计算机研究与发展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202330483
61(11): 2846−2869, 2024

神经渲染及其硬件加速综述

严忻恺[1,3] 霍宇驰[1,2] 鲍虎军[1,2]
[1](计算机辅助设计与图形学国家重点实验室(浙江大学) 杭州 310058)
[2](之江实验室 杭州 311121)
[3](江苏省专用集成电路设计重点实验室(无锡)(江苏信息职业技术学院) 江苏无锡 214153)
(yanxinkai@zju.edu.cn)

Survey on Neural Rendering and Its Hardware Acceleration

Yan Xinkai[1,3], Huo Yuchi[1,2], and Bao Hujun[1,2]
[1](*State Key Lab of CAD & CG* (*Zhejiang University*), *Hangzhou* 310058)
[2](*Zhejiang Lab*, *Hangzhou* 311121)
[3](*Jiangsu Key Laboratory of ASIC Design* (*Wuxi*) (*Jiangsu Vocational College of Information Technology*), *Wuxi*, *Jiangsu* 214153)

Abstract Neural rendering is a new image and video generation method based on deep learning. It combines the deep learning model with the physical knowledge of computer graphics to obtain a controllable and realistic scene model, and realize the control of scene attributes such as lighting, camera parameters, posture and so on. On the one hand, neural rendering can not only make full use of the advantages of deep learning to accelerate the traditional forward rendering process, but also provide new solutions for specific tasks such as inverse rendering and 3D reconstruction. On the other hand, the design of innovative hardware structures that adapt to the neural rendering pipeline breaks through the parallel computing and power consumption bottleneck of existing graphics processors, which is expected to provide important support for future key areas such as virtual and augmented reality, film and television creation and digital entertainment, artificial intelligence and metaverse. In this paper, we review the technical connotation, main challenges, and research progress of neural rendering. On this basis, we analyze the common requirements of neural rendering pipeline for hardware acceleration and the characteristics of the current hardware acceleration architecture, and then discuss the design challenges of neural rendering processor architecture. Finally, the future development trend of neural rendering processor architecture is prospected.

Key words neural rendering; hardware acceleration; neural rending processing unit (NRPU); multilayer perceptron (MLP); ray marching; Hash table

摘 要 神经渲染是一种基于深度学习的新兴图像和视频生成方法,它将深度学习模型和计算机图形学的物理模型相结合,从而获得可控和逼真的场景表示和渲染,实现对诸如光照、相机参数、姿态等场景属性的控制.一方面,神经渲染既可以充分利用深度学习的优势,实现传统前向渲染流程的加速,也可以为影像的逆向渲染和3维场景重建等特定任务提供新的解决方案;另一方面设计适应神经渲染流水线的创新硬件结构,突破现有图形处理器的并行计算和功耗瓶颈,有望为虚拟现实和增强现实、影视创作和数字娱乐、人工智能和元宇宙等未来重点领域提供重要支撑.综述了神经渲染的技术内涵、主要挑战和研究进展,在此基础上分析了神经渲染流水线对硬件加速支持的共性需求和当前主流神经渲染硬件加速结构的

收稿日期: 2023-06-12; **修回日期**: 2023-12-08
基金项目: 之江实验室重点研究项目(K2022PG1BB01)
The work was supported by the Key Research Project of Zhejiang Lab (K2022PG1BB01).
通信作者: 鲍虎军(bao@cad.zju.edu.cn)

特点，进而讨论了神经渲染处理器架构的设计挑战，最后展望了其未来的发展趋势.

关键词 神经渲染；硬件加速；神经渲染处理器；多层感知机；光线步进；哈希表

中图法分类号 TP303

渲染是指计算机图形学中从3维场景生成2维图片的过程，而逆渲染是渲染的逆向过程，即从一系列输入的2维图片中重建3维场景. 传统渲染/逆渲染已经有成熟的软件栈和图形处理单元GPU支持，可用于游戏、电影、可视化展览等应用场景以及3维内容的制作和生成. 但是传统渲染/逆渲染仍存在依赖专业人员手工操作、专业软件复杂等问题，难以满足其他行业人员对3维内容生成和可视化渲染的需求. 随着深度神经网络的迅速兴起和发展，深度学习在计算机视觉、图像处理、自然语言处理等众多人工智能的交叉领域中取得了重大进步，因此将深度学习与计算机图形学相结合也顺理成章地成为热门研究方向. 一方面利用深度神经网络基于数据驱动的自动化设计方法来替代专业人员的人工设计，同时充分利用大量训练数据中提取的先验特征来加速渲染过程；另一方面基于深度学习的方法可以使整个逆渲染过程融入到基于梯度的优化中从而实现深度神经网络端到端的训练，使得逆渲染速度更快、结果更鲁棒.

神经渲染作为深度学习和计算机图形学的融合方法应运而生. Eslami等人[1]提出神经渲染的概念，该团队引入了一个生成查询网络(generative query network, GQN). GQN将不同数量的图像和对应的相机参数作为输入，然后将完整的场景信息编码为一个向量，并将这个向量作为生成网络的输入，得到正确遮挡的任意视图输出. GQN创新性的在不需要人力标注数据的情况下从数据中学习了一种强大的神经渲染器，能够产生准确的、全新视角的场景图像，虽然它的渲染效果有限，但是启发了大量的后续工作. Tewari等人[2]给出了神经渲染的定义：

神经渲染(neural rendering)是一种基于深度学习的图像和视频生成方法，它将深度神经网络模型和计算机图形学的物理知识相结合，从而获得可控和逼真的场景模型，实现对例如光照、相机参数、姿态等场景属性的控制.

和其他深度学习领域相比，神经渲染的重点是将传统渲染中的物理/数学知识和神经网络结构设计相结合，以及将计算机图形学领域的知识融入到神经网络的设计当中. 一般来说，神经渲染将神经网络看作一个通用函数逼近工具，通过输入真实世界场景的数据以及构建损失函数进行训练来模拟逼近物理/数学定律. 所以与传统渲染不同，神经网络对函数的逼近拟合效果直接影响到渲染质量，这也意味着训练输入数据集的数量、分布和质量都与最后的渲染效果密切相关.

1 综述现状

1.1 相关综述

Eslami等人[1]讨论了机器学习可以在哪里应用、如何改进经典的渲染流水线，以及神经渲染训练需要哪些数据. 文中简要介绍了物理图像生成和深度生成模型这2个领域的相关基础知识，展示了计算机图形学与机器学习混合模式的优点和局限性，该综述进而讨论了神经渲染的方法分类：控制、计算机图形学模块、显示或隐式控制、多模态合成和通用性这5个方向，并展开介绍了基于这5个方向的神经渲染相关应用，包括新视角合成、重光照、场景变形和合成等. 该文作者在文中表明该综述的目的是希望提升神经渲染相关的研究热度，从而有助于开发下一代神经渲染和图形应用程序.

近年来神经辐射场(neural radiance fields, NeRF)[3]及其变种优化相关研究的热度也应验了深度思维团队的预测. Tewari等人[4]对神经渲染的不同场景表示进行了全面的概述，简要介绍了经典渲染流水线和机器学习的组件的基本原理，重点介绍了将经典渲染与可学习3维表示相结合的高级神经渲染相关方法. 该综述中引用的相关文献主要集中在近2年的NeRF及其变种的优化工作，涉及的应用范围从刚性和非刚性场景的自由视角视频，到形状和材料编辑、光场和重照明以及数字人替身生成等. 此外Tewari等人[4]还探讨了神经渲染方法所带来的社会影响，包括在工业研究和生产、社会伦理和环境等各个层面.

与上文的2篇综述侧重不同，Wang等人[5]专注讨论了神经渲染的前向应用，即将深度神经网络与渲染组件相结合的渲染流水线相关应用. 文中简要介绍了基于物理的渲染和深度生成网络的理论基础，重点介绍了神经网络如何取代或增强传统渲染流水

线中渲染器的工作和这种组合的优缺点,涵盖了从环境遮挡生成、体积和次表面渲染、多场景表示渲染、全局光照渲染、直接光照渲染、与人相关渲染和渲染后处理等应用的通用和特定方法,并表明了传统图形渲染流水线可以部分或完全被深度学习增强渲染所取代的观点.

