(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 113034698 B
(45)授权公告日 2025.05.06

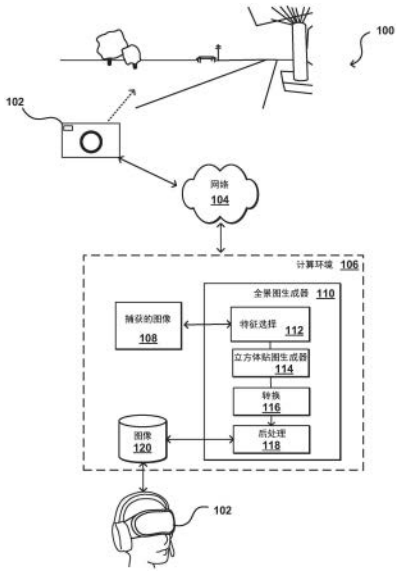
(21)申请号 202011538415.6
(22)申请日 2020.12.23
(65)同一申请的已公布的文献号
申请公布号 CN 113034698 A
(43)申请公布日 2021.06.25
(30)优先权数据
16/726,545 2019.12.24 US
(73)专利权人 辉达公司
地址 美国加利福尼亚州
(72)发明人 S·帕德什 P·P·科塔里 V·V·盖克瓦德
(74)专利代理机构 北京市磐华律师事务所 11336
专利代理师 汪洋
(51)Int.Cl.
G06T 19/00(2011.01)
G06T 11/00(2006.01)
G06N 3/0442(2023.01)
G06N 3/0475(2023.01)
G06N 3/045(2023.01)
G06N 3/0455(2023.01)
G06N 3/047(2023.01)
G06N 3/08(2023.01)
G06N 3/094(2023.01)
(56)对比文件
Shuran Song et al.Im2Pano3D: Extrapolating 360。Structure and Semantics Beyond the Field of View.《IEEE/CVF Conference on Computer Vision and Panttern Recognition》.2018,第3847-3856页.
审查员 朱佳彧
权利要求书3页 说明书79页 附图41页

(54)发明名称

使用一个或更多个神经网络生成全景图

(57)摘要

本申请涉及使用一个或更多个神经网络生成全景图。提出了用于从单个图像生成全景图的装置、系统和技术。在至少一个实施例中,一个或更多个生成神经网络用于使用从单个输入图像提取的特征来生成球形全景图像。



CN 113034698 B

1.一种处理器,包括:

一个或更多个电路,用于使用一个或更多个神经网络以至少部分地基于包括与在一个或更多个二维2D图像内的一个或更多个第一对象的一个或更多个特征相似的一个或更多个特征的一个或更多个第二对象产生包括所述一个或更多个第一对象的第一全景图像,其中所述一个或更多个第一对象在所述一个或更多个2D图像的扩展部分中。

2.根据权利要求1所述的处理器,其中,所述一个或更多个2D图像表示场景,并且其中,门控网络至少部分地基于经确定的场景类型来选择所述神经网络中的至少一个。

3.根据权利要求1所述的处理器,其中,所述一个或更多个神经网络包括用于至少部分地基于所述一个或更多个2D图像的代表性特征来生成外推图像内容的一个或更多个生成对抗网络(GAN)或变分自编码器(VAE)。

4.根据权利要求3所述的处理器,其中,所述一个或更多个2D图像和所述外推图像内容用于生成所述一个或更多个2D图像的立方体贴图表示。

5.根据权利要求4所述的处理器,其中,生成网络被用于将所述立方体贴图转换成全景图像。

6.根据权利要求5所述的处理器,其中,所述全景图像是球形全景图像,并且所述一个或更多个神经网络还用于执行球形全景图像的后处理,以使所述球形全景图像具有用于指定用途的格式。

7.一种系统,包括:

一个或更多个处理器,用于使用一个或更多个神经网络以至少部分地基于包括与在一个或更多个二维2D图像内的一个或更多个第一对象的一个或更多个特征相似的一个或更多个特征的一个或更多个第二对象产生包括所述一个或更多个第一对象的第一全景图像,其中所述一个或更多个第一对象在所述一个或更多个2D图像的扩展部分中。

8.根据权利要求7所述的系统,其中,所述一个或更多个2D图像表示场景,并且其中,门控网络至少部分地基于经确定的场景类型来选择所述神经网络中的至少一个。

9.根据权利要求7所述的系统,其中,所述一个或更多个神经网络包括用于至少部分地基于所述一个或更多个2D图像的代表性特征来生成外推图像内容的一个或更多个生成对抗网络(GAN)或变分自编码器(VAE)。

10.根据权利要求9所述的系统,其中,所述一个或更多个2D图像和所述外推图像内容用于生成所述一个或更多个2D图像的立方体贴图表示。

11.根据权利要求10所述的系统,其中,生成网络被用于将所述立方体贴图转换成全景图像。

12.根据权利要求11所述的系统,其中,所述全景图像是球形全景图像,并且所述一个或更多个处理器还用于执行球形全景图像的后处理,以使所述球形全景图像具有用于指定用途的格式。

13.一种方法,包括:

使用一个或更多个神经网络以至少部分地基于包括与在一个或更多个二维2D图像内的一个或更多个第一对象的一个或更多个特征相似的一个或更多个特征的一个或更多个第二对象产生包括所述一个或更多个第一对象的第一全景图像,其中所述一个或更多个第一对象在所述一个或更多个2D图像的扩展部分中。

14.根据权利要求13所述的方法,其中,所述一个或更多个2D图像表示场景,并且所述方法还包括:

由门控网络至少部分地基于经确定的场景类型来选择所述神经网络中的至少一个。

15.根据权利要求13所述的方法,其中,所述一个或更多个神经网络包括用于至少部分地基于所述一个或更多个2D图像的代表性特征来生成外推图像内容的一个或更多个生成对抗网络(GAN)或变分自编码器(VAE)。

16.根据权利要求15所述的方法,其中,所述一个或更多个2D图像和所述外推图像内容用于生成所述一个或更多个2D图像的立方体贴图表示。

17.根据权利要求16所述的方法,其中,生成网络被用于将所述立方体贴图转换成全景图像。

18.根据权利要求17所述的方法,其中,所述全景图像是球形全景图像,所述方法还包括:

执行球形全景图像的后处理,以使所述球形全景图像具有用于指定用途的格式。

19.一种非暂时性机器可读介质,在其上存储一组指令,所述一组指令如果由一个或更多个处理器执行,则使所述一个或更多个处理器至少:

使用一个或更多个神经网络以至少部分地基于包括与在一个或更多个二维2D图像内的一个或更多个第一对象的一个或更多个特征相似的一个或更多个特征的一个或更多个第二对象产生包括所述一个或更多个第一对象的第一全景图像,其中所述一个或更多个第一对象在所述一个或更多个2D图像的扩展部分中。

20.根据权利要求19所述的非暂时性机器可读介质,其中,所述一个或更多个2D图像表示场景,并且其中,如果所述指令被执行,则还使所述一个或更多个处理器:

由门控网络至少部分地基于经确定的场景类型来选择所述神经网络中的至少一个。

21.根据权利要求20所述的非暂时性机器可读介质,其中,所述一个或更多个神经网络包括用于至少部分地基于所述一个或更多个2D图像的代表性特征来生成外推图像内容的一个或更多个生成对抗网络(GAN)或变分自编码器(VAE)。

22.根据权利要求21所述的非暂时性机器可读介质,其中,所述一个或更多个2D图像和所述外推图像内容用于生成所述一个或更多个2D图像的立方体贴图表示。

23.根据权利要求22所述的非暂时性机器可读介质,其中,生成网络被用于将所述立方体贴图转换成全景图像。

24.根据权利要求23所述的非暂时性机器可读介质,其中,所述全景图像是球形全景图像,并且如果所述指令被执行,还使所述一个或更多个处理器:

执行球形全景图像的后处理,以使所述球形全景图像具有用于指定用途的格式。

25.一种全景图生成系统,包括:

一个或更多个处理器,用于使用一个或更多个神经网络以至少部分地基于包括与在一个或更多个二维2D图像内的一个或更多个第一对象的一个或更多个特征相似的一个或更多个特征的一个或更多个第二对象产生包括所述一个或更多个第一对象的第一全景图像,其中所述一个或更多个第一对象在所述一个或更多个2D图像的扩展部分中;以及

存储器,用于存储所述一个或更多个神经网络的网络参数。

26.根据权利要求25所述的全景图生成系统,其中,所述一个或更多个2D图像表示场

景,并且其中,门控网络至少部分地基于经确定的场景类型来选择所述神经网络中的至少一个。

27.根据权利要求26所述的全景图生成系统,其中,所述一个或更多个神经网络包括用于至少部分地基于所述一个或更多个2D图像的代表性特征来生成外推图像内容的一个或更多个生成对抗网络(GAN)或变分自编码器(VAE)。

28.根据权利要求27所述的全景图生成系统,其中,所述一个或更多个2D图像和所述外推图像内容用于生成所述一个或更多个2D图像的立方体贴图表示。

29.根据权利要求28所述的全景图生成系统,其中,生成网络被用于将所述立方体贴图转换成全景图像。

30.根据权利要求29所述的全景图生成系统,其中,所述全景图像是球形全景图像,并且所述一个或更多个处理器还用于执行球形全景图像的后处理,以使所述球形全景图像具有用于指定用途的格式。

使用一个或更多个神经网络生成全景图

技术领域

[0001] 至少一个实施例涉及用于执行和促进人工智能的处理资源。例如,至少一个实施例涉及用于根据本文描述的各种新颖技术来训练神经网络的处理器或计算系统。

背景技术

[0002] 诸如虚拟现实等的显示技术使用户能够获得场景的完整360度视图。不幸的是,由于许多用户相机仅捕获单个视点的二维图像,因此许多用户难以生成可用于此类显示技术的内容。类似地,不可能从先前捕获的单个图像返回并捕获全景图像。

附图说明

[0003] 将参考附图描述根据本公开的各种实施例,其中:

[0004] 图1示出了根据至少一个实施例的用于生成全景图像的系统;

[0005] 图2示出了根据至少一个实施例的全景图生成过程的图像;

[0006] 图3示出了根据至少一个实施例的球形全景图生成过程的图像;

[0007] 图4示出了根据至少一个实施例的用于生成全景图像的过程;

[0008] 图5示出了根据至少一个实施例的用于生成球形全景图像的过程;

[0009] 图6A示出了根据至少一个实施例的推理和/或训练逻辑;

[0010] 图6B示出了根据至少一个实施例的推理和/或训练逻辑;

[0011] 图7示出了根据至少一个实施例的示例数据中心系统;

[0012] 图8示出了根据至少一个实施例的计算机系统;

[0013] 图9示出了根据至少一个实施例的计算机系统;

[0014] 图10示出了根据至少一个实施例的计算机系统;

[0015] 图11示出了根据至少一个实施例的计算机系统;

[0016] 图12A示出了根据至少一个实施例的计算机系统;

[0017] 图12B示出了根据至少一个实施例的计算机系统;

[0018] 图12C示出了根据至少一个实施例的计算机系统;

[0019] 图12D示出了根据至少一个实施例的计算机系统;

[0020] 图12E和12F示出了根据至少一个实施例的共享编程模型;

[0021] 图13示出了根据至少一个实施例的示例性集成电路和相关的图形处理器;

[0022] 图14A-14B示出了根据至少一个实施例的示例性集成电路和相关的图形处理器;

[0023] 图15A-15B示出了根据至少一个实施例的另外的示例性图形处理器逻辑;

[0024] 图16示出了根据至少一个实施例的计算机系统;

[0025] 图17A示出了根据至少一个实施例的并行处理器;

[0026] 图17B示出了根据至少一个实施例的分区单元;

[0027] 图17C示出了根据至少一个实施例的处理集群;

[0028] 图17D示出了根据至少一个实施例的图形多处理器;

[0029] 图18示出了根据至少一个实施例的多图形处理单元(GPU)系统;

[0030] 图19示出了根据至少一个实施例的图形处理器;

[0031] 图20示出了根据至少一个实施例的处理器的微架构;

[0032] 图21示出了根据至少一个实施例的深度学习应用处理器;

[0033] 图22示出了根据至少一个实施例的示例性神经形态处理器;

[0034] 图23和图24示出了根据至少一个实施例的图形处理器的至少一部分;

[0035] 图25示出了根据至少一个实施例的图形处理器核心的至少一部分;

[0036] 图26A-26B示出了根据至少一个实施例的图形处理器核心的至少一部分;

[0037] 图27示出了根据至少一个实施例的并行处理单元(“PPU”);

[0038] 图28示出了根据至少一个实施例的通用处理集群(“GPC”);

[0039] 图29示出了根据至少一个实施例的并行处理单元(“PPU”)的存储器分区单元;

[0040] 图30示出了根据至少一个实施例的流式多处理器;

[0041] 图31是根据至少一个实施例的用于高级计算管线的示例数据流程图;

[0042] 图32是根据至少一个实施例的用于在高级计算管线中训练,调整,实例化和部署机器学习模型的示例系统的系统图;

[0043] 图33包括根据至少一个实施例的用于处理成像数据的高级计算管线3210A的示例说明;

[0044] 图34A包括根据至少一个实施例的支持超声设备的虚拟仪器的示例数据流程图;

[0045] 图34B包括根据至少一个实施例的支持CT扫描仪的虚拟仪器的示例数据流程图;

[0046] 图35A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程图;以及

[0047] 图35B是根据至少一个实施例的利用预训练的注释模型来增强注释工具的客户端-服务器架构的示例说明。

具体实施方式

[0048] 在至少一个实施例中,位置或场景的单个图像可以用于生成全景图像。在至少一个实施例中,可以使用两个或更多个图像来生成全景图,只要存在至少一些重叠量或使这些图像的特征相关联的方式即可。在至少一个实施例中,输入图像可以是任何尺寸或分辨率的二维(2D)或三维(3D)图像。此外,该图像可以表示要为其生成全景图像的场景的任何选定部分或百分比。在至少一个实施例中,图1示出了用于生成全景图像数据的系统100。在至少一个实施例中,相机102可以捕获与特定位置,场景或区域相关联的一个或更多个图像。在至少一个实施例中,相机可以是能够捕获位置的图像或视频数据的任何设备,其中图像可以包括视频数据的帧。在至少一个实施例中,尽管相机可能已经存在或与计算环境连接,例如智能电话中的相机,但是捕获的图像可以被传送到计算环境106。在至少一个实施例中,图像数据可以通过至少一个网络104(例如,蜂窝网络或因特网)从相机102发送。在至少一个实施例中,用户可以通过硬连线,无线,对等或局域网连接将相机102连接到计算环境106,例如笔记本或台式计算机。在至少一个实施例中,计算环境106可以是由单独方提供的云或资源环境,例如云提供商或全景图生成服务。

[0049] 在至少一个实施例中,由相机102捕获的图像108可用于生成全景图像,例如对于

虚拟和增强现实应用程序有用的二维全景图或等矩形全景图像。在至少一个实施例中,无监督和半无监督机器学习(可以包括使用生成对抗网络(GAN)或变分自编码器(VAE))可以用于获取标准数字图像并生成全景图像,例如全景360-度单图像。在至少一个实施例中,还可以向系统100生成的图像注入适用的元数据,以使这些图像与各种全景图像兼容共享和消费应用程序或平台,例如可从诸如Google和Facebook之类的公司获得。

[0050] 在至少一个实施例中,想要生成用于位置或场景的全景图像的用户可以捕获图像。在至少一个实施例中,也可以捕获附加图像以供使用。在至少一个实施例中,可以将输入图像输入到全景生成器110,该全景生成器可以以硬件和/或软件来提供。在至少一个实施例中,全景生成器110可以对该图像进行一些预处理,以改善清晰度,纵横比,颜色平衡,颜色深度或其他这样的图像参数或纵横比。在至少一个实施例中,该图像将由特征选择模块112分析,以便识别和选择输入图像108的代表性特征。

[0051] 在至少一个实施例中,全景生成器110可以接受单个图像作为输入,例如宽度和高度为至少256个像素的RGB图像。在至少一个实施例中,输入图像可以尽可能代表给定场景,以便为该场景提供最准确的全景图生成。在至少一个实施例中,使用各种尺寸的输入图像的大量集合来训练特征选择模块,该特征选择模块可以包括生成对抗网络(GAN)。在至少一个实施例中,训练数据可以包括至少一些标签。在至少一个实施例中,该训练数据可以包括成对的具有对应表示的单个图像(或图像集),例如立方体贴图全景表示,以用于训练该生成模型。在至少一个实施例中,可以使用对应的等角矩形表示来训练转换模型。在至少一个实施例中,可以以不同的方式裁剪立方体贴图或全景图以生成用于训练的单个图像。

[0052] 在至少一个实施例中,特征选择步骤,过程或模块112用于学习每个方向上的输入图像的特征表示,并选择适当的模型以生成其等效全景图。在至少一个实施例中,模型可以尝试基于重建概率来确定场景的特征。在至少一个实施例中,这可以使用具有前馈选通或决策网络的专家混合(MoE)模型来实现,该模型基于该网络检测到的场景特征来分配专家。在至少一个实施例中,这些专家可以是负责重建该输入图像的变分自编码器(VAE)。在至少一个实施例中,基于最低的重建概率来分配最后一组专家。在至少一个实施例中,这样的分配有助于根据场景级特征来分配专家。在至少一个实施例中,第一专家模型可能专门研究外景,例如可能涉及沙漠或草山,而第二专家模型可能专门研究室内空间,例如客厅或餐厅内部。

[0053] 在至少一个实施例中,生成全景包括基于输入图像的特征生成附加图像内容。在至少一个实施例中,可能存在针对不同类型的场景或环境训练的多个生成模型,并且可能期望针对这些场景或环境训练特定模型,以便不仅提高准确性或生成或外推(extrapolated)的特征,而且减少可能会在生成的图像部分中出现的噪声量。在至少一个实施例中,具有特定训练的模型可以改善模型能够添加到场景中的特征的数量,质量和类型数量。在至少一个实施例中,可以训练模型以针对具有特定类型的特征的特定类型或场景组(例如,域)生成特征,除了避免所生成的图像中的过多噪声,对这些类型设置很少的限制。在至少一个实施例中,门控网络可以分析输入图像并确定应该使用哪个或哪些专家模型来处理该图像并生成全景图所需的图像内容。在至少一个实施例中,门控网络可以随着时间学习哪些专家模型专门研究具有特定类型特征的场景。

[0054] 在至少一个实施例中,特征选择过程可以分析输入图像以提取要渲染的场景的代

表性特征,并且消除或最小化这些图像中存在的噪声。在至少一个实施例中,使用无监督机器学习技术来执行降维(dimensionality reduction),以促进更快的结果和更少的资源需求。在至少一个实施例中,降维还可以用作压缩的度量,以存储具有大量特征的用例的表示。在至少一个实施例中,降维被用作特征选择的量度以从样品中识别和选择相关特征。在至少一个实施例中,自动编码器可以用于实现足够的降维。在至少一个实施例中,从降维处理输出的原始特征可以用作生成算法的输入。

[0055] 在至少一个实施例中,第二步骤,过程或模块114涉及用于中间表示的代表图像的生成,例如立方体贴图的每个面的六个图像。在至少一个实施例中,可以将360度场景解释为六个垂直视图的组合,包括前视图,后视图,顶视图,底视图,左视图和右视图。在至少一个实施例中,该生成模型可用于为输入特征分配方向,并在每个方向上生成用于中间表示(例如立方体贴图表示)的图像。在至少一个实施例中,立方体贴图结构中的每个位置均保持任何相关观看标准,例如360度观看,可能包括诸如焦点和角度之类的标准。在至少一个实施例中,垂直位置相互依赖,如生成模型所强制的。在至少一个实施例中,GAN可以用作图像生成的生成模型。在至少一个实施例中,可以使用变分自编码器(VAE)作为生成模型来组合这些选择和生成阶段。在至少一个实施例中,该生成模型可以根据各种因素中的任何一种来固有地调整生成的图像。在至少一个实施例中,垂直视场可能是该调整的一个因素,例如可以在从前向位置向下45度的角度处捕获给定图像。在至少一个实施例中,生成模型将识别该角度并将最终图像调整为主要面向前方。在至少一个实施例中,缺少上下文信息可以是一个因素,例如通常缺少一个方向上的内容。在至少一个实施例中,这可以对应于常规图像在“后”方向上缺失上下文,由此模型可以外推(extrapolate)该输入图像的特征以有效地环绕此立方体贴图,从而通过例如扩展,复制并生成新的兼容特征来填补两者之间的空白以完成该场景。在至少一个实施例中,图像生成过程涉及创建多个图像,其中变化可能与所涉及的外推程度有关。在至少一个实施例中,这可以通过将生成模型配置为假设图像属于六个垂直方向外的一个垂直方向,并生成其余五个图像来实现,总计每个输入图像至少六个全景图像。在至少一个实施例中,通过省略例如顶视图,后视图和底视图,这种方法可扩展到180度图像。

[0056] 在至少一个实施例中,诸如可以由立方体贴图生成器114使用的生成算法用于生成每个正交方向或可以映射到立方体贴图的六个总方向的代表图像。在至少一个实施例中,该生成器可以是具有约束的条件GAN。在至少一个实施例中,立方体贴图用作环境或场景的表示,其中该场景的部分投影到立方体的侧面并存储为六个正方形纹理,可用于在任何方向上投影该场景的图像。在至少一个实施例中,可以使用生成模型来将方向(从确定的视点或原点)分配给从输入图像108确定的特征,并生成用于立方体贴图表示的侧面的图像。在至少一个实施例中,可以生成除立方体贴图以外的表示,或者模型可以使用这些特征来直接生成球形全景图像,如本文其他地方所讨论的。

[0057] 在至少一个实施例中,立方体贴图结构中的每个位置将满足用于完整360度视图的标准,其中那些标准可以与诸如焦点和角度之类的因素有关。在至少一个实施例中,垂直位置彼此依赖,这可以由立方体贴图生成器114的生成模型来强制。在至少一个实施例中,可以使用变分自编码器(VAE)来组合选择和生成阶段以直接从输入图像108产生立方体贴图。在至少一个实施例中,可以利用矢量量化的VAE(VQ-VAE)。在至少一个实施例中,生成对

抗网络(GAN)可以用于图像生成。在至少一个实施例中,从图像特征生成立方体贴图使得能够通过该生成过程来填补空白。在至少一个实施例中,系统100从从输入图像108提取的核心特征的集合中学习场景的表示。在至少一个实施例中,神经网络可以推断该场景的特征并将那些特征编码到可用于在360度范围内重建整个场景的潜空间中。

[0058] 在至少一个实施例中,条件GAN可以假定输入图像对应于要生成的立方体贴图的正面。在至少一个实施例中,该GAN然后可以使用该正面作为参考来生成相关的垂直面部图像。在至少一个实施例中,鉴别器可以分析生成的图像以尝试确定该图像是真实的还是生成的,并且将真实的确定视为有效的生成图像。在至少一个实施例中,可以使用重叠的可变百分比。在至少一个实施例中,模型可以假定对于给定的输入图像,在顶面,正面和右侧之间存在重叠。在至少一个实施例中,该模型然后可以在其他方向上外推。在至少一个实施例中,模型可以采用三个重叠方向的集合,并据此外推其余方向。在至少一个实施例中,这种方法将不会生成单个全景图而是多个可能的全景图。在至少一个实施例中,生成多个全景图可以帮助处理可以以多种方式解释的特征,例如瓷砖可能被视为浴室内部或游泳池的一部分,这将导致非常不同的全景图。在至少一个实施例中,模型可以以不同的方式解释这些特征,并对每个特征进行推断,然后由用户或过程确定或选择合适的全景图。在至少一个实施例中,生成模型可以基于百分比或基于图像大小来工作,并且也可以是可调的。在至少一个实施例中,网络可以确定该输入图像将占最终全景图的百分比。在至少一个实施例中,网络可以假设输入图像将表示最终立方体贴图图像的一个框,并且可以相应地设置立方体贴图的整体尺寸。然而,在至少一个实施例中,网络还可以保留定制单个输入图像所代表的立方体贴图面的百分比的能力。在至少一个实施例中,用户或应用程序可以指示图像的立方体面的百分比,并且可以指定该立方体面的边界内的位置。在至少一个实施例中,GAN可以将输入图像视为输入约束(例如,地面实况数据),以用于推断特征以填充全景图中的其余区域。

[0059] 在至少一个实施例中,下一步骤,过程或模块116涉及从立方体贴图或其他中间表示转换为等矩形全景表示。在至少一个实施例中,可以将经训练的GAN用于该转换,以便使该图像中的真实感最大化并且使失真最小化。在至少一个实施例中,GAN的使用还可以允许例如在大约1度到大约180度的范围内的可变和可配置的垂直视场(FOV),这可以至少部分地取决于该GAN的训练。这在至少一个实施例中,可以以这种方式从立方体贴图生成圆柱形全景表示,因为使用GAN可以实现训练控制的实现灵活性。在至少一个实施例中,可以使用专用模型来支持每种用例,例如可以将生成模型应用于120度球形全景转换,用于180度转换的单独模型以及用于圆柱全景表示的单独模型。

[0060] 在至少一个实施例中,过程或模块118可以使用后处理步骤,将元数据注入到生成的全景图(例如360度图像)中,以符合图像处理,共享和消费平台或可能需要与某些标准或准则兼容的其他应用程序。在至少一个实施例中,这可以包括遵守Adobe的XMP标准以呈现图像,并且可以在此步骤将必需的元数据注入该图像中。在至少一个实施例中,这里也可以解决任何其他后处理措施。在至少一个实施例中,可以将生成的图像存储到本地图像存储库120中以用于后续检索。在至少一个实施例中,可以检索该图像(直接或通过单独的应用程序,服务或设备)以在适当的显示机制(例如虚拟现实(VR)头戴式耳机102)上呈现。然后,在至少一个实施例中,用户可以移动他或她的头部以获得该空间的不同VR视图,其中由相

机102捕获这些原始图像。

[0061] 在至少一个实施例中,训练图像集可以用于训练和测试。在至少一个实施例中,来自两个域的图像被输入到该系统,使得可以在诸如ImageNet的大数据集上对GAN模型进行预训练。在至少一个实施例中,可以为生成阶段提供常规图像和对应的立方体贴图表示。在至少一个实施例中,可以为转换阶段提供相关的立方体贴图表示和具有指定视场的相应全景表示。在至少一个实施例中,测试阶段仅需要规则图像作为输入。在至少一个实施例中,这样的系统可以被部署为托管的Web服务,或者被部署为处理图像的VR解决方案的一部分。在至少一个实施例中,这样的系统还可以是视频游戏系统的一部分,例如来自NVIDIA Corporation的GeForce Now,以协助执行诸如处理游戏内屏幕截图的任务。

[0062] 在至少一个实施例中,可以提供输入图像202,其包括场景的视图,这里是具有树木,草地和建筑物的风景,如图2的视图200所示。在至少一个实施例中,可以选择这些特征,并使用适当的生成模型,用于在该原始图像的区域206之外生成相似类型的特征,以生成包括原始输入图像202中未包含或未表示的新内容的一个或更多个区域208,210的全景图像204。在至少一个实施例中,用户可以指定全景图的类型以及其他方面,例如大小,分辨率和类型。在至少一个实施例中,用户还可以指定输入图像在全景图中的放置。例如,用户可能具有指定是将输入图像水平居中还是垂直居中的能力,以及沿该图像内容应占据的任一维度的百分比。

[0063] 在至少一个实施例中,这种方法可以用于生成如图3的视图300所示的球形全景图。在至少一个实施例中,单个输入图像302可以被接收,并且用于生成用于立方体贴图304的图像。在至少一个实施例中,输入图像302的图像内容可以表示立方体贴图304的一部分,例如可以是示出的正面的一部分。在至少一个实施例中,该图像内容在立方体贴图304中的放置和大小可以手动或自动执行。在至少一个实施例中,该图像内容还可以是一个以上立方体贴图面的一部分。在至少一个实施例中,该立方体贴图304可以包括来自填充了该立方体贴图304的侧面的输入图像302的图像内容的区域之外的内容。在至少一个实施例中,该立方体贴图然后可以被转换成球形表示306,其可以通过适当的查看器或演示机制进行查看。

[0064] 在至少一个实施例中,可以生成图像以具有根据各种标准的等矩形360度图像的特征,因为在所有方向上可以具有2:1的纵横比和均匀的焦点。在至少一个实施例中,这样的图像还可以适当地具有纬度和/或经度失真。在至少一个实施例中,可以接收多个图像作为输入,并且系统100应该能够生成全景图,只要它能够在所有方向上关联和外推特征即可。在至少一个实施例中,这样的生成过程可以是生成性的并且是自调整的,能够适应失真和聚焦角变化。在至少一个实施例中,可以在生成立方体贴图之前对输入图像进行归一化,以允许输入图像中的角度和方向变化,以及允许多个输入图像之间没有重叠。在至少一个实施例中,系统100所使用的模型本质上是生成性的,从而使它们能够对该场景进行假设并生成缺少图像数据的任何区域或方向的内容。

[0065] 在至少一个实施例中,如图4所示,可以利用用于生成全景图像的过程400。在至少一个实施例中,可以接收402包括位置,场景或环境的视图的图像。在至少一个实施例中,可以分析该图像以确定用于处理特定类型的场景的模型。在至少一个实施例中,部分地执行使用可以确定图像的代表性特征特征提取模型(或本文提出的其他机制)并选择适合那些

类型的特征的模型。在至少一个实施例中,该模型可以确定404该输入图像在中间表示(诸如立方体贴图)中的放置。在至少一个实施例中,确定的图像特征可以与相同或不同的生成模型一起使用以生成立方体贴图或其他这样的表示。在至少一个实施例中,生成模型使用确定的特征来外推406图像数据以生成用于填充立方体贴图的剩余部分的图像内容。在至少一个实施例中,可以使用转换模型(或本文提出的其他机制)来处理该立方体贴图,以执行408等矩形转换以生成球形全景图像。在至少一个实施例中,可以执行410至少一些量的后处理,以将该全景图像放置为能够在目标应用程序或设备中显示的格式。在至少一个实施例中,可以由经训练的模型替代地执行任何格式处理,作为全景图生成过程的一部分。在至少一个实施例中,可以在没有立方体贴图或类似中间表示的单个模型中执行特征提取和全景生成。

[0066] 在至少一个实施例中,如图5所示,可以利用用于生成全景图像的过程500。在至少一个实施例中,可以接收502场景的图像。在至少一个实施例中,分析该图像以识别每个图像中的代表性特征,例如表示那些图像中的对象或内容的唯一或核心特征。在至少一个实施例中,生成模型使用这些特征来生成504从那些特征外推出的附加图像内容。在至少一个实施例中,可以至少部分地基于这些代表性特征和附加图像内容来生成506全景图像。在至少一个实施例中,可以通过与执行全景图像生成相同或不同的生成模型来生成中间表示(诸如立方体贴图)。在至少一个实施例中,可以执行一些后处理以使该全景图与目标应用程序,设备或格式兼容。

[0067] 推理和训练逻辑

[0068] 图6A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615。下面结合图6A和/或6B提供关于推理和/或训练逻辑615的细节。

[0069] 在至少一个实施例中,推理和/或训练逻辑615可以包括但不限于代码和/或数据存储器601,以存储前向和/或输出权重和/或输入/输出数据和/或其他参数以在一个或更多个实施例的方面中配置用于训练和/或用于推理的神经网络的神经元或层。在至少一个实施例中,训练逻辑615可以包括或耦合到代码和/或数据存储器601以存储图形代码或其他软件以控制时序和/或顺序,在其中加载权重和/或其他参数信息以配置包括整数和/或浮点单元(统称为算术逻辑单元(ALU))的逻辑。在至少一个实施例中,代码(诸如图形代码)基于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/或数据存储器601存储在使用一个或更多个实施例的方面的训练和/或推理期间的输入/输出数据和/或权重参数的正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一层的权重参数和/或输入/输出数据。在至少一个实施例中,代码和/或数据存储器601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。

[0070] 在至少一个实施例中,代码和/或数据存储器601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/或代码和/或数据存储601可以是高速缓冲存储器、动态随机可寻址存储器(“DRAM”)、静态随机可寻址存储器(“SRAM”)、非易失性存储器(例如闪存)或其他存储器。在至少一个实施例中,对代码和/或编码和/或数据存储601是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,可以取决于存储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批量大小或这些因素的某种组合。

[0071] 在至少一个实施例中,推理和/或训练逻辑615可以包括但不限于代码和/或数据存储器605,以存储与神经元或层相对应的向后和/或输出权重和/或输入/输出数据,对应于一个或更多个实施例的方面中训练和/或用于推理的神经网络的神经元或层。在至少一个实施例中,代码和/或数据存储器605存储使用一个或更多个实施例的各方面在训练和/或推理期间的输入/输出数据和/或权重参数在向后传播期间与一个或更多个实施例一起训练或使用的神经网络的每一层的权重参数和/或输入/输出数据。在至少一个实施例中,训练逻辑615可以包括或耦合到代码和/或数据存储器605以存储图形代码或其他软件以控制时序和/或顺序,在其中加载权重和/或其他参数信息将在以配置包括整数和/或浮点单元(统称为算术逻辑单元(ALU))的逻辑、。在至少一个实施例中,代码(诸如图形代码)基于该代码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/或数据存储器605的任何部分可以与其他片上或片外数据存储器一起包括,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/或数据存储器605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,数据存储605可以是高速缓存存储器、DRAM、SRAM、非易失性存储器(例如闪存)或其他存储器。在至少一个实施例中,代码和/或数据存储605是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。

[0072] 在至少一个实施例中,代码和/或数据存储器601以及代码和/或数据存储器605可以是分开的存储结构。在至少一个实施例中,代码和/或数据存储器601以及代码和/或数据存储器605可以是相同的存储结构。在至少一个实施例中,代码和/或数据存储器601以及代码和/或数据存储器605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/或数据存储器601和代码和/或数据存储器605的任何部分可以与其他片上或片外数据存储器包括在一起,包括处理器的L1、L2或L3高速缓存或系统存储器。

[0073] 在至少一个实施例中,推理和/或训练逻辑615可以包括但不限于一个或更多个算术逻辑单元(“ALU”)610,包括整数和/或浮点单元,以至少部分地基于训练和/或推理代码(例如,图形代码)或由其指示来执行逻辑和/或数学运算,其结果可能会产生(例如,来自神经网络内部的层或神经元的输出值)存储在激活存储器620中的激活,其是存储在代码和/或数据存储器601和/或代码和/或数据存储器605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ALU 610执行的线性代数和/或基于矩阵的数学生成在激活存储器620中存储的激活,其中存储在代码和/或数据存储605中和/或代码和/或将数据存储601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605或代码和/或数据存储601其他片上或片外存储器中。

[0074] 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中包括一个或更多个ALU 610,而在另一实施例中,一个或更多个ALU 610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 610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/或数据存储器601,代码和/或数据存储器605以及激活存储器620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,激活存储器620的任何部分可以与其他片上或片外数据存储器包括在一起,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。

[0075] 在至少一个实施例中,激活存储器620可以是高速缓冲存储器、DRAM、SRAM、非易失性存储器(例如,闪存)或其他存储器。在至少一个实施例中,激活存储器620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器,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,选择激活存储器620是处理器的内部还是外部,例如,或者包含DRAM、SRAM、闪存或其他存储器类型,。在至少一个实施例中,图6A中所示的推理和/或训练逻辑615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Intel Corp的Nervana®(例如“Lake Crest”)处理器。在至少一个实施例中,图6A所示的推理和/或训练逻辑615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。

[0076] 图6B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615。在至少一个实施例中,推理和/或训练逻辑615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6B中所示的推理和/或训练逻辑615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Intel Corp的Nervana®(例如“Lake Crest”)处理器。在至少一个实施例中,图6B中所示的推理和/或训练逻辑615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615包括但不限于代码和/或数据存储器601以及代码和/或数据存储器605,其可以用于存储代码(例如,图形代码)、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6B中所示的至少一个实施例中,代码和/或数据存储器601以及代码和/或数据存储器605中的每一个都分别与专用的计算资源(例如计算硬件602和计算硬件606)相关联,。在至少一个实施例中,计算硬件602和计算硬件606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/或数据存储器601和代码和/或数据存储器605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器620中。

[0077] 在至少一个实施例中,代码和/或数据存储器601和605以及相应的计算硬件602和

606中的每一个分别对应于神经网络的不同层,使得从代码和/或数据存储器601和计算硬件602的一个“存储/计算对601/602”得到的激活提供作为代码和/或数据存储605和计算硬件606的“存储/计算对605/606”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601/602和605/606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615中可以包括在存储计算对601/602和605/606之后或与之并行的附加存储/计算对(未示出)。

[0078] 数据中心

[0079] 图7示出了示例数据中心700,其中可以使用至少一个实施例。在至少一个实施例中,数据中心700包括数据中心基础设施层710、框架层720、软件层730和应用层740。

[0080] 在至少一个实施例中,如图7所示,数据中心基础设施层710可以包括资源协调器712、分组的计算资源714和节点计算资源(“节点C.R.s”)716(1)-716(N),其中“N”代表任何完整的正整数。在至少一个实施例中,节点CR 716(1)-716(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),内存设备(例如动态只读存储器),存储器设备(例如固态硬盘或磁盘驱动器),网络输入/输出(“NW I/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.716(1)-716(N)中的一个或更多个节点C.R.可以是具有一个或更多个上述计算资源的服务器。

[0081] 在至少一个实施例中,分组的计算资源714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714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、内存或存储器资源。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。

[0082] 在至少一个实施例中,资源协调器712可以配置或以其他方式控制一个或更多个节点C.R.716(1)-716(N)和/或分组的计算资源714。在至少一个实施例中,资源协调器712可以包括用于数据中心700的软件设计基础结构(“SDI”)管理实体。在至少一个实施例中,资源协调器可以包括硬件、软件或其某种组合。

[0083] 在至少一个实施例中,如图7所示,框架层720包括作业调度器722、配置管理器724、资源管理器726和分布式文件系统728。在至少一个实施例中,框架层720可以包括支持软件层730的软件732和/或应用程序层740的一个或更多个应用程序742的框架。在至少一个实施例中,软件732或应用程序742可以分别包括基于Web的服务软件或应用程序,例如由Amazon Web Services,Google Cloud和Microsoft Azure提供的服务或应用程序。在至少一个实施例中,框架层720可以是但不限于一种免费和开放源软件网络应用框架,例如可以利用分布式文件系统728来进行大范围数据处理(例如“大数据”)的Apache Spark™(以下称为“Spark”)。在至少一个实施例中,作业调度器722可以包括Spark驱动器,以促进对数据中心700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724可以能够配置不同的层,例如软件层730和包括Spark和用于支持大规模数据处理的分布式文件系统728的框架层720。在至少一个实施例中,资源管理器726能够管理映射到或分配用于支持

分布式文件系统728和作业调度器722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710上的分组的计算资源714。在至少一个实施例中,资源管理器726可以与资源协调器712协调以管理这些映射的或分配的计算资源。

[0084] 在至少一个实施例中,包括在软件层730中的软件732可以包括由节点C.R.716(1)-716(N)的至少一部分,分组计算资源714和/或框架层720的分布式文件系统728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。

[0085] 在至少一个实施例中,应用层740中包括的应用程序742可以包括由节点C.R.716(1)-716(N)的至少一部分、分组的计算资源714和/或框架层720的分布式文件系统728使用的一种或更多种类型的应用程序。一种或更多种类型的应用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。

[0086] 在至少一个实施例中,配置管理器724、资源管理器726和资源协调器712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700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。

[0087] 在至少一个实施例中,数据中心700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700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700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。

[0088] 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。

[0089] 推理和/或训练逻辑615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。下面结合图6A和/或6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例中,推理和/或训练逻辑615可以在系统图7中使用,至少部分地基于使用神经网络训练操作、神经网络功能和/或架构或本文所述的神经网络用例计算出的权重参数来推理或预测操作。

[0090] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0091] 计算机系统

[0092] 图8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800,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800可以包括但不限于组件,例如处理器802,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800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(Intel Corporation of Santa Clara, California)获得的PENTIUM®处理器家族、Xeon™、Itanium®、XScale™和/或StrongARM™,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800可以执行可从华盛顿州雷蒙德市的微软公司(Microsoft Corporation of Redmond, Wash.)n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。

[0093] 实施例可以用在其他设备中,例如手持设备和嵌入式应用。手持设备的一些示例包括蜂窝电话、互联网协议(Internet Protocol)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。

[0094] 在至少一个实施例中,计算机系统800可包括但不限于处理器802,该处理器802可包括但不限于一个或更多个执行单元808,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800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800可以是多处理器系统。在至少一个实施例中,处理器802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,或任何其他处理器设备,例如数字信号处理器。在至少一个实施例中,处理器802可以耦合到处理器总线810,该处理器总线810可以在处理器802与计算机系统800中的其他组件之间传输数据信号。

[0095] 在至少一个实施例中,处理器802可以包括但不限于1级(“L1”)内部高速缓存存储器(“cache”)804。在至少一个实施例中,处理器802可以具有单个内部高速缓存或多级内部缓存。在至少一个实施例中,高速缓冲存储器可以驻留在处理器802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806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。

[0096] 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的执行单元808,其也位于处理器802中。在至少一个实施例中,处理器802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808可以包括用于处理封装指令集809的逻辑。在至少一个实施例中,通过将封装指令集809包括在通用处理器802的指令集中,以及要执行指令的相关电路,可以使用通用处理器802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。

[0097] 在至少一个实施例中,执行单元808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800可以包括但不限于存储器820。在至少一个实施例中,存储器820可以被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820可以存储由处理器802可以执行的由数据信号表示的指令819和/或数据821。

[0098] 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810和存储器820。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”)816,并且处理器802可以经由处理器总线810与MCH 816通信。在至少一个实施例中,MCH 816可以提供到存储器820的高带宽存储器路径818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 816可以在处理器802、存储器820和计算机系统800中的其他组件之间启动数据信号,并且在处理器总线810、存储器820和系统I/O 822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 816可以通过高带宽存储器路径818耦合到存储器820,并且图形/视频卡812可以通过加速图形端口(Accelerated Graphics Port)(“AGP”)互连814耦合到MCH 816。

[0099] 在至少一个实施例中,计算机系统800可以使用系统I/O 822作为专有集线器接口总线来将MCH 816耦合到I/O控制器集线器(“ICH”)830。在至少一个实施例中,ICH 830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820、芯片组和处理器802的高速I/O总线。示例可以包括但不限于音频控制器829、固件集线器(“Flash BIOS”)828、无线收发器826、数据存储器824、包含用户输入的传统I/O控制器823和键盘接口825、串行扩展端口827(例如通用串行总线(USB))和网络控制器834。数据存储装置824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。

[0100] 在至少一个实施例中,图8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8可以示出示例性片上系统(“SoC”)。在至少一个实施例中,图8中所示的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800的一个或更多个组件使用计算快速链路(CXL)互连来互连。

[0101] 推理和/或训练逻辑615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。下面结合图6A和/或6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例中,推理和/或训练逻辑615可以在系统图8中使用,用于至少部分地基于使用神经网络训练操作、神经网络功能和/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。

[0102] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0103] 图9是示出根据至少一个实施例的用于利用处理器910的电子设备900的框图。在至少一个实施例中,电子设备900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。

[0104] 在至少一个实施例中,系统900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910。在至少一个实施例中,处理器910使用总线或接口耦合,诸如1℃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9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9可以示出示例性片上系统(“SoC”)。在至少一个实施例中,图9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9的一个或更多个组件使用计算快速链路(CXL)互连线来互连。

[0105] 在至少一个实施例中,图9可以包括显示器924、触摸屏925、触摸板930、近场通信单元(“NFC”)945、传感器集线器940、热传感器946、快速芯片组(“EC”)935、可信平台模块(“TPM”)938、BIOS/固件/闪存(“BIOS,FW Flash”)922、DSP 960、驱动器920(例如固态磁盘(“SSD”)或硬盘驱动器(“HDD”))、无线局域网单元(“WLAN”)950、蓝牙单元952、无线广域网单元(“WWAN”)956、全球定位系统(GPS)955、相机(“USB 3.0相机”)954(例如USB 3.0相机)和/或实现在例如LPDDR3标准低功耗双倍数据速率(“LPDDR”)存储单元(“LPDDR3”)915。这些组件可以各自以任何合适的方式实现。

[0106] 在至少一个实施例中,其他组件可以通过以上讨论的组件通信地耦合到处理器910。在至少一个实施例中,加速度计941、环境光传感器(“ALS”)942、罗盘943和陀螺仪944可以可通信地耦合到传感器集线器940。在至少一个实施例中,热传感器939、风扇937、键盘936和触摸板930可以通信地耦合到EC935。在至少一个实施例中,扬声器963、耳机964和麦克风(“mic”)965可以通信地耦合到音频单元(“音频编解码器和D类放大器”)962,其又可以通信地耦合到DSP960。在至少一个实施例中,音频单元964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”)957可以通信地耦合到WWAN单元956。在至少一个实施例中,组件(诸如WLAN单元950和蓝牙单元952以及WWAN单元956)可以被实现为下一代形式因素(NGFF)。

[0107] 推理和/或训练逻辑615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。下面结合图6A和/或6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例中,推理和/或训练逻辑615可以在系统图9中使用,用于至少部分地基于使用神经网络训练操作、神经网络功能和/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。

[0108] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0109] 图10示出了根据至少一个实施例的计算机系统1000。在至少一个实施例中,计算机系统1000配置为实现贯穿本公开描述的各种过程和方法。

[0110] 在至少一个实施例中,计算机系统1000包括但不限于至少一个中央处理单元(“CPU”)1002,该中央处理单元(“CPU”)1002连接到使用任何合适协议实现的通信总线1010,诸如PCI(“外围设备互联”)、外围组件互连Express(“PCI-Express”)、AGP(“加速图形端口”)、超传输或任何其他总线或点对点通信协议。在至少一个实施例中,计算机系统1000

包括但不限于主存储器1004和控制逻辑(例如,实现为硬件、软件或其组合),并且数据可以采取随机存取存储器(“RAM”)的形式存储在主存储器1004中。在至少一个实施例中,网络接口子系统(“网络接口”)1022提供到其他计算设备和网络的接口,用于从计算机系统1000接收数据并将数据传输到其他系统。

[0111] 在至少一个实施例中,计算机系统1000在至少一个实施例中包括但不限于输入设备1008、并行处理系统1012和显示设备1006,它们可以使用常规的阴极视线管(“CRT”)、液晶显示器(“LCD”)、发光二极管(“LED”)、等离子显示器或其他合适的显示技术实现。在至少一个实施例中,从输入设备1008(诸如键盘、鼠标、触摸板、麦克风等)接收用户输入。在至少一个实施例中,前述模块中的每一个可以位于单个半导体平台上以形成处理系统。

[0112] 推理和/或训练逻辑615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。下面结合图6A和/或6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例中,推理和/或训练逻辑615可以在系统图10中使用,以至少部分地基于使用神经网络训练操作、神经网络功能和/或架构或本文所述的神经网络用例计算出的权重参数来进行推理或预测操作。

[0113] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0114] 图11示出了根据至少一个实施例的计算机系统1100。在至少一个实施例中,计算机系统1100包括但不限于计算机1110和USB盘1120。在至少一个实施例中,计算机1110可以包括但不限于任何数量和类型的处理器(未示出)和存储器(未示出)。在至少一个实施例中,计算机1110包括但不限于服务器、云实例、膝上型计算机和台式计算机。

[0115] 在至少一个实施例中,USB盘1120包括但不限于处理单元1130、USB接口1140和USB接口逻辑1150。在至少一个实施例中,处理单元1130可以是任何指令执行系统、装置或能够执行指令的设备。在至少一个实施例中,处理单元1130可以包括但不限于任何数量和类型的处理核心(未示出)。在至少一个实施例中,处理核心1130包括专用集成电路(“ASIC”),该专用集成电路被优化为执行与机器学习相关联的任何数量和类型的操作。例如,在至少一个实施例中,处理核心1130是张量处理单元(“TPC”),其被优化以执行机器学习推理操作。在至少一个实施例中,处理核心1130是视觉处理单元(“VPU”),其被优化以执行机器视觉和机器学习推理操作。

[0116] 在至少一个实施例中,USB接口1140可以是任何类型的USB连接器或USB插座。例如,在至少一个实施例中,USB接口1140是用于数据和电源的USB 3.0Type-C插座。在至少一个实施例中,USB接口1140是USB 3.0Type-A连接器。在至少一个实施例中,USB接口逻辑1150可以包括使处理单元1130能够经由USB连接器1140与设备(例如计算机1110)相连接的任何数量和类型的逻辑。

[0117] 推理和/或训练逻辑615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。下面结合图6A和/或6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例中,推理和/或训练逻辑615可以在系统图11中使用,至少部分地基于使用神经网络训练操作、神经网络功能和/或架构或本文所述的神经网络用例计算出的权重参数来推理或预测操作。

[0118] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0119] 图12A示出了示例性架构,其中多个GPU 1210-1213通过高速链路1240-1243(例如,总线、点对点互连等)通信地耦合到多个多核心处理器1205-1206。在一个实施例中,高速链路1240-1243支持4GB/s、30GB/s、80GB/s或更高的通信吞吐量。可以使用各种互连协议,包括但不限于PCIe 4.0或5.0和NVLink 2.0。

[0120] 另外,在一个实施例中,GPU 1210-1213中的两个或更多个通过高速链路1229-1230互连,可以使用与用于高速链路1240-1243的协议/链路相同或不同的协议/链路来实现。类似地,两个或更多个多核心处理器1205-1206可以通过高速链路1228连接,该高速链路可以是以20GB/s、30GB/s、120GB/s或更高的速度运行的对称多处理器(SMP)总线。可替代地,图12A中所示的各种系统组件之间的所有通信可以使用相同的协议/链路(例如,通过共同的互连结构)来完成。

[0121] 在一个实施例中,每个多核心处理器1205-1206分别经由存储器互连1226-1227通信地耦合至处理器存储器1201-1202,并且每个GPU 1210-1213分别通过GPU存储器互连1250-1253通信地耦合至GPU存储器1220-1223。存储器互连1226-1227和1250-1253可以利用相同或不同的存储器访问技术。作为示例而非限制,处理器存储器1201-1202和GPU存储器1220-1223可以是易失性存储器,诸如动态随机存取存储器(DRAM)(包括堆叠的DRAM)\图形DDR SDRAM(GDDR)(例如,GDDR5、GDDR6)或高带宽存储器(HBM)和/或可以是非易失性存储器,例如3D XPoint或Nano-Ram。在一个实施例中,处理器存储器1201-1202的某些部分可以是易失性存储器,而另一部分可以是非易失性存储器(例如,使用两级存储器(2LM)层次结构)。

[0122] 如下所述,尽管各种处理器1205-1206和GPU 1210-1213可以分别物理耦合到特定的存储器1201-1202、1220-1223,可以实现统一的存储器架构,其中相同的虚拟系统地址空间(也称为“有效地址”空间)分布在各种物理存储器之间。例如,处理器存储器1201-1202每个可以包括64GB的系统存储器地址空间,而GPU存储器1220-1223每个可以包括32GB的系统存储器地址空间(在此示例中总共为256GB可寻址存储器)。

[0123] 图12B示出了根据一个示例性实施例的用于多核心处理器1207和图形加速模块1246之间的互连的附加细节。图形加速模块1246可以包括集成在通过高速链路1240耦合到处理器1207的线卡上的一个或更多个GPU芯片。可替代地,图形加速模块1246可以与处理器1207集成在同一封装或芯片上。

[0124] 在至少一个实施例中,所示的处理器1207包括多个核心1260A-1260D,每个核心具有转换后备缓冲区1261A-1261D和一个或更多个高速缓存1262A-1262D。在至少一个实施例中,核心1260A-1260D可以包括用于执行未示出的指令和处理数据的各种其他组件。高速缓存1262A-1262D可以包括1级(L1)和2级(L2)高速缓存。此外,一个或更多个共享高速缓存1256可以被包括在高速缓存1262A-1262D中并且由一组核心1260A-1260D共享。例如,处理器1207的一个实施例包括12个核心,每个核心具有其自己的L1高速缓存、十二个共享的L2高速缓存和十二个共享的L3高速缓存。在这一实施例中,两个相邻核心共享一个或更多个L2和L3高速缓存。处理器1207和图形加速模块1246与系统存储器1214连接,其可以包括图

12A的处理器存储器1201-1202。

[0125] 经由相干总线1264上的核心间通信,为存储在各种高速缓存1262A-1262D、1256和系统存储器1214中的数据和指令保持相干。例如,每个高速缓存可具有与其关联的高速缓存相干逻辑/电路以响应于所检测到的对特定高速缓存线的读取或写入通过相干总线1264进行通信。在一种实现中,在相干总线1264上实现了高速缓存监听协议以监听高速缓存访问。

