程200B的属性分布差距未针对分布变量进行优化,过程200C的任务训练也可提高准确性。另外,在同时实施过程200B的分布训练和过程200C的任务训练的情况下,实现了最准确的结果。

[0063] 现在参考图5A-5B,图5A-5B是示出了根据本公开的一些实施例的当使用与基线模型相比由分布变换器模型生成的合成数据集时下游任务网络性能的示例说明。例如,图像502A和502B示出了根据本文所述的过程100和200使用变换的场景图110训练的任务网络的预测,而图像504A和504B仅说明使用场景图106生成的场景训练的相同任务网络的预测,而不使用分布变换器108。如图5A-5B所示,图像502A和502B比图像504A和504B描绘了更精确的对象检测。图像502A和502B是从图14的底部列采样的线条图,并且图像504A和504B是从2019年4月1日提交的美国临时申请No.62/827,313和2018年11月16日提交的美国临时申请No.62/768,846的图14的顶部栏中采样的线条图,所述专利通过引用整体并入本文。

[0064] 现在参考图6-7,本文所述的方法600和700的每个框包括可以使用硬件、固件和/或软件的任何组合来执行的计算过程。例如,各种功能可以由执行存储在存储器中的指令的处理器来实现。所述方法600和700也可以具体化为存储在计算机存储介质上的计算机可用指令。仅举几例,所述方法600和700可以由独立应用程序、服务或托管服务(独立地或者与另一个托管服务结合地)或者另一个产品的插件提供。另外,通过示例的方式,关于图1和图2A-2C的处理器描述了方法600和700。然而,附加地或者可替代地,这些方法600和700可以由在任何一个过程中的任意一种系统(例如,使用图8的示例计算设备800的某些或全部组件)或者任意系统或过程组合(包括但不限于本文描述的那些系统)执行。

[0065] 图6是示出了根据本公开的一些实施例的用于训练分布变换器模型以计算变换的场景图方法600的流程图。在框B602处,方法600包括从场景语法中采样场景图。例如,可以从概率语法102中采样场景图106。

[0066] 在框B604处,方法600包括将表示所述场景图的第一数据应用于第一机器学习模型。例如,场景图106可以应用于分布变换器108。

[0067] 在框B606处,方法600包括使用所述第一机器学习模型来计算表示变换的场景图的第二数据。例如,分布变换器108可以至少部分地基于对场景图106的处理来计算变换的场景图110。

[0068] 在框B608处,方法600包括渲染表示图像的第三数据。例如,渲染器可以渲染与合成数据集相对应的图像114。

[0069] 在框B610处,方法600包括生成表示对应的地面实况的第四数据。例如,渲染器112或另一组件可以通过使用图像数据、变换的场景图110和/或其他信息来生成地面实况116。

[0070] 在框B612处,方法600包括使用第三数据和第四数据训练第二机器学习模型。例如,任务网络训练器118可以使用图像114和地面实况116来训练任务网络。

[0071] 图7是示出了根据本公开的一些实施例的用于使用分布变换器模型生成的合成数据集来训练下游任务网络方法700的流程图。在框B702处,方法700包括使用机器学习模型来计算表示变换的场景图的第一数据。例如,分布变换器108可以使用将来自概率语法102采样的场景图106作为输入,来生成变换的场景图110。

[0072] 在框B704处,方法700包括至少部分地基于第一数据渲染表示合成图像的第二数据。例如,可以使用渲染器112,并且至少部分地基于变换的场景图110来渲染所生成的合成场景。

[0073] 在框B706处,方法700包括通过将合成属性分布与真实世界属性分布进行比较来确定差异。例如,可以通过将对应于生成合成场景210的分布214与对应于真实场景212的分布216进行比较来确定MMD 218。

[0074] 在框B708处,方法700包括至少部分地基于所述差异,生成表示网络更新信息的第三数据。例如,MMD 218可用于生成地面实况或其他网络训练信息,用于更新分布变换器108的参数。

[0075] 在框B710处,方法700包括使用第三数据训练机器学习模型。例如,可以将由此产生的差异或其他网络更新信息反向传播到分布变换器108,以更新分布变换器的参数。结果,随着时间的推移,与所生成的合成场景210相对应的分布214可以变得与真实场景212相对应的分布216更相似。

[0076] 图8为适合用于实现本公开的一些实施例的示例计算设备800的框图。计算设备800可以包括直接或间接耦合下列设备的总线802:内存804,一个或更多个中央处理单元(CPU) 806,一个或更多个图形处理单元(GPU) 808,通信接口810,输入/输出(I/O) 端口812,输入/输出组件814,电源816,以及一个或更多个呈现组件818(例如显示器)。

[0077] 尽管图8的各个框被示为经由具有线路的总线802连接,但是这并不意图是限制性的,并且仅仅为了清楚起见。例如,在一些实施例中,诸如显示设备之类的呈现组件818可以被认为是I/O组件814(例如如果显示器为触摸屏)。作为另一个示例,CPU 806和/或GPU 808可以包括内存(例如,内存804可以表示除了GPU 808、CPU 806和/或其他组件的内存以外的存储设备)。换言之,图8的计算设备仅仅是说明性的。在诸如“工作站”、“服务器”、“膝上型电脑”、“台式机”、“平板电脑”、“客户端设备”、“移动设备”、“手持式设备”、“游戏控制台”、“电子控制单元(ECU)”、“虚拟现实系统”和/或其他设备或系统类型之类的类别之间不进行区分,因为所有这些都考虑在图8的计算设备的范围内。

[0078] 总线802可以表示一条或更多条总线,例如地址总线、数据总线、控制总线或者其

组合。总线802可以包括一种或更多种总线类型,例如行业标准架构 (ISA) 总线、扩展行业标准架构 (EISA) 总线、视频电子标准协会 (VESA) 总线、外围组件互连 (PCI) 总线、外围组件互连快速 (PCIe) 总线,和/或另一种类型的总线。

[0079] 内存804可以包括各种各样的计算机可读介质中的任何介质。计算机可读介质可以是可由计算设备800访问的任何可用介质。计算机可读介质可以包括易失性和非易失性介质以及可移除和不可移除介质。举例而言且非限制性地,计算机可读介质可以包括计算机存储介质和通信介质。

[0080] 计算机存储介质可以包括易失性和非易失性介质和/或可移除和不可移除介质,其以用于存储诸如计算机可读指令、数据结构、程序模块和/或其他数据类型之类的信息的任何方法或技术实现。例如,内存804可以存储计算机可读指令(例如其表示程序和/或程序元素,例如操作系统)。计算机存储介质可以包括但不限于RAM、ROM、EEPROM、闪存或者其他存储技术、CD-ROM、数字多功能盘 (DVD) 或其他光盘存储装置、磁带盒、磁带、磁盘存储装置或其他磁存储设备,或者可以用来存储期望的信息且可以由计算设备800访问的任何其他介质。当在本文使用时,计算机存储介质并不包括信号本身。