1.2 本文范围

不同于神经渲染算法应用的相关综述,本文更关注神经渲染在前向渲染和逆渲染2个方面与传统渲染流水线相结合或构建新型神经渲染流水线对硬件加速的共性需求.为了能更清晰地理解,本文首先简要概述了神经渲染的相关理论基础,然后介绍了神经渲染在前向渲染、逆渲染和后处理应用中具有代表性的研究成果.随后本文详细分析了神经渲染应用在计算和存储方面对硬件加速的共性需求,介绍了近期基于神经渲染硬件加速的相关工作,讨论了神经渲染处理器架构的设计挑战.

2 神经渲染

在给定高质量的场景描述前提下,传统渲染方法可以为各种复杂的现实世界现象渲染出逼真的图像,但是构建高质量的场景模型需要大量的手工工作.而神经渲染方法将特定场景条件(如视点、光照、布局等)相对应的图像作为数据输入,通过训练来构建一个基于“神经网络”的场景表示,并在新的场景属性下渲染这种表示,从而合成新的图像.神经渲染的“深度神经网络”通过学习来逼近真实函数,使得训练得到的场景表示不受简单场景建模近似的限制,可以针对高质量的新图像进行优化.

2.1 理论基础

本节讨论神经渲染工作的理论基础,首先介绍了传统计算机图形学中基于物理的渲染方法,然后概述了基于深度神经网络的生成模型的方法和3维场景的表示方法.

2.1.1 基于物理的渲染

1)光线投射.传统图形流水线将图像形成建模为现实世界中的物理过程:光源发射的光子与场景中的物体相互作用,形成由几何和材料属性决定的双向散射分布函数(bidirectional scattering distribution function, BSDF),然后由相机记录下来.这个过程被称为光线传输,可以用经典渲染方程[6]来表示:

$$L_{\mathrm{o}}(\boldsymbol{p},\omega_{\mathrm{o}},\lambda,t)=L_{\mathrm{e}}(\boldsymbol{p},\omega_{\mathrm{o}},\lambda,t)+L_{\mathrm{r}}(\boldsymbol{p},\omega_{\mathrm{o}},\lambda,t), \tag{1}$$

其中L_{o}表示从表面发出的亮度,它是位置$\boldsymbol{p}$、光线方向ω_{o}、波长λ和时间t的函数,L_{e}表示从表面直接发出的亮度,L_{r}表示入射光与表面反射率的相互作用:

$$L_{\mathrm{r}}(\boldsymbol{p},\omega_{\mathrm{o}},\lambda,t)=\int_{\Omega}f_{\mathrm{r}}(\boldsymbol{p},\omega_{\mathrm{i}},\omega_{\mathrm{o}},\lambda,t)L_{\mathrm{i}}(\boldsymbol{p},\omega_{\mathrm{i}},\lambda,t)(\omega_{\mathrm{i}}\boldsymbol{n})\mathrm{d}\omega_{\mathrm{i}}, \tag{2}$$

其中$\boldsymbol{n}$表示表面法线,ω_{i}表示入射光方向,L_{i}表示入射亮度,f_{r}表示BSDF函数,Ω表示表面点周围的半球.经典渲染方程是一个积分方程,最精确的近似是基于蒙特卡罗算法通过采样穿过场景的光线路径进行模拟.

2)前向渲染.将包括摄像机、灯光、表面几何和材料在内的场景定义转换为模拟摄像机图像的过程称为前向渲染.计算机图形学为渲染方程提供了各种近似,当前2种最常见的前向渲染方法是光栅化和光线追踪.光栅化是一个前向过程,它实际上是将3维场景转换到三角形,再投射到屏幕上,即使用透视投影将三角形的3维表示变成2维表示.光线追踪是一个递归过程,在这个过程中光线从图像像素向后投射到虚拟场景中,通过递归地从交叉点投射新光线来模拟反射和折射.当前主流GPU主要基于光栅化实现渲染过程的硬件加速,这是因为光栅化对并行计算友好、适合硬件加速且具有良好的内存一致性.然而由于在全局照明和其他复杂光(如景深、运动模糊)等场景下,光线追踪的效果要好于光栅化,所以近年来硬件厂商也在其GPU中也添加了光线追踪硬件加速结构,从而支持在实时渲染流水线中支持光线追踪[7-8],通过光栅化与光线追踪混合渲染来获得更好的渲染效果.

3)逆渲染.从真实数据中估计不同模型参数(如相机、几何、材料、光参数)以生成新的视图,编辑材料或照明的过程被称为逆渲染,通俗来说逆渲染要解决的问题就是在已有图片的条件下如何构建生成该图片的3维场景.由于数学复杂性或计算开销,所以经典渲染中使用的预定义物理模型或数据结构并不总是准确地再现现实世界物理过程的所有特征,这是逆渲染面临的主要问题.而深度神经网络可以在统计上近似逆渲染的物理过程,使得输出数据与训练数据更接近,从而能更准确地再现真实世界的效果.

2.1.2 深度神经网络

传统计算机图形学方法侧重于使用物理/数学公式建模场景和模拟光传输来生成图像,而深度神经网络可以通过学习真实世界中的图像分布,从统计概率的角度来解决这个问题.

在机器学习领域中经常使用多层感知机(multilayer perceptron, MLP)作为通用函数逼近器[9].MLP是一种传统的全连接神经网络.在场景表示的上下文中,MLP

将空间中的坐标作为输入，并产生与该坐标对应的一些值作为输出，这种类型的网络也称为基于坐标的神经网络. 基于 MLP 的函数逼近器既可用于表示表面或体积，也可用于存储其他属性，自 NeRF 之后大量的神经渲染工作都采用 MLP 作为函数逼近器.

深度生成网络擅长生成具有类似训练集统计数据的随机真实图像，传统的生成对抗网络（generative adversarial nets, GANs）[10] 从随机向量中合成具有类似训练数据的虚拟图像，但这对于场景渲染来说是远远不够的. 当前神经渲染将深度生成网络使用感知距离[11] 进行训练或使用条件生成对抗网络（conditional generative adversarial nets, cGANs）网络进行学习，它们都使用预训练辅助网络来定义一个有效的训练目标，从而训练出更好的生成网络和渲染效果.

源于自然语言处理的 Transformer 模型[12] 提出的自感知机制将序列中的任意 2 个位置之间的距离缩小为 1 个常量，从而解决长期依赖问题，通过矩阵计算可以直接计算每个词之间的相关性，不需要通过隐藏层传递. 这种思想启发并简化了神经渲染使用 MLP 学习从位置到特定值的映射的任务，还可以用于推理几何和外观.

2.1.3 3 维场景表示

为了对 3 维场景中的物体建模，已经提出了许多不同的场景表示方法，主要分为显式表示和隐式表示 2 类，而显式和隐式表示方法都可以用于场景的表面表示和体表示，主流 3 维数据格式包括：深度图、体素[13]、点云和网格.

1）深度图. 深度图是包含与视点场景对象表面距离有关信息的图像通道，通道本身类似于灰度图像，每个像素值是传感器测出距离物体的实际距离. 由于深度图中包含的都是场景中最前面物体表面的信息，所以不需要额外计算可见性. 此外深度图对神经网络非常友好，是一种非常适合训练和通过神经网络计算的格式.

2）体素. 体素通常用于表示体积. 它们可以存储几何占用率、具有体积效果（比如透明度）场景的密度值和场景的外观[14] 等. 传统渲染很少用到体素，因为它既要计算可见性又要计算阴影，但体素对神经渲染相对友好，因为神经网络可以使用体素来学习端到端的渲染流水线.

3）点云. 点云是欧几里得空间的一组元素，场景的连续表面可以用点云离散化，即点云的每个元素代表表面上的一个样本点 (x, y, z)，此外点云还可以表示体积（如存储不透明度或密度值）. 在神经渲染中，点云可以存储一些可学习的特征[15-16]，从而将单个特征投影和融合到 2 维特征图中，主要应用于降噪、风格化和新视图合成等场景[16-18]. 但由于点云数据的数量、顺序与表面点的采样方式相关，所以对神经网络相对不友好.

4）网格. 多边形网格表示表面的分段线性逼近，在传统图形学中大多数图形编辑工具都使用网格作为标准表示. 为了和传统渲染流水线兼容，许多神经渲染方法使用网格表示.

2.1.4 神经场景表示

在神经渲染的背景下，使用神经网络来近似表面或体积表示函数的场景表示称为神经场景表示. 表面和体积表示都可以存储额外的信息，如颜色或辐射度等.