[0126] 在一个实施例中,代理电路1225将图形加速模块1246通信地耦合到相干总线1264,从而允许图形加速模块1246作为核心1260A-1260D的对等方参与高速缓存相干协议。特别地,接口1235通过高速链路1240(例如,PCIe总线、NVLink等)提供到代理电路1225的连接,并且接口1237将图形加速模块1246连接到链路1240。

[0127] 在一种实现中,加速器集成电路1236代表图形加速模块1246的多个图形处理引擎1231、1232、N提供高速缓存管理、存储器访问、环境管理和中断管理服务。图形处理引擎1231、1232、N可以各自包括单独的图形处理单元(GPU)。可选地,图形处理引擎1231、1232、N可包括GPU内的不同类型的图形处理引擎,例如图形执行单元、媒体处理引擎(例如,视频编码器/解码器)、采样器和blit引擎。在至少一个实施例中,图形加速模块1246可以是具有多个图形处理引擎1231-1232、N或图形处理引擎1231-1232的GPU,N可以是集成在通用封装、线卡或芯片上的各个GPU。

[0128] 在一个实施例中,加速器集成电路1236包括存储器管理单元(MMU)1239,用于执行各种存储器管理功能,例如虚拟到物理的存储器转换(也称为有效到实际的存储器转换)和用于访问系统存储器1214的存储器访问协议。MMU1239还可以包括转换后备缓冲区(TLB)(未示出),用于将虚拟/有效地址缓存为物理/实际地址转换。在一种实现中,高速缓存1238存储命令和数据,以供图形处理引擎1231-1232、N有效访问。在一个实施例中,存储在高速缓存1238和图形存储器1233-1234、M中的数据与核心高速缓存1262A-1262D、1256和系统存储器1214保持一致。如上所述,这可以经由代理电路1225代表高速缓存1238和存储器1233-1234、M来完成(例如,将与处理器缓存1262A-1262D,1256上的高速缓存行的修改/访问有关的更新发送到高速缓存1238,并从高速缓存1238接收更新)。

[0129] 一组寄存器1245存储由图形处理引擎1231-1232、N执行的线程的环境数据,并且环境管理电路1248管理线程环境。例如,环境管理电路1248可以执行保存和还原操作以在环境切换期间保存和还原各种线程的环境(例如,其中保存第一线程并且存储第二线程,使得第二线程可以由图形处理引擎执行)。例如,在环境切换器上,环境管理电路1248可以将当前寄存器值存储到存储器中的指定区域(例如,由环境指针标识)。然后,当返回环境时,它可以恢复寄存器值。在一个实施例中,中断管理电路1247接收并处理从系统设备接收的中断。

[0130] 在一个实施例中,MMU 1239将来自图形处理引擎1231的虚拟/有效地址转换为系统存储器1214中的实际/物理地址。加速器集成电路1236的一个实施例支持多个(例如4、8、16)图形加速器模块1246和/或其他加速器设备。图形加速器模块1246可以专用于在处理器1207上执行的单个应用,或者可以在多个应用之间共享。在一个实施例中,提出了一种虚拟化的图形执行环境,其中图形处理引擎1231-1232、N的资源与多个应用程序或虚拟机(VM)共享。在至少一个实施例中,可以基于与虚拟机和/或应用程序相关联的处理要求和优先

级,将资源细分为“片”,将其分配给不同的虚拟机和/或应用程序。

[0131] 在至少一个实施例中,加速器集成电路1236充当用于图形加速模块1246的系统的桥梁,并提供地址转换和系统存储器高速缓存服务。另外,加速器集成电路1236可以为主机处理器提供虚拟化设施,以管理图形处理引擎1231-1232、N的虚拟化,中断和存储器管理。

[0132] 因为图形处理引擎1231-1232、N的硬件资源被显式映射到主机处理器1207看到的实际地址空间,所以任何主机处理器都可以使用有效地址值直接寻址这些资源。在一个实施例中,加速器集成电路1236的一个功能是物理分隔图形处理引擎1231-1232、N,使得它们在系统中表现为独立的单元。

[0133] 在至少一个实施例中,一个或更多个图形存储器1233-1234、M分别耦合到图形处理引擎1231-1232、N中的每一个。图形存储器1233-1234、M存储由图形处理引擎1231-1232、N中的每一个处理的指令和数据。图形存储器1233-1234、M可以是易失性存储器,例如DRAM(包括堆叠的DRAM)、GDDR存储器(例如,GDDR5、GDDR6)或HBM,和/或可以是非易失性存储器,例如3D XPoint或Nano-Ram。

[0134] 在一个实施例中,为了减少链路1240上的数据流量,使用偏置技术以确保存储在图形存储器1233-1234、M中的数据将是图形处理引擎1231-1232、N最频繁使用的数据,并且最好不被核心1260A-1260D使用(至少不经常使用)。类似地,偏置机制尝试将核心(最好不是图形处理引擎1231-1232、N)所需的数据保留在核心和系统存储器1214的高速缓存1262A-1262D、1256中。

[0135] 图12C示出了另一示例性实施例,其中加速器集成电路1236被集成在处理器1207内。在至少该实施例中,图形处理引擎1231-1232、N经由接口1237和接口1235通过高速链路1240直接通信到加速器集成电路1236(其中,再次可以使用任何形式的总线或接口协议)。加速器集成电路1236可以执行与关于图12B描述的相同的操作,但是鉴于其紧密靠近相干总线1264和高速缓存1262A-1262D、1256,可能具有较高的吞吐量。至少一个实施例支持不同的编程模型,包括专用过程编程模型(无图形加速模块虚拟化)和共享编程模型(具有虚拟化),其可以包括由加速器集成电路1236控制的编程模型和由图形加速模块1246控制的编程模型。

[0136] 在至少一个实施例中,图形处理引擎1231-1232、N专用于单个操作系统下的单个应用程序或过程。在至少一个实施例中,单个应用程序可以将其他应用请求集中到图形处理引擎1231-1232、N,从而在VM/分区内提供虚拟化。

[0137] 在至少一个实施例中,图形处理引擎1231-1232、N可以由多个VM/应用程序分区共享。在至少一个实施例中,共享模型可以使用系统管理程序来虚拟化图形处理引擎1231-1232、N以允许被每个操作系统访问。对于没有管理程序的单分区系统,操作系统拥有图形处理引擎1231-1232、N。在至少一个实施例中,操作系统可以虚拟化图形处理引擎1231-1232、N以提供对每个进程或应用程序的访问。

[0138] 在至少一个实施例中,图形加速模块1246或单独的图形处理引擎1231-1232、N使用过程句柄来选择过程元素。在至少一个实施例中,处理元件被存储在系统存储器1214中,并且可使用本文所述的有效地址到实地址转换技术来寻址。在至少一个实施例中,过程句柄可以是在向图形处理引擎1231-1232、N注册其环境时向主机过程提供的特定于实现的值(即,调用系统软件以将过程元素添加到过程元素链表)。在至少一个实施例中,过程句柄的

低16位可以是过程元素链表中过程元素的偏移量。

[0139] 图12D示出了示例性加速器集成切片1290。如本文所使用的,“片”包括加速器集成电路1236的处理资源的指定部分。系统存储器1214内的应用有效地址空间1282存储过程元素1283。在一个实施例中,响应于来自处理器1207上执行的应用程序1280的GPU调用1281而存储过程元素1283。过程元素1283包含对应应用程序1280的处理状态。包含在处理元素1283中的工作描述符(WD)1284可以是应用程序请求的单个作业或可能包含指向作业队列的指针。在至少一个实施例中,WD 1284是指向应用程序地址空间1282中的作业请求队列的指针。

[0140] 图形加速模块1246和/或各个图形处理引擎1231-1232、N可以由系统中的全部或部分进程共享。在至少一个实施例中,可以包括用于建立处理状态并将WD 1284发送到图形加速模块1246以在虚拟化环境中开始作业的基础设施。

[0141] 在至少一个实施例中,专用进程编程模型是针对实现的。在该模型中,单个进程拥有图形加速模块1246或单个图形处理引擎1231。由于图形加速模块1246由单个进程拥有,因此管理程序为拥有的分区初始化加速器集成电路1236,并且当分配图形加速模块1246时操作系统对加速器集成电路1236进行初始化以用于拥有的分区。

[0142] 在操作中,加速器集成切片1290中的WD获取单元1291获取下一个WD 1284,其中包括要由图形加速模块1246的一个或更多个图形处理引擎完成的工作的指示。来自WD 1284的数据可以存储在寄存器1245被MMU 1239、中断管理电路1247和/或环境管理电路1248使用,如图所示。例如,MMU 1239的一个实施例包括用于访问OS虚拟地址空间1285内的段/页表1286的段/页面漫游电路。中断管理电路1247可以处理从图形加速模块1246接收到的中断事件1292。当执行图形操作时,由图形处理引擎1231-1232、N产生的有效地址1293由MMU 1239转换为实际地址。

[0143] 在一个实施例中,为每个图形处理引擎1231-1232、N和/或图形加速模块1246复制相同的寄存器组1245,并且可以由系统管理程序或操作系统来初始化。这些复制的寄存器中的每一个都可以包含在加速器集成切片1290中。表1中显示了可由管理程序初始化的示例性寄存器。

[0144] 表1-管理程序初始化的寄存器

[0145]

1	片控制寄存器
2	实址(RA)计划的处理区域指针
3	授权掩码覆盖寄存器
4	中断向量表输入偏移
5	中断向量表入口限制
6	状态寄存器
7	逻辑分区ID
8	实址(RA)管理程序加速器利用率记录指针
9	存储描述寄存器

[0146] 表2中示出了可以由操作系统初始化的示例性寄存器。

[0147] 表2-操作系统初始化寄存器

[0148]

1	进程和线程识别
2	有效地址（EA）环境保存/还原指针
3	虚拟地址（VA）加速器利用率记录指针
4	虚拟地址（VA）存储分段表指针
5	权威面具

[0149]

6	工作描述符

[0150] 在一个实施例中,每个WD 1284特定于特定的图形加速模块1246和/或图形处理引擎1231-1232、N。它包含图形处理引擎1231-1232、N进行工作或工作所需的所有信息,或者它可以是指向存储器位置的指针,其中应用程序建立了要完成的工作的命令队列。

[0151] 图12E示出了共享模型的一个示例性实施例的附加细节。该实施例包括管理程序实地址空间1298,其中存储了过程元素列表1299。可以通过管理程序1296来访问管理程序实际地址空间1298,其中管理程序1296对图形加速模块引擎进行虚拟化以用于操作系统1295。

[0152] 在至少一个实施例中,共享的编程模型允许系统中全部或部分分区的全部或部分进程使用图形加速模块1246。存在两个共享图形加速模块1246的编程模型由多个进程和分区共享:时间片共享和图形方向共享。

[0153] 在此模型中,系统管理程序1296拥有图形加速模块1246,并且其功能可用于所有操作系统1295。为使图形加速模块1246支持系统管理程序1296进行虚拟化,图形加速模块1246可遵循以下规定:1)应用程序的作业请求必须是自主的(即,不需要在作业之间保持状态),或者图形加速模块1246必须提供环境保存和还原机制。2)应用程序的作业请求由图形加速模块1246保证以在指定的时间内完成,包括任何翻译错误,或者图形加速模块1246提供了抢占作业处理的能力。3)在有定向共享编程模型中操作时,必须确保图形加速模块1246在进程之间的公平性。

[0154] 在至少一个实施例中,需要应用程序1280进行具有图形加速模块1246类型、工作描述符(WD)、权限掩码寄存器(AMR)值和环境保存/恢复区域指针(CSRP)的操作系统1295系统调用。在至少一个实施例中,图形加速模块1246类型描述了用于系统调用的目标加速功能。在至少一个实施例中,图形加速模块1246的类型可以是针对系统的值。在至少一个实施例中,WD是专门为图形加速模块1246格式化的,并且可以采用图形加速模块1246命令、指向用户定义结构的有效地址指针、指向命令队列的有效地址指针或任何其他数据结构的形式来描述要由图形加速模块1246完成的工作。在一个实施例中,AMR值是要用于当前过程的AMR状态。在至少一个实施例中,传递给操作系统的值类似于AMR的应用程序设置。如果加速器集成电路1236和图形加速模块1246实现不支持用户权限掩码覆盖寄存器(UAMOR),则操

作系统可以在管理程序调用中传递AMR之前将当前UAMOR值应用于AMR值。管理程序1296可以在将AMR放入过程元素1283之前,可选地应用当前的授权掩码覆盖寄存器(AMOR)值。在至少一个实施例中,CSRP是寄存器1245之一,其中包含应用程序有效地址空间1282中某个区域的有效地址,用于图形加速模块1246保存和恢复环境状态。如果不需要在作业之间保存状态或当作业被抢占时,则此指针是可选的。在至少一个实施例中,环境保存/恢复区域可以是固定的系统存储器。

[0155] 在接收到系统调用之后,操作系统1295可以验证应用程序1280已经被注册并且被授予使用图形加速模块1246的权限。然后,操作系统1295使用表3中所示的信息来调用系统管理程序1296。

[0156] 表3-OS到管理程序(Hypervisor)的调用参数

[0157]

1	工作描述符(WD)
2	授权屏蔽寄存器(AMR)值(可能被屏蔽)
3	有效地址(EA)环境保存/还原区域指针(CSRP)
4	进程ID(PID)和可选线程ID(TID)
5	虚拟地址(VA)加速器利用率记录指针(AURP)
6	存储分段表指针(SSTP)的虚拟地址
7	逻辑中断服务号码(LISN)

[0158] 在接收到系统管理程序调用后,系统管理程序1296验证操作系统1295已注册并被授予使用图形加速模块1246的权限。系统管理程序1296然后将过程元素1283放入对应的图形加速模块1246类型的过程元素链表中。过程元素可能包含表4中所示的信息。

[0159] 表4-过程元素信息

[0160]

1	工作描述符（WD）

[0161]

2	授权屏蔽寄存器（AMR）值（可能被屏蔽）。
3	有效地址（EA）环境保存/还原区域指针（CSRP）
4	进程 ID（PID）和可选线程 ID（TID）
5	虚拟地址（VA）加速器利用率记录指针（AURP）
6	存储器分段表指针（SSTP）的虚拟地址
7	逻辑中断服务号码（LISN）
8	从管理程序调用参数派生的中断向量表
9	状态寄存器（SR）值
10	逻辑分区 ID（LPID）
11	实地址（RA）管理程序加速器利用率记录指针
12	存储器描述符寄存器（SDR）

[0162] 在至少一个实施例中,管理程序初始化多个加速器集成切片1290寄存器1245。

[0163] 如图12F所示,在至少一个实施例中,使用统一存储器,该统一存储器可通过用于访问物理处理器存储器1201-1202和GPU存储器1220-1223的公共虚拟存储器地址空间来寻址。在该实现中,在GPU 1210-1213上执行的操作利用相同的虚拟/有效存储器地址空间来访问处理器存储器1201-1202,反之亦然,从而简化了可编程性。在一个实施例中,虚拟/有效地址空间的第一部分被分配给处理器存储器1201,第二部分被分配给第二处理器存储器1202,第三部分被分配给GPU存储器1220,等等。在至少一个实施例中,整个虚拟/有效存储器空间(有时称为有效地址空间)由此分布在处理器存储器1201-1202和GPU存储器1220-1223的每一个中,从而允许任何处理器或GPU访问具有映射到该存储器的虚拟地址的任何物理存储器。

[0164] 在一个实施例中,一个或更多个MMU 1239A-1239E内的偏置/相干管理电路1294A-1294E确保一个或更多个主机处理器(例如1205)和GPU 1210-1213的高速缓存之间的高速缓存相干,并实现指示物理存储器的偏置技术,其中应存储某些类型的数据。尽管在图12F中示出了偏置/相干管理电路1294A-1294E的多个实例,可以在一个或更多个主机处理器1205的MMU内和/或在加速器集成电路1236内实现偏置/相干电路。

[0165] 一个实施例允许附有GPU的存储器1220-1223被映射为系统存储器的一部分,并使用共享虚拟存储器(SVM)技术进行访问,但是没有遭受与完整的系统高速缓存相干相关的性能缺陷。在至少一个实施例中,将附接的GPU的存储器1220-1223作为系统存储器来访问

而没有繁重的高速缓存相干开销的能力为GPU卸载提供了有益的操作环境。这种安排允许主机处理器1205软件设置操作数并访问计算结果,而不会产生传统I/O DMA数据副本的开销。此类传统副本涉及驱动程序调用、中断和存储器映射的I/O(MMIO)访问,相对于简单的存储器访问而言,这些访问效率均较低。在至少一个实施例中,在没有高速缓存相干开销的情况下访问GPU附接的存储器1220-1223的能力对于卸载计算的执行时间是至关重要的。例如,在具有大量流写入存储器流量的情况下,缓存相干开销会大大降低GPU 1210-1213看到的有效写入带宽。在至少一个实施例中,操作数设置的效率、结果访问的效率以及GPU计算的效率可以在确定GPU卸载的效率中起作用。

[0166] 在至少一个实施例中,GPU偏置和主处理器偏置的选择由偏置跟踪器数据结构驱动。可以使用例如偏置表,该偏置表可以是页面粒度结构(即,以存储器页面的粒度来控制),每个GPU连接的存储器页面包括1或2位。在至少一个实施例中,可以在一个或更多个GPU附接的存储器1220-1223的被盗存储器范围中实现偏置表,在GPU 1210-1213中具有或不具有偏置缓存(例如,以高速缓存频繁/最近使用的偏置表条目)。可替代地,可以在GPU内维护整个偏置表。

[0167] 在至少一个实施例中,在实际访问GPU存储器之前访问与对GPU附接的存储器1220-1223的每次访问相关联的偏置表条目,从而导致以下操作。首先,来自GPU 1210-1213的本地请求在GPU偏置中找到其页面被直接转发到相应的GPU存储器1220-1223。来自GPU的在主机偏置中找到其页面的本地请求被转发到处理器1205(例如,通过如上所述的高速链路)。在一个实施例中,来自处理器1205的在主机处理器偏置中找到所请求页面的请求完成了与普通存储器读取类似的请求。备选地,可以将针对GPU偏置的页面的请求转发给GPU 1210-1213。在至少一个实施例中,如果GPU当前不使用页面,则GPU可以将页面转换为主机处理器偏置。在至少一个实施例中,页面的偏置状态可以通过基于软件的机制、基于硬件辅助的软件的机制,或者在有限的情况下,可以纯粹基于硬件的机制来改变。

[0168] 一种用于改变偏置状态的机制采用API调用(例如OpenCL),该调用反过来又调用GPU的设备驱动程序,该设备驱动程序反过来又向GPU发送消息(或排队命令描述符),将其引导到更改偏置状态,并在某些转换中在主机中执行缓存刷新操作。在至少一个实施例中,高速缓存刷新操作用于从主机处理器1205偏置到GPU偏置的转变,但是不用于相反的转变。

[0169] 在一个实施例中,通过暂时渲染由主机处理器1205无法缓存的GPU偏置的页面,来保持缓存相干。为了访问这些页面,处理器1205可以请求从GPU 1210进行访问,其可以立即授予或不授予访问权限。因此,为了减少处理器1205与GPU 1210之间的通信,有利的是确保GPU偏置的页面是GPU所需要的页面,而不是主机处理器1205所需要的页面,反之亦然。

[0170] 推理和/或训练逻辑615用于执行一个或更多个实施例。下面结合图6A和/或6B提供关于推理和/或训练逻辑615的细节。

[0171] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0172] 图13是示出根据至少一个实施例的可使用一个或更多个IP核心制造的芯片集成电路1300上的示例性系统的框图。在至少一个实施例中,集成电路1300包括一个或更多个应用处理器1305(例如,CPU)、至少一个图形处理器1310,并且可以另外包括图像处理器

1315和/或视频处理器1320,其中任意一个可能是模块化IP核心。在至少一个实施例中,集成电路1300包括外围或总线逻辑,其包括USB控制器1325、UART控制器1330、SPI/SDIO控制器1335和I^2S/I^2C控制器1340。在至少一个实施例中,集成电路1300可以包括显示设备1345耦合到高清多媒体接口(HDMI)控制器1350和移动工业处理器接口(MIPI)显示接口1355中的一个或更多个。在至少一个实施例中,存储器可以由闪存子系统1360提供,包括闪存和闪存控制器。在至少一个实施例中,可以经由存储器控制器1365提供存储器接口以用于访问SDRAM或SRAM存储器设备。在至少一个实施例中,一些集成电路还包括嵌入式安全引擎1370。

[0173] 推理和/或训练逻辑615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。下面结合图6A和/或6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例中,推理和/或训练逻辑615可以在集成电路1300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。

[0174] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0175] 图14A-14B示出了根据本文所述的各种实施例的示例性集成电路和相关联的图形处理器,其可以使用一个或更多个IP核心来制造。除了图示之外,在至少一个实施例中可以包括其他逻辑和电路,包括附加的图形处理器/核心、外围接口控制器或通用处理器核心。

[0176] 图14A-14B是示出根据本文描述的实施例的在SoC内使用的示例性图形处理器的框图。图14A示出了根据至少一个实施例的芯片集成电路上系统的示例性图形处理器1410,其可以使用一个或更多个IP核心来制造。图14B示出了根据至少一个实施例的芯片集成电路上系统的另外示例性图形处理器1440,其可以使用一个或更多个IP核心来制造。在至少一个实施例中,图14A的图形处理器1410是低功耗图形处理器核心。在至少一个实施例中,图14B的图形处理器1440是更高性能的图形处理器核心。在至少一个实施例中,每个图形处理器1410、1440可以是图13的图形处理器1310的变体。

[0177] 在至少一个实施例中,图形处理器1410包括顶点处理器1405和一个或更多个片段处理器1415A-1415N(例如1415A、1415B、1415C、1415D至1415N-1和1415N)。在至少一个实施例中,图形处理器1410可以经由单独的逻辑来执行不同的着色器程序,使得顶点处理器1405被优化以执行针对顶点着色器程序的操作,而一个或更多个片段处理器1415A-1415N执行片段(例如,像素)着色操作用于片段或像素或着色器程序。在至少一个实施例中,顶点处理器1405执行3D图形管线的顶点处理阶段并生成图元和顶点数据。在至少一个实施例中,片段处理器1415A-1415N使用由顶点处理器1405生成的图元和顶点数据来生成在显示设备上显示的帧缓冲区。在至少一个实施例中,片段处理器1415A-1415N被优化以执行如在OpenGL API中所提供的片段着色器程序,其可以用于执行与在Direct 3D API中所提供的像素着色器程序类似的操作。

[0178] 在至少一个实施例中,图形处理器1410附加地包括一个或更多个存储器管理单元(MMU)1420A-1420B、高速缓存1425A-1425B和电路互连1430A-1430B。在至少一个实施例中,一个或更多个MMU 1420A-1420B提供用于图形处理器1410的虚拟到物理地址的映射,包括

用于顶点处理器1405和/或片段处理器1415A-1415N,其可以引用存储在存储器中的顶点或图像/纹理数据,除了存储在一个或更多个高速缓存1425A-1425B中的顶点或图像/纹理数据之外。在至少一个实施例中,一个或更多个MMU1420A-1420B可以与系统内的其他MMU同步,包括与图像13的一个或更多个应用处理器1305、图像处理器1315和/或视频处理器1320相关联的一个或更多个MMU,使得每个处理器1305-1320可以参与共享或统一的虚拟存储器系统。在至少一个实施例中,一个或更多个电路互连1430A-1430B使图形处理器1410能够经由SoC的内部总线或经由直接连接与SoC内的其他IP核心相连接。

[0179] 在至少一个实施例中,图形处理器1440包括图14A的图形处理器1410的一个或更多个MMU 1420A-1420B、高速缓存1425A-1425B和电路互连1430A-1430B。在至少一个实施例中,图形处理器1440包括一个或更多个着色器核心1455A-1455N(例如,1455A、1455B、1455C、1455D、1455E、1455F、直到1455N-1和1455N),其提供了统一的着色器核心架构,其中单个核心或类型或核心可以执行所有类型的可编程着色器代码,包括用于实现顶点着色器、片段着色器和/或计算着色器的着色器程序代码。在至少一个实施例中,多个着色器核心可以变化。在至少一个实施例中,图形处理器1440包括核心间任务管理器1445,其充当线程分派器以将执行线程分派给一个或更多个着色器核心1455A-1455N和分块单元1458,以加速基于图块渲染的分块操作,其中在图像空间中细分了场景的渲染操作,例如,以利用场景内的局部空间相干或优化内部缓存的使用。

[0180] 推理和/或训练逻辑615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。下面结合图6A和/或6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例中,推理和/或训练逻辑615可以在集成电路14A和/或14B中用于至少部分地基于使用神经网络训练操作、神经网络功能或架构,或本文所述的神经网络用例计算的权重参数来进行推理或预测操作。

[0181] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0182] 图15A-15B示出了根据本文描述的实施例的附加的示例性图形处理器逻辑。在至少一个实施例中,图15A示出了可以包括在图13的图形处理器1310内的图形核心1500,在至少一个实施例中,可以是图14B中统一的着色器核心1455A-1455N。图15B示出了在至少一个实施例中的适用于在多芯片模块上部署的高度并行的通用图形处理单元1530。

[0183] 在至少一个实施例中,图形核心1500包括共享指令高速缓存1502、纹理单元1518和高速缓存/共享存储器1520,它们是图形核心1500内的执行资源所共有的。在至少一个实施例中,图形核心1500可以包括多个切片1501A-1501N或每个核心的分区,图形处理器可以包括图形核心1500的多个实例。切片1501A-1501N可以包括支持逻辑,该支持逻辑包括本地指令高速缓存1504A-1504N、线程调度器1506A-1506N、线程分派器1508A-1508N和一组寄存器1510A-1510N。在至少一个实施例中,切片1501A-1501N可以包括一组附加功能单元(AFU 1512A-1512N)、浮点单元(FPU 1514A-1514N)、整数算术逻辑单元(ALU 1516-1516N)、地址计算单元(ACU 1513A-1513N)、双精度浮点单元(DPFPU 1515A-1515N)和矩阵处理单元(MPU 1517A-1517N)。

[0184] 在至少一个实施例中,FPU 1514A-1514N可以执行单精度(32位)和半精度(16位)

浮点运算,而DPFPU 1515A-1515N可以执行双精度(64位)浮点运算点操作。在至少一个实施例中,ALU 1516A-1516N可以以8位、16位和32位精度执行可变精度整数运算,并且可以被配置用于混合精度运算。在至少一个实施例中,MPU 1517A-1517N还可被配置用于混合精度矩阵运算,包括半精度浮点运算和8位整数运算。在至少一个实施例中,MPU 1517A-1517N可以执行各种矩阵操作以加速机器学习应用框架,包括使得能够支持加速的通用矩阵到矩阵乘法(GEMM)。在至少一个实施例中,AFU 1512A-1512N可以执行浮点数或整数单元不支持的附加逻辑运算,包括三角运算(例如,Sine、Cosine等)。

[0185] 推理和/或训练逻辑615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。下面结合图6A和/或6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例中,推理和/或训练逻辑615可以在图形核心1500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。

[0186] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0187] 图15B示出了在至少一个实施例中的通用处理单元(GPGPU)1530,其可以被配置为使得高度并行的计算操作能够由图形处理单元阵列来执行。在至少一个实施例中,GPGPU 1530可以直接链路到GPGPU 1530的其他实例,以创建多GPU集群以提高用于深度神经网络的训练速度。在至少一个实施例中,GPGPU 1530包括主机接口1532以实现与主机处理器的连接。在至少一个实施例中,主机接口1532是PCI Express接口。在至少一个实施例中,主机interjace1532可以是厂商专用的通信接口或通信结构。在至少一个实施例中,GPGPU 1530从主机处理器接收命令,并使用全局调度器1534将与那些命令相关联的执行线程分派给一组计算集群1536A-1536H。在至少一个实施例中,计算集群1536A-1536H共享高速缓冲存储器1538。在至少一个实施例中,高速缓冲存储器1538可以用作计算集群1536A-1536H内的高速缓冲存储器的高级高速缓存。

[0188] 在至少一个实施例中,GPGPU 1530包括经由一组存储器控制器1542A-1542B与计算集群1536A-1536H耦合的存储器1544A-1544B。在至少一个实施例中,存储器1544A-1544B可以包括各种类型的存储器设备,包括动态随机存取存储器(DRAM)或图形随机存取存储器,例如同步图形随机存取存储器(SGRAM),包括图形双倍数据速率(GDDR)存储器。

[0189] 在至少一个实施例中,计算集群1536A-1536H各自包括一组图形核心,诸如图15A的图形核心1500,其可以包括多种类型的整数和浮点逻辑单元,其可以以包括用于在适用于机器学习计算的精度范围内执行计算操作。例如,在至少一个实施例中,每个计算集群1536A-1536H中的浮点单元的至少一个子集可以配置为执行16位或32位浮点运算,而不同的浮点单元的子集可以配置为执行64位浮点运算。

[0190] 在至少一个实施例中,GPGPU 1530的多个实例可以被配置为操作为计算集群。在至少一个实施例中,计算集群1536A-1536H用于同步和数据交换的通信在实施例之间变化。在至少一个实施例中,GPGPU 1530的多个实例通过主机接口1532进行通信。在至少一个实施例中,GPGPU 1530包括I/O集线器1539,其将GPGPU 1530与GPU链路1540耦合,使得能够直接连接至GPGPU1530的其他的实例。在至少一个实施例中,GPU链路1540耦合到专用GPU到

GPU桥接器,其使得能够在GPGPU 1530的多个实例之间进行通信和同步。在至少一个实施例中,GPU链路1540与高速互连耦合,以向其他GPGPU或并行处理器发送和接收数据。在至少一个实施例中,GPGPU1530的多个实例位于单独的数据处理系统中,并经由可经由主机接口1532访问的网络设备进行通信。在至少一个实施例中,GPU链路1540可被配置为能够连接到主机处理器,附加或替代主机接口1532。

[0191] 在至少一个实施例中,GPGPU 1530可以配置为训练神经网络。在至少一个实施例中,可以在推理平台内使用GPGPU 1530。在其中使用GPGPU 1530进行推理的至少一个实施例中,相对于当使用GPGPU训练神经网络时,GPGPU可以包括更少的计算集群1536A-1536H。在至少一个实施例中,与存储器1544A-1544B相关联的存储器技术可以在推理和训练配置之间有所不同,其中更高带宽的存储器技术专用于训练配置。在至少一个实施例中,GPGPU 1530的推理配置可以支持推理特定指令。例如,在至少一个实施例中,推理配置可以提供对一个或更多个8位整数点积指令的支持,该指令可以在部署的神经网络的推理操作期间使用。

[0192] 推理和/或训练逻辑615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。下面结合图6A和/或6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例中,推理和/或训练逻辑615可以在GPGPU 1530中用于至少部分地基于本文描述的使用神经网络训练操作、神经网络功能和/或架构或神经网络用例计算的权重参数来推理或预测操作。

[0193] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0194] 图16是示出根据至少一个实施例的计算系统1600的框图。在至少一个实施例中,计算系统1600包括具有一个或更多个处理器1602的处理子系统1601和经由可以包括存储器集线器1605的互连路径通信的系统存储器1604。在至少一个实施例中,存储器集线器1605可以是芯片组组件内的单独组件,也可以集成在一个或更多个处理器1602内。在至少一个实施例中,存储器集线器1605通过通信链路1606与I/O子系统1611耦合。在一实施例中,I/O子系统1611包括I/O集线器1607,其可以使计算系统1600能够接收来自一个或更多个输入设备1608的输入。在至少一个实施例中,I/O集线器1607可以使能显示控制器,其包括在一个或更多个处理器1602中,用于向一个或更多个显示设备1610A提供输出。在至少一个实施例中,与I/O集线器1607耦合的一个或更多个显示设备1610A可以包括本地、内部或嵌入式显示设备。

[0195] 在至少一个实施例中,处理子系统1601包括经由总线或其他通信链路1613耦合到存储器集线器1605的一个或更多个并行处理器1612。在至少一个实施例中,通信链路1613可以是许多基于标准的通信链路技术或协议中的一种,例如但不限于PCI Express,或者可以是针对供应商的通信接口或通信结构。在至少一个实施例中,一个或更多个并行处理器1612形成计算集中的并行或向量处理系统,该系统可以包括大量的处理核心和/或处理集群,例如多集成核心(MIC)处理器。在至少一个实施例中,一个或更多个并行处理器1612形成可以将像素输出到经由I/O集线器1607耦合的一个或更多个显示设备1610A之一的图形处理子系统。在至少一个实施例中,一个或更多个并行处理器1612还可以包括显示控制器

和显示接口(未示出),以使得能够直接连接到一个或更多个显示设备1610B。

[0196] 在至少一个实施例中,系统存储单元1614可以连接到I/O集线器1607,以提供用于计算系统1600的存储机制。在至少一个实施例中,I/O开关1616可以用于提供接口机制,以实现I/O集线器1607与其他组件之间的连接,例如可以集成到平台中的网络适配器1618和/或无线网络适配器1619,以及可以通过一个或更多个附加设备1620添加的各种其他设备。在至少一个实施例中,网络适配器1618可以是以太网适配器或另一有线网络适配器。在至少一个实施例中,无线网络适配器1619可以包括Wi-Fi、蓝牙、近场通信(NFC)的一个或更多个或其他包括一个或更多个无线电的网络设备。

[0197] 在至少一个实施例中,计算系统1600可以包括未明确示出的其他组件,包括USB或其他端口连接、光存储驱动器、视频捕获设备等,也可以连接到I/O集线器1607。在至少一个实施例中,对图16中的各个组件进行互连的通信路径可以使用任何合适的协议来实现,诸如基于PCI(外围组件互连)的协议(例如,PCI-Express),或其他总线或点对点通信接口和/或协议(例如,NV-链路高速互连或互连协议)。

[0198] 在至少一个实施例中,一个或更多个并行处理器1612包括针对图形和视频处理而优化的电路(包括例如视频输出电路),并构成图形处理单元(GPU)。在至少一个实施例中,一个或更多个并行处理器1612包括针对通用处理而优化的电路。在至少一个实施例中,计算系统1600的组件可以与单个集成电路上的一个或更多个其他系统元件集成。例如,在至少一个实施例中,一个或更多个并行处理器1612、存储器集线器1605、处理器1602和I/O集线器1607可以被集成到片上系统(SoC)集成电路中。在至少一个实施例中,计算系统1600的组件可以被集成到单个封装中以形成系统级封装(SIP)配置。在至少一个实施例中,计算系统1600的组件的至少一部分可以被集成到多芯片模块(MCM)中,该多芯片模块可以与其他多芯片模块互连到模块化计算系统中。

[0199] 推理和/或训练逻辑615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。下面结合图6A和/或6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例中,推理和/或训练逻辑615可以在系统图1600中使用,用于至少部分地基于使用神经网络训练操作、神经网络功能和/或架构或本文所述的神经网络用例计算出的权重参数来推理或预测操作。

[0200] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0201] 处理器

[0202] 图17A示出了根据至少一个实施例的并行处理器1700。在至少一个实施例中,并行处理器1700的各种组件可以使用一个或更多个集成电路设备来实现,例如可编程处理器、专用集成电路(ASIC)或现场可编程门阵列(FPGA)。在至少一个实施例中,所示的并行处理器1700是根据示例性实施例的图16所示的一个或更多个并行处理器1612的变体。

[0203] 在至少一个实施例中,并行处理器1700包括并行处理单元1702。在至少一个实施例中,并行处理单元1702包括I/O单元1704,其使得能够与其他设备进行通信,包括并行处理单元1702的其他实例。在至少一个实施例中,I/O单元1704可以直接连接到其他设备。在至少一个实施例中,I/O单元1704通过使用集线器或交换机接口(例如,存储器集线器2805)

与其他设备连接。在至少一个实施例中,存储器集线器2805与I/O单元1704之间的连接形成通信链路2813。在至少一个实施例中,I/O单元1704与主机接口1706和存储器交叉开关1716连接,其中主机接口1706接收用于执行处理操作的命令,而存储器交叉开关1716接收用于执行存储器操作的命令。

[0204] 在至少一个实施例中,当主机接口1706经由I/O单元1704接收命令缓冲区时,主机接口1706可以引导工作操作以执行那些命令到前端1708。在至少一个实施例中,前端1708与调度器1710耦合,调度器1710配置成将命令或其他工作项分配给处理集群阵列1712。在至少一个实施例中,调度器1710确保在将任务分配给处理集群阵列1712之前,处理集群阵列1712被正确地配置并且处于有效状态。在至少一个实施例中,调度器1710通过在微控制器上执行的固件逻辑来实现。在至少一个实施例中,微控制器实现的调度器1710可配置成以粗粒度和细粒度执行复杂的调度和工作分配操作,从而实现对在处理阵列1712上执行的线程的快速抢占和环境切换。在至少一个实施例中,主机软件可以证明用于通过多个图形处理门铃之一在处理阵列1712上进行调度的工作负载。在至少一个实施例中,工作负载然后可以由包括调度器1710的微控制器内的调度器1710逻辑在处理阵列1712上自动分配。

[0205] 在至少一个实施例中,处理集群阵列1712可以包括多达“N”个处理集群(例如,集群1714A、集群1714B到集群1714N)。在至少一个实施例中,处理集群阵列1712的每个集群1714A-1714N可以执行大量并发线程。在至少一个实施例中,调度器1710可以使用各种调度和/或工作分配算法将工作分配给处理集群阵列1712的集群1714A-1714N,其可以根据每种程序或计算类型产生的工作负载而变化。在至少一个实施例中,调度可以由调度器1710动态地处理,或者可以在配置为由处理集群阵列1712执行的程序逻辑的编译期间部分地由编译器逻辑来辅助。在至少一个实施例中,可将处理集群阵列1712的不同的集群1714A-1714N分配用于处理不同类型的程序或用于执行不同类型的计算。

[0206] 在至少一个实施例中,处理集群阵列1712可以配置成执行各种类型的并行处理操作。在至少一个实施例中,处理集群阵列1712配置成执行通用并行计算操作。例如,在至少一个实施例中,处理集群阵列1712可以包括执行处理任务的逻辑,该处理任务包括对视频和/或音频数据的过滤,执行建模操作,包括物理操作以及执行数据转换。

[0207] 在至少一个实施例中,处理集群阵列1712配置成执行并行图形处理操作。在至少一个实施例中,处理集群阵列1712可以包括附加逻辑以支持这种图形处理操作的执行,包括但不限于执行纹理操作的纹理采样逻辑,以及镶嵌逻辑和其他顶点处理逻辑。在至少一个实施例中,处理集群阵列1712可以配置成执行与图形处理有关的着色器程序,例如但不限于顶点着色器、曲面细分着色器、几何着色器和像素着色器。在至少一个实施例中,并行处理单元1702可以经由I/O单元1704从系统存储器传送数据以进行处理。在至少一个实施例中,在处理期间,可以在处理期间将传送的数据存储到片上存储器(例如,并行处理器存储器1722),然后将其写回到系统存储器。

[0208] 在至少一个实施例中,当并行处理单元1702用于执行图形处理时,调度器1710可以配置成将处理工作负载划分为近似相等大小的任务,以更好地将图形处理操作分配给处理集群阵列1712的多个集群1714A-1714N。在至少一个实施例中,处理集群阵列1712的部分可以配置成执行不同类型的处理。例如,在至少一个实施例中,第一部分可以配置成执行顶点着色和拓扑生成,第二部分可以配置成执行镶嵌和几何着色,并且第三部分可以配置成

执行像素着色或其他屏幕空间操作,以生成用于显示的渲染图像。在至少一个实施例中,可以将由集群1714A-1714N中的一个或更多个产生的中间数据存储在缓冲区中,以允许在集群1714A-1714N之间传输中间数据以进行进一步处理。

[0209] 在至少一个实施例中,处理集群阵列1712可以经由调度器1710接收要执行的处理任务,该调度器1710从前端1708接收定义处理任务的命令。在至少一个实施例中,处理任务可以包括要被处理的数据的索引,例如表面(补丁)数据、原始数据、顶点数据和/或像素数据,以及状态参数和定义如何处理数据的命令(例如,要执行什么程序)。在至少一个实施例中,调度器1710可以配置成获取与任务相对应的索引,或者可以从前端1708接收索引。在至少一个实施例中,前端1708可以配置成确保在启动由传入命令缓冲区(例如,批处理缓冲区、推入缓冲区等)指定的工作负载之前,处理集群阵列1712配置成有效状态。

[0210] 在至少一个实施例中,并行处理单元1702的一个或更多个实例中的每一个可以与并行处理器存储器1722耦合。在至少一个实施例中,可以经由存储器交叉开关1716访问并行处理器存储器1722,所述存储器交叉开关1716可以接收来自处理集群阵列1712以及I/O单元1704的存储器请求。在至少一个实施例中,存储器交叉开关1716可以经由存储器接口1718访问并行处理器存储器1722。在至少一个实施例中,存储器接口1718可以包括多个分区单元(例如,分区单元1720A、分区单元1720B、直到分区单元1720N),其可各自耦合至并行处理器存储器1722的一部分(例如,存储单元)。在至少一个实施例中,多个分区单元1720A-1720N为配置为等于存储单元的数量,使得第一分区单元1720A具有对应的第一存储单元1724A,第二分区单元1720B具有对应的存储单元1724B,第N分区单元1720N具有对应的第N存储单元1724N。在至少一个实施例中,分区单元1720A-1720N的数量可以不等于存储设备的数量。

[0211] 在至少一个实施例中,存储器单元1724A-1724N可以包括各种类型的存储器设备,包括动态随机存取存储器(DRAM)或图形随机存取存储器,例如同步图形随机存取存储器(SGRAM),包括图形双倍数据速率(GDDR)存储器。在至少一个实施例中,存储器单元1724A-1724N还可包括3D堆叠存储器,包括但不限于高带宽存储器(HBM)。在至少一个实施例中,可以跨存储器单元1724A-1724N来存储诸如帧缓冲区或纹理映射的渲染目标,从而允许分区单元1720A-1720N并行地写入每个渲染目标的部分,以有效地使用并行处理器存储器1722的可用带宽。在至少一个实施例中,可以排除并行处理器存储器1722的本地实例,以有利于利用系统存储器与本地高速缓存存储器结合的统一存储器设计。

[0212] 在至少一个实施例中,处理集群阵列1712的集群1714A-1714N中的任何一个都可以处理将被写入并行处理器存储器1722内的任何存储器单元1724A-1724N中的数据。在至少一个实施例中,存储器交叉开关1716可以配置为将每个集群1714A-1714N的输出传输到任何分区单元1720A-1720N或另一个集群1714A-1714N,集群1714A-1714N可以对输出执行其他处理操作。在至少一个实施例中,每个集群1714A-1714N可以通过存储器交叉开关1716与存储器接口1718通信,以从各种外部存储设备读取或写入各种外部存储设备。在至少一个实施例中,存储器交叉开关1716具有到存储器接口1718的连接以与I/O单元1704通信,以及到并行处理器存储器1722的本地实例的连接,从而使不同处理集群1714A-1714N内的处理单元与系统存储器或不是并行处理单元1702本地的其他存储器进行通信。在至少一个实施例中,存储器交叉开关1716可以使用虚拟通道来分离集群1714A-1714N和分区单元

1720A-1720N之间的业务流。

[0213] 在至少一个实施例中,可以在单个插入卡上提供并行处理单元1702的多个实例,或者可以将多个插入卡互连。在至少一个实施例中,并行处理单元1702的不同实例可以配置成相互操作,即使不同实例具有不同数量的处理核心,不同数量的本地并行处理器存储器和/或其他配置差异。例如,在至少一个实施例中,并行处理单元1702的一些实例可以包括相对于其他实例而言更高精度的浮点单元。在至少一个实施例中,结合并行处理单元1702或并行处理器1700的一个或更多个实例的系统可以以各种配置和形式因素来实现,包括但不限于台式机、膝上型计算机或手持式个人计算机、服务器、工作站、游戏机和/或嵌入式系统。

[0214] 图17B是根据至少一个实施例的分区单元1720的框图。在至少一个实施例中,分区单元1720是图17A的分区单元1720A-1720N之一的实例。在至少一个实施例中,分区单元1720包括L2高速缓存1721、帧缓冲区接口1725和光栅操作单元(“ROP”)1726。L2高速缓存1721是读/写高速缓存,其配置成执行从存储器交叉开关1716和ROP 1726接收的加载和存储操作。在至少一个实施例中,L2高速缓存1721将读取未命中和紧急回写请求输出到帧缓冲区接口1725以进行处理。在至少一个实施例中,还可以经由帧缓冲区接口1725将更新发送到帧缓冲区以进行处理。在至少一个实施例中,帧缓冲区接口1725与并行处理器存储器中的存储器单元(诸如图17B的存储器单元1724A-1724N(例如,在并行处理器存储器1722内))之一相互作用。

[0215] 在至少一个实施例中,ROP 1726是一种处理单元,其执行光栅操作,诸如模版、z测试、混合等。在至少一个实施例中,ROP 1726然后输出存储在图形存储器中的处理后的图形数据。在至少一个实施例中,ROP 1726包括压缩逻辑以压缩被写入存储器的深度或颜色数据并解压缩从存储器读取的深度或颜色数据。在至少一个实施例中,压缩逻辑可以是利用多种压缩算法中的一种或更多种的无损压缩逻辑。ROP 1726执行的压缩逻辑可以基于要压缩的数据的统计特性而变化。例如,在至少一个实施例中,基于每图块基础上的深度和颜色数据执行增量颜色压缩。

[0216] 在至少一个实施例中,ROP 1726包括在每个处理集群内(例如,图17A的集群1714A-1714N),而不是在分区单元1720内。在至少一个实施例中,通过存储器交叉开关1716而不是像素片段数据传输对像素数据的读取和写入请求。在至少一个实施例中,经处理的图形数据可以在显示设备上(诸如图16的一个或更多个显示设备1610之一)显示,由处理器1602路由以供进一步处理,或者由图17A的并行处理器1700内的处理实体之一路由以供进一步处理。

[0217] 图17C是根据至少一个实施例的并行处理单元内的处理集群1714的框图。在至少一个实施例中,处理集群是图17A的处理集群1714A-1714N之一的实例。在至少一个实施例中,一个或更多个处理集群1714中的一个或更多个可以配置成并行执行许多线程,其中“线程”是指在特定的一组输入数据上执行的特定程序的实例。在至少一个实施例中,单指令多数据(SIMD)指令发布技术用于支持大量线程的并行执行而无需提供多个独立的指令单元。在至少一个实施例中,使用单指令多线程(SIMT)技术来支持并行执行大量一般同步的线程,这使用了公共指令单元,该公共指令单元配置成向每个处理集群内的一组处理引擎发出指令。

[0218] 在至少一个实施例中,可以通过将处理任务分配给SIMT并行处理器的管线管理器1732来控制处理集群1714的操作。在至少一个实施例中,管线管理器1732从图17A的调度器1710接收指令,通过图形多处理器1734和/或纹理单元1736管理这些指令的执行。在至少一个实施例中,图形多处理器1734是SIMT并行处理器的示例性实例。然而,在至少一个实施例中,处理集群1714内可以包括不同架构的各种类型的SIMT并行处理器。在至少一个实施例中,在处理集群1714内可以包括图形多处理器1734的一个或更多个实例。在至少一个实施例中,图形多处理器1734可以处理数据,并且数据交叉开关1740可以用于将处理后的数据分发到多个可能的目的(包括其他着色器单元)地之一。在至少一个实施例中,管线管理器1732可以通过指定要针对数据交叉开关1740分配的处理数据的目的地来促进处理数据的分配。

[0219] 在至少一个实施例中,处理集群1714内的每个图形多处理器1734可以包括相同的一组功能执行逻辑(例如,算术逻辑单元、负载存储单元等)。在至少一个实施例中,可以以管线方式配置功能执行逻辑,其中可以在先前的指令完成之前发出新的指令。在至少一个实施例中,功能执行逻辑支持多种操作,包括整数和浮点算术、比较操作、布尔运算、移位和各种代数函数的计算。在至少一个实施例中,可以利用相同的功能单元硬件来执行不同的操作,并且可以存在功能单元的任何组合。

[0220] 在至少一个实施例中,传送到处理集群1714的指令构成线程。在至少一个实施例中,跨一组并行处理引擎执行的一组线程是线程组。在至少一个实施例中,线程组在不同的输入数据上执行程序。在至少一个实施例中,线程组内的每个线程可被分配给图形多处理器1734内的不同处理引擎。在至少一个实施例中,线程组可包括比图形多处理器1734内的多个处理引擎更少的线程。在至少一个实施例中,当线程组包括的线程数少于处理引擎的数量时,一个或更多个处理引擎在正在处理该线程组的循环期间可能是空闲的。在至少一个实施例中,线程组还可以包括比图形多处理器1734内的多个处理引擎更多的线程。在至少一个实施例中,当线程组包括比图形多处理器1734内的处理引擎更多的线程时,可以在连续的时钟周期内执行处理。在至少一个实施例中,可以在图形多处理器1734上同时执行多个线程组。

[0221] 在至少一个实施例中,图形多处理器1734包括内部高速缓冲存储器,以执行加载和存储操作。在至少一个实施例中,图形多处理器1734可以放弃内部高速缓存并使用处理集群1714内的高速缓存存储器(例如,L1高速缓存1748)。在至少一个实施例中,每个图形多处理器1734还可以访问分区单元(例如,图17A的分区单元1720A-1720N)内的L2高速缓存,这些分区单元在所有处理集群1714之间共享并且可以用于在线程之间传输数据。在至少一个实施例中,图形多处理器1734还可以访问片外全局存储器,其可以包括本地并行处理器存储器和/或系统存储器中的一个或更多个。在至少一个实施例中,并行处理单元1702外部的任何存储器都可以用作全局存储器。在至少一个实施例中,处理集群1714包括图形多处理器1734的多个实例,它们可以共享可以存储在L1高速缓存1748中的公共指令和数据。

[0222] 在至少一个实施例中,每个处理集群1714可以包括配置成将虚拟地址映射为物理地址的存储器管理单元(“MMU”)1745。在至少一个实施例中,MMU 1745的一个或更多个实例可以驻留在图17A的存储器接口1718内。在至少一个实施例中,MMU 1745包括一组页表条目(PTE),其用于将虚拟地址映射到图块的物理地址以及可选地映射到高速缓冲存储器线。在

至少一个实施例中,MMU 1745可以包括地址转换后备缓冲区(TLB)或可以驻留在图形多处理器1734或L1高速缓存或处理集群1714内的高速缓存。在至少一个实施例中,处理物理地址以分配表面数据访问局部性,以便在分区单元之间进行有效的请求交织。在至少一个实施例中,高速缓存行索引可以用于确定对高速缓存线的请求是命中还是未命中。

[0223] 在至少一个实施例中,可以配置处理集群1714,使得每个图形多处理器1734耦合到纹理单元1736,以执行纹理映射操作,例如,确定纹理样本位置、读取纹理数据以及过滤纹理数据。在至少一个实施例中,根据需要从内部纹理L1高速缓存(未示出)或从图形多处理器1734内的L1高速缓存中读取纹理数据,并从L2高速缓存、本地并行处理器存储器或系统存储器中获取纹理数据。在至少一个实施例中,每个图形多处理器1734将处理后的任务输出到数据交叉开关1740,以将处理后的任务提供给另一处理集群1714以进行进一步处理或将处理后的任务存储在L2高速缓存、本地并行处理器存储器、或经由存储器交叉开关1716的系统存储器中。在至少一个实施例中,preROP1742(光栅前操作单元)配置成从图形多处理器1734接收数据,将数据引导至ROP单元,该ROP单元可以与本文所述的分区单元(例如,图17A的分区单元1720A-1720N)一起定位。在至少一个实施例中,PreROP1742单元可以执行用于颜色混合的优化、组织像素颜色数据以及执行地址转换。

[0224] 推理和/或训练逻辑615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。下面结合图6A和/或6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例中,推理和/或训练逻辑615可以在图形处理集群1714中用于至少部分地基于使用本文描述的神经网络训练操作、神经网络功能和/或架构或神经网络用例计算的权重参数来进行推理或预测操作。

[0225] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0226] 图17D示出了根据至少一个实施例的图形多处理器1734。在至少一个实施例中,图形多处理器1734与处理集群1714的管线管理器1732耦合。在至少一个实施例中,图形多处理器1734具有执行管线,该执行管线包括但不限于指令高速缓存1752、指令单元1754、地址映射单元1756、寄存器文件1758、一个或更多个通用图形处理单元(GPGPU)核心1762和一个或更多个加载/存储单元1766。GPGPU核心1762和加载/存储单元1766与高速缓存存储器1772和共享存储器1770通过存储器和高速缓存互连1768耦合。

