



(12) 发明专利申请

(10) 申请公布号 CN 116206042 A
(43) 申请公布日 2023. 06. 02

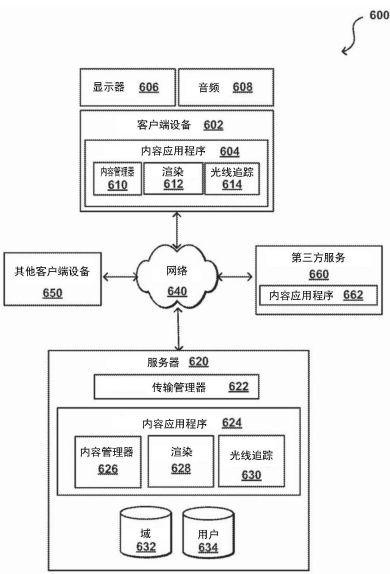
(21) 申请号 202210893578.9
(22) 申请日 2022.07.27
(30) 优先权数据
17/536,531 2021.11.29 US
(71) 申请人 辉达公司
地址 美国加利福尼亚州
(72) 发明人 P·A·戈特龙 C·A·瓦克泰
(74) 专利代理机构 北京市磐华律师事务所
11336
专利代理师 高伟
(51) Int.Cl.
G06T 15/06 (2011.01)
G06N 5/04 (2023.01)
G06N 3/08 (2023.01)
G06N 20/00 (2019.01)

权利要求书2页 说明书32页 附图14页

(54) 发明名称
空间哈希一致采样

(57) 摘要

本公开涉及空间哈希一致采样。本文提出的方法减少在模拟中使用空间哈希时出现诸如闪烁之类的时间伪影。通过确保在算法的单独执行中使用相同的模拟点,即使在执行顺序可能不同的情况下,也可以至少部分避免闪烁。这可能涉及为空间哈希映射的每个哈希单元选择单个代表点,其中将针对该代表点位置执行该哈希单元的模拟,而不管执行顺序如何。此哈希映射条目的位置索引和选择器值都存储在一个值中,其中较低位存储位置索引,较高位表示选择器值。将选择器值存储在较高的权重位中可确保原子最大化操作将主要考虑选择器值,并且仅在选择器值相等的情况下使用位置索引。



1. 一种计算机实现的方法,包括:
将要渲染的内容中的位置子集分配给多个哈希映射条目;
确定相应哈希映射条目中要从其执行模拟的位置;
将用于所述相应哈希映射条目的选择器值和位置索引值存储在组合值中;以及
提供所述组合值以确定在所述模拟期间要针对所述相应哈希映射条目执行的计算操作的所述位置。
2. 根据权利要求1所述的计算机实现的方法,其中所述组合值中较低位数存储所述位置索引,较高位数存储所述选择器值。
3. 根据权利要求1所述的计算机实现的方法,还包括:
通过对所述要渲染的内容的位置集执行空间哈希来确定所述多个哈希映射条目。
4. 根据权利要求1所述的计算机实现的方法,其中所述位置对应于要渲染的图像中的像素,并且其中要执行的所述操作涉及从由所述组合哈希值确定的像素位置追踪用于相应哈希映射条目的光线。
5. 根据权利要求1所述的计算机实现的方法,其中所述选择器值是使用哈希函数或伪随机数或确定性数字生成器生成的最大值或最小值。
6. 根据权利要求1所述的计算机实现的方法,还包括:
从所确定的位置对所述相应哈希映射条目执行所述模拟,其中所确定的位置对于所述内容的一组实例中的每一个都是相同的,而不论所述计算操作的执行计划如何。
7. 根据权利要求1所述的计算机实现的方法,其中为所述位置计算的数据被应用于针对所述相应哈希映射条目中其他位置的相关点。
8. 一种系统,包括:
至少一个处理器;以及
存储器,其包括指令,所述指令在由所述至少一个处理器执行时,使所述系统:
将要生成的内容的位置子集分配给多个哈希映射单元;
根据确定的选择器值,确定相应哈希映射单元内要从其执行模拟的位置;
将针对相应哈希映射单元的所确定的位置的所述选择器值和索引值存储在组合值中;
以及
提供所述组合值,以确定所述模拟期间针对所述相应哈希映射单元的计算操作的所述位置。
9. 根据权利要求8所述的系统,其中所述组合值中较低位数存储所述位置索引,较高位数存储所述选择器值。
10. 根据权利要求8所述的系统,其中所述指令在被执行时进一步使所述系统:
通过对所述要生成的内容的位置集执行空间哈希来确定所述多个哈希映射单元。
11. 根据权利要求8所述的系统,其中所述位置对应于要渲染的图像中的像素,并且其中要执行的所述操作涉及从由所述组合哈希值确定的像素位置追踪用于相应哈希映射条目的光线。
12. 根据权利要求8所述的系统,其中所述选择器值是使用哈希函数或伪随机数或确定性数字生成器生成的最大值或最小值。
13. 根据权利要求8所述的系统,其中所述指令在被执行时进一步使所述系统:

从所确定的位置对所述相应哈希映射单元执行所述模拟,其中所确定的位置对于所述内容的一组实例中的每一个都是相同的,而不论所述计算操作的执行计划如何。

14. 根据权利要求8所述的系统,其中针对所述位置确定的模拟数据被应用于针对所述相应哈希映射单元中其他位置的相关点。

15. 根据权利要求8所述的系统,其中所述系统包括以下至少一个:

用于执行模拟操作的系统;

用于执行模拟操作以测试或验证自主机器应用程序的系统;

用于渲染图形输出的系统;

用于执行深度学习操作的系统;

使用边缘设备实现的系统;

包含一个或多个虚拟机 (VM) 的系统;

至少部分地在数据中心中实现的系统;或者

至少部分地使用云计算资源实现的系统。

16. 一种非暂时性计算机可读存储介质,其包括指令,所述指令如果由一个或多个处理器执行,使得所述一个或多个处理器:

将要渲染的图像的像素子集分配给多个哈希映射单元;

根据所确定的选择器值,确定相应哈希映射单元内的要执行光线追踪的像素位置;

将用于相应哈希映射单元的所确定的像素位置的选择器值和像素索引值存储在组合值中;以及

提供所述组合值,以确定光线追踪期间所述相应哈希映射单元的执行线程的像素位置,其中从所述像素位置追踪所述相应哈希映射单元的光线。

17. 根据权利要求17所述的非暂时性计算机可读存储介质,其中所述指令如果被执行,则进一步使得所述一个或多个处理器:

使用所述组合值中较低位数存储所述像素索引,使用较高位数存储所述选择器值。

18. 根据权利要求17所述的非暂时性计算机可读存储介质,其中所述指令如果被执行,则进一步使得所述一个或多个处理器:

通过对所述要渲染的图像的一组像素执行空间哈希来确定所述多个哈希映射单元。

19. 根据权利要求17所述的非暂时性计算机可读存储介质,其中所述指令如果被执行,则进一步使得所述一个或多个处理器:

确定要包括在所述多个哈希映射单元的各个单元中的像素的数量。

20. 根据权利要求17所述的非暂时性计算机可读存储介质,其中所述选择器值是使用哈希函数或伪随机数生成器生成的最大值或最小值。

空间哈希一致采样

背景技术

[0001] 空间哈希是一种实用工具,可用于在大规模并行环境中高效存储和检索稀疏空间数据。哈希函数本质上可以在整个哈希映射中相对均匀地传播信息。在渲染图像的情况下,空间哈希映射条目可以覆盖要生成的最终图像中的几个像素。可以在单独的计算步骤中处理这些像素中的每一个,其中每个线程向哈希映射添加信息。例如,通过追踪穿过虚拟场景的样本G 光线,每个线程可能有助于对半球函数的积分进行蒙特卡罗估计。当这些样本取自伪随机或确定性序列(如准蒙特卡罗(QMC))时,每个线程将覆盖该序列的一段。根据这些线程的调度,QMC序列段到线程的映射可能会由于时间顺序的不同而从一个执行更改为另一个执行,因为调度可能是不确定的。由于每个像素对应于要渲染的场景中的至少一个相应点,因此执行之间的这种更改通常会导致追踪不同的光线集,这可能导致时间伪影,例如重复执行相同的处理时的图像闪烁。

附图说明

- [0002] 将参考附图描述根据本公开的各种实施例,其中:
- [0003] 图1A、1B和1C示出了根据各种实施例的包含闪烁伪影的图像;
- [0004] 图2A、2B和2C示出了根据各种实施例的可以为图像中的表面确定的哈希单元;
- [0005] 图3A和3B示出了根据至少一个实施例的确定用于光线追踪的代表点的示例方法;
- [0006] 图4A和4B示出了根据至少一个实施例的用于存储点确定数据的方法;
- [0007] 图5示出了根据至少一个实施例的用于在图像渲染中执行光线追踪的过程;
- [0008] 图6示出了根据至少一个实施例的可用于确定和/或执行任务的分布式系统的组件;
- [0009] 图7A示出了根据至少一个实施例的推理和/或训练逻辑;
- [0010] 图7B示出了根据至少一个实施例的推理和/或训练逻辑;
- [0011] 图8示出了根据至少一个实施例的示例数据中心系统;
- [0012] 图9示出了根据至少一个实施例的计算机系统;
- [0013] 图10示出了根据至少一个实施例的计算机系统;
- [0014] 图11示出了根据一个或多个实施例的图形处理器的至少部分;
- [0015] 图12示出了根据一个或多个实施例的图形处理器的至少部分;
- [0016] 图13是根据至少一个实施例的高级计算管线的示例数据流图;
- [0017] 图14是根据至少一个实施例的用于在高级计算管线中训练、适配、实例化和部署机器学习模型的示例系统的系统图;以及
- [0018] 图15A和图15B示出了根据至少一个实施例的用于训练机器学习模型的过程的数据流图,以及利用预训练的注释模型来增强注释工具的客户端-服务器架构。

具体实施方式

[0019] 在以下描述中,将描述各种实施例。为了解释的目的,阐述了具体配置和细节,以

提供对实施例的彻底理解。然而,对于本领域技术人员来说,实施例可以在没有具体细节的情况下实施也是显而易见的。此外,可以省略或简化公知特征,以便不模糊所描述的实施例。

[0020] 根据各种实施例的方法可用于生成基本上没有至少某些类型的伪影的内容,其中所生成的内容可包括一个或更多个图像、视频、纹理图、增强现实 (AR) 或虚拟现实 (VR) 内容,或其他此类二维或三维内容,以及其他类型的输出,例如,一个或更多个光探头。在至少一个实施例中,这可以包括在使用空间哈希模拟光传输或其他形式的计算机生成的视觉时减少伪影(例如闪烁)的存在。通过确保在算法的单独执行中追踪同一组光线,即使执行顺序可能不同,也可以至少部分地避免闪烁。这可能涉及为一组哈希单元格中的每一个选择一个代表点。空间哈希映射条目可以覆盖诸如图像之类的内容实例的多个像素或其他点或区域。可以在该哈希映射条目所跨越的区域内选择一个点,从该位置追踪所有光线(或模拟光传输),而不管线程调度或计算顺序如何。可以为该哈希映射条目存储像素索引和选择器值。可以使用伪随机函数为哈希映射项覆盖的每个点计算不同的选择器值。条目的选择器值存储迄今为止遇到的最大(或最小)选择器值,而像素索引存储已获得此选择器的像素的索引。为了更新选择器和像素索引,可以将这两个值捆绑在一起,形成一个32位的值,其中较低的 25位存储像素索引,较高的7位表示选择器值。将选择器值存储在较高的权重位中可确保原子最大值操作将主要考虑选择器值,并且仅在选择器值相等的情况下使用像素索引。例如,在模拟或渲染期间,可以为每个执行线程定位相应的条目,可以确定对应于该条目的像素索引,并且可以相应地获取虚拟场景的附加空间信息,例如该像素处的曲面位置和法线。然后,该线程可以原子地保留伪随机或确定性序列的一段,并追踪其采样光线。由于给定条目的所有线程都将追踪来自同一位置的光线,因此调度顺序变得无关紧要,如果以不同的顺序重新运行相同的计算步骤,则生成的图像将保持一致。

[0021] 在各种实施例的范围内,也可以使用各种其他这样的功能,这对于本领域的普通技术人员来说,根据本文包含的教导和建议是显而易见的。

[0022] 图1A-1C示出了图像或视频帧序列中闪烁伪影的示例。在图1A中的第一图像100中,示出了由场景中的一个或更多个虚拟光源照明的立方体 102。如上所述,光模拟可用于确定如何照亮或遮蔽该立方体的每个像素,这包括至少部分基于例如立方体上该点的照明量的计算积分的估计来确定颜色或像素值。这种模拟也可用于确定其他信息,如可能与全局照明、环境光遮挡、着色器效果等相关的信息。应该理解,立方体类型的对象可能不会如图所示频繁地展示闪烁,但为了简单解释,给出了此示例。由于在许多情况下,对所有入射光线进行采样或以解析方式计算所有像素的积分是不现实的,特别是对于诸如在线游戏之类的时间敏感的应用程序,通常会执行一定量的采样,例如作为诸如照明等方面的代表性度量,然后也可以应用于附近的像素。这可以包括将代表性照明值应用于对象的哈希单元中的像素。然而,如图1B和图1C的后续图像120、140中所示,在生成不同的图像或视频帧时采样不同的光线可导致确定不同数量的代表性照明,这可导致不同哈希单元122、142内的像素在帧之间具有略微不同的阴影。这种频繁的颜色调整通常被称为闪烁,其效果至少部分取决于入射到给定像素位置的不同光线之间的照明值的差异。

[0023] 为了避免此类问题,本文所示的至少一些实施例提供了在算法的单独执行中独立于顺序重做相同计算步骤的能力,这可以利用空间哈希。图2A 示出了对象的部分的图像

200,这里是车辆的底侧。空间哈希可以应用于这样的图像以生成如图2B所示的该图像的表示220,其中相邻单元的组被分配给不同的哈希单元。在一种这样的方法中,空间哈希映射条目E可以覆盖要生成的图像的n个像素,例如3x3像素242的阵列240或网格,如图2C所示。对方向进行采样的有效方法包括存储一个计数器值c,该计数器值表示序列中的当前索引,例如准蒙特卡罗(QMC)序列,用于对空间哈希映射条目E覆盖的其表面上方的方向进行采样。在光线追踪阶段,为条目E启动n个线程,例如该条目中的每个像素一个线程。每个线程可以保留采样序列的一段,例如,通过原子方式增加计数器c的值。然后,每个线程可以从通过其相应像素可见的点追踪采样光线,并将其贡献添加到哈希映射条目E。使用这样的原子函数可以确保操作的原子性,但不保证该操作的执行顺序。如果每个线程追踪10条光线,那么在一次执行中,线程0可能是第一个计划发出原子加法调用的线程。然后,线程0将c的值从0更改为10,并追踪序列中的前10个样本。线程1可以追踪样本10到19,依此类推。在另一次执行中,可以首先调度线程1,并追踪光线0到9,而线程0将追踪光线10到19。由于光线来自条目E所覆盖区域内稍有不同的位置,因此结果会因执行的不同而不同,从而导致动画中出现闪烁的伪影。

[0024] 根据各种实施例的方法可以尝试通过例如有效地为每个哈希单元选择单个代表点来解决现有方法中的这些和其他缺陷。这样,就可以使用并行采样过程来提供一致、可重复的结果。当哈希映射不能跨帧重用,例如在动画过程中,提供这样的结果可能特别重要。

[0025] 在至少一个实施例中,在哈希映射的条目分配期间,可以从空间哈希映射条目E跨越的区域内选择单个点。然后,可以从该单个点的位置追踪所有计算,例如可能涉及基于单个点的空间位置的光线追踪,而不管线程调度如何。这可以在至少一些实施例中通过在单个哈希条目中存储两个附加值来实现,包括像素索引和选择器值。选择单个索引的一种简单方法是从设置为特定值的像素索引开始,并使用简单的原子比较和交换来存储接触该条目的第一线程的索引。然而,由于调度,这可能会导致相同的不确定性问题。例如,也可以将点保持在最靠近条目跨距的体积中心的位置。反过来,这可能会导致混叠伪影,尤其是在以混乱或高频内容为特征的场景中。

[0026] 根据至少一个实施例的方法可以替代地利用伪随机或确定性函数。在这种方法中,可以为条目所覆盖的每个点计算不同的值(例如选择器值),并且可以选择具有最高(或最小)选择器值的像素的索引,例如通过使用原子最大/最小函数。例如,可以使用空间哈希方案计算该值,而无需离散化。在另一个示例中,可以使用提供确定性结果的伪随机生成器。条目的选择器值可以存储迄今为止遇到的最大(或最小)选择器值,并且像素索引可以用于存储已获得该选择器的像素的索引。

[0027] 在某些图形硬件上,以原子方式更新选择器和像素索引值可能具有挑战性。为了应对这些挑战,两个值都可以捆绑成一个值。这可以是32位值,例如,其中较低的25位用于存储像素索引,较高的7位用于表示选择器值。作为示例,25位允许2D图像的像素索引高达3300万像素,除其他此类选项外,可支持高达8K的分辨率。将选择器值存储在较高权重位中也有助于确保原子最小/最大操作将主要考虑选择器值,并且仅在选择器值相等的情况下采用像素索引。一些实施例可以替代地利用64位值来嵌入选择器和像素标识符两者,潜在代价是额外的存储器和处理时间。

[0028] 在光线追踪阶段,单个线程可以找到其条目,并获得对应于该条目的像素索引。该线程还可以获取该像素位置的曲面位置和法线。然后,该线程可以原子地保留伪随机序列的一段,并追踪其采样光线。由于至少在本实施例中,给定条目的所有线程将追踪来自同一位置的光线,因此调度顺序变得不相关,并且生成的图像应保持一致。因此,虽然先前的方法侧重于在静态场景中的多个帧上逐步细化估计,但它们不处理动态场景的特定情况,对于动态场景,必须在每个帧上重置哈希映射(或其部分)。本文提出的方法至少可以解决这个问题,例如,关于使用空间哈希的动画场景的实时渲染。

[0029] 使用空间哈希的一个优点是,数据可以存储在三维空间中,而不是存储在2D图像中。可以在常规哈希映射中生成多个单元或哈希条目,其中每个区域覆盖空间中的一个区域。这种方法对于确定场景的照明等目的非常实用,因为场景中的所有表面上的照明往往相对平滑。可以收集、组合和重用给定哈希单元中像素的信息,以提供对这些像素的照明的非常平滑的估计,否则每个像素的信息可能很少,这可能会导致图像具有大量噪声或其他此类伪影。在图3的情况300中,示出了包括多个点的表面的部分 310。这些点的位置可以映射到三维(3D)空间,该空间可概括为 n 维。哈希函数可以获取这些点的坐标,并生成长度为32位的一维(1D)值。该一维值提供了一个位置来存储关于该点的信息,例如到达该点的光量。其他点也可以这样做,部分地受此特定哈希映射中的条目数目限制。在这样的映射中,可能不相关的两个点可以共享相同的哈希键。这种方法还可以用于检测或避免冲突,因为可以利用在相同输入数据上计算的另一个哈希键。可以分析该第二哈希键以确定是否存在冲突,如果存在冲突,则可以标识哈希映射中的另一个位置以用于存储相应的数据。

[0030] 可以使用许多不同的方法来对三维中的点进行哈希运算。在至少一个实施例中,可以对点的第一坐标进行哈希运算,然后添加第二坐标,然后执行另一哈希运算,然后执行第三坐标和另一哈希运算,从而产生三维哈希值。在至少一个实施例中,位置信息还可以离散化,使得同一单元中的所有点最终位于同一哈希映射条目中。如上所述,现有方法的一个问题是,可以在给定哈希单元内的不同点(例如, p_1 、 p_2 、 p_3)估计照明,这可能导致这些点被独立处理。在至少一个实施例中,每个点可以有一个执行线程。可能有许多样本被确定为在一系列样本中被采集,这对于光线追踪可能涉及一系列伪随机光线。这可以包括对第一个点追踪光线1、2,对第二个点追踪光线3、4,对第三个点追踪光线5、6。然而,在某些硬件中,例如在 GPU 中,可能无法保证第一个点将追踪光线1、2,或者第二个点将追踪不同采样的光线3、4。这些追踪可能取决于所需的比例和硬件方面等因素,因此在两次不同的执行之间,应用程序可能不会最终追踪单个点的相同光线。如前所述,当呈现图像或视频帧序列时,追踪的光线的差异可能导致伪影,例如闪烁。因此,部分取决于要追踪的光线的调度,对于不同的帧可能无法确定相同的结果。这些单元也可以独立闪烁,这可能会对至少一些演示文稿或观众造成视觉干扰。

[0031] 这些点中的每一个都可以对应于给定哈希条目内的不同像素,该哈希条目可以覆盖多个像素。然后,给定哈希单元中的每个点都可以贡献于为该哈希单元计算的照明估计。在一些应用中,每个像素可能有一个这样的点,而在其他应用中,每两个或四个像素可能有一个这样的点,以及其他这样的选项。在至少一个实施例中,可以配置要包括在给定哈希单元中的像素数量,用户或应用程序可以设置这些像素数量,例如选择一个 3×3 或 9×9 像素阵列。哈希单元中的每个像素也可以对应于一个独立的线程。每个像素可以在伪随机样本序