符号距离函数（signed distance function, SDF）是最常用的隐式曲面表示，它会返回空间中任何一个点到物体表面的最短距离，距离为 0 的所有点即为物体表面，而距离小于 0 的点代表在物体内部，距离大于 0 的点代表在物体外.

神经隐式表面[19-21] 在神经渲染中由符号距离函数和 MLP 相结合得到，并被用于形状建模. MLP 将连续坐标映射到符号距离值，这种表示方法已经在神经场景表示和渲染中得到了广泛的应用. 神经隐式表面表示有很多优点，如内存效率高、鲁棒性较强、可以减少显式计算量、理论上可以以无限分辨率表示几何等.

神经体素. 神经体素（neural volumetric）和神经隐式表面类似，它表示使用神经网络定义体素而不是使用体素网格存储特征或其他变量，在神经渲染中一般使用基于坐标的神经网络对场景进行体建模. 由于 MLP 网络可用于将体积进行参数化表示，所以神经体素表示可能比显式体素网格表示更高效.

体渲染（volume rendering）基于光线投射[13]，主要用于云和烟这些非刚性物体的渲染建模，它把光子与粒子发生作用的过程分为吸收、放射、外散射和内散射 4 种类型. NeRF[3] 及其相关工作[22-27] 已证明体渲染在从多视图输入数据中学习场景表示方面具有较好的效果，可以在基于神经渲染的逆渲染框架中使用.

2.2 应 用

本节按照前向渲染、逆渲染和后处理的分类简要介绍了神经渲染技术在这 3 类领域内的最新或代表性应用.

2.2.1 前向渲染

图 1 给出了传统前向渲染流水线示意图，分为场

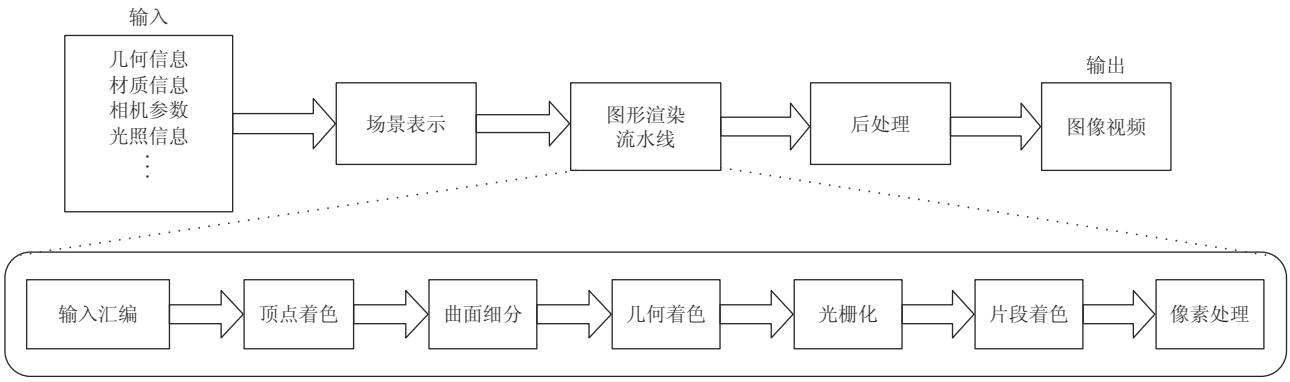


Fig. 1 Pipeline of forward rendering

图 1 前向渲染流水线

景表示、图形渲染流水线和后处理 3 个阶段. 神经渲染技术在前向渲染中的作用通常为使用深度神经网络对传统渲染过程中的各子模块进行增强, 如体素场景表示、全局光照等.

表 1 给出了神经渲染在各渲染子模块的相关研究工作以及网络的输入数据类型.

图 2 给出了 RenderNet[28] 实现的端到端神经体素渲染过程. 它首先将输入的体素网格、相机姿态和光源位置转换成相机坐标; 然后使用 3 维卷积神经网络(3D convolutional neural network, 3D CNN)将相机坐标转换为 4 维张量(H, W, D, C)表示的神经体素; 接着使用投影单元将神经体素转换成 3 维张量表示的神经像素(对深度维度 D 和特征通道维度 C 做矩阵变维处理); 最后使用 2D CNN 网络进行反卷积运算将投影的神经体素渲染成图片. neural voxel renderer[29] 提出了一种基于深度学习的渲染方法将体素化场景映射到高质量图像. 文献 [29] 设计了 2 个神经渲染器 NVR 和 NVR+来渲染场景, NVR 和 Render Net 非常相似, 区别在于 NVR 的投影单元使用 2 层 MLP 对光位置进行处理, 从而实现对光照信息的编码. NVR+将在 2.2.2 节中介绍.

Table 1 Application of Neural Rendering in Forward Rendering

表 1 神经渲染在前向渲染中的应用

阶段	子模块	工作	输入数据	泛化性
场景表示	基于体素	RenderNet[28]	体素、相机姿态、光源位置	通用
		neural voxel renderer[29]	体素、相机姿态、光源位置	通用
	基于点云	neural point-based graphic[30]	点云、描述符、相机姿态	通用
		neural point cloud rendering[31]	点云、相机姿态	特殊
		neural light transport[32]	点云、场景特征	特殊
	基于网络	differentiable volumetric[33]	像素、	特殊
		neural implicit surfaces[34]	图片集/SDF	特殊
全局光照	间接光照	neural light transport[32]	点云、场景特征	特殊
		deep illumination[35]	法线图、直接光照、漫反射图、深度图	通用
		neural control variates[36]	相机姿态、概率密度	通用
	直接光照	neural screen space rendering[37]	材料、几何形状、光照	通用
空间效果	环境阴影	deep shading[38]	位置、法线、反射率、颜色	通用
	环境遮挡	AOGAN[39]	位置、法线	通用

点云神经图形 (neural point-based graphic)[30] 提出了一种基于点云的真实场景外观建模方法, 使用原始点云作为场景的几何表示, 并通过编码局部几何和外观以及使用训练好的神经描述符来增强每个点云. 文中使用 Z-Buffer 在几种分辨率下的光栅化点作为神经渲染网络输入数据, 使用神经描述符作为伪颜色, 网络架构类似于 U-Net, 通过传递光栅化来渲染图像. 该模型通过优化神经渲染网络的参数和通过反向传播损失函数的神经描述符来适应新场景. 神经点云(neural point cloud)[31] 在此基础上提出了一种通过多平面投影(multi-plane projection, MPP) 的神经点云渲染流水线, 通过将 3 维点投影到摄像机视锥的分层体中的方法来方便神经网络自动学习 3 维点的可见性. 其网络框架由 2 个模块组成: 基于多平面的体素化和多平面渲染. 基于多平面的体素化模块根据图像大小和预定义的平面数量, 将摄像机视锥的 3 维空间均匀划分为小视锥体素, 同时对每个小视锥进行聚合操作, 生成多平面 3 维表示; 多平面渲染模