[0227] 在至少一个实施例中,指令高速缓存1752从管线管理器1732接收要执行的指令流。在至少一个实施例中,将指令高速缓存在指令高速缓存1752中并将其分派以供指令单元1754执行。在一个实施例中,指令单元1754可以分派指令作为线程组(例如,线程束),将每个线程组分配给GPGPU核心1762内的不同执行单元。在至少一个实施例中,指令可以通过在统一地址空间内指定地址来访问任何本地、共享或全局地址空间。在至少一个实施例中,地址映射单元1756可以用于将统一地址空间中的地址转换成可以由加载/存储单元1766访问的不同的存储器地址。

[0228] 在至少一个实施例中,寄存器文件1758为图形多处理器1734的功能单元提供了一组寄存器。在至少一个实施例中,寄存器文件1758为连接到图形多处理器1734的功能单元(例如,GPGPU核心1762、加载/存储单元1766)的数据路径的操作数提供了临时存储。在至少

一个实施例中,在每个功能单元之间划分寄存器文件1758,使得为每个功能单元分配寄存器文件1758的专用部分。在至少一个实施例中,寄存器文件1758在图形多处理器1734正在执行的不同线程束之间划分。

[0229] 在至少一个实施例中,GPGPU核心1762可以各自包括用于执行图形多处理器1734的指令的浮点单元(FPU)和/或整数算术逻辑单元(ALU)。GPGPU核心1762在架构上可以相似或架构可能有所不同。在至少一个实施例中,GPGPU核心1762的第一部分包括单精度FPU和整数ALU,而GPGPU核心的第二部分包括双精度FPU。在至少一个实施例中,FPU可以实现用于浮点算法的IEEE 754-2008标准或启用可变精度浮点算法。在至少一个实施例中,图形多处理器1734可以另外包括一个或更多个固定功能或特殊功能单元,以执行特定功能,诸如复制矩形或像素混合操作。在至少一个实施例中,GPGPU核心中的一个或更多个也可以包括固定或特殊功能逻辑。

[0230] 在至少一个实施例中,GPGPU核心1762包括能够对多组数据执行单个指令的SIMD逻辑。在至少一个实施例中,GPGPU核心1762可以物理地执行SIMD4、SIMD8和SIMD16指令,并且在逻辑上执行SIMD1、SIMD2和SIMD32指令。在至少一个实施例中,用于GPGPU核心的SIMD指令可以在编译时由着色器编译器生成,或者在执行针对单程序多数据(SPMD)或SIMT架构编写和编译的程序时自动生成。在至少一个实施例中,可以通过单个SIMD指令来执行为SIMT执行模型配置的程序的多个线程。例如,在至少一个实施例中,可以通过单个SIMD8逻辑单元并行执行执行相同或相似操作的八个SIMT线程。

[0231] 在至少一个实施例中,存储器和高速缓存互连1768是将图形多处理器1734的每个功能单元连接到寄存器文件1758和共享存储器1770的互连网络。在至少一个实施例中,存储器和高速缓存互连1768是交叉开关互连,其允许加载/存储单元1766在共享存储器1770和寄存器文件1758之间实现加载和存储操作。在至少一个实施例中,寄存器文件1758可以以与GPGPU核心1762相同的频率操作,从而在GPGPU核心1762和寄存器文件1758之间进行数据传输的延迟非常低。在至少一个实施例中,共享存储器1770可以用于启用在图形多处理器1734内的功能单元上执行的线程之间的通信。在至少一个实施例中,高速缓存存储器1772可以用作例如数据高速缓存,以高速缓存在功能单元和纹理单元1736之间通信的纹理数据。在至少一个实施例中,共享存储器1770也可以用作程序管理的高速缓存。在至少一个实施例中,除了存储在高速缓存存储器1772中的自动高速缓存的数据之外,在GPGPU核心1762上执行的线程还可以以编程方式将数据存储在共享存储器中。

[0232] 在至少一个实施例中,如本文所述的并行处理器或GPGPU通信地耦合到主机/处理器核心,以加速图形操作、机器学习操作、图案分析操作以及各种通用GPU(GPGPU)功能。在至少一个实施例中,GPU可以通过总线或其他互连(例如,诸如PCIe或NVLink的高速互连)通信地耦合到主机处理器/核心。在至少一个实施例中,GPU可以与核心集成在相同的封装或芯片上,并通过内部处理器总线/互连(即,封装或芯片的内部)通信地耦合到核心。在至少一个实施例中,不管GPU连接的方式如何,处理器核心可以以工作描述符中包含的命令/指令序列的形式向GPU分配工作。在至少一个实施例中,GPU然后使用专用电路/逻辑来有效地处理这些命令/指令。

[0233] 推理和/或训练逻辑615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。下面结合图6A和/或图6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例

中,推理和/或训练逻辑615可以在图形多处理器1734中用于至少部分地基于使用本文描述的神经网络训练操作、神经网络功能和/或架构或神经网络用例计算的权重参数来进行推理或预测操作,。

[0234] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0235] 图18示出了根据至少一个实施例的多GPU计算系统1800。在至少一个实施例中,多GPU计算系统1800可包括经由主机接口开关1804耦合到多个通用图形处理单元(GPGPU)1806A-D的处理器1802。在至少一个实施例中,主机接口开关1804是PCI高速开关设备,其将处理器1802耦合到PCI高速总线,处理器1802可以通过PCI高速总线与GPGPU1806A-D通信。GPGPU 1806A-D可以经由一组高速点对点GPU到GPU链路1816来互连。在至少一个实施例中,GPU到GPU链路1816经由专用GPU链路连接到GPGPU 1806A-D中的每一个。在至少一个实施例中,P2P GPU链路1816使得能够在每个GPGPU 1806A-D之间直接进行通信,而无需通过处理器1802所连接的主机接口总线1804进行通信。在至少一个实施例中,在GPU到GPU业务流指向P2P GPU链路1816的情况下,主机接口总线1804保持可用于系统存储器访问或者例如经由一个或更多个网络设备与多GPU计算系统1800的其他实例通信。虽然在至少一个实施例中,GPGPU 1806A-D经由主机接口开关1804连接到处理器1802,但是在至少一个实施例中,处理器1802包括对P2P GPU链路1816的直接支持,并且可以直接连接到GPGPU 1806A-D。

[0236] 推理和/或训练逻辑615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。下面结合图6A和/或图6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例中,推理和/或训练逻辑615可以在多GPU计算系统1800中用于至少部分地基于使用本文所述的神经网络训练操作,神经网络功能和/或或架构或神经网络用例计算出的权重参数来进行推理或预测操作。

[0237] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0238] 图19是根据至少一个实施例的图形处理器1900的框图。在至少一个实施例中,图形处理器1900包括环形互连1902、管线前端1904、媒体引擎1937和图形核心1980A-1980N。在至少一个实施例中,环形互连1902将图形处理器1900耦合到其他处理单元,包括其他图形处理器或一个或更多个通用处理器核心。在至少一个实施例中,图形处理器1900是集成在多核心处理系统内的许多处理器之一。

[0239] 在至少一个实施例中,图形处理器1900经由环形互连1902接收多批命令。在至少一个实施例中,输入命令由管线前端1904中的命令流转化器1903解释。在至少一个实施例中,图形处理器1900包括可缩放执行逻辑,以经由图形核心1980A-1980N执行3D几何处理和媒体处理。在至少一个实施例中,对于3D几何处理命令,命令流转化器1903将命令提供给几何管线1936。在至少一个实施例中,对于至少一些媒体处理命令,命令流转化器1903将命令提供给视频前端1934,其与媒体引擎1937耦合。在至少一个实施例中,媒体引擎1937包括用于视频和图像后处理的视频质量引擎(VQE)1930,以及用于提供硬件加速媒体数据编码和解码的多格式编码/解码(MFX)1933引擎。在至少一个实施例中,几何管线1936和媒体引擎

1937各自生成用于由至少一个图形核心1980A提供的线程执行资源的执行线程。

[0240] 在至少一个实施例中,图形处理器1900包括以模块化核心1980A-1980N(有时称为核心切片)为特征的可缩放线程执行资源,每个模块核心具有多个子核心1950A-1950N、1960A-1960N(有时称为核心子切片)。在至少一个实施例中,图形处理器1900可以具有任意数量的图形核心1980A至1980N。在至少一个实施例中,图形处理器1900包括具有至少第一子核心1950A和第二子核心1960A的图形核心1980A。在至少一个实施例中,图形处理器1900是具有单个子核心(例如1950A)的低功率处理器。在至少一个实施例中,图形处理器1900包括多个图形核心1980A-1980N,每个图形核心包括一组第一子核心1950A-1950N和一组第二子核心1960A-1960N。在至少一个实施例中,第一子核心1950A-1950N中的每个子核心至少包括第一组执行单元1952A-1952N和媒体/纹理采样器1954A-1954N。在至少一个实施例中,第二子核心1960A-1960N中的每个子核心至少包括第二组执行单元1962A-1962N和采样器1964A-1964N。在至少一个实施例中,每个子核心1950A-1950N、1960A-1960N共享一组共享资源1970A-1970N。在至少一个实施例中,共享资源包括共享高速缓冲存储器和像素操作逻辑。

[0241] 推理和/或训练逻辑615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。下面结合图6A和/或图6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例中,推理和/或训练逻辑615可以在图形处理器1900中用于至少部分地基于使用本文描述的神经网络训练操作、神经网络功能和/或架构或神经网络用例计算的权重参数来推理或预测操作。

[0242] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0243] 图20是示出根据至少一个实施例的用于处理器2000的微架构的框图,该处理器2000可以包括执行指令的逻辑电路。在至少一个实施例中,处理器2000可以执行指令,包括x86指令、ARM指令、用于专用集成电路(ASIC)的专用指令等。在至少一个实施例中,处理器2000可以包括用于存储封装数据的寄存器,例如作为加利福尼亚州圣克拉拉市英特尔公司采用MMX技术启用的微处理器中的64位宽MMX™寄存器。在至少一个实施例中,整数和浮点数形式可用的MMX寄存器可以与封装的数据元素一起运行,所述封装的数据元素伴随单指令多数据(“SIMD”)和流式SIMD扩展(“SSE”)指令。在至少一个实施例中,与SSE2、SSE3、SSE4、AVX或更高版本(一般称为“SSEx”)技术有关的128位宽XMM寄存器可以保存此类封装数据操作数。在至少一个实施例中,处理器2000可以执行指令以加速机器学习或深度学习算法、训练或推理。

[0244] 在至少一个实施例中,处理器2000包括有序前端(“前端”)2001,以提取要执行的指令并准备稍后在处理器管线中使用的指令。在至少一个实施例中,前端2001可以包括几个单元。在至少一个实施例中,指令预取器2026从存储器中获取指令并将指令提供给指令解码器2028,指令解码器2028又对指令进行解码或解释。例如,在至少一个实施例中,指令解码器2028将接收到的指令解码为机器可执行的所谓的“微指令”或“微操作”(也称为“微操作”或“微指令”)的一个或更多个操作。在至少一个实施例中,指令解码器2028将指令解析为操作码以及相应的数据和控制字段,其可以由微架构用来使用以根据至少一个实施例

来执行操作。在至少一个实施例中,跟踪高速缓存2030可以将解码的微指令组装成微指令队列2034中的程序排序的序列或追踪以供执行。在至少一个实施例中,当追踪高速缓存2030遇到复杂指令时,微码ROM 2032提供完成操作所需的微指令。

[0245] 在至少一个实施例中,可以将一些指令转换成单个微操作,而另一些指令则需要几个微操作来完成全部操作。在至少一个实施例中,如果需要多于四个的微指令来完成一条指令,则指令解码器2028可以访问微码ROM 2032以执行指令。在至少一个实施例中,可以将指令解码为少量的微指令以在指令解码器2028处进行处理。在至少一个实施例中,如果需要多个微指令完成操作,则可以将指令存储在微码ROM 2032中。在至少一个实施例中,追踪高速缓存器2030参考入口点可编程逻辑阵列(“PLA”)以确定正确的微指令指针,用于根据至少一个实施例从微码ROM 2032读取微码序列以完成一个或更多个指令。在至少一个实施例中,在微码ROM 2032完成对指令的微操作排序之后,机器的前端2001可以恢复从追踪高速缓存2030获取微操作。

[0246] 在至少一个实施例中,乱序执行引擎(“乱序引擎”)2003可以准备用于执行的指令。在至少一个实施例中,乱序执行逻辑具有多个缓冲区,以使指令流平滑并重新排序,以在指令沿管线下降并被调度执行时优化性能。在至少一个实施例中,乱序执行引擎2003包括但不限于分配器/寄存器重命名器2040、存储器微指令队列2042、整数/浮点微指令队列2044、存储器调度器2046、快速调度器2002、慢速/通用浮点调度器(“慢速/通用FP调度器”)2004和简单浮点调度器(“简单FP调度器”)2006。在至少一个实施例中,快速调度器2002、慢速/通用浮点调度器2004和简单浮点调度器2006也统称为“微指令调度器2002、2004、2006”。在至少一个实施例中,分配器/寄存器重命名器2040分配每个微指令按顺序执行所需要的机器缓冲区和资源。在至少一个实施例中,分配器/寄存器重命名器2040将逻辑寄存器重命名为寄存器文件中的条目。在至少一个实施例中,分配器/寄存器重命名器2040还为两个微指令队列之一中的每个微指令分配条目,存储器微指令队列2042用于存储器操作和整数/浮点微指令队列2044用于非存储器操作,在存储器调度器2046和微指令调度器2002、2004、2006的前面。在至少一个实施例中,微指令调度器2002、2004、2006基于它们的从属输入寄存器操作数源的就绪性和需要完成的执行资源微指令的可用性来确定何时准备好执行微指令。在至少一个实施例中,至少一个实施例的快速调度器2002可以在主时钟周期的每个一半上调度,而慢速/通用浮点调度器2004和简单浮点调度器2006可以在每个主处理器时钟周期调度一次。在至少一个实施例中,微指令调度器2002、2004、2006对调度端口进行仲裁,以调度用于执行的微指令。

[0247] 在至少一个实施例中,执行块2011包括但不限于整数寄存器文件/支路网络2008、浮点寄存器文件/支路网络(“FP寄存器文件/支路网络”)2010、地址生成单元(“AGU”)2012和2014、快速算术逻辑单元(“快速ALU”)2016和2018、慢速算术逻辑单元(“慢速ALU”)2020、浮点ALU(“FP”)2022和浮点移动单元(“FP移动”)2024。在至少一个实施例中,整数寄存器文件/支路网络2008和浮点寄存器文件/旁路网络2010在本文中也称为“寄存器文件2008、2010”。在至少一个实施例中,AGU 2012和2014、快速ALU 2016和2018、慢速ALU 2020、浮点ALU 2022和浮点移动单元2024在本文中也称为“执行单元2012、2014、2016、2018、2020、2022和2024”。在至少一个实施例中,执行框b11可以包括但不限于任意数量(包括零)和类型的寄存器文件、支路网络、地址生成单元和执行单元(以任何组合)。

[0248] 在至少一个实施例中,寄存器文件2008、2010可以布置在微指令调度器2002、2004、2006与执行单元2012、2014、2016、2018、2020、2022和2024之间。在至少一个实施例中,整数寄存器文件/支路网络2008执行整数运算。在至少一个实施例中,浮点寄存器文件/支路网络2010执行浮点操作。在至少一个实施例中,寄存器文件2008、2010中的每一个可以包括但不限于支路网络,该支路网络可以绕过或转发尚未写入寄存器文件中的刚刚完成的结果到新的从属对象。在至少一个实施例中,寄存器文件2008、2010可以彼此通信数据。在至少一个实施例中,整数寄存器文件/支路网络2008可以包括但不限于两个单独的寄存器文件、一个寄存器文件用于低阶32位数据,第二寄存器文件用于高阶32位数据。在至少一个实施例中,浮点寄存器文件/支路网络2010可以包括但不限于128位宽的条目,因为浮点指令通常具有宽度为64至128位的操作数。

[0249] 在至少一个实施例中,执行单元2012、2014、2016、2018、2020、2022、2024可以执行指令。在至少一个实施例中,寄存器文件2008、2010存储微指令需要执行的整数和浮点数据操作数值。在至少一个实施例中,处理器2000可以包括但不限于任何数量的执行单元2012、2014、2016、2018、2020、2022、2024及其组合。在至少一个实施例中,浮点ALU 2022和浮点移动单元2024,可以执行浮点、MMX、SIMD、AVX和SSE或其他操作,包括专门的机器学习指令。在至少一个实施例中,浮点ALU 2022可以包括但不限于64位乘64位浮点除法器,以执行除法、平方根和余数微操作。在至少一个实施例中,可以用浮点硬件来处理涉及浮点值的指令。在至少一个实施例中,可以将ALU操作传递给快速ALU 2016、2018。在至少一个实施例中,快速ALUS 2016、2018可以以半个时钟周期的有效延迟执行快速操作。在至少一个实施例中,大多数复杂的整数运算进入慢速ALU 2020,因为慢速ALU 2020可以包括但不限于用于长延迟类型操作的整数执行硬件,例如乘法器、移位、标志逻辑和分支处理。在至少一个实施例中,存储器加载/存储操作可以由AGUS 2012、2014执行。在至少一个实施例中,快速ALU 2016、快速ALU 2018和慢速ALU 2020可以对64位数据操作数执行整数运算。在至少一个实施例中,可以实现快速ALU2016、快速ALU 2018和慢速ALU 2020以支持包括十六、三十二、128、256等的各种数据位大小。在至少一个实施例中,浮点ALU 2022和浮点移动单元2024可以实现为支持具有各种宽度的位的一定范围的操作数。在至少一个实施例中,浮点ALU 2022和浮点移动单元2024可以结合SIMD和多媒体指令对128位宽封装数据操作数进行操作。

[0250] 在至少一个实施例中,微指令调度器2002、2004、2006在父加载完成执行之前调度从属操作。在至少一个实施例中,由于可以在处理器2000中推测性地调度和执行微指令,处理器2000还可以包括用于处理存储器未命中的逻辑。在至少一个实施例中,如果数据高速缓存中的数据加载未命中,则可能存在在管线中正在运行的从属操作,其使调度器暂时没有正确的数据。在至少一个实施例中,一种重放机制追踪踪并重新执行使用不正确数据的指令。在至少一个实施例中,可能需要重放从属操作并且可以允许完成独立操作。在至少一个实施例中,处理器的至少一个实施例的调度器和重放机制也可以设计为捕获用于文本串比较操作的指令序列。

[0251] 在至少一个实施例中,术语“寄存器”可以指代可以用作识别操作数的指令的一部分的机载处理器存储位置。在至少一个实施例中,寄存器可以是那些可以从处理器外部使用的寄存器(从程序员的角度来看)。在至少一个实施例中,寄存器可能不限于特定类型的电路。相反,在至少一个实施例中,寄存器可以存储数据、提供数据并执行本文描述的功能。

在至少一个实施例中,本文描述的寄存器可以通过处理器内的电路使用多种不同技术来实现,例如专用物理寄存器、使用寄存器重命名动态分配的物理寄存器、专用和动态分配的物理寄存器的组合等。在至少一个实施例中,整数寄存器存储32位整数数据。至少一个实施例的寄存器文件还包含八个用于封装数据的多媒体SIMD寄存器。

[0252] 推理和/或训练逻辑615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。下面结合图6A和/或图6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例中,可以将推理和/或训练逻辑615的部分或全部并入执行块2011以及示出或未示出的其他存储器或寄存器。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执行框2011中示出的一个或更多个ALU。此外,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,该寄存器和/或寄存器配置执行块2011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。

[0253] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0254] 图21示出了根据至少一个实施例的深度学习应用处理器2100。在至少一个实施例中,深度学习应用处理器2100使用指令,如果由深度学习应用处理器2100执行,则指令使深度学习应用处理器2100执行贯穿本公开描述的一些或全部过程和技术。在至少一个实施例中,深度学习应用处理器2100是专用集成电路(ASIC)。在至少一个实施例中,应用处理器2100执行矩阵乘法运算或者“硬连线”到硬件中,作为执行一个或更多个指令或两者的结果。在至少一个实施例中,深度学习应用处理器2100包括但不限于处理集群2110(1)-2110(12)、芯片间链路(“ICL”)2120(1)-2120(12)、芯片间控制器(“ICC”)2130(1)-2130(2)、存储器控制器(“Mem Ctrlrs”)2142(1)-2142(4)、高带宽存储器物理层(“HBM PHY”)2144(1)-2144(4)、管理控制器中央处理单元(“管理控制器CPU”)2150、串行外围设备接口、内部集成电路和通用输入/输出块(“SPI、I2C、GPIO”),外围组件互连快速控制器和直接存储器访问块(“PCIe控制器和DMA”)2170、以及十六通道外围组件互连快速端口(“PCI Express x16”)2180。

[0255] 在至少一个实施例中,处理集群2110可以执行深度学习操作,包括基于一种或更多种训练技术计算的权重参数的推理或预测操作,包括本文所述的那些技术。在至少一个实施例中,每个处理集群2110可以包括但不限于任何数量和类型的处理器。在至少一个实施例中,深度学习应用处理器2100可以包括任何数量和类型的处理集群2100。在至少一个实施例中,芯片间链路2120是双向的。在至少一个实施例中,芯片间链路2120和芯片间控制器2130使多个深度学习应用处理器2100能够交换信息,包括从执行一个或更多个神经网络中体现的一种或更多种机器学习算法而产生的激活信息。在至少一个实施例中,深度学习应用处理器2100可以包括任意数量(包括零)和类型的ICL 2120和ICC 2130。

[0256] 在至少一个实施例中,HBM2 2140提供总共32GB的存储器。HBM2 2140(i)与存储器控制器2142(i)和HBM PHY 2144(i)都相关联。在至少一个实施例中,任何数量的HBM2 2140可以提供任何类型和总量的高带宽存储器,并且可以与任何数量(包括零)和类型的存储器控制器2142和HBM PHY 2144相关联。在至少一个实施例中,可以用任何数量和类型的块替换SPI、I2C、GPIO 2160、PCIe控制器和DMA 2170和/或PCIe2180,以任何技术上可行的方式

实现任何数量和类型的通信标准。

[0257] 推理和/或训练逻辑615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。下面结合图6A和/或6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例中,深度学习应用处理器2100用于训练机器学习模型(例如神经网络),以预测或推理提供给深度学习应用处理器2100的信息。在至少一个实施例中,深度学习应用处理器2100用于基于已经由另一处理器或系统或由深度学习应用处理器2100训练的经训练的机器学习模型(例如,神经网络)来推理或预测信息。在至少一个实施例中,处理器2100可以用于执行本文所述的一个或更多个神经网络用例。

[0258] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0259] 图22是根据至少一个实施例的神经形态处理器2200的框图。在至少一个实施例中,神经形态处理器2200可以从神经形态处理器2200外部的源接收一个或更多个输入。在至少一个实施例中,这些输入可以被传输到神经形态处理器2200内的一个或更多个神经元2202。在至少一个实施例中,可以使用包括一个或更多个算术逻辑单元(ALU)的电路或逻辑来实现神经元2202及其组件。在至少一个实施例中,神经形态处理器2200可以包括但不限于成千上万个神经元2202的实例,但是可以使用任何合适数量的神经元2202。在至少一个实施例中,神经元2202的每个实例可以包括神经元输入2204和神经元输出2206。在至少一个实施例中,神经元2202可以生成可以传输到神经元2202的其他实例的输入的输出。在至少一个实施例中,神经元输入2204和神经元输出2206可以经由突触2208互连。

[0260] 在至少一个实施例中,神经元2202和突触2208可以互连,使得神经形态处理器2200操作以处理或分析由神经形态处理器2200接收的信息。在至少一个实施例中,当通过神经元输入2204接收到的输入超过阈值时,神经元2202可以发送输出脉冲(或“触发”或“峰值”)。在至少一个实施例中,神经元2202可以对在神经元输入2204处接收到的信号进行求和或积分。例如,在至少一个实施例中,神经元2202可以实现为有泄漏的积分-触发神经元,其中如果求和(称为“膜电位”)超过阈值,则神经元2202可以使用诸如sigmoid或阈值函数的传递函数来产生输出(或“触发”)。在至少一个实施例中,泄漏的积分-触发神经元可以将在神经元输入2204处接收到的信号求和成膜电位,并且可以应用衰减因子(或泄漏)以减小膜电位。在至少一个实施例中,如果在神经元输入2204处接收到足够快以超过阈值的多个输入信号(即,在膜电势衰减得太低而不能触发之前),则泄漏的积分-触发神经元可能会触发。在至少一个实施例中,神经元2202可以使用接收输入、将输入积分到膜电位、并衰减膜电位的电路或逻辑来实现。在至少一个实施例中,可以对输入求平均,或者可以使用任何其他合适的传递函数。此外,在至少一个实施例中,神经元2202可以包括但不限于当将传递函数应用于神经元输入2204的结果超过阈值时在神经元输出2206处产生输出尖峰的比较器电路或逻辑。在至少一个实施例中,一旦神经元2202触发,它可以通过例如将膜电位复位为0或另一合适的默认值来忽略先前接收的输入信息。在至少一个实施例中,一旦膜电位被重置为0,则神经元2202可以在合适的时间段(或修复期)之后恢复正常操作。

[0261] 在至少一个实施例中,神经元2202可以通过突触2208互连。在至少一个实施例中,突触2208可以操作以将从第一神经元2202的输出的信号传输到第二神经元2202的输入。在

至少一个实施例中,神经元2202可以在一个以上的突触2208实例上传输信息。在至少一个实施例中,神经元输出2206的一个或更多个实例可以通过突触2208的实例连接到同一神经元2202中神经元输入2204的实例。在至少一个实施例中,相对于突触2208的那个实例,神经元2202的实例产生要在突触2208的实例上传输的输出可以被称为“突触前神经元”。在至少一个实施例中,相对于突触2208的实例,神经元2202的实例接收通过突触2208的实例传输的输入可以被称为“突触后神经元”。在至少一个实施例中,关于突触2208的各种实例,因为神经元2202的实例可以接收来自一个或更多个突触2208实例的输入,并且还可以通过一个或更多个突触2208实例传输输出,因此神经元2202的单个实例可以既是“突触前神经元”又是“突触后神经元”。

[0262] 在至少一个实施例中,神经元2202可以被组织成一层或更多层。神经元2202的每个实例可以具有一个神经元输出2206,该神经元输出2206可以通过一个或更多个突触2208扇出到一个或更多个神经元输入2204。在至少一个实施例中,第一层2210中的神经元2202的神经元输出2206可以连接到第二层2212中的神经元2202的神经元输入2204。在至少一个实施例中,层2210可以被称为“前馈层”。在至少一个实施例中,在第一层2210的实例中神经元2202的每个实例可以扇出到第二层2212中的神经元2202的每个实例。在至少一个实施例中,第一层2210可以被称为“完全连接的前馈层”。在至少一个实施例中,在第二层2212的每个实例中的神经元2202的每个实例扇出到少于在第三层2214中的神经元2202的所有实例。在至少一个实施例中,第二层2212可以被称为“稀疏连接的前馈层”。在至少一个实施例中,第二层2212中的神经元2202可以扇出到多个其他层中的神经元2202,包括(相同)第二层2212中的神经元2202。在至少一个实施例中,第二层2212可以被称为“循环层”。在至少一个实施例中,神经形态处理器2200可以包括但不限于循环层和前馈层的任何合适的组合,包括但不限于稀疏连接的前馈层和完全连接的前馈层。

[0263] 在至少一个实施例中,神经形态处理器2200可以包括但不限于可重新配置的互连架构或专用硬连线互连,以将突触2208连接到神经元2202。在至少一个实施例中,神经形态处理器2200可以包括但不限于电路或逻辑,其根据神经网络拓扑结构和神经元扇入/扇出,允许根据需要将突触分配给不同神经元2202。例如,在至少一个实施例中,可以使用互连结构(诸如片上网络)或通过专用连接将突触2208连接到神经元2202。在至少一个实施例中,可以使用电路或逻辑来实现突触互连及其组件。

[0264] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0265] 图23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2300包括一个或更多个处理器2302和一个或更多个图形处理器2308,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2302或处理器核心2307的服务器系统。在至少一个实施例中,系统2300是结合在片上系统(SoC)集成电路内的处理平台,以用于移动、手持或嵌入式设备。

[0266] 在至少一个实施例中,系统2300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2300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2300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2300是电视或机顶盒设备,其具有一个或更多个处理器2302以及由一个或更多个图形处理器2308生成的图形界面。

[0267] 在至少一个实施例中,一个或更多个处理器2302每个包括一个或更多个处理器核心2307,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2307中的每一个被配置为处理特定指令集2309。在至少一个实施例中,指令集2309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,多个处理器核心2307可以各自处理不同的指令集2309,该指令集2309可以包括有助于仿真其他指令集的指令。在至少一个实施例中,处理器核心2307还可以包括其他处理设备,例如数字信号处理器(DSP)。

[0268] 在至少一个实施例中,处理器2302包括高速缓存存储器2304。在至少一个实施例中,处理器2302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2302的各个组件之间共享。在至少一个实施例中,处理器2302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),其可以使用已知的高速缓存相干技术在处理器核心2307之间共享该逻辑。在至少一个实施例中,处理器2302中另外包括寄存器文件2306,处理器2302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2306可以包括通用寄存器或其他寄存器。

[0269] 在至少一个实施例中,一个或更多个处理器2302与一个或更多个接口总线2310耦合,以在处理器2302与系统2300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2310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2310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,PCI Express)、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2302包括集成存储器控制器2316和平台控制器集线器2330。在至少一个实施例中,存储器控制器2316促进存储设备与系统2300的其他组件之间的通信,而平台控制器集线器(PCH)2330通过本地I/O总线提供到I/O设备的连接。

[0270] 在至少一个实施例中,存储设备2320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过程存储器。在至少一个实施例中,存储设备2320可以用作系统2300的系统存储器,以存储数据2322和指令2321,以在一个或更多个处理器2302执行应用或过程时使用。在至少一个实施例中,存储器控制器2316还与可选的外部图形处理器2312耦合,其可以与处理器2302中的一个或更多个图形处理器2308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2311可以连接至处理器2302。在至少一个实施例中,显示设备2311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便携式计算机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。在至少一个实施例中,显示设备2311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。

[0271] 在至少一个实施例中,平台控制器集线器2330使外围设备能够通过高速I/O总线

连接到存储设备2320和处理器2302。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2346、网络控制器2322、固件接口2328、无线收发器2326、触摸传感器2325、数据存储设备2324(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2324可以经由存储器接口(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、PCI Express)。在至少一个实施例中,触摸传感器2325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2326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2328使能与系统固件的通信,并且可以是例如统一的可扩展固件接口(UEFI)。在至少一个实施例中,网络控制器2334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2310耦合。在至少一个实施例中,音频控制器2346是多通道高清晰度音频控制器。在至少一个实施例中,系统2300包括可选的遗留I/O控制器2340,用于将遗留(例如,个人系统2(PS/2))设备耦合到系统。在至少一个实施例中,平台控制器集线器2330还可以连接到一个或更多个通用串行总线(USB)控制器2342,该控制器连接输入设备,诸如键盘和鼠标2343组合、相机2344或其他USB输入设备。

[0272] 在至少一个实施例中,存储器控制器2316和平台控制器集线器2330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2312。在至少一个实施例中,平台控制器集线器2330和/或存储控制器2316可以在一个或更多个处理器2302的外部。例如,在至少一个实施例中,系统2300可以包括外部存储控制器2316和平台控制器集线器2330,其可以配置成在与处理器2302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。

[0273] 推理和/或训练逻辑615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。下面结合图6A和/或6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例中,推理和/或训练逻辑615的部分或全部可以被合并到图形处理器2300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种或更多种体现在图形处理器2312中的ALU。此外,在至少一个实施例中,可以使用除图6A或图6B所示的逻辑以外的逻辑来完成本文所述的推理和/或训练操作。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2300的ALU以执行一种或更本文描述的多种机器学习算法、神经网络架构、用例或训练技术。

[0274] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0275] 图24是根据至少一个实施例的具有一个或更多个处理器核心2402A-2402N、集成存储器控制器2414和集成图形处理器2408的处理器2400的框图。在至少一个实施例中,处理器2400可以包括直至并包括由虚线框表示的附加核心2402N的附加核心。在至少一个实施例中,每个处理器核心2402A-2402N包括一个或更多个内部高速缓存单元2404A-2404N。在至少一个实施例中,每个处理器核心还可以访问一个或更多个共享高速缓存的单元2406。

[0276] 在至少一个实施例中,内部高速缓存单元2404A-2404N和共享高速缓存单元2406表示处理器2400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2404A-2404N可以包括每个处理器核心内的至少一级指令和数据以及共享中级缓存中的

一级或更多级缓存,例如2级(L2)、3级(L3)、4级(L4)或其他级别的缓存,其中在外部存储器之前将最高级别的缓存归类为LLC。在至少一个实施例中,高速缓存相干逻辑维持各种高速缓存单元2406和2404A-2404N之间的相干。

[0277] 在至少一个实施例中,处理器2400还可包括一组一个或更多个总线控制器单元2416和系统代理核心2410。在至少一个实施例中,一个或更多个总线控制器单元2416管理一组外围总线,例如一个或更多个PCI或PCI Express总线。在至少一个实施例中,系统代理核心2410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2410包括一个或更多个集成存储器控制器2414,以管理对各种外部存储器设备(未示出)的访问。

[0278] 在至少一个实施例中,一个或更多个处理器核心2402A-2402N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2410包括用于在多线程处理期间协调和操作核心2402A-2402N的组件。在至少一个实施例中,系统代理核心2410可以另外包括电源控制单元(PCU),该电源控制单元包括逻辑和组件以调节处理器核心2402A-2402N和图形处理器2408的一个或更多个电源状态。

[0279] 在至少一个实施例中,处理器2400另外包括图形处理器2408以执行图形处理操作。在至少一个实施例中,图形处理器2408与共享高速缓存单元2406和包括一个或更多个集成存储器控制器2414的系统代理核心2410耦合。在至少一个实施例中,系统代理核心2410还包括用于驱动图形处理器输出到一个或更多个耦合的显示器的显示器控制器2411。在至少一个实施例中,显示器控制器2411也可以是经由至少一个互连与图形处理器2408耦合的独立模块,或者可以集成在图形处理器2408内。

[0280] 在至少一个实施例中,基于环的互连单元2412用于耦合处理器2400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2408经由I/O链路2413与环形互连2412耦合。

[0281] 在至少一个实施例中,I/O链路2413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2418(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2402A-2402N和图形处理器2408中的每一个使用嵌入式存储器模块2418作为共享的最后一级高速缓存。

[0282] 在至少一个实施例中,处理器核心2402A-2402N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2402A-2402N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2402A-2402N执行公共指令集,而一个或更多个其他处理器核心2402A-24-02N执行公共指令集或不同指令集的子集。在至少一个实施例中,就微架构而言,处理器核心2402A-2402N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2400可以实现在一个或更多个芯片上或被实现为SoC集成电路。

[0283] 推理和/或训练逻辑615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。下面结合图6A和/或图6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例中,可以将推理和/或训练逻辑615的部分或全部合并到处理器2400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在图24中的图形处理器2312、图形核心2402A-2402N或其他组件中体现的一个或更多个ALU。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6A或6B所示的逻辑以外的逻辑来完成。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2400的ALU以执行一种或更多种机器学习算法、神经网络架构、用例或本文介绍的训练技术。

[0284] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0285] 图25是根据本文所述的至少一个实施例的图形处理器核心2500的硬件逻辑的框图。在至少一个实施例中,图形处理器核心2500被包括在图形核心阵列内。在至少一个实施例中,图形处理器核心2500(有时称为核心切片)可以是模块化图形处理器内的一个或更多个图形核心。在至少一个实施例中,图形处理器核心2500是一个图形核心切片的示例,并且本文所述的图形处理器可以基于目标功率和性能包络线包括多个图形核心切片。在至少一个实施例中,每个图形核心2500可以包括与多个子核心2501A-2501F耦合的固定功能块2530,也称为子切片,其包括通用和固定功能逻辑的模块块。

[0286] 在至少一个实施例中,固定功能块2530包括几何/固定功能管线2536,例如,在较低性能和/或较低功率的图形处理器实施方式中,该几何/固定功能管线2536可以由图形处理器2500中的所有子核心共享。在至少一个实施例中,几何/固定功能管线2536包括3D固定功能管线、视频前端单元,线程产生器和线程分派器以及管理统一返回缓冲区的统一返回缓冲区管理器。

[0287] 在固定的至少一个实施例中,功能块2530还包括图形SoC接口2537、图形微控制器2538和媒体管线2539。在固定的至少一个实施例中,图形SoC接口2537提供了图形核心2500以及片上集成电路系统中的其他处理器核心之间的接口。在至少一个实施例中,图形微控制器2538是可编程子处理器,其可配置为管理图形处理器2500的各种功能,包括线程分派、调度和抢占。在至少一个实施例中,媒体管线2539包括有助于对包括图像和视频数据的多媒体数据进行解码、编码、预处理和/或后处理的逻辑。在至少一个实施例中,媒体管线2539经由对子核心2501-2501F内的计算或采样逻辑的请求来实现媒体操作。

[0288] 在至少一个实施例中,SoC接口2537使图形核心2500能够与通用应用处理器核心(例如,CPU)和/或SoC内的其他组件通信,包括存储器层次结构元素,诸如共享的最后一级高速缓存、系统RAM和/或嵌入式片上或封装DRAM。在至少一个实施例中,SoC接口2537还可以使得能够与SoC内的固定功能设备(例如,相机成像管线)进行通信,并且使得能够使用和/或实现可以在图形核心2500和SoC内部的CPU之间共享的全局存储器原子。在至少一个实施例中,SoC接口2537还可以实现用于图形核心2500的电源管理控制,并且启用图形核心2500的时钟域与SoC内的其他时钟域之间的接口。在至少一个实施例中,SoC接口2537使得能够从命令流转化器和全局线程分派器接收命令缓冲区,其配置为向图形处理器内的一个或更多个图形核心中的每一个提供命令和指令。在至少一个实施例中,当要执行媒体操作时,可以将命令和指令分派给媒体管线2539,或者当要执行图形处理操作时,可以将其分配给几何形状和固定功能管线(例如,几何形状和固定功能管线2536、几何形状和固定功能管线2514)。

[0289] 在至少一个实施例中,图形微控制器2538可以配置为对图形核心2500执行各种调度和管理任务。在至少一个实施例中,图形微控制器2538可以在子核心2501A-2501F中的执

行单元(EU)阵列2502A-2502F、2504A-2504F内的各种图形并行引擎上执行图形和/或计算工作负载调度。在至少一个实施例中,在包括图形核心2500的SoC的CPU核心上执行的主机软件可以提交多个图形处理器门铃之一的工作负载,其调用适当的图形引擎上的调度操作。在至少一个实施例中,调度操作包括确定接下来要运行哪个工作负载、将工作负载提交给命令流转化器、抢先在引擎上运行的现有工作负载、监控工作负载的进度以及在工作负载完成时通知主机软件。在至少一个实施例中,图形微控制器2538还可以促进图形核心2500的低功率或空闲状态,从而为图形核心2500提供在图形核心2500内独立于操作系统和/或系统上的图形驱动程序软件的跨低功率状态转换的保存和恢复寄存器的能力。

[0290] 在至少一个实施例中,图形核心2500可以具有比所示的子核心2501A-2501F多或少达N个模块化子核心。对于每组N个子核心,在至少一个实施例中,图形核心2500还可以包括共享功能逻辑2510、共享和/或高速缓冲存储器2512、几何/固定功能管线2514以及附加的固定功能逻辑2516以加速各种图形和计算处理操作。在至少一个实施例中,共享功能逻辑2510可以包括可由图形核心2500内的每个N个子核心共享的逻辑单元(例如,采样器、数学和/或线程间通信逻辑)。在至少一个实施例中,固定、共享和/或缓存存储器2512可以是图形核心2500内的N个子核心2501A-2501F的最后一级高速缓存,并且还可以用作可由多个子核心访问的共享存储器。在至少一个实施例中,可以包括几何/固定功能管线2514来代替固定功能块2530内的几何/固定功能管线2536,并且可以包括相同或相似的逻辑单元。

[0291] 在至少一个实施例中,图形核心2500包括附加的固定功能逻辑2516,其可以包括供图形核心2500使用的各种固定功能加速逻辑。在至少一个实施例中,附加的固定功能逻辑2516包括用于仅位置着色中使用的附加的几何管线。在仅位置着色中,存在至少两个几何管线,而在几何/固定功能管线2516、2536内的完整几何管线和剔除管线中,其是可以包括在附加的固定功能逻辑2516中的附加几何管线。在至少一个实施例中,剔除管线是完整几何管线的修整版。在至少一个实施例中,完整管线和剔除管线可以执行应用程序的不同实例,每个实例具有单独的环境。在至少一个实施例中,仅位置着色可以隐藏被丢弃的三角形的长剔除运行,从而在某些情况下可以更早地完成着色。例如,在至少一个实施例中,附加固定功能逻辑2516中的剔除管线逻辑可以与主应用程序并行执行位置着色器,并且通常比完整管线更快地生成关键结果,因为剔除管线获取并遮蔽顶点的位置属性,无需执行光栅化和将像素渲染到帧缓冲区。在至少一个实施例中,剔除管线可以使用生成的临界结果来计算所有三角形的可见性信息,而与这些三角形是否被剔除无关。在至少一个实施例中,完整管线(在这种情况下可以称为重播管线)可以消耗可见性信息来跳过剔除的三角形以仅遮盖最终传递到光栅化阶段的可见三角形。

[0292] 在至少一个实施例中,附加的固定功能逻辑2516还可包括机器学习加速逻辑,例如固定功能矩阵乘法逻辑,用于实现包括用于机器学习训练或推理的优化。

[0293] 在至少一个实施例中,在每个图形子核心2501A-2501F内包括一组执行资源,其可用于响应于图形管线、媒体管线或着色器程序的请求来执行图形、媒体和计算操作。在至少一个实施例中,图形子核心2501A-2501F包括多个EU阵列2502A-2502F、2504A-2504F,线程分派和线程间通信(TD/IC)逻辑2503A-2503F,3D(例如,纹理)采样器2505A-2505F,媒体采样器2506A-2506F,着色器处理器2507A-2507F和共享本地存储器(SLM)2508A-2508F。EU阵列2502A-2502F、2504A-2504F每个都包含多个执行单元,这些执行单元是通用图形处理单

元,能够为图形、媒体或计算操作提供服务,执行浮点和整数/定点逻辑运算,包括图形、媒体或计算着色器程序。在至少一个实施例中,TD/IC逻辑2503A-2503F为子核心内的执行单元执行本地线程分派和线程控制操作,并促进在子核心的执行单元上执行的线程之间的通信。在至少一个实施例中,3D采样器2505A-2505F可以将与纹理或其他3D图形相关的数据读取到存储器中。在至少一个实施例中,3D采样器可以基于与给定纹理相关联的配置的采样状态和纹理格式来不同地读取纹理数据。在至少一个实施例中,媒体采样器2506A-2506F可以基于与媒体数据相关联的类型和格式来执行类似的读取操作。在至少一个实施例中,每个图形子核心2501A-2501F可以可替代地包括统一的3D和媒体采样器。在至少一个实施例中,在每个子核心2501A-2501F内的执行单元上执行的线程可以利用每个子核心内的共享本地存储器2508A-2508F,以使在线程组内执行的线程能够使用片上存储器的公共池来执行。

[0294] 推理和/或训练逻辑615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。下面结合图6A和/或图6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例中,推理和/或训练逻辑615的部分或全部可以被合并到图形处理器2510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在图形处理器2312、图形微控制器2538、几何和固定功能管线2514和2536或图24中的其他逻辑中体现的一个或更多个ALU。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6A或图6B所示的逻辑以外的逻辑来完成。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,所述寄存器配置图形处理器2500的ALU以执行一种或更多种本文介绍的机器学习算法、神经网络架构、用例或训练技术。

[0295] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0296] 图26A-26B示出了根据至少一个实施例的包括图形处理器核心的处理元件的阵列的线程执行逻辑2600。图26A示出了至少一个实施例,其中使用了线程执行逻辑2600。图26B示出了根据至少一个实施例的执行单元的示例性内部细节。

[0297] 如图26A中所示,在至少一个实施例中,线程执行逻辑2600包括着色器处理器2602、线程分派器2604、指令高速缓存2606、包括多个执行单元2608A-2608N的可缩放执行单元阵列、采样器2610、数据高速缓存2612和数据端口2614。在至少一个实施例中,可缩放执行单元阵列可以例如基于工作负载的计算要求,通过启用或禁用一个或更多个执行单元(例如,执行单元2608A、2608B、2608C、2608D、从2608N-1到2608N中的任何一个)来动态缩放。在至少一个实施例中,可缩放执行单元通过链路到每个执行单元的互连结构互连。在至少一个实施例中,线程执行逻辑2600包括通过指令高速缓存2606、数据端口2614、采样器2610和执行单元2608A-2608N中的一个或更多个到存储器(诸如系统存储器或高速缓冲存储器)的一个或更多个连接。在至少一个实施例中,每个执行单元(例如2608A)是独立的可编程通用计算单元,其能够执行多个同时的硬件线程,同时针对每个线程并行处理多个数据元素。在至少一个实施例中,执行单元2608A-2608N的阵列可缩放以包括任意数量的单独执行单元。

[0298] 在至少一个实施例中,执行单元2608A-2608N主要用于执行着色器程序。在至少一

个实施例中,着色器处理器2602可以处理各种着色器程序并经由线程分派器2604来分派与着色器程序相关联的执行线程。在至少一个实施例中,线程分派器2604包括用于仲裁来自图形和媒体管线的线程初始化庆祝以及在执行单元2608A-2608N中的一个或更多个执行单元上实例化请求的线程的逻辑。例如,在至少一个实施例中,几何管线可以将顶点、镶嵌或几何着色器分派到线程执行逻辑以进行处理。在至少一个实施例中,线程分派器2604还可以处理来自执行着色器程序的运行时线程产生请求。

[0299] 在至少一个实施例中,执行单元2608A-2608N支持一种指令集,该指令集包括对许多标准3D图形着色器指令的本机支持,从而使图形库(例如Direct 3D和OpenGL)中的着色器程序只需最少的翻译即可执行。在至少一个实施例中,执行单元支持顶点和几何处理(例如,顶点程序、几何程序、顶点着色器)、像素处理(例如,像素着色器、片段着色器)和通用处理(例如,计算和媒体着色器)。在至少一个实施例中,每个执行单元2608A-2608N包括一个或更多个算术逻辑单元(ALU),能够执行多发出单指令多数据(SIMD),并且多线程操作实现了高效的执行环境尽管有更高的延迟存储器访问。在至少一个实施例中,每个执行单元内的每个硬件线程具有专用的高带宽寄存器文件和相关的独立线程状态。在至少一个实施例中,执行是每个时钟到管线的多次发出,管线能够进行整数、单精度和双精度浮点运算、SIMD分支功能、逻辑运算、先验运算和其他其他运算。在至少一个实施例中,在等待来自存储器或共享功能之一的数据时,执行单元2608A-2608N内的依赖性逻辑使等待线程休眠直到返回了所请求的数据。在至少一个实施例中,当等待线程正在休眠时,硬件资源可以专用于处理其他线程。例如,在至少一个实施例中,在与顶点着色器操作相关联的延迟期间,执行单元可以对像素着色器、片段着色器或另一类型的着色器程序(包括不同的顶点着色器)执行操作。

[0300] 在至少一个实施例中,执行单元2608A-2608N中的每一个执行单元在数据元素的阵列上进行操作。在至少一个实施例中,多个数据元素是“执行大小”或指令的通道数。在至少一个实施例中,执行通道是用于指令内的数据元素访问、屏蔽和流控制的执行的逻辑单元。在至少一个实施例中,多个通道可以独立于用于特定图形处理器的多个物理算术逻辑单元(ALU)或浮点单元(FPU)。在至少一个实施例中,执行单元2608A-2608N支持整数和浮点数据类型。

[0301] 在至少一个实施例中,执行单元指令集包括SIMD指令。在至少一个实施例中,各种数据元素可以作为封装数据类型存储在寄存器中,并且执行单元将基于元素的数据大小来处理各种元素。例如,在至少一个实施例中,当对256位宽的向量进行操作时,将向量的256位存储在寄存器中,并且执行单元对向量进行操作,作为四个单独的64位封装数据元素(四字(QW)大小数据元素)、八个单独的32位封装数据元素(双字(DW)大小数据元素)、十六个单独的16位封装数据元素(单词(W)大小数据元素)或三十二个单独的8位数据元素(字节(B)大小的数据元素)。然而,在至少一个实施例中,不同的向量宽度和寄存器大小是可能的。

[0302] 在至少一个实施例中,一个或更多个执行单元可以被组合成具有执行对于融合EU的线程控制逻辑(2607A-2607N)的融合执行单元2609A-2609N。在至少一个实施例中,可以将多个EU合并成一个EU组。在至少一个实施例中,融合EU组中的每个EU可以配置为执行单独的SIMD硬件线程。融合的EU组中的EU的数量可以根据各种实施例而变化。在至少一个实施例中,每个EU可以执行各种SIMD宽度,包括但不限于SIMD8、SIMD16和SIMD32。在至少一个

实施例中,每个融合图形执行单元2609A-2609N包括至少两个执行单元。例如,在至少一个实施例中,融合执行单元2609A包括第一EU 2608A、第二EU 2608B以及第一EU 2608A和第二EU 2608B共有的线程控制逻辑2607A。在至少一个实施例中,线程控制逻辑2607A控制在融合图形执行单元2609A上执行的线程,从而允许融合执行单元2609A-2609N内的每个EU使用公共指令指针寄存器来执行。

[0303] 在至少一个实施例中,一个或更多个内部指令高速缓存(例如2606)被包括在线程执行逻辑2600中以高速缓存用于执行单元的线程指令。在至少一个实施例中,包括一个或更多个数据高速缓存(例如2612)以在线程执行期间高速缓存线程数据。在至少一个实施例中,包括采样器2610以提供用于3D操作的纹理采样和用于媒体操作的媒体采样。在至少一个实施例中,采样器2610包括专门的纹理或媒体采样功能,以在将采样数据提供给执行单元之前在采样过程中处理纹理或媒体数据。

[0304] 在执行期间,在至少一个实施例中,图形和媒体管线通过线程产生和分派逻辑将线程发起请求发送到线程执行逻辑2600。在至少一个实施例中,一旦一组几何对象已经被处理并光栅化成像素数据,则在着色器处理器2602内的像素处理器逻辑(例如,像素着色器逻辑、片段着色器逻辑等)被调用以进一步计算输出信息并且导致将结果写入输出表面(例如,颜色缓冲区、深度缓冲区、模板缓冲区等)。在至少一个实施例中,像素着色器或片段着色器计算要在光栅化对象上插值的各种顶点属性的值。在至少一个实施例中,着色器处理器2602内的像素处理器逻辑然后执行应用程序接口(API)提供的像素或片段着色器程序。在至少一个实施例中,为了执行着色器程序,着色器处理器2602经由线程分派器2604将线程分派到执行单元(例如2608A)。在至少一个实施例中,着色器处理器2602使用采样器2610中的纹理采样逻辑来访问存储在存储器中的纹理贴图中的纹理数据。在至少一个实施例中,对纹理数据和输入几何数据的算术运算为每个几何片段计算像素颜色数据,或者丢弃一个或更多个像素以进行进一步处理。

[0305] 在至少一个实施例中,数据端口2614提供了一种用于线程执行逻辑2600的存储器访问机制,以将处理后的数据输出到存储器以在图形处理器输出管线上进行进一步处理。在至少一个实施例中,数据端口2614包括或耦合到一个或更多个高速缓存存储器(例如,数据高速缓存2612)以高速缓存数据以便经由数据端口进行存储器访问。

[0306] 如图26B所示,在至少一个实施例中,图形执行单元2608可以包括指令获取单元2637、通用寄存器文件阵列(GRF)2624、架构寄存器文件阵列(ARF)2626、线程仲裁器2622、发送单元2630、分支单元2632、一组SIMD浮点单元(FPU)2634,以及在至少一个实施例中,一组专用整数SIMD ALU2635。在至少一个实施例中,GRF 2624和ARF 2626包括一组与可以在图形执行单元2608中活跃的每个同时硬件线程相关联的通用寄存器文件和架构寄存器文件。在至少一个实施例中,在ARF 2626中维护每个线程架构状态,而在线程执行期间使用的数据存储在GRF 2624中。在至少一个实施例中,每个线程的执行状态,包括每个线程的指令指针,可以被保存在ARF 2626中的线程专用寄存器中。

[0307] 在至少一个实施例中,图形执行单元2608具有一种架构,该架构是同时多线程(SMT)和细粒度交错多线程(IMT)的组合。在至少一个实施例中,架构具有模块化配置,该模块化配置可以在设计时基于同时线程的目标数量和每个执行单元的寄存器数量来进行微调,其中执行单元资源在用于执行多个同时线程的逻辑上分配。