列中拾取多个样本。计数器可以与给定的哈希单元相关联,哈希单元可以从默认值(如0)开始。要为给定点执行的第一个线程可能需要追踪十条光线,这将导致此计数器递增0到10,并追踪光线0-9。要执行的下一个线程将以原子方式将计数器递增到10,并追踪样本10-19,对于第三个线程也是如此。这些线程中的每一个都可以引用存储器中的单个值,可以使用原子函数对其进行更新。这些原子值允许以其他线程无法干预的方式对特定值进行查找,以及对这些值进行添加。例如,当GPU上有多个线程并行运行时,可能无法保证这些原子加法的执行顺序。可能没有任何方案可以保证每次执行这些原子相加添加时线程执行都以相同的方式进行调度。

[0032] 根据各种实施例的方法可以尝试确保为给定线程的每次执行追踪相同的光线。一种方法可以选择一个点,从该点追踪该单元的所有光线,而不是从给定哈希单元内的所有不同点进行采样。即使不同像素对应于给定哈希单元的不同点,也可以执行此操作。因此,一种方法可以从单元内的各个点中选择一个感兴趣的点,然后仅从该选定点追踪光线。每个线程仍然可以追踪确定数量的光线,但这些光线将全部来自空间中的同一点。这种方法在调度中可能仍然具有至少一定量的随机性,但这种随机性的影响被最小化,因为哈希单元的所有光线都来自同一个地方,这样就可以为给定哈希单元的每次执行追踪相同的光线。

[0033] 可以使用各种方法来选择给定哈希单元内的代表点或兴趣点。如果只有一个单元,则可以做出始终选择第一点的选择,例如如图4A所示的单元400中的左上角点,并且可以从该位置执行所有追踪。然而,这种方法的一个潜在问题是,在图像上对多个哈希单元使用相同的角点可能会导致一些混叠问题,因为可能存在某些从未采样的特征,这可能会导致分辨率或图像质量的损失。例如,为了避免丢失小的或精细的特征,在至少一些实施例中,期望在选择代表点时提供至少一些量的随机化。

[0034] 在至少一个实施例中,可以采取类似于空间哈希运算的点随机化方法。可以为每个兴趣点计算哈希值。如图3A所示,这可以由 $H_i(p_i)$ 表示。可能存在与每个点相关联的伪随机索引,并且可以任意地或根据选择标准选择值,例如选择具有最大哈希值的点。如图4B的示例情况450所示,这可以对应于被选择为具有最大哈希值的每个单元中的不同点位置452。然后可以从这些位置追踪每个相应单元的光线。这种方法可以提供变化,有助于创建更统一的模式,从而更有可能捕获精细特征或细节的信息。在至少一个实施例中,可以使用伪随机生成器,或将随时间保持一致且具有确定性的其他技术,代替用于将值指派给各个点的哈希函数。在至少一个实施例中,需要为每个像素缓存(或以其他方式存储)至少两条信息,包括哈希值(或伪随机数等)和像素索引。这些值可用于定位或识别兴趣点。然而,一个潜在的问题是,原子函数通常只作用于较小的值,对于DirectX 或Vulkan等应用程序,这些值只能是32位的值。如果确定 p_i 是单元中的最大值点,例如,可以使用诸如AtomicMax的函数来原子地存储,如图4B所示。此32位AtomicMax值可用于存储两个单独的值,包括各自的哈希值或伪随机值(此处为MaxH或导致该点被选为兴趣点的最大哈希值)和像素索引(此处为Indx)。在至少一个实施例中,这可以包括使用7位作为MaxH值,使用25位作为各个点的索引值。这种方法可以将这些值捆绑在一起,并使用单个AtomicMax函数调用进行更新。可以假设将保留此具有最大哈希值的32位值,然后AtomicMax函数可以对该值的最强位进行操作。如果两个点具有相同的哈希,则该过程也可以取两个像素索引中的最大值,这使

得在至少一些实施例中能够具有单个原子调用。应该理解,也可以使用其他位大小,例如64位原子值,它可能使用32位数字来存储 MaxH和Indx值。然而,在至少一些实施例中,可以优选较小的比特大小以减少空间、时间和处理需求。可以使用哈希算法,因为这样的算法可以提供相对随机和独立的值,或者至少对于每个点值根本上不相同,并且可以在图像空间中生成良好的模式。在至少一个实施例中,可将用于哈希单元的相同空间哈希方案应用于哈希单元的图像点的浮点坐标。在至少一个实施例中,MaxH值是选择器值,使得其完全独立于照明。照明和其他信息将存储在哈希映射中的其他位置,该值允许确定采样点。

[0035] 在至少一个实施例中,哈希单元可以具有多达128个像素。任何更多像素都可能导致该方法恢复为也使用像素索引来管理冲突值。然后可以使用像素索引来确定要选择的像素或点,因为选择过程可以选择最大的像素索引。如上所述,哈希元素本身的大小是可配置的,尽管在某些实现方式中,10x10像素以上的大小可能太大,最终可能会遗漏太多细微的细节。

[0036] 这种方法还可以利用一种分量来分离这两个混合值。空间哈希方案可以首先计算图像的所有单元,然后一遍执行一个操作,例如追踪来自这些单元的各个识别点的光线。在追踪任何光线之前,系统可以确定给定单元中的哪个像素是最大像素,或者是具有最高哈希值或伪随机值的像素。然后,系统可以尝试确定通过该像素可以看到3D空间中的哪个点,然后可以追踪来自该像素的光线。在许多实现方式中,查找该值并提取25个较弱的位以获得感兴趣像素的索引将相对简单。

[0037] 图5示出了用于对待渲染图像执行光线追踪的示例过程500,该过程可以根据各种实施例执行。应当理解,对于本文讨论的此过程和其他过程,除非另有明确说明,否则在各种实施例的范围内,可以以类似或替代顺序或至少部分并行地执行附加、更少或替代步骤。此外,尽管该过程是关于渲染图像来描述的,但应当理解,利用空间哈希运算进行一致采样的优点还可以有利地用于其他应用、内容类型或用途。在该示例中,为图像中的待渲染场景的视图确定502数据。例如,这可以包括要在图像中至少部分地表示的一个或多个对象的数据。可以使用空间哈希算法来生成图像的哈希映射,并且可以将该图像中的像素子集分配504给各个哈希映射单元。在至少一个实施例中,每个单元可以包括多达指定数量的像素或点,例如 3x3网格中的九个点。哈希单元中的每个像素可以对应于3D空间中的对应点,并且可以为这些点中的每个点生成哈希值或伪随机数506(在本文讨论和建议的其他此类选项中)。然后,可以使用508原子最大值函数为这些哈希映射单元中的每一个选择代表点或兴趣点。这种方法允许在一个单元内相对随机地选择点,但随后在独立于线程调度的多个执行中一致地选择。然后,可以将最大哈希值(或其他选择器值)与像素索引值一起存储510在组合值中,例如单个32位值。可以将组合值提供512给光线追踪过程(或硬件等),以确定像素位置来用于每次执行的各个哈希映射单元的光线追踪。然后,可以从代表点和应用于哈希映射单元的各个像素的照明来执行514光线追踪,以用于渲染图像以进行呈现或其他此类目的。

[0038] 如前所述,本文提出的各种方法足够轻量级,可以在客户端设备上实时执行,例如个人计算机或游戏控制台。可以对在该客户端设备上生成的或从外部源接收的内容执行这种处理,例如通过至少一个网络接收的流式传输内容。源可以是任何适当的源,例如游戏主机、流媒体提供商、第三方内容提供商或其他客户端设备,以及其他此类选项。在一些情况

下,该内容的处理和/或渲染可以由这些其他设备、系统或实体之一执行,然后提供给客户端设备(或另一个这样的接收者)用于呈现或另一个这样的使用。

[0039] 作为示例,图6示出了可用于提供、生成、修改、编码和/或传输内容的示例网络配置600。在至少一个实施例中,客户端设备602可以使用客户端设备602上的内容应用604的组件和本地存储在该客户端设备上的数据来生成或接收会话的内容。在至少一个实施例中,在内容服务器620(例如,云服务器或边缘服务器)上执行的内容应用程序624(例如,图像生成或编辑应用程序)可以启动与至少客户端设备602相关联的会话,这可以利用会话管理器626和存储在用户数据库634中的用户数据,并且可以使内容管理器626确定内容632。内容生成或管理应用程序626可以确定要生成的图像的数据,该数据可以传递给光线追踪组件630和渲染引擎628,以及其他此类组件、模块或处理,以生成要发送给客户端设备602或其他此类接收者的图像数据、视频帧或其他此类内容。可以使用适当的传输管理器622将该内容的至少一部分传输到客户端设备602,以通过下载、流式传输或另一这样的传输信道进行发送。编码器可用于在发送到客户端设备602之前对该数据中的至少一些进行编码和/或压缩。在至少一个实施例中,接收此类内容的客户端设备602可以将此内容提供给相应的内容应用程序604,该内容应用程序604还可以或可选地包括内容管理器610、渲染引擎612或光线追踪硬件或软件614。解码器还可用于解码通过网络640接收的用于经由客户端设备602呈现的数据,例如通过显示器606的图像或视频内容,以及通过至少一个音频播放设备608(例如扬声器或耳机)的音频,例如声音和音乐。在至少一个实施例中,该内容中的至少一部分可能已经存储在客户端设备602上、渲染在客户端设备602上或可供客户端设备602访问,使得至少该部分内容不需要通过网络640进行传输,例如该内容之前可能已经下载或本地存储在硬盘或光盘上。在至少一个实施例中,诸如数据流的传输机制可用于将该内容从服务器620或用户数据库634传输到客户端设备602。在至少一个实施例中,该内容的至少一部分可以从另一个源获得或流式传输,例如第三方服务660,其还可以包括用于生成或提供内容的内容应用程序662。在至少一个实施例中,可以使用多个计算设备或一个或更多个计算设备内的多个处理器来执行该功能的部分,例如可以包括CPU和GPU的组合。

[0040] 在该示例中,这些客户端设备可以包括任何适当的计算设备,如可以包括台式计算机、笔记本计算机、机顶盒、流式传输设备、游戏控制台、智能手机、平板计算机、VR耳机、AR护目镜、可穿戴计算机或智能电视。每个客户端设备可以跨至少一个有线或无线网络提交请求,所述网络可以包括因特网、以太网、局域网(LAN)或蜂窝网络,以及其他此类选项。在该示例中,这些请求可以提交到与云提供商相关联的地址,所述云提供商可以在云提供商环境下操作或控制一个或更多个电子资源,所述云提供商环境诸如可以包括数据中心或服务器群。在至少一个实施例中,该请求可以由位于网络边缘并且在与云提供商环境相关联的至少一个安全层之外的至少一个边缘服务器接收或处理。通过这种方式,可以通过使得客户端设备能够与距离更近的服务器交互来减少延迟,同时还可以提高云提供商环境中资源的安全性。

[0041] 在至少一个实施例中,此类系统可以用于执行图形渲染操作。在其他实施例中,此类系统可以用于其他目的,诸如用于提供图像或视频内容以测试或验证自主机器应用,或者用于执行深度学习操作。在至少一个实施例中,此类系统可以使用边缘设备来实现,或者可以结合一个或更多个虚拟机(VM)。在至少一个实施例中,此类系统可以至少部分地在数

据中心中或至少部分地使用云计算资源来实现。

[0042] 推理和训练逻辑

[0043] 图7A示出了用于执行与一个或更多个实施例相关联的推理和/或训练操作的推理和/或训练逻辑715。下面结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。

[0044] 在至少一个实施例中,推理和/或训练逻辑715可以包括但不限于代码和/或数据存储701,用于存储前向和/或输出权重和/或输入/输出数据,和/或在一个或更多个实施例的方面中配置被训练为和/或用于推理的神经网络的神经元或层的其他参数。在至少一个实施例中,训练逻辑715可以包括或耦合到用于存储图形代码或其他软件以控制时序和/或顺序的代码和/或数据存储701,其中权重和/或其他参数信息被加载以配置逻辑,包括整数和/或浮点单元(统称为算术逻辑单元(ALU))。在至少一个实施例中,代码(诸如图代码)基于该代码所对应的神经网络的架构将权重或其他参数信息加载到处理器ALU中。在至少一个实施例中,代码和/或数据存储701存储在使用一个或更多个实施例的方面训练和/或推理期间的输入/输出数据和/或权重参数的前向传播期间结合一个或更多个实施例训练或使用的神经网络的每个层的权重参数和/或输入/输出数据。在至少一个实施例中,代码和/或数据存储701的任何部分都可以包括在其他片上或片外数据存储内,包括处理器的L1、L2或L3高速缓存或系统存储器。

[0045] 在至少一个实施例中,代码和/或数据存储701的任何部分可以在一个或更多个处理器或其他硬件逻辑设备或电路的内部或外部。在至少一个实施例中,代码和/或数据存储701可以是高速缓存存储器、动态随机可访问存储器(“DRAM”)、静态随机可访问存储器(“SRAM”)、非易失性存储器(例如闪存)或其他存储。在至少一个实施例中,对代码和/或数据存储701是在处理器的内部还是外部的选择,例如,或者由DRAM、SRAM、闪存或某种其他存储类型组成,可以取决于片上或片外的可用存储空间,正在执行训练和/或推理功能的延迟要求,在神经网络的推理和/或训练中使用的数据的批大小或这些因素的某种组合。

[0046] 在至少一个实施例中,推理和/或训练逻辑715可以包括但不限于代码和/或数据存储705,用于存储与在一个或更多个实施例的方面中被训练为和/或用于推理的神经网络的神经元或层相对应的反向和/或输出权重和/或输入/输出数据。在至少一个实施例中,在使用一个或更多个实施例的方面训练和/或推理期间,代码和/或数据存储705存储在输入/输出数据和/或权重参数的反向传播期间结合一个或更多个实施例训练或使用的神经网络的每个层的权重参数和/或输入/输出数据。在至少一个实施例中,训练逻辑715可以包括或耦合到用于存储图代码或其他软件以控制时序和/或顺序的代码和/或数据存储705,其中权重和/或其他参数信息被加载以配置逻辑,该逻辑包括整数和/或浮点单元(统称为算术逻辑单元(ALU))。在至少一个实施例中,代码(诸如图代码)基于该代码所对应的神经网络的架构将权重或其他参数信息加载到处理器ALU中。在至少一个实施例中,代码和/或数据存储705的任何部分可以与其他片上或片外数据存储一起包括,包括处理器的L1、L2或L3高速缓存或系统存储器。在至少一个实施例中,代码和/或数据存储705的任何部分可以在一个或更多个处理器或其他硬件逻辑设备或电路上的内部或外部。在至少一个实施例中,代码和/或数据存储705可以是高速缓存存储器、DRAM、SRAM、非易失性存储器(例如闪存)或其他存储。在至少一个实施例中,代码和/或数据存储705是在处理器的内部还是外部的选择,例如,是由DRAM、SRAM、闪存还是其他某种存储类型组成,取决于可用存储是片上还是片

外,正在执行的训练和/或推理功能的延迟要求,在神经网络的推理和/或训练中使用的数据批量大小或这些因素的某种组合。

[0047] 在至少一个实施例中,代码和/或数据存储701以及代码和/或数据存储705可以是分开的存储结构。在至少一个实施例中,代码和/或数据存储 701以及代码和/或数据存储705可以是相同的存储结构。在至少一个实施例中,代码和/或数据存储701以及代码和/或数据存储705可以是部分相同的存储结构和部分分离的存储结构。在至少一个实施例中,代码和/或数据存储701以及代码和/或数据存储705的任何部分可以与其他片上或片外数据存储包括在一起,包括处理器的L1、L2或L3高速缓存或系统存储器。

[0048] 在至少一个实施例中,推理和/或训练逻辑715可以包括但不限于一个或更多个算术逻辑单元(“ALU”)710(包括整数和/或浮点单元),用于至少部分地基于训练和/或推理代码(例如,图代码)或由其指示来执行逻辑和/或数学运算,其结果可能会产生存储在激活存储720中的激活(例如,来自神经网络内部的层或神经元的输出值),其是存储在代码和/或数据存储701和/或代码和/或数据存储705中的输入/输出和/或权重参数数据的函数。在至少一个实施例中,激活响应于执行指令或其他代码,由 ALU710执行的线性代数和/或基于矩阵的数学生成在激活存储720中存储的激活,其中存储在代码和/或数据存储705中和/或代码和/或数据存储701 中的权重值用作具有其他值的操作数,例如偏置值、梯度信息、动量值或其他参数或超参数,可以将任何或所有这些存储在代码和/或数据存储705 或代码和/或数据存储701或其他片上或片外存储中。

[0049] 在至少一个实施例中,一个或更多个处理器或其他硬件逻辑设备或电路包括一个或更多个ALU 710,而在另一实施例中,一个或更多个ALU 710可以在处理器或其他硬件逻辑设备或使用它们(例如协处理器)的电路外。在至少一个实施例中,可以将一个或更多个ALU 710包括在处理器的执行单元之内,或者以其他方式包括在由处理器的执行单元可访问的 ALU组中,该处理器的执行单元可以在同一处理器内或者分布在不同类型的不同处理器之间(例如,中央处理单元、图形处理单元、固定功能单元等)。在至少一个实施例中,代码和/或数据存储701、代码和/或数据存储 705以及激活存储720可以在同一处理器或其他硬件逻辑设备或电路上,而在另一实施例中,它们可以在不同的处理器或其他硬件逻辑设备或电路或相同和不同处理器或其他硬件逻辑设备或电路的某种组合中。在至少一个实施例中,激活存储720的任何部分可以与其他片上或片外数据存储包括在一起,包括处理器的L1、L2或L3高速缓存或系统存储器。此外,推理和/或训练代码可以与处理器或其他硬件逻辑或电路可访问的其他代码一起存储,并可以使用处理器的提取、解码、调度、执行、退出和/或其他逻辑电路来提取和/或处理。

[0050] 在至少一个实施例中,激活存储720可以是高速缓存存储器、DRAM、SRAM、非易失性存储器(例如,闪存)或其他存储。在至少一个实施例中,激活存储720可以完全地或部分地在一个或更多个处理器或其他逻辑电路内部或外部。在至少一个实施例中,可以取决于片上或片外可用的存储,进行训练和/或推理功能的延迟要求,在推理和/或训练神经网络中使用的数据的批量大小或这些因素的某种组合,选择激活存储720是处理器的内部还是外部,例如,或者包含DRAM、SRAM、闪存或其他存储类型。在至少一个实施例中,图7a中所示的推理和/或训练逻辑715可以与专用集成电路(“ASIC”)结合使用,例如来自Google的Tensorflow®处理单元、来自Graphcore™的推理处理单元(IPU)或来自IntelCorp的

Nervana® (例如“LakeCrest”)处理器。在至少一个实施例中,图7a所示的推理和/或训练逻辑715可与中央处理单元(“CPU”)硬件,图形处理单元(“GPU”)硬件或其他硬件(例如现场可编程门阵列(“FPGA”))结合使用。

[0051] 图7b示出了根据至少一个或更多个实施例的推理和/或训练逻辑715。在至少一个实施例中,推理和/或训练逻辑715可以包括但不限于硬件逻辑,其中计算资源被专用或以其他方式唯一地连同对应于神经网络内的一层或更多层神经元的权重值或其他信息一起使用。在至少一个实施例中,图7b中所示的推理和/或训练逻辑715可以与专用集成电路(ASIC)结合使用,例如来自Google的**TensorFlow®**处理单元,来自Graphcore™的推理处理单元(IPU)或来自IntelCorp的**Nervana®** (例如“LakeCrest”)处理器。在至少一个实施例中,图7b中所示的推理和/或训练逻辑715可以与中央处理单元(CPU)硬件、图形处理单元(GPU)硬件或其他硬件(例如现场可编程门阵列(FPGA))结合使用。在至少一个实施例中,推理和/或训练逻辑715包括但不限于代码和/或数据存储701以及代码和/或数据存储705,其可以用于存储代码(例如,图代码)、权重值和/或其他信息,包括偏置值、梯度信息、动量值和/或其他参数或超参数信息。在图7b中所示的至少一个实施例中,代码和/或数据存储701以及代码和/或数据存储705中的每一个都分别与专用计算资源(例如计算硬件702和计算硬件706)相关联。在至少一个实施例中,计算硬件702和计算硬件706中的每一个包括一个或更多个ALU,这些ALU仅分别对存储在代码和/或数据存储701和代码和/或数据存储705中的信息执行数学函数(例如线性代数函数),执行函数的结果被存储在激活存储720中。

[0052] 在至少一个实施例中,代码和/或数据存储701和705以及相应的计算硬件702和706中的每一个分别对应于神经网络的不同层,使得从代码和/或数据存储701和计算硬件702的一个“存储/计算对701/702”得到的激活提供作为代码和/或数据存储705和计算硬件706的下一个“存储/计算对705/706”的输入,以便反映神经网络的概念组织。在至少一个实施例中,每个存储/计算对701/702和705/706可以对应于一个以上的神经网络层。在至少一个实施例中,在推理和/或训练逻辑715中可以包括在存储计算对701/702和705/706之后或与之并行的附加存储/计算对(未示出)。

[0053] 数据中心

[0054] 图8示出了可以使用至少一个实施例的示例数据中心800。在至少一个实施例中,数据中心800包括数据中心基础设施层810、框架层820、软件层830和应用层840。