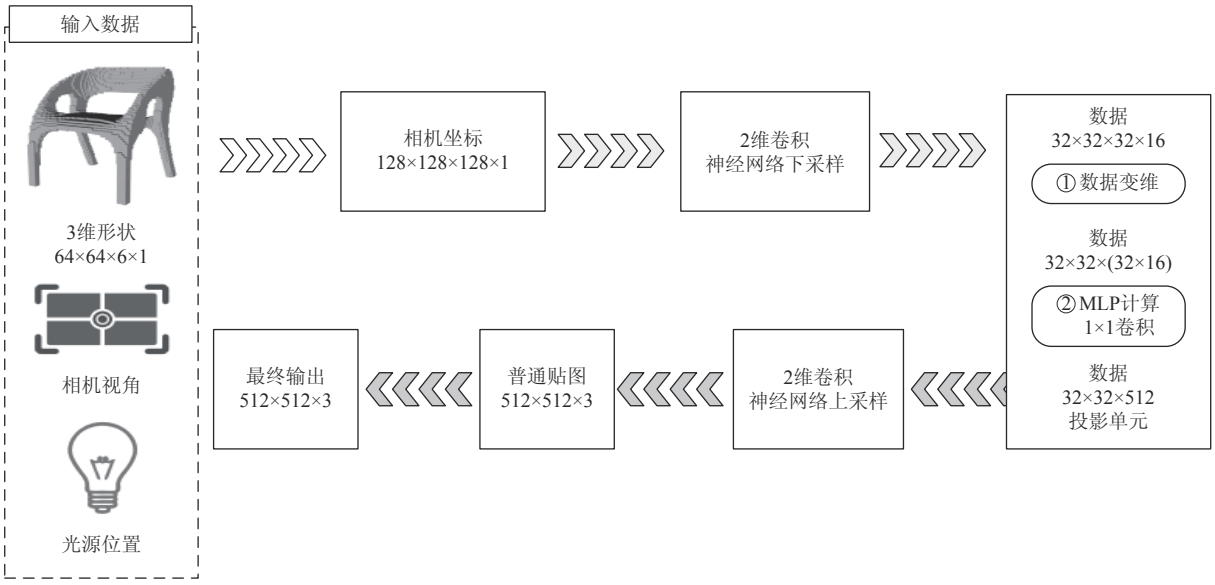


Fig. 2 Network architecture of RenderNet

图 2 RenderNet 网络结构

块作为渲染网络为每个平面预测 4 通道输出(RGB+混合权重),最终根据混合权重混合所有平面产生渲染图像. 神经光传输(neural light transport, NLT)[32] 提出一种使用神经网络学习静态和动态 3 维场景中的光传输的方法. 图 3 给出了 NLT 的具体模型结构. 文中的神经渲染网络架构主要有 3 个模块: 1)光传输层; 2)视角变换、环境遮挡过滤和投影层; 3)图像合成层.其中光传输层使用 MLP 来实现带特征的点云到处理后特征的转换,图像合成层使用类似带残差块的 U-Net 架构来实现图像渲染. 与现有 2 维图像域方法相比,支持在 3 维和 2 维空间中进行推理,从而实现全局光照效果和 3 维场景几何处理,并且该模型能够产生静态和动态场景的真实感渲染.

可微体渲染(differentiable volumetric rendering, DVR)[33] 提出了一种隐式形状和纹理表示的可微渲染公式. 该文作者设计了一个占用网络,为 3 维空间中的每个点分配了占用概率,使用等值面提取技术来提取物体表面,同时直接使用纹理场[40] 从 RGB 图像中学习隐式形状和纹理表示. Yariv 等人[34] 将体积密度函数定义为拉普拉斯的累积分布函数(cumulative distribution function, CDF),应用于 SDF 表示. 该文作者认为这种简单的密度表示有 3 个好处: 1)为在神经体渲染过程中学习的几何提供了有用的归纳偏差; 2)有助于对不透明度近似误差进行约束,从而导致对观察光线的精确采样; 3)允许在体素渲染中对形状和外观进行有效的无监督解缠. 其网络框架由 2 个 MLP 组成: 第 1 个 MLP 用于近似几何 SDF 和 256 维的全局几何特征,第 2 个 MLP 用于表示场景辐射场.

深度光照(deep illumination, DI)[35] 在实时应用中用条件生成对抗网络 cGAN 来近似全局光照,提出了

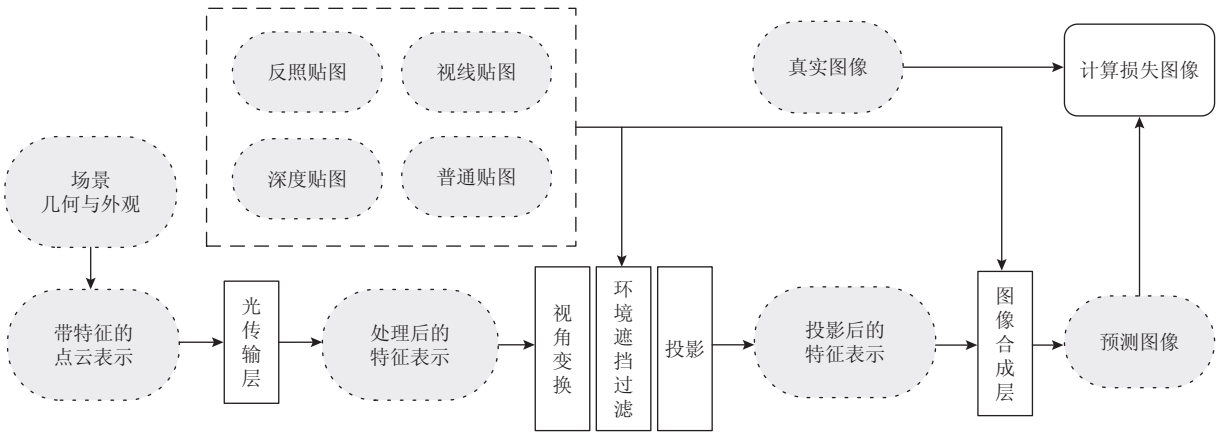


Fig. 3 Model architecture of neural light transport

图 3 神经光传输模型结构

一个从 Gbuffer(深度图、法线图和漫反射图)和直接光照到任何全局光照的映射网络,其中生成模型可用于学习从屏幕空间缓冲区到 3 维环境的高级照明模型的密度估计. 神经控制变量(neural control variates, NCV)[36] 提出了一种在参数蒙特卡罗积分中降低无偏方差的方法,使用神经网络来学习一个接近渲染方程的函数、一个神经采样器来产生采样概率,以及一个神经网络来推断积分方程的解. 神经直接光照渲染器(neural direct-illumination renderer, NDR)[37] 可以在远处照明下渲染具有不透明材料的任何几何形状的直接照明图像. 图 4 给出了 NDR 具体的模型结构. 首先将屏幕空间缓冲区(材料、几何形状和光照)输入到 CNN 网络中学习从像素缓冲区到渲染图像的映射,然后 NDR 预测阴影图,并将其与反照贴图相结合来渲染图像.

深度阴影(deep shading)[38] 提出了一种生成多种渲染效果的新技术,包括利用延迟着色缓冲区和卷积神经网络的环境阴影,使用深度神经网络作为渲染器,将延迟着色缓冲区作为输入,生成特定的渲染结果. AOGAN[39] 提出一种端到端的 GAN 用于产生逼真的环境阴影,通过将 VGG 结构的感知损失与 GAN 结构的对抗损失结合起来探索了生成模型中感知损失对环境阴影精度的重要性,同时引入了一个以位置和正常着色缓冲区为输入的自注意力模块来提高训练的泛化性.

2.2.2 逆渲染

图 5 给出了逆渲染流水线示意图,分为场景观察、逆渲染流水线和后处理 3 个阶段. 神经渲染技术在逆渲染中的作用通常为实现深度神经网络,如新视图合成、重照明、材质编辑等.

表 2 给出了神经渲染在逆渲染的相关研究工作以及对应的生成网络类型和 3 维场景表示类型.

neural voxel renderer[29] 提出的 NVR+网络在 NVR 网络的基础上增加了溅射处理网络和神经重渲染网络. 溅射处理网络首先通过分裂目标视图中的彩色体素中心来合成图像,然后将该图像和 NVR 生成的特征向量相连一起传递给 2 维卷积编码器,最终结果由神经重渲染网络 U-Net 处理以生成输出图像. Deepvoxels[41] 对 3 维场景的视图外观进行编码,实现了端到端的逆渲染. 它首先使用 2 维 U-Net 网络提取 2 维特征映射,然后基于可微分提升层将 2 维特征提升到 3 维特征体,接着在特征融合后使用 3 维 U-Net 对特征体进行处理,通过可微重投影层将特征体映射到 2 个目标视图的摄像机坐标系中. 然后遮挡网络计算每个体素的软可见性,最后使用 2 维 U-Net 渲染网络生成 2 个最终输出图像. neural volumes[42] 提出了