[0308] 在至少一个实施例中,图形执行单元2608可以共同发布多个指令,每个指令可以是不同的指令。在至少一个实施例中,图形执行单元线程2608的线程仲裁器2622可以将指令分派到发送单元2630、分支单元2642或SIMD FPU 2634之一以供执行。在至少一个实施例中,每个执行线程可以访问GRF 2624中的128个通用寄存器,其中每个寄存器可以存储32个字节,可以作为32位数据元素的SIMD 8元素向量进行访问。在至少一个实施例中,每个执行单元线程可以访问GRF 2624中的4KB,尽管实施例不限于此,并且在其他实施例中可以提供更多或更少的寄存器资源。在至少一个实施例中,尽管每个执行单元的线程数量也可以根据实施例而变化,但是最多可以同时执行七个线程。在其中七个线程可以访问4KB的至少一个实施例中,GRF 2624可以存储总共28KB。在至少一个实施例中,灵活的寻址模式可以允许将寄存器一起寻址以有效地建立更宽的寄存器或表示跨步的矩形块数据结构。

[0309] 在至少一个实施例中,经由由消息传递发送单元2630执行的“发送”指令来调度存储器操作、采样器操作和其他更长延迟的系统通信。在至少一个实施例中,将分支指令分派到专用分支单元2632促进SIMD发散和最终收敛。

[0310] 在至少一个实施例中,图形执行单元2608包括一个或更多个SIMD浮点单元(FPU)2634,以执行浮点操作。在至少一个实施例中,FPU 2634还支持整数计算。在至少一个实施例中,FPU 2634可以SIMD执行多达M个32位浮点(或整数)运算,或者SIMD执行多达2M个16位整数或16位浮点运算。在至少一个实施例中,FPU中的至少一个提供扩展的数学能力以支持高吞吐量的先验数学函数和双精度64位浮点。在至少一个实施例中,还存在一组8位整数SIMD ALU 2635,并且可以被专门优化以执行与机器学习计算相关的操作。

[0311] 在至少一个实施例中,可以在图形子核心分组(例如,子切片)中实例化图形执行单元2608的多个实例的阵列。在至少一个实施例中,执行单元2608可以跨多个执行通道执行指令。在至少一个实施例中,在图形执行单元2608上执行的每个线程在不同的通道上执行。

[0312] 推理和/或训练逻辑615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。下面结合图6A和/或图6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例中,推理和/或训练逻辑615的部分或全部可以被结合到执行逻辑2600中。此外,在至少一个实施例中,可以使用除了图6A或图6B中所示的逻辑之外的逻辑来完成在此描述的推理和/或训练操作。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,其配置执行逻辑2600的ALU以执行一种或更多种机器学习算法、神经网络架构、用例或本文介绍的训练技术。

[0313] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0314] 图27示出了根据至少一个实施例的并行处理单元(“PPU”)2700。在至少一个实施例中,PPU 2700配置有机器可读代码,该机器可读代码如果由PPU 2700执行,则使得PPU 2700执行贯穿本公开描述的一些或全部过程和技术。在至少一个实施例中,PPU 2700是在一个或更多个集成电路设备上实现的多线程处理器,并且利用多线程作为被设计为处理在多个线程上并行执行的计算机可读指令(也称为机器可读指令或简单的指令)的延迟隐藏技术。在至少一个实施例中,线程是指执行线程,并且是被配置为由PPU 2700执行的一组指

令的实例。在至少一个实施例中,PPU 2700是图形处理单元(“GPU”),图形处理单元配置为实现用于处理三维(“3D”)图形数据的图形渲染管线,以便生成用于在显示设备(诸如液晶显示器(“LCD”)设备)上显示的二维(“2D”)图像数据。在至少一个实施例中,PPU 2700用于执行计算,诸如线性代数运算和机器学习运算。图27仅出于说明性目的示出了示例并行处理器,并且应被解释为在本公开的范围内设想的处理器架构的非限制性示例,并且可以采用任何适当的处理器来对其进行补充和/或替代。

[0315] 在至少一个实施例中,一个或更多个PPU 2700配置成加速高性能计算(“HPC”)、数据中心和机器学习应用程序。在至少一个实施例中,PPU2700配置成加速深度学习系统和应用程序,包括以下非限制性示例:自动驾驶汽车平台、深度学习、高精度语音、图像、文本识别系统、智能视频分析、分子模拟、药物发现、疾病诊断、天气预报、大数据分析、天文学、分子动力学模拟、财务建模、机器人技术、工厂自动化、实时语言翻译、在线搜索优化以及个性化用户推荐等。

[0316] 在至少一个实施例中,PPU 2700包括但不限于输入/输出(“I/O”)单元2706、前端单元2710、调度器单元2712、工作分配单元2714、集线器2716、交叉开关(“Xbar”)2720、一个或更多个通用处理集群(“GPC”)2718和一个或更多个分区单元(“存储器分区单元”)2722。在至少一个实施例中,PPU 2700通过一个或更多个高速GPU互连(“GPU互连”)2708连接到主机处理器或其他PPU 2700。在至少一个实施例中,PPU 2700通过互连2702连接到主机处理器或其他外围设备。在一实施例中,PPU 2700连接到包括一个或更多个存储器设备(“存储器”)2704的本地存储器。在至少一个实施例中,存储器设备2704包括但不限于一个或更多个动态随机存取存储器(“DRAM”)设备。在至少一个实施例中,一个或更多个DRAM设备配置和/或可配置为高带宽存储器(“HBM”)子系统,并且在每个设备内堆叠有多个DRAM管芯。

[0317] 在至少一个实施例中,高速GPU互连2708可以指代系统使用其来进行缩放的基于线的多通道通信链路,并包括与一个或更多个中央处理单元结合的一个或更多个PPU 2700(“CPU”),支持PPU 2700和CPU之间的缓存相干以及CPU主控。在至少一个实施例中,高速GPU互连2708通过集线器2716将数据和/或命令传输到PPU 2700的其他单元,例如一个或更多个复制引擎、视频编码器、视频解码器、电源管理单元和/或在图27中可能未明确示出的其他组件。

[0318] 在至少一个实施例中,I/O单元2706配置为通过系统总线2702从主机处理器(图27中未示出)发送和接收通信(例如,命令、数据)。在至少一个实施例中,I/O单元2706直接通过系统总线2702或通过一个或更多个中间设备(例如存储器桥)与主机处理器通信。在至少一个实施例中,I/O单元2706可以经由系统总线2702与一个或更多个其他处理器(例如一个或更多个PPU 2700)通信。在至少一个实施例中,I/O单元2706实现外围组件互连Express(“PCIe”)接口,用于通过PCIe总线进行通信。在至少一个实施例中,I/O单元2706实现用于与外部设备通信的接口。

[0319] 在至少一个实施例中,I/O单元2706对经由系统总线2702接收的分组进行解码。在至少一个实施例中,至少一些分组表示被配置为使PPU 2700执行各种操作的命令。在至少一个实施例中,I/O单元2706如命令所指定的那样将解码的命令发送到PPU 2700的各种其他单元。在至少一个实施例中,命令被发送到前端单元2710和/或被发送到集线器2716或PPU 2700的其他单元,例如一个或更多个复制引擎、视频编码器、视频解码器、电源管理单

元等(图27中未明确示出)。在至少一个实施例中,I/O单元2706配置为在PPU 2700的各种逻辑单元之间路由通信。

[0320] 在至少一个实施例中,由主机处理器执行的程序在缓冲区中对命令流进行编码,该缓冲区将工作负载提供给PPU 2700以进行处理。在至少一个实施例中,工作负载包括指令和要由那些指令处理的数据。在至少一个实施例中,缓冲区是可由主机处理器和PPU 2700两者访问(例如,读/写)的存储器中的区域—主机接口单元可以配置为访问经由I/O单元2706通过系统总线2702传输的存储器请求连接到系统总线2702的系统存储器中的缓冲区。在至少一个实施例中,主机处理器将命令流写入缓冲区,然后将指示命令流开始的指针发送给PPU 2700,使得前端单元2710接收指向一个或更多个命令流指针并管理一个或更多个命令流,从命令流中读取命令并将命令转发到PPU 2700的各个单元。

[0321] 在至少一个实施例中,前端单元2710耦合到调度器单元2712,该调度器单元2712配置各种GPC 2718以处理由一个或更多个命令流定义的任务。在至少一个实施例中,调度器单元2712配置为跟踪与调度器单元2712管理的各种任务有关的状态信息,其中状态信息可以指示任务被分配给哪个GPC 2718,任务是活跃的还是非活跃的,与任务相关联的优先级等等。在至少一个实施例中,调度器单元2712管理在一个或更多个GPC 2718上执行的多个任务。

[0322] 在至少一个实施例中,调度器单元2712耦合到工作分配单元2714,该工作分配单元2714配置为分派任务以在GPC 2718上执行。在至少一个实施例中,工作分配单元2714跟踪从调度器单元2712接收到的多个调度任务并且工作分配单元2714管理每个GPC 2718的待处理任务池和活跃任务池。在至少一个实施例中,待处理任务池包括多个时隙(例如32个时隙),这些时隙包含分配给要由特定的GPC 2718处理的任务;活跃任务池可包括用于由GPC 2718主动处理的任务的多个时隙(例如4个时隙),以使随着GPC 2718中的一个完成任务的执行,该任务将从GPC 2718的活动任务池中逐出,并且从待处理任务池中选择其他任务之一,并安排其在GPC 2718上执行。在至少一个实施例中,如果活跃任务在GPC 2718上处于空闲状态,例如在等待数据依赖性解决时,则活跃任务从GPC 2718中驱逐并返回到待处理任务池,同时选择了待处理任务池中的另一个任务并调度在GPC 2718上执行。

[0323] 在至少一个实施例中,工作分配单元2714经由XBar2720与一个或更多个GPC 2718通信。在至少一个实施例中,XBar2720是互连网络,其将PPU 2700的许多单元耦合到PPU 2700的其他单元,并且可以配置为将工作分配单元2714耦合到特定的GPC2718。在至少一个实施例中,一个或更多个PPU 2700的其他单元也可以通过集线器2716连接到XBar2720。

[0324] 在至少一个实施例中,任务由调度器单元2712管理,并由工作分配单元2714分配给GPC 2718之一。GPC2718配置为处理任务并产生结果。在至少一个实施例中,结果可以由GPC 2718中的其他任务消耗,通过XBar2720路由到不同的GPC 2718或存储在存储器2704中。在至少一个实施例中,结果可以通过分区单元2722写到存储器2704中,其实现了用于向存储器2704写入数据或从存储器2704读取数据的存储器接口。在至少一个实施例中,结果可以经由高速GPU互连2708传输到另一PPU 2704或CPU。在至少一个实施例中,PPU 2700包括但不限于U个分区单元2722,其等于耦合到PPU 2700的分离且不同的存储设备2704的数量。在至少一个实施例中,下面结合图29更详细地描述分区单元2722。

[0325] 在至少一个实施例中,主机处理器执行驱动器核心,该驱动程序核心实现应用程

序编程接口(API),该应用程序编程接口使在主机处理器上执行的一个或更多个应用程序能够调度操作以在PPU 2700上执行。在一个实施例中,多个计算应用由PPU 2700同时执行,并且PPU 2700为多个计算应用程序提供隔离、服务质量("QoS")和独立的地址空间。在至少一个实施例中,应用程序生成指令(例如,以API调用的形式),该指令使驱动器核心生成一个或更多个任务以供PPU 2700执行,并且驱动器核心将任务输出至由PPU 2700处理的一个或更多个流。在至少一个实施例中,每个任务包括一个或更多个相关线程组,其可以被称为线程束(warp)。在至少一个实施例中,线程束包括可以并行执行的多个相关线程(例如32个线程)。在至少一个实施例中,协作线程可以指代多个线程,包括用于执行任务并且通过共享存储器交换数据的指令。在至少一个实施例中,结合图29根据至少一个实施例更详细地描述了线程和协作线程。

[0326] 推理和/或训练逻辑615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。下面结合图6A和/或6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例中,深度学习应用处理器用于训练机器学习模型(诸如神经网络),以预测或推理提供给PPU 2700的信息。在至少一个实施例中,PPU 2700用于基于已由另一处理器或系统或PPU 2700训练过的训练过的机器学习模型(例如,神经网络)推理或预测信息。在至少一个实施例中,PPU 2700可用于执行本文所述的一个或更多个神经网络用例。

[0327] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0328] 图28示出了根据至少一个实施例的通用处理集群("GPC")2800。在至少一个实施例中,GPC 2800是图27的GPC 2718。在至少一个实施例中,每个GPC 2800包括但不限于用于处理任务的多个硬件单元,并且每个GPC 2800包括但不限于管线管理器2802、预光栅操作单元("PROP")2804、光栅引擎2808、工作分配交叉开关("WDX")2816、存储器管理单元("MMU")2818、一个或更多个数据处理集群("DPC")2806,以及部件的任何合适组合。

[0329] 在至少一个实施例中,GPC 2800的操作由管线管理器2802控制。在至少一个实施例中,管线管理器2802管理一个或更多个DPC 2806的配置,以处理分配给GPC 2800的任务。在至少一个实施例中,管线管理器2802配置一个或更多个DPC 2806中的至少一个以实现图形渲染管线的至少一部分。在至少一个实施例中,DPC 2806配置为在可编程流式多处理器("SM")2814上执行顶点着色器程序。在至少一个实施例中,管线管理器2802配置为将从工作分配单元接收的数据包路由到GPC 2800内的适当逻辑单元,以及在至少一个实施例中,可以将一些数据包路由到PROP 2804和/或光栅引擎2808中的固定功能硬件单元,而可以将其他数据包路由到DPC 2806以由原始引擎2812进行处理。在至少一个实施例中,管线管理器2802配置DPC 2806中的至少一个以实现神经网络模型和/或计算管线。

[0330] 在至少一个实施例中,PROP单元2804配置为在至少一个实施例中将由光栅引擎2808和DPC 2806生成的数据路由到分区单元2722中的光栅操作("ROP")单元,上面结合图27更详细地描述。在至少一个实施例中,PROP单元2804配置为执行用于颜色混合的优化、组织像素数据、执行地址转换等等。在至少一个实施例中,光栅引擎2808包括但不限于配置为执行各种光栅操作的多个固定功能硬件单元,并且在至少一个实施例中,光栅引擎2808包括但不限于设置引擎、粗光栅引擎、剔除引擎、裁剪引擎、精细光栅引擎、图块聚合引擎及其

任意合适的组合。在至少一个实施例中,设置引擎接收变换后的顶点并生成与由顶点定义的几何图元相关联的平面方程;平面方程式被传送到粗光栅引擎以生成基本图元的覆盖信息(例如,图块的x、y覆盖范围掩码);粗光栅引擎的输出将传输到剔除引擎,在剔除引擎中与z测试失败的图元相关联的片段将被剔除,并传输到剪切引擎,在剪切引擎中剪切位于视锥范围之外的片段。在至少一个实施例中,将经过裁剪和剔除的片段传递给精细光栅引擎,以基于设置引擎生成的平面方程式生成像素片段的属性。在至少一个实施例中,光栅引擎2808的输出包括将由任何适当的实体(例如,由在DPC 2806内实现的片段着色器)处理的片段。

[0331] 在至少一个实施例中,包括在GPC 2800中的每个DPC 2806包括但不限于M管线控制器("MPC")2810;图元引擎2812;一个或更多个SM 2814;及其任何合适的组合。在至少一个实施例中,MPC 2810控制DPC 2806的操作,将从管线管理器2802接收的分组路由到DPC 2806中的适当单元。在至少一个实施例中,将与顶点相关联的分组路由到图元引擎2812,图元引擎2812配置为从存储器中获取与顶点关联的顶点属性;相反,可以将与着色器程序相关联的数据包发送到SM 2814。

[0332] 在至少一个实施例中,SM 2814包括但不限于可编程流式处理器,其配置为处理由多个线程表示的任务。在至少一个实施例中,SM 2814是多线程的并且配置为同时执行来自特定线程组的多个线程(例如32个线程),并且实现单指令多数据("SIMD")架构,其中将一组线程(例如,线程束)中的每个线程配置为基于相同的指令集来处理不同的数据集。在至少一个实施例中,线程组中的所有线程执行相同的指令。在至少一个实施例中,SM 2814实施单指令多线程("SIMT")架构,其中一组线程中的每个线程配置为基于相同的指令集来处理不同的数据集,但是其中线程组中的各个线程允许在执行期间发散。在至少一个实施例中,为每个线程束维护程序计数器、调用栈和执行状态,从而当线程束中的线程发散时,实现线程束和线程束内的串行执行之间的并发性。在另一个实施例中,为每个单独的线程维护程序计数器、调用栈和执行状态,从而使得在线程束内和线程束之间的所有线程之间具有相等的并发性。在至少一个实施例中,为每个单独的线程维持执行状态,并且可以收敛并并行地执行执行相同指令的线程以提高效率。下面更详细地描述SM 2814的至少一个实施例。

[0333] 在至少一个实施例中,MMU 2818在GPC 2800和存储器分区单元(例如,图27的分区单元2722)之间提供接口,并且MMU 2818提供虚拟地址到物理地址的转换、存储器保护以及存储器请求的仲裁。在至少一个实施例中,MMU 2818提供一个或更多个转换后备缓冲区("TLB"),用于执行虚拟地址到存储器中的物理地址的转换。

[0334] 推理和/或训练逻辑615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。下面结合图6A和/或图6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例中,深度学习应用处理器用于训练机器学习模型(诸如神经网络),以预测或推理提供给GPC 2800的信息。在至少一个实施例中,GPC 2800用于基于已由另一处理器或系统或GPC 2800训练过的机器学习模型(例如,神经网络)推理或预测信息。在至少一个实施例中,GPC 2800可用于执行本文所述的一个或更多个神经网络用例。

[0335] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成

全景图像。

[0336] 图29示出了根据至少一个实施例的并行处理单元(“PPU”)的存储器分区单元2900。在至少一个实施例中,存储器分区单元2900包括但不限于光栅操作(“ROP”)单元2902;二级(“L2”)高速缓存2904;存储器接口2906;及其任何合适的组合。在至少一个实施例中,存储器接口2906耦合到存储器。在至少一个实施例中,存储器接口2906可以实现32、64、128、1024位数据总线,或者类似的实现方式用于高速数据传输。在至少一个实施例中,PPU包括U个存储器接口2906,每对分区单元2900一个存储器接口2906,其中每对分区单元2900连接到对应的存储器设备。例如,在至少一个实施例中,PPU可以连接至多达Y个存储器设备,例如高带宽存储器堆栈或图形双数据速率版本5同步动态随机地址存储器(“GDDR5 SDRAM”)。

[0337] 在至少一个实施例中,存储器接口2906实现高带宽存储器第二代(“HBM2”)存储器接口,并且Y等于U的一半。在至少一个实施例中,HBM2存储器堆栈与PPU位于同一物理封装上,与传统的GDDR5 SDRAM系统相比,可提供大量功率并节省面积。在至少一个实施例中,每个HBM2堆栈包括但不限于四个存储器管芯,且Y等于4,每个HBM2堆栈包括每个管芯两个128位通道,用于总共8个通道和1024位的数据总线宽度。在至少一个实施例中,存储器支持单错误校正双错误检测(“SECDED”)错误校正码(“ECC”)以保护数据。在至少一个实施例中,ECC为对数据损坏敏感的计算应用提供更高的可靠性。

[0338] 在至少一个实施例中,PPU实现了多级存储器层次结构。在至少一个实施例中,存储器分区单元2900支持统一存储器以为中央处理单元(“CPU”)和PPU存储器提供单个统一虚拟地址空间,从而实现虚拟存储器系统之间的数据共享。在至少一个实施例中,追踪PPU对位于其他处理器上的存储器的访问频率,以确保将存储器页面移动到更频繁地访问页面的PPU的物理存储器。在至少一个实施例中,高速GPU互连2708支持地址转换服务,其允许PPU直接访问CPU的页表,并通过PPU提供对CPU存储器的完全访问。

[0339] 在至少一个实施例中,复制引擎在多个PPU之间或PPU与CPU之间传输数据。在至少一个实施例中,复制引擎可以为未被映射到页表中的地址生成页面错误,并且存储器分区单元2900然后为页面错误提供服务,将地址映射到页表中,之后复制引擎执行传输。在至少一个实施例中,为多个处理器之间的多个复制引擎操作固定(即不可分页)存储器,从而实质上减少了可用存储器。在至少一个实施例中,在硬件页面故障的情况下,可以将地址传递给复制引擎,而无需考虑是否驻留存储器页面,并且复制过程是透明的。

[0340] 根据至少一个实施例,来自图27的存储器2704或其他系统存储器的数据由存储器分区单元2900获取,并将其存储在L2高速缓存2904中,L2高速缓存2904位于芯片上并且在各种GPC之间共享。在至少一个实施例中,每个存储器分区单元2900包括但不限于与对应的存储器设备相关联的L2高速缓存的至少一部分。在至少一个实施例中,在GPC内的各个单元中实现较低级别的高速缓存。在至少一个实施例中,每个SM 2814可以实现一级(“L1”)高速缓存,其中L1高速缓存是专用于特定SM 2814的私有存储器,并且从L2高速缓存2904中获取数据并将其存储在每个L1高速缓存中,用于在SM 2814的功能单元中进行处理。在至少一个实施例中,L2高速缓存2904耦合到存储器接口2906和XBar2720。

[0341] 在至少一个实施例中,ROP单元2902执行与像素颜色有关的图形光栅操作,诸如颜色压缩、像素混合等。在至少一个实施例中,ROP单元2902结合光栅引擎2808实施深度测试,

从光栅引擎2808的剔除引擎接收与像素片段相关联的样本位置的深度。在至少一个实施例中,针对在与片段关联的样本位置的深度缓冲区中的相应深度测试深度。在至少一个实施例中,如果片段通过了针对样本位置的深度测试,则ROP单元2902更新深度缓冲区,并将深度测试的结果发送给光栅引擎2808。将意识到,分区单元2900的数量可以不同于GPC的数量,因此,可以在至少一个实施例中将每个ROP单元2902耦合到每个GPC。在至少一个实施例中,ROP单元2902追踪从不同GPC接收到的分组,并确定通过XBar2720将ROP单元2902产生的结果路由到哪个。

[0342] 图30示出了根据至少一个实施例的流式多处理器(“SM”)3000。在至少一个实施例中,SM 3000是图28的SM 2814。在至少一个实施例中,SM 3000包括但不限于指令高速缓存3002;一个或更多个调度器单元3004;寄存器文件3008;一个或更多个处理核心(“核心”)3010;一个或更多个特殊功能单元(“SFU”)3012;一个或更多个加载/存储单元(“LSU”)3014;互连网络3016;共享存储器/一级(“L1”)高速缓存3018;及其任何合适的组合。在至少一个实施例中,工作分配单元调度任务以在并行处理单元(“PPU”)的通用处理集群(“GPC”)上执行,并且每个任务被分配给GPC内部的特定数据处理集群(“DPC”),并且如果任务与着色器程序相关联,则将任务分配给SM 3000之一。在至少一个实施例中,调度器单元3004从工作分配单元接收任务并管理分配给SM 3000的一个或更多个线程块的指令调度。在至少一个实施例中,调度器单元3004调度线程块以作为并行线程的线程束来执行,其中,每个线程块被分配至少一个线程束。在至少一个实施例中,每个线程束执行线程。在至少一个实施例中,调度器单元3004管理多个不同的线程块,将线程束分配给不同的线程块,然后在每个时钟周期内将来自多个不同的协作组的指令分派给各种功能单元(例如,处理核心3010、SFU 3012和LSU 3014)。

[0343] 在至少一个实施例中,合作组可以指用于组织通信线程组的编程模型,其允许开发人员表达线程正在通信的粒度,从而能够表达更丰富、更有效的并行分解。在至少一个实施例中,协作启动API支持线程块之间的同步以执行并行算法。在至少一个实施例中,常规编程模型的应用提供了用于同步协作线程的单一、简单的构造:跨线程块的所有线程的屏障(例如,syncthreads()函数)。但是,在至少一个实施例中,程序员可以在小于线程块粒度的情形下来定义线程组,并在所定义的组内进行同步,以实现更高的性能、设计灵活性以及以集合组范围功能接口的形式实现软件重用。在至少一个实施例中,协作组使程序员能够以子块(即,小到单个线程)和多块粒度明确定义线程组,并执行集合操作,例如对协作组中的线程进行同步。在至少一个实施例中,编程模型支持跨软件边界的干净组合,从而库和实用程序功能可以在其本地环境中安全地同步,而不必进行关于收敛的假设。在至少一个实施例中,协作组图元使协作并行的新图案成为可能,包括但不限于生产者-消费者并行,机会主义并行以及整个线程块网格上的全局同步。

[0344] 在至少一个实施例中,调度单元3006配置为将指令发送到功能单元中的一个或更多个,并且调度器单元3004包括但不限于两个调度单元3006,该两个调度单元3006使得来自相同线程束的两个不同指令能够在每个时钟周期被调度。在至少一个实施例中,每个调度器单元3004包括单个调度单元3006或附加调度单元3006。

[0345] 在至少一个实施例中,每个SM 3000在至少一个实施例中包括但不限于寄存器文件3008,该寄存器文件3008为SM 3000的功能单元提供了一组寄存器。在至少一个实施例

中,寄存器文件3008在每个功能单元之间划分,从而为每个功能单元分配寄存器文件3008的专用部分。在至少一个实施例中,寄存器文件3008在由SM 3000执行的不同线程束之间划分,并且寄存器文件3008为连接到功能单元的数据路径的操作数提供临时存储。在至少一个实施例中,每个SM 3000包括但不限于多个L个处理核心3010。在至少一个实施例中,SM 3000包括但不限于大量(例如128个或更多)不同的处理核心3010。在至少一个实施例中,每个处理核心3010在至少一个实施例中包括但不限于全管线、单精度、双精度和/或混合精度处理单元,其包括但不限于浮点算术逻辑单元和整数算术逻辑单元。在至少一个实施例中,浮点算术逻辑单元实现用于浮点算术的IEEE 754-2008标准。在至少一个实施例中,处理核心3010包括但不限于64个单精度(32位)浮点核心、64个整数核心、32个双精度(64位)浮点核心和8个张量核心。

[0346] 根据至少一个实施例,张量核心配置为执行矩阵运算。在至少一个实施例中,一个或更多个张量核心包括在处理核心3010中。在至少一个实施例中,张量核心配置为执行深度学习矩阵算术,例如用于神经网络训练和推理的卷积运算。在至少一个实施例中,每个张量核心在4×4矩阵上操作并且执行矩阵乘法和累加运算D=A×B+C,其中A、B、C和D是4×4矩阵。

[0347] 在至少一个实施例中,矩阵乘法输入A和B是16位浮点矩阵,并且累加矩阵C和D是16位浮点或32位浮点矩阵。在至少一个实施例中,张量核心对具有32位浮点累积的16位浮点输入数据进行操作。在至少一个实施例中,16位浮点乘法使用64个运算,并得到全精度乘积,然后使用32位浮点加法与其他中间乘积累加起来,以进行4x4x4矩阵乘法。在至少一个实施例中,张量核心用于执行由这些较小的元件构成的更大的二维或更高维度的矩阵运算。在至少一个实施例中,API(诸如CUDA 9C++API)公开专门的矩阵加载、矩阵乘法和累加以及矩阵存储操作,以有效地使用来自CUDA-C++程序的张量核心。在至少一个实施例中,在CUDA级别,线程束级别接口假定跨越所有32个线程束线程的16×16大小的矩阵。

[0348] 在至少一个实施例中,每个SM 3000包括但不限于执行特殊功能(例如,属性评估、倒数平方根等)的M个SFU 3012。在至少一个实施例中,SFU 3012包括但不限于配置为遍历分层树数据结构的树遍历单元。在至少一个实施例中,SFU 3012包括但不限于配置为执行纹理映射过滤操作的纹理单元。在至少一个实施例中,纹理单元配置为从存储器中加载纹理映射(例如,纹理像素的2D阵列)和采样纹理映射,以产生采样的纹理值以供由SM 3000执行的着色器程序使用。在至少一个实施例中,将纹理映射存储在共享存储器/L1高速缓存3018中。在至少一个实施例中,根据至少一个实施例,纹理单元使用mip映射(mip-maps)(例如,细节级别不同的纹理映射)来实现纹理操作(诸如过滤操作)。在至少一个实施例中,每个SM 3000包括但不限于两个纹理单元。

[0349] 在至少一个实施例中,每个SM 3000包括但不限于实现共享存储器/L1高速缓存3018与寄存器文件3008之间的加载和存储操作的N个LSU 3014。在至少一个实施例中,每个SM 3000包括但不限于互连网络3016,互连网络3016将每个功能单元连接到寄存器文件3008,并且LSU 3014连接到寄存器文件3008和共享存储器/L1高速缓存3018。在至少一个实施例中,互连网络3016是交叉开关,其可以配置为将任何功能单元连接到寄存器文件3008中的任何寄存器,并且将LSU 3014连接到寄存器文件3008和共享存储器/L1高速缓存3018中的存储器位置。

[0350] 在至少一个实施例中,共享存储器/L1高速缓存3018是片上存储器的阵列,其在至少一个实施例中允许SM 3000与图元引擎之间以及SM3000中的线程之间的数据存储和通信。在至少一个实施例中,共享存储器/L1高速缓存3018包括但不限于128KB的存储容量,并且位于从SM 3000到分区单元的路径中。在至少一个实施例中,共享存储器/L1高速缓存3018在至少一个实施例中用于高速缓存读取和写入。在至少一个实施例中,共享存储器/L1高速缓存3018、L2高速缓存和存储器中的一个或更多个是后备存储器。

[0351] 在至少一个实施例中,将数据高速缓存和共享存储器功能组合到单个存储器块中,为两种类型的存储器访问提供了改进的性能。在至少一个实施例中,容量由不使用共享存储器的程序使用或将其用作高速缓存,例如如果共享存储器配置为使用一半容量,则纹理和加载/存储操作可以使用剩余容量。根据至少一个实施例,在共享存储器/L1高速缓存3018内的集成使共享存储器/L1高速缓存3018能够用作用于流传输数据的高吞吐量管线,同时提供对频繁重用的数据的高带宽和低延迟访问。在至少一个实施例中,当配置用于通用并行计算时,与图形处理相比,可以使用更简单的配置。在至少一个实施例中,绕过固定功能图形处理单元,从而创建了更加简单的编程模型。在至少一个实施例中,在通用并行计算配置中,工作分配单元直接将线程的块分配和分布给DPC。在至少一个实施例中,块中的线程执行相同的程序,在计算中使用唯一的线程ID以确保每个线程生成唯一的结果,使用SM 3000执行程序并执行计算,使用共享存储器/L1高速缓存3018在线程之间进行通信,以及使用LSU 3014通过共享存储器/L1高速缓存3018和存储器分区单元来读写全局存储器。在至少一个实施例中,当被配置用于通用并行计算时,SM 3000向调度器单元3004写入可以用来在DPC上启动新工作的命令。

[0352] 在至少一个实施例中,PPU被包括在台式计算机、膝上型计算机、平板电脑、服务器、超级计算机、智能电话(例如,无线、手持设备)、个人数字助理(“PDA”)、数码相机、车辆、头戴式显示器、手持式电子设备等中或与之耦合。在至少一个实施例中,PPU被实现在单个半导体衬底上。在至少一个实施例中,PPU与一个或更多个其他设备(例如附加的PPU、存储器、精简指令集计算机(“RISC”)CPU,一个或更多个存储器管理单元(“MMU”)、数模转换器(“DAC”)等)一起被包括在片上系统(“SoC”)中。

[0353] 在至少一个实施例中,PPU可以被包括在包括一个或更多个存储设备的图形卡上。图形卡可以配置为与台式计算机主板上的PCIe插槽相连接。在至少一个实施例中,PPU可以是包括在主板的芯片组中的集成图形处理单元(“iGPU”)。

[0354] 推理和/或训练逻辑615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。下面结合图6A和/或6B提供关于推理和/或训练逻辑615的细节。在至少一个实施例中,深度学习应用处理器用于训练机器学习模型(诸如神经网络),以预测或推理提供给SM 3000的信息。在至少一个实施例中,SM 3000用于基于已由另一处理器或系统或由SM 3000训练过的机器学习模型(例如,神经网络)推理或预测信息。在至少一个实施例中,SM 3000可用于执行一个或更多个本文所述的神经网络用例。

[0355] 推理和/或训练逻辑615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。在至少一个实施例中,该逻辑可以与这些图的组件一起使用,以从单个输入图像生成全景图像。

[0356] 在至少一个实施例中,单个半导体平台可以指唯一的单一的基于半导体的集成电

路或芯片。在至少一个实施例中,可以使用具有增加的连接性的多芯片模块,其模拟芯片上的操作,并且相对于利用传统的中央处理单元(“CPU”)和总线实施方式进行了实质性的改进。在至少一个实施例中,根据用户的需求,各种模块也可以分开放置或以半导体平台的各种组合放置。

[0357] 在至少一个实施例中,将以机器可读的可执行代码或计算机控制逻辑算法的形式的计算机程序存储在主存储器1004和/或辅助存储器中。根据至少一个实施例,如果由一个或更多个处理器执行,则计算机程序使系统1000能够执行各种功能。在至少一个实施例中,存储器1004、存储器和/或任何其他存储器是计算机可读介质的可能示例。在至少一个实施例中,辅助存储器可以指代任何合适的存储设备或系统,例如硬盘驱动器和/或可移除存储驱动器,代表软盘驱动器、磁带驱动器、光盘驱动器、数字多功能盘(“DVD”)驱动器、记录设备、通用串行总线(“USB”)闪存等。在至少一个实施例中,各种先前附图的架构和/或功能是在CPU 1002;并行处理系统1012;能够具有至少两个CPU 1002的能力的至少一部分的集成电路;并行处理系统1012;芯片组(例如,一组设计成工作并作为执行相关功能的单元出售的集成电路等);以及集成电路的任何适当组合的环境中实现的。

[0358] 在至少一个实施例中,各种先前附图的架构和/或功能在通用计算机系统、电路板系统、专用于娱乐目的的游戏控制台系统、专用系统等的环境中实现。在至少一个实施例中,计算机系统1000可以采取台式计算机、膝上型计算机、平板电脑、服务器、超级计算机、智能电话(例如,无线,、手持设备)、个人数字助理(“PDA”)、数码相机、车辆、头戴式显示器、手持式电子设备、移动电话设备、电视、工作站、游戏机、嵌入式系统和/或任何其他类型的逻辑的形式。

[0359] 在至少一个实施例中,并行处理系统1012包括但不限于多个并行处理单元(“PPU”)1014和相关联的存储器1016。在至少一个实施例中,PPU1014通过互连1018和开关1020或多路复用器连接到主机处理器或其他外围设备。在至少一个实施例中,并行处理系统1012在可并行化的PPU 1014上分配计算任务,例如,作为跨多个图形处理单元(“GPU”)线程块的计算任务分布的一部分。在至少一个实施例中,在PPU 1014中的一些或全部之间共享和访问存储器(例如,用于读取和/或写入访问),尽管这种共享存储器可能引发相对于使用本地存储器和驻留在PPU 1014上的寄存器的性能损失。在至少一个实施例中,通过使用命令(诸如__syncthreads())来同步PPU 1014的操作,其中块中的所有线程(例如,跨多个PPU 1014执行)在进行之前到达某个代码执行点。

[0360] 虚拟化的计算平台

[0361] 公开了实施例,其涉及用于高级计算的虚拟化计算平台,诸如医疗应用中的图像推断和图像处理。没有限制,实施例可以包括放射线照相,磁共振成像(MRI),核医学,超声,超声检查,弹性成像,光声成像,断层扫描,超声心动图,功能近红外光谱和磁性粒子成像,或其组合。在至少一个实施例中,本文所述的虚拟化计算平台和相关过程可以附加地或替代地用于但不限于法医科学分析,地下探测和成像(例如,石油勘探,考古,古生物学等),地形,海洋学,地质学,骨病学,气象学,智能区域或对象跟踪和监视,传感器数据处理(例如RADAR,SONAR,LIDAR等)和/或基因组学和基因测序。

[0362] 参考图31是根据至少一个实施例的用于生成和部署图像处理和推理管线的过程3100的示例数据流程图。在至少一个实施例中,处理3100可以被部署以与成像设备,处理设

备,基因组设备,基因测序设备,放射学设备和/或一个或多个设施3102处的其他设备类型一起使用,例如医疗设施,医院,医疗机构,诊所,研究或诊断实验室等。在至少一个实施例中,过程3100可以被部署以对测序数据执行基因组分析和推理。可以使用本文所述的系统和过程进行的基因组分析的实例包括但不限于变体调用,突变检测和基因表达定量。过程3100可以在训练系统3104和/或部署系统3106内执行。在至少一个实施例中,训练系统3104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,以供在部署系统3106中使用。在至少一个实施例中,部署系统3106可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3102的基础架构要求。在一个实施例中,部署系统3106可以提供简化的平台,用于选择,定制和实现与设施3102处的成像设备(例如,MRI,CT扫描,X射线,超声等)或测序设备一起使用的虚拟仪器。在至少一个实施例中,虚拟仪器可以包括软件定义的应用程序,用于执行关于成像数据的一个或更多个处理操作,所示成像数据由成像设备,测序设备,放射学设备和/或其他设备类型生成。在至少一个实施例中,管线中的一个或更多个应用可以在应用的执行期间使用或调用部署系统3106的服务(例如,推理,可视化,计算,AI等)。

[0363] 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3102处生成(并存储在设施3102处的一个或更多个图片归档和通信系统(PACS)服务器上)的数据3108(例如成像数据)在设施3102处训练机器学习模型,可以使用来自另一设施(例如,不同的医院,实验室,诊所等)或其组合的成像或测序数据3108来训练。在至少一个实施例中,训练系统3104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3106的工作的,可部署的机器学习模型。

[0364] 在至少一个实施例中,模型注册表3124可以由可以支持版本控制和对象元数据的对象存储支持。在至少一个实施例中,可以通过来自云平台内的云存储(例如,图32的云3226)兼容应用程序编程接口(API)来访问对象存储。在至少一个实施例中,模型注册表3124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载,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,该方法允许具有适当证书的用户将模型与应用程序相关联,使得可以将模型作为应用程序的容器化实例的执行的一部分来执行。

[0365] 在至少一个实施例中,训练管线3204(图32)可以包括设施3102正在训练他们自己的机器学习模型或具有需要被优化或更新的现有机器学习模型的场景。在至少一个实施例中,可以接收由一个或更多个成像设备,测序设备和/或其他设备类型生成的成像数据3108。在至少一个实施例中,一旦接收到成像数据3108,则AI辅助注释3110可用于帮助生成与成像数据3108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3110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)),可以对其进行训练以生成与某些类型的成像数据3108(例如,来自某些设备)相对应的注释和/或成像数据3108中某些类型的异常。在至少一个实施例中,然后可以直接使用AI辅助注释3110,或可以使用注释工具(例如,由研究人员,临床医生,医生,科学家等)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,在一些示例中,标记的临床数据3112(例如,由临床医生,医生,科学家,技术人员等提供的注释)可以用作用于训练机器学习模型的

地面实况数据。在至少一个实施例中,AI辅助注释3110,标记的临床数据3112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3116,并且可以由部署系统3106使用,如本文所述。

[0366] 在至少一个实施例中,训练管线3204(图32)可以包括以下场景:设施3102需要机器学习模型以用于为部署系统3106中的一个或更多个应用程序执行一个或更多个处理任务,但是设施3102当前可能没有这样的机器学习模型(或者可能没有为此目的而优化,有效或有效的模型)。在至少一个实施例中,可以从模型注册表3124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3124可以包括机器学习模型,该机器学习模型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,可以在来自不同于设施3102的设施(例如,位于远程的设施)的成像数据上训练模型注册表3124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,两个位置或任何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(例如,遵守HIPAA法规,隐私法规等)。在至少一个实施例中,一旦在一个位置上训练了模型或部分地训练了模型,就可以将机器学习模型添加到模型注册表3124中。在至少一个实施例中,然后可以对机器学习模型进行重新训练或更新,可以在模型注册表3124中使用经过重新训练或更新的模型。在至少一个实施例中,然后可以从模型注册表3124中选择机器学习模型,并称为输出模型3116,并且可以在部署系统3106中使用以为部署系统的一个或更多个应用程序执行一个或更多个处理任务。

[0367] 在至少一个实施例中,训练管线3204(图32),场景可以包括设施3102,该设施3102需要机器学习模型以用于为部署系统3106中的一个或更多个应用程序执行一个或更多个处理任务,但是设施3102当前可能没有这样的机器学习模型(或者可能没有为此目的而优化,有效或有效的模型)。在至少一个实施例中,由于群体,遗传变异,用于训练机器学习的训练数据的鲁棒性,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3124中选择的机器学习模型可能未针对设施3102处生成的成像数据3108进行微调或优化。在至少一个实施例中,AI辅助注释3110可以用于帮助生成与成像数据3108相对应的注释,以用作用于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,标记的临床数据3112(例如,由临床医生,医生,科学家等提供的注释)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3114。在至少一个实施例中,模型训练3114-例如AI辅助注释3110,标记的诊所数据3112或它们的组合-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,经训练的机器学习模型可以被称为输出模型3116,并且可以由部署系统3106使用,如本文所述。

[0368] 在至少一个实施例中,部署系统3106可以包括软件3118,服务3120,硬件3122和/或其他组件,特征和功能。在至少一个实施例中,部署系统3106可以包括软件“堆栈”,使得软件3118可以建立在服务3120的顶部并且可以使用服务3120来执行一些或全部处理任务,并且服务3120和软件3118可以建立在硬件3122的顶部,并使用硬件3122执行部署系统3106的处理,存储和/或其他计算任务。在至少一个实施例中,软件3118可以包括任意数量的不同容器,其中每个容器可以执行一个应用程序的实例化。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,对于每种类型的成像设备(例如,CT,MRI,X射线,超声,超声检查,超声心动图等),测序设备,放射线学设备,基因组学设备等,可以有任何数量的容器,所述容器可以针对由设备生成的成像数据3108(或其他数据类型,例如本文描述的那些数据)执行数据处理任务。在至少一个实施例中,除了接收和配置供每个容器和/或供设施3102在通过管线处理(例如,将输出转换回可用的数据类型,例如医学数字成像和通信(DICOM)数据,放射线信息系统(RIS)数据,临床信息系统(CIS))数据,远程过程调用(RPC)数据,基本符合表示状态传输(REST)接口的数据,基本符合基于文件的接口的数据和/或原始数据,以在设施3102进行存储和显示)后使用的成像数据之外,还可以基于处理成像数据3108所需的不同的容器的选择来定义高级处理和推理管线。在至少一个实施例中,软件3118内的容器的组合(例如,构成管线的容器)可以被称为虚拟仪器(如本文中更详细地描述),并且虚拟仪器可以利用服务3120和硬件3122,以执行在容器中实例化的应用程序的部分或全部处理任务。

[0369] 在至少一个实施例中,数据处理管线可以响应于推断请求(例如,来自部署系统3106的用户(例如临床医生,医生,放射线医生等)的请求)而以DICOM,RIS,CIS,REST兼容,RPC,原始和/或其他格式接收输入数据(例如,成像数据3108)。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,测序设备,放射学设备,基因组学设备和/或其他设备类型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,以准备数据以供一个或更多个应用程序处理。在至少一个实施例中,可以对管线的一个或更多个推理任务或其他处理任务的输出执行后处理,以准备用于下一应用程序的输出数据和/或准备用于传输和/或由用户使用的输出数据(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来执行,例如训练或部署的神经网络,其可以包括训练系统3104的输出模型3116。

[0370] 在至少一个实施例中,数据处理管线的任务可以封装在一个或更多个容器中,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的,全功能的实例。在至少一个实施例中,容器或应用程序可以被发布到容器注册表的私有(例如,有限访问)区域中(在此更详细地描述),并且训练或部署的模型可以被存储在模型注册表3124中并与一个或更多的应用程序相关联。在至少一个实施例中,应用程序的图像(例如,容器图像)可以在容器注册表中可用,并且一旦被用户从容器注册表中选择以部署在管线中,则可以使用图像来生成由用户系统使用的应用程序的实例的容器。

[0371] 在至少一个实施例中,开发人员(例如,软件开发人员,临床医生,医生等)可以开发,发布和存储用于对所提供的数据执行图像处理和/或推理的应用程序(例如,作为容器)。在至少一个实施例中,可以使用与系统相关联的软件开发套件(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/或容器与系统兼容或兼容)。在至少一个实施例中,可以使用SDK来对开发的应用程序进行本地测试(例如,在第一设施处,在来自第一设施的数据上),该SDK可以支持至少某些服务3120作为系统(例如,图32的系统3200)。在至少一个实施例中,因为DICOM对象可以包含一到数百个图像或其他数据类型的任何地方,并且由于数据的变化,所以开发人员可以负责管理(例如,设置构造,将预处理构建为应用程序等)提取和准备传入的DICOM数据。在至少一个实施例中,一旦通过系统3200的验证(例如,

准确性,安全性,患者隐私等),应用程序就可以在容器注册表中可用,由用户(例如,医院,诊所,实验室,医疗保健供应商等)选择和/或实施,以针对用户设施(例如第二设施)上的数据执行一个或更多个处理任务。

[0372] 在至少一个实施例中,开发人员然后可以通过网络共享应用程序或容器,以供系统(例如,图32的系统3200)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3124中。在至少一个实施例中,请求实体(例如,医疗机构中的用户)-提供推理或图像处理请求的人-可以浏览容器注册表和/或模型注册表3124用于应用程序,容器,数据集,机器学习模型等,选择所需的元件组合以包含在数据处理管线中,并提交成像处理请求。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/或可以包括在处理请求时将执行的一个或更多个应用程序和/或机器学习模型的选择。在至少一个实施例中,然后可以将请求传递到部署系统3106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3106进行的处理可以包括从容器注册表和/或模型注册表3124中引用选择的元件(例如,应用程序,容器,模型等)。在至少一个实施例中,一旦由管线生成结果,结果可以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)。在至少一个实施例中,放射线医师可以从包括任何数量的应用程序和/或容器的数据处理管线接收结果,其中结果可以包括X射线,CT扫描,MRI等中的异常检测。

[0373] 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3120。在至少一个实施例中,服务3120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3120可以提供软件3118中一个或更多个应用所共有的功能,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3120提供的功能可以动态且更有效地运行,同时还通过允许应用程序并行处理数据(例如,使用并行计算平台3230(图32))来很好地缩放。在至少一个实施例中,不是要求共享服务3120所提供的相同功能的每个应用程序具有服务3120的相应实例,而是可以在各种应用程序之间共享服务3120。在至少一个实施例中,作为非限制性示例,服务可以包括可以用于执行检测或分段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,该数据增强服务可以提供GPU加速的数据(例如,DICOM,RIS,CIS,REST兼容,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,该可视化服务可以添加图像渲染效果-例如,光线跟踪,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,该虚拟仪器服务为虚拟仪器的管线内的其他应用程序提供波束形成,分段,推理,成像和/或支持。

[0374] 在至少一个实施例中,在服务3120包括AI服务(例如,推理服务)的情况下,与用于异常检测(例如,肿瘤,生长异常,瘢痕形成等)的应用程序相关联的一个或更多个机器学习模型可以通过调用(例如,作为API调用)推理服务(例如,推理服务器)执行,以执行一个或更多个机器学习模型或作为应用程序执行的一部分来处理。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

理服务来执行用于执行与分割任务相关联的一个或更多个处理操作的机器学习模型。在至少一个实施例中,可以简化实现包括分段应用程序和异常检测应用程序的高级处理和推理管线的软件3118,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。

[0375] 在至少一个实施例中,硬件3122可以包括GPU,CPU,图形卡,AI/深度学习系统(例如,AI超级计算机,例如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3122为部署系统3106中的软件3118和服务3120提供有效的,特定目的的支持。在至少一个实施例中,可以实现对GPU处理的使用以进行本地处理(例如,在设施3102),在AI/深度学习系统中,在云系统中和/或在部署系统3106的其他处理组件中,以提高图像处理,图像重建,分割,MRI检查,中风或心脏病发作检测(例如实时),渲染中的图像质量等的效率,准确性和功效。在至少一个实施例中,设施可包括成像设备,基因组设备,测序设备和/或本地的其他设备类型可以利用GPU生成代表对象解剖结构的成像数据。在至少一个实施例中,作为非限制性示例,可以针对深度学习,机器学习和/或高性能计算针对GPU处理优化软件3118和/或服务3120。在至少一个实施例中,部署系统3106和/或训练系统3104的计算环境中的至少一些可以在数据中心中利用GPU优化的软件(例如,NVIDIA的DGX系统的硬件和软件的组合)来在一个或更多个超级计算机或高性能计算系统中执行)。在至少一个实施例中,数据中心可以符合HIPAA的规定,使得关于患者数据的隐私的成像数据和/或其他患者数据的接收,处理和传输安全地处理。在至少一个实施例中,硬件3122可以包括可被调用以并行执行数据处理的任意数量的GPU,如本文所述。在至少一个实施例中,云平台还可包括用于GPU优化执行的深度学习任务,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编排系统(例如,KUBERNETES),以实现无缝缩放和负载平衡。

[0376] 图32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3200的系统图。在至少一个实施例中,系统3200可以用于实现图31的过程3100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3200可以包括训练系统3104和部署系统3106。在至少一个实施例中,可以使用软件3118,服务3120和/或硬件3122来实现训练系统3104和部署系统3106,如本文所述。

[0377] 在至少一个实施例中,系统3200(例如,训练系统3104和/或部署系统3106)可以在云计算环境中(例如,使用云3226)实现。在至少一个实施例中,系统3200可以关于医疗服务设施在本地实现,或者作为云和本地计算资源的组合来实现。在至少一个实施例中,在实现云计算的实施例中,患者数据可以与系统3200的一个或更多个组件分离或不被系统3200的一个或更多个组件处理,这将导致处理不符合HIPAA和/或其他数据处理和隐私法规或法律。在至少一个实施例中,可以通过制定的安全措施或协议将对云3226中的API的访问限制为授权用户。在至少一个实施例中,安全协议可以包括可以由认证服务(例如,AuthN,AuthZ,Gluecon等)签名的网络令牌,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(本文所描述的)或系统3200的其他实例可以被限制为已经被审核或被授权用于交互的一组公共IP。

[0378] 在至少一个实施例中,系统3200的各个组件可以使用各种不同的网络类型中的任何一种在彼此之间进行通信,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,可以通过数据总线,无线数据协议(Wi-Fi),有线数据协议(例如以太网)等进行系统3200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)。

[0379] 在至少一个实施例中,训练系统3104可以执行训练管线3204,类似于本文关于图31所描述的那些。在至少一个实施例中,其中部署系统3106将在部署管线3210中使用一个或更多个机器学习模型,训练管线3204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3206(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3204的结果,可以生成一个或更多个输出模型3116。在至少一个实施例中,训练管线3204可以包括任何数量的处理步骤,例如但不限于成像数据(或其他输入数据)的转换或适配(例如,使用DICOM适配器3202A将DICOM图像转换为适合于通过相应的机器学习模型处理的另一种格式,例如神经影像信息技术倡议(NIfTI)格式),AI辅助注释3110,成像数据的标记或注释3108以生成标记的临床数据3112,从模型注册表中选择模型,模型训练3114,训练,重新训练或更新模型和/或其他处理步骤。在至少一个实施例中,对于由部署系统3106使用的不同的机器学习模型,可以使用不同的训练管线3204。在至少一个实施例中,类似于关于图31所描述的第一示例的训练管线3204可用于第一机器学习模型,类似于关于图31描述的第二示例的训练管线3204可以用于第二机器学习模型,并且类似于关于图31描述的第三示例的训练管线3204可以用于第三机器学习模型。在至少一个实施例中,可以根据每个相应的机器学习模型的要求来使用训练系统3104内的任务的任何组合。在至少一个实施例中,机器学习模型中的一个或更多个可以已经被训练并且准备好部署,因此机器学习模型可以不由训练系统3104进行任何处理,并且可以由部署系统3106来实现。

[0380] 在至少一个实施例中,取决于实施方式或实施例,一个或更多个输出模型3116和/或一个或更多个预训练模型3206可以包括任何类型的机器学习模型。在至少一个实施例中,但不限于,系统3200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表示聚类,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,长期/短期记忆(LSTM),Hopfield,Boltzmann,深层信念,反卷积,生成对抗,液体状态机等)的一个或更多个机器学习模型和/或其他类型的机器学习模型。