[0081] 通信介质可以在诸如载波之类的调制数据信号或其他传输机制中包含计算机可读指令、数据结构、程序模块和/或其他数据类型,并且包括任何信息输送介质。术语“调制数据信号”可以指这样的信号,该信号使它的特性中的一个或多个以这样的将信息编码到该信号中的方式设置或改变。举例而言且非限制性地,通信介质可以包括诸如有线网络或直接有线连接之类的有线介质,以及诸如声音、RF、红外和其他无线介质之类的无线介质。任何以上所述的组合也应当包含在计算机可读介质的范围内。

[0082] CPU 806可以被配置为执行计算机可读指令以便控制计算设备800的一个或多个组件执行本文描述的方法和/或过程中的一个或多个。CPU806中的每一个可以包括能够同时处理大量软件线程的一个或多个核(例如一个、两个、四个、八个、二十八个、七十二个等等)。CPU 806可以包括任何类型的处理器,并且可以包括不同类型的处理器,这取决于实现的计算设备800的类型(例如具有用于移动设备的较少核的处理器以及具有用于服务器的更多核的处理器)。例如,取决于计算设备800的类型,处理器可以是使用精简指令集计算 (RISC) 实现的ARM处理器或者使用复杂指令集计算 (CISC) 实现的x86处理器。除了一个或多个微处理器或者诸如数学协处理器之类的补充协处理器之外,计算设备800还可以包括一个或多个CPU 806。

[0083] GPU 808可以由计算设备800用来渲染图形(例如3D图形)。GPU 808可以包括能够同时处理数百或数千个软件线程的数百或数千个核。GPU808可以响应于渲染命令(例如经由主机接口接收的来自CPU 806的渲染命令)而生成用于输出图像的像素数据。GPU 808可以包括诸如显示内存之类的用于存储像素数据的图形内存。显示内存可以作为内存804的部分而被包括。GPU 808可以包括(例如经由链路)并行操作的两个或更多GPU。当组合在一起时,每个GPU 808可以生成用于输出图像的不同部分或者用于不同输出图像的像素数据(例如,第一GPU用于第一图像,第二GPU用于第二图像)。每个GPU可以包括它自己的内存,或者可以与其他GPU共享内存。

[0084] 在其中计算设备800不包括GPU 808的示例中,CPU 806可以用来渲染图形。在任何实施例中,CPU 806和/或GPU 808可以与诸如游戏引擎之类的图形引擎结合使用以渲染合

成数据集D(θ),以用于确定MMD 218和/或训练任务网络。例如,由于GPU 808具有渲染照片级逼真的图像的能力,尤其是使用诸如光线追踪之类的技术,因此合成数据集不仅在上下文方面而且在视觉外观上都可以更类似于真实世界的场景(例如,缩小域差距)。

[0085] 通信接口810可以包括一个或更多个接收器、发送器和/或收发器,其使得计算设备800能够经由电子通信网络与其他计算设备通信,包括有线和/或无线通信。通信接口810可以包括使能够通过若干不同网络中的任何网络进行通信的组件和功能,所述网络例如无线网络(例如Wi-Fi、Z波、蓝牙、蓝牙LE、ZigBee等等)、有线网络(例如通过以太网通信)、低功率广域网(例如LoRaWAN、SigFox等等)和/或因特网。

[0086] I/O端口812可以使得计算设备800能够逻辑地耦合到包括I/O组件814、呈现组件818和/或其他组件在内的其他设备,其中一些可以内置到(例如集成到)计算设备800中。说明性I/O组件814包括麦克风、鼠标、键盘、操纵杆、游戏垫、游戏控制器、碟形卫星天线、扫描仪、打印机、无线设备等等。I/O组件814可以提供处理用户生成的空中手势、语音或其他生理输入的自然用户接口(NUI)。在一些实例中,输入可以传输至适当的网络元件以便进一步处理。NUI可以实现语音识别、手写笔识别、面部识别、生物特征识别、屏幕上和邻近屏幕的手势识别、空中手势、头部和眼睛跟踪以及与计算设备800的显示器关联的触摸识别(如下文更详细地描述的)的任意组合。计算设备800可以包括诸如立体照相机系统之类的深度照相机、红外照相机系统、RGB照相机系统、触摸屏技术以及这些的组合,以用于手势检测和识别。此外,计算设备800可以包括使能运动检测的加速度计或陀螺仪(例如作为惯性测量单元(IMU)的部分)。在一些示例中,加速度计或陀螺仪的输出可以由计算设备800用来渲染沉浸式增强现实或者虚拟现实。

[0087] 电源816可以包括硬接线电源、电池电源或者其组合。电源816可以向计算设备800供电以使得计算设备800的组件能够操作。

[0088] 呈现组件818可以包括显示器(例如监视器、触摸屏、电视屏幕、平视显示器(HUD)、其他显示器类型或者其组合)、扬声器和/或其他呈现组件。呈现组件818可以接收来自其他组件(例如GPU 808、CPU 806等等)的数据,并且输出该数据(例如作为图像、视频、声音等等)。

[0089] 本公开可以在由计算机或者诸如个人数字助理或其他手持式设备之类的其他机器执行的、包括诸如程序模块之类的计算机可执行指令的机器可使用指令或者计算机代码的一般背景下进行描述。通常,包括例程、程序、对象、组件、数据结构等等的程序模块指的是执行特定任务或者实现特定抽象数据类型的代码。本公开可以在各种各样的系统配置中实践,这些配置包括手持式设备、消费电子器件、通用计算机、更专业的计算设备等等。本公开也可以在其中任务由通过通信网络链接的远程处理设备执行的分布式计算环境(例如,云环境)中实践。例如,过程100和200或其部分可以使用云基础设施来执行,例如能够进行有效图形处理和渲染的基于云的GPU服务器。

[0090] 如在本文中使用的,“和/或”关于两个或更多元素的叙述应当解释为仅指一个元素或者元素组合。例如,“元素A、元素B和/或元素C”可以包括仅仅元素A,仅仅元素B,仅仅元素C,元素A和元素B,元素A和元素C,元素B和元素C,或者元素A、B和C。此外,“元素A或元素B中的至少一个”可以包括元素A中的至少一个,元素B中的至少一个,或者元素A中的至少一个和元素B中的至少一个。进一步,“元素A和元素B中的至少一个”可以包括元素A中的至少

一个,元素B中的至少一个,或者元素A中的至少一个和元素B中的至少一个。

[0091] 这里详细地描述了本公开的主题以满足法定要求。然而,描述本身并非意在限制本公开的范围。相反地,本发明人已经设想到,要求保护的主题也可以以其他的方式具体化,以包括与本文中结合其他当前或未来技术描述的步骤不同的步骤或者相似的步骤的组合。而且,尽管术语“步骤”和/或“块”在本文中可以用来隐含采用的方法的不同元素,但是这些术语不应当被解释为暗示本文公开的各个步骤之中或之间的任何特定顺序,除非明确描述了各步骤的顺序。