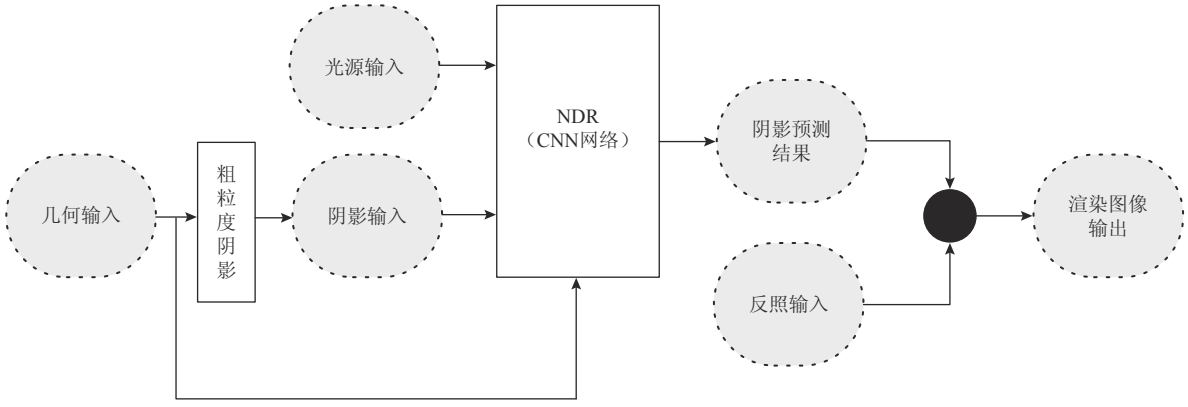


Fig. 4 Model architecture of neural direct-illumination renderer

图 4 NDR 模型结构

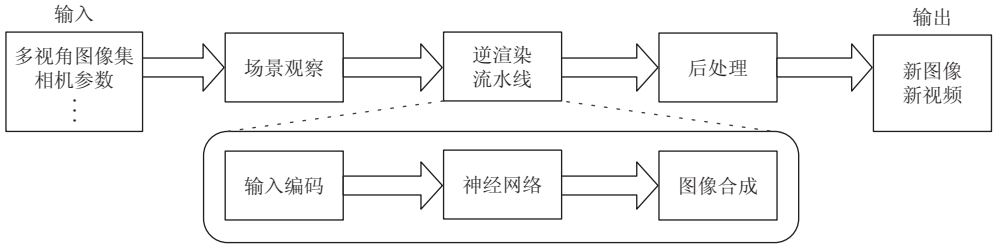


Fig. 5 Pipeline of inverse rendering

图 5 逆渲染流水线

Table 2 Application of Neural Rendering in Inverse Rendering

表 2 神经渲染在逆渲染中的应用

功能分类	工作	生成模型	3 维场景表示	输入数据类型
新视图合成(体素类)	neural voxel renderer[29]	U-Net	体素	体素
	Deepvoxels[41]	U-Net	体素	体素
	neural volumes[42]		体素	图像
新视图合成(NeRF 类)	NeRF[3], AutoInt[43]		MLP	图像相机姿态
	neural sparse voxel fields[44], KiloNeRF[45]		神经体素+网格	图像相机姿态
	baking NeRF[46], FastNeRF[47], GeoNeRF[48], PlenOctrees[49], light field networks[50], instant neural graphics primitives[51]		网格	图像相机姿态
	D-NeRF[24], neural scene flow fields[52]		MLP	视频相机姿态
	GIRAFFE[23]	CNN	神经特征场	图像
	pixelNeRF[53]		神经体素	图像相机姿态
	IBRNet[54], common objects in 3D[55]		神经体素	图像相机姿态
新视图合成和场景重建(SDF 类)	DeepSDF[19]	CNN	SDF	图像相机姿态
	I2-SDF[21]	MLP	SDF	图像相机姿态
	scene representation networks[56]	CNN	SDF	图像相机姿态
重光照和材质编辑	I2-SDF[21]	MLP	SDF	图像相机姿态
	neural reflectance fields[57]	CNN	神经体素	图像光参数
	neural light transport[58]	MLP	神经体素	图像光参数
	NeLF[59]	MLP	神经特征向量	肖像相机姿态

一种基于学习的方法来表示动态对象，它采用一个编码器-解码器网络，将输入图像转换为 3 维体素表示同时使用可微射线进行操作，从而实现端到端训练. neural volumes 的创新性在于任何输入图像都可以通过重渲染的损失作为监督，不需要显式重建或跟踪目标.

NeRF[3] 标志着基于 MLP 网络的场景表示在应用于单场景逼真新视图合成方面的突破，图 6 给出了 NeRF 模型结构及可微渲染过程. NeRF 直接应用体素渲染模型从 MLP 中合成图像，MLP 将位置和方向映射到体密度和颜色，然后根据对输入图像的像素级渲染损失优化 MLP 权重，以表示每个新的输入场景. NeRF 最大的创新是引入了位置编码(positional encoding)，使得基于 MLP 的场景表示在优化过程中可以有效地对场景进行微分压缩，从而获得比离散 3 维体素表示更高的分辨率，同时可以在不增加 MLP 权重数量的情况下表示更高频率的信号. 但是由于 NeRF 中单个点的颜色和体密度都需要神经网络计算，带来了巨大的计算开销，导致 NeRF 的渲染速度慢. 此外 NeRF 还存在泛化性差、只支持静态场景和需要大量样本训练等问题. 部分工作[44-47,49] 针对 NeRF 架构进行优化来提升渲染速度；AutoInt[43] 和 light field networks[50] 通过训练 MLP 自身来提升渲染速度；Müller 等人[51] 则针对 NeRF 训练速度进行优化，引入了多分辨率哈希编码来减轻 MLP 训练负担，从而加快训练速度.

Pumarola 等人[24] 和 Li 等人[52] 针对 NeRF 静态场景进行改进，D-NeRF[24] 将时间作为输入，将学习过程分为 2 个主要阶段：将场景编码为规范空间和在特定时间将规范表示编码映射到变形的场景，同时使用全连接网络学习；neural scene flow fields[52] 将动态场景建模为外观、几何体和 3 维场景运动的时变连续函数. 过去几年有大量针对 NeRF 泛化性进行优化的工作，GIRAFFE[23] 学习 2 维图像中每个像素的局部特征，然后将这些特征投影到 3 维点，从而产生通用和丰富的点表示；GeoNeRF[48] 使用 Transformer 来渲染推理几何和外观，同时使用体渲染来绘制图像细节，从而实现了一种基于 NeRF 的可泛化真实感新视图合成方法 pixelNeRF[53] 在体渲染框架中利用特征在多个视图上聚合进而以 NeRF 方式渲染使 MLP 产生颜色和密度场，当在多个场景上进行训练时，它们会学习用于重建的场景先验，从而能够从少

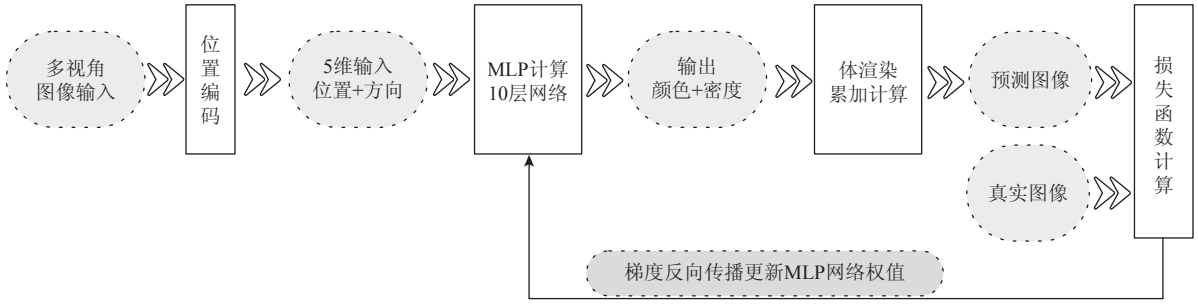


Fig. 6 Model architecture and differentiable render procedure of NeRF

图 6 NeRF 模型结构及可微渲染过程

量视角高保真地重建场景; IBRNet[54]引入 Ray-Transformer 网络来学习通用的视图插值函数, 支持泛化到新场景; NeRFormer[55]采用 Transformer 网络实现在少量视图的情况下重建物体.

DeepSDF[19]提出一种可学习的连续符号距离函数(SDF)表示, 能从部分和有噪声的 3 维输入数据中实现高质量的形状表示、插值和补全. 场景表示网络(scene representation networks, SRNs)[56]主要由 MLP 场景表示模块、光线步进 LSTM 模块和像素生成器模块组成, 该网络基于 SDF 隐式函数实现了在没有显式监督的情况下对一堆图像进行端到端训练, 从而实现新视图合成、少样本重建以及形状和外观插值. I^2-SDF[21]基于整体神经 SDF 框架从多视角图像中恢复基本形状、入射辐射度和材料, 通过基于表面的可微蒙特卡罗光线跟踪和辐射原语分割, 将神经辐射场分解为场景的空间变化材质, 在室内场景重建、新视图合成和场景编辑方面显示出优越质量.