[0381] 在至少一个实施例中,训练管线3204可以包括AI辅助注释,如本文至少关于图35B更详细地描述的。在至少一个实施例中,可以通过许多技术来生成标记的临床数据3112(例如,传统注释)。在至少一个实施例中,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,适于为地面实况生成注释或标签的另一种程序,和/或可以手绘生成标签或其他注释,在一些示例中。在至少一个实施例中,地面实况数据可以是合成产生的(例如,从计算机模型或渲染产生的),真实产生的(例如,从真实世界的数据设计和产生的),机器自动化的(例如,使用特征分析和学习以从数据中提取特征,然后生成标签),人工注释(例如,贴标机或注释专家,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3108(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

3104生成的对应的地面实况数据。在至少一个实施例中,AI辅助注释可以作为部署管线3210的一部分执行;补充或代替训练管线3204中包括的AI辅助注释。在至少一个实施例中,系统3200可包括多层平台,该多层平台可包括可以执行一项或多项医学成像和诊断功能的诊断应用程序(或其他应用程序类型)的软件层(例如,软件3118)。在至少一个实施例中,系统3200可以通信地耦合到(例如,经由加密的链路)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3200可以被配置为从PACS服务器(例如,通过DICOM适配器3202或其他数据类型适配器,例如RIS,CIS,REST兼容,RPC,原始等)访问和引用数据(例如,DICOM数据,RIS数据,原始数据,CIS数据,符合REST的数据,RPC数据,原始数据等)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。

[0382] 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/或认证的API,通过这些,可以从一个或更多个外部环境(例如,设施3102)援引(例如,调用)应用程序或容器。在至少一个实施例中,应用程序然后可以调用或执行一个或更多个服务3120以执行与各个应用程序相关联的计算,AI或可视化任务,并且软件3118和/或服务3120可以利用硬件3122来以有效和高效的方式执行处理任务。

[0383] 在至少一个实施例中,部署系统3106可以执行部署管线3210。在至少一个实施例中,部署管线3210可以包括可以顺序地,非顺序地或以其他方式应用于由成像设备,测序设备,基因组设备等(包括如上所述的AI辅助注释)生成成像数据(或其他数据类型)的任意数量的应用程序。在至少一个实施例中,如本文所述,用于单个设备的部署管线3210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,取决于从设备生成的数据中期望的信息,可以有一个以上的部署管线3210。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3210,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3210。

[0384] 在至少一个实施例中,可用于部署管线3210的应用程序可包括可用于对来自设备的成像数据或其他数据执行处理任务的任何应用程序。在至少一个实施例中,不同的应用程序可以负责图像增强,分割,重建,异常检测,对象检测,特征检测,治疗计划,剂量测定,射束计划(或其他放射治疗程序)和/或其他分析,图像处理或推理任务。在至少一个实施例中,部署系统3106可以为每个应用程序定义构造,使得部署系统3106(例如,医疗设施,实验室,诊所等)的用户可以理解构造并且使应用程序适于在其各自设施内的实现。在至少一个实施例中,可以选择用于图像重建的应用程序以包括在部署管线3210中,但是由成像设备生成的数据类型可以与在应用程序内使用的数据类型不同。在至少一个实施例中,可以在部署管线3210内使用DICOM适配器3202B(和/或DICOM读取器)或另一数据类型的适配器或读取器(例如,RIS,CIS,REST兼容,RPC,原始等),以将数据转换为部署系统3106中的应用程序可使用的形式。在至少一个实施例中,可以累积和预处理对DICOM,RIS,CIS,REST兼容,RPC,原始和/或其他数据类型库的访问,包括解码,提取和/或执行任何卷积,色彩校正,清晰度,伽玛和/或其他对数据的扩充。在至少一个实施例中,DICOM,RIS,CIS,REST兼容,RPC,和/或原始数据可以是无序的,并且可以执行预通过以组织或分类收集的数据。在至少一个实施例中,因为各种应用程序可以共享共同的图像操作,所以在一些实施例中,可以使用数据增强库(例如,作为服务3120之一)来加速这些操作。在至少一个实施例中,为了避免依赖

于CPU处理的常规处理方法的瓶颈,并行计算平台3230可以用于这些处理任务的GPU加速。

[0385] 在至少一个实施例中,图像重建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3124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用于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3200的其他特征,例如服务3120和硬件3122,部署管线3210可以甚至更加用户友好,提供更容易的集成并且产生更准确,有效和及时的结果。

[0386] 在至少一个实施例中,部署系统3106可以包括用户界面3214(例如,图形用户界面,网络界面等),该用户界面3214可以用于选择要包括在一个或更多个部署管线3210中的应用程序,布置应用程序,修改或改变应用程序或参数或其构造,在设置和/或部署期间使用一个或更多个部署管线3210并与之交互,和/或以其他方式与部署系统3106交互。在至少一个实施例中,尽管关于训练系统3104未示出,用户界面3214(或不同的用户界面)可以用于选择在部署系统3106中使用的模型,在训练系统3104中选择用于训练或重新训练的模型,和/或用于与训练系统3104交互。

[0387] 在至少一个实施例中,除了应用程序编排系统3228之外,还可以使用管线管理器3212管理一个或更多个部署管线3210的应用程序或容器与服务3120和/或硬件3122之间的交互。在至少一个实施例中,管线管理器3212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3120和/或从应用程序或服务到硬件3122的交互。在至少一个实施例中,尽管被示为包括在软件3118中,但是并不旨在限制,并且在一些示例中(例如,如图33所示),管线管理器3212可以被包括在服务3120中。在至少一个实施例中,应用程序编排系统3228(例如,Kubernetes,DOCKER等)可以包括容器编排系统,该系统可以将应用程序分组在容器中作为逻辑单元,以进行协调,管理,扩展和部署。在至少一个实施例中,通过将来自一个或更多个部署管线3210的应用程序(例如,重建应用程序,分段应用程序等)与各个容器相关联,每个应用程序可以在自包含的环境中(例如,在内核等级处)执行,以提高速度和效率。

[0388] 在至少一个实施例中,每个应用程序和/或容器(或其图像)可以单独开发,修改和部署(例如,第一用户或开发人员可以开发,修改和部署第一应用程序,第二用户或开发人员可以开发,修改和部署与第一用户或开发人员分开的第二应用程序),这可以使专注于并关注单个应用程序和/或一个或更多个容器的任务,而不会被另外一个或更多个应用程序或一个或更多个容器的任务阻碍。在至少一个实施例中,管线管理器3212和应用程序编排系统3228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/或输出是由系统(例如,基于应用程序或容器的构造)已知的,应用程序编排系统3228和/或管线管理器3212可以促进每个应用程序或容器之间的通信以及资源共享。在至少一个实施例中,因为一个或更多个部署管线3210中的一个或更多个应用程序或容器可以共享相同的服务和资源,所以应用程序编排系统3228可以协调,负载平衡并确定各种应用程序或容器之间的服务或资源的共享。在至少一个实施例中,调度器可以用于跟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,并在应用程序之间分配资源。在一些示例中,调度器(和/或应用程序编排系统3228的其他组件)可以基于施加在系统上的约束(例如,用户约束),例如服务质量(QoS),针对数据输出的需求紧迫性(例如,确定是执行实时处理还是延迟处理)等,来确定资源可用性和分配。

[0389] 在至少一个实施例中,由部署系统3106中的应用程序或容器利用和共享的服务3120可以包括计算服务3216,AI服务3218,可视化服务3220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)服务3120中的一个或更多个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3216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,可以利用一个或更多个计算服务3216来执行并行处理(例如,使用并行计算平台3230),以通过一个或更多个应用程序和/或单个应用程序的一个或更多个任务来基本同时处理数据。在至少一个实施例中,并行计算平台3230(例如,NVIDIA的CUDA)可以实现在GPU(GPGPU)(例如,GPU 3222)上的通用计算。在至少一个实施例中,并行计算平台3230的软件层可以提供对GPU的虚拟指令集和并行计算元件的访问,以执行计算内核。在至少一个实施例中,并行计算平台3230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/或在单个容器内的不同处理任务之间共享存储器。在至少一个实施例中,可以为多个容器和/或容器内的多个过程生成进程间通信(IPC)调用,以使用来自并行计算平台3230的共享存储器段的相同数据(例如,其中应用程序的多个不同阶段或多个应用程序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,同时,在不同时间等)。在至少一个实施例中,由于数据被用作处理的结果来生成新数据,因此可以在各种应用程序之间存储和共享数据的新位置的该信息。在至少一个实施例中,数据的位置以及更新或修改的数据的位置可以是容器中如何理解有效载荷的定义的一部分。

[0390] 在至少一个实施例中,可以利用AI服务3218来执行推理服务,以执行与应用程序相关联的一个或更多个机器学习模型(例如,被赋予执行应用程序的一个或更多个处理任务的任务)。在至少一个实施例中,AI服务3218可以利用AI系统3224来执行一个或更多个机器学习模型(例如,诸如CNN的神经网络)以用于分割,重构,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,一个或更多个部署管线3210的应用程序可以使用来自训练系统3104的输出模型3116中的一个或更多个和/或其他应用程序模型来对成像数据(例如,DICOM数据,RIS数据,CIS数据,REST兼容数据,RPC数据,原始数据等)进行推理。在至少一个实施例中,使用应用程序编排系统3228(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,第一类别可以包括可以实现更高服务水平协议的高优先级/低延迟路径,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,该标准优先级路径可以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请求。在至少一个实施例中,应用程序协调系统3228可以基于用于AI服务3218的不同推理任务的优先级路径来分配资源(例如,服务3120和/或硬件3122)。

[0391] 在至少一个实施例中,共享存储可以被安装到系统3200内的AI服务3218。在至少一个实施例中,共享存储可以用作高速缓存(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3106的一组API实例可以接收到请求,并且可以选择一个或更多个实例(例如,用于最佳配合,用于负载平衡等)以处理请求。在至少一个实施例中,为了处理请求,可以将请求输入数据库中,可以从模型注册表3124中找到机器学习模型(如果尚未在高速缓存中),验证步骤可以确保将适当的机器学习模型加载到其中。高速缓存(例如,共享存储)和/或模型的副本可以保存到高速缓存中。在至少一个实施例中,如果应用程序还没有运行或者没有足够的实例,则可以使用调度器(例如,管线管理器3212)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以执行模型,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,在其中推理服务器被集群化的拉模型中,只要负载平衡是有利的,就可以缓存模型。在至少一个实施例中,推理服务器可以被静态地加载在对应的分布式服务器中。

[0392] 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(以及可选地模型的多个版本)相关联。在至少一个实施例中,如果当接收到对模型执行推理的请求时不存在推理服务器的实例,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于不同的模型,只要推理服务器作为不同的实例运行即可。

[0393] 在至少一个实施例中,在应用程序执行期间,可以接收针对给定应用程序的推理请求,并且可以加载(如果尚未加载)容器(例如,托管推理服务器的实例),以及启动程序可能会被调用。在至少一个实施例中,容器中的预处理逻辑可以对传入的数据(例如,使用一个或更多个CPU和/或一个或更多个GPU)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,一旦准备好数据进行推理,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X射线)的单个推理调用,或者可能需要对数百个图像(例如,胸部CT)进行推理。在至少一个实施例中,应用程序可以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,像素水平分割,体素水平分割,生成可视化或生成文本以总结发现。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,某些模型可能具有实时(TAT<1分钟)优先级,而其他模型可能具有较低的优先级(例如TAT<10分钟)。在至少一个实施例中,可以从请求机构或实体测量模型执行时间,并且可以包括伙伴网络遍历时间以及在推理服务上的执行。

[0394] 在至少一个实施例中,服务3120与推理应用程序之间的请求传送可以被隐藏在软件开发工具包(SDK)的后面,并且可以通过队列来提供鲁棒的传输。在至少一个实施例中,由API将请求放置在队列中,用于单个应用程序/租户ID组合,并且SDK将从队列中拉出请求并将请求提供给应用程序。在至少一个实施例中,可以在SDK从中拾取队列的环境中提供队列的名称。在至少一个实施例中,通过队列的异步通信可能是有用的,因为它可以允许应用程序的任何实例在可用时拾取工作。结果可以通过队列传回,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段的能力,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,其按收到的顺序处理任务。在至少一个实施例中,应用程序可以在云3226中生成的GPU加速的实例上运行,并且推理服务可以在GPU上执行推理。

[0395] 在至少一个实施例中,可视化服务3220可被利用来生成可视化以查看应用程序和/或一个或更多个部署管线3210的输出。在至少一个实施例中,可视化服务3220可利用GPU 3222来生成可视化。在至少一个实施例中,可视化服务3220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体积渲染,3D体积重建,2D断层摄影切片,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,以生成虚拟交互式显示器或环境(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3220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,光线跟踪,光栅化,内部光学器件等)。

[0396] 在至少一个实施例中,硬件3122可以包括GPU 3222,AI系统3224,云3226和/或用于执行训练系统3104和/或部署系统3106的任何其他硬件。在至少一个实施例中,GPU 3222(例如NVIDIA的TESLA和/或QUADRO GPU)可以包括可用于执行计算服务3216,AI服务3218,可视化服务3220,其他服务和/或软件3118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3218,GPU 3222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进行后处理,和/或执行推理(例如执行机器学习模型)。在至少一个实施例中,云3226,AI系统3224和/或系统3200的其他组件可以使用GPU3222。在至少一个实施例中,云3226可以包括用于深度学习任务的GPU优化平台。在至少一个实施例中,AI系统3224可以使用GPU,并且可以使用一个或更多个AI系统3224来执行云3226(或至少一部分负责深度学习或推理的任务)。同样,尽管硬件3122被示为离散的,这不意味着限制,并且硬件3122的任何组件可以与硬件3122的任何其他组件组合或利用。

[0397] 在至少一个实施例中,AI系统3224可以包括被配置用于推理,深度学习,机器学习和/或其他人工智能任务的专用计算系统(例如,超级计算机或HPC)。在至少一个实施例中,除了CPU,RAM,存储器和/或其他组件,功能部件或功能之外,AI系统3224(例如,NVIDIA的DGX)可以包括可以使用多个GPU 3222执行的GPU优化的软件(例如,软件栈)。在至少一个实施例中,可以在云3226中(例如,在数据中心中)实现一个或更多个AI系统3224,以执行系统3200的一些或全部基于AI的处理任务。

[0398] 在至少一个实施例中,云3226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3200的处理任务的GPU优化的平台。在至少一个实施例中,云3226可以包括一个或更多个AI系统3224,用于执行系统3200的一个或更多个基于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3226可以与利用多个GPU的应用程序编排系统3228集成,以实现应用程序和服务3120之间的无缝缩放和负载平衡。在至少一个实施例中,云3226可以负责执行系统3200的至少一些服务3120,包括计算服务3216,AI服务3218和/或可视化服务3220,如本文所述。在至少一个实施例中,云3226可以执行小批量和大批量推理(例如,执行NVIDIA的TENSOR RT),提供加速的并行计算API和平台3230(例如,NVIDIA的CUDA),执行应用程序编排系统3228(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3200提供其他功能。

[0399] 在至少一个实施例中,为了保护患者的机密性(例如,在非现场使用患者数据或记录的情况下),云3226可以包括注册表-例如深度学习容器注册表。在至少一个实施例中,注

册表可以存储用于对患者数据执行预处理,后处理或其他处理任务的应用程序的实例化的容器。在至少一个实施例中,云3226可以接收包括患者数据以及容器中的传感器数据的数据,仅对那些容器中的传感器数据执行所请求的处理,然后将结果输出和/或可视化转发给适当的参与者和/或设备(例如用于可视化或诊断的本地医疗设备),而无需提取,存储或访问患者数据。在至少一个实施例中,根据HIPAA和/或其他数据法规来保留患者数据的机密性。

[0400] 图33包括根据至少一个实施例的用于处理成像数据的部署管线3210A的示例说明。在至少一个实施例中,系统3200,尤其是部署系统3106,可以用于将一个或更多个部署管线3210A定制,更新和/或集成到一个或更多个生产环境中。在至少一个实施例中,图33的部署管线3210A包括可以由设施(例如,医院,诊所,实验室,研究环境等)处的特定用户(或用户团队)定制定义的部署管线3210A的非限制性示例。在至少一个实施例中,为了定义用于CT扫描仪3302的部署管线3210A,用户可以例如从容器注册表中选择一个或更多个应用程序,该应用程序执行关于由CT扫描仪3302生成的成像数据的特定功能或任务。在至少一个实施例中,应用程序可以作为可以利用系统3200的服务3120和/或硬件3122的容器的容器应用到部署管线3210A。此外,部署管线3210A可以包括附加处理任务或应用程序,其可以被实现为准备供应用程序使用的数据(例如DICOM适配器3202B和DICOM阅读器3306可在部署管线3210A中使用,以准备供CT重建3308,器官分割3310等使用的数据)。在至少一个实施例中,可以定制或选择部署管线3210A以用于一致的部署,一次使用或另一频率或间隔。在至少一个实施例中,用户可能期望在特定间隔上具有针对几个对象的CT重建3308和器官分割3310,并且因此可以在该时间段内部署管线3210A。在至少一个实施例中,用户可以针对来自系统3200的每个请求选择用户想要针对该请求对该数据执行处理的应用程序。在至少一个实施例中,可以以任何间隔调整部署管线3210A,并且由于系统3200内的容器结构的适应性和可伸缩性,这可以是无缝的过程。

[0401] 在至少一个实施例中,图33的部署管线3210A可以包括CT扫描仪3302,其生成患者或对象的成像数据。在至少一个实施例中,来自CT扫描仪3302的成像数据可以存储在与容纳CT扫描仪3302的设施相关的一个或更多个PACS服务器3304上。一个或更多个PACS服务器3304可以包括可以在设施处与成像模态(例如CT扫描仪3302)直接接口的软件和/或硬件组件。在至少一个实施例中,DICOM适配器3202B可以允许使用DICOM协议发送和接收DICOM对象。在至少一个实施例中,DICOM适配器3202B可以帮助准备或配置来自一个或更多个PACS服务器3304的DICOM数据,以供部署管线3210A使用。在至少一个实施例中,一旦通过DICOM适配器3202B处理了DICOM数据,管线管理器3212就可以将数据路由到部署管线3210A。在至少一个实施例中,DICOM读取器3306可以从DICOM数据(例如,如在可视化3316A中所示的原始正弦图数据)提取图像文件和任何相关联的元数据。在至少一个实施例中,可以将提取的工作文件存储在高速缓存中,以供部署管线3210A中的其他应用程序更快地处理。在至少一个实施例中,一旦DICOM读取器3306完成了提取和/或存储数据,就可以将完成信号传送到管线管理器3212。在至少一个实施例中,管线管理器3212然后可以发起或调用其他部署管线3210A中的一个或更多个应用程序或容器。

[0402] 在至少一个实施例中,一旦数据(例如,原始正弦图数据)可由CT重建3308应用程序处理,就可以执行CT重建3308应用程序和/或容器。在至少一个实施例中,CT重建3308可

以从高速缓存中读取原始正弦图数据,从原始正弦图数据中重建图像文件(例如,如可视化3316B所示),并且将得到的图像文件存储在高速缓存中。在至少一个实施例中,在重建完成时,可以向管线管理器3212发信号通知重建任务已完成。在至少一个实施例中,一旦重建完成,并且重建的图像文件可以被存储在高速缓存器(或其他存储设备)中,则器官分割3310应用程序和/或容器可以由管线管理器3212触发。在至少一个实施例中,器官分割3310应用程序和/或容器可以从缓存中读取图像文件,将图像文件规范化或转换为适合推理的格式(例如,将图像文件转换为机器学习模型的输入分辨率),然后对归一化的图像运行推理。在至少一个实施例中,为了对归一化的图像运行推理,器官分割3310应用程序和/或容器可以依赖服务3120,并且管线管理器3212和/或应用程序编排系统3228可以通过器官分割3310应用程序和/或容器来促进服务3120的使用。例如,器官分割3310应用程序和/或容器可以利用AI服务3218对归一化图像执行推理,并且AI服务3218可以利用硬件3122(例如,AI系统3224)来执行AI服务3218。在至少一个实施例中,推论的结果可以是可以存储在高速缓存(或其他存储设备)中的掩码文件(例如,如可视化3316C所示)。

[0403] 在至少一个实施例中,一旦处理DICOM数据和/或从DICOM数据中提取的数据的应用程序已经完成处理,就可以为管线管理器3212生成信号。在至少一个实施例中,管线管理器3212然后可以执行DICOM写入器3312以从高速缓存(或其他存储设备)读取结果,将结果打包成DICOM格式(例如,作为DICOM输出3314),以供产生请求的设施处的用户使用。在至少一个实施例中,然后可以将DICOM输出3314发送到DICOM适配器3202B,以准备DICOM输出3314用于存储在一个或更多个PACS服务器3304上(例如,以供设施处的DICOM查看器查看)。在至少一个实施例中,响应于对重建和分段的请求,可视化3316B和3316C可以被生成并且对于用户可用于诊断,研究和/或用于其他目的。

[0404] 尽管在部署管线3210A中被示为连续的应用程序,但是在至少一个实施例中,可以并行地处理CT重建3308和器官分割3310的应用程序。在至少一个实施例中,在应用程序彼此不依赖并且数据可用于每个应用程序的情况下(例如,在DICOM阅读器3306提取数据之后),应用程序可以在同一时间,基本同一时间执行,或者有一些重叠。在至少一个实施例中,在两个或更多个应用程序需要类似服务3120的情况下,系统3200的调度器可用于负载平衡并在各种应用程序之间分配计算或处理资源。在至少一个实施例中,在一些实施例中,并行计算平台3230可以用于对应用程序执行并行处理,以减少部署管线3210A的运行时间以提供实时结果。

[0405] 在至少一个实施例中,并且参照图34A-34B,部署系统3106可以被实现为一种或更多种虚拟仪器,以利用成像设备(例如,CT扫描仪,X射线机,MRI机器等)),测序设备,基因组设备和/或其他设备类型执行不同的功能,例如图像处理,分割,增强,AI,可视化和推理。在至少一个实施例中,系统3200可以允许创建和提供虚拟仪器,该虚拟仪器可以包括软件定义的部署管线3210,该软件定义的部署管线3210可以接收由一个或更多个设备生成的原始/未处理的输入数据并输出处理/重建的数据。在至少一个实施例中,代表虚拟仪器的部署管线3210(例如3210A和3210B)可以例如通过利用机器学习模型将智能实现到管线中,以向系统提供容器化的推理支持。在至少一个实施例中,虚拟仪器可以执行任意数量的容器,每个容器都包括应用程序的实例。在至少一个实施例中,例如在需要实时处理的地方,表示虚拟仪器的部署管线3210可以是静态的(例如,可以设置容器和/或应用程序),而在其他示

例中,可以从应用程序或资源池(例如,在容器注册表中)选择(例如,基于每个请求)虚拟仪器的容器和/或应用程序。

[0406] 在至少一个实施例中,系统3200可以被实例化或执行为在例如部署在放射线机,成像设备和/或设施中的其他设备类型旁边或与之通信的计算系统中的设施处的一个或更多个虚拟仪器。然而,在至少一个实施例中,可以在本地数据中心(例如,位于本地的数据中心)和/或在云环境中(例如,在云3226中)的设备本身的计算系统(例如,与成像设备集成在一起的计算系统)内实例化或执行本地安装。在至少一个实施例中,在一些示例中,可以由超级计算机或其他HPC系统实例化作为虚拟仪器操作的部署系统3106。在至少一个实施例中,本地安装可以允许用于实时处理的高带宽使用(例如,通过较高吞吐量的本地通信接口,例如以太网上的RF)。在至少一个实施例中,在虚拟仪器支持超声设备或其他成像模态的情况下,实时或近实时处理可能特别有用,其中期望或要求立即可视化以进行准确的诊断和分析。在至少一个实施例中,当本地需求超过内部部署容量或能力时,云计算架构可能能够动态爆发到云计算服务提供商或其他计算集群。在至少一个实施例中,如在此关于训练系统3104所描述的,云架构在被实施时可以被调整用于训练神经网络或其他机器学习模型。在至少一个实施例中,具有适当的训练管线,机器学习模型在处理其支持的设备中的其他数据时可能会不断学习和改进。在至少一个实施例中,可以使用附加数据,新数据,现有机器学习模型和/或新的或更新的机器学习模型来持续地改进虚拟仪器。

[0407] 在至少一个实施例中,计算系统可以包括本文描述的硬件3122中的一些或全部,并且硬件3122可以以包括在设备内的多种方式中的任一种来分布,作为耦合到并位于设备附近的计算设备的一部分,在设施的本地数据中心中和/或云3226中。在至少一个实施例中,因为部署系统3106和关联的应用程序或容器是在软件中创建的(例如,作为应用程序的离散容器化实例),虚拟仪器的行为,操作和配置以及虚拟仪器生成的输出可以根据需要进行修改或自定义,而无需更改或改变虚拟仪器支持的设备的原始输出。

[0408] 图34A包括根据至少一个实施例的支持超声设备的虚拟仪器的示例数据流程图。在至少一个实施例中,部署管线3210B可以利用系统3200的服务3120中的一个或更多个。在至少一个实施例中,部署管线3210B和服务3120可以利用本地或云3226中的系统的硬件3122。在一个实施例中,尽管未示出,但是过程3400可以通过管线管理器3212,应用程序编排系统3228和/或并行计算平台3230来促进。

[0409] 在至少一个实施例中,过程3400可以包括从超声设备3402接收成像数据。在至少一个实施例中,可以以DICOM格式(或其他格式,例如RIS,CIS,REST兼容,RPC,原始等)存储在一个或更多个PACS服务器上,并可以由系统3200接收,以通过部署管线3210选择或定制用于超声设备3402的虚拟仪器(例如,虚拟超声)的进行处理。在至少一个实施例中,可以直接从成像设备(例如,超声设备3402)接收成像数据,并由虚拟仪器对其进行处理。在至少一个实施例中,通信地耦合在成像设备和虚拟仪器之间的传感器或其他信号转换器可以将由成像设备生成的信号数据转换成可以由虚拟仪器处理的图像数据。在至少一个实施例中,原始数据和/或图像数据可以被应用程序于DICOM读取器3306,以提取数据供部署管线3210B的应用程序或容器使用。在至少一个实施例中,DICOM读取器3306可以利用数据增强库3414(例如,NVIDIA的DALI)作为服务3120(例如,作为一种或更多种计算服务3216之一)来提取,调整大小,重新缩放和/或以其他方式准备供应用程序或容器使用的数据。

[0410] 在至少一个实施例中,一旦准备好数据,就可以执行重建3406应用程序和/或容器以将来自超声设备3402的数据重建成图像文件。在至少一个实施例中,在重建3406之后或与重建3406同时,可以执行检测3408应用程序和/或容器以用于异常检测,对象检测,特征检测和/或与数据有关的其他检测任务。在至少一个实施例中,可以在检测3408期间使用在重建3406期间生成的图像文件以识别异常,对象,特征等。在至少一个实施例中,检测3408应用程序可以利用推理引擎3416(例如,作为一个或更多个AI服务3218之一)对数据执行推理以生成检测。在至少一个实施例中,检测3408应用程序可以执行或调用一个或更多个机器学习模型(例如,来自训练系统3104)。

[0411] 在至少一个实施例中,一旦重构3406和/或检测3408完成,从这些应用程序和/或容器输出的数据可用于生成可视化3410,例如在工作站或显示终端上显示的可视化3412(例如,灰度输出)。在至少一个实施例中,可视化可以允许技术人员或其他用户相对于超声设备3402可视化部署管线3210B的结果。在至少一个实施例中,可以通过利用系统3200的渲染组件3418来执行可视化3410(例如,一个或更多个可视化服务3220之一)。在至少一个实施例中,渲染组件3418可以执行2D,OpenGL或光线跟踪服务以生成可视化3412。

[0412] 图34B包括根据至少一个实施例的支持CT扫描仪的虚拟仪器的示例数据流程图。在至少一个实施例中,部署管线3210C可以利用系统3200的服务3120中的一个或更多个。在至少一个实施例中,部署管线3210C和服务3120可以利用本地或云3226中的系统的硬件3122。在一个实施例中,虽然未示出,但是可以通过管线管理器3212,应用程序编排系统3228和/或并行计算平台3230来促进过程3420。

[0413] 在至少一个实施例中,过程3420可以包括CT扫描仪3422,其生成可以由DICOM读取器3306(例如,在处理等之后直接经由PACS服务器3304直接)接收的原始数据。在至少一个实施例中,虚拟CT(由部署管线3210C实例化)可以包括用于监视患者(例如,患者运动检测AI 3426)和/或用于调整或优化CT扫描仪3422的暴露(例如,使用曝光控制AI 3424)的第一实时管线。在至少一个实施例中,一个或更多个应用程序(例如3424和3426)可以利用服务3120,例如一个或更多个AI服务3218。在至少一个实施例中,曝光控制AI 3424应用程序(或容器)的输出和/或患者运动检测AI 3426应用程序(或容器)可以用作对CT扫描仪3422和/或技术人员的反馈,以调整曝光(或CT扫描仪3422的其他设置)和/或通知患者减少运动。

[0414] 在至少一个实施例中,部署管线3210C可以包括用于分析由CT扫描器3422生成的数据的非实时管线。在至少一个实施例中,第二管线可以包括CT重建3308应用程序和/或容器,粗略检测AI 3428应用程序和/或容器,精细检测AI 3432应用程序和/或容器(例如,通过粗略检测AI 3428检测到某些结果的情况),可视化3430应用程序和/或容器以及DICOM编写器3312(和/或其他数据类型编写器,例如RIS,CIS,REST兼容,RPC,原始文件等)应用程序和/或容器。在至少一个实施例中,由CT扫描器3422产生的原始数据可以通过部署管线3210C的管线(被实例化为虚拟CT仪器)以产生结果。在至少一个实施例中,来自DICOM写入器3312的结果可以被发送用于显示和/或可以被存储在一个或更多个PACS服务器3304上以供技术人员,从业者或其他用户以后检索,分析或显示。

[0415] 图35A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3500的数据流程图。在至少一个实施例中,可以使用作为非限制性示例的图32的系统3200来执行过程3500。在至少一个实施例中,过程3500可以利用本文所述的系统3200的服

务3120和/或硬件3122。在至少一个实施例中,由过程3500生成的改进模型3512可以由部署系统3106针对部署管线3210中的一个或更多个容器化应用程序执行。

[0416] 在至少一个实施例中,模型训练3114可以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3506,和/或与输入数据相关联的新的地面实况数据)对初始模型3504(例如,预训练模型)进行重新训练或更新。在至少一个实施例中,为了重新训练或更新初始模型3504,可以重置或删除初始模型3504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3504可以具有从先前训练中保留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3114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3114期间,通过重置或替换初始模型3504的输出或损失层,可以在新的客户数据集3506(例如,图31的图像数据3108)上生成预测时,基于与输出或损失层的准确性相关联的损失计算来为新数据集更新和重新调整参数。

[0417] 在至少一个实施例中,可以将预训练的模型3206存储在数据存储或注册表中(例如,图31的模型注册表3124)。在至少一个实施例中,预训练模型3206可能已经至少部分地在除设施执行过程3500之外的一个或更多个设施处被训练。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,预训练模型3206可能已在本地使用本地生成的客户或患者数据进行了训练。在至少一个实施例中,可以使用云3226和/或其他硬件3122来训练预训练模型3206,但是机密的,受隐私保护的患者数据可以不被传送到云3226(或其他非本地硬件),或被其使用,或被其任何部件访问。在至少一个实施例中,在使用来自一个以上设施的患者数据来训练预训练模型3206的情况下,在针对来自另一设施的患者或客户数据进行训练之前,可以针对每个设施分别对预训练模型3206进行训练。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),或客户或患者数据包含在公共数据集中的情况下,来自任何数量的设施的客户或患者数据可以用于在例如数据中心或其他云计算基础设施中在本地和/或非本地训练预训练的模型3206。

[0418] 在至少一个实施例中,在选择的应用程序以在部署管线3210中使用时,用户可能也选择的机器学习模型以用于特定的应用程序。在至少一个实施例中,用户可能没有要使用的模型,因此用户可以选择要与应用程序一起使用的预训练模型3206。在至少一个实施例中,未优化预训练模型3206以在用户设施的客户数据集3506上生成准确结果(例如,基于患者多样性,人口统计学,所使用的医学成像设备的类型等)。在至少一个实施例中,在将预训练的模型3206部署到部署管线3210中与一个或更多个应用程序一起使用之前,可以对预训练的模型3206进行更新,重新训练和/或微调以在各个设施处使用。

[0419] 在至少一个实施例中,用户可以选择要被更新,重新训练和/或微调的预训练模型3206,并且预训练模型3206可以被称为用于训练流程3500中系统3504的初始模型3104。在至少一个实施例中,客户数据集3506(例如,成像数据,基因组数据,测序数据或由设施中的设备生成的其他数据类型)可以用于在初始模型3504上执行模型训练3114(可以包括但不限于,转移学习)以生成精炼模型3512。在至少一个实施例中,可以由训练系统3104生成与客户数据集3506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,从业人员在设施中(例如,如图31中标记的临床数据3112所示)生成地面

实况数据。

[0420] 在至少一个实施例中,在一些示例中,AI辅助注释3110可以用于生成地面实况数据。在至少一个实施例中,AI辅助注释3110(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)来生成针对客户数据集的建议的或预测的地面实况数据。在至少一个实施例中,用户3510可以在计算设备3508上的用户界面(图形用户界面(GUI))内使用注释工具。

[0421] 在至少一个实施例中,用户3510可以经由计算设备3508与GUI交互以编辑或微调(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。

[0422] 在至少一个实施例中,一旦顾客数据集3506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3114期间被使用以生成精炼模型3512。在至少一个实施例中,顾客数据集3506可以被多次应用程序到初始模型3504,并且地面实况数据可以被用来更新初始模型3504的参数,直到对于精炼模型3512达到可接受的精度水平为止。在至少一个实施例中,一旦生成了精炼模型3512,就可以在用于执行关于医学成像数据的一个或更多个处理任务的设施处的一个或更多个部署管线3210内部署精炼模型3512。

[0423] 在至少一个实施例中,可以将精炼模型3512上传到模型注册表3124中的预训练模型3206,以由另一设施选择。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任意次数地进一步完善精炼模型3512,以生成更通用的模型。

[0424] 图35B是根据至少一个实施例的,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3532的示例说明。在至少一个实施例中,可以基于客户端-服务器架构3532来实例化AI辅助的注释工具3536。在至少一个实施例中,成像应用程序中的注释工具3536可以帮助放射线医生,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,该软件工具帮助用户3510识别原始图像3534(例如,在3D MRI或CT扫描中)的特定感兴趣器官上的一些极端点,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3538被存储在数据存储器中,并且被用作(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3508发送用于AI辅助注释3110的极限点时,深度学习模型例如可以接收该数据作为输入并且返回分割器官或异常的推理结果。在至少一个实施例中,预先实例化的注释工具,例如可以通过对诸如注释助手服务器3540之类的服务器进行API调用(例如,API调用3544)来增强图35B中的AI辅助注释工具3536B,例如,该服务器可以包括存储在注释模型注册表中的一组预训练模型3542。在至少一个实施例中,注释模型注册表可以存储预训练的模型3542(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。这些模型可以通过使用训练管线3204来进一步更新。在至少一个实施例中,随着添加新的标记的临床数据3112,预安装的注释工具可以随着时间的推移而改进。

[0425] 一个或更多个硬件结构615用于执行一个或更多个实施例。下面结合图6A和/或6B提供关于硬件结构(x)615的细节。

[0426] 其他变型在本公开的精神内。因此,尽管公开的技术易于进行各种修改和替代构造,但是其某些示出的实施例在附图中示出并且已经在上面进行了详细描述。但是,应当理

解,无意将公开内容限制为所公开的一种或更多种特定形式,而是相反,其意图是涵盖落入如所附权利要求书所定义的公开内容的精神和范围内的所有修改、替代构造和等同物。

[0427] 除非另有说明,除非另有说明或显然与环境矛盾,否则在描述所公开的实施例的环境中(特别是在所附权利要求的环境中)对术语“一”,“一个”和“该”以及类似指代的使用应解释为涵盖单数和复数,并且不作为术语的定义。术语“包含”,“具有”,“包括”和“内含”应解释为开放式术语(意思是“包括但不限于”)。术语“连接”在未经修改时指的是物理连接,应理解为部分或全部包含在,连接到或连接在一起的部分或全部,即使有任何介入。除非在此另外指出,否则本文中数值范围的引用仅旨在用作分别指代落入该范围内的每个单独值的速记方法,并且每个单独值都被并入说明书中,就如同其在本文中被单独叙述一样。除非环境另外指出或矛盾,否则术语“组”(例如“一组项目”)或“子集”的使用应解释为包括一个或更多个成员的非空集合。此外,除非环境另外指出或矛盾,否则相应集合的术语“子集”不一定表示相应集合的适当子集,而是子集和相应集合可以相等。

[0428] 除非以其他方式明确指出或与环境明显矛盾,否则诸如“A、B和C中的至少一个”或“A、B和C的至少一个”形式的词组等联合语言在环境中理解为通常用来表示项目,术语等可以是A或B或C,也可以是A和B和C集合的任何非空子集。例如,在具有三个成员,连接短语“A、B和C中的至少一个”和“A、B和C的至少一个”是指以下任意集合:{A},{B},{C},{A,B},{A,C},{B,C},{A,B,C}。因此,这种联合语言通常不意图暗示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。另外,除非另有说明或与环境矛盾,否则术语“多个”表示复数的状态(例如,“多个项目”表示多个项目)。复数是至少两个项目,但是当明确地或通过环境指示时可以是多个。此外,除非另有说明或从环境中清楚得知,否则短语“基于”是指“至少部分基于”而不是“仅基于”。

[0429] 除非本文另外指出或与环境明显矛盾,否则本文描述的过程的操作可以任何合适的顺序执行。在至少一个实施例中,诸如本文所述的那些过程(或其变形和/或其组合)的过程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,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)通过硬件或其组合在一个或更多个处理器上共同执行。在至少一个实施例中,代码例如以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更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),但包括暂时性信号的收发器中的非暂时性数据存储电路(例如,缓冲区、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)被存储在其上存储有可执行指令(或其他存储器以存储可执行指令)的一组一个或更多个非暂时性计算机可读存储介质上,当由计算机系统的一个或更多个处理器执行(即,由于被执行)而导致的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以及缺少所有代码的多个非暂时性计算机可读存储介质的一个或更多个单个非暂时性存储介质,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,并且主中央处理单元(“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指令的不同子

集。

[0430] 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描述的过程的操作的一个或更多个服务,并且这样的计算机系统配置有能够实现操作的适用的硬件和/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,并且在另一实施例中,是一种分布式计算机系统,其包括以不同方式操作的多个设备,使得分布式计算机系统执行本文所述的操作,并且使得单个设备不执行所有操作。

[0431] 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,并且不对公开的范围构成限制,除非另有声明。说明书中的任何语言都不应被解释为表示对于实施公开必不可少的任何未要求保护的要素。

[0432] 本文引用的所有参考文献,包括出版物、专利申请和专利,均以引用的方式并入本文,如同每个参考文献被单独且具体地指出以引用的方式并入本文一样。

[0433] 在描述和权利要求中,可以使用术语“耦合”和“连接”及其派生词。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”可用于指示两个或更多个元件彼此直接或间接物理或电接触。“耦合”也可能意味着两个或更多个元素彼此不直接接触,但仍彼此协作或交互。

[0434] 除非另外特别说明,否则可以理解,在整个说明书中,诸如“处理”、“计算”、“运算”、“确定”等,是指计算机或计算系统的动作和/或过程。或类似的电子计算设备,将计算系统的寄存器和/或存储器中表示为物理量(例如电子)的数据处理和/或转换为类似表示为计算系统的存储器、寄存器或其他此类信息存储、传输或显示设备中的物理量的其他数据。

[0435] 以类似的方式,术语“处理器”可以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成可以存储在寄存器和/或存储器中的其他电子数据的任何设备或设备的一部分。作为非限制性示例,“处理器”可以是CPU或GPU。“计算平台”可以包括一个或更多个处理器。如本文所使用的,“软件”过程可以包括例如随时间执行工作的软件和/或硬件实体,诸如任务、线程和智能代理。而且,每个过程可以指代多个过程,以连续地或间歇地顺序地或并行地执行指令。因为系统可以体现一种或更多种方法并且方法可以被认为是系统,术语“系统”和“方法”在本文中可互换使用。

[0436] 在本文件中,可以参考获得、获取、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中。可以以多种方式来完成获得、获取、接收或输入模拟和数字数据,例如通过接收作为函数调用或对应用程序接口的调用的参数的数据。在一些实施方式中,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串行或并行接口传输数据来完成。在另一实施方式中,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获得、获取、接收或输入模拟或数字数据的过程。也可以参考提供、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作为函数调用的输入或输出参数,应用程序编程接口或进程间通信机制的参数进行传输来实现。

[0437] 尽管上面的讨论阐述了所描述的技术的示例实现,但是其他架构可以用于实现所描述的功能,并且意图在本公开的范围内。此外,尽管出于讨论目的在上面定义了具体的责任分配,但是根据情况,可以以不同的方式分配和划分各种功能和职责。

[0438] 此外,尽管已经用针对结构特征和/或方法动作的语言描述了主题,但是应该理解,所附权利要求书所要求保护的主题不必限于所描述的特定特征或动作。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。

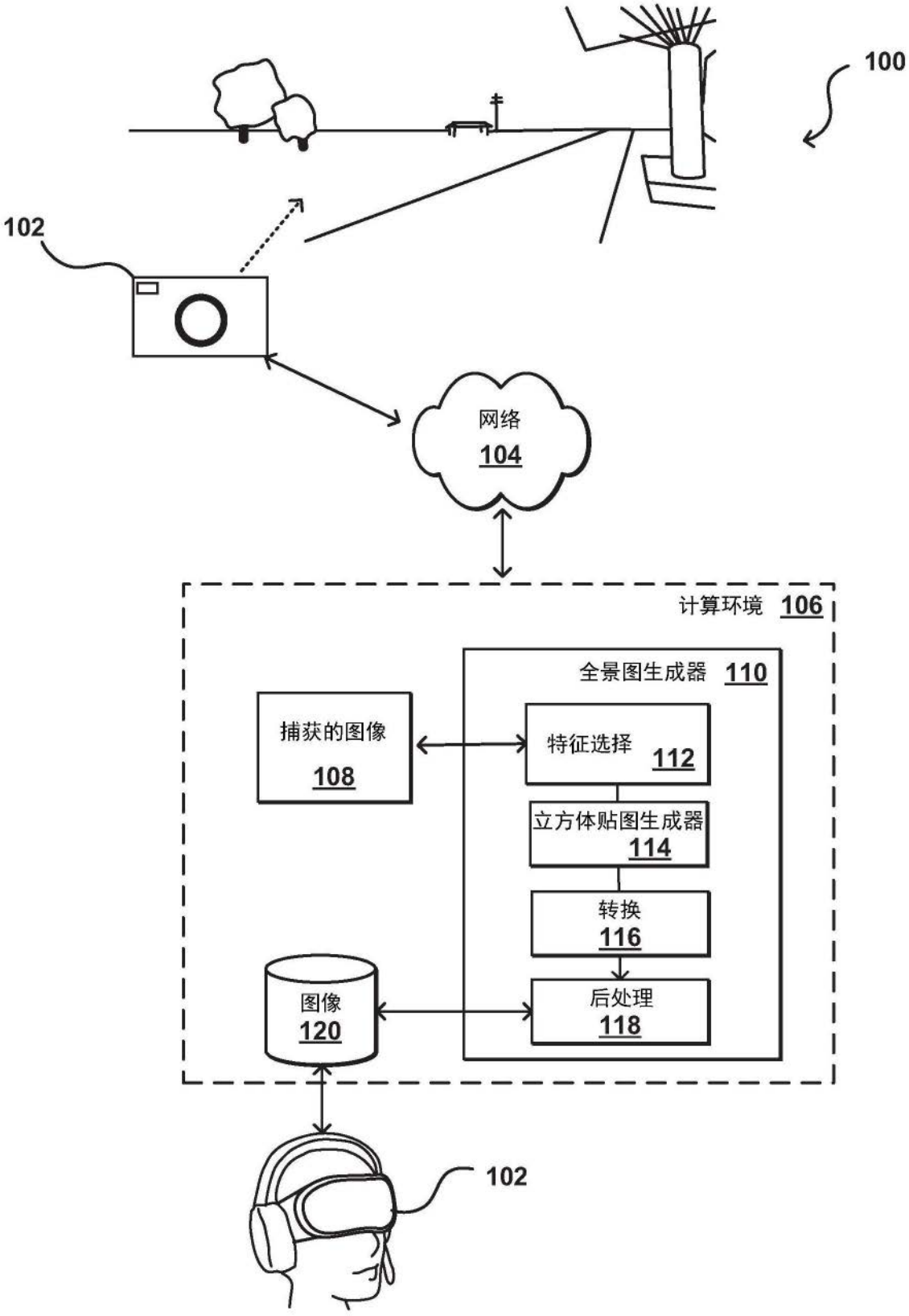

100
102
网络
104
计算环境 106
全景图生成器 110
捕获的图像
108
特征选择 112
立方体贴图生成器
114
转换
116
图像
120
后处理
118
102

图1

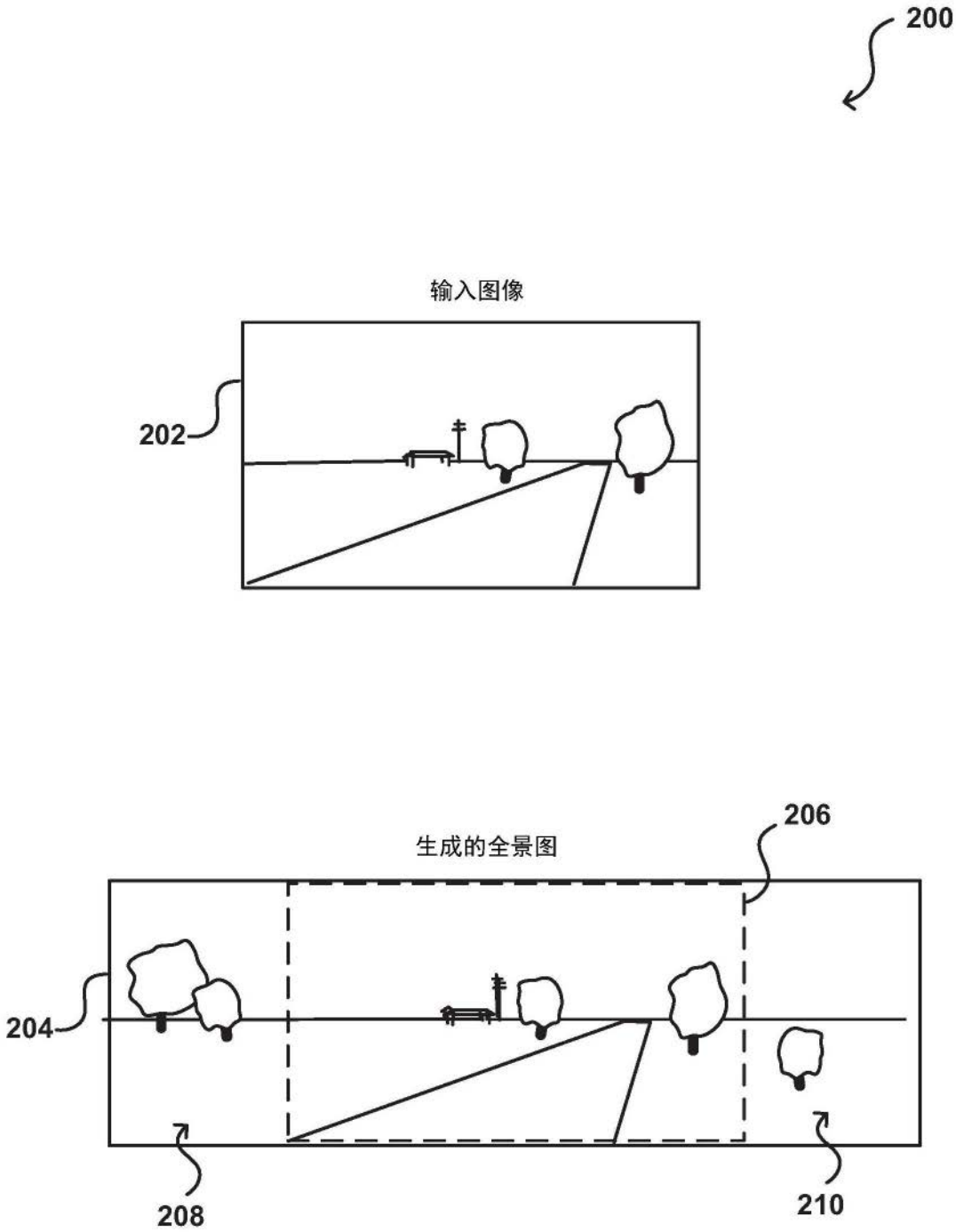


图2

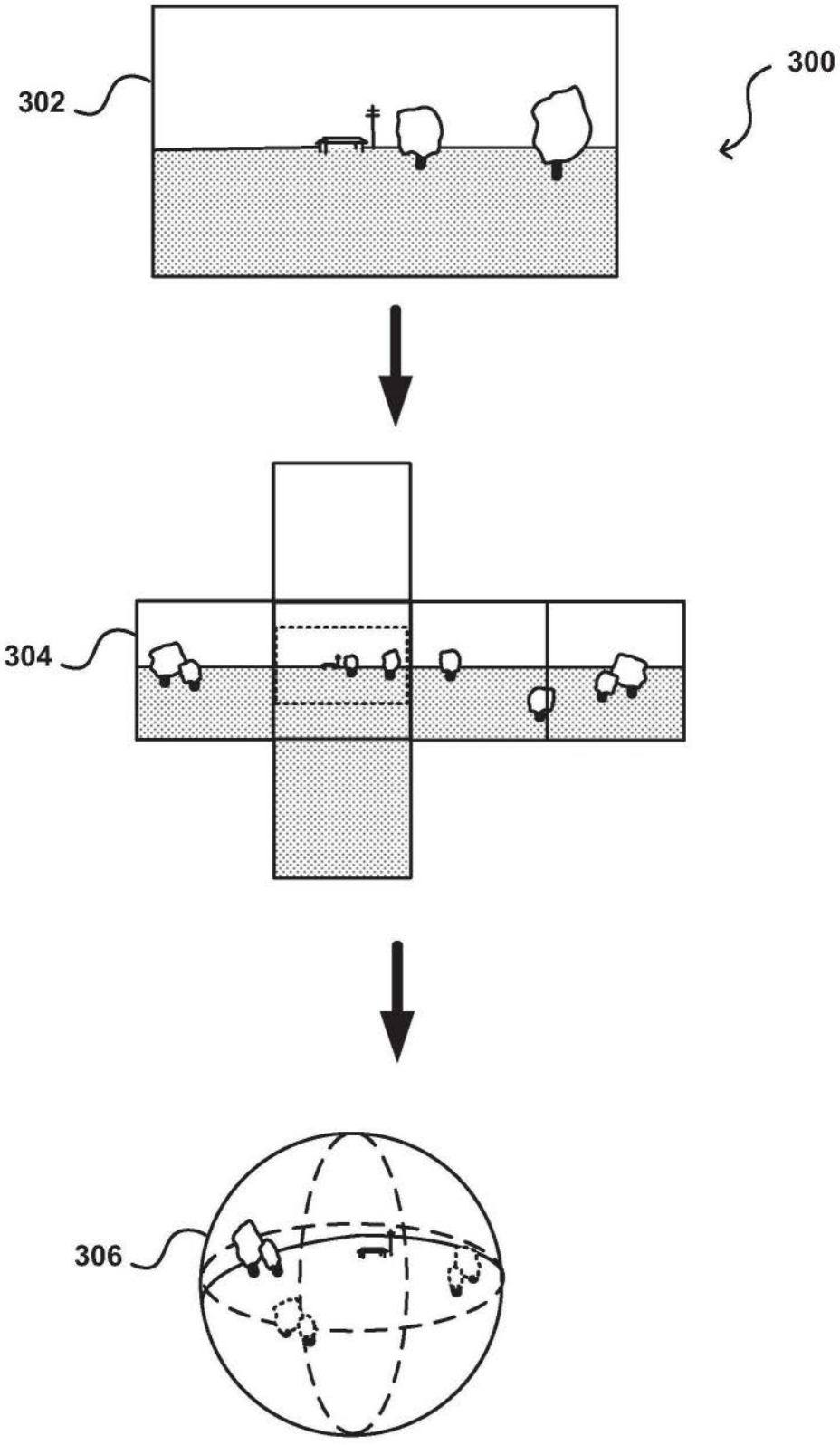


图3

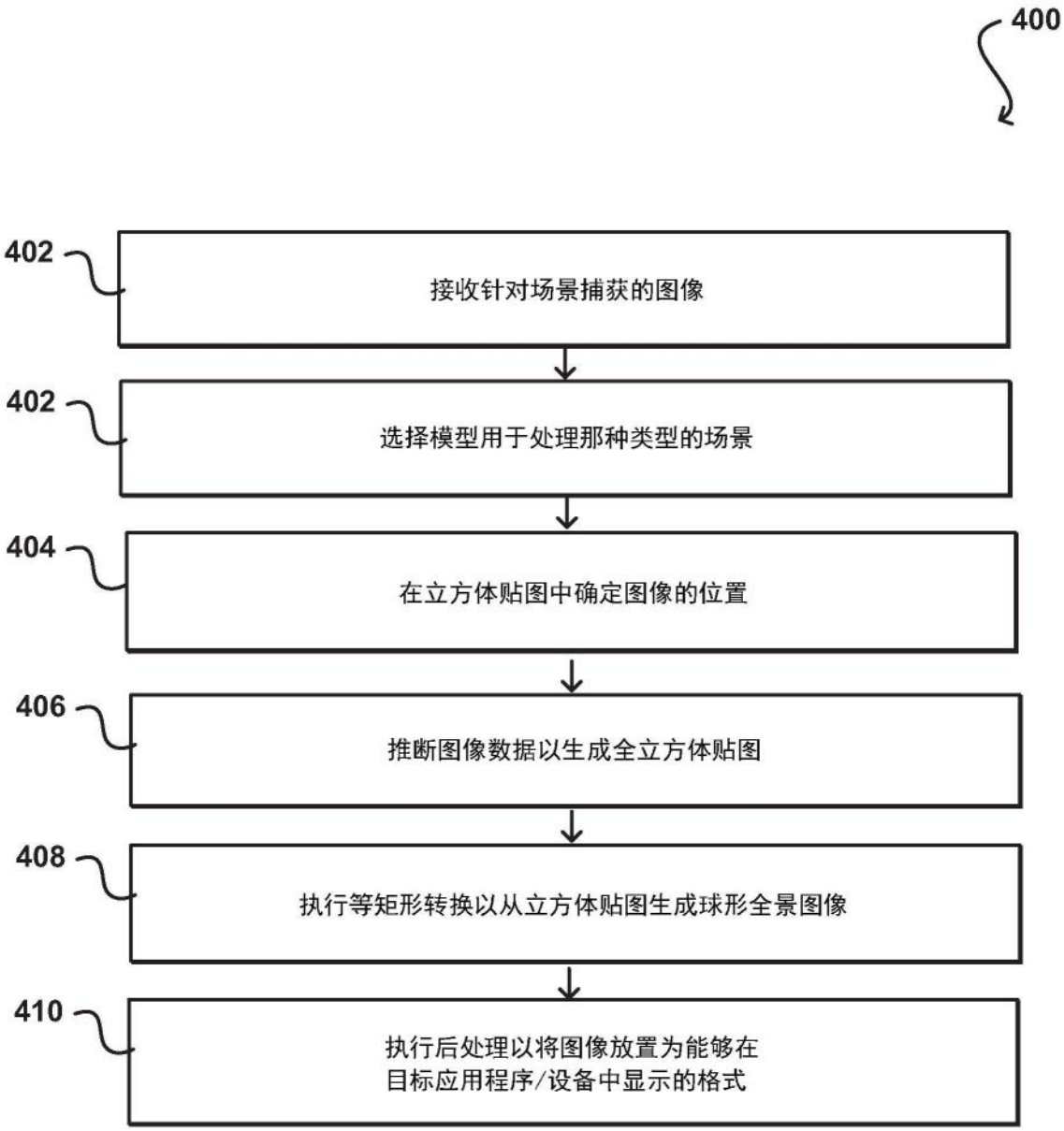

400
402
接收针对场景捕获的图像
402
选择模型用于处理那种类型的场景
404
在立方体贴图中确定图像的位置
406
推断图像数据以生成全立方体贴图
408
执行等矩形转换以从立方体贴图生成球形全景图像
410
执行后处理以将图像放置为能够在
目标应用程序/设备中显示的格式