[0055] 在至少一个实施例中,如图8所示,数据中心基础设施层810可以包括资源协调器812、分组的计算资源814和节点计算资源(“节点C.R.”) 816(1)-816(N),其中“N”代表任何正整数。在至少一个实施例中,节点C.R.816(1)-816(N)可以包括但不限于任何数量的中央处理单元(“CPU”)或其他处理器(包括加速器、现场可编程门阵列(FPGA)、图形处理器等),存储器设备(例如动态只读存储器)、存储设备(例如固态硬盘或磁盘驱动器),网络输入/输出(“NWI/O”)设备,网络交换机,虚拟机(“VM”),电源模块和冷却模块等。在至少一个实施例中,节点C.R.816(1)-816(N)中的一个或更多个节点C.R.可以是一个或更多个上述计算资源的服务器。

[0056] 在至少一个实施例中,分组的计算资源814可以包括容纳在一个或更多个机架内的节点C.R.的单独分组(未示出),或者容纳在各个地理位置的数据中心内的许多机架(也

未示出)。分组的计算资源814内的节点C.R. 的单独分组可以包括可以被配置或分配为支持一个或更多个工作负载的分组的计算、网络、存储器或存储资源。在至少一个实施例中, 可以将包括 CPU或处理器的几个节点C.R. 分组在一个或更多个机架内, 以提供计算资源来支持一个或更多个工作负载。在至少一个实施例中, 一个或更多个机架还可以包括任何数量的电源模块、冷却模块和网络交换机, 以任意组合。

[0057] 在至少一个实施例中, 资源协调器812可以配置或以其他方式控制一个或更多个节点C.R. 816 (1) -816 (N) 和/或分组的计算资源814。在至少一个实施例中, 资源协调器812可以包括用于数据中心800的软件设计基础结构 (“SDI”) 管理实体。在至少一个实施例中, 资源协调器108 可以包括硬件、软件或其某种组合。

[0058] 在至少一个实施例中, 如图8所示, 框架层820包括作业调度器822、配置管理器824、资源管理器826和分布式文件系统828。在至少一个实施例中, 框架层820可以包括支持软件层830的软件832和/或应用程序层840 的一个或更多个应用程序842的框架。在至少一个实施例中, 软件832或应用程序842可以分别包括基于Web的服务软件或应用程序, 例如由AmazonWebServices, GoogleCloud和MicrosoftAzure提供的服务或应用程序。在至少一个实施例中, 框架层820可以是但不限于一种免费和开放源软件网络应用程序框架, 例如可以利用分布式文件系统828来进行大范围数据处理 (例如“大数据”) 的ApacheSpark™ (以下称为“Spark”)。在至少一个实施例中, 作业调度器832可以包括Spark驱动器, 以促进对数据中心800的各个层所支持的工作负载进行调度。在至少一个实施例中, 配置管理器824可以能够配置不同的层, 例如软件层830和包括Spark和用于支持大规模数据处理的分布式文件系统828的框架层820。在至少一个实施例中, 资源管理器826能够管理映射到或分配用于支持分布式文件系统828和作业调度器822的集群或分组计算资源。在至少一个实施例中, 集群或分组计算资源可以包括数据中心基础设施层810上的分组计算资源 814。在至少一个实施例中, 资源管理器826可以与资源协调器812协调以管理这些映射的或分配的计算资源。

[0059] 在至少一个实施例中, 包括在软件层830中的软件832可以包括由节点C.R. 816 (1) -816 (N) 的至少一部分, 分组的计算资源814和/或框架层820的分布式文件系统828使用的软件。一种或更多种类型的软件可以包括但不限于Internet网页搜索软件、电子邮件病毒扫描软件、数据库软件和流视频内容软件。

[0060] 在至少一个实施例中, 应用层840中包括的一个或更多个应用程序842 可以包括由节点C.R. 816 (1) -816 (N) 的至少一部分、分组计算资源814 和/或框架层820的分布式文件系统828使用的一种或更多种类型的程序。一种或更多种类型的程序可以包括但不限于任何数量的基因组学应用程序、认知计算和机器学习应用程序, 包括训练或推理软件, 机器学习框架软件 (例如PyTorch、TensorFlow、Caffe等) 或其他与一个或更多个实施例结合使用的机器学习应用程序。

[0061] 在至少一个实施例中, 配置管理器824、资源管理器826和资源协调器812中的任何一个可以基于以任何技术上可行的方式获取的任何数量和类型的数据来实现任何数量和类型的自我修改动作。在至少一个实施例中, 自我修改动作可以减轻数据中心800的数据中心操作员做出可能不好的配置决定并且可以避免数据中心的未充分利用和/或执行差的部分。

[0062] 在至少一个实施例中, 数据中心800可以包括工具、服务、软件或其他资源, 以根据

本文所述的一个或更多个实施例来训练一个或更多个机器学习模型或者使用一个或更多个机器学习模型来预测或推理信息。例如,在至少一个实施例中,可以通过使用上文关于数据中心800描述的软件和计算资源,根据神经网络架构通过计算权重参数来训练机器学习模型。在至少一个实施例中,通过使用通过本文所述的一种或更多种训练技术计算出的权重参数,可以使用上面与关于数据中心800所描述的资源,使用对应于一个或更多个神经网络的经训练的机器学习模型来推理或预测信息。

[0063] 在至少一个实施例中,数据中心可以使用CPU、专用集成电路(ASIC)、GPU、FPGA或其他硬件来使用上述资源来执行训练和/或推理。此外,上述的一个或更多个软件和/或硬件资源可以配置成一种服务,以允许用户训练或执行信息推理,例如图像识别、语音识别或其他人工智能服务。

[0064] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑 715的细节。在至少一个实施例中,推理和/或训练逻辑715可以在系统图 8的系统中使用,用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构或本文所述的神经网络用例计算的权重参数来推理或预测操作。

[0065] 此类组件可以用于使用基于光线追踪的重要性采样渲染图像,可以通过硬件进行加速。

[0066] 计算机系统

[0067] 图9是示出根据至少一个实施例示例性计算机系统的框图,该示例性计算机系统可以是具有互连的设备和组件的系统,片上系统(SOC)或它们的某种形成有处理器的组合,该处理器可以包括执行单元以执行指令。在至少一个实施例中,根据本公开,例如本文所述的实施例,计算机系统 900可以包括但不限于组件,例如处理器902,其执行单元包括逻辑以执行用于过程数据的算法。在至少一个实施例中,计算机系统900可以包括处理器,例如可从加利福尼亚圣塔克拉拉的英特尔公司(IntelCorporationofS antaClara, California)获得的PENTIUM®处理器家族、Xeon™、Itanium ®、XScale™和/或StrongARMTM, Intel®Core™或Intel®Nervana™微处理器,尽管也可以使用其他系统(包括具有其他微处理器的PC、工程工作站、机顶盒等)。在至少一个实施例中,计算机系统900可以执行可从华盛顿州雷蒙德的微软公司(MicrosoftCorporationofRedmond, Wash.)获得的WINDOWS操作系统版本,尽管其他操作系统(例如UNIX和Linux)、嵌入式软件和/或图形用户界面也可以使用。

[0068] 实施例可以用在其他设备中,例如手持设备和嵌入式应用程序。手持设备的一些示例包括蜂窝电话、互联网协议(InternetProtocol)设备、数码相机、个人数字助理(“PDA”)和手持PC。在至少一个实施例中,嵌入式应用程序可以包括微控制器、数字信号处理器(“DSP”)、片上系统、网络计算机(“NetPC”)、机顶盒、网络集线器、广域网(“WAN”) 交换机,或根据至少一个实施例可以执行一个或更多个指令的任何其他系统。

[0069] 在至少一个实施例中,计算机系统900可包括但不限于处理器902,该处理器902可包括但不限于一个或更多个执行单元908,以根据本文描述的技术执行机器学习模型训练和/或推理。在至少一个实施例中,计算机系统900是单处理器台式机或服务器系统,但是在另一实施例中,计算机系统900可以是多处理器系统。在至少一个实施例中,处理器902可以

包括但不限于复杂指令集计算机 (“CISC”) 微处理器、精简指令集计算 (“RISC”) 微处理器、超长指令字 (“VLIW”) 微处理器、实现指令集组合的处理器, 或任何其他处理设备, 例如数字信号处理器。在至少一个实施例中, 处理器902可以耦合到处理器总线910, 该处理器总线910 可以在处理器902与计算机系统900中的其他组件之间传输数据信号。

[0070] 在至少一个实施例中, 处理器902可以包括但不限于1级 (“L1”) 内部高速缓存存储器 (“高速缓存”) 904。在至少一个实施例中, 处理器 902可以具有单个内部高速缓存或多级内部高速缓存。在至少一个实施例中, 高速缓存存储器可以驻留在处理器902的外部。根据特定的实现和需求, 其他实施例也可以包括内部和外部高速缓存的组合。在至少一个实施例中, 寄存器文件906可以在各种寄存器中存储不同类型的数据, 包括但不限于整数寄存器、浮点寄存器、状态寄存器和指令指针寄存器。

[0071] 在至少一个实施例中, 包括但不限于执行整数和浮点运算的逻辑执行单元908, 其也位于处理器902中。在至少一个实施例中, 处理器902还可以包括微码 (“ucode”) 只读存储器 (“ROM”), 用于存储某些宏指令的微代码。在至少一个实施例中, 执行单元908可以包括用于处理封装指令集909的逻辑。在至少一个实施例中, 通过将封装指令集909包括在通用处理器的指令集中, 以及要执行指令的相关电路, 可以使用处理器 902中的封装数据来执行许多多媒体应用程序使用的操作。在一个或更多个实施例中, 可以通过使用处理器的数据总线的全宽度来在封装的数据上执行操作来加速和更有效地执行许多多媒体应用程序, 这可能不需要在处理器的数据总线上传输较小的数据单元来一次执行一个数据元素的一个或更多个操作。

[0072] 在至少一个实施例中, 执行单元908也可以用在微控制器、嵌入式处理器、图形设备、DSP和其他类型的逻辑电路中。在至少一个实施例中, 计算机系统900可以包括但不限于存储器920。在至少一个实施例中, 存储器920可以实现为动态随机存取存储器 (“DRAM”) 设备、静态随机存取存储器 (“SRAM”) 设备、闪存设备或其他存储设备。在至少一个实施例中, 存储器920可以存储由处理器902可以执行的由数据信号表示的指令919和/或数据921。

[0073] 在至少一个实施例中, 系统逻辑芯片可以耦合到处理器总线910和存储器920。在至少一个实施例中, 系统逻辑芯片可以包括但不限于存储器控制器集线器 (“MCH”) 916, 并且处理器902可以经由处理器总线910 与MCH 916通信。在至少一个实施例中, MCH 916可以提供到存储器920 的高带宽存储器路径918以用于指令和数据存储以及用于图形命令、数据和纹理的存储。在至少一个实施例中, MCH 916可以在处理器902、存储器920和计算机系统900中的其他组件之间启动数据信号, 并且在处理器总线910、存储器920和系统I/O 922之间桥接数据信号。在至少一个实施例中, 系统逻辑芯片可以提供用于耦合到图形控制器的图形端口。在至少一个实施例中, MCH 916可以通过高带宽存储器路径918耦合到存储器920, 并且图形/视频卡912可以通过加速图形端口 (Accelerated Graphics Port) (“AGP”) 互连914耦合到MCH 916。

[0074] 在至少一个实施例中, 计算机系统900可以使用系统I/O 922, 所述系统I/O 922是专有集线器接口总线来将MCH 916耦合到I/O控制器集线器 (“ICH”) 930。在至少一个实施例中, ICH 930可以通过本地I/O总线提供与某些I/O设备的直接连接。在至少一个实施例中, 本地I/O总线可以包括但不限于用于将外围设备连接到存储器920、芯片组和处理器902的高速I/O总线。示例可以包括但不限于音频控制器929、固件集线器 (“FlashBIOS”) 928、

无线收发器926、数据存储924、包含用户输入和键盘接口的传统I/O控制器923、串行扩展端口927 (例如通用串行总线 (USB) 端口) 和网络控制器934。数据存储924可以包括硬盘驱动器、软盘驱动器、CD-ROM设备、闪存设备或其他大容量存储设备。

[0075] 在至少一个实施例中,图9示出了包括互连的硬件设备或“芯片”的系统,而在其他实施例中,图9可以示出示例性片上系统 (SoC)。在至少一个实施例中,设备可以与专有互连、标准化互连 (例如,PCIe) 或其某种组合互连。在至少一个实施例中,计算机系统900的一个或更多个组件使用计算快速链路 (CXL) 互连来互连。

[0076] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关的推理和/或训练操作。下文结合图7A和/或图7B提供关于推理和/或训练逻辑715 的细节。在至少一个实施例中,推理和/或训练逻辑715可以在图9的系统中使用,用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构或本文所述的神经网络用例计算的权重参数来推理或预测操作。

[0077] 此类组件可以用于使用基于光线追踪的重要性采样渲染图像,可以通过硬件进行加速。

[0078] 图10是示出根据至少一个实施例的用于利用处理器1010的电子设备 1000的框图。在至少一个实施例中,电子设备1000可以是,例如但不限于,笔记本电脑、塔式服务器、机架服务器、刀片服务器、膝上型计算机、台式机、平板电脑、移动设备、电话、嵌入式计算机或任何其他合适的电子设备。

[0079] 在至少一个实施例中,系统1000可以包括但不限于通信地耦合到任何合适数量或种类的组件、外围设备、模块或设备的处理器1010。在至少一个实施例中,处理器1010使用总线或接口耦合,诸如I²C总线、系统管理总线 (“SMBus”)、低引脚数 (LPC) 总线、串行外围接口 (“SPI”)、高清音频 (“HDA”) 总线、串行高级技术附件 (“SATA”) 总线、通用串行总线 (“USB”) (1、2、3版) 或通用异步接收器/发送器 (“UART”) 总线。在至少一个实施例中,图10示出了系统,该系统包括互连的硬件设备或“芯片”,而在其他实施例中,图10可以示出示例性片上系统 (SoC)。在至少一个实施例中,图10中所示的设备可以与专有互连线、标准化互连 (例如,PCIe) 或其某种组合互连。在至少一个实施例中,图10的一个或更多个组件使用计算快速链路 (CXL) 互连线来互连。

[0080] 在至少一个实施例中,图10可以包括显示器1024、触摸屏1025、触摸板1030、近场通信单元 (“NFC”) 1045、传感器集线器1040、热传感器1046、快速芯片组 (“EC”) 1035、可信平台模块 (“TPM”) 1038、BIOS/固件/闪存 (“BIOS,FWFlash”) 1022、DSP 1060、驱动器1020 (例如固态硬盘 (“SSD”) 或硬盘驱动器 (“HDD”))、无线局域网单元 (“WLAN”) 1050、蓝牙单元 1052、无线广域网单元 (“WWAN”) 1056、全球定位系统 (GPS) 1055、相机 (“USB3.0相机”) 1054 (例如 USB3.0相机) 和/或以例如LPDDR3标准实现的低功耗双倍数据速率 (“LPDDR”) 存储器单元 (“LPDDR3”) 1015。这些组件可以各自以任何合适的方式实现。

[0081] 在至少一个实施例中,其他组件可以通过上文所述的组件通信地耦合到处理器 1010。在至少一个实施例中,加速度计1041、环境光传感器 (“ALS”) 1042、罗盘1043和陀螺仪1044可以通信地耦合到传感器集线器1040。在至少一个实施例中,热传感器1039、风扇 1037、键盘1036 和触摸板1030可以通信地耦合到EC 1035。在至少一个实施例中,扬声器 1063、耳机1064和麦克风 (“mic”) 1065可以通信地耦合到音频单元 (“音频编解码器和D类放

大器”) 1062,其又可以通信地耦合到DSP 1060。在至少一个实施例中,音频单元1064可以包括例如但不限于音频编码器/解码器(“编解码器”)和D类放大器。在至少一个实施例中,SIM卡(“SIM”) 1057可以通信地耦合到WWAN单元1056。在至少一个实施例中,组件(诸如WLAN单元1050和蓝牙单元1052以及WWAN单元1056)可以被实现为下一代形式因素(NGFF)。

[0082] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。下文结合图7a和/或图7b提供关于推理和/或训练逻辑 715的细节。在至少一个实施例中,推理和/或训练逻辑715可以在图10 系统中用于至少部分地基于使用神经网络训练操作、神经网络功能和/或架构或本文描述的神经网络用例计算的权重参数来推理或预测操作。

[0083] 此类组件可以用于使用基于光线追踪的重要性采样渲染图像,可以通过硬件进行加速。

[0084] 图11是根据至少一个实施例的处理系统的框图。在至少一个实施例中,系统1100包括一个或更多个处理器1102和一个或更多个图形处理器 1108,并且可以是单处理器台式机系统、多处理器工作站系统或具有大量处理器1102或处理器核心1107的服务器系统。在至少一个实施例中,系统1100是结合在片上系统(SoC)集成电路内的处理平台,以在移动、手持或嵌入式设备使用。

[0085] 在至少一个实施例中,系统1100可以包括或结合在基于服务器的游戏平台中,包括游戏和媒体控制台的游戏控制台、移动游戏控制台、手持游戏控制台或在线游戏控制台。在至少一个实施例中,系统1100是移动电话、智能电话、平板计算设备或移动互联网设备。在至少一个实施例中,处理系统1100还可包括与可穿戴设备耦合或集成在可穿戴设备中,例如智能手表可穿戴设备、智能眼镜设备、增强现实设备或虚拟现实设备。在至少一个实施例中,处理系统1100是电视或机顶盒设备,其具有一个或更多个处理器1102以及由一个或更多个图形处理器1108生成的图形界面。

[0086] 在至少一个实施例中,一个或更多个处理器1102每个包括一个或更多个处理器核心1107,以处理指令,该指令在被执行时执行针对系统和用户软件的操作。在至少一个实施例中,一个或更多个处理器核心1107中的每一个被配置为处理特定指令组1109。在至少一个实施例中,指令组1109 可以促进复杂指令集计算(CISC)、精简指令集计算(RISC),或通过超长指令字(VLIW)进行计算。在至少一个实施例中,处理器核心1107可以各自处理不同的指令组1109,该指令组可以包括有助于仿真其他指令组的指令。在至少一个实施例中,处理器核心1107还可以包括其他处理设备,例如数字信号处理器(DSP)。

[0087] 在至少一个实施例中,处理器1102包括高速缓存存储器1104。在至少一个实施例中,处理器1102可以具有单个内部高速缓存或多个级别的内部高速缓存。在至少一个实施例中,高速缓存存储器在处理器1102的各个组件之间共享。在至少一个实施例中,处理器1102还使用外部高速缓存(例如,三级(L3)高速缓存或最后一级高速缓存(LLC))(未示出),可以使用已知的高速缓存一致性技术在处理器核心1107之间共享该外部高速缓存。在至少一个实施例中,处理器1102中另外包括寄存器文件1106,处理器可以包括用于存储不同类型的数据的不同类型的寄存器(例如,整数寄存器、浮点寄存器、状态寄存器和指令指针寄存器)。在至少一个实施例中,寄存器文件1106可以包括通用寄存器或其他寄存器。

[0088] 在至少一个实施例中,一个或更多个处理器1102与一个或更多个接口总线1110耦

合,以在处理器1102与系统1100中的其他组件之间传输通信信号,例如地址、数据或控制信号。在至少一个实施例中,接口总线1110 在一个实施例中可以是处理器总线,例如直接媒体接口 (DMI) 总线的版本。在至少一个实施例中,接口总线1110不限于DMI总线,并且可以包括一个或更多个外围组件互连总线(例如,PCI,PCIExpress)、存储器总线或其他类型的接口总线。在至少一个实施例中,处理器1102包括集成存储器控制器1116和平台控制器集线器1130。在至少一个实施例中,存储器控制器1116促进存储器设备与处理系统1100的其他组件之间的通信,而平台控制器集线器 (PCH) 1130通过本地I/O总线提供到I/O设备的连接。

[0089] 在至少一个实施例中,存储器设备1120可以是动态随机存取存储器 (DRAM) 设备、静态随机存取存储器 (SRAM) 设备、闪存设备、相变存储设备或具有适当的性能以用作处理器存储器。在至少一个实施例中,存储设备1120可以用作处理系统1100的系统存储器,以存储数据1122 和指令1121,以在一个或更多个处理器1102执行应用程序或过程时使用。在至少一个实施例中,存储器控制器1116还与可选的外部图形处理器1112 耦合,其可以与处理器1102中的一个或更多个图形处理器1108通信以执行图形和媒体操作。在至少一个实施例中,显示设备1111可以连接至处理器1102。在至少一个实施例中,显示设备1111可以包括内部显示设备中的一个或更多个,例如在移动电子设备或膝上型设备或通过显示器接口(例如显示端口(DisplayPort)等)连接的外部显示设备中。在至少一个实施例中,显示设备1111可以包括头戴式显示器(HMD),诸如用于虚拟现实(VR)应用或增强现实(AR)应用中的立体显示设备。

[0090] 在至少一个实施例中,平台控制器集线器1130使外围设备能够通过高速I/O总线连接到存储设备1120和处理器1102。在至少一个实施例中,I/O外围设备包括但不限于音频控制器1146、网络控制器1134、固件接口 1128、无线收发器1126、触摸传感器1125、数据存储设备1124(例如,硬盘驱动器、闪存等)。在至少一个实施例中,数据存储设备1124可以经由存储接口(例如,SATA)或经由外围总线来连接,诸如外围组件互连总线(例如,PCI、PCIe)。在至少一个实施例中,触摸传感器1125可以包括触摸屏传感器、压力传感器或指纹传感器。在至少一个实施例中,无线收发器1126可以是Wi-Fi收发器、蓝牙收发器或移动网络收发器,诸如 3G、4G或长期演进(LTE)收发器。在至少一个实施例中,固件接口1128 使能与系统固件的通信,并且可以是例如统一可扩展固件接口(UEFI)。在至少一个实施例中,网络控制器1134可以启用到有线网络的网络连接。在至少一个实施例中,高性能网络控制器(未示出)与接口总线1110耦合。在至少一个实施例中,音频控制器1146是多通道高清晰度音频控制器。在至少一个实施例中,处理系统1100包括可选的传统(legacy) I/O控制器1140,用于将传统(例如,个人系统2(PS/2))设备耦合到系统1100。在至少一个实施例中,平台控制器集线器1130还可以连接到一个或更多个通用串行总线(USB)控制器1142,该控制器连接输入设备,诸如键盘和鼠标1143组合、相机1144或其他USB输入设备。