神经反射场[57]提出了 NeRF 的第 1 个扩展以实现重光照, 将场景表示为体积密度场、表面法线和双向反射分布函数(bidirectional reflectance distribution function, BRDF)场, 从而允许在任意光照条件下渲染场景, 采用的方法是在每个 3 维位置使用预测的表面法线和 BRDF 来评估有多少入射光线从该位置的粒子反射到相机, 且通过只训练与相机共存的单点光照明的物体的图像来减少 MLP 计算量. 神经反射和可见场[58]训练 MLP 来近似任何输入 3 维位置和 2 维入射光方向的光源可见性, 通过可见光 MLP 只需要对每个入射光方向进行 1 次查询, 从而能够从具有明显阴影和自聚焦效果的图像中恢复场景的重光照模型. NeLF[59]提出一种人像视图合成和重照明系统, 用神经网络来预测 3 维空间中的光传输场, 并从预测的神经光传输场在新的环境照明下从新的相机视图产生人像, 从而实现同时对给定的多视角人像的视图合成和重光照.

2.2.3 后处理

由于更高的分辨率和刷新率以及更逼真的效果给实时渲染带来了很大的挑战, 所以后处理是实时渲染常用的技术手段之一. 传统后处理主要采用线性插值方法, 如双线性插值算法[60]、双 3 次插值算法[61]等, 神经渲染在后处理阶段使用神经网络来减轻渲染流水线的负担. 超分辨率和插帧技术能够以较低的分辨率或帧速率渲染流水线工作, 并通过深度学习方法恢复目标分辨率和帧速率. 基于深度学习的后处理方法利用卷积运算可以更有效地自动提取图像细节特征, 解决底层视觉问题, 生成质量更优的高分辨率图像(降噪、去模糊等), 比如使用残差网络来增强层间连接, 将低分辨率图像特征信息传到更深层, 缓解梯度消失和特征丢失问题, 使用 GAN 中多个生成器的感知损失和对抗损失信息, 提升高分辨率图片的真实感.

NVIDIA 公司于 2019 年发布了深度学习超级采样(deep learning super sampling, DLSS)技术, 目前已经更新到 3.0 版本[62], 它利用基于深度学习的超分辨率技术, 在减少像素渲染的基础上构建更清晰、更高分辨率的图像. DLSS 卷积自动编码帧生成器会接收 4 个输入数据: 当前游戏帧、前游戏帧、光流加速器生成的光流场, 以及游戏引擎数据(例如运动矢量和深度). 对于每一帧, DLSS 帧生成网络都会决定如何使用来自游戏运动矢量、光流场和后续游戏帧的信息生成中间帧. Zhang 等人[63]使用一种 CNN 和 Transformer 的混合网络架构, 通过统一操作显式提取运动和外观信息, 这种混合架构可以降低帧间注意力的计算复杂度, 并保留细节的底层结构信息. Bi 等人[64]提出一种提高低质量合成图像视觉真实感的方法, 作者首先以基于物理的渲染为目标, 学习以监督的方式预测准确的阴影, 然后使用改进的 cycleGAN 网络[65]来进一步增加纹理和阴影的真实感.

3 硬件加速

随着神经渲染技术被广泛应用于实时渲染的各个领域, 神经渲染所需的计算量也随着所使用神经网络复杂度的增加而增长, 所以在硬件层面考虑如何更好地加速神经渲染应用也成为研究热点. 在 2023 年计算机体系结构顶会 ISCA(International Symposium on Computer Architecture)新兴视觉/图形/AR-VR 子专题录用的 6 篇论文中, 有 4 篇是神经渲染硬件加速的相关工作, 这也表明神经渲染硬件加速结构已经逐渐成为重要的研究热点.

硬件架构师应分步思考 3 个问题: 1)神经渲染任务的共性需求和性能瓶颈; 2)现有的通用处理器(CPU)、图形处理器(GPU)或通用图形处理器(general-purpose GPU, GPGPU)能否满足神经渲染任务的计算和访存要求, 新兴的人工智能处理器 neural processing unit(NPU)、tensor processing unit(TPU)是否能具备神经渲染任务所需的算子或特殊硬件支持; 3)如果 2)的答案是否定, 那么神经渲染是否需要专用硬件加速的支持.

3.1 共性算法

神经渲染任务使用了各种类型的神经网络算法和传统图形学中的部分典型算法，本节分析了第2节中介绍的各神经渲染技术的共性需求.

3.1.1 神经网络

由于神经渲染是一种基于深度学习的技术，所以神经渲染算法所采用的深度神经网络是各个算法的核心，表3总结了本文第2节中介绍的各项工作所使用的神经网络类型以及主干网络的架构组成.

MLP主要由全连接层(fully connected layers, FC)和激活层(activation layers)组成；CNN主要由卷积层(convolution layers)、激活层、归一化层(batch normalization layers, BN)、池化层(pooling layers)和全连接层组成；GAN是一个由生成器(G)和判别器(D)两个模型组成的系统，生成器和判别器的主要网络架构为卷积层；U-Net是一种端到端的编码器-解码器结构，编码器主要由卷积层组成，解码器主要由反卷积层组成；Transformer本质上也是一个编码器-解码器的结构，Transformer中最重要的结构是自注意力(self-attention)模块，它需要进行2次矩阵-矩阵乘计算；多头注意力(multi-head attention)模块将H个自注意力模块的输出进行拼接处理和线性变换得到最终的特征矩阵；特征矩阵经过归一化残差(add & norm)层后再被送入前馈神经网络(feed forward neural network, FFN)，FFN是一个具有2层FC层的网络.

Table 3 Types of Neural Network in Neural Rendering Application

表3 神经渲染应用中的神经网络类型

神经网络	工作	残差层	拼接层
MLP	GIRAFFE[23],neural light transport[32], neural relighting[58],NeLF[59]	需要	需要
	DeepSDF[19],pixelNeRF[53],IBRNet[54], neural radiosity[66], common objects in 3D[55],NeRF[3],NeRF in the wild[22], RenderNet[28] ,neural voxel renderer[29],neural volumes[42] neural sparse voxel fields[44],KiloNeRF[45],baking neural radiance fields[46], FastNeRF[47],PlenOctrees[49],AutoInt[43],light field networks[50], instant neural graphics primitives[51],neural scene flow fields[52], scene representation networks[56],extracting motion and appearance[63] ,Instant-3D[67]	不需要	需要
CNN	GIRAFFE[23], RenderNet [28],neural voxel renderer[29,32],texture fields[40], neural reflectance fields[57],extracting motion and appearance[63]	需要	需要
	neural volumes[42],neural point cloud rendering[31], neural screen space rendering[37],deep shading[38]	不需要	需要
	neural control variates[36],AOGAN[39],pixelNeRF[53], scene representation networks[56]	需要	不需要
	neural light-transport field[59]	不需要	不需要
GAN	deep illumination[35]	不需要	需要
	AOGAN[39]	不需要	不需要
U-Net	Deepvoxels[41],neural point-based graphics[30],deep illumination[35]	不需要	需要
Transformer	IBRNet[54],common objects in 3D[55],GeoNeRF[48], extracting motion and appearance[63],GeoNeRF[68]	需要	需要

注：残差层（resblk）是将第1层单元的输入直接与第2层单元输出线性堆叠在一起后再激活的计算层，可以缓解梯度消失问题；拼接层（concat）是将多个输入数据连接到一个输出数据，实现了输入数据的拼接，可以改变数据维度、增加通道数.