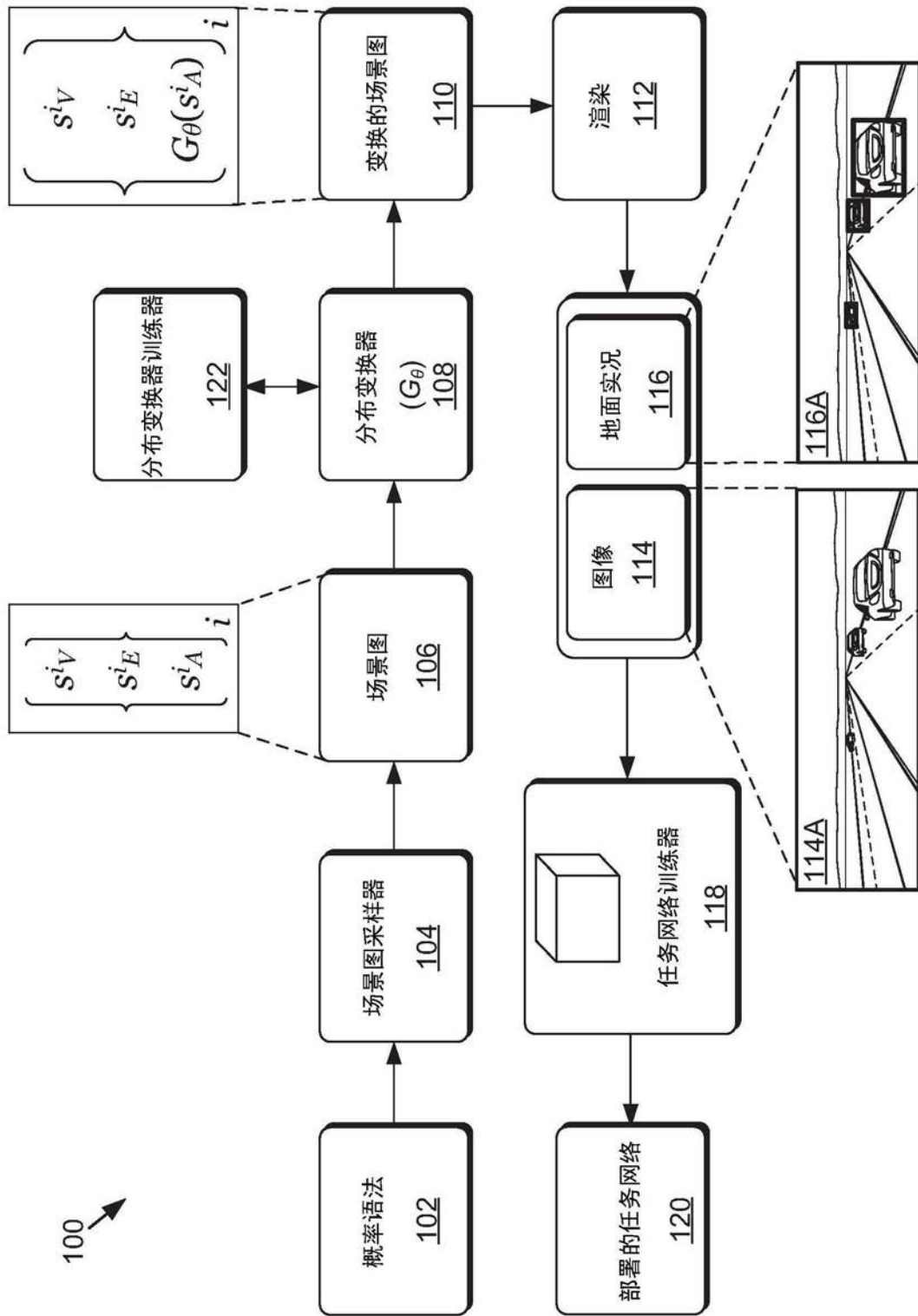


图1

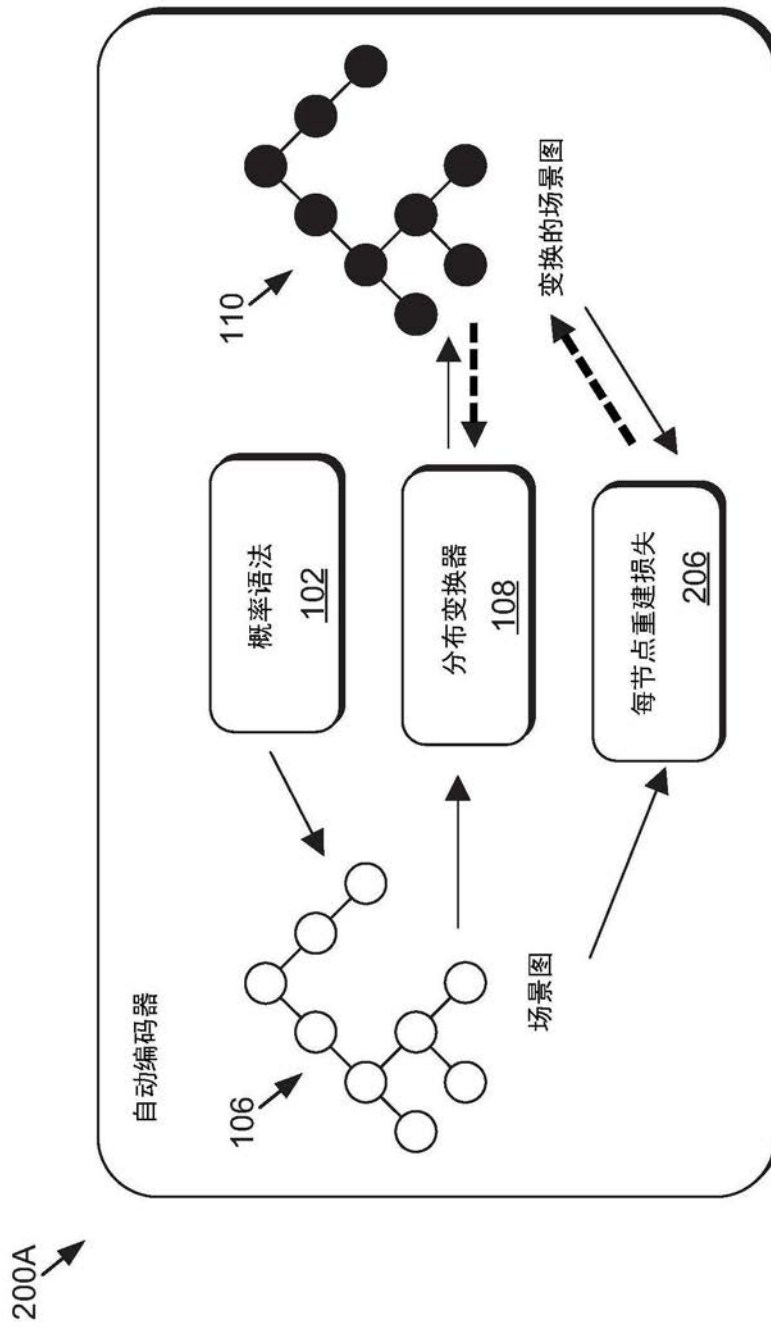