[0091] 在至少一个实施例中,存储器控制器1116和平台控制器集线器1130 的实例可以集成到离散的外部图形处理器中,例如外部图形处理器1112。在至少一个实施例中,平台控制器集线器1130和/或存储器控制器1116可以在一个或更多个处理器1102的外部。例如,在至少一个实施例中,系统 1100可以包括外部存储器控制器1116和平台控制器集线器1130,其可以配置成在与处理器1102通信的系统芯片组中的存储器控制器集线器和外围控制器集线器。

[0092] 推理和/或训练逻辑715用于执行与一个或多个实施例相关联的推理和/或训练操作。下文结合图7A和/或图7B提供关于推理和/或训练逻辑 715的细节。在至少一个实施例中,部分或全部推理和/或训练逻辑715可以结合到图形处理器1100中。例如,在至少一个实施例中,本文描述的训练和/或推理技术可以使用一个或多个ALU,所述ALU体现在图形处理器中。此外,在至少一个实施例中,本文描述的推理和/或训练操作可以使用除图7A或图7B所示的逻辑之外的逻辑来完成。在至少一个实施例中,权重参数可以存储在片上或片外存储器和/或寄存器(示出或未示出)中,其配置图形处理器的ALU,以执行一种或更多种本文所述的机器学习算法、神经网络架构、用例或训练技术。

[0093] 此类组件可以用于使用基于光线追踪的重要性采样渲染图像,可以通过硬件进行加速。

[0094] 图12是根据至少一个实施例的具有一个或多个处理器核心 1202A-1202N、集成存储器控制器1214和集成图形处理器1208的处理器 1200的框图。在至少一个实施例中,处理器1200可以包含附加核心,多达并包括以虚线框表示的附加核心1202N。在至少一个实施例中,每个处理器核心1202A-1202N包括一个或多个内部高速缓存单元1204A-1204N。在至少一个实施例中,每个处理器核心还可以访问一个或多个共享高速缓存单元1206。

[0095] 在至少一个实施例中,内部高速缓存单元1204A-1204N和共享高速缓存单元1206表示处理器1200内的高速缓存存储器层次结构。在至少一个实施例中,高速缓存存储器单元1204A-1204N可以包括每个处理器核心内的至少一级指令和数据高速缓存以及共享中级高速缓存中的一级或更多级缓存,例如2级(L2)、3级(L3)、4级(L4)或其他级别的高速缓存,其中将外部存储器之前的最高级别的高速缓存归类为LLC。在至少一个实施例中,高速缓存一致性逻辑维持各种高速缓存单元1206和1204A-1204N 之间的一致性。

[0096] 在至少一个实施例中,处理器1200还可包括一组一个或多个总线控制器单元1216和系统代理核心1210。在至少一个实施例中,一个或多个总线控制器单元1216管理一组外围总线,例如一个或多个PCI或 PCIe总线。在至少一个实施例中,系统代理核心1210为各种处理器组件提供管理功能。在至少一个实施例中,系统代理核心1210包括一个或多个集成存储器控制器1214,以管理对各种外部存储器设备(未示出)的访问。

[0097] 在至少一个实施例中,一个或多个处理器核心1202A-1202N包括对多线程同时进行的支持。在至少一个实施例中,系统代理核心1210包括用于在多线程处理期间协调和操作核心1202A-1202N的组件。在至少一个实施例中,系统代理核心1210可以另外包括电源控制单元(PCU),该电源控制单元包括用于调节处理器核心1202A-1202N和图形处理器1208的一个或多个电源状态的逻辑和组件。

[0098] 在至少一个实施例中,处理器1200还包括用于执行图处理操作的图形处理器1208。在至少一个实施例中,图形处理器1208与共享高速缓存单元1206和包括一个或多个集成存储器控制器1214的系统代理核心1210耦合。在至少一个实施例中,系统代理核心1210还包括用于驱动图形处理器输出到一个或多个耦合的显示器的显示器控制器1211。在至少一个实施例中,显示器控制器1211也可以是经由至少一个互连与图形处理器1208耦合的独立模块,或者可以集成在图形处理器1208内。

[0099] 在至少一个实施例中,基于环的互连单元1212用于耦合处理器1200 的内部组件。在至少一个实施例中,可以使用替代性互连单元,例如点对点互连、交换互连或其他技术。

在至少一个实施例中,图形处理器1208 经由I/O链路1213与环形互连1212耦合。

[0100] 在至少一个实施例中,I/O链路1213代表多种I/O互连中的至少一种,包括促进各种处理器组件与高性能嵌入式存储器模块1218 (例如eDRAM 模块) 之间的通信的封装I/O互连。在至少一个实施例中,处理器核心 1202A-1202N和图形处理器1208中的每一个使用嵌入式存储器模块1218 作为共享的最后一级高速缓存。

[0101] 在至少一个实施例中,处理器核心1202A-1202N是执行公共指令集架构的同质核心。在至少一个实施例中,处理器核心1202A-1202N在指令集架构 (ISA) 方面是异构的,其中一个或更多个处理器核心1202A-1202N 执行公共指令集,而一个或更多个其他处理器核心1202A-1202N执行公共指令集的子集或不同指令集。在至少一个实施例中,就微架构而言,处理器核心1202A-1202N是异构的,其中具有相对较高功耗的一个或更多个核心与具有较低功耗的一个或更多个功率核心耦合。在至少一个实施例中,处理器1200可以在一个或更多个芯片上实现或被实现为SoC集成电路。

[0102] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。下文结合图7a和/或图7b提供关于推理和/或训练逻辑 715的细节。在至少一个实施例中,部分或全部推理和/或训练逻辑715可以结合到处理器1200中。例如,在至少一个实施例中,本文描述的训练和 /或推理技术可以使用一个或更多个ALU,所述ALU体现在图12中的图形处理器1512、图形核心1202A-1202N或其他组件中。此外,在至少一个实施例中,本文描述的推理和/或训练操作可以使用除图7A或图7B所示的逻辑之外的逻辑来完成。在至少一个实施例中,权重参数可以存储在片上或片外存储器和/或寄存器(示出或未示出)中,其配置图形处理器1200 的ALU以执行一种或更多种本文所述的机器学习算法、神经网络架构、用例或训练技术。

[0103] 此类组件可以用于使用基于光线追踪的重要性采样渲染图像,可以通过硬件进行加速。

[0104] 虚拟化计算平台

[0105] 图13是根据至少一个实施例的生成和部署图像处理和推理管线的过程1300的示例数据流程图。在至少一个实施例中,过程1300可以被部署用于在一个或更多个设施1302处与成像设备、处理设备和/或其他设备类型一起使用。过程1300可以在训练系统1304和/或部署系统1306内执行。在至少一个实施例中,训练系统1304可以用于执行机器学习模型(例如,神经网络、对象检测算法、计算机视觉算法等)的训练、部署和实现,以用于部署系统1306。在至少一个实施例中,部署系统1306可以被配置为在分布式计算环境中卸载处理和计算资源,以减少设施1302的基础设施需求。在至少一个实施例中,管线中的一个或更多个应用程序在应用程序执行期间,可以使用或调用部署系统1306的服务(例如,推理、可视化、计算、AI等)。

[0106] 在至少一个实施例中,在高级处理和推理管线中使用的一些应用程序可以使用机器学习模型或其他AI来执行一个或更多个处理步骤。在至少一个实施例中,可以使用在设施1302处生成(并存储在设施1302处的一个或更多个图片存档和通信系统 (PACS) 服务器上)的数据1308 (例如成像数据) 在设施1302处训练机器学习模型,可以使用于自另一个或更多个设施的成像或测序数据1308来训练机器学习模型,或其组合。在至少一个实施例中,训练系统1304可以用于提供应用程序、服务和/或其他资源,以生成用于部署系统1306的工

作的、可部署的机器学习模型。

[0107] 在至少一个实施例中,模型注册表1324可以由对象存储支持,该对象存储可以支持版本控制和对象元数据。在至少一个实施例中,可以从云平台内通过例如云存储(例如,图14的云1426)兼容的应用程序编程接口(API)来访问对象存储。在至少一个实施例中,模型注册表1324内的机器学习模型可以由与API交互的系统的开发者或合作伙伴上传、列出、修改或删除。在至少一个实施例中,API可以提供对方法的访问,所述方法允许具有适当凭证的用户将模型与应用程序相关联,使得模型可以作为应用程序的容器化实例化的执行的一部分来执行。

[0108] 在至少一个实施例中,训练管线1404(图14)可以包括以下情形:其中设施1302正在训练他们自己的机器学习模型,或者具有需要优化或更新的现有机器学习模型。在至少一个实施例中,可以接收由成像设备、测序设备和/或其他类型设备生成的成像数据1308。在至少一个实施例中,一旦接收到成像数据1308,AI辅助注释1310就可以用于帮助生成与成像数据1308相对应的注释,以用作机器学习模型的地面实况数据。在至少一个实施例中,AI辅助注释1310可以包括一个或更多个机器学习模型(例如,卷积神经网络(CNN)),可以对该机器学习模型进行训练,以生成对应于某些类型的成像数据1308(例如,来自某些设备)的注释。在至少一个实施例中,然后AI辅助注释1310可以被直接使用,或者可以使用注释工具进行调整或微调,以生成地面实况数据。在至少一个实施例中,AI辅助注释1310、标记的临床数据1312或其组合可以用作训练机器学习模型的地面实况数据。在至少一个实施例中,经训练的机器学习模型可以被称为输出模型1316,并且可以由部署系统1306使用,如本文所述。

[0109] 在至少一个实施例中,训练管线1404(图14)可以包括以下情形:其中设施1302需要机器学习模型,以用于执行用于部署系统1306中的一个或更多个应用程序的一个或更多个处理任务,但是设施1302当前可能没有这种机器学习模型(或者可能没有为此目的而优化的、高效的或有效的模型)。在至少一个实施例中,可以从模型注册表1324中选择现有的机器学习模型。在至少一个实施例中,模型注册表1324可以包括机器学习模型,其被训练为对成像数据执行各种不同的推理任务。在至少一个实施例中,可以在来自不同的设施(例如,位于远处的设施)而不是设施1302的成像数据上训练模型注册表1324中的机器学习模型。在至少一个实施例中,机器学习模型可能已经在来自一个位置、两个位置或任意数量的位置的成像数据上进行训练。在至少一个实施例中,当在来自特定位置的成像数据上进行训练时,可以在该位置处进行训练,或者至少以保护成像数据的机密性或限制成像数据从场外转移的方式进行训练。在至少一个实施例中,一旦在一个位置处训练了模型或部分地训练了模型,则可以将机器学习模型添加到模型注册表1324。在至少一个实施例中,然后可以在任意数量的其他设施处对机器学习模型进行重新训练或更新,经重新训练或更新的模型可以在模型注册表1324中使用。在至少一个实施例中,然后可以从模型注册表1324中选择机器学习模型(并称为输出模型1316),并且可以在部署系统1306中,以执行用于部署系统的一个或更多个应用程序的一个或更多个处理任务。

[0110] 在至少一个实施例中,在训练管线1404(图14)中,场景可以包括需要机器学习模型的设施1302,以用于执行用于部署系统1306中的一个或更多个应用程序的一个或更多个处理任务,但是设施1302当前可能没有这样的机器学习模型(或者可能没有优化的、高效的

或有效的模型)。在至少一个实施例中,由于用于训练机器学习模型的训练数据的种群差异、稳健性,训练数据异常的多样性,和/或训练数据的其他问题,从模型注册表 1324中选择的机器学习模型可能不会针对在设施1302处生成的成像数据 1308进行微调或优化。在至少一个实施例中,AI辅助注释1310可以用于帮助生成与成像数据1308相对应的注释,以用作训练或更新机器学习模型的地面实况数据。在至少一个实施例中,标记的临床数据1312可以用作训练机器学习模型的地面实况数据。在至少一个实施例中,重新训练或更新机器学习模型可以称为模型训练1314。在至少一个实施例中,模型训练 1314(例如AI辅助注释1310,标记的临床数据1312或其组合)可以用作重新训练或更新机器学习模型的地面实况数据。在至少一个实施例中,训练的机器学习模型可以被称为输出模型1316,并且可以由部署系统1306 使用,如本文所述。

[0111] 在至少一个实施例中,部署系统1306可以包括软件1318、服务1320、硬件1322和/或其他组件、特征和功能。在至少一个实施例中,部署系统 1306可以包括软件“栈”,以使软件1318可以构建在服务1320的顶部上,并且可以使用服务1320来执行一些或全部处理任务,并且服务1320和软件1318可以构建在硬件1322的顶部上,并使用硬件1322来执行部署系统的处理、存储和/或其他计算任务。在至少一个实施例中,软件1318可以包括任意数量的不同容器,其中每个容器可以执行应用程序的实例化。在至少一个实施例中,每个应用程序可以在高级处理和推理管线中执行一个或更多个处理任务(例如,推理、对象检测、特征检测、分割、图像增强、校准等)。在至少一个实施例中,除了接收和配置成像数据以供每个容器使用和/或在通过管线处理后由设施1302使用的容器以外,还可以基于对处理成像数据1308想要的或所需的不同容器的选择来定义高级处理和推理管线(例如,以将输出转换回可用的数据类型。在至少一个实施例中,软件1318内的容器组合(例如,其构成管线)可以被称为虚拟仪器(如本文中更详细地描述的),并且虚拟仪器可以利用服务1320和硬件1322来执行容器中实例化的应用程序的部分或全部处理任务。

[0112] 在至少一个实施例中,数据处理管线可以响应于推理请求(例如,来自部署系统1306的用户的请求)接收具体格式的输入数据(例如,成像数据1308)。在至少一个实施例中,输入数据可以代表由一个或更多个成像设备生成的一个或更多个图像、视频和/或其他数据表示。在至少一个实施例中,数据可以作为数据处理管线的一部分进行预处理,以准备数据用于由一个或更多个应用程序处理。在至少一个实施例中,可以在管线的一个或更多个推理任务或其他处理任务的输出上执行后处理,以准备下一个应用程序的输出数据和/或准备输出数据,以供用户传输和/或使用(例如作为对推理请求的响应)。在至少一个实施例中,推理任务可以由一个或更多个机器学习模型执行,例如经训练或部署的神经网络,所述模型可以包括训练系统1304的输出模型1316。

[0113] 在至少一个实施例中,数据处理管线的任务可以封装在容器中,每个容器表示能够引用机器学习模型的应用程序和虚拟化计算环境的离散、全功能实例化。在至少一个实施例中,容器或应用程序可被发布到容器注册表(本文更详细地描述)的私有(例如,有限访问)区域中,并且经训练或部署的模型可存储在模型注册表1324中,并与一个或更多个应用程序相关联。在至少一个实施例中,应用程序的图像(例如,容器图像)可在容器注册表中使用,并且一旦用户从容器注册表中选择图像以用于在管线中部署,则该图像可用于生成用于应用程序的实例化的容器,以供用户的系统使用。

[0114] 在至少一个实施例中,开发者(例如,软件开发、临床医生、医生等)可以开发、发布和存储应用程序(例如,作为容器),用于对所提供的数据执行图像处理和/或推理。在至少一个实施例中,可以使用与系统相关联的软件开发工具包(SDK)来执行开发、发布和/或存储(例如,以确保开发的应用程序和/或容器符合系统或与系统兼容)。在至少一个实施例中,所开发的应用程序可以使用SDK在本地测试(例如,在第一设施处,对来自第一设施的数据进行测试),所述SDK作为系统(例如图14中的系统1400)可以支持至少某些服务1320。在至少一个实施例中,由于 DICOM对象可能包含一到数百个图像或其他数据类型,并且由于数据的变化,因此开发者可负责管理(例如,设置构造,用于将预处理构建到应用程序中等)传入的数据的提取和准备。在至少一个实施例中,一旦通过系统1400的验证(例如,为了准确性),应用程序便会在容器注册表中可用,以供用户选择和/或实现,以对用户的设施(例如第二设施)处的数据执行一个或更多个处理任务。

[0115] 在至少一个实施例中,开发者随后可以通过网络共享应用程序或容器,以供系统(例如,图14的系统1400)的用户访问和使用。在至少一个实施例中,可以将完成并经过验证的应用程序或容器存储在容器注册表中,并且可以将相关的机器学习模型存储在模型注册表1324中。在至少一个实施例中,请求实体(其提供推理或图像处理请求)可以浏览容器注册表和/或模型注册表1324,以获得应用程序、容器、数据集、机器学习模型等,选择所需的元素组合以包含在数据处理管线中,并提交图像处理请求。在至少一个实施例中,请求可以包括执行请求所必需的输入数据(以及在一些示例中与患者相关的数据),和/或可以包括对在处理请求时要执行的应用程序和/或机器学习模型的选择。在至少一个实施例中,然后将请求传递到部署系统1306的一个或更多个组件(例如,云),以执行数据处理管线的处理。在至少一个实施例中,由部署系统1306进行的处理可以包括引用从容器注册表和/或模型注册表1324中选择的元素(例如,应用程序、容器、模型等)。在至少一个实施例中,一旦通过管线生成结果,结果可返回给用户以供参考(例如,用于在本地、本地工作站或终端上执行的查看应用程序套件中进行查看)。

[0116] 在至少一个实施例中,为了帮助处理或执行管线中的应用程序或容器,可以利用服务1320。在至少一个实施例中,服务1320可以包括计算服务、人工智能(AI)服务、可视化服务和/或其他服务类型。在至少一个实施例中,服务1320可以提供软件1318中的一个或更多个应用程序所共有的功能,因此可以将功能抽象为可以被应用程序调用或利用的服务。在至少一个实施例中,由服务1320提供的功能可以动态且更有效地运行,同时还可以通过允许应用程序并行地处理数据(例如,使用图14中的并行计算平台 1430)来很好地缩放。在至少一个实施例中,不是要求共享服务1320提供的相同功能的每个应用程序都必须具有服务1320的相对应实例,而是可以在各种应用程序之间和之中共享服务1320。在至少一个实施例中,作为非限制性示例,服务可包括可用于执行检测或分割任务的推理服务器或引擎。在至少一个实施例中,可以包括模型训练服务,其可以提供机器学习模型训练和/或重新训练能力。在至少一个实施例中,可以进一步包括数据增强服务,其可以提供GPU加速的数据(例如,DICOM、RIS、CIS、符合REST、RPC、原始等)提取、调整大小、缩放和/或其他增强。在至少一个实施例中,可以使用可视化服务,其可以添加图像渲染效果(例如光线追踪、光栅化、去噪、锐化等),以向二维(2D)和/或三维(3D)模型添加真实感。在至少一个实施例中,可以包括虚拟仪器服务,其为虚拟仪器的管线内的其他应用程序提供波束赋形、分割、推理、成

像和/或支持。

[0117] 在至少一个实施例中,在服务1320包括AI服务(例如,推理服务)的情况下,作为应用程序执行的一部分,可以通过调用(例如,作为API 调用)推理服务(例如,推理服务器),以执行一个或更多个机器学习模型或其处理,来执行一个或更多个机器学习模型。在至少一个实施例中,在另一应用程序包括用于分割任务的一个或更多个机器学习模型的情况下,应用程序可以调用推理服务来执行机器学习模型,以用于执行与分割任务相关联的一个或更多个处理操作。在至少一个实施例中,实现高级处理和推理管线的软件1318,其包括分割应用程序和异常检测应用程序,可以被流水线化,因为每个应用程序可以调用相同的推理服务来执行一个或更多个推理任务。

[0118] 在至少一个实施例中,硬件1322可包括GPU、CPU、图形卡、AI/ 深度学习系统(例如,AI超级计算机,诸如NVIDIA的DGX)、云平台或其组合。在至少一个实施例中,可以使用不同类型的硬件1322,以为部署系统1306中的软件1318和服务1320提供高效的、专门构建的支持。在至少一个实施例中,可以实现使用GPU处理来在AI/深度学习系统内、云系统中、和/或部署系统1306的其他处理组件中进行本地处理(例如,在设施1302处),以提高图像处理和生成的效率、准确性和效能。在至少一个实施例中,作为非限制性示例,关于深度学习、机器学习和/或高性能计算,可以针对GPU处理优化软件1318和/或服务1320。在至少一个实施例中,部署系统1306和/或训练系统1304的计算环境中的至少一些可以在具有GPU优化的软件(例如,NVIDIADGX系统的硬件和软件组合)的数据中心、一个或更多个超级计算机或高性能计算机系统中执行。在至少一个实施例中,如本文所述,硬件1322可包括任意数量的GPU,所述GPU可被调用以并行执行数据处理。在至少一个实施例中,云平台还可包括用于深度学习任务的GPU优化执行、机器学习任务或其他计算任务的GPU处理。在至少一个实施例中,可以使用AI/深度学习超级计算机和/或GPU优化的软件(例如,如在NVIDIA的DGX系统上提供的)作为硬件抽象和缩放平台,来执行云平台(例如,NVIDIA的NGC)。在至少一个实施例中,云平台可以在多个GPU上集成应用程序容器集群系统或协调系统(例如,KUBERNETES),以实现无缝缩放和负载均衡。