深度神经网络的主要共性算子如表4所示，主要有卷积/反卷积、矩阵-矩阵乘、矩阵向量乘、池化/反池化、Element-Wise乘/加操作、激活函数以及归一化. 卷积、矩阵-矩阵乘等算子是神经网络中计算量占比最高的算子，需要硬件设计专用运算单元进行加速，如脉动阵列、点积、多级累加树等专用单元. 这些硬件单元相较于通用处理器的单指令多数据(single instruction multi data, SIMD)和图形处理器的单指令多线程(single instruction multi thread, SIMT)的数据并行方式具有更好的加速效果；对于池化/反池化、激活函数等算子，虽然它们计算量占比不大，但由于其串行执行的特性会占用程序关键路径的运行时间，所以也需要设计专用硬件单元来进行加速，比如并行查表、超越函数/插值等特殊运算单元；对于

Table 4 Neural Network Operator Requirement of Neural Rendering Application

表4 神经渲染应用中神经网络的算子需求

算子	主要作用	算子特征	硬件加速需求	神经渲染过程
卷积/反卷积	特征提取、数据升维、图像合成	计算需求高	张量/矩阵计算单元	训练/渲染
矩阵-矩阵乘	注意力机制、坐标转换、特征映射	计算需求高	张量/矩阵计算单元	训练/渲染
矩阵-向量乘	反向传播、注意力机制、概率生成	计算需求高	张量/矩阵计算单元	训练/渲染
池化/反池化（最大、平均）	数据降维、下采样、扩大感受野	位于关键路径	通用计算单元、执行流水线	训练/渲染
Element-Wise乘/加操作	残差模块、偏置	访存需求高	通用计算单元、数据传输单元	训练/渲染
激活函数（softmax, ReLU, tanh, sigmoid等）	概率生成、函数逼近	位于关键路径	查表单元、执行流水线	训练/渲染
归一化	梯度消失	位于关键路径	通用计算单元、执行流水线	训练/渲染

Element-Wise 操作等算子, 它们计算量不大但访存带宽需求大, 虽然不需要设计专用硬件计算加速单元, 但需要优化访存通路来提高算子执行效率, 例如设计专用片上网络结构或优化访存单元以支持多种数据传输模式.

3.1.2 光线步进和体渲染

光线步进是指光线从起点出发逐步前进, 每到达一步光线就停下来做一次计算, 然后继续前进直到逼近终点, 它常用于隐式表面、点云或体素渲染. 许多神经渲染任务(尤其是 NeRF 类神经渲染任务)都采用光线步进进行编辑, 从而替代光线追踪或光栅化, 表 5 列举了使用光线步进技术的相关工作.

Table 5 Ray Marching in Neural Rendering Application

表 5 神经渲染应用中的光线步进

工作	类型	作用	神经渲染过程
neural volumes[42]	累积-光线步进	渲染体素	渲染
NeRF[3] 及其变种[46-55]	逐光线-光线步进	渲染体素	渲染
common objects in 3D[55]	辐射吸收-光线步进	渲染透明度场	渲染
scene representation networks[56]	可微-光线步进	渲染体素	训练

一般来说神经渲染任务中的光线步进主要分 3 步执行: 首先确认需要生成的光线数量, 然后确定当前光线传播的步进点, 最后根据坐标查询 MLP 网络来计算颜色, 写回像素. 光线步进的核心计算是 GetDist 函数, 这个函数输入空间中某一点的坐标, 返回此点与空间中最近物体的距离, 一般采用 SDF 函数来计算空间中点到球面的距离, 如式(3)所示:

$$\Phi_{\text{circle}}(\boldsymbol{x}) = \|\boldsymbol{x}-\boldsymbol{c}\| - r, \tag{3}$$

其中 $\boldsymbol{x}$ 为空间中一点坐标, $\boldsymbol{c}$ 为球体球心坐标, r 为球体半径. 当函数值大于 0 时表明坐标位于场景内各个物体外, 当函数值小于 0 时表明坐标位于场景内各个物体内, 当函数值等于 0 时表明坐标位于场景内各个物体表面上. 由于式(3)中的 3 维坐标计算从几何上可以转换为向量求模问题, 所以可以将光线步进计算转换为向量乘-平方根-标量加计算, 即光线步进的计算需求为通用计算单元, 无需设计专用硬件加速单元.

神经渲染任务尤其是 NeRF 类任务中需要使用传统体渲染计算, 即通过将 MLP 网络查询到的颜色和密度结果经过体渲染公式计算出采样点的最终颜色, 离散体渲染计算公式如式(4)所示:

$$\hat{C}(r) = \sum_{k=1}^{N} T_k(1-\exp(-\sigma_k(t_{k+1}-t_k)))c_k, \tag{4}$$

其中 $T_k = \exp\left(-\sum_{j=1}^{k-1}\sigma_j(t_{j+1}-t_j)\right)$, 从式(4)中可以得出体渲染计算的需求为通用计算单元和查表单元, 与神经网络计算的硬件加速单元需求一致.

3.1.3 线性插值

除了神经网络、光线步进和体渲染计算之外, 线性插值也是共性需求之一, 表 6 中给出了神经渲染工作中使用插值计算的情况.

Table 6 Interpolation in Neural Rendering Application

表 6 神经渲染应用中的插值

工作	插值类型	数据类型	作用	神经渲染过程
Deepvoxels[41]	3 线性插值	体素、特征	体素、特征插值	渲染
Instant NGP[51]	d 线性插值	体素	哈希编码插值	训练
Deepvoxels[41],neural point-based graphics[30], deep illumination[35]	双线性插值	特征	反卷积插值	渲染
传统后处理	双线性/双 3 次插值	像素	像素插值	渲染

线性插值计算不仅可以应用于神经渲染端到端渲染流水线, 也可以应用于神经渲染流水线+传统后处理流水线中, 线性插值的计算需求为通用计算单元和特殊函数单元, 与光线步进和神经网络中池化层和逐点操作层的硬件加速需求一致. 此外 Instant NGP[51] 引入的多分辨率哈希编码证明可以加速 NeRF 类神经渲染工作的训练过程, 而哈希编码问题可以转换为硬件查表, 这与神经网络中激活函数算子对硬件加速的需求一致.

3.2 需求分析

根据 3.1 节中讨论的共性需求, 本节将神经渲染任务从训练和渲染 2 个过程分别进行讨论.

3.2.1 训练过程

神经渲染前向任务的训练过程和其他基于深度学习应用的训练过程基本相同, 采用反向传播(backward propagation)算法进行训练, 数据类型为单精度浮点或半精度浮点数据, 其步骤一般是: 首先以监督或无监督的方式在大型数据集上预训练模型, 然后通过细粒度调优将预训练模型在测试集上适应特定的下游任务. 神经逆渲染任务的训练过程和基于深度学习应用的训练过程略有不同.

图 7 给出了 NeRF[3], Instant NGP[51], Instant-3D[67] 的训练流程对比, 原始 NeRF 中一个场景的训练通常需要 10 万到 30 万次迭代收敛, 每个场景的批处理规模即光线采样点规模为 192 点/像素乘以 4 096 个像素, 所以每次迭代都需要进行 786 432 次 MLP 查询,