图4

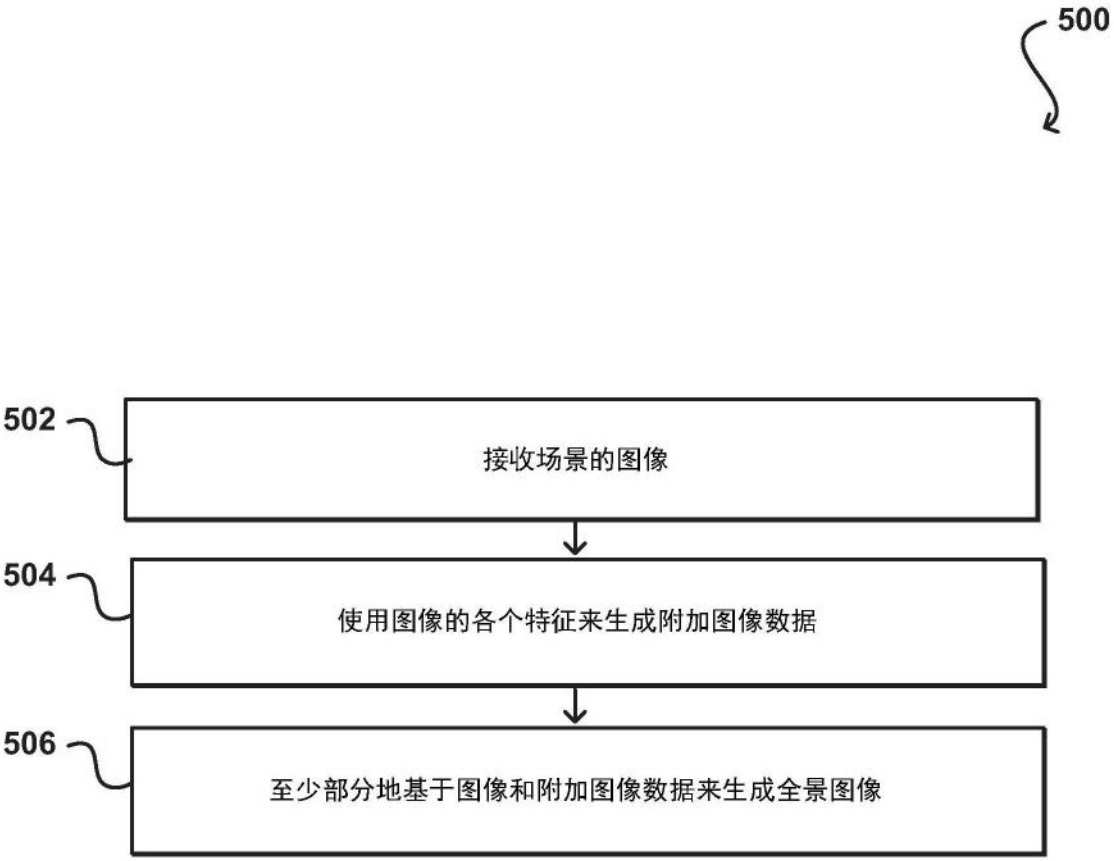

500
502
接收场景的图像
504
使用图像的各个特征来生成附加图像数据
506
至少部分地基于图像和附加图像数据来生成全景图像

图5

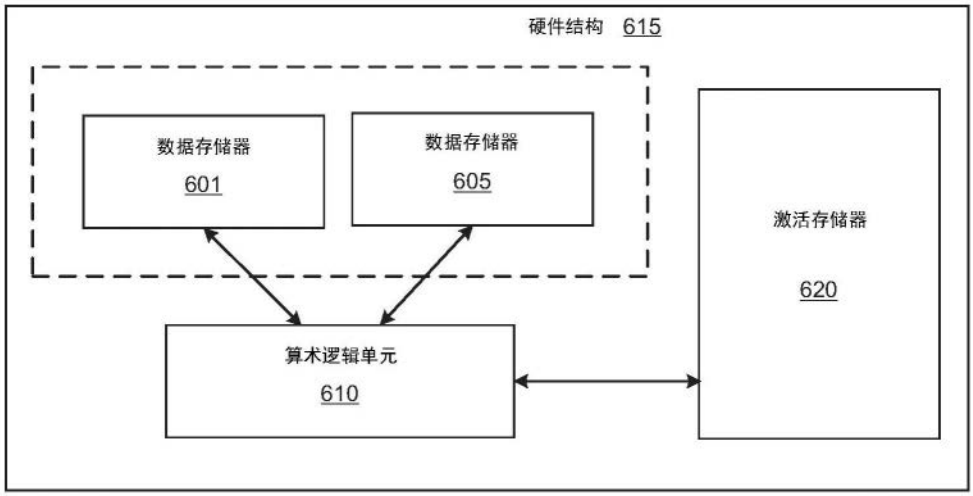

硬件结构 615
数据存储器
601
数据存储器
605
激活存储器
620
算术逻辑单元
610

图6A

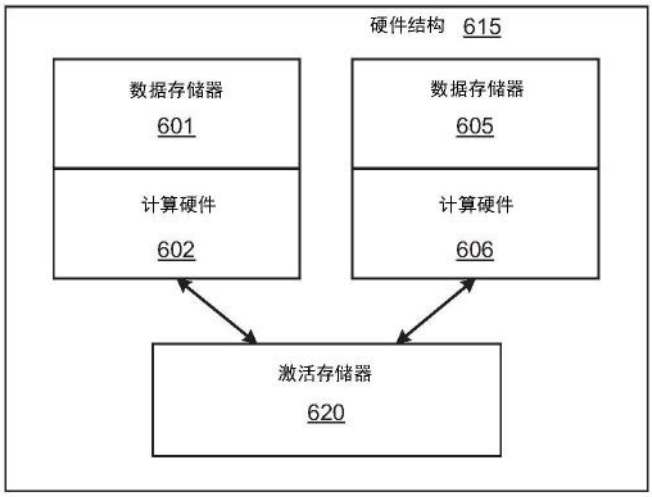


图6B

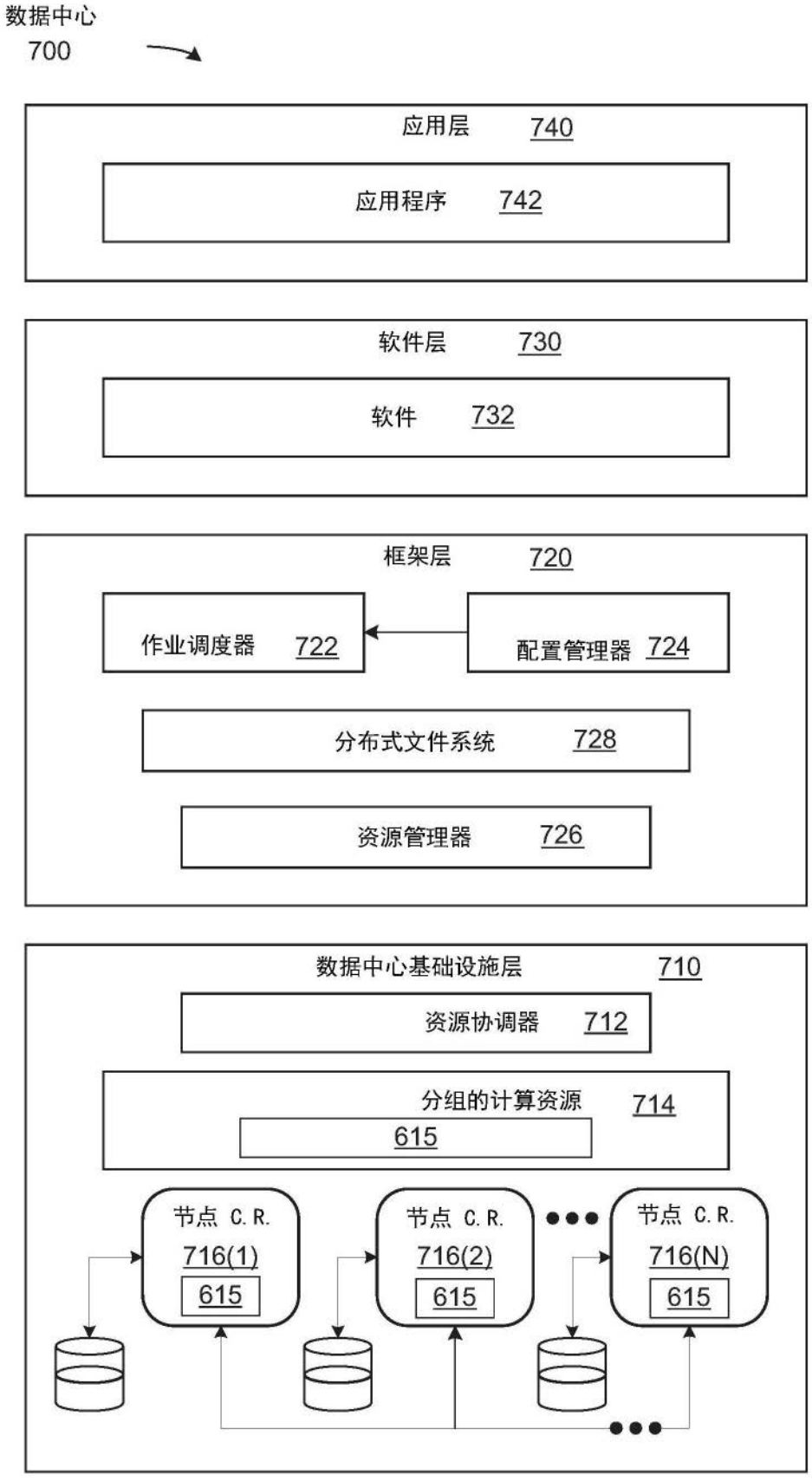

数据中心
700
应用层 740
应用程序 742
软件层 730
软件 732
框架层 720
作业调度器 722
配置管理器 724
分布式文件系统 728
资源管理器 726
数据中心基础设施层 710
资源协调器 712
分组的计算资源 714
615
节点 C. R.
716(1)
615
节点 C. R.
716(2)
615
节点 C. R.
716(N)
615

图7

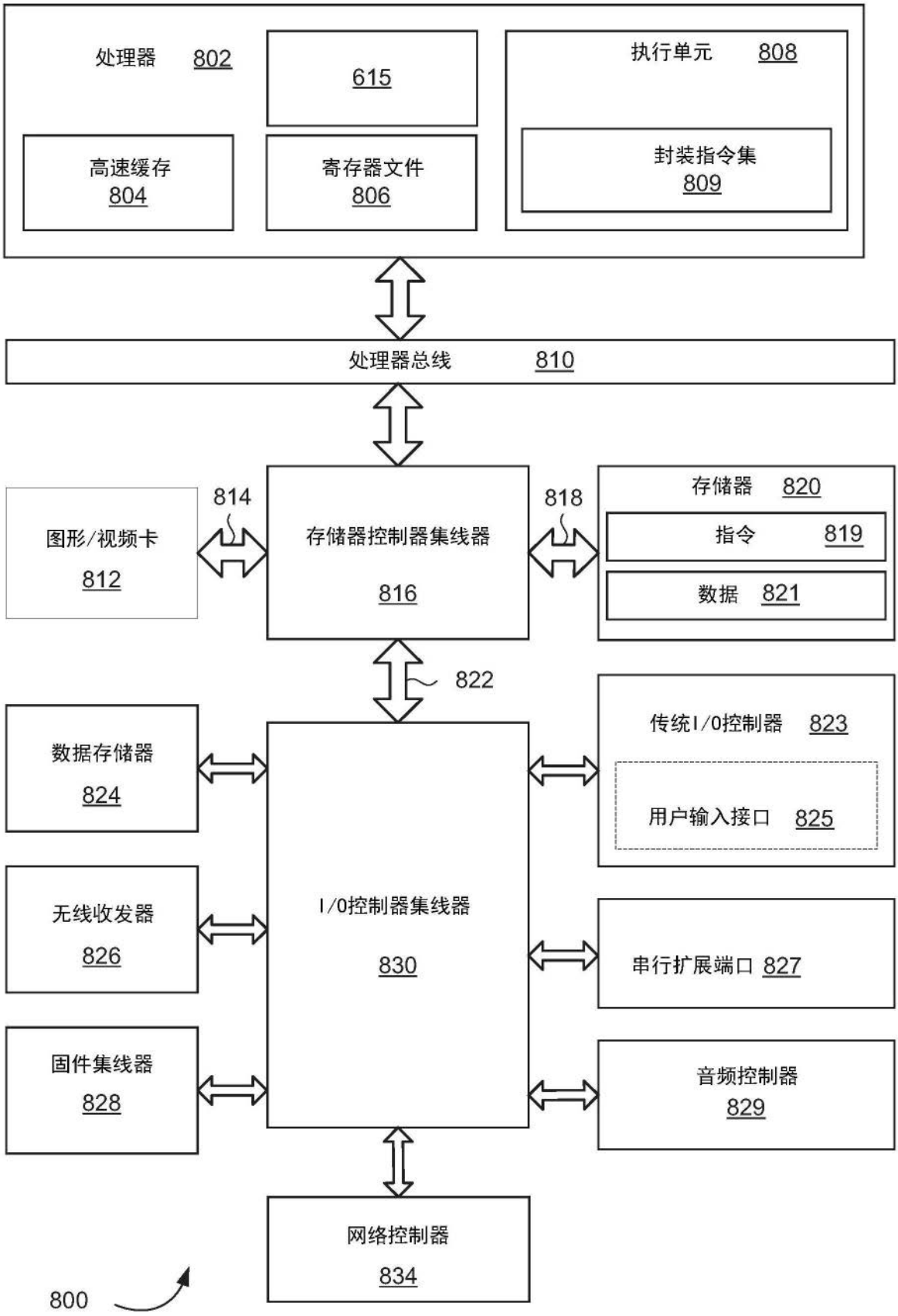

处理器 802
615
执行单元 808
高速缓存 804
寄存器文件 806
封装指令集 809
处理器总线 810
814
818
图形/视频卡 812
存储器控制器集线器 816
存储器 820
指令 819
数据 821
822
数据存储器 824
传统I/O控制器 823
用户输入接口 825
无线收发器 826
I/O控制器集线器 830
串行扩展端口 827
固件集线器 828
音频控制器 829
网络控制器 834
800

图8

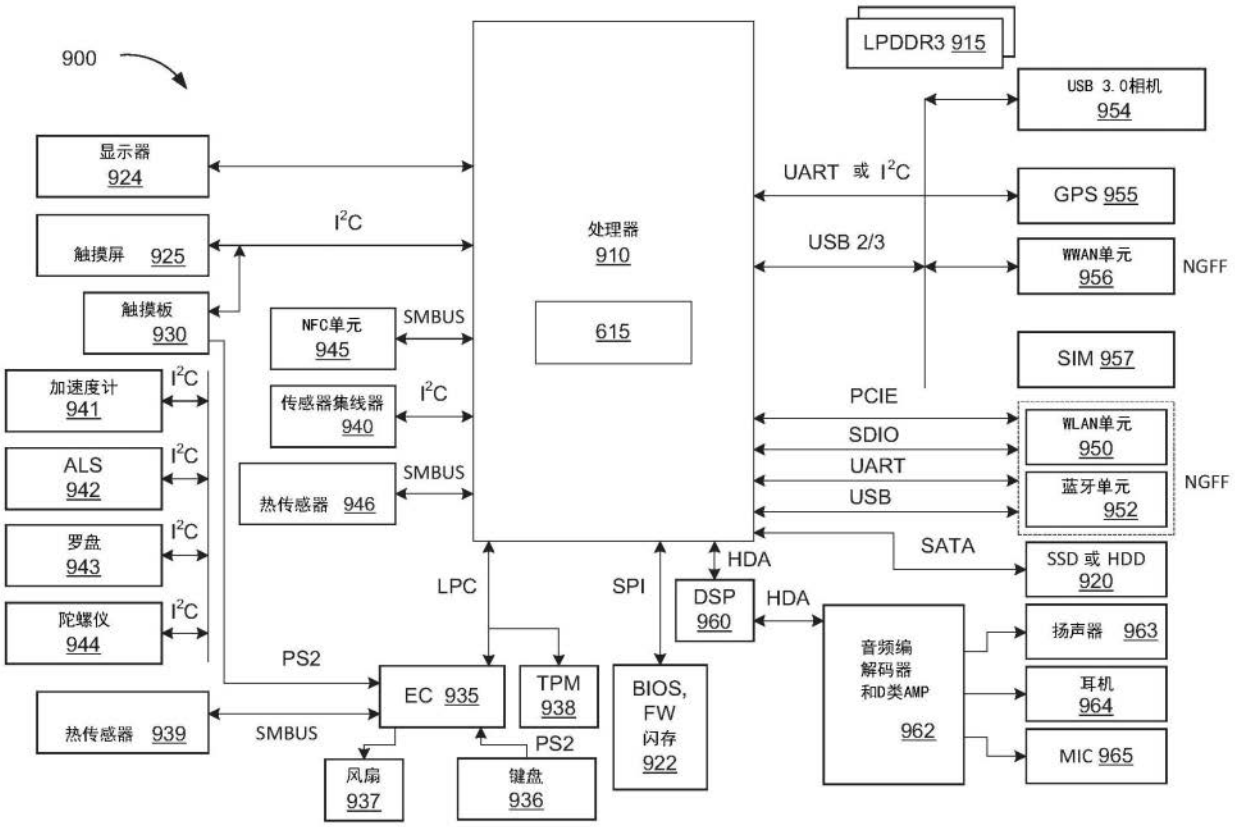

900
显示器 924
触摸屏 925
触摸板 930
加速度计 941
ALS 942
罗盘 943
陀螺仪 944
热传感器 939
NFC单元 945
传感器集线器 940
热传感器 946
处理器 910
615
LPDDR3 915
USB 3.0相机 954
GPS 955
WWAN单元 956
NGFF
SIM 957
WLAN单元 950
蓝牙单元 952
NGFF
SSD 或 HDD 920
扬声器 963
耳机 964
MIC 965
音频编解码器和D类AMP 962
DSP 960
BIOS, FW 闪存 922
EC 935
TPM 938
风扇 937
键盘 936
I²C
SMBUS
UART 或 I²C
USB 2/3
PCIE
SDIO
UART
USB
SATA
HDA
LPC
SPI
PS2

图9

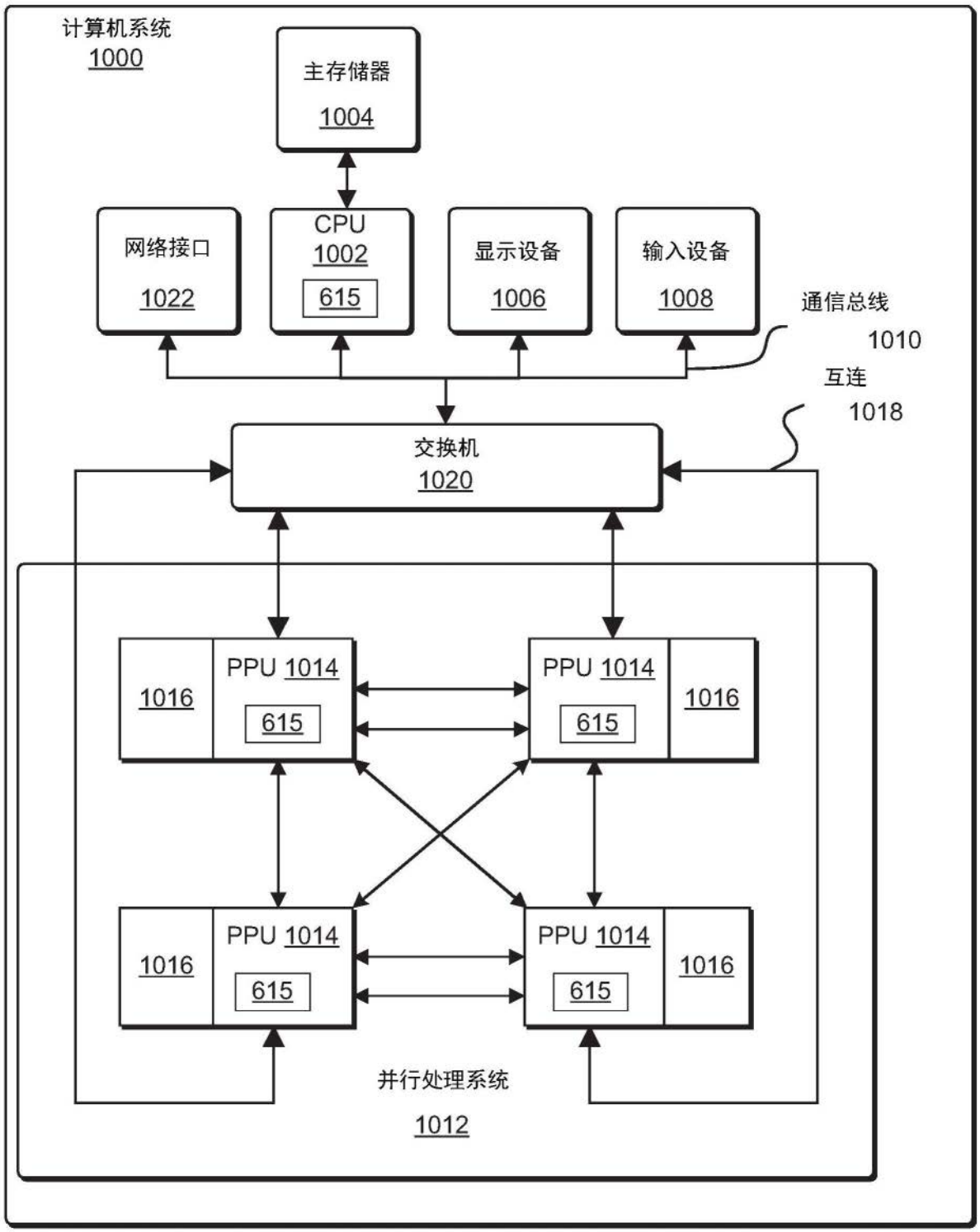

计算机系统
1000
主存储器
1004
网络接口
1022
CPU
1002
615
显示设备
1006
输入设备
1008
通信总线
1010
互连
1018
交换机
1020
1016
PPU 1014
615
PPU 1014
615
1016
1016
PPU 1014
615
PPU 1014
615
1016
并行处理系统
1012

图10

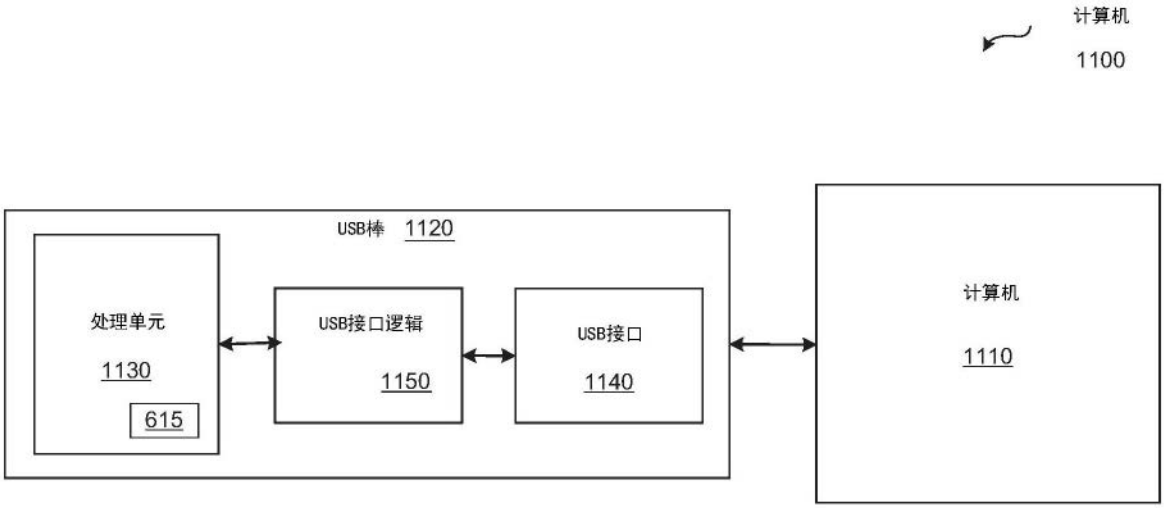

计算机
1100
USB棒 1120
处理单元
1130
615
USB接口逻辑
1150
USB接口
1140
计算机
1110

图11

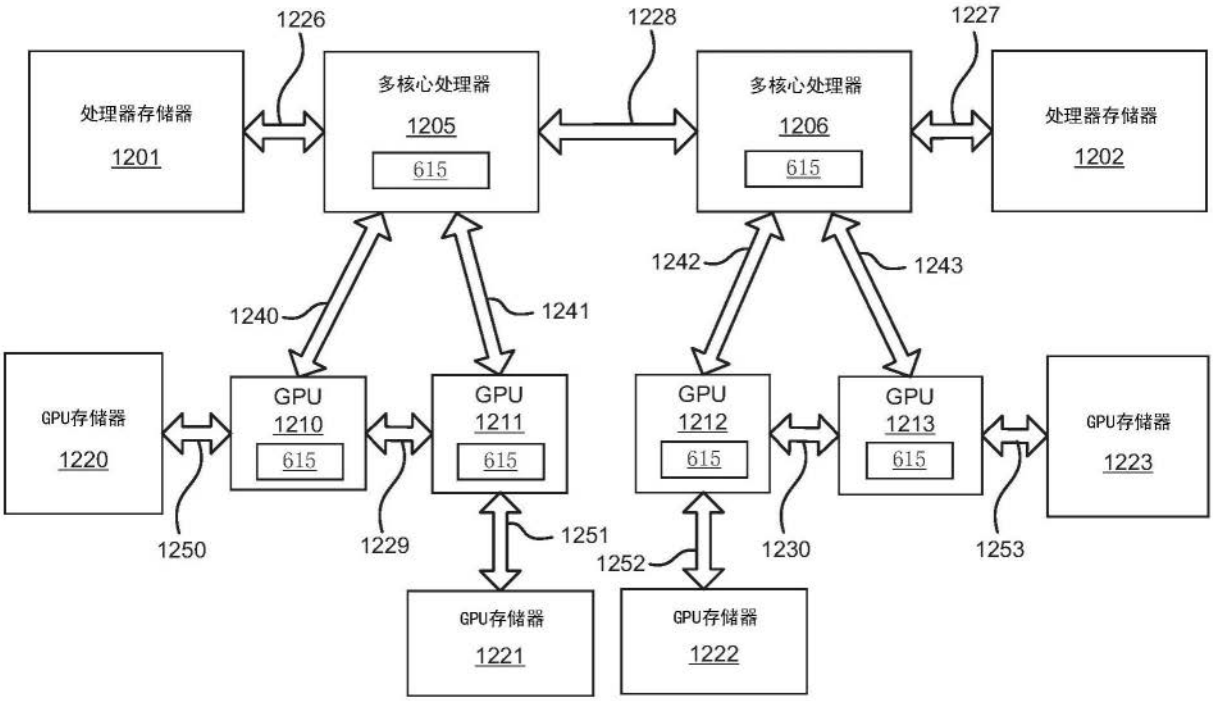

1226
1228
1227
处理器存储器
1201
多核心处理器
1205
615
多核心处理器
1206
615
处理器存储器
1202
1242
1243
1240
1241
GPU存储器
1220
GPU
1210
615
GPU
1211
615
GPU
1212
615
GPU
1213
615
GPU存储器
1223
1250
1229
1251
1252
1230
1253
GPU存储器
1221
GPU存储器
1222

图12A

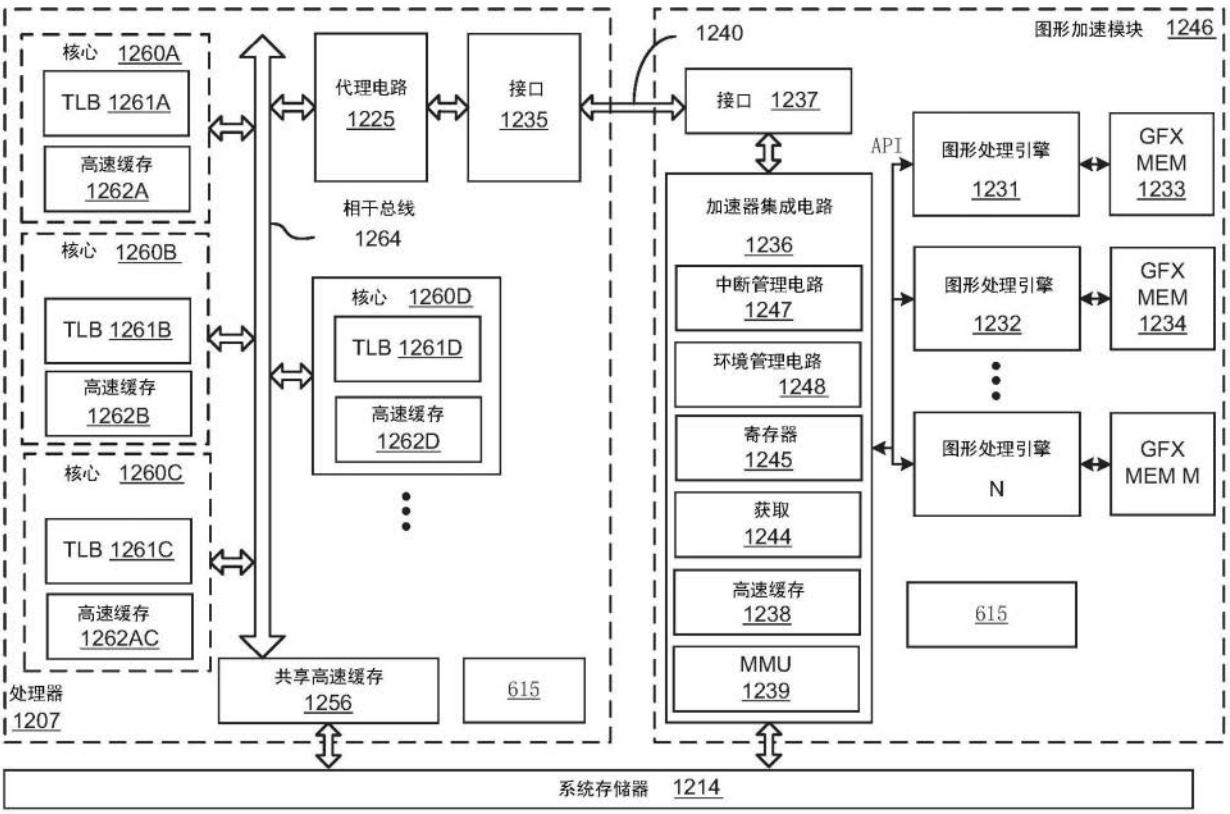

核心 1260A
TLB 1261A
高速缓存 1262A
核心 1260B
TLB 1261B
高速缓存 1262B
核心 1260C
TLB 1261C
高速缓存 1262AC
处理器 1207
代理电路 1225
接口 1235
相干总线 1264
核心 1260D
TLB 1261D
高速缓存 1262D
共享高速缓存 1256
615
1240
接口 1237
加速器集成电路 1236
中断管理电路 1247
环境管理电路 1248
寄存器 1245
获取 1244
高速缓存 1238
MMU 1239
图形加速模块 1246
API
图形处理引擎 1231
图形处理引擎 1232
图形处理引擎 N
GFX MEM 1233
GFX MEM 1234
GFX MEM M
615
系统存储器 1214

图12B

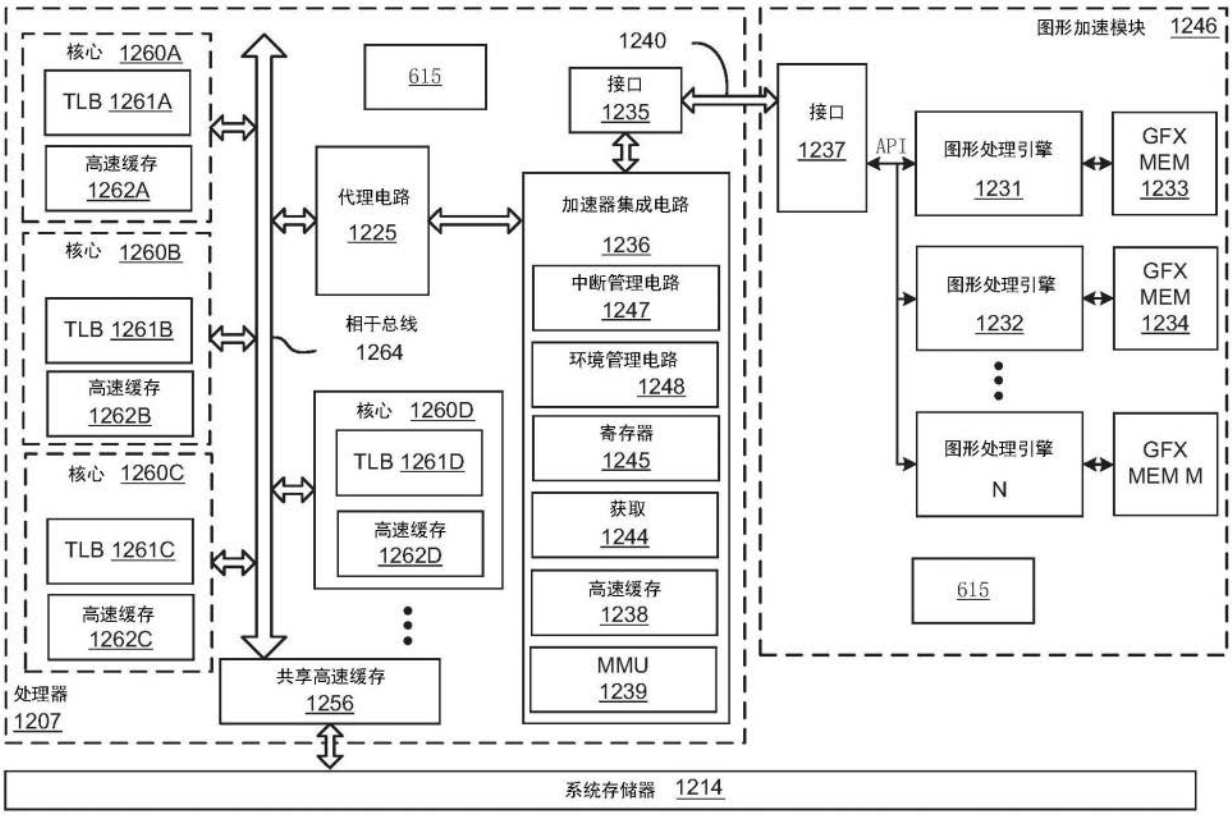

核心 1260A
TLB 1261A
高速缓存 1262A
核心 1260B
TLB 1261B
高速缓存 1262B
核心 1260C
TLB 1261C
高速缓存 1262C
处理器 1207
615
代理电路 1225
相干总线 1264
核心 1260D
TLB 1261D
高速缓存 1262D
共享高速缓存 1256
1240
接口 1235
加速器集成电路 1236
中断管理电路 1247
环境管理电路 1248
寄存器 1245
获取 1244
高速缓存 1238
MMU 1239
图形加速模块 1246
接口 1237
API
图形处理引擎 1231
图形处理引擎 1232
图形处理引擎 N
GFX MEM 1233
GFX MEM 1234
GFX MEM M
615
系统存储器 1214

图12C

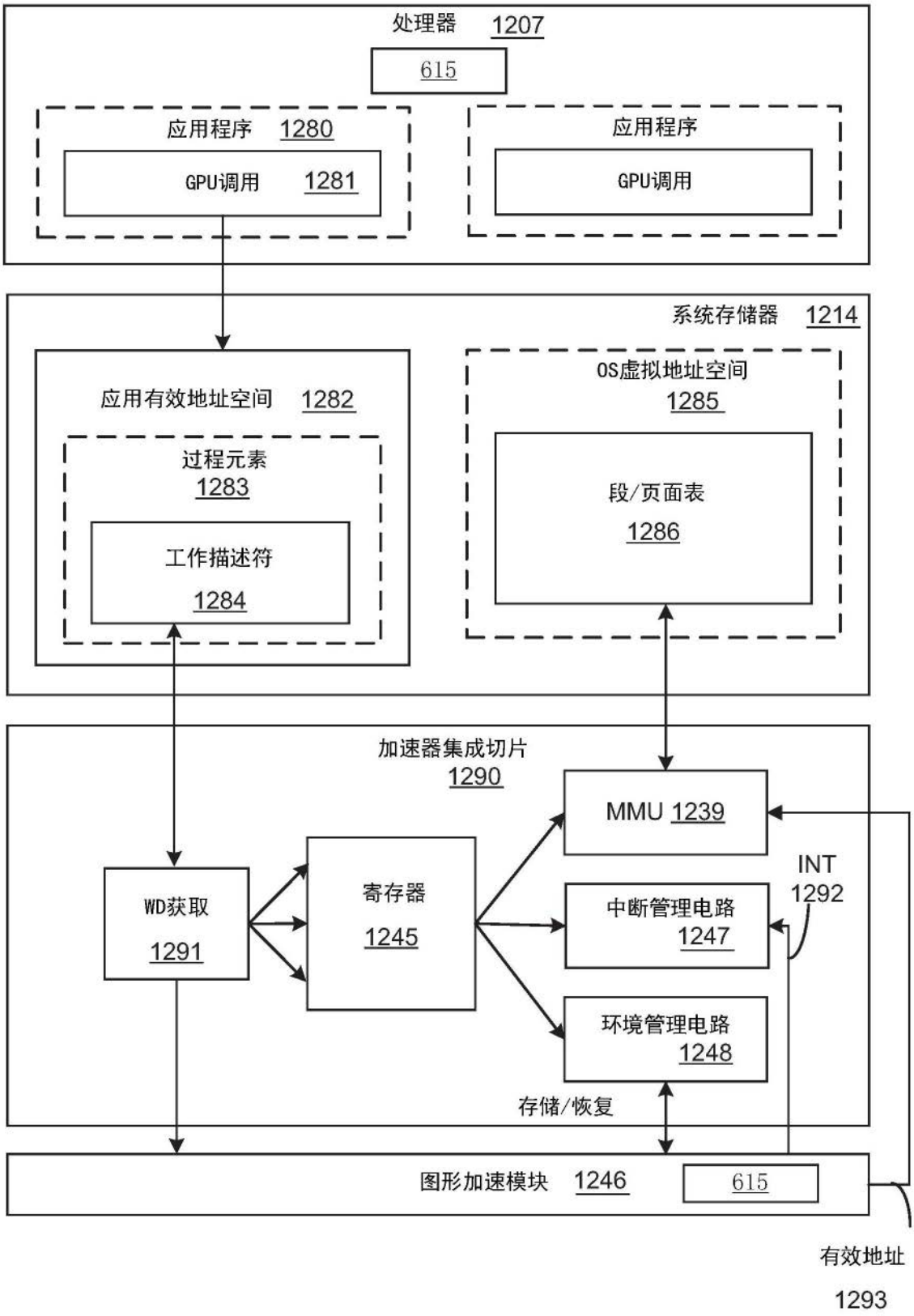

处理器 1207
615
应用程序 1280
GPU调用 1281
应用程序
GPU调用
系统存储器 1214
应用有效地址空间 1282
过程元素
1283
工作描述符
1284
OS虚拟地址空间
1285
段/页面表
1286
加速器集成切片
1290
MMU 1239
INT
1292
WD获取
1291
寄存器
1245
中断管理电路
1247
环境管理电路
1248
存储/恢复
图形加速模块 1246
615
有效地址
1293

图12D

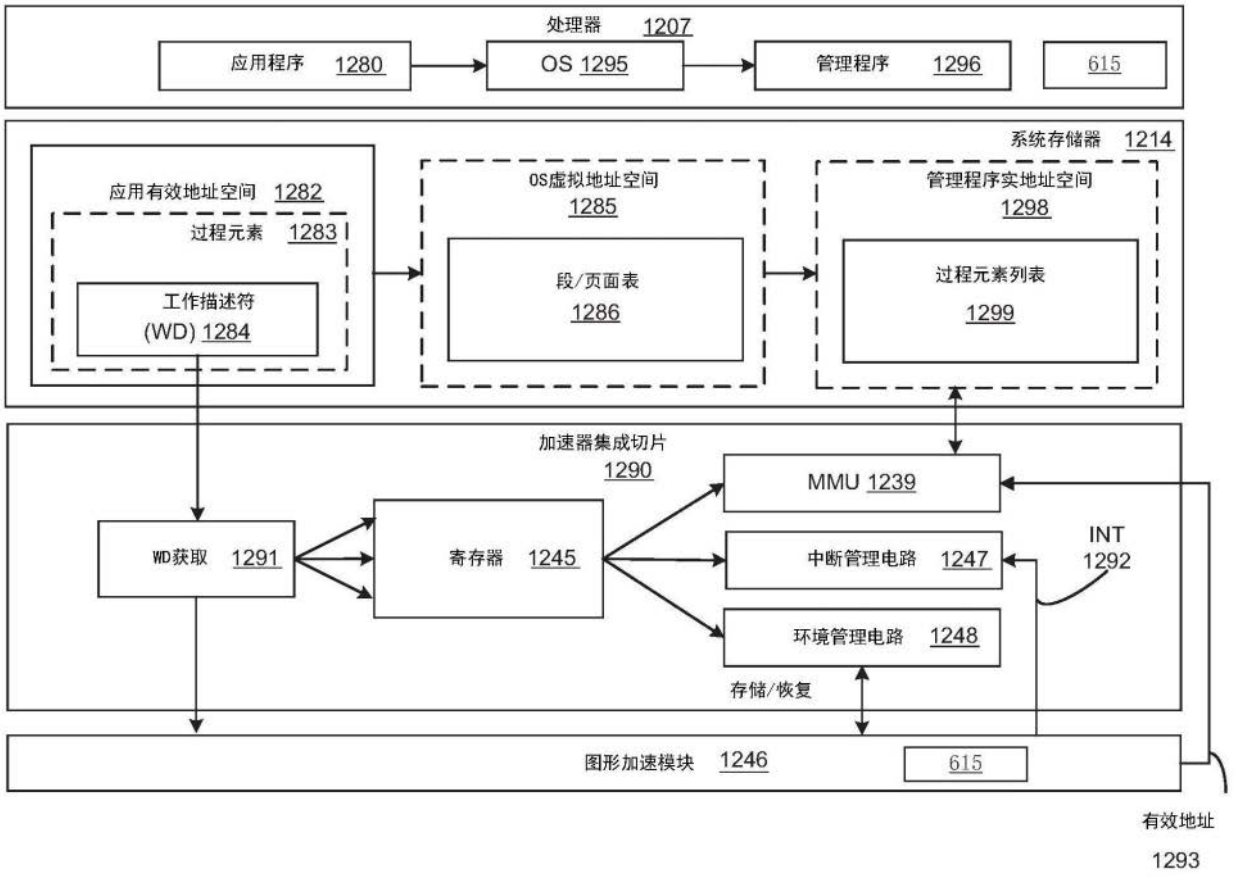

处理器 1207
应用程序 1280
OS 1295
管理程序 1296
615
系统存储器 1214
应用有效地址空间 1282
过程元素 1283
工作描述符 (WD) 1284
OS虚拟地址空间 1285
段/页面表 1286
管理程序实地址空间 1298
过程元素列表 1299
加速器集成切片 1290
MMU 1239
WD获取 1291
寄存器 1245
中断管理电路 1247
INT 1292
环境管理电路 1248
存储/恢复
图形加速模块 1246
615
有效地址 1293

图12E

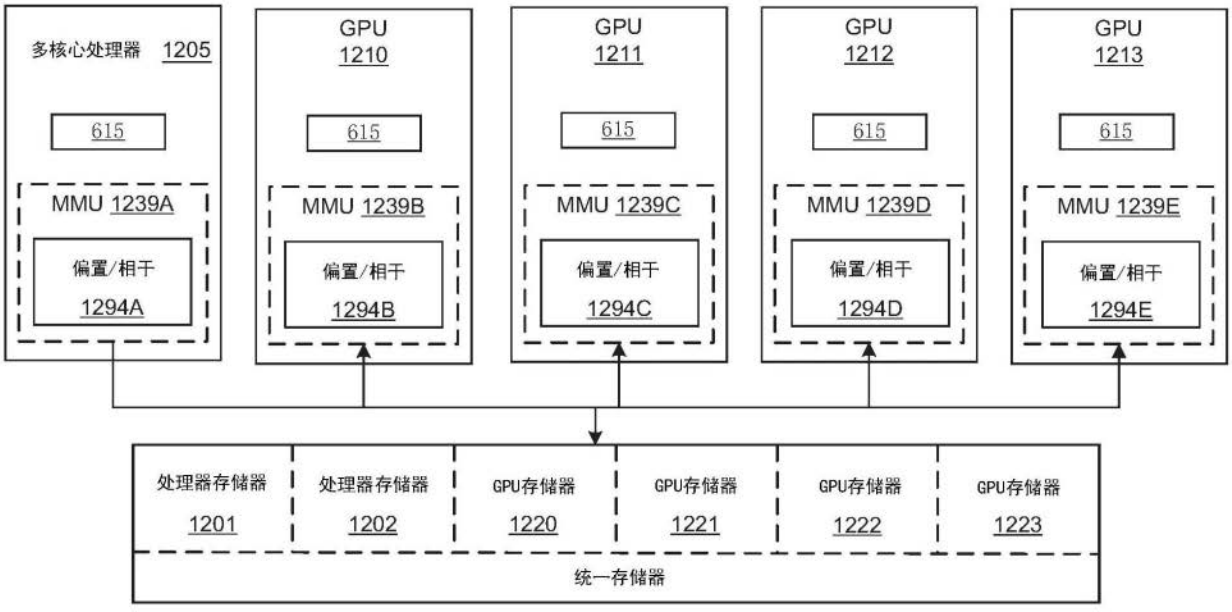

多核心处理器 1205
615
MMU 1239A
偏置/相干 1294A
GPU 1210
615
MMU 1239B
偏置/相干 1294B
GPU 1211
615
MMU 1239C
偏置/相干 1294C
GPU 1212
615
MMU 1239D
偏置/相干 1294D
GPU 1213
615
MMU 1239E
偏置/相干 1294E
处理器存储器 1201
处理器存储器 1202
GPU存储器 1220
GPU存储器 1221
GPU存储器 1222
GPU存储器 1223
统一存储器

图12F

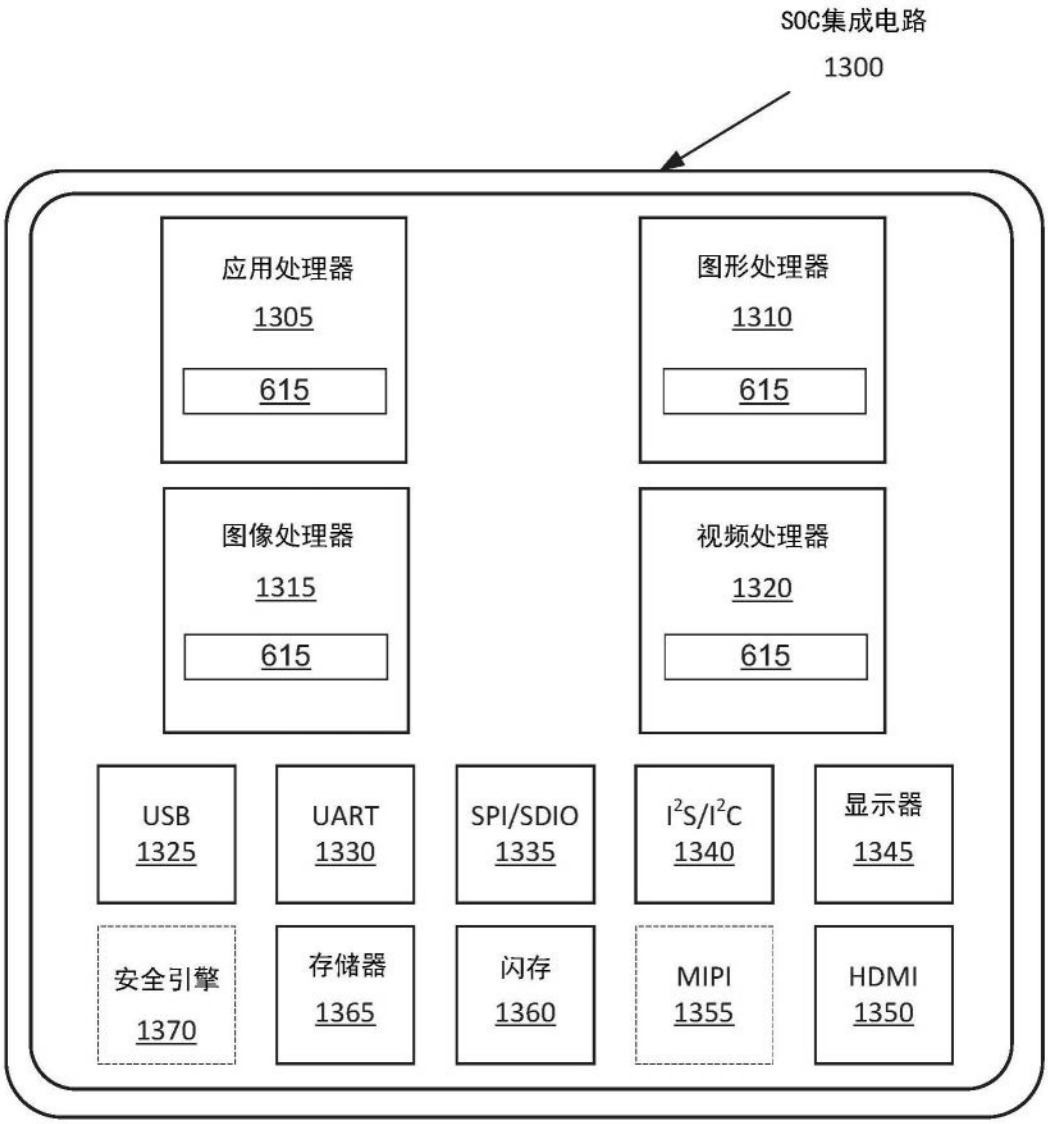

SOC集成电路
1300
应用处理器
1305
615
图形处理器
1310
615
图像处理器
1315
615
视频处理器
1320
615
USB
1325
UART
1330
SPI/SDIO
1335
I²S/I²C
1340
显示器
1345
安全引擎
1370
存储器
1365
闪存
1360
MIPI
1355
HDMI
1350