[0119] 图14是根据至少一个实施例的用于生成和部署成像部署管线的示例系统1400的系统图。在至少一个实施例中,系统1400可以用于实现图13 的过程1300和/或其他过程,包括高级处理和推理管线。在至少一个实施例中,系统1400可以包括训练系统1304和部署系统1306。在至少一个实施例中,可以使用软件1318、服务1320和/或硬件1322,来实现训练系统 1304和部署系统1306,如本文所述。

[0120] 在至少一个实施例中,系统1400(例如,训练系统1304和/或部署系统1306)可以在云计算环境中(例如,使用云1426)实现。在至少一个实施例中,系统1400可以在本地实现(关于医疗服务设施),或者作为云计算资源和本地计算资源的组合来实现。在至少一个实施例中,可以通过制定安全措施或协议,将对云1426中的API的访问权限限制为授权用户。

[0121] 在至少一个实施例中,安全协议可以包括网络令牌,其可以由认证(例如, AuthN、AuthZ、Gluecon等)服务签名,并且可以携带适当的授权。在至少一个实施例中,虚拟仪器的API(本文中描述)或系统1400的其他实例可以被限制为已被审核或授权用于交互的一组公共IP。

[0122] 在至少一个实施例中,系统1400的各个组件可以使用多种不同网络类型中的任何

一种在彼此之间进行通信,所述不同网络类型包括但不限于经由有线和/或无线通信协议的局域网(LAN)和/或广域网(WAN)。在至少一个实施例中,系统1400的设施和组件之间的通信(例如,用于发送推理请求、用于接收推理请求的结果等)可以通过一个或更多个数据总线、无线数据协议(Wi-Fi)、有线数据协议(例如以太网)等进行传送。

[0123] 在至少一个实施例中,类似于本文关于图13所描述的,训练系统1304 可以执行训练管线1404。在至少一个实施例中,其中部署系统1306将在部署管线1410中使用一个或更多个机器学习模型,训练管线1404可用于训练或重新训练一个或更多个(例如,预训练的)模型,和/或实现一个或更多个预训练模型1406(例如,无需重新训练或更新)。在至少一个实施例中,作为训练管线1404的结果,可以生成输出模型1316。在至少一个实施例中,训练管线1404可以包括任意数量的处理步骤,诸如但不限于成像数据(或其他输入数据)的转换或适配。在至少一个实施例中,对于由部署系统1306使用的不同的机器学习模型,可以使用不同的训练管线1404。在至少一个实施例中,类似于关于图13描述的第一示例的训练管线1404 可用于第一机器学习模型,类似于关于图13描述的第二示例的训练管线 1404可用于第二机器学习模型,类似于关于图13描述的第三示例的训练管线1404可用于第三机器学习模型。在至少一个实施例中,可以根据每个相对应机器学习模型的要求来使用训练系统1304内任务的任何组合。在至少一个实施例中,一个或更多个机器学习模型可能已经被训练并准备好用于部署,因此训练系统1304可能不会对机器学习模型进行任何处理,并且一个或更多个机器学习模型可以由部署系统1306来实现。

[0124] 在至少一个实施例中,根据实现方式或实施例,输出模型1316和/ 或预训练模型1406可包括任何类型的机器学习模型。在至少一个实施例中并且不限于此,系统1400使用的机器学习模型可以包括使用线性回归、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、Boltzmann、深度信念、反卷积、生成对抗、液体状态机等),和/或其他类型的机器学习模型。

[0125] 在至少一个实施例中,训练管线1404可以包括AI辅助注释,如本文关于至少图15B更详细描述。在至少一个实施例中,可以通过任何数量的技术来生成标记的临床数据1312(例如,传统注释)。在至少一个实施例中,在一些示例中可以在绘图程序(例如,注释程序)、计算机辅助设计(CAD)程序、标记程序、适用于生成地面实况的注释或标签的另一类型的应用程序中、和/或可以手绘,生成标签或其他注释。在至少一个实施例中,地面实况数据可以被合成产生(例如,从计算机模型或渲染生成)、真实产生(例如,从真实世界数据设计和生成)、机器自动产生(例如,使用特征分析和学习从数据中提取特征,然后生成标签)、人工注释(例如,标记器或注释专家,定义标签的位置)和/或其组合。在至少一个实施例中,对于成像数据1308(或机器学习模型使用的其他数据类型)的每个实例,可以存在由训练系统1304生成的相对应的地面实况数据。在至少一个实施例中,可以作为部署管线1410的一部分执行AI辅助注释;补充或代替训练管线1404中包括的AI辅助注释。在至少一个实施例中,系统1400 可以包括多层平台,所述多层平台可以包括诊断应用程序(或其他应用程序类型)的软件层(例如软件1318),其可以执行一个或更多个医学成像和诊断功能。在至少一个实施例中,系统1400可以通信地耦合到(例如,经由加密链路)一个或更多个设施的PACS服务器网络。在至少一个实施例中,系统1400可被配置为从PACS服务器访问和引用数据,以执

行操作,例如训练机器学习模型、部署机器学习模型、图像处理、推理和/或其他操作。

[0126] 在至少一个实施例中,软件层可以被实现为安全的、加密的和/或经认证的API,通过所述API可以从外部环境(例如,设施1302)援引(involve)(例如,调用(call))应用程序或容器。在至少一个实施例中,应用程序随后可以调用或执行一个或多个服务1320,以执行与各自的应用程序相关联的计算、AI或可视化任务,并且软件1318和/或服务1320可以利用硬件1322以有效和高效的方式执行处理任务。

[0127] 在至少一个实施例中,部署系统1306可以执行部署管线1410。在至少一个实施例中,部署管线1410可以包括任意数量的应用程序,所述应用程序可以是顺序的、非顺序的,或者以其他方式应用于成像数据(和/或其他数据类型)-包括AI辅助注释,所述成像数据由成像设备、测序设备、基因组学设备等生成,如上所述。在至少一个实施例中,如本文所述,用于个体设备的部署管线1410可以被称为用于设备的虚拟仪器(例如,虚拟超声仪器、虚拟CT扫描仪器、虚拟测序仪器等)。在至少一个实施例中,对于单个设备,可以存在不止一个部署管线1410,这取决于从设备生成的数据所期望的信息。在至少一个实施例中,在期望从MRI机器检测到异常的情况下,可以存在第一部署管线1410,并且在期望从MRI机器的输出进行图像增强的情况下,可以存在第二部署管线1410。

[0128] 在至少一个实施例中,图像生成应用程序可以包括包括使用机器学习模型的处理任务。在至少一个实施例中,用户可能希望使用他们自己的机器学习模型,或者从模型注册表1324中选择机器学习模型。在至少一个实施例中,用户可以实现他们自己的机器学习模型或选择机器学习模型,以包含在执行处理任务的应用程序中。在至少一个实施例中,应用程序可以是可选择的和可定制的,并且通过定义应用程序的构造,针对特定用户的应用程序的部署和实现被呈现为更加无缝的用户体验。在至少一个实施例中,通过利用系统1400的其他特征(例如服务1320和硬件1322),部署管线1410可以更加用户友好,提供更容易的集成,并且产生更准确、高效和及时的结果。

[0129] 在至少一个实施例中,部署系统1306可以包括用户接口1414(例如,图形用户接口、Web接口等),所述用户接口可以被用于选择要包括在部署管线1410中的应用程序、布置应用程序、修改或改变应用程序或其参数或构造、在设置和/或部署期间使用部署管线1410以及与其交互,和/或以其他方式与部署系统1306交互。在至少一个实施例中,尽管没有关于训练系统1304示出,但是用户接口1414(或不同的用户接口)可用于选择在部署系统1306中使用的模型、用于选择用于在训练系统1304中训练或重新训练的模型,和/或用于以其他方式与训练系统1304交互。

[0130] 在至少一个实施例中,除了应用程序协调系统1428之外,还可以使用管线管理器1412来管理部署管线1410的应用程序或容器与服务1320 和/或硬件1322之间的交互。在至少一个实施例中,管线管理器1412可以被配置为促进从应用程序到应用程序、从应用程序到服务1320,和/或从应用程序或服务到硬件1322的交互。在至少一个实施例中,尽管示出为包括在软件1318中,这并不旨在进行限制,并且在一些示例中(例如,如图12所示),管线管理器1412可以被包括在服务1320中。在至少一个实施例中,应用程序协调系统1428(例如,Kubernetes、DOCKER等)可以包括容器协调系统,其可以将应用程序分组到容器中,作为用于协调、管理、缩放和部署的逻辑单元。在至少一个实施例中,通过将来自部署管线1410的应用程序(例如,重建应用程序、分割应用程序等)与各个容器相关联,每个应用程序可以

在自包含的环境(例如,在内核级)中执行,以提高速度和效率。

[0131] 在至少一个实施例中,每个应用程序和/或容器(或其映像)可以被单独开发、修改和部署(例如,第一用户或开发者可以开发、修改和部署第一应用程序,第二用户或开发者可以开发、修改和部署与第一用户或开发者分开的第二应用程序),这可以允许专注并关注单个应用程序和/或容器的任务,而不受另一个应用程序或容器的任务的阻碍。在至少一个实施例中,管线管理器1412和应用程序协调系统1428可以辅助不同容器或应用程序之间的通信和协作。在至少一个实施例中,只要每个容器或应用程序的预期输入和/或输出是系统已知的(例如,基于应用程序或容器的构造),则应用程序协调系统1428和/或管线管理器1412可以促进每个应用程序或容器之间和之中的通信以及资源的共享。在至少一个实施例中,由于部署管线1410中的一个或多个应用程序或容器可以共享相同的服务和资源,因此应用程序协调系统1428可以在各个应用程序或容器之间和之中进行协调、负载均衡,并确定服务或资源的共享。在至少一个实施例中,调度器可用于追踪应用程序或容器的资源需求、这些资源的当前使用或计划使用,以及资源可用性。因此,在至少一个实施例中,考虑到系统的需求和可用性,调度器可以将资源分配给不同的应用程序,并在应用程序之间和之中分配资源。在一些示例中,调度器(和/或应用程序协调系统1428的其他组件)可以基于施加在系统上的约束(例如,用户约束)来确定资源可用性和分布,例如服务质量(QoS)、对数据输出的迫切需求(例如,以确定是执行实时处理还是延迟处理)等。

[0132] 在至少一个实施例中,由部署系统1306中的应用程序或容器利用并由其共享的服务1320,可以包括计算服务1416、AI服务1418、可视化服务1420和/或其他服务类型。在至少一个实施例中,应用程序可以调用(例如,执行)一个或多个服务1320,以执行针对应用程序的处理操作。在至少一个实施例中,应用程序可以利用计算服务1416来执行超级计算或其他高性能计算(HPC)任务。在至少一个实施例中,可以利用一个或多个计算服务1416来执行并行处理(例如,使用并行计算平台1430),以通过一个或多个应用程序和/或单个应用程序的一个或多个任务基本上同时地处理数据。在至少一个实施例中,并行计算平台1430(例如,NVIDIA的CUDA)可以在GPU(GPGPU)(例如,GPU1422)上实现通用计算。在至少一个实施例中,并行计算平台1430的软件层可以提供对GPU的虚拟指令集和并行计算元素的访问,以执行计算内核。在至少一个实施例中,并行计算平台1430可以包括存储器,并且在一些实施例中,可以在多个容器之间和之中,和/或在单个容器内的不同处理任务之间和之中共享存储器。在至少一个实施例中,可以为多个容器和/或容器内的多个进程生成进程间通信(IPC)调用,以使用于自并行计算平台1430的共享存储器段的相同数据(例如,其中一应用程序或多个应用程序的多个不同阶段正在处理相同的信息)。在至少一个实施例中,不是复制数据并将数据移动到存储器中的不同位置(例如,读/写操作),而是可以将存储器相同位置中的相同数据用于任何数量的处理任务(例如,在同一时间、不同时间等)。在至少一个实施例中,由于作为处理的结果数据被用于生成新数据,因此数据的新位置的该信息可以在各个应用程序之间存储和共享。在至少一个实施例中,数据的位置以及经更新或修改的数据的位置可以是理解容器中的有效负载的定义的一部分。

[0133] 在至少一个实施例中,可以利用AI服务1418来执行推理服务,该推理服务用于执行与应用程序相关联的机器学习模型(例如,任务为执行应用程序的一个或多个处理任务)。在至少一个实施例中,AI服务1418可以利用AI系统1424来执行机器学习模型(例如,

诸如CNN之类的神经网络)以用于分割、重建、对象检测、特征检测、分类和/或其他推理任务。在至少一个实施例中,部署管线1410的应用程序可以使用于自训练系统1304的一个或更多个输出模型1316和/或应用程序的其他模型,来对成像数据执行推理。在至少一个实施例中,使用应用程序协调系统1428(例如,调度器)进行推理的两个或更多个示例可以是可用的。在至少一个实施例中,第一类别可以包括高优先级/低延时路径,其可以实现更高服务水平协议,例如用于在紧急情况下对紧急请求执行推理,或者在诊断过程中用于放射科医生。在至少一个实施例中,第二类别可以包括标准优先级路径,其可用于可能不紧急的请求或者可以在稍后的时间执行分析的情况。在至少一个实施例中,应用程序协调系统1428可以基于优先级路径来分配资源(例如,服务1320和/或硬件1322),以用于AI服务1418的不同推理任务。

[0134] 在至少一个实施例中,共享存储器可以被安装到系统1400中的AI 服务1418。在至少一个实施例中,共享存储器可以操作为高速缓存(或其他存储设备类型),并且可以用于处理来自应用程序的推理请求。在至少一个实施例中,当提交推理请求时,部署系统1306的一组API实例可以接收请求,并且可以选择一个或更多个实例(例如,为了最佳拟合、为了负载均衡等)来处理请求。在至少一个实施例中,为了处理请求,可以将请求输入到数据库中,如果尚未在高速缓存中,则可以从模型注册表1324 定位机器学习模型,验证步骤可以确保将适当的机器学习模型加载到高速缓存中(例如,共享存储),和/或可以将模型的副本保存到高速缓存中。在至少一个实施例中,如果应用程序尚未运行或没有足够的应用程序的实例,则可使用调度器(例如,管线管理器1412的调度器)来启动在请求中引用的应用程序。在至少一个实施例中,如果尚未启动推理服务器来执行模型,则可以启动推理服务器。每个模型可以启动任意数量的推理服务器。在至少一个实施例中,在将推理服务器聚类的拉(pull)模型中,每当负载均衡有利时,就可以将模型高速缓存。在至少一个实施例中,推理服务器可以静态加载到相应的分布式服务器中。

[0135] 在至少一个实施例中,可以使用在容器中运行的推理服务器来执行推理。在至少一个实施例中,推理服务器的实例可以与模型(并且可选地与模型的多个版本)相关联。在至少一个实施例中,如果在接收到对模型执行推理的请求时推理服务器的实例不存在,则可以加载新实例。在至少一个实施例中,当启动推理服务器时,可以将模型传递到推理服务器,使得可以使用相同的容器来服务不同的模型,只要推理服务器作为不同的实例运行即可。

[0136] 在至少一个实施例中,在应用程序执行期间,可以接收对给定应用程序的推理请求,并且可以加载(如果尚未加载的话)容器(例如,托管推理服务器的实例),以及可以调用启动程序。在至少一个实施例中,容器中的预处理逻辑可以(例如,使用CPU和/或GPU)对传入的数据进行加载、解码和/或执行任何附加的预处理。在至少一个实施例中,一旦数据准备好进行推理,容器就可以根据需要对数据进行推理。在至少一个实施例中,这可以包括对一个图像(例如,手部X光线)的单个推理调用,或可要求对数百个图像(例如,胸部CT)进行推理。在至少一个实施例中,应用程序可在完成之前总结结果,其可以包括但不限于单个置信度得分、像素级分割、体素级分割、生成可视化或生成文本以总结结果。在至少一个实施例中,可以为不同的模型或应用程序分配不同的优先级。例如,一些模型可具有实时(TAT小于1分钟)优先级,而其他模型可具有较低的优先级(例如,TAT小于10分钟)。在至少一个实

施例中,模型执行时间可以从请求机构或实体进行测量,并且可以包括合作网络遍历时间以及推理服务的执行时间。

[0137] 在至少一个实施例中,请求在服务1320和推理应用程序之间的传送可以隐藏在软件开发工具包(SDK)后面,并且可以通过队列提供鲁棒的传输。在至少一个实施例中,将通过API将请求放置在队列中,以用于个体应用程序/租户ID组合,并且SDK将从队列中拉取请求并将请求提供给应用程序。在至少一个实施例中,在SDK将从中拾取队列的环境中,可以提供队列的名称。在至少一个实施例中,通过队列的异步通信可能有用,因为它可以允许应用程序的任何实例在其可用时拾取工作。可以通过队列将结果传送回去,以确保没有数据丢失。在至少一个实施例中,队列还可以提供对工作进行分割的能力,因为最高优先级的工作可以进入与应用程序的大多数实例连接的队列,而最低优先级的工作可以进入与单个实例连接的队列,所述实例按照接收到的顺序处理任务。在至少一个实施例中,应用程序可以在GPU加速的实例上运行,所述实例在云1426中生成,并且推理服务可以在GPU上执行推理。

[0138] 在至少一个实施例中,可以利用可视化服务1420来生成用于查看应用程序和/或部署管线1410输出的可视化。在至少一个实施例中,可视化服务1420可以利用GPU1422来生成可视化。在至少一个实施例中,可视化服务1420可以实现诸如光线追踪之类的渲染效果,以生成更高质量的可视化。在至少一个实施例中,可视化可以包括但不限于2D图像渲染、3D体渲染、3D体重建、2D层析切片、虚拟现实显示、增强现实显示等。在至少一个实施例中,可以使用虚拟化环境来生成虚拟交互显示或环境(例如,虚拟环境),以供系统用户(例如,医生、护士、放射科医生等)进行交互。在至少一个实施例中,可视化服务1420可以包括内部可视化器、电影和/或其他渲染或图像处理能力或功能(例如,光线追踪、光栅化、内部光学器件等)。

[0139] 在至少一个实施例中,硬件1322可以包括GPU 1422、AI系统1424、云1426和/或用于执行训练系统1304和/或部署系统1306的任何其他硬件。在至少一个实施例中,GPU 1422(例如,NVIDIA的TESLA和/或QUADRO GPU)可包括可用于执行计算服务1416、AI服务1418、可视化服务1420、其他服务和/或软件1318的任何特征或功能的处理任务的任意数量的GPU。例如,对于AI服务1418,GPU 1422可用于对成像数据(或机器学习模型使用的其他数据类型)执行预处理,对机器学习模型的输出执行后处理和/或执行推理(例如以执行机器学习模型)。在至少一个实施例中,云1426、AI系统1424和/或系统1400的其他组件可以使用GPU1422。在至少一个实施例中,云1426可以包括用于深度学习任务的GPU优化的平台。在至少一个实施例中,AI系统1424可以使用GPU,并且可以使用一个或更多个AI系统1424来执行云1426(或者任务为深度学习或推理的至少部分)。同样,尽管硬件1322被示出为离散组件,但这并不意图是限制,并且硬件 1322的任何组件可以与硬件1322的任何其他组件组合,或由硬件1322的任何其他组件利用。

[0140] 在至少一个实施例中,AI系统1424可包括专门构建的计算系统(例如,超级计算机或HPC),该计算系统配置用于推理、深度学习、机器学习和/或其他人工智能任务。在至少一个实施例中,除了CPU、RAM、存储器和/或其他组件、特征或功能之外,AI系统1424(例如,NVIDIA的 DGX)还可以包括可以使用多个GPU1422来执行分GPU优化的软件(例如,软件栈)。在至少一个实施例中,可以在云1426中(例如,在数据中心中)实现一个或更多个AI系统1424,以执行系统1400的一些或全部基于AI的处理任务。

[0141] 在至少一个实施例中,云1426可以包括GPU加速的基础设施(例如,NVIDIA的NGC),其可以提供用于执行系统1400的处理任务的GPU优化的平台。在至少一个实施例中,云1426可以包括AI系统1424,其用于执行系统1400的一个或多个基于AI的任务(例如,作为硬件抽象和缩放平台)。在至少一个实施例中,云1426可以与利用多个GPU的应用程序协调系统1428集成,以实现应用程序和服务1320之间和之中的无缝缩放和负载均衡。在至少一个实施例中,如本文所述,云1426可以负责执行系统1400的至少一些服务1320,包括计算服务1416、AI服务1418和/或可视化服务1420。在至少一个实施例中,云1426可以执行大小批的推理(例如,执行NVIDIA的TENSORRT),提供加速的并行计算API和平台1430(例如,NVIDIA的CUDA),执行应用程序协调系统1428(例如,KUBERNETES),提供图形渲染API和平台(例如,用于光线追踪,2D图形、3D图形和/或其他渲染技术以产生更高质量的电影效果),和/或可以为系统1400提供其他功能。

[0142] 图15A示出了根据至少一个实施例的用于训练、重新训练或更新机器学习模型的过程1500的数据流图。在至少一个实施例中,可以使用作为非限制性示例的图14的系统1400来执行过程1500。在至少一个实施例中,过程1500可以利用系统1400的服务1320和/或硬件1322,如本文所述。在至少一个实施例中,由过程1500生成的精炼模型1512可以由部署系统1306针对部署管线1410中的一个或多个容器化的应用程序执行。