图2A

200B

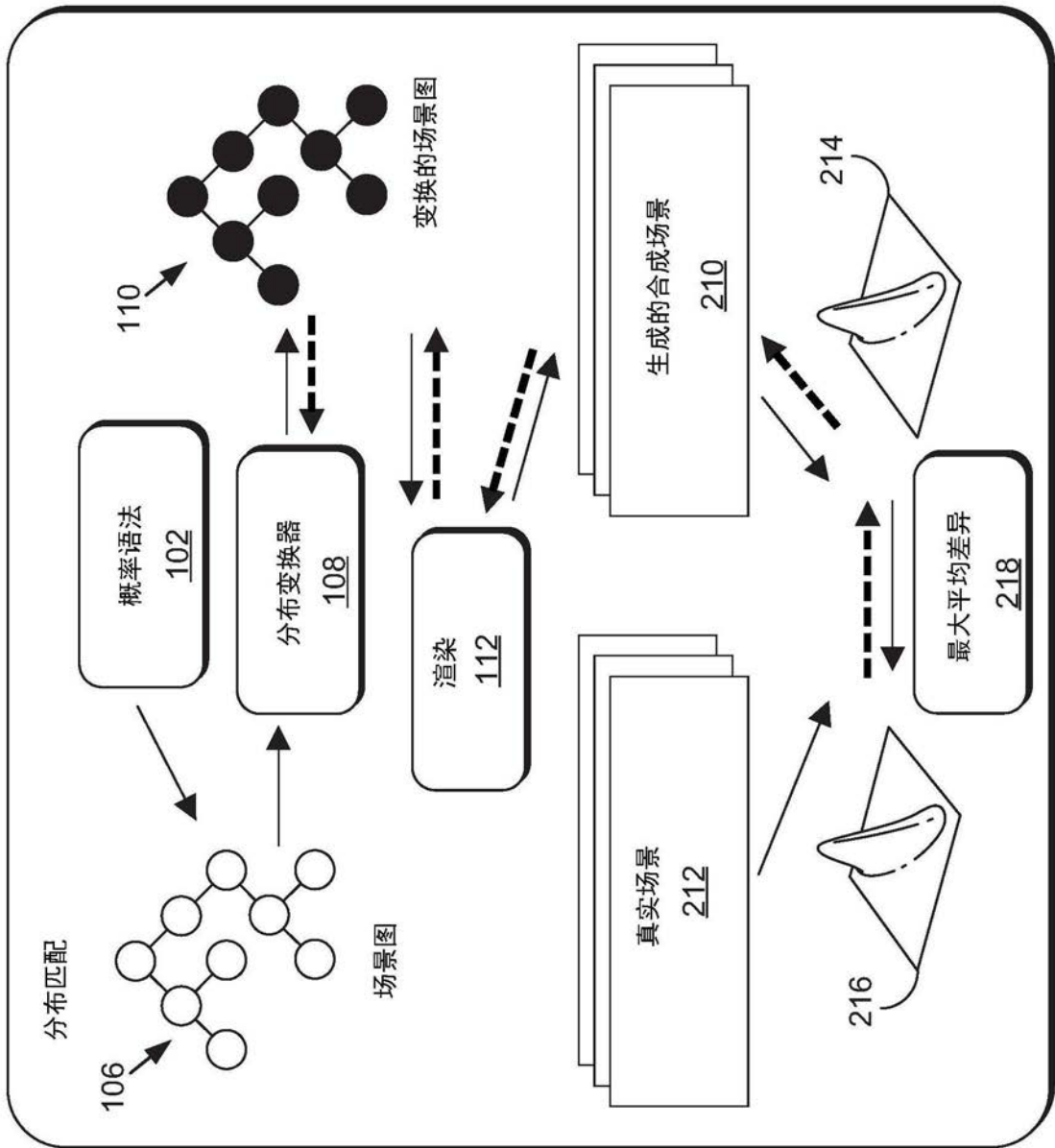


图2B

200C

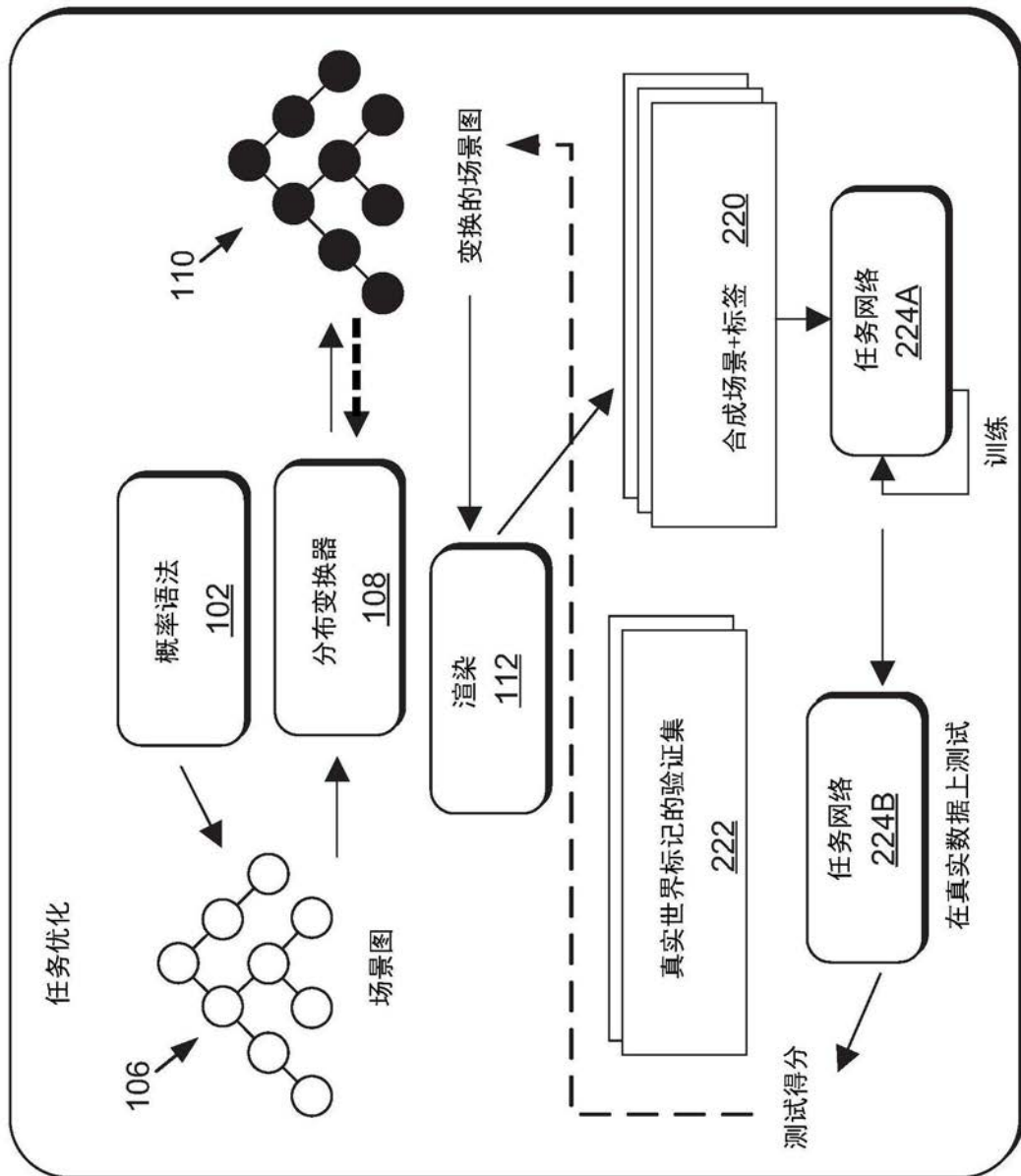