图13

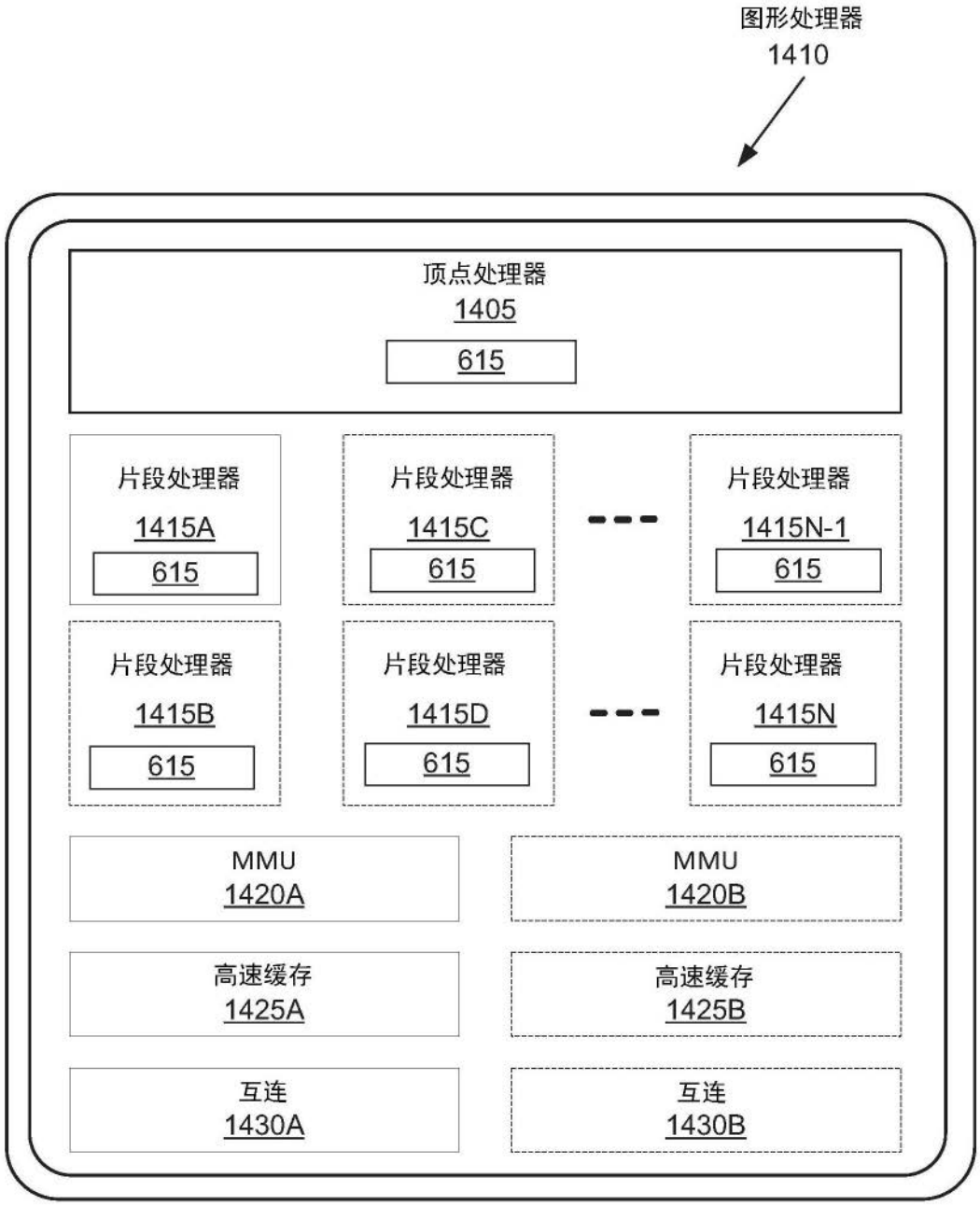

图形处理器
1410
顶点处理器
1405
615
片段处理器
1415A
615
片段处理器
1415C
615
片段处理器
1415N-1
615
片段处理器
1415B
615
片段处理器
1415D
615
片段处理器
1415N
615
MMU
1420A
MMU
1420B
高速缓存
1425A
高速缓存
1425B
互连
1430A
互连
1430B

图14A

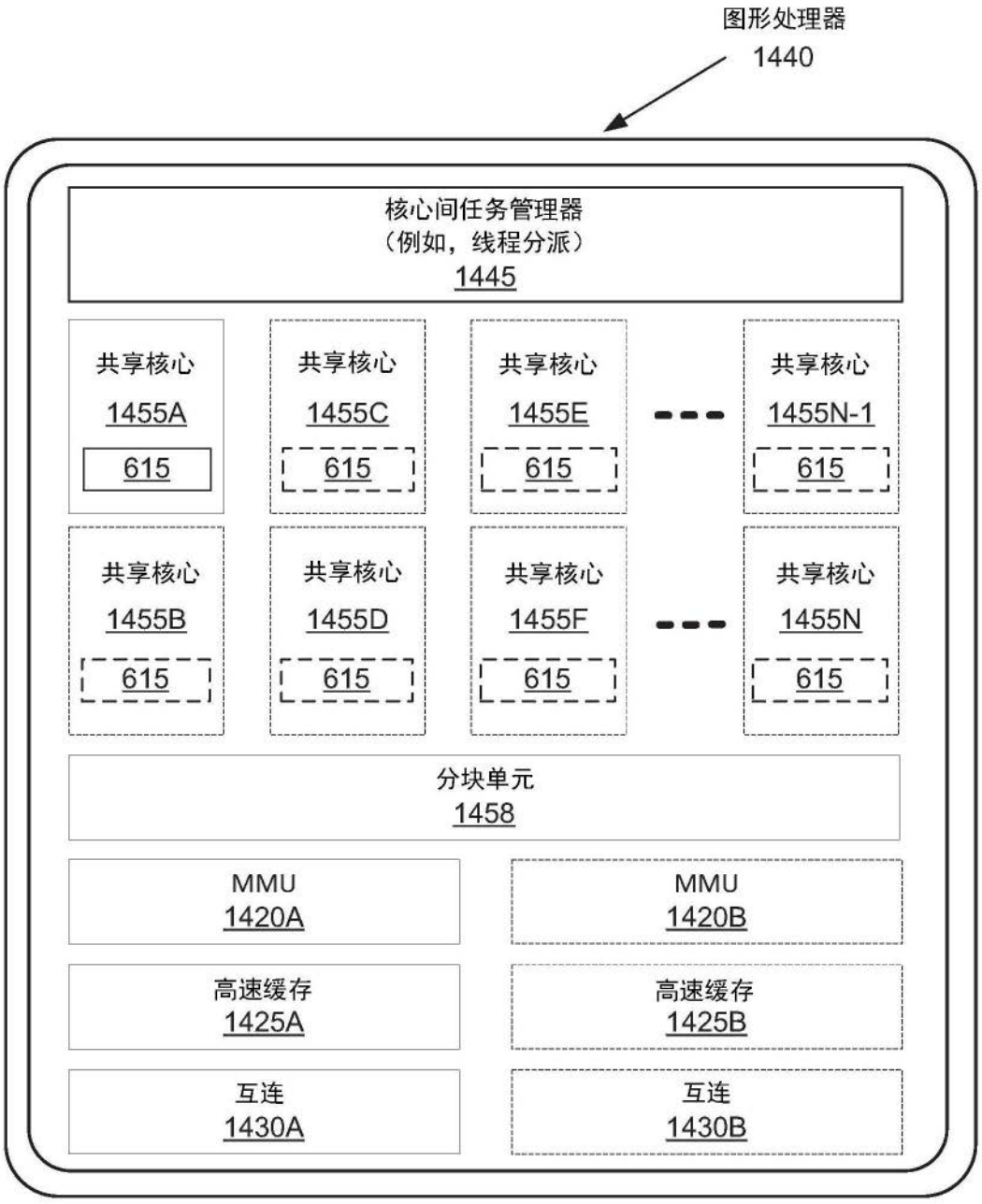

图形处理器
1440
核心间任务管理器
（例如，线程分派）
1445
共享核心
1455A
615
共享核心
1455C
615
共享核心
1455E
615
共享核心
1455N-1
615
共享核心
1455B
615
共享核心
1455D
615
共享核心
1455F
615
共享核心
1455N
615
分块单元
1458
MMU
1420A
MMU
1420B
高速缓存
1425A
高速缓存
1425B
互连
1430A
互连
1430B

图14B

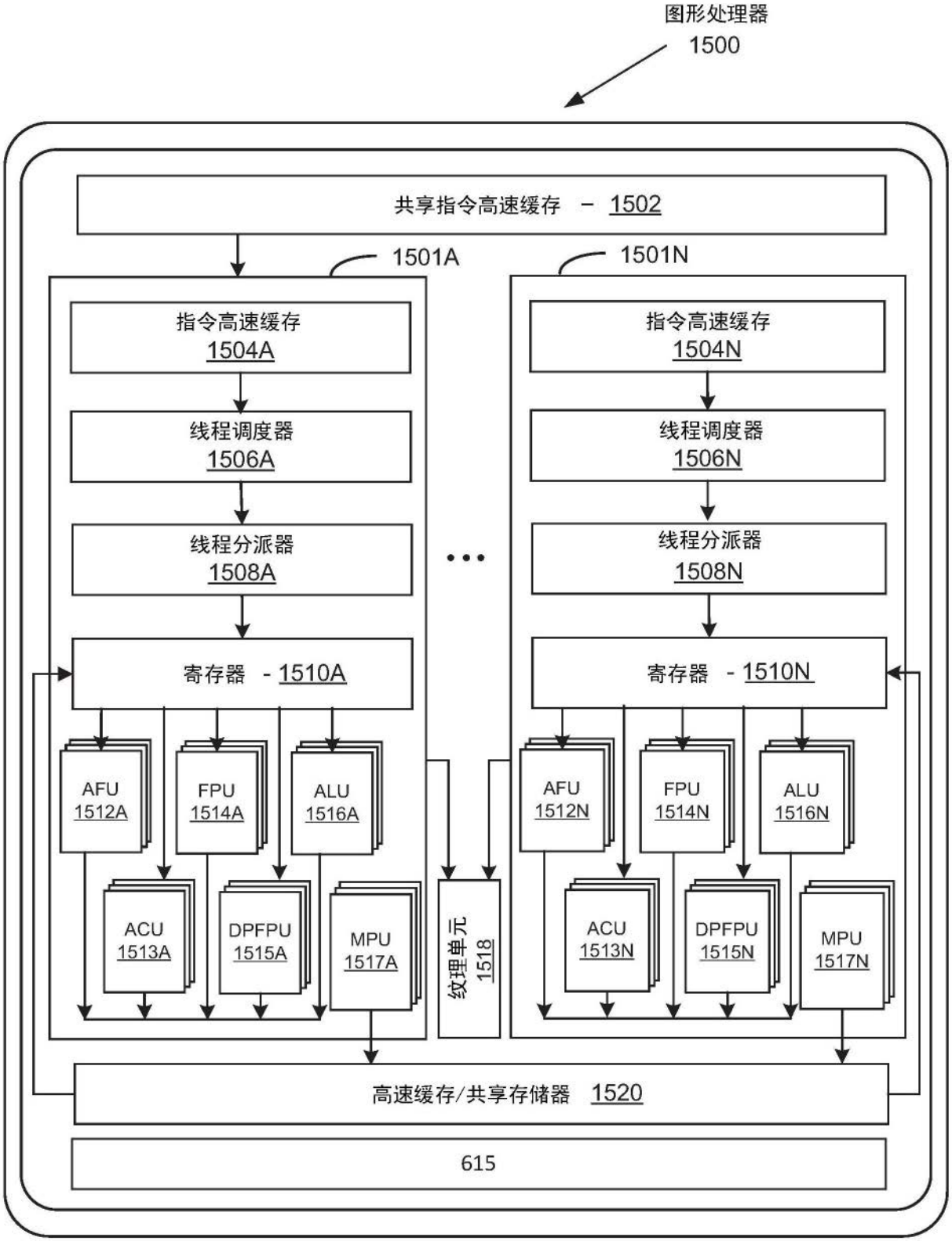

图形处理器
1500
共享指令高速缓存 - 1502
1501A
1501N
指令高速缓存
1504A
线程调度器
1506A
线程分派器
1508A
寄存器 - 1510A
AFU
1512A
FPU
1514A
ALU
1516A
ACU
1513A
DPFPU
1515A
MPU
1517A
纹理单元
1518
指令高速缓存
1504N
线程调度器
1506N
线程分派器
1508N
寄存器 - 1510N
AFU
1512N
FPU
1514N
ALU
1516N
ACU
1513N
DPFPU
1515N
MPU
1517N
高速缓存/共享存储器 1520
615

图15A

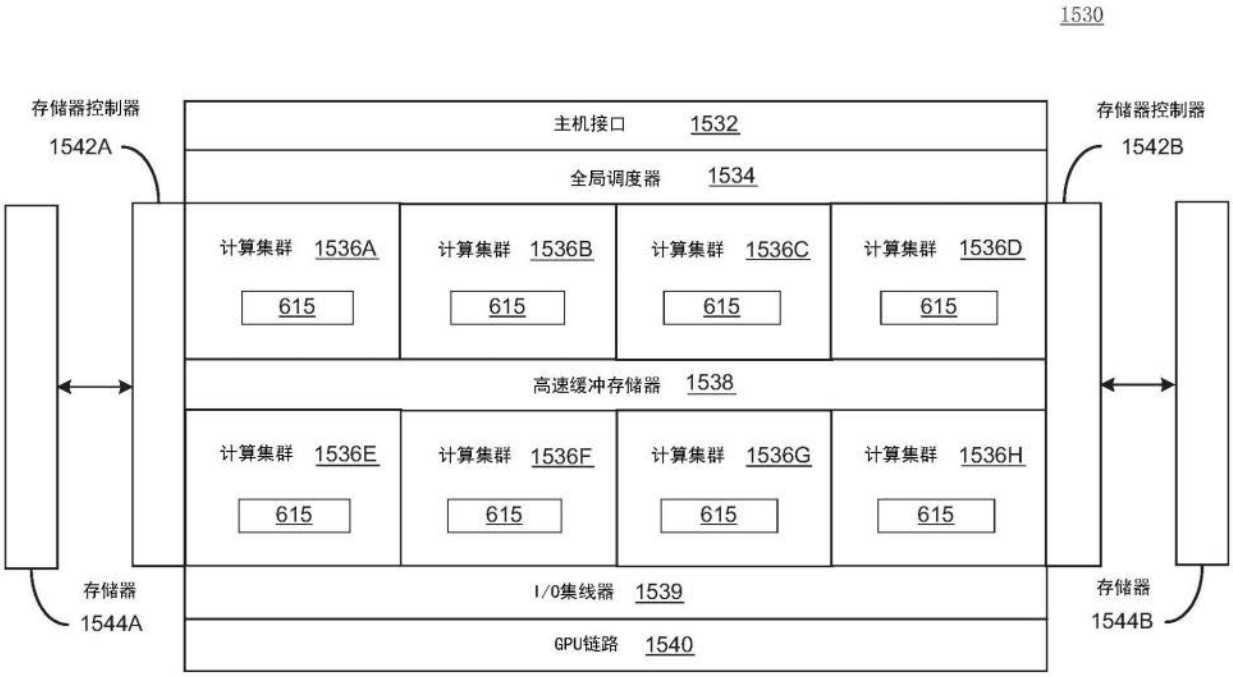

1530
存储器控制器
1542A
主机接口 1532
全局调度器 1534
存储器控制器
1542B
计算集群 1536A
615
计算集群 1536B
615
计算集群 1536C
615
计算集群 1536D
615
高速缓冲存储器 1538
计算集群 1536E
615
计算集群 1536F
615
计算集群 1536G
615
计算集群 1536H
615
存储器
1544A
I/O集线器 1539
GPU链路 1540
存储器
1544B

图15B

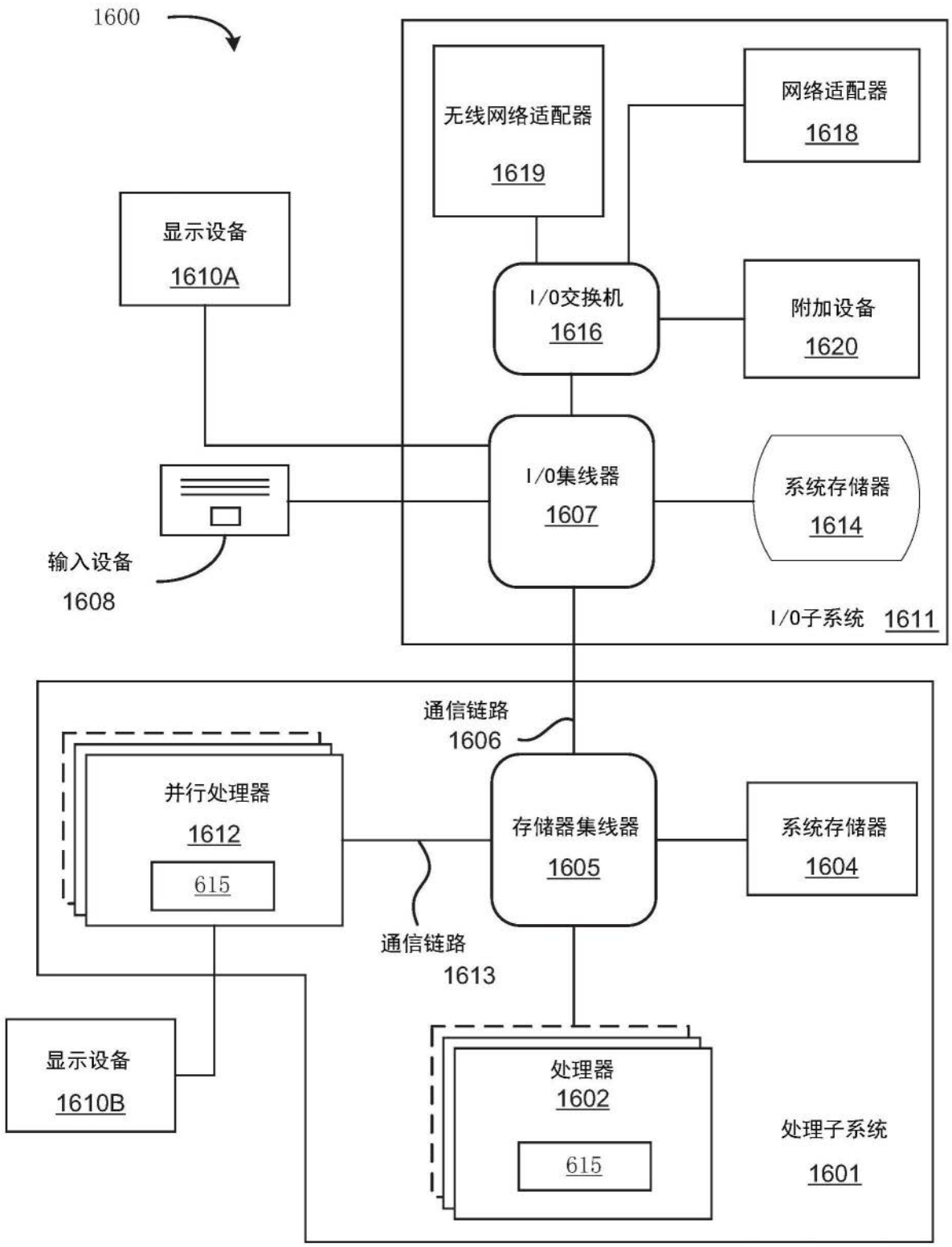

1600
无线网络适配器
1619
网络适配器
1618
显示设备
1610A
I/O交换机
1616
附加设备
1620
I/O集线器
1607
系统存储器
1614
输入设备
1608
I/O子系统 1611
通信链路
1606
并行处理器
1612
615
存储器集线器
1605
系统存储器
1604
通信链路
1613
显示设备
1610B
处理器
1602
615
处理子系统
1601

图16

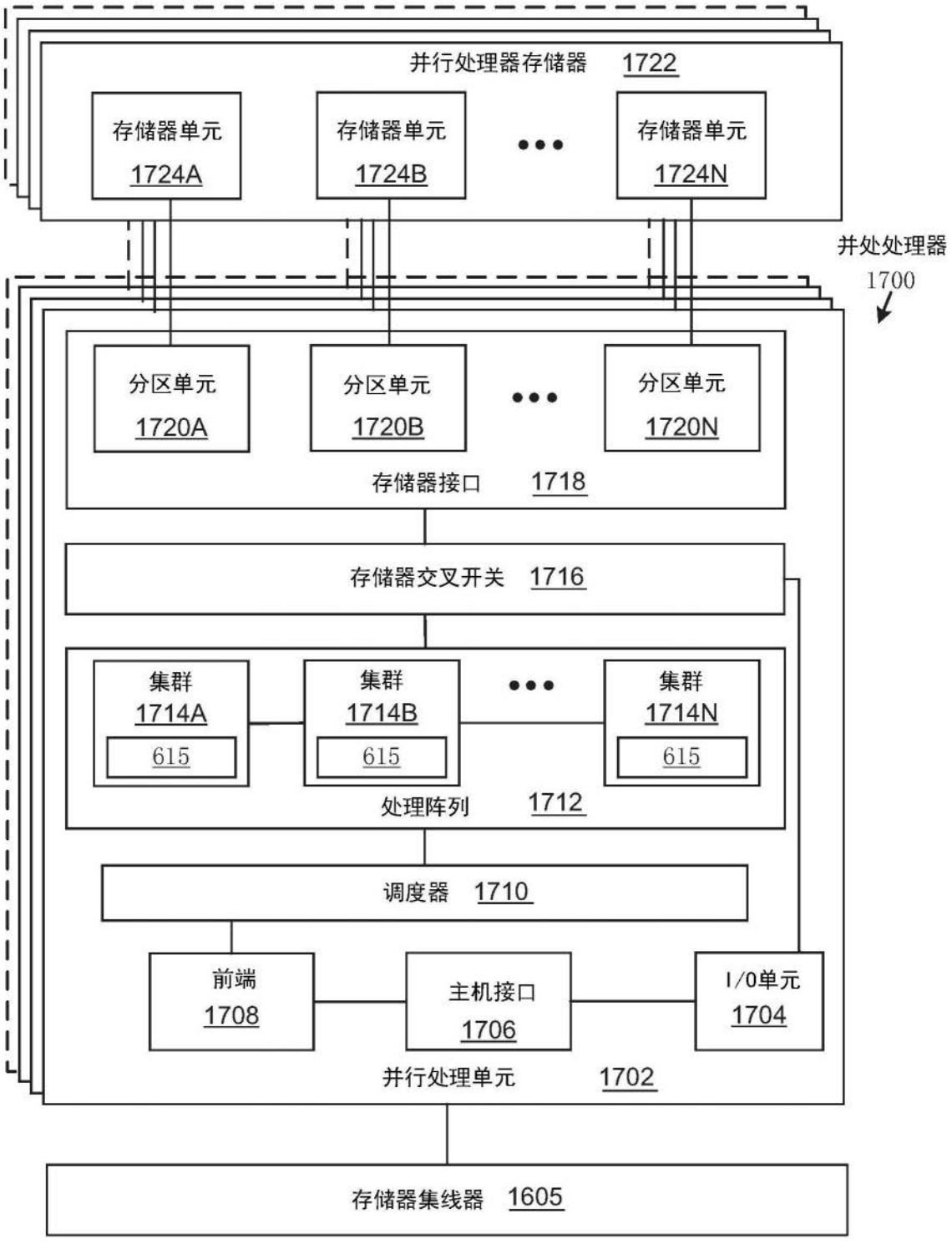

并行处理器存储器 1722
存储器单元 1724A
存储器单元 1724B
存储器单元 1724N
并处处理器 1700
分区单元 1720A
分区单元 1720B
分区单元 1720N
存储器接口 1718
存储器交叉开关 1716
集群 1714A
615
集群 1714B
615
集群 1714N
615
处理阵列 1712
调度器 1710
前端 1708
主机接口 1706
I/O单元 1704
并行处理单元 1702
存储器集线器 1605

图17A

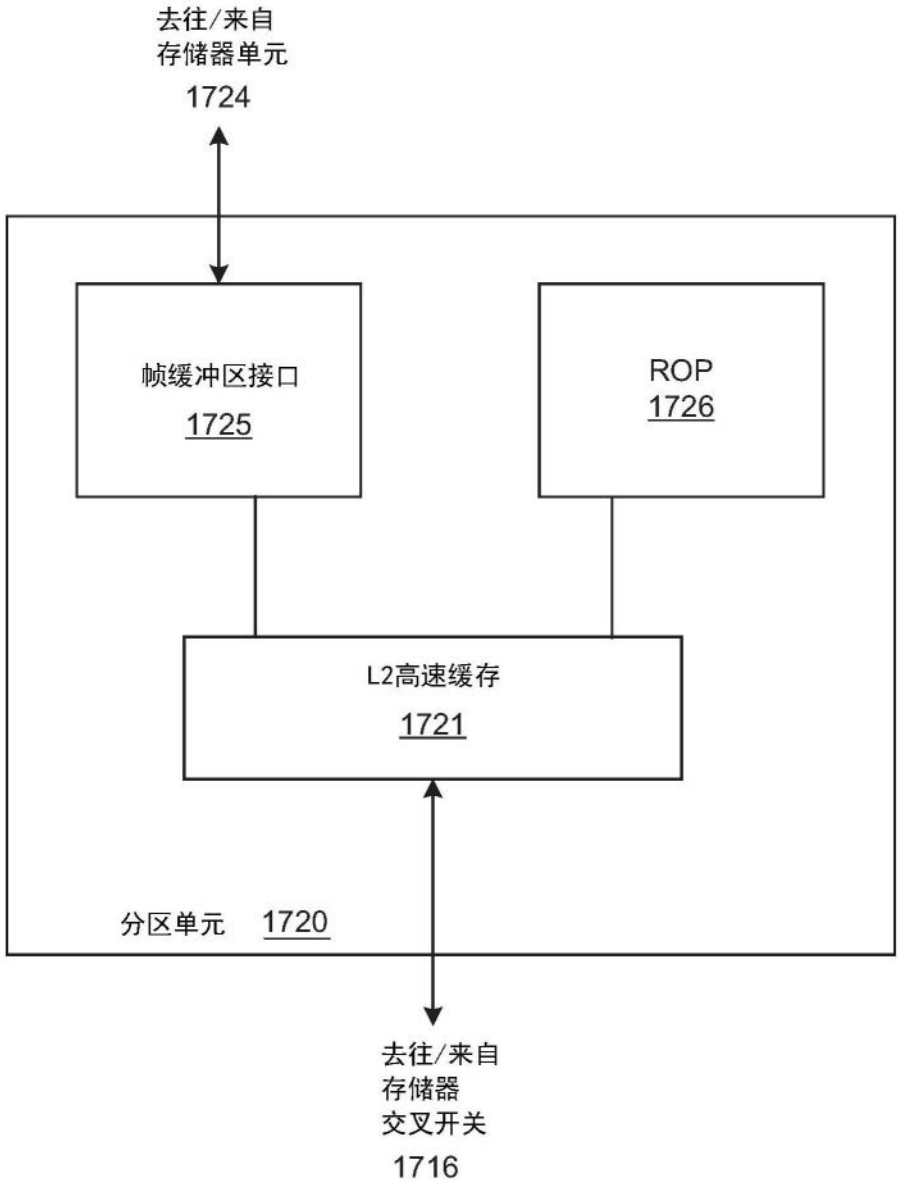

去往/来自
存储器单元
1724
帧缓冲区接口
1725
ROP
1726
L2高速缓存
1721
分区单元 1720
去往/来自
存储器
交叉开关
1716

图17B

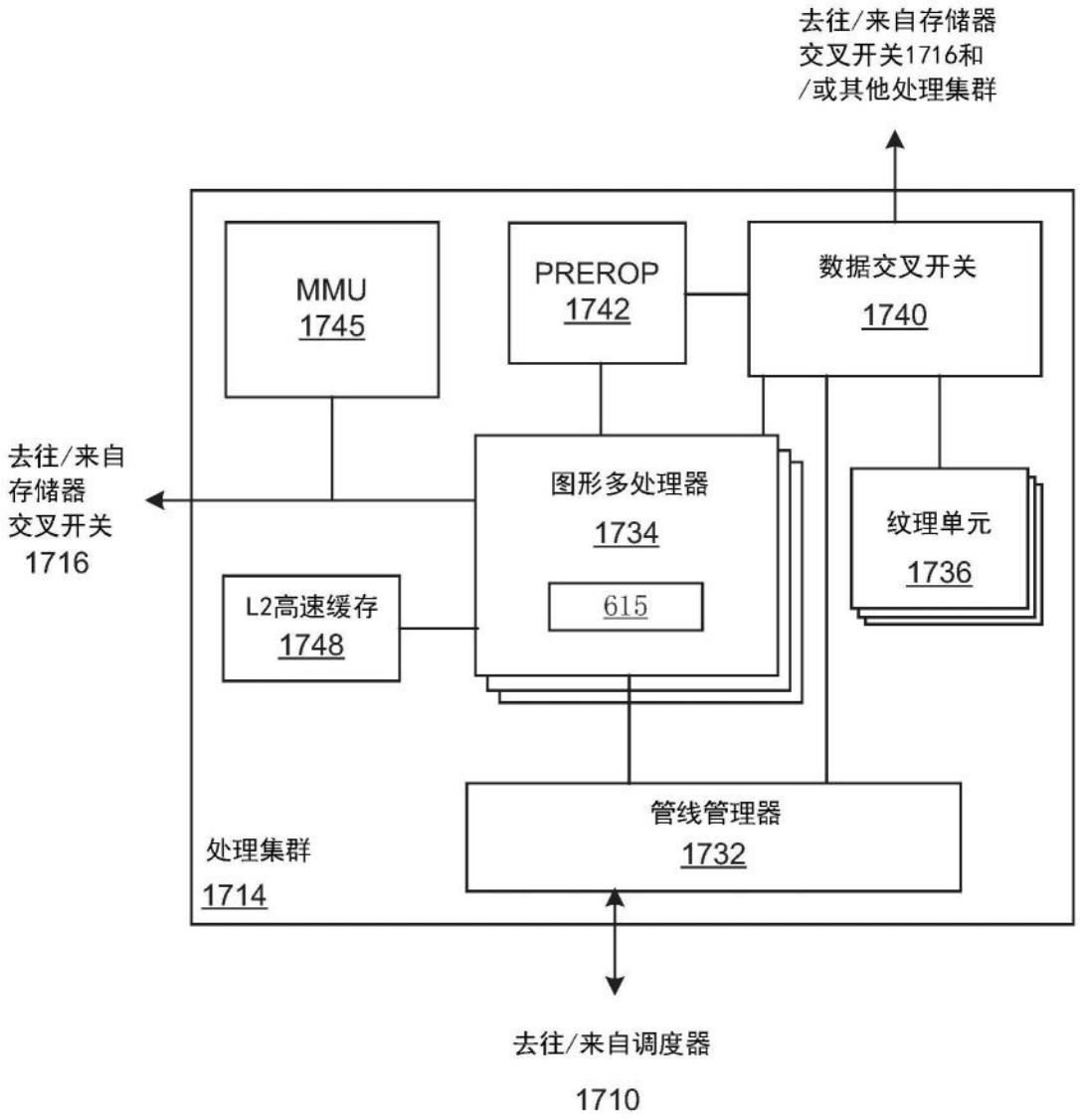

去往/来自存储器
交叉开关1716和
/或其他处理集群
MMU
1745
PREROP
1742
数据交叉开关
1740
去往/来自
存储器
交叉开关
1716
图形多处理器
1734
615
纹理单元
1736
L2高速缓存
1748
管线管理器
1732
处理集群
1714
去往/来自调度器
1710

图17C

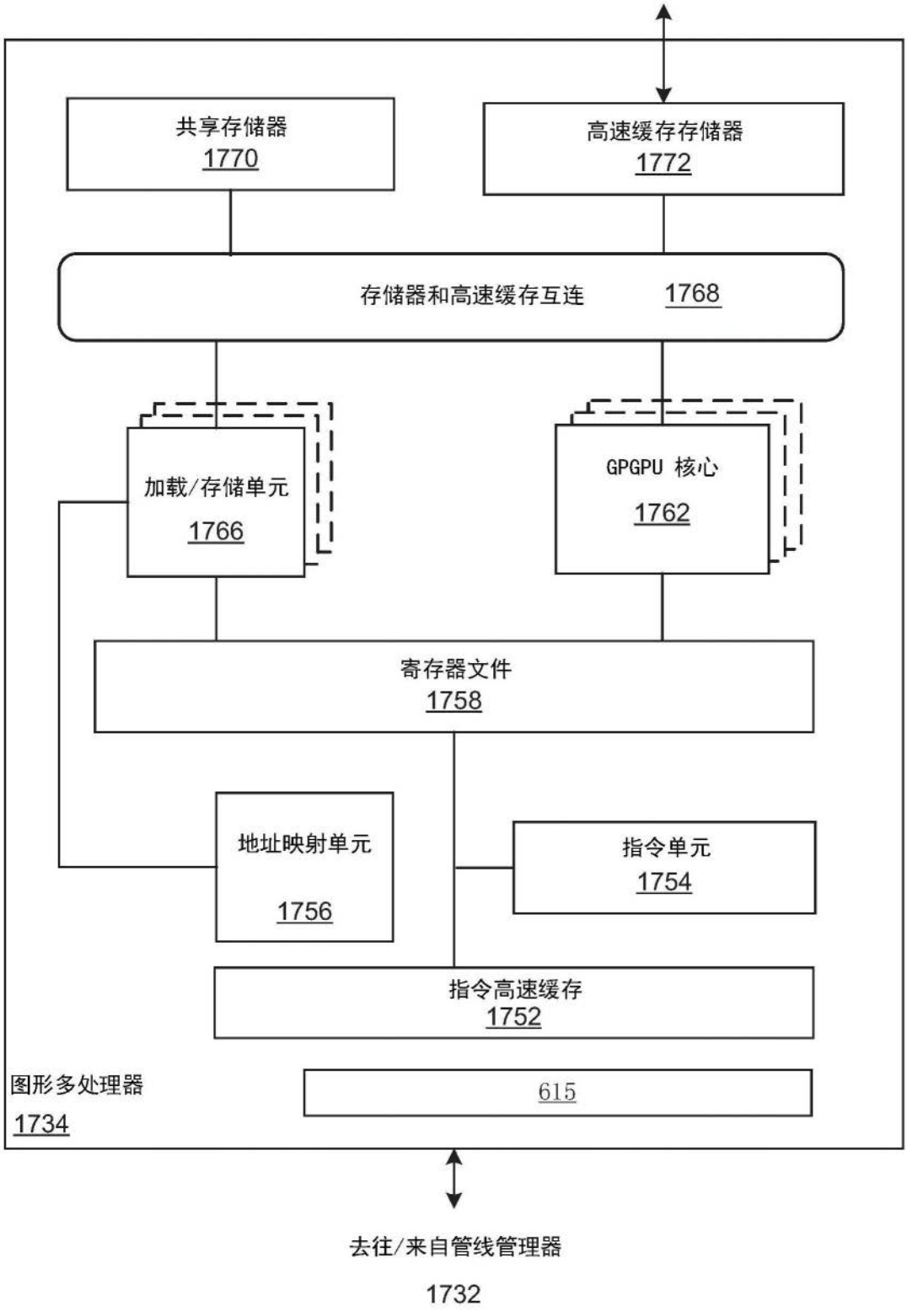

共享存储器
1770
高速缓存存储器
1772
存储器和高速缓存互连 1768
加载/存储单元
1766
GPGPU 核心
1762
寄存器文件
1758
地址映射单元
1756
指令单元
1754
指令高速缓存
1752
图形多处理器
1734
615
去往/来自管线管理器
1732

图17D

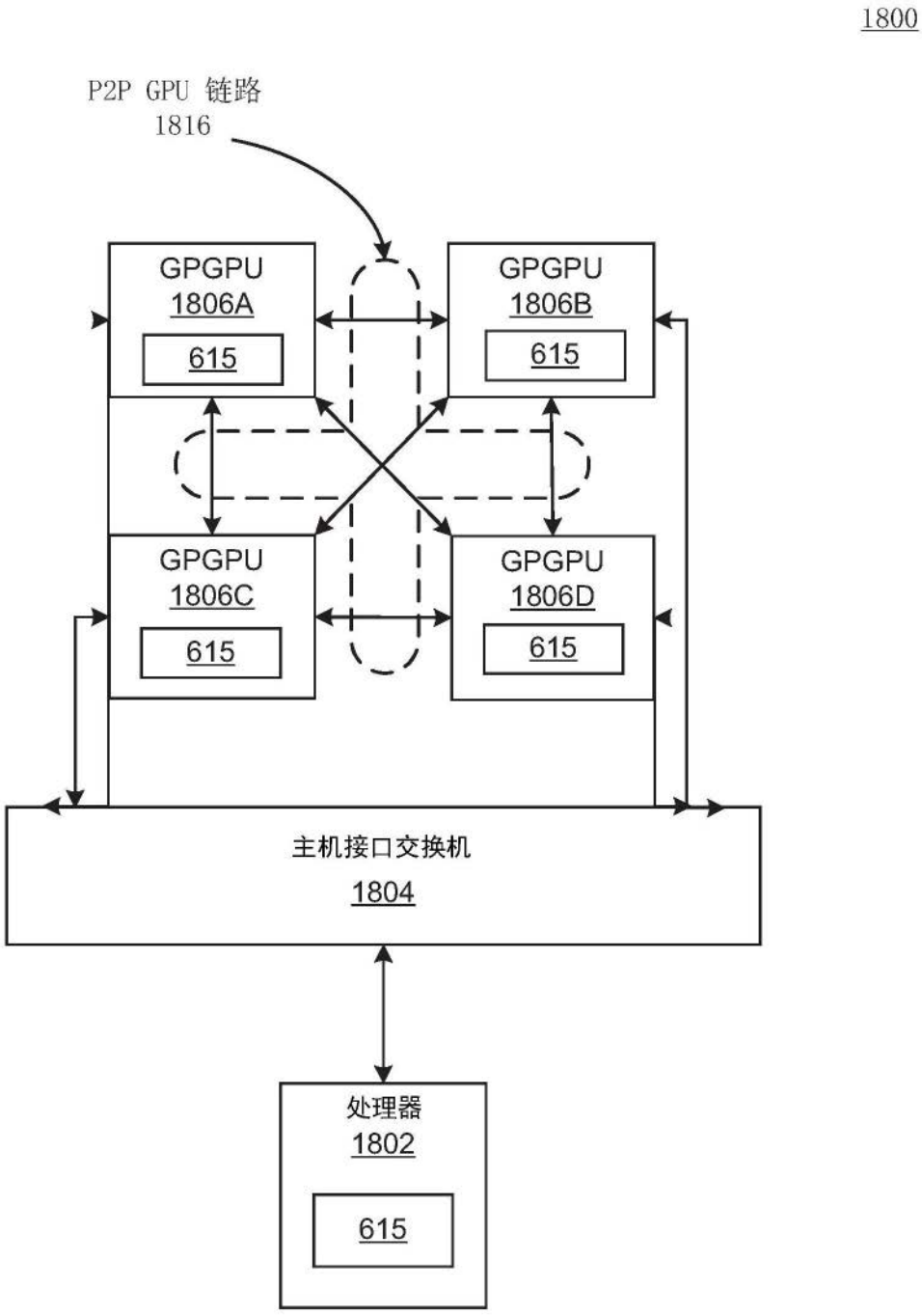

1800
P2P GPU 链路
1816
GPGPU
1806A
615
GPGPU
1806B
615
GPGPU
1806C
615
GPGPU
1806D
615
主机接口交换机
1804
处理器
1802
615

图18

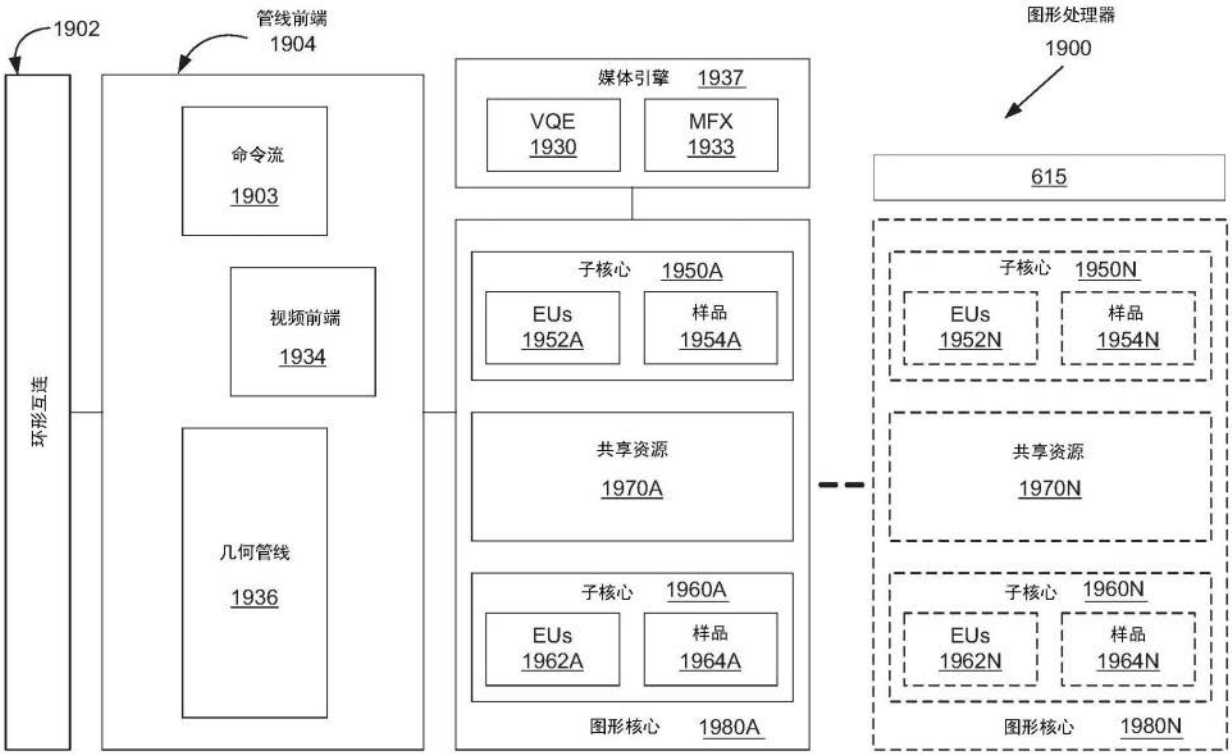

1902
管线前端
1904
命令流
1903
视频前端
1934
几何管线
1936
环形互连
媒体引擎 1937
VQE
1930
MFX
1933
子核心 1950A
EUs
1952A
样品
1954A
共享资源
1970A
子核心 1960A
EUs
1962A
样品
1964A
图形核心 1980A
图形处理器
1900
615
子核心 1950N
EUs
1952N
样品
1954N
共享资源
1970N
子核心 1960N
EUs
1962N
样品
1964N
图形核心 1980N

图19

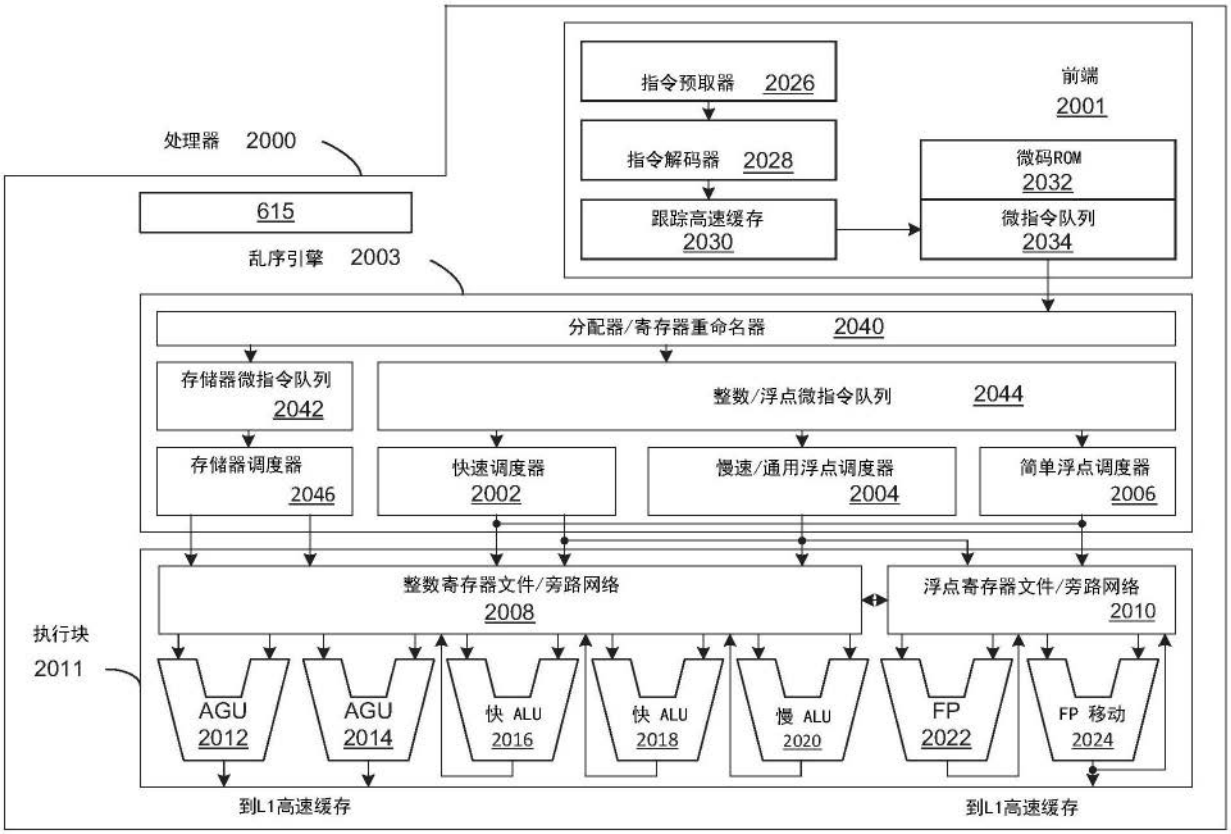

处理器 2000
615
乱序引擎 2003
指令预取器 2026
指令解码器 2028
跟踪高速缓存 2030
前端 2001
微码ROM 2032
微指令队列 2034
分配器/寄存器重命名器 2040
存储器微指令队列 2042
整数/浮点微指令队列 2044
存储器调度器 2046
快速调度器 2002
慢速/通用浮点调度器 2004
简单浮点调度器 2006
整数寄存器文件/旁路网络 2008
浮点寄存器文件/旁路网络 2010
执行块 2011
AGU 2012
AGU 2014
快 ALU 2016
快 ALU 2018
慢 ALU 2020
FP 2022
FP 移动 2024
到L1高速缓存
到L1高速缓存

图20

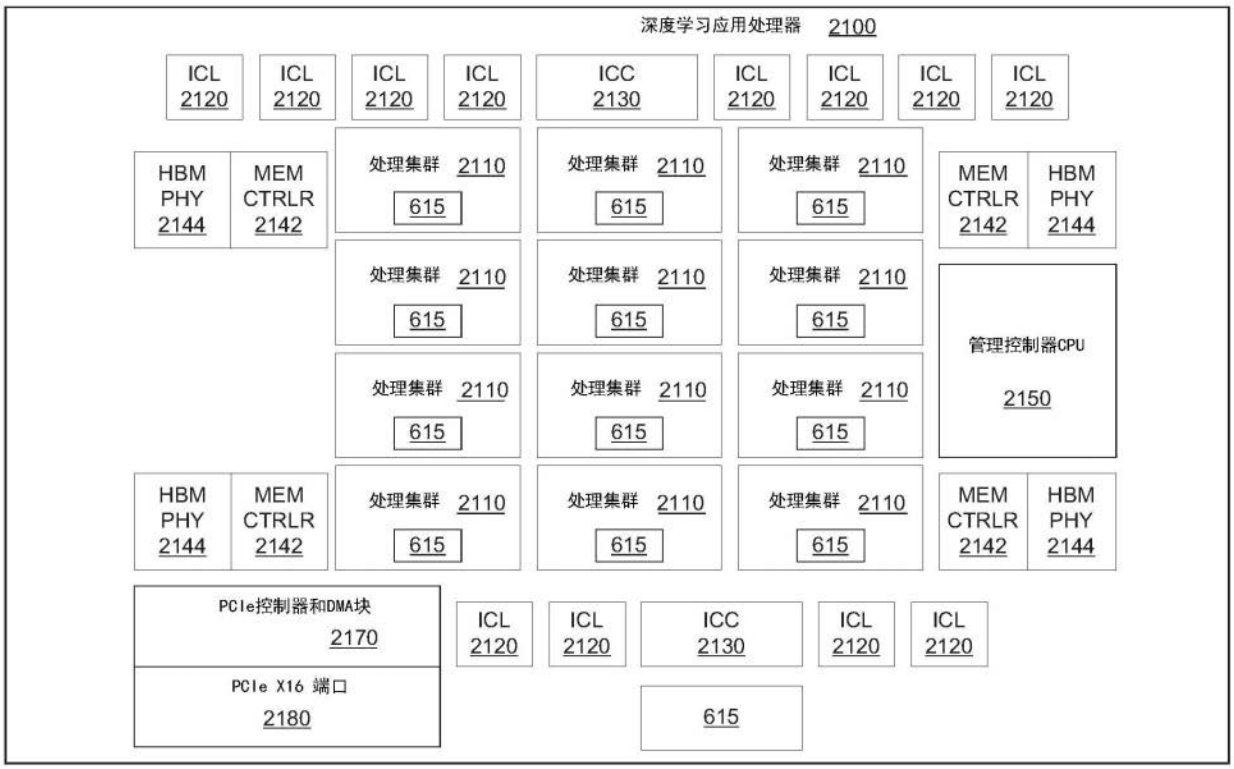

深度学习应用处理器 2100
ICL 2120
ICC 2130
HBM PHY 2144
MEM CTRLR 2142
处理集群 2110
615
管理控制器CPU 2150
PCIe控制器和DMA块 2170
PCIe X16 端口 2180