[0143] 在至少一个实施例中,模型训练1314可包括使用新的训练数据(例如,新的输入数据(诸如客户数据集1506),和/或与输入数据相关联的新的地面实况数据)重新训练或更新初始模型1504(例如,预训练模型)。在至少一个实施例中,为了重新训练或更新初始模型1504,可以重置或删除初始模型1504的输出或损失层,和/或用更新的或新的输出或损失层代替。在至少一个实施例中,初始模型1504可以具有从先前的训练中保留下来的先前精细调整的参数(例如,权重和/或偏差),因此训练或重新训练1314可能不需要花费与从头开始训练模型一样长的时间或不需要那么多的处理。在至少一个实施例中,在模型训练1314期间,通过重置或替换初始模型1504的输出或损失层,在新的客户数据集1506(例如图13的图像数据1308)上生成预测时,可以基于与输出或损失层的精度相关联的损失计算,更新和重新调整新数据集的参数。

[0144] 在至少一个实施例中,可以将经预训练的模型1406存储在数据存储或注册表中(例如,图13的模型注册表1324)。在至少一个实施例中,经预训练的模型1406可能已经至少部分地在除了执行过程1500的设施之外的一个或多个设施处被训练。在至少一个实施例中,为了保护患者、受试者或不同设施的客户的隐私和权利,经预训练的模型1406可能已经使用本地生成的客户或患者数据在本地进行了训练。在至少一个实施例中,可以使用云1426和/或其他硬件1322来训练经预训练的模型1406,但是机密的、受隐私保护的的患者数据可以不被传送到云1426的任何组件(或其他非本地硬件)、由其使用或由其访问。在至少一个实施例中,如果使用于自不止一个设施的患者数据来训练经预训练的模型1406,则在来自另一设施的患者或客户数据上进行训练之前,经预训练的模型1406可能已经针对每个设施进行了单独训练。在至少一个实施例中,例如在客户或患者数据已发布隐私问题(例如,通过放弃,用于实验用途等),或者其中客户或患者数据包括在公共数据集中的情况下,来自任意数量的设施的客户或患者数据可以用于在本地和/或外部训练经预训练的模型1406,例如在数据中心中或其他云计算基础设施中。

[0145] 在至少一个实施例中,在选择应用程序以在部署管线1410中使用时,用户还可以选择用于特定应用程序的机器学习模型。在至少一个实施例中,用户可能没有模型以使用,因此用户可以选择要与应用程序一起使用的经预训练的模型1406。在至少一个实施例中,经预训练的模型1406可能没有被优化用于在用户设施的客户端数据集1506上生成准确的结果(例如,基于患者多样性、人口统计、所使用的医学成像设备的类型等)。在至少一个实施例中,在将经预训练的模型1406部署到部署管线1410中以与一个或多个应用程序一起使用之前,经预训练的模型1406可以被更新、重新训练和/或微调,以用于在各个设施处使用。

[0146] 在至少一个实施例中,用户可以选择要更新、重新训练和/或微调的经预训练的模型1406,并且经预训练的模型1406可以称为过程1500中训练系统1304的初始模型1504。在至少一个实施例中,客户端数据集1506(例如,成像数据、基因组数据、测序数据或由设施处的设备生成的其他数据类型)可用于对初始模型1504执行模型训练1314(其可包括但不限于传递学习),以生成精炼模型1512。在至少一个实施例中,可以由训练系统1304生成与客户端数据集1506相对应的地面实况数据。在至少一个实施例中,可以至少部分地由临床医生、科学家、医生、从业者在设施处生成地面实况数据(例如,如图13中的标记的临床数据1312)。

[0147] 在至少一个实施例中,在一些示例中可以使用AI辅助注释1310来生成地面实况数据。在至少一个实施例中,AI辅助注释1310(例如,使用AI辅助注释SDK实现)可以利用机器学习模型(例如,神经网络)来生成用于客户端数据集的建议或预测的地面实况数据。在至少一个实施例中,用户1510可以在计算设备1508上的用户界面(图形用户界面(GUI))内使用注释工具。

[0148] 在至少一个实施例中,用户1510可以经由计算设备1508与GUI交互,以编辑或微调注释或自动注释。在至少一个实施例中,多边形编辑特征可以用于将多边形的顶点移动到更精确或微调的位置。

[0149] 在至少一个实施例中,一旦客户端数据集1506具有相关联的地面实况数据,则地面实况数据(例如,来自AI辅助注释、手动标记等)可以在模型训练1314期间用于生成精炼模型1512。在至少一个实施例中,客户端数据集1506可以被应用到初始模型1504任意次数,并且地面实况数据可以用于更新初始模型1504的参数,直到对于精炼模型1512达到可接受的精度水平为止。在至少一个实施例中,一旦生成精炼模型1512,就可以在设施处的一个或多个部署管线1410内部署精炼模型1512,以用于执行关于医学成像数据的一个或多个处理任务。

[0150] 在至少一个实施例中,可以将精炼模型1512上传到模型注册表1324中的经预训练的模型1406,以由另一个设施选择。在至少一个实施例中,他的过程可以在任意数量的设施处完成,使得可以在新数据集上对精炼模型1512进一步精炼任意次数,以生成更通用的模型。

[0151] 图15B是根据至少一个实施例的用于利用经预训练的注释模型来增强注释工具的客户-服务器架构1532的示例图示。在至少一个实施例中,可以基于客户-服务器架构1532来实例化AI辅助注释工具1536。在至少一个实施例中,成像应用程序中的注释工具1536可以帮助放射科医生,例如识别器官和异常。在至少一个实施例中,成像应用程序可以包括软件工具,作为非限制性示例,所述软件工具帮助用户1510识别原始图像1534中(例如,在3DMRI或CT扫描中)的特定感兴趣器官上的几个极值点,并接收特定器官的所有2D切

片的自动注释结果。在至少一个实施例中,结果可以作为训练数据1538存储在数据存储中,并且用作(例如但不限于)用于训练的地面实况数据。在至少一个实施例中,当计算设备1508发送用于AI辅助注释1310的极值点时,例如,深度学习模型可以接收该数据作为输入并返回分割器官或异常的推理结果。在至少一个实施例中,预实例化的注释工具(例如图15B中的AI辅助注释工具1536B)可以通过对服务器(诸如注释助手服务器1540)进行API调用(例如API调用1544)来增强,注释助手服务器1540可包括存储在例如注释模型注册表中的一组经预训练的模型1542。在至少一个实施例中,注释模型注册表可以存储经预训练的模型1542(例如,机器学习模型,诸如深度学习模型),其被预训练以对特定器官或异常执行AI辅助注释。在至少一个实施例中,可以通过使用训练管线1404来进一步更新这些模型。在至少一个实施例中,随着添加新标记的临床数据1312,可以随时间改进预安装的注释工具。

[0152] 此类组件可以用于使用基于光线追踪的重要性采样渲染图像,可以通过硬件进行加速。

[0153] 其他变体符合本发明的精神。因此,虽然所公开的技术容易受到各种修改和替代构造的影响,但其某些图示实施例在附图中示出,并已在上文中详细描述。然而,应当理解,无意将公开限制为特定形式或所公开的形式,相反,意图是涵盖属于所附权利要求书中定义的公开精神和范围内的所有修改、替代构造和等价物。

[0154] 在描述所公开的实施例的上下文中(特别是在以下权利要求的上下文中),术语“一”和“一个”以及“所述”和类似指代的使用应被解释为涵盖单数和复数,除非本文中另有说明或与上下文明确矛盾,而不是作为术语的定义。除非另有说明,否则术语“包含”、“具有”、“包括”和“包含”应解释为开放式术语(指“包括但不限于”)。术语“连接”未经修改并指物理连接时,应解释为部分或全部包含在内、附接或连接在一起,即使存在某些介入。除非本文另有说明,否则本文中数值范围的引用仅旨在用作分别指代范围内每个单独数值的简写方法,并且每个单独数值被纳入说明书中,如同其在本文中被单独叙述一样。除非上下文另有说明或与上下文矛盾,否则术语“集”(例如“项目项”)或“子集”的使用应解释为包含一个或更多个成员的非空集合。此外,除非上下文另有说明或与上下文矛盾,否则术语相应集的“子集”不一定表示对应集的适当子集,但子集和对应集可以相等。

[0155] 除非以其他方式明确说明或与上下文明确矛盾,否则诸如“A、B和C中的至少一个”或“A、B与C中的至少一个”形式的短语之类的连接语与上下文一起理解,通常用于表示项目、条款等,其可以是A或B或C,也可以是A、B和C集合的任何非空子集。例如,在具有三个成员的集合的示例中,连接短语“A、B和C中的至少一个”和“A、B与C中的至少一个”指以下集合中的任何一个: {A}、{B}、{C}、{A、B}、{A、C}、{B、C}、{A、B、C}。因此,这种连接语言通常并不意味着某些实施例要求存在A中的至少一个、B中的至少一个和C中的至少一个。此外,除非另有说明或与上下文相矛盾,否则术语“多个”表示复数状态(例如,“多个项目”表示多个项目)。复数是至少两个项目,但当如果明确指示或通过上下文指示时可以更多。此外,除非另有说明或上下文另有明确说明,否则短语“基于”是指“至少部分基于”而非“仅基于”。

[0156] 除非本文另有说明或与上下文明确矛盾,否则本文描述的过程的操作可以按任何适当的顺序进行。在至少一个实施例中,在配置有可执行指令的一个或更多个计算机系统的控制下执行诸如本文所述的那些过程(或其变体和/或组合)之类的过程,并且被实现为代码(例如,可执行指令、一个或更多个计算机程序或一个或更多个应用程序),该代码通过

硬件或其组合在一个或多个处理器上共同执行。在至少一个实施例中,代码以例如计算机程序的形式存储在计算机可读存储介质上,该计算机程序包括可由一个或多个处理器执行的多个指令。在至少一个实施例中,计算机可读存储介质是非暂时性计算机可读存储介质,其不包括暂时性信号(例如,传播的瞬态电或电磁传输),但包括瞬态信号收发器内的非暂时性数据存储电路(例如,缓冲器、高速缓存和队列)。在至少一个实施例中,代码(例如,可执行代码或源代码)存储在一组一个或多个非暂时性计算机可读存储介质上,该存储介质上存储了可执行指令(或用于存储可执行指令的其他存储器),当由计算机系统的一个或多个处理器执行时(即,由于执行的结果),使计算机系统执行本文所述的操作。在至少一个实施例中,一组非暂时性计算机可读存储介质包括多个非暂时性计算机可读存储介质和多个非暂时性计算机可读存储介质的一个或多个单个非暂时性存储介质缺少所有代码,而多个非暂时性计算机可读存储介质共同存储所有代码。在至少一个实施例中,执行可执行指令,使得不同的指令由不同的处理器执行——例如,非暂时性计算机可读存储介质存储指令,主中央处理单元(“CPU”)执行一些指令,而图形处理单元(“GPU”)执行其他指令。在至少一个实施例中,计算机系统的不同组件具有单独的处理器,并且不同的处理器执行不同的指令子集。

[0157] 因此,在至少一个实施例中,计算机系统被配置为实现单独地或共同地执行本文所述过程的操作的一个或多个服务,并且这样的计算机系统被配置为具有能够执行操作的适用硬件和/或软件。此外,实现本公开的至少一个实施例的计算机系统是单个设备,并且在另一个实施例中是分布式计算机系统,其包括不同操作的多个设备,使得分布式计算机系统执行本文所述的操作,并且单个设备不执行所有操作。

[0158] 本文提供的任何和所有示例或示例性语言(例如,“诸如”)的使用仅旨在更好地阐明本公开的实施例,并且除非另有声明,否则不会对本公开的范围构成限制。说明书中的任何语言都不应被解释为表示任何未要求保护的元素对实践公开内容是必不可少的。

[0159] 本文引用的所有参考文献,包括出版物、专利申请和专利,在此均通过引用并入本文,其程度就如同每个参考文献被单独且具体地指示为以引用的方式并入本文并且其全部内容在本文中阐述一样。

[0160] 在说明书和权利要求书中,可以使用术语“耦合”和“连接”及它们的派生词。应当理解,这些术语可能不旨在作为彼此的同义词。相反,在特定示例中,“连接”或“耦合”可用于指示两个或多个元件彼此直接或间接物理或电接触。“耦合”还可能意味着两个或多个元素彼此不直接接触,但仍相互合作或交互。

[0161] 除非另有特别说明,否则可以理解,在整个说明书中,例如“处理”、“计算”、“计算”、“确定”等之类的术语,是指操作和/或转换表示为物理数据(例如电子数据)的计算机或计算系统或类似电子计算设备的动作和/或过程,将计算系统寄存器和/或存储器中的量转换为其他数据,类似地表示为计算系统存储器、寄存器或其他此类信息存储、传输或显示设备中的物理量。

[0162] 以类似方式,术语“处理器”可指处理来自寄存器和/或存储器的电子数据并将该电子数据转换为可存储在寄存器和/或存储器中的其他电子数据的任何设备或设备的部分。作为非限制性示例,“处理器”可以是CPU或GPU。“计算平台”可以包括一个或多个处理器。如本文所用,“软件”过程可包括例如随时间执行工作的软件和/或硬件实体,例如任

务、线程和智能代理。此外，每个过程可以指多个过程，用于顺序或并行、连续或间歇地执行指令。术语“系统”和“方法”在本文中互换使用，只要系统可以体现一个或更多个方法，并且方法可以被视为一个系统。

[0163] 在本文件中，可参考获取、获得、接收或将模拟或数字数据输入子系统、计算机系统或计算机实现的机器。可以通过多种方式获得、获取、接收或输入模拟和数字数据，例如通过接收数据作为函数调用的参数或对应用程序编程接口的调用来实现。在一些实现中，可以通过串行或并行接口传输数据来完成获得、获取、接收或输入模拟或数字数据的过程。在另一个实现中，获得、获取、接收或输入模拟或数字数据的过程可以通过经由计算机网络将数据从提供实体传输到获取实体来完成。还可以参考提供、输出、传输、发送或呈现模拟或数字数据。在各种示例中，提供、输出、传输、发送或呈现模拟或数字数据的过程可以通过将数据传输为函数调用的输入或输出参数、应用程序编程接口的参数或进程间通信机制来实现。

[0164] 尽管本文的描述阐述了所描述的技术的示例实现，但其他架构可用于实现所描述的功能，并且旨在落入本公开的范围内。此外，尽管出于描述的目的，在上文定义了具体的职责分配，但根据具体情况，可以以不同的方式分配和划分各种职能和责任。

[0165] 此外，尽管主题已经用特定于结构特征和/或方法行为的语言描述，但应当理解，所附权利要求中要求保护的主体不一定限于所描述的特定特征或行为。而是，公开了特定的特征和动作作为实现权利要求的示例性形式。