图2C

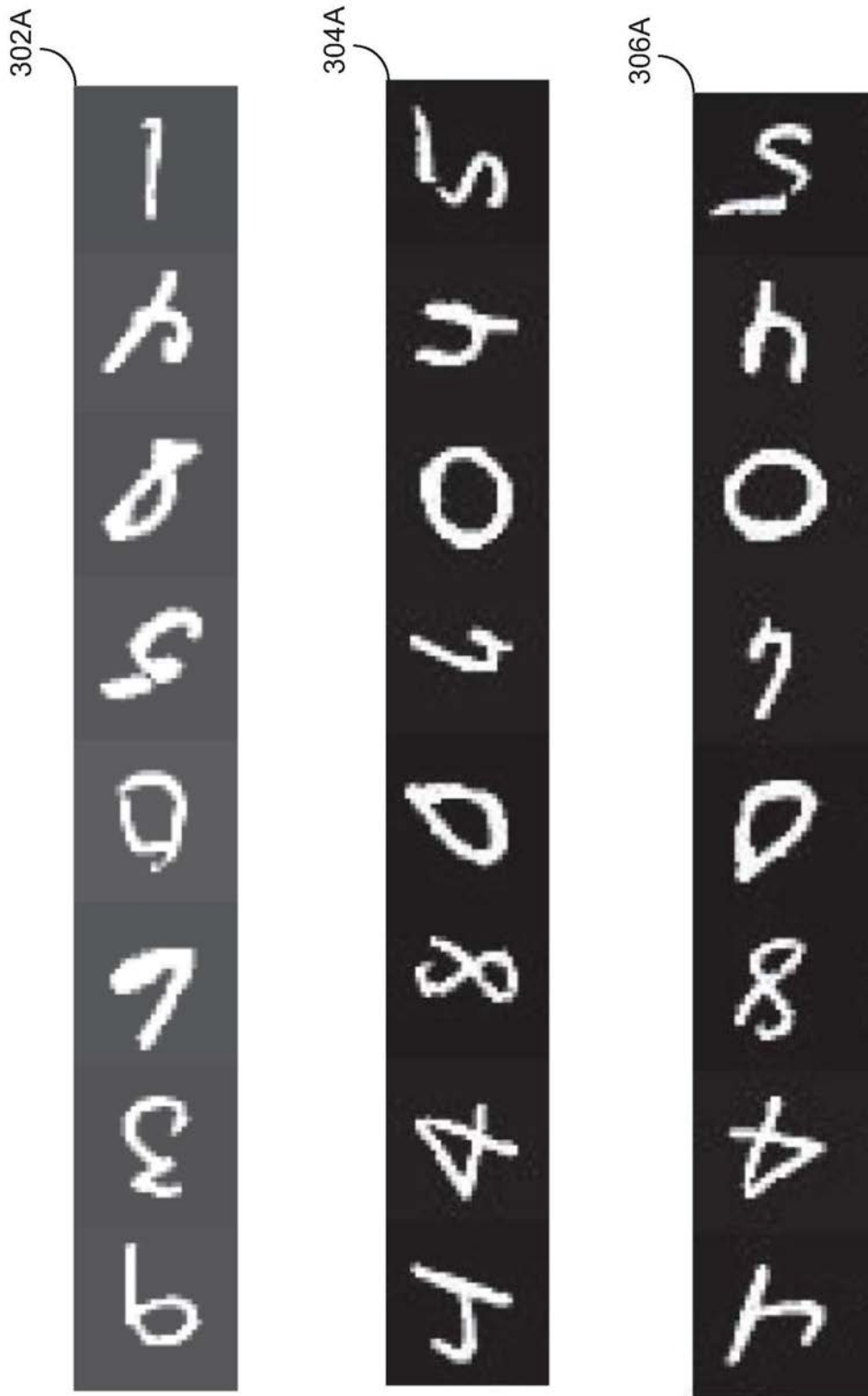


图3A

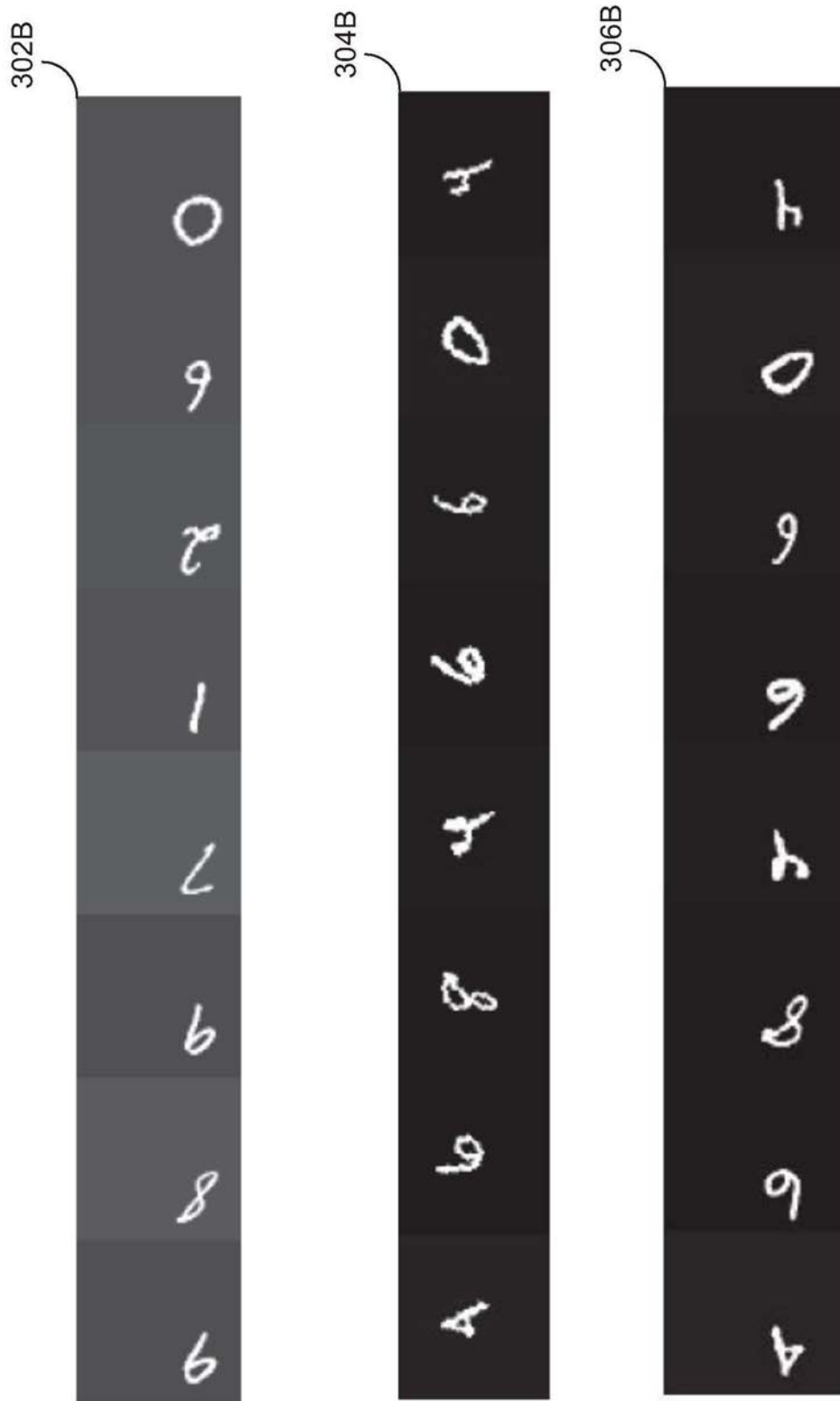