图21

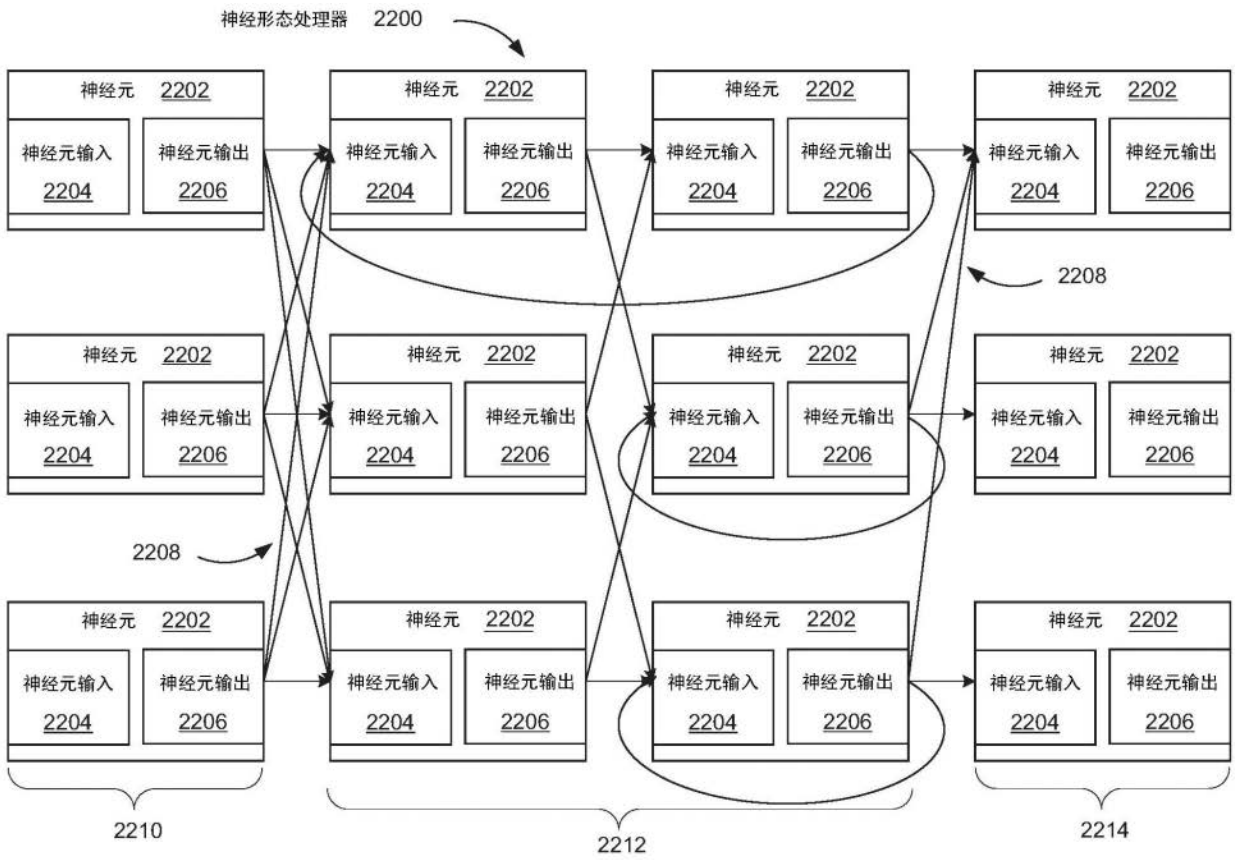

神经形态处理器 2200
神经元 2202
神经元输入 2204
神经元输出 2206
2208
2208
2210
2212
2214

图22

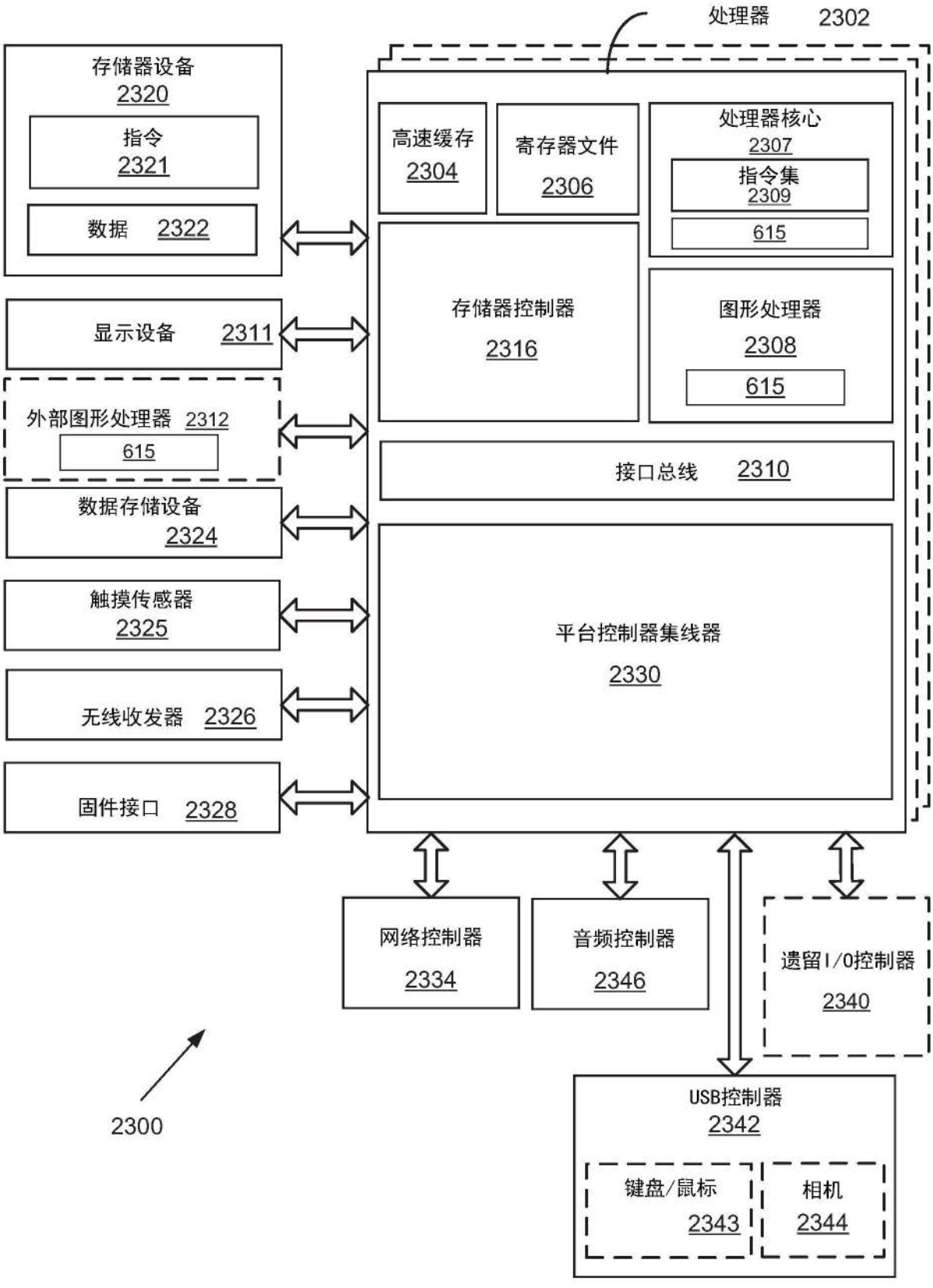

处理器 2302
存储器设备
2320
指令
2321
数据 2322
显示设备 2311
外部图形处理器 2312
615
数据存储设备
2324
触摸传感器
2325
无线收发器 2326
固件接口 2328
高速缓存
2304
寄存器文件
2306
处理器核心
2307
指令集
2309
615
存储器控制器
2316
图形处理器
2308
615
接口总线 2310
平台控制器集线器
2330
网络控制器
2334
音频控制器
2346
遗留I/O控制器
2340
USB控制器
2342
键盘/鼠标
2343
相机
2344
2300

图23

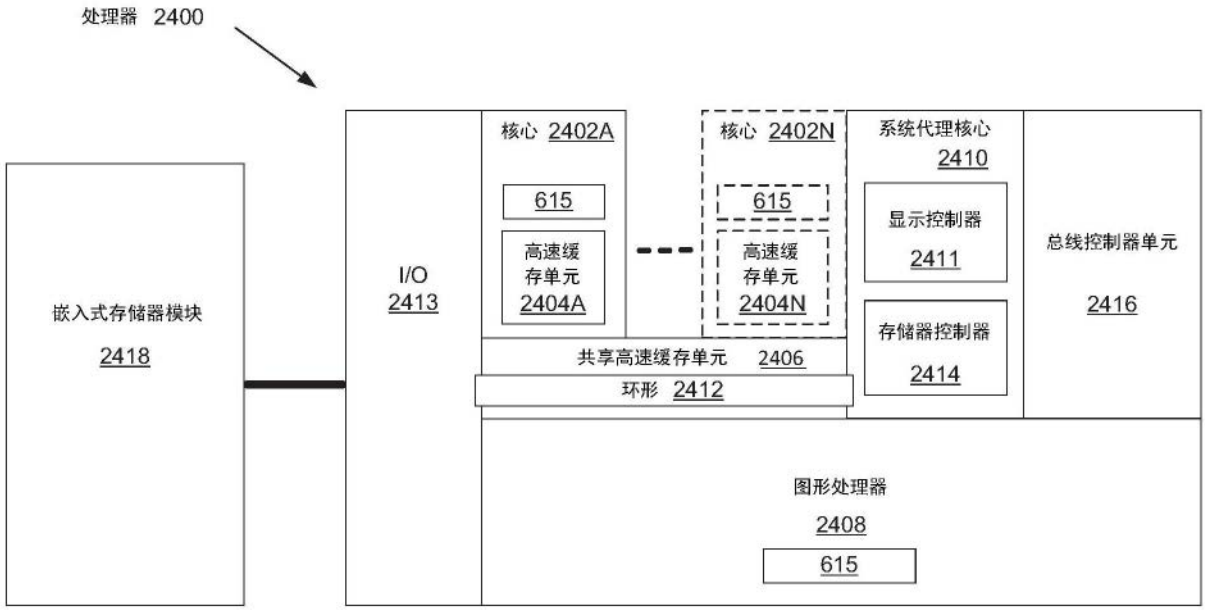

处理器 2400
嵌入式存储器模块
2418
I/O
2413
核心 2402A
615
高速缓
存单元
2404A
核心 2402N
615
高速缓
存单元
2404N
系统代理核心
2410
显示控制器
2411
存储器控制器
2414
总线控制器单元
2416
共享高速缓存单元 2406
环形 2412
图形处理器
2408
615

图24

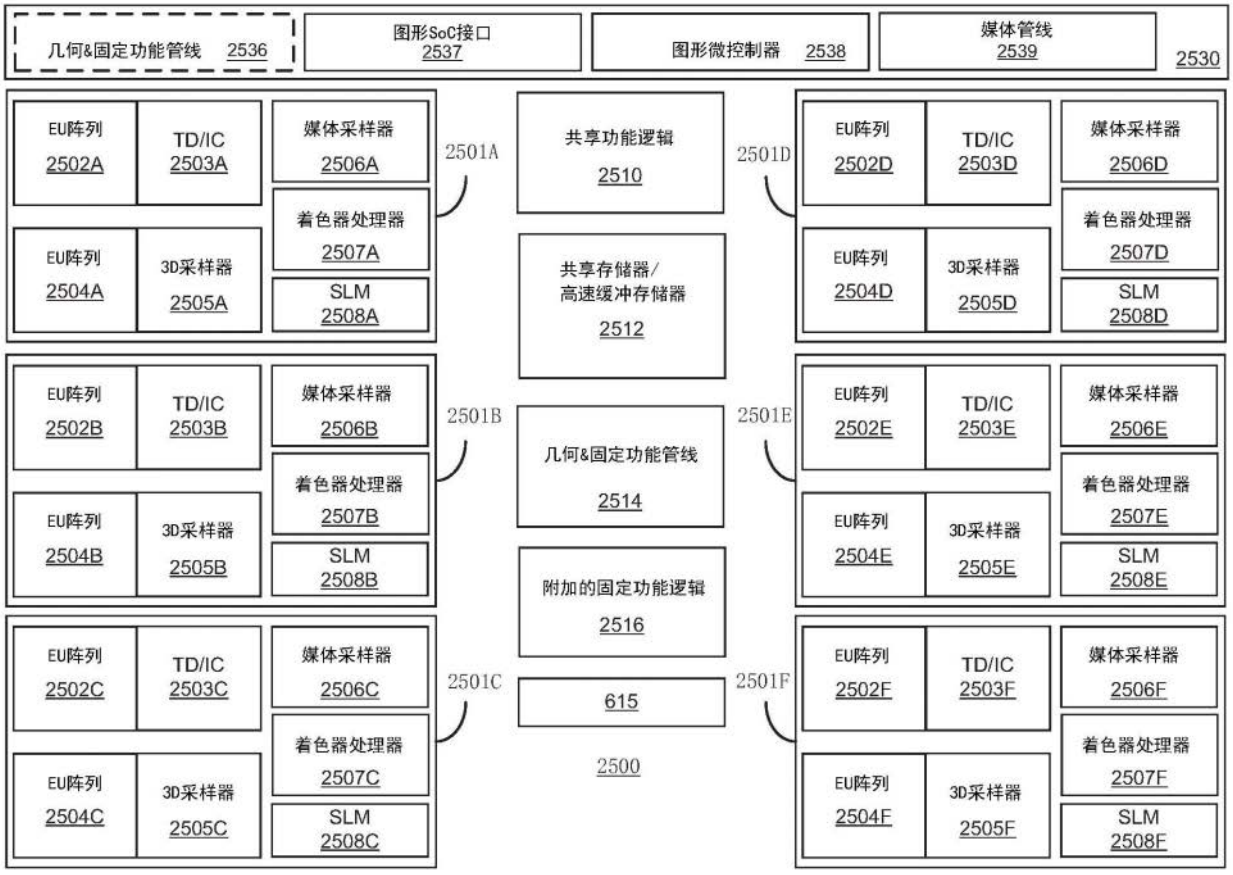

几何&固定功能管线 2536
图形SoC接口
2537
图形微控制器 2538
媒体管线
2539
2530
EU阵列
2502A
TD/IC
2503A
媒体采样器
2506A
着色器处理器
2507A
SLM
2508A
EU阵列
2504A
3D采样器
2505A
2501A
共享功能逻辑
2510
共享存储器/
高速缓冲存储器
2512
几何&固定功能管线
2514
附加的固定功能逻辑
2516
615
2500
2501D
EU阵列
2502D
TD/IC
2503D
媒体采样器
2506D
着色器处理器
2507D
SLM
2508D
EU阵列
2504D
3D采样器
2505D
EU阵列
2502B
TD/IC
2503B
媒体采样器
2506B
着色器处理器
2507B
SLM
2508B
EU阵列
2504B
3D采样器
2505B
2501B
2501E
EU阵列
2502E
TD/IC
2503E
媒体采样器
2506E
着色器处理器
2507E
SLM
2508E
EU阵列
2504E
3D采样器
2505E
EU阵列
2502C
TD/IC
2503C
媒体采样器
2506C
着色器处理器
2507C
SLM
2508C
EU阵列
2504C
3D采样器
2505C
2501C
2501F
EU阵列
2502F
TD/IC
2503F
媒体采样器
2506F
着色器处理器
2507F
SLM
2508F
EU阵列
2504F
3D采样器
2505F

图25

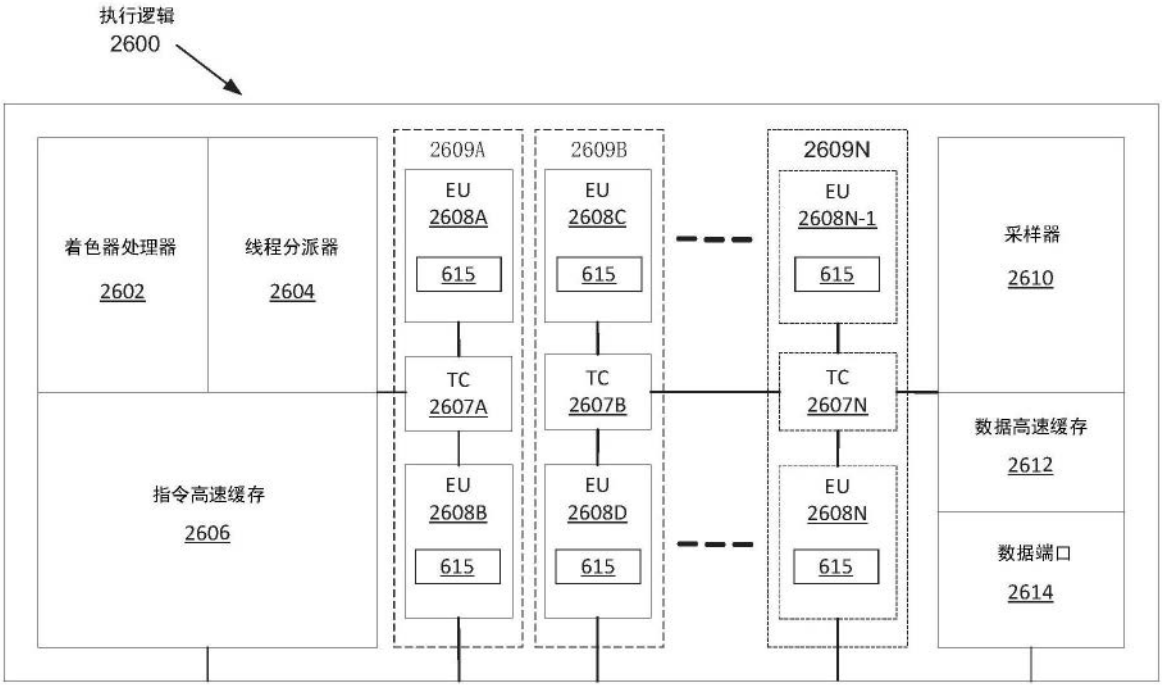

执行逻辑
2600
着色器处理器
2602
线程分派器
2604
指令高速缓存
2606
2609A
EU
2608A
615
TC
2607A
EU
2608B
615
2609B
EU
2608C
615
TC
2607B
EU
2608D
615
2609N
EU
2608N-1
615
TC
2607N
EU
2608N
615
采样器
2610
数据高速缓存
2612
数据端口
2614

图26A

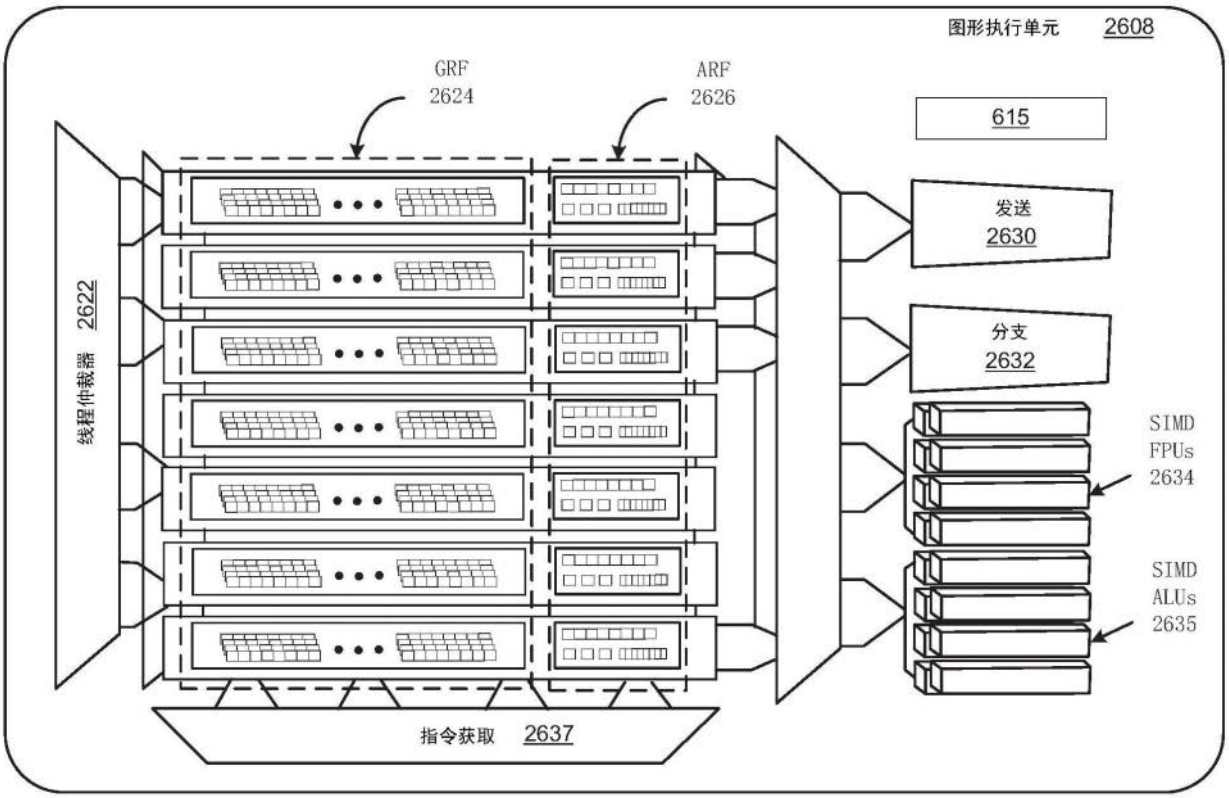

图形执行单元 2608
GRF
2624
ARF
2626
615
发送
2630
分支
2632
SIMD
FPUs
2634
SIMD
ALUs
2635
线程仲裁器 2622
指令获取 2637

图26B

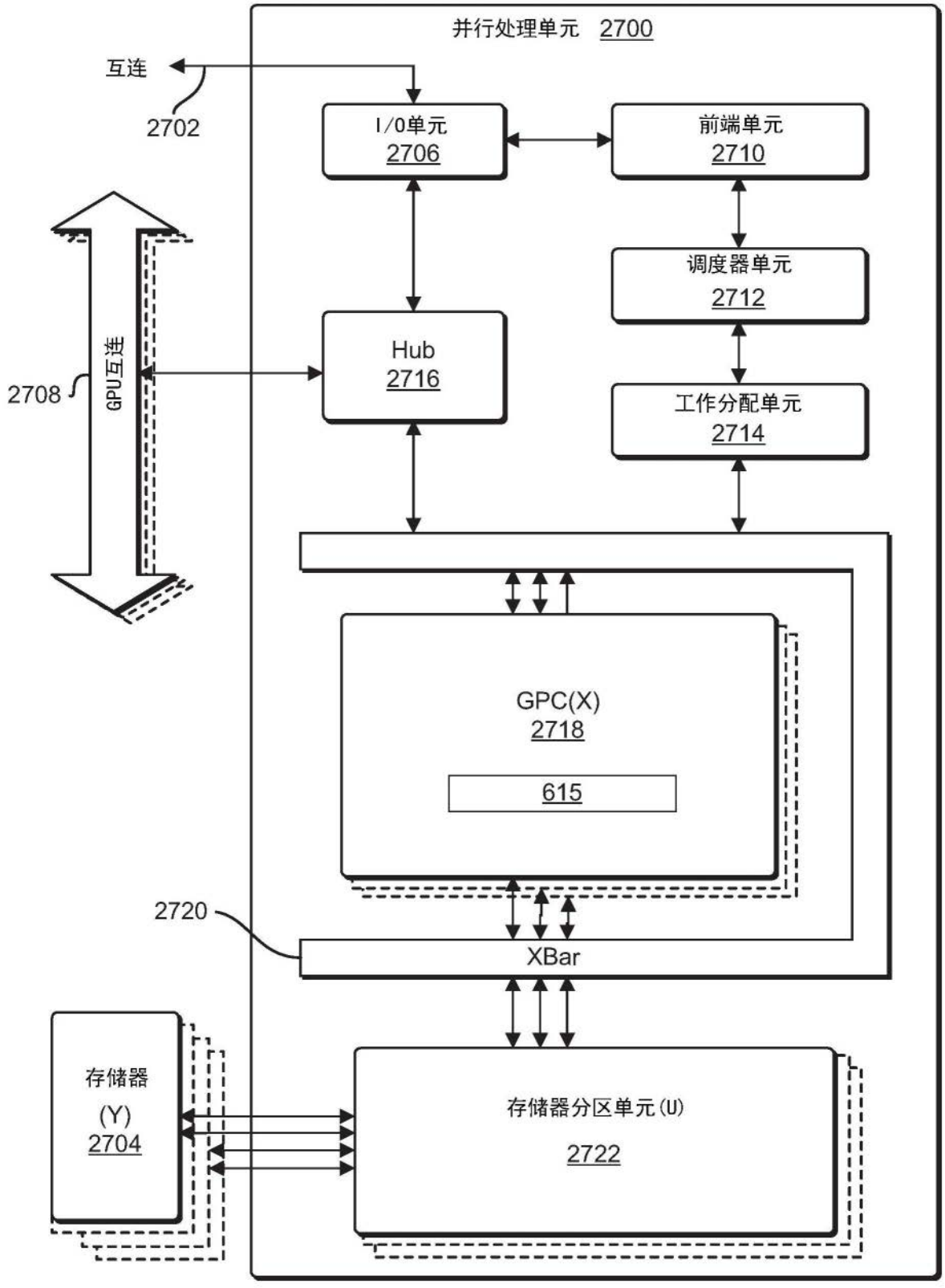

并行处理单元 2700
互连
2702
I/O单元
2706
前端单元
2710
调度器单元
2712
Hub
2716
工作分配单元
2714
2708
GPU互连
GPC(X)
2718
615
2720
XBar
存储器
(Y)
2704
存储器分区单元(U)
2722

图27

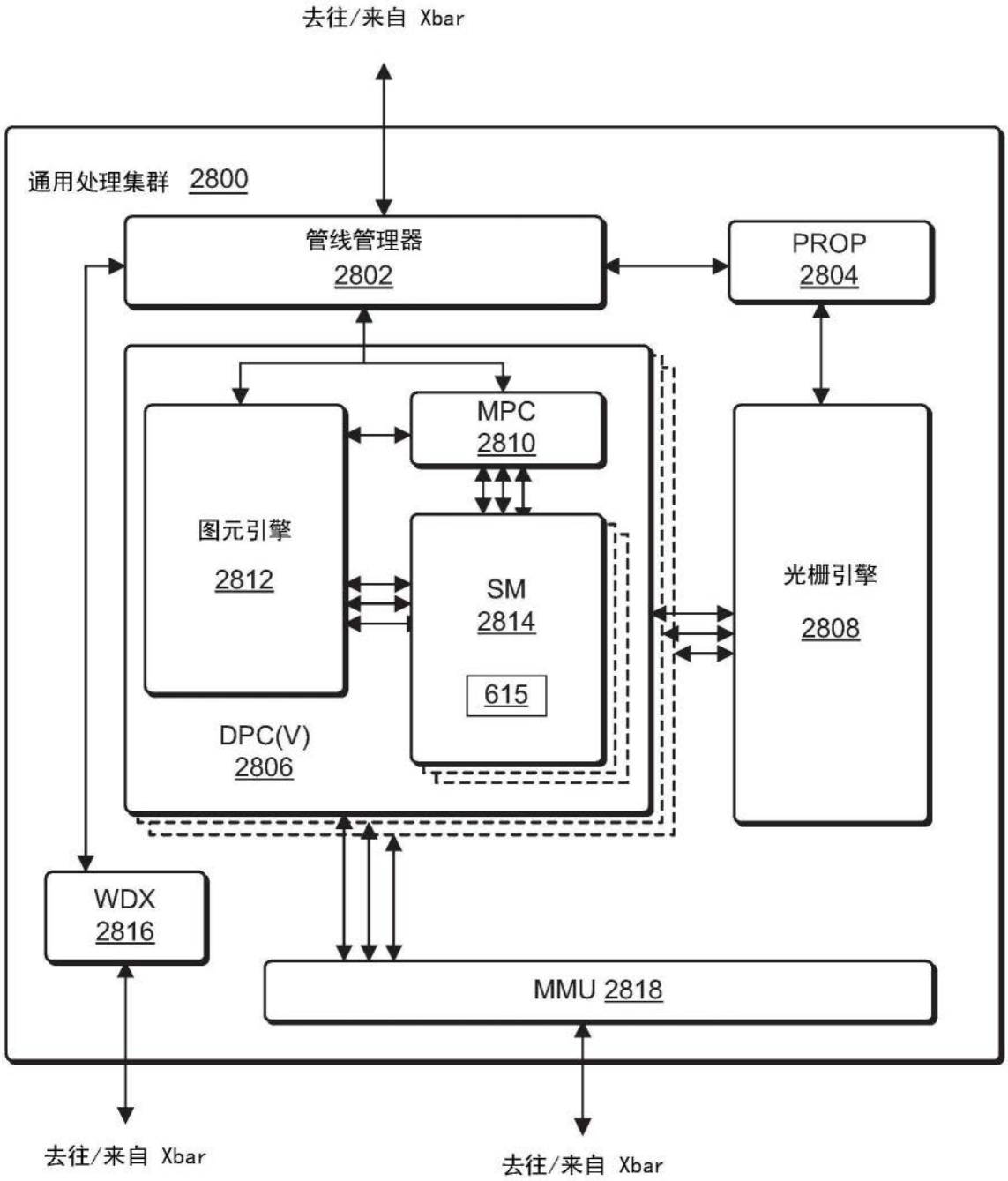

去往/来自 Xbar
通用处理集群 2800
管线管理器
2802
PROP
2804
MPC
2810
图元引擎
2812
SM
2814
615
光栅引擎
2808
DPC(V)
2806
WDX
2816
MMU 2818
去往/来自 Xbar
去往/来自 Xbar

图28

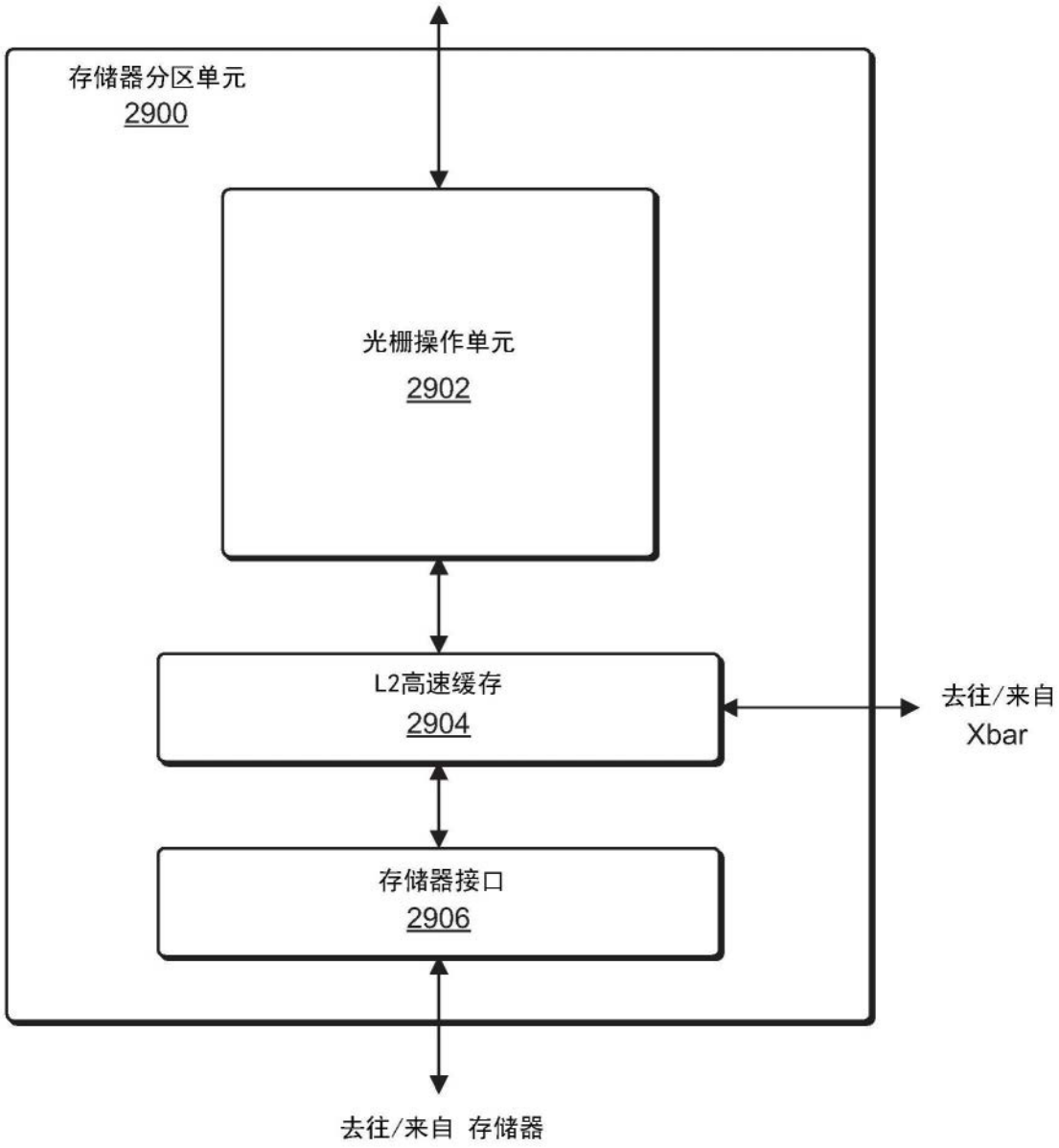

存储器分区单元
2900
光栅操作单元
2902
L2高速缓存
2904
去往/来自
Xbar
存储器接口
2906
去往/来自 存储器

图29

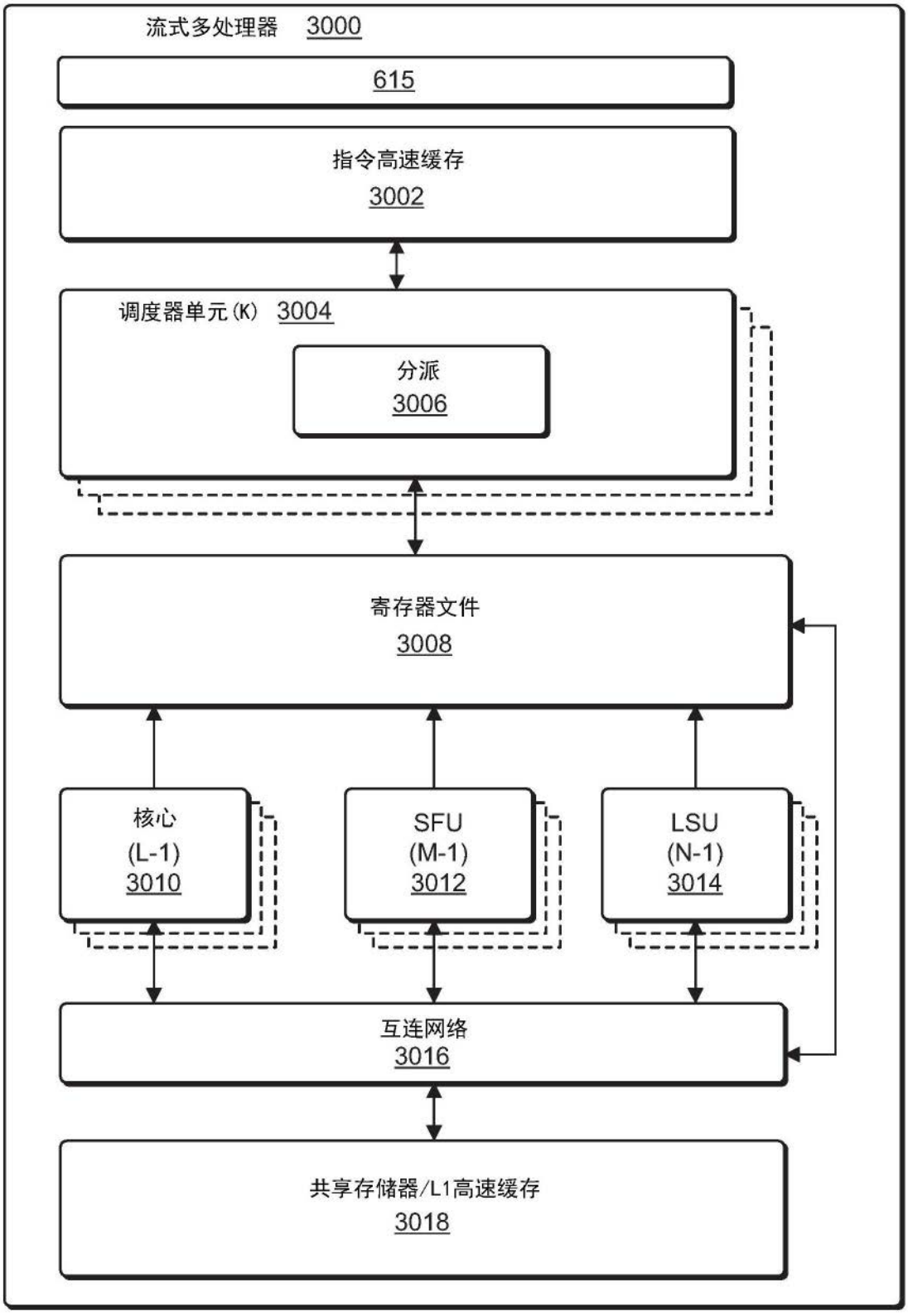

流式多处理器 3000
615
指令高速缓存
3002
调度器单元(K) 3004
分派
3006
寄存器文件
3008
核心
(L-1)
3010
SFU
(M-1)
3012
LSU
(N-1)
3014
互连网络
3016
共享存储器/L1高速缓存
3018

图30

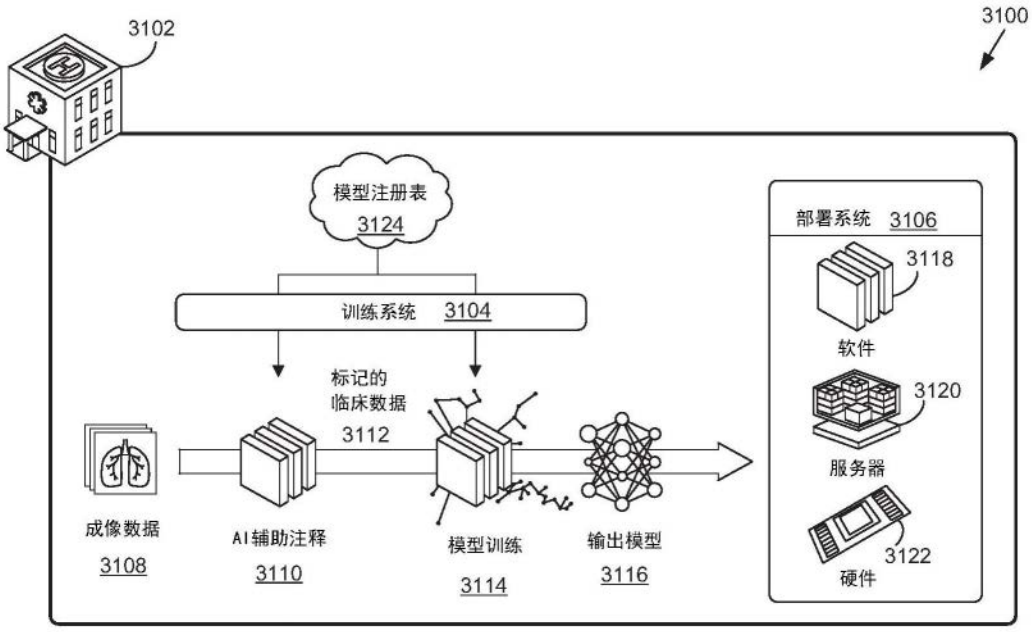

3100
3102
模型注册表
3124
训练系统 3104
标记的
临床数据
3112
成像数据
3108
AI辅助注释
3110
模型训练
3114
输出模型
3116
部署系统 3106
3118
软件
3120
服务器
3122
硬件

图31

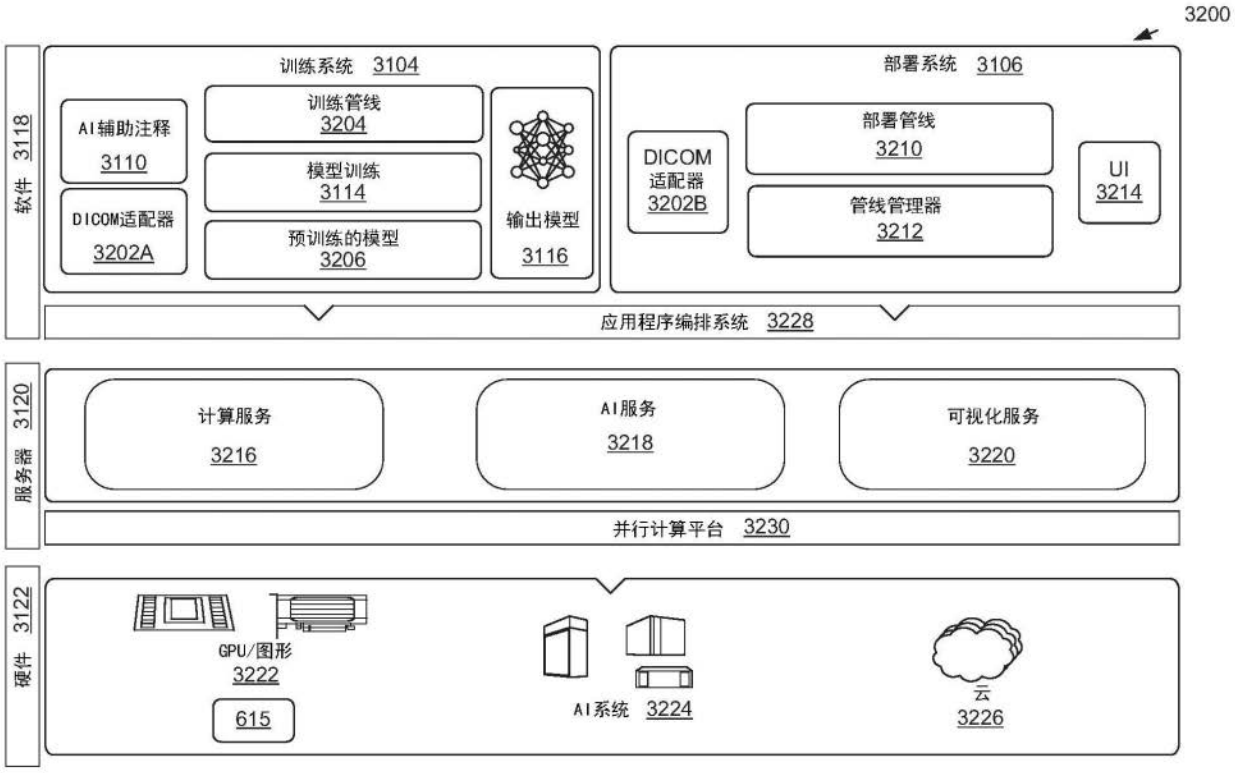

3200
软件 3118
训练系统 3104
AI辅助注释
3110
DICOM适配器
3202A
训练管线
3204
模型训练
3114
预训练的模型
3206
输出模型
3116
部署系统 3106
DICOM
适配器
3202B
部署管线
3210
管线管理器
3212
UI
3214
应用程序编排系统 3228
服务器 3120
计算服务
3216
AI服务
3218
可视化服务
3220
并行计算平台 3230
硬件 3122
GPU/图形
3222
615
AI系统 3224
云
3226

图32

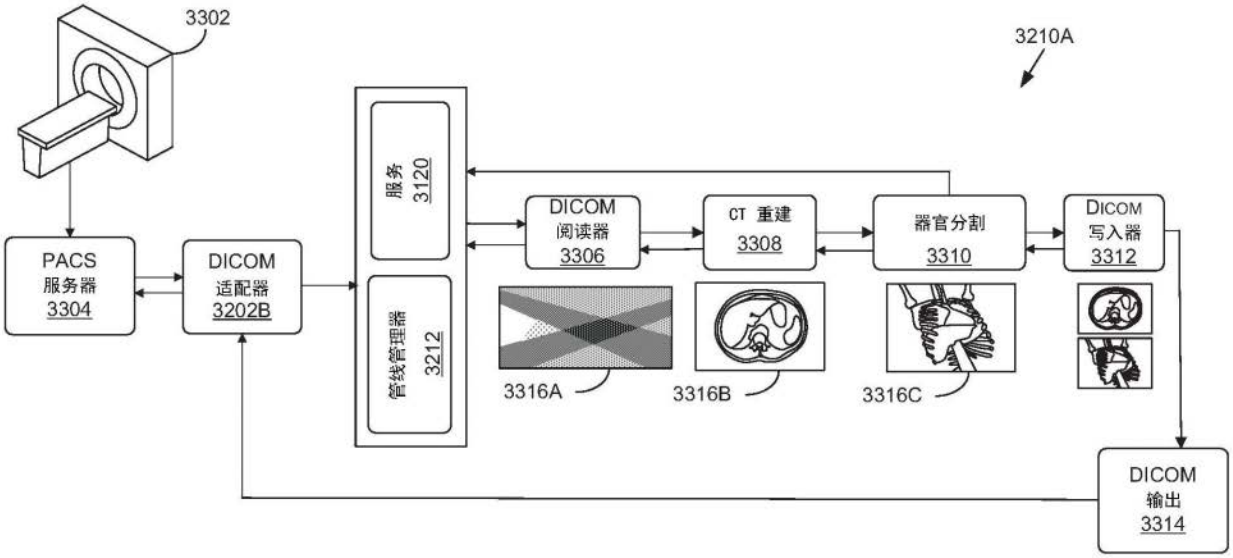

3302
3210A
服务
3120
管线管理器
3212
PACS
服务器
3304
DICOM
适配器
3202B
DICOM
阅读器
3306
CT 重建
3308
器官分割
3310
DICOM
写入器
3312
3316A
3316B
3316C
DICOM
输出
3314

图33

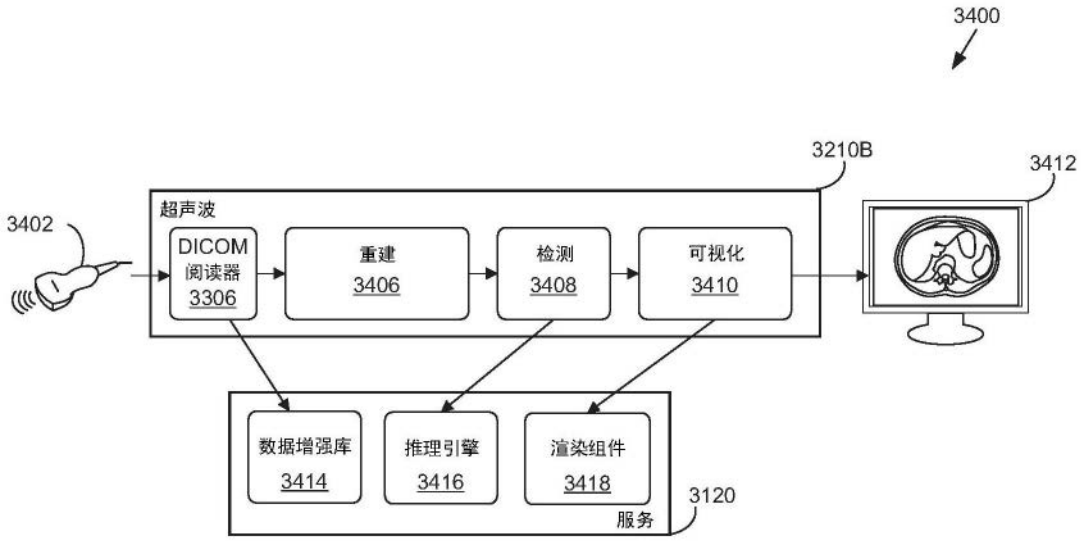

3400
3210B
3412
超声波
3402
DICOM
阅读器
3306
重建
3406
检测
3408
可视化
3410
数据增强库
3414
推理引擎
3416
渲染组件
3418
3120
服务

图34A

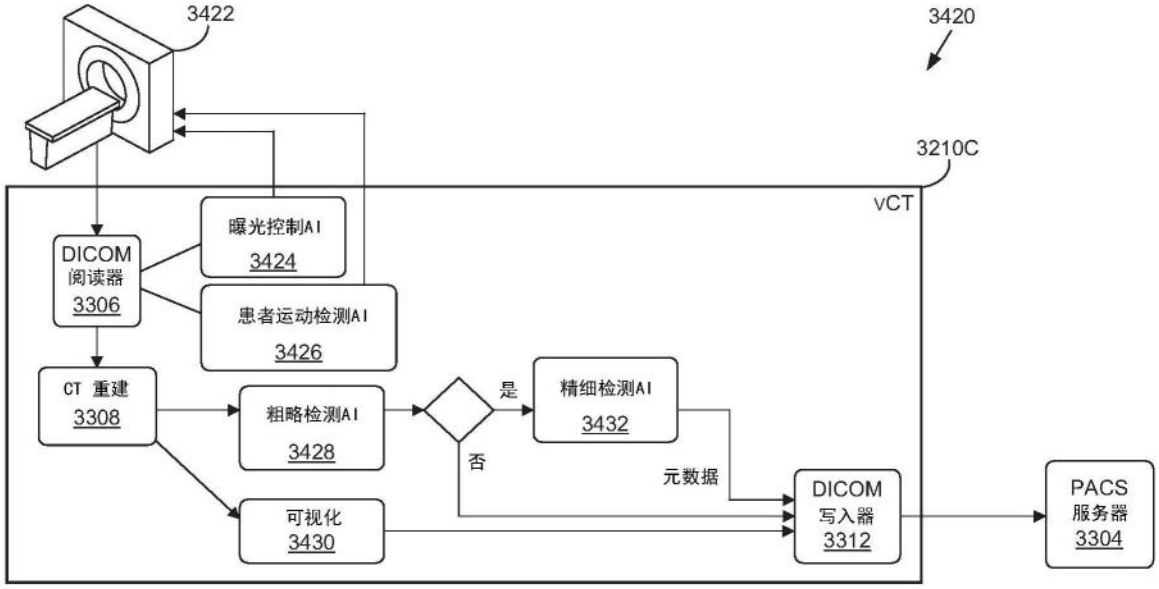

3422
3420
3210C
vCT
曝光控制AI
3424
DICOM
阅读器
3306
患者运动检测AI
3426
CT 重建
3308
粗略检测AI
3428
是
精细检测AI
3432
否
元数据
可视化
3430
DICOM
写入器
3312
PACS
服务器
3304

图34B

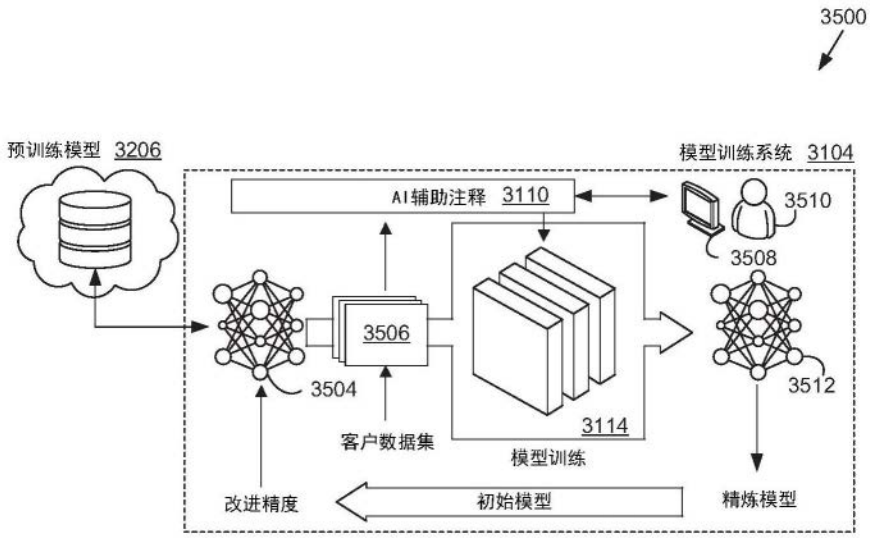

3500
预训练模型 3206
模型训练系统 3104
AI辅助注释 3110
3510
3508
3506
3504
3512
3114
客户数据集
模型训练
改进精度
初始模型
精炼模型

图35A

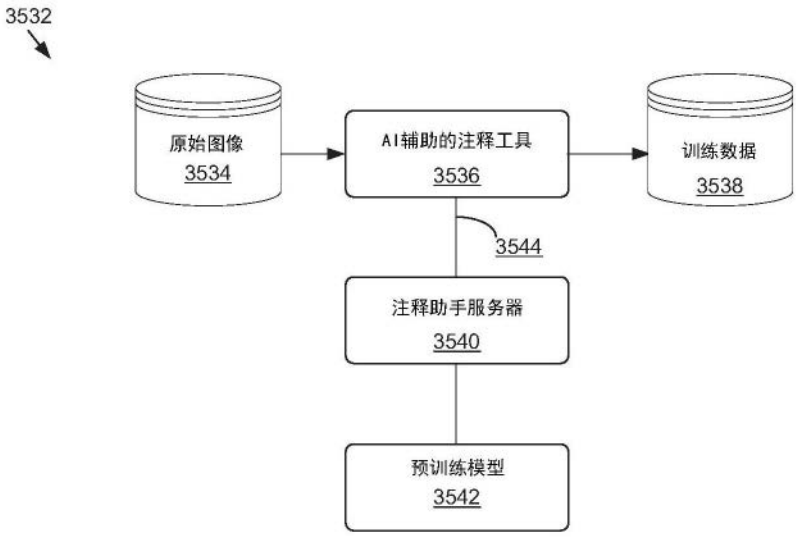

3532
原始图像
3534
AI辅助的注释工具
3536
训练数据
3538
3544
注释助手服务器
3540
预训练模型
3542

图35B