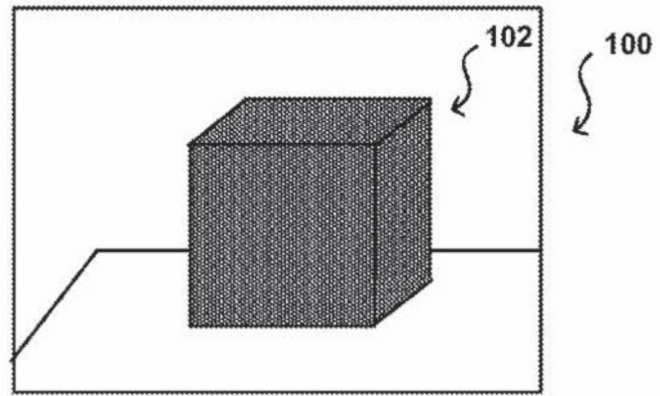


图1A

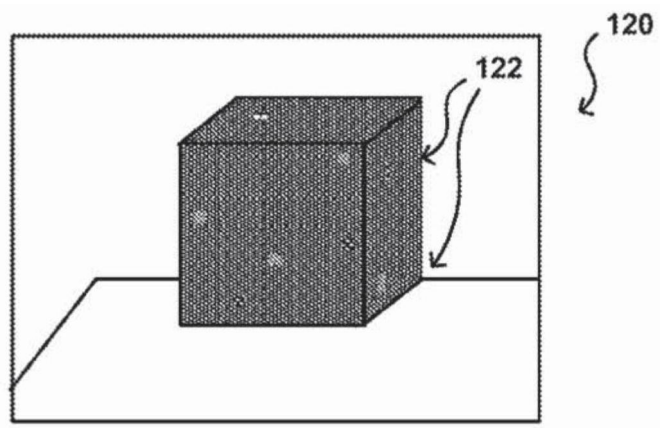


图1B

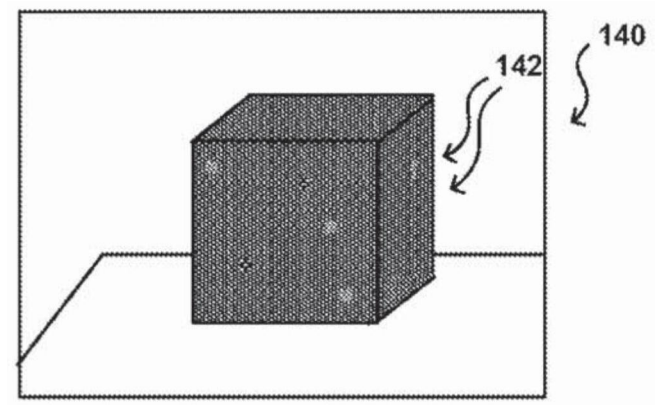


图1C

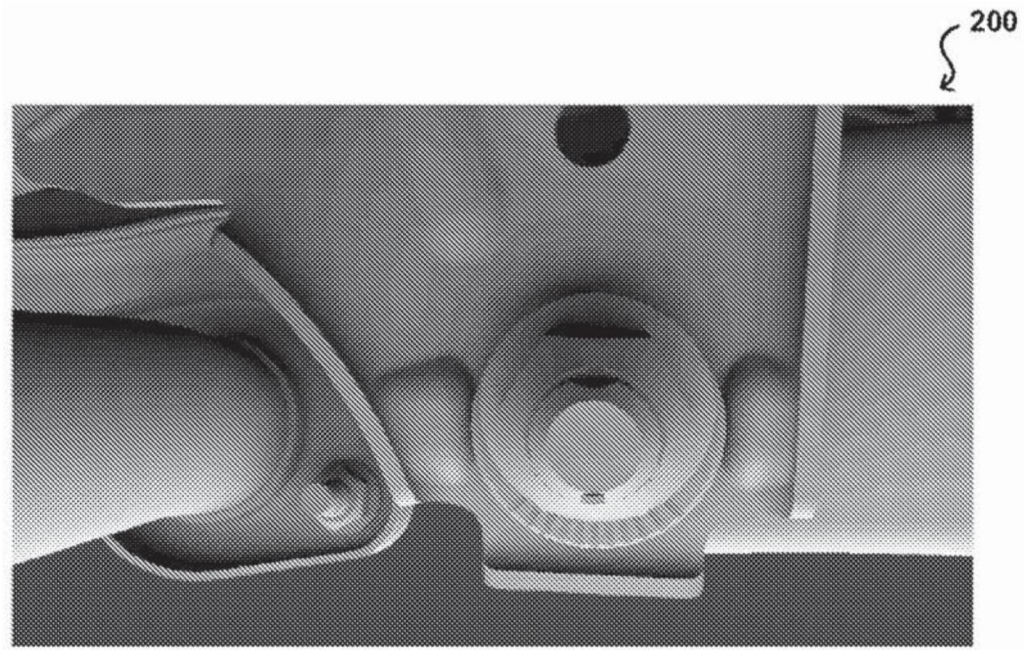


图2A

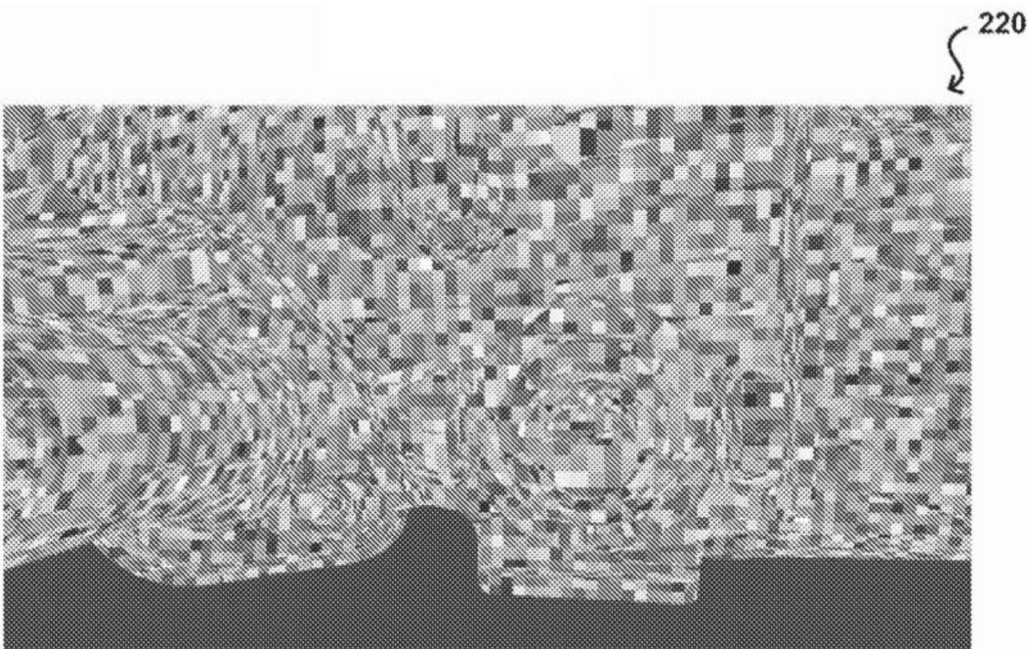


图2B

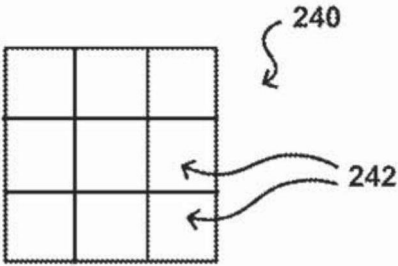


图2C

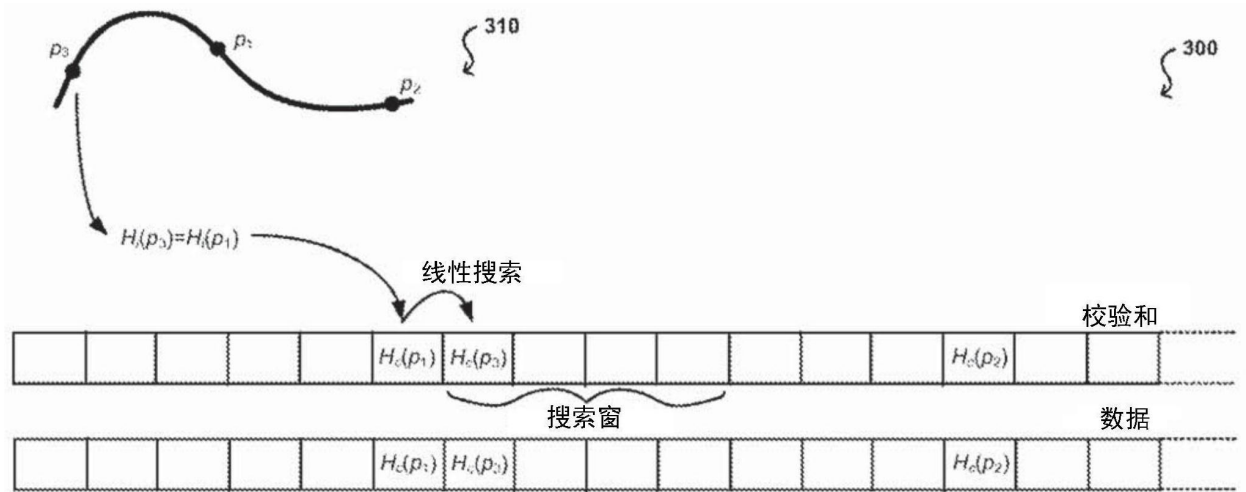


图3A

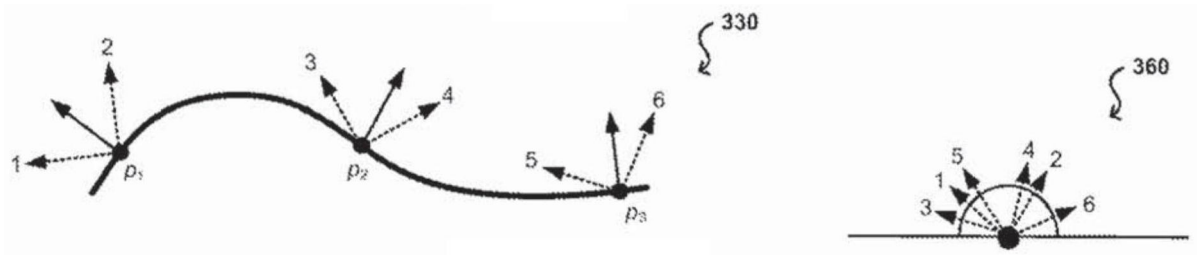


图3B

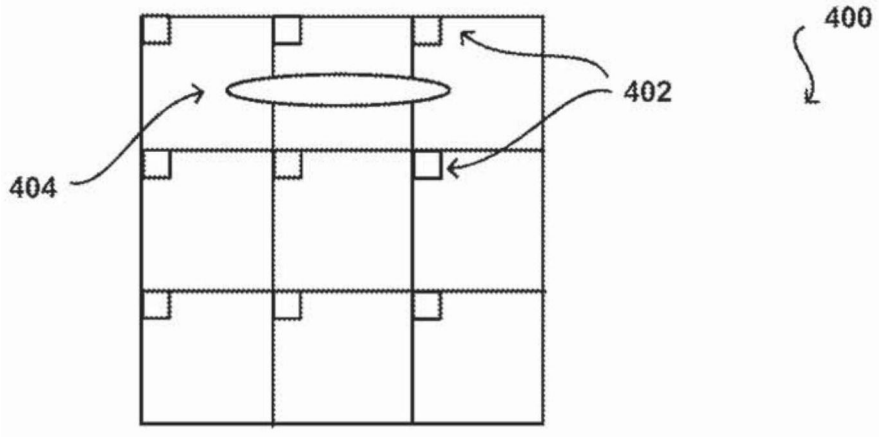


图4A

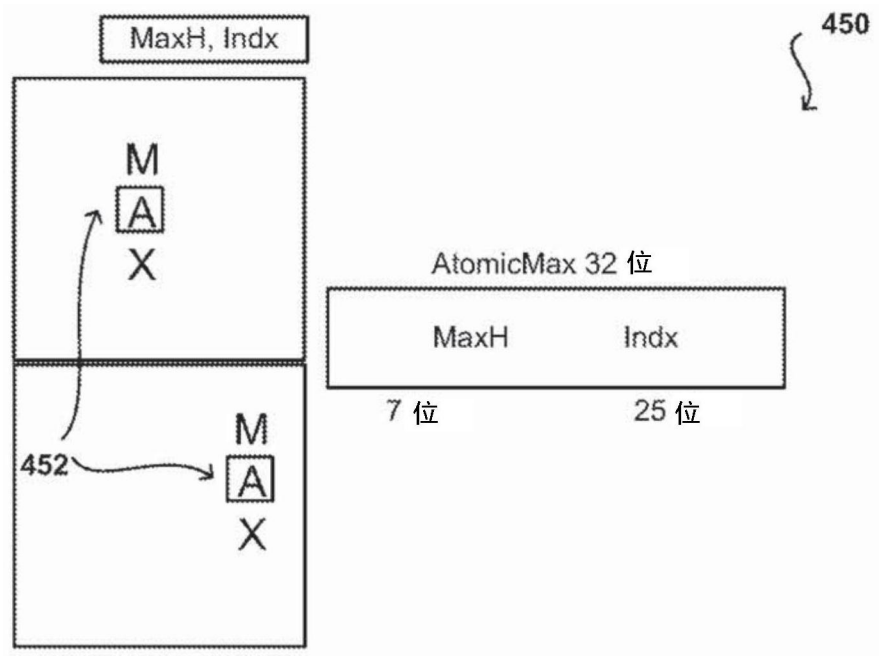


图4B

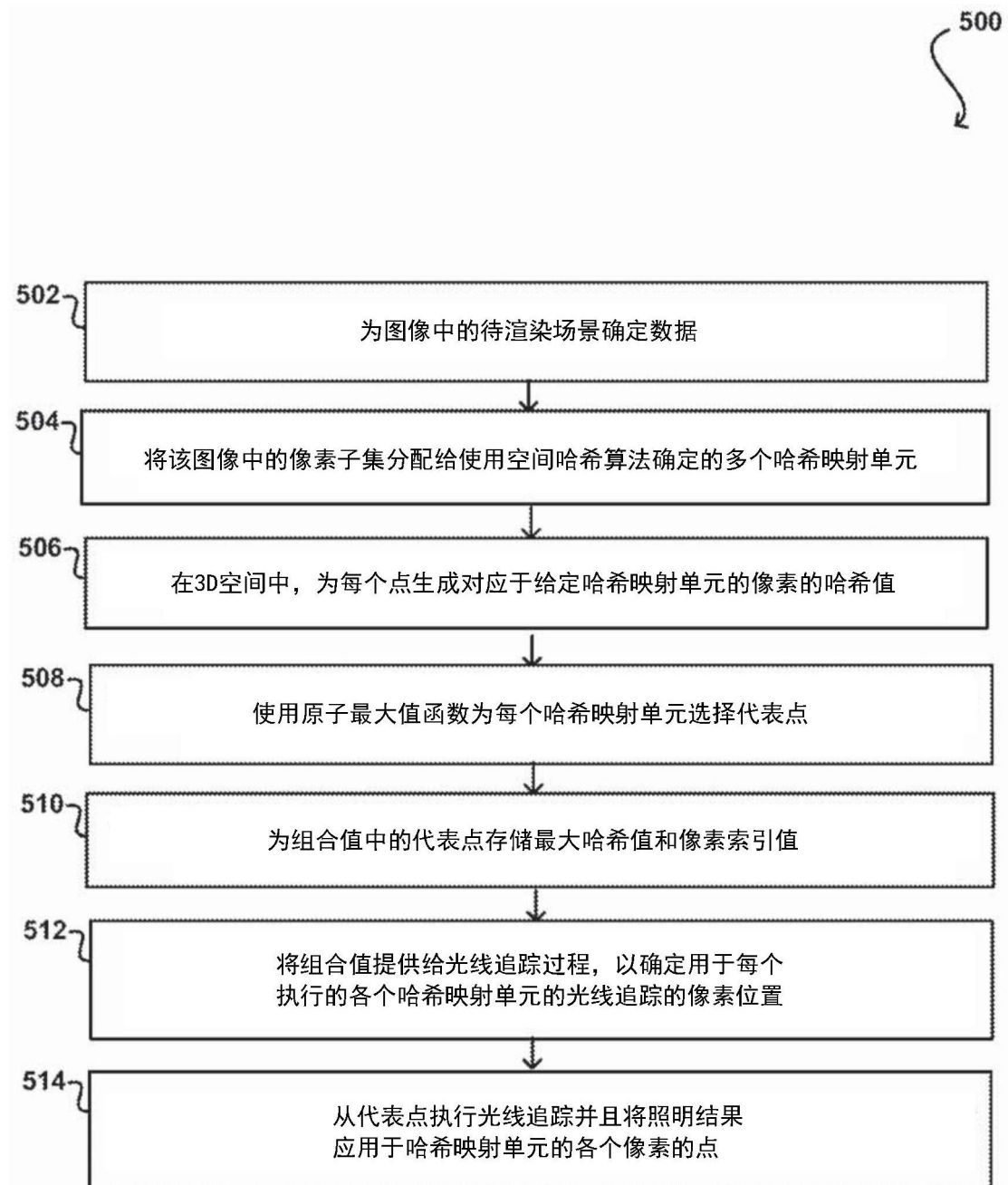


图5

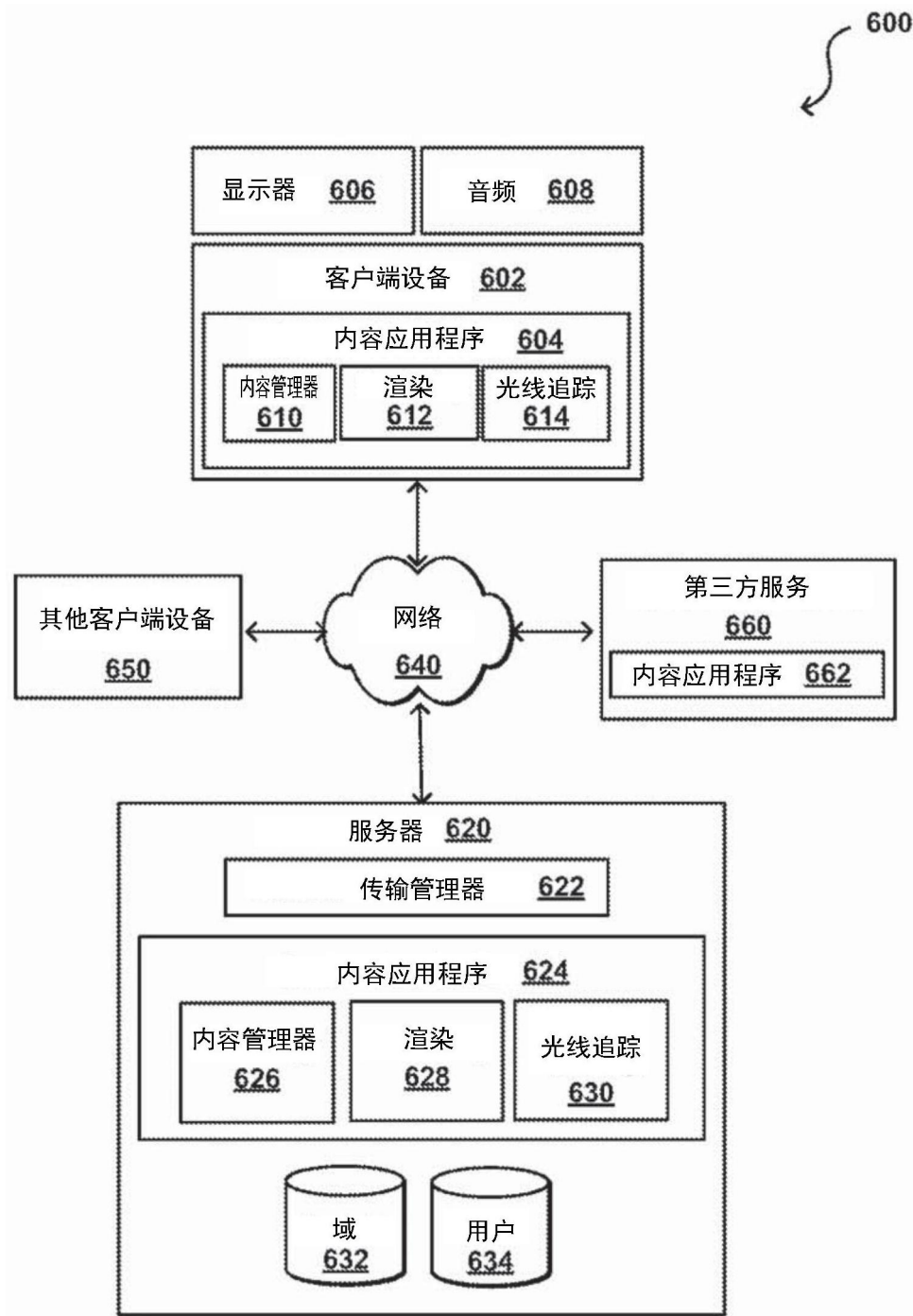