图3B

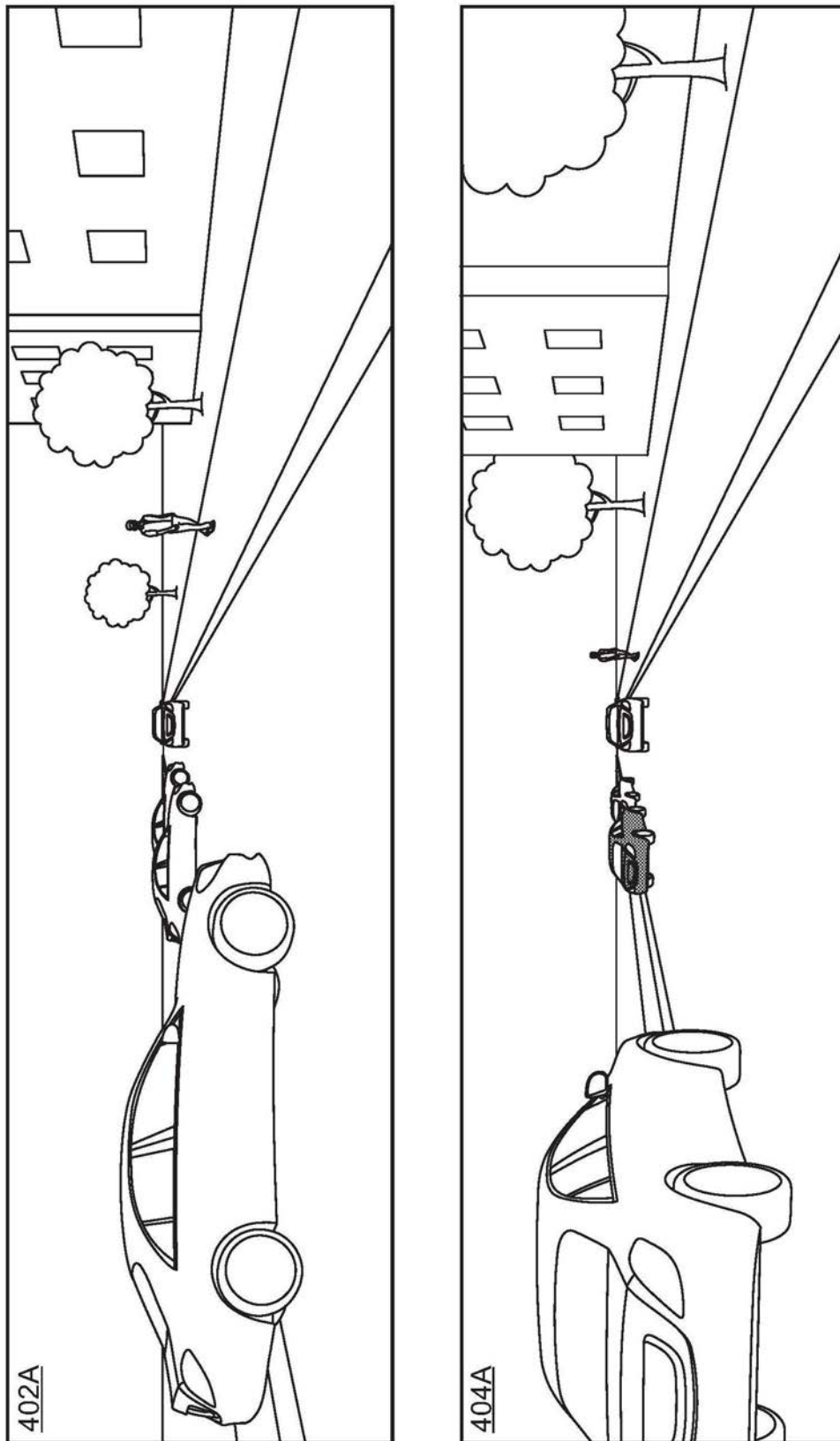


图4A