图6

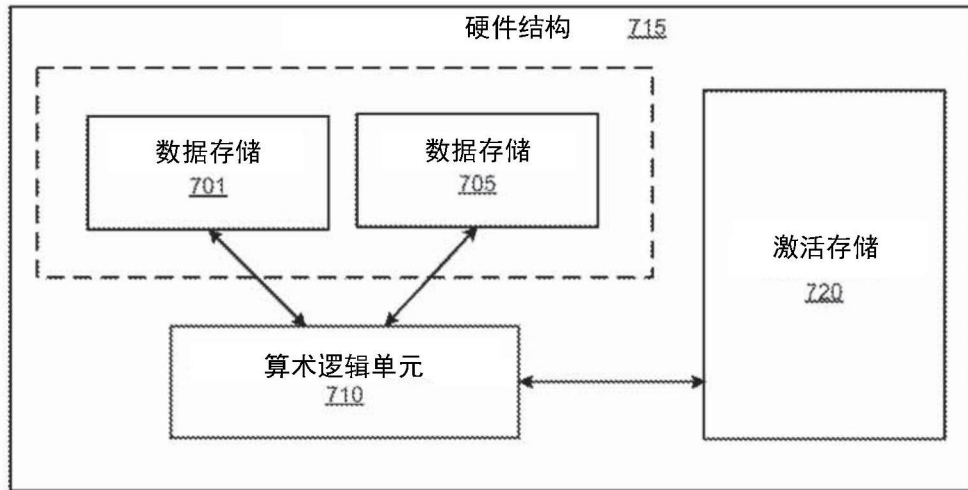


图7A

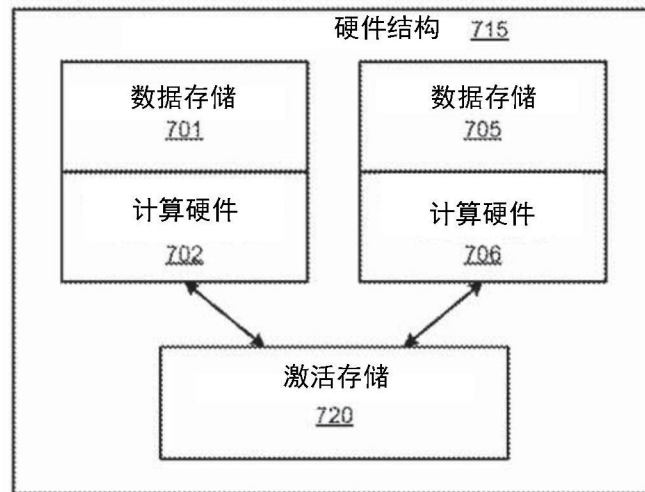


图7B

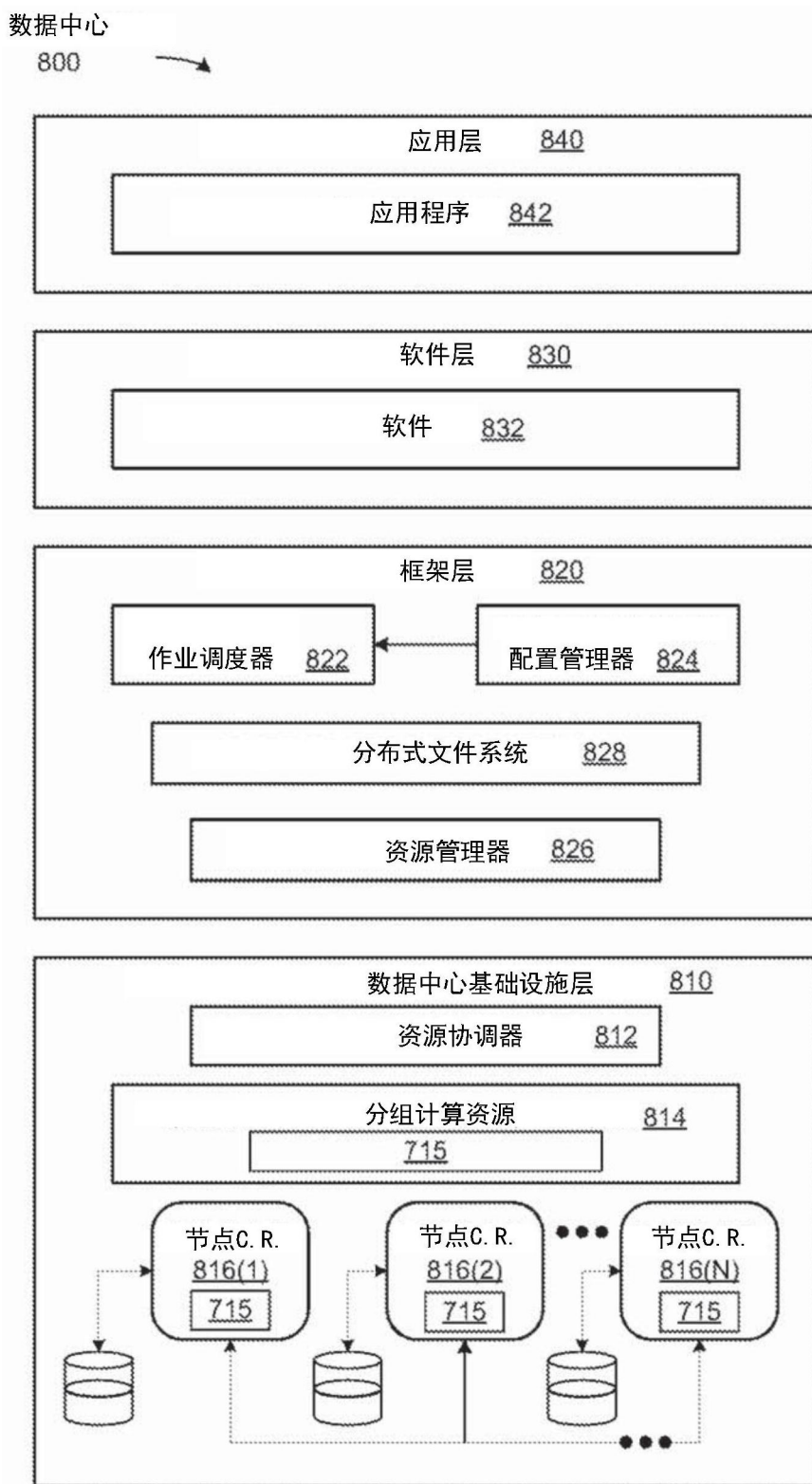


图8

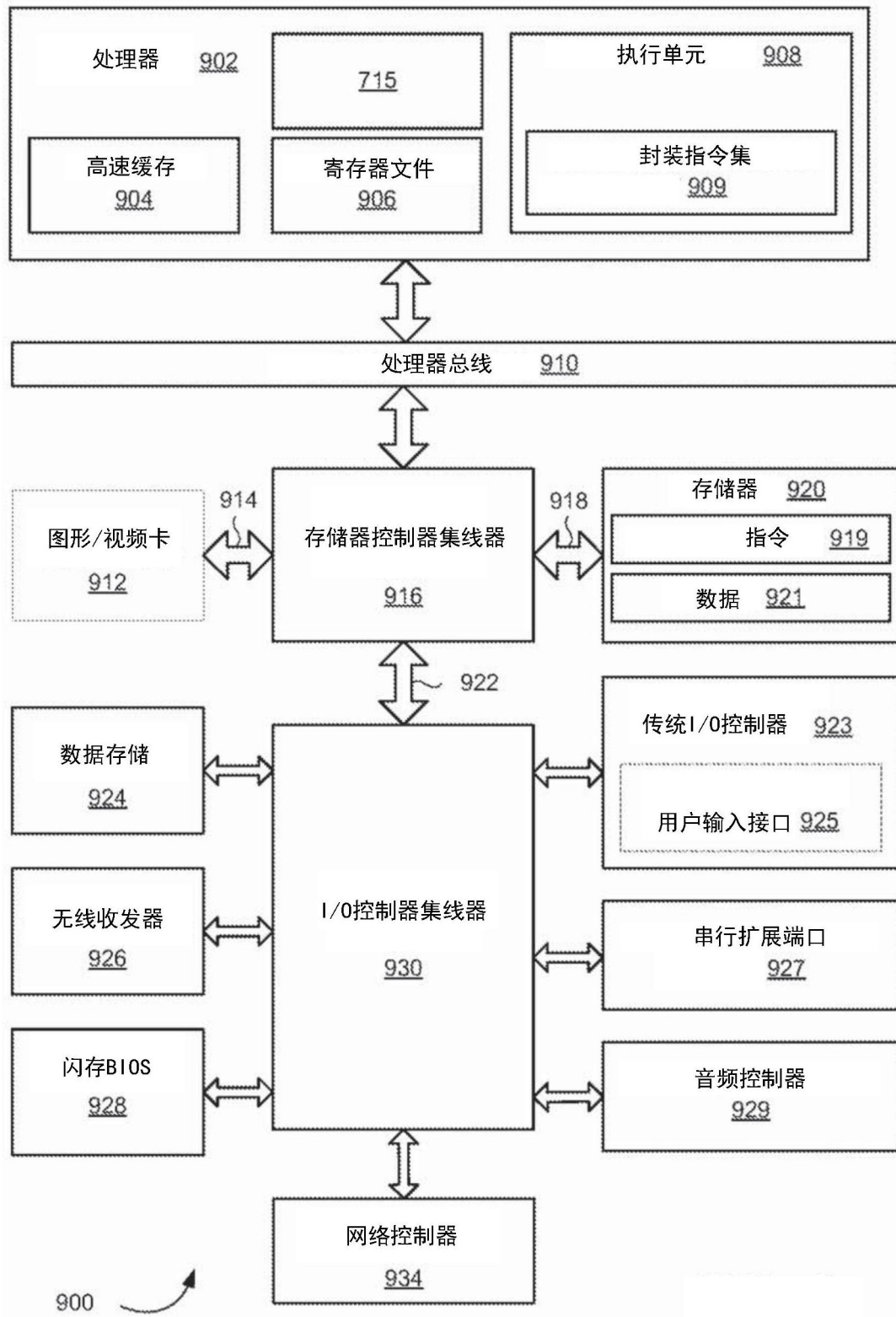


图9

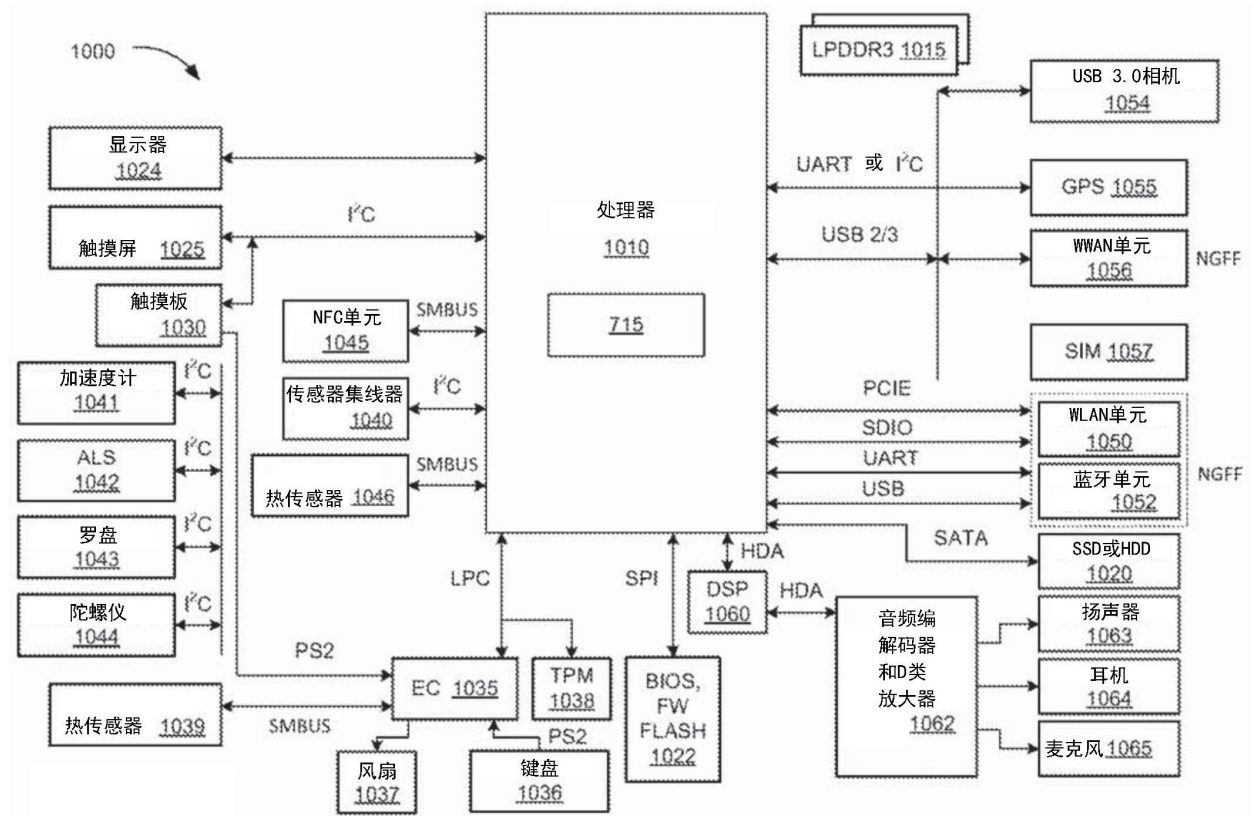


图10

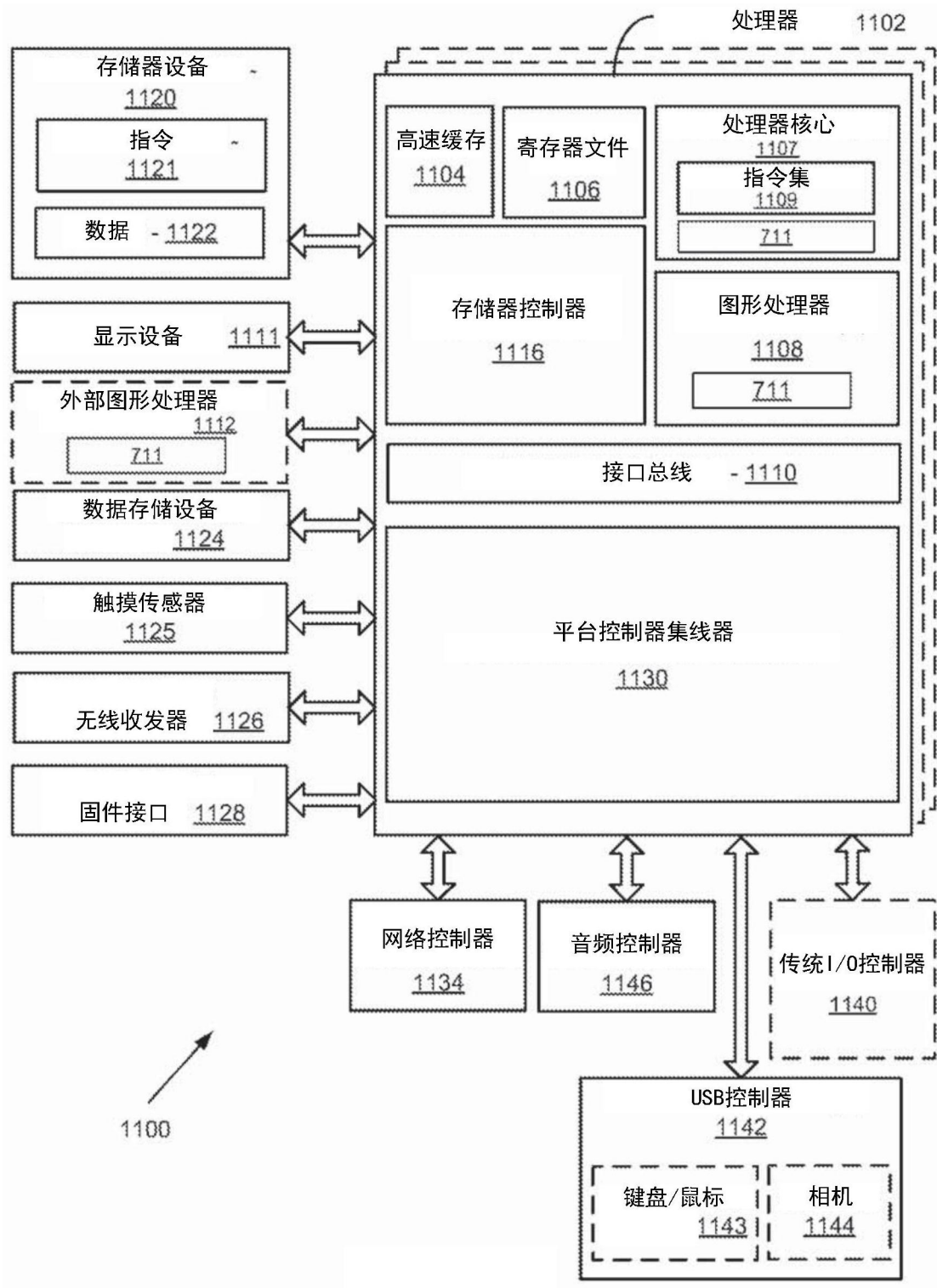


图11

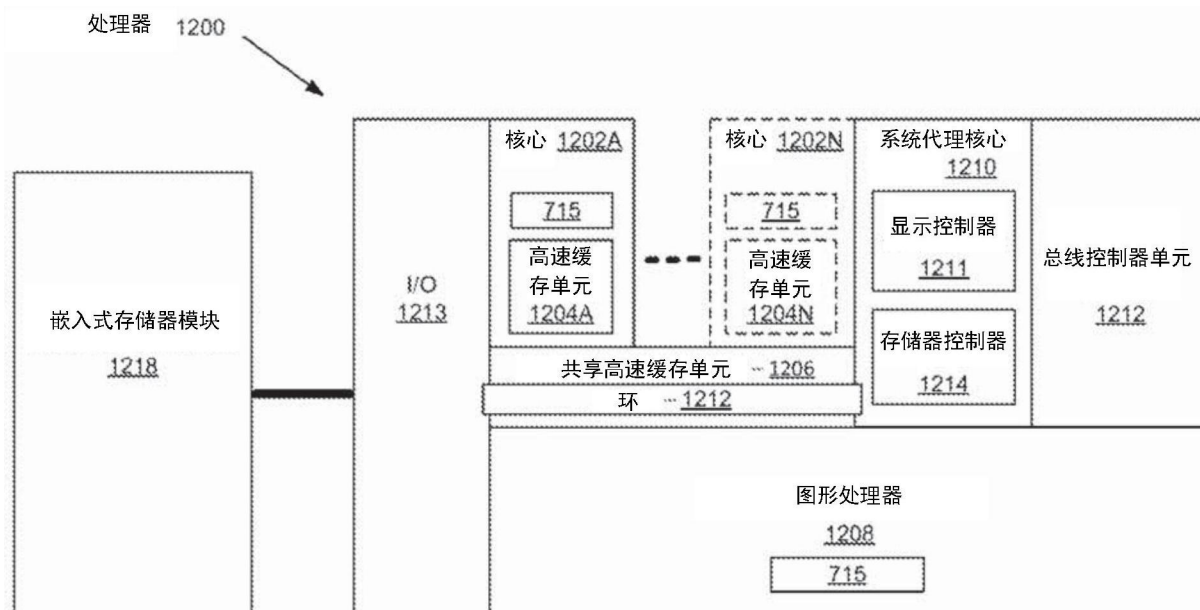


图12

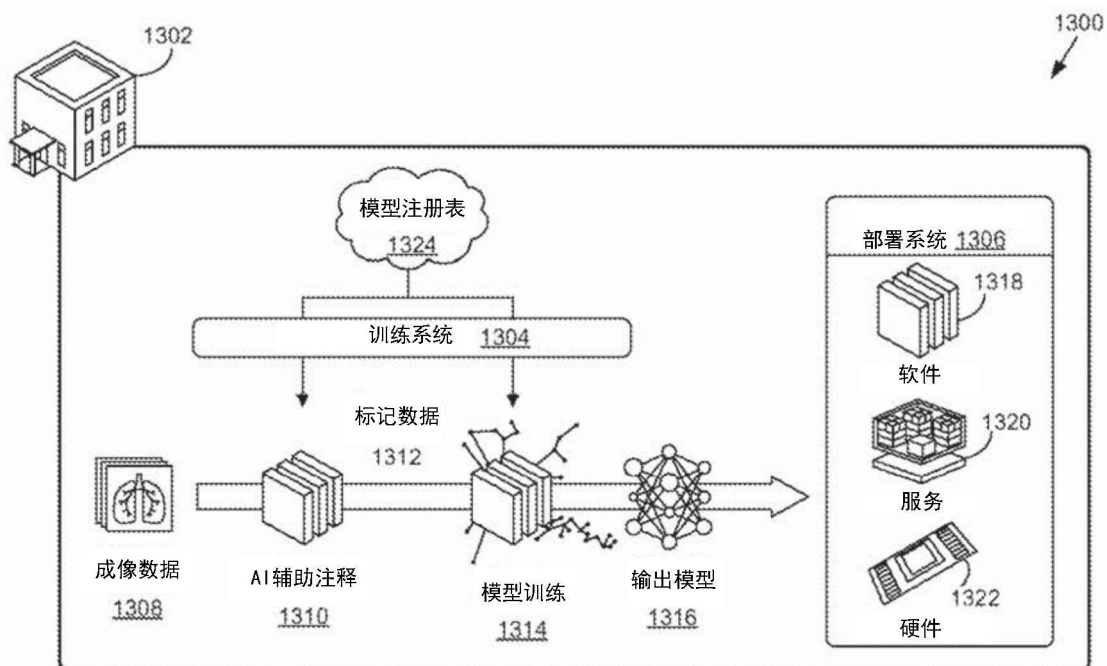


图13

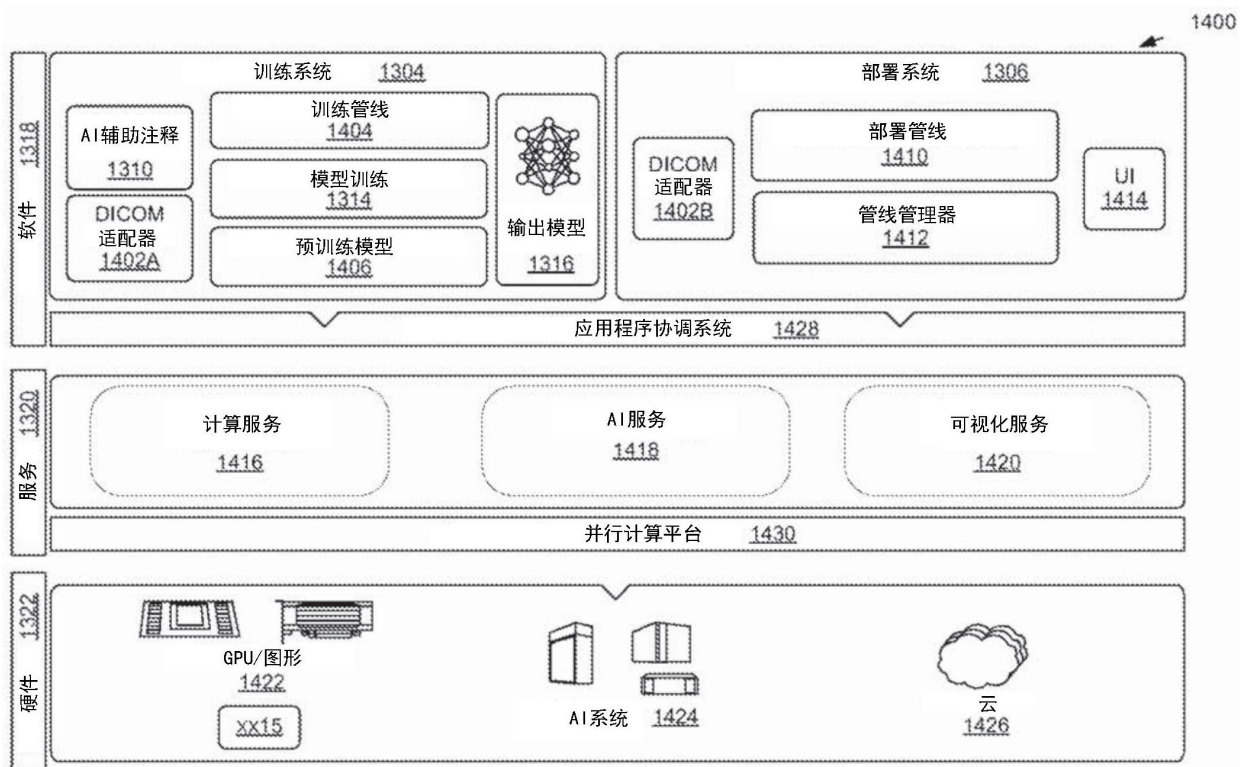


图14

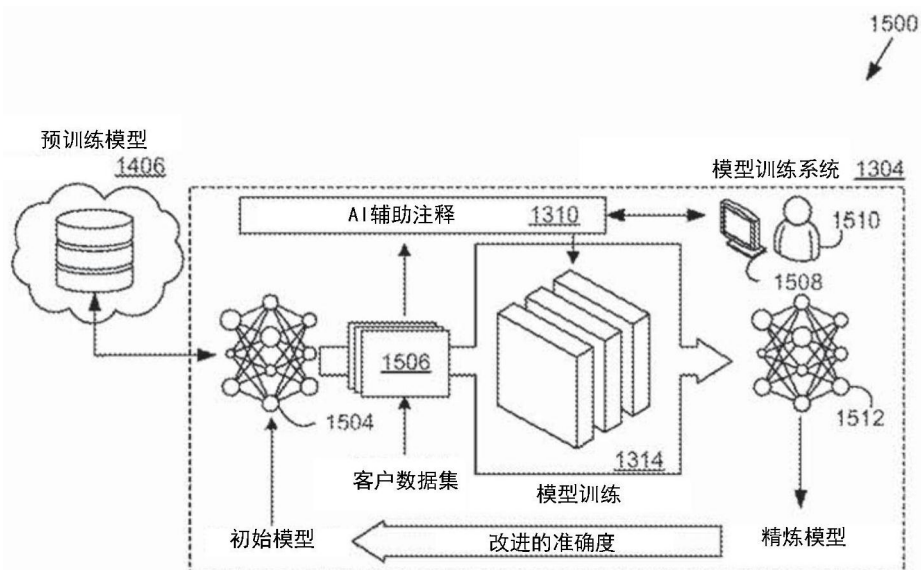


图15A

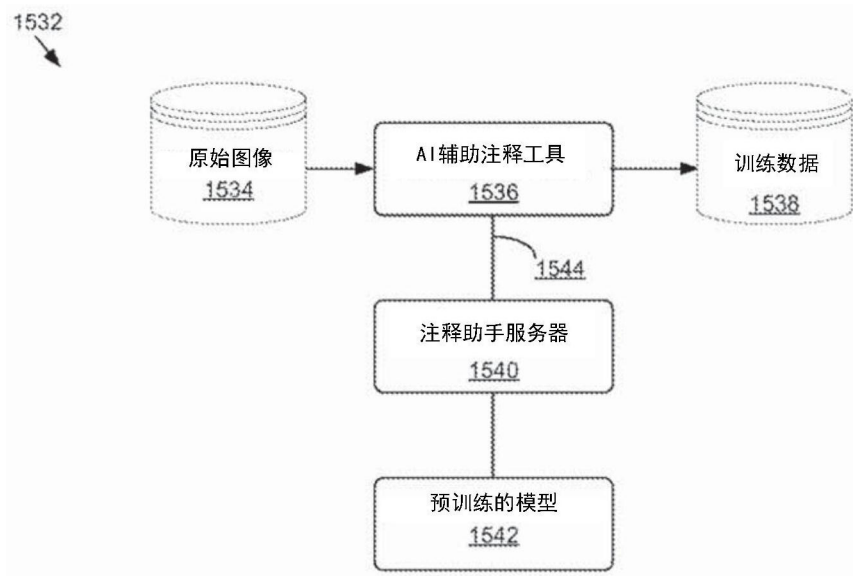


图15B