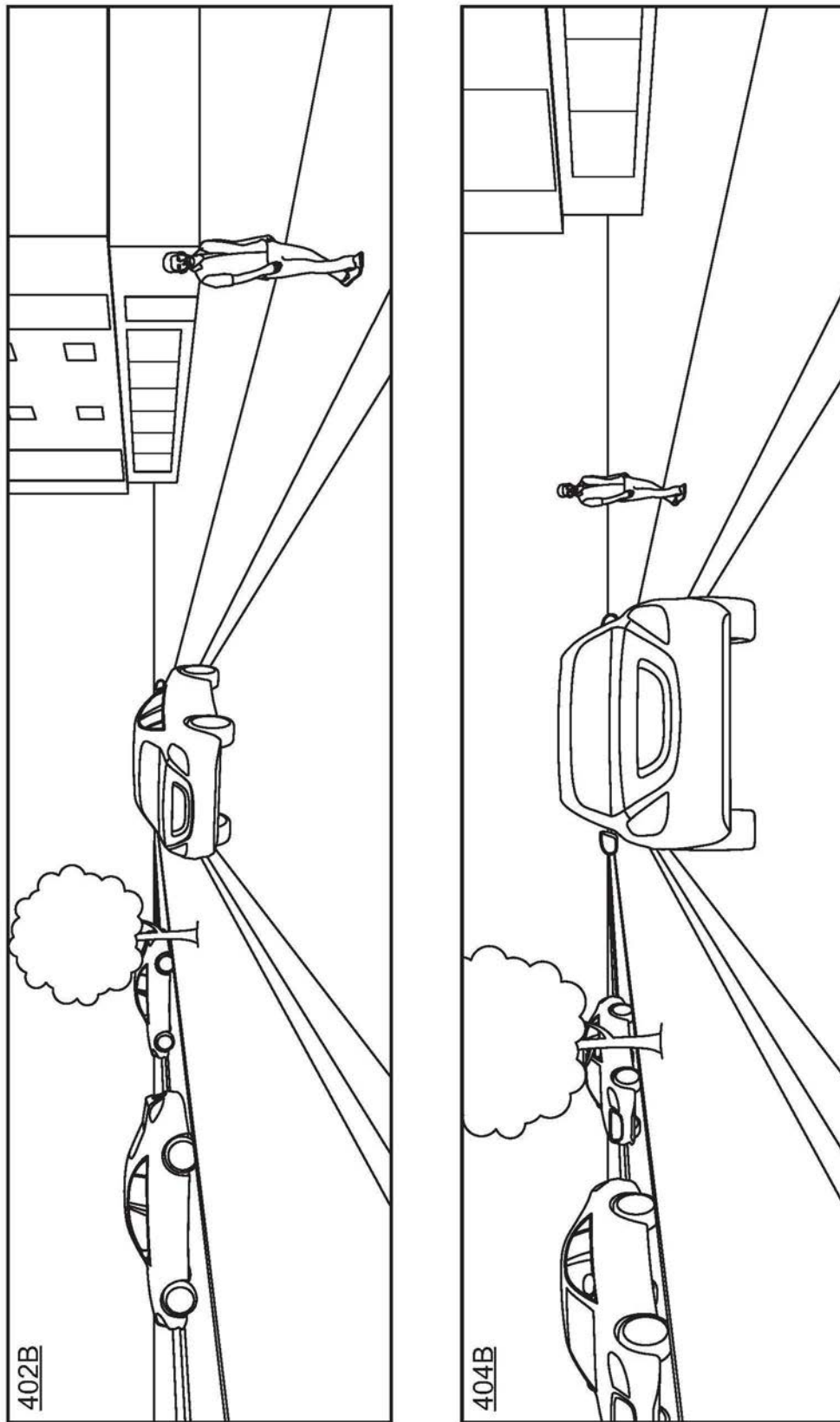


图4B

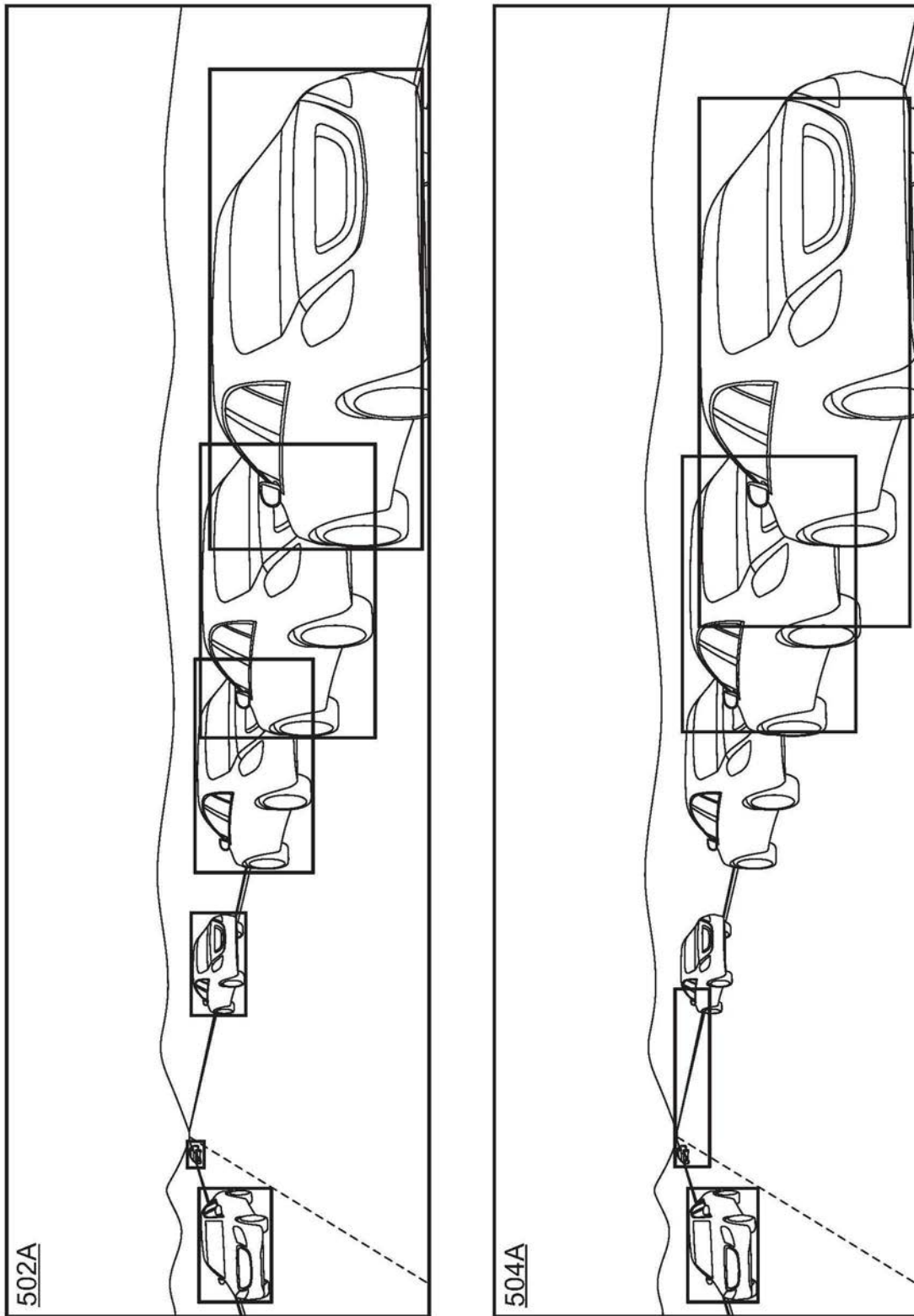


图5A

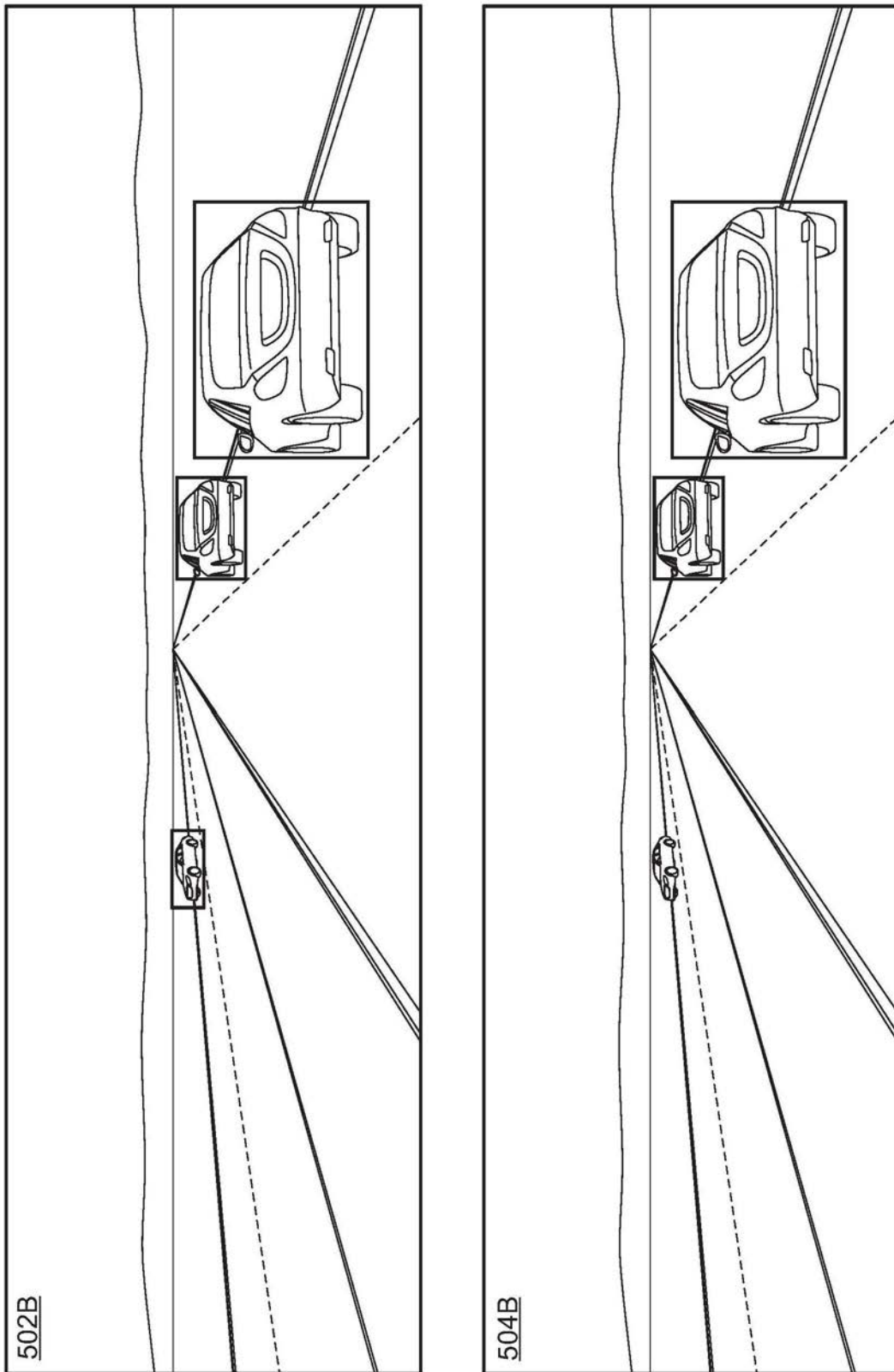


图5B



图6



图7

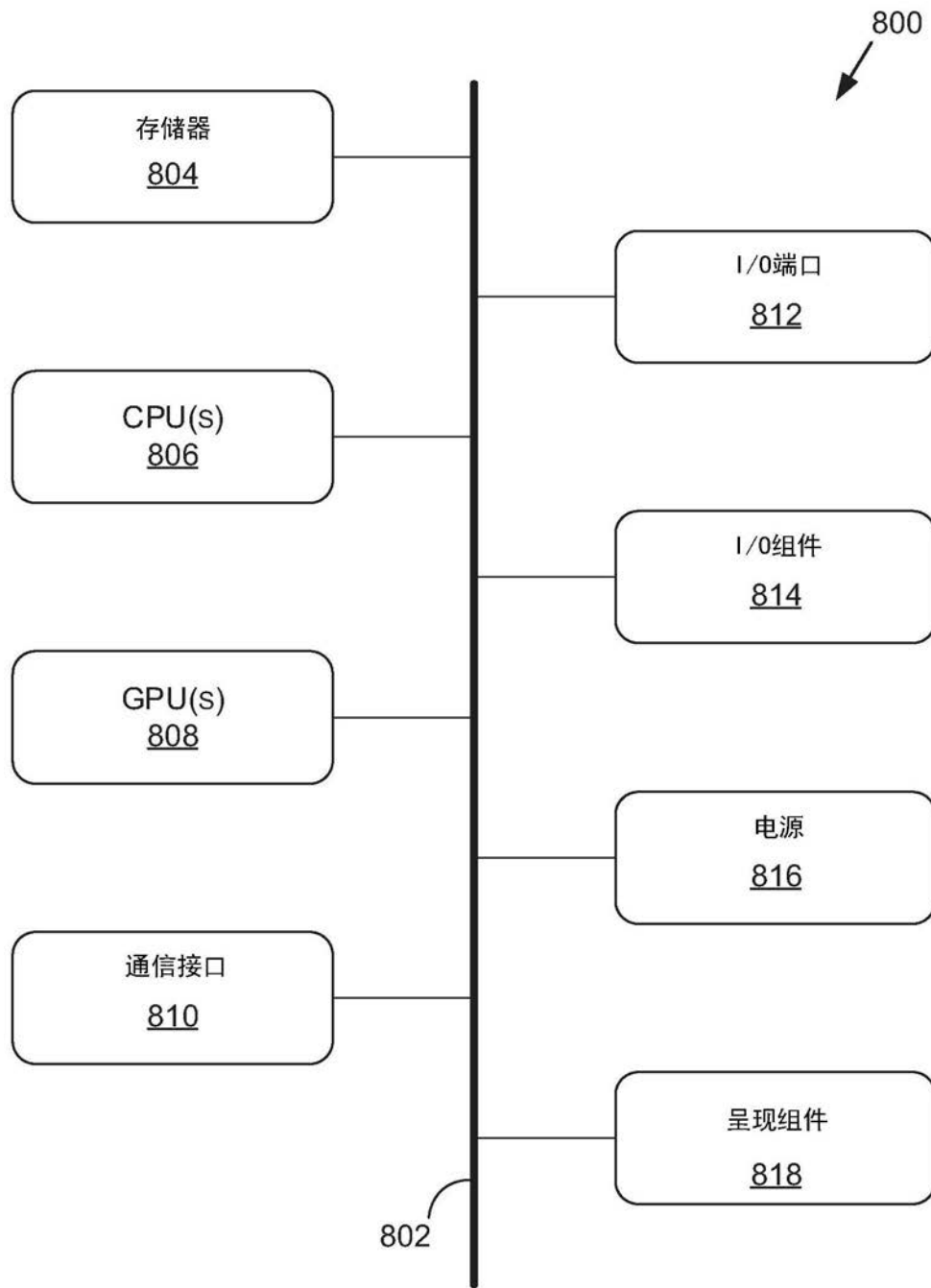


图8