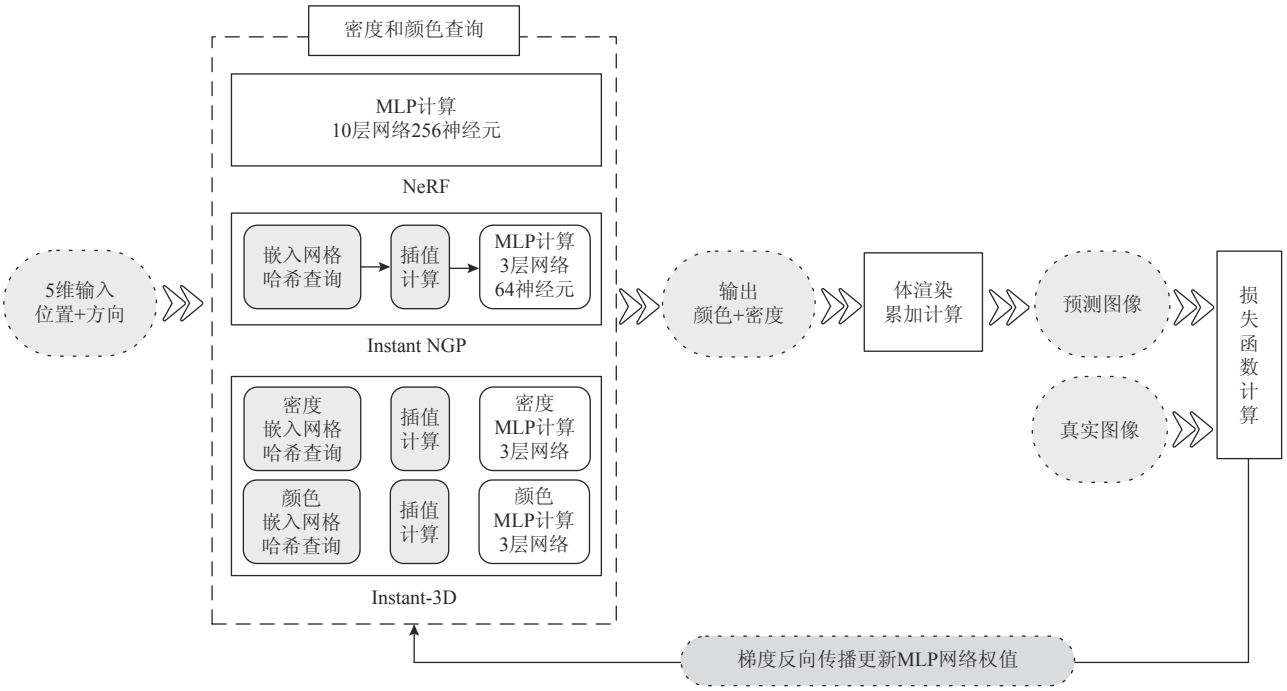


Fig. 7 Training process versus of NeRF, Instant NGP and Instant-3D

图 7 NeRF，Instant NGP 和 Instant-3D 训练过程对比

训练计算量接近 10^{17} 量级，使用单块 NVIDIA V100 GPU 需要大约 1~2 天的训练时长；Instant NGP 通过算法优化在相同的硬件平台上进行 5 s 的训练效果已经接近原始 NeRF 花费数小时的训练效果；Li 等人[67]通过软硬件协同设计来加速 NeRF 训练过程，他们基于 Instant NGP 提出了 Instant-3D 算法和对应的硬件加速结构，通过在不影响重建质量的前提下解耦合嵌入式网格的颜色和密度分支来压缩 NeRF 计算提高训练效率，从而在 AR/VR 设备上实现了质量基本不变的情况下每个场景 1.6 s 的训练时间和 1.9 W 的训练功耗，相比于传统 NeRF 类应用的训练效率提升了 41~248 倍. 后续针对神经渲染训练的工作基本都采用了哈希编码[69-70]作为训练加速手段.

当前针对各类神经网络的训练已经有很多综述性工作，周飞燕等人[71]讨论了 CNN 的监督学习、无监督学习训练方法；王恩东等人[72]讨论了单机训练和分布式训练面临的主要挑战，梳理了各研究分支优化算法和代表性成果；张浩宇等人[73]和殷炯等人[74]分别讨论了 Transformer 网络架构在图像和视频任务训练方面的相关工作，所以本文不再针对神经网络的通用性训练问题进行讨论.

3.2.2 渲染过程

神经渲染任务的渲染过程和其他基于深度学习应用的推理过程近似，数据类型为半精度浮点或整型，其步骤一般是:首先是编码输入，然后是神经网络查询，最后是生成像素输出. 渲染过程主要使用神经网络用于特征提取、投影坐标转换、图像合成等功能. 对于使用 MLP 坐标投影+CNN 生成网络作为渲染框架的算法需要一定的卷积和矩阵算力，例如 RenderNet 使用 1 层 32×32×512 规模的 MLP 网络进行坐标投影，8 层 CNN 网络用于生成 512×512×3 的 RGB 图像. 对于使用 U-Net 作为生成模型的神经渲染算法，推理过程要使用预训练好的 U-Net 进行图像生成，需要一定的卷积和通用计算算力，例如图 8 给出的 Neural Light Transport 渲染生成过程，它使用步长为 2 的卷积进行下采样，然后使用线性插值进行上采样，同时使用 3 次跳跃连接(原始 U-Net 有 4 次跳跃连接)来融合信息从而合成图像. 跳跃连接层也可以认为是残差层的一种形式，它的本质为逐元素(element-wise)运算，此外图 8 中跳跃连接层和拼接层需要保存输入图像以及 2 层中间特征图，这也增加了访存开销.

综上所述神经渲染应用前向渲染过程的硬件加速需求为矩阵运算、卷积运算以及通用计算能力，不同任务具体的计算和访存量是由目标场景中的体素数量和渲染图片的分辨率所决定的.

神经逆渲染应用的渲染过程主要使用神经网络用于输入编码和场景查询以及使用光线步进算法用于光线追踪. 以原始 NeRF 为例，对每条光线选取了 64 点(粗粒度场景)+128 点(细粒度场景)共 192 点的

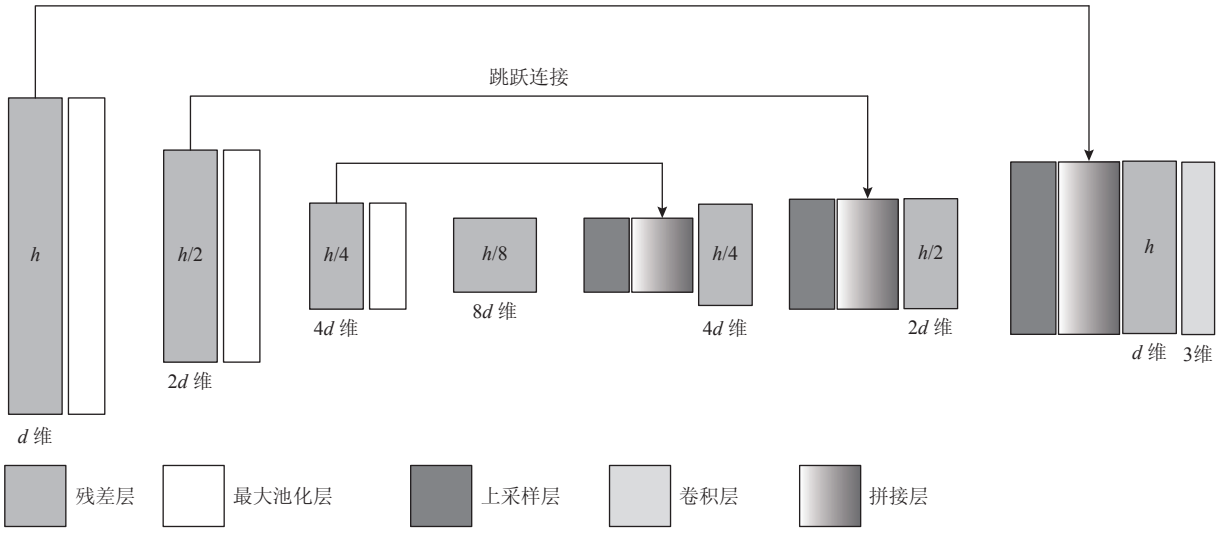


Fig. 8 Network architecture of image synthesis layer

图 8 图像合成层网络架构

光线步进点，对于每个光线步进点的 MLP 网络如图 9 所示. 首先将 3 维坐标向量进行输入编码生成 60 维向量送入 MLP 网络处理，网络包含了 8 层 FC 层(其中第 5 层需要再次输入 60 维向量加强坐标信息)，每层的通道数量为 256，所以得到输出为一个 256 维的特征向量和 σ，然后在 256 维特征向量和相机视角方向输入编码生成的 24 维向量进行连接后送入一个通道数量为 256 加上一个通道数量为 128 的 FC 层计算 RGB 颜色. 为了获取高质量的渲染效果，设定每张图像生成光线数量为 762 000，所以渲染 1 帧图片需要 1.5 亿到 2 亿次光线步进和 MLP 网络计算，在 NVIDIA V100 GPU 上需要 30 s 才能渲染 1 帧图像.

FastNeRF[47] 将 NeRF 中顺序执行的坐标编码 MLP 网络和光线方向 MLP 网络拆分为 2 个并行计算的 MLP 网络(一个产生 d 维神经辐射映射输出的 8 层位置相关 MLP 网络和一个产生 1 维方向特征输出的 4 层方向相关 MLP 网络)实现渲染加速，该文在 NVIDIA RTX 3090 GPU 上的实验结果显示 FastNeRF 比原始 NeRF 的渲染速度快 3 000 倍，但值得一提的是 FastNeRF 的 MLP 网络权值所占用缓存为 54 GB，可以放在 GPU 的片上缓存中从而提升渲染速度. MobileNeRF[75] 把 NeRF 的 MLP 网络拆分为编码器(前 8 层)和解码器(后 2 层)，预先存储好编码器的输出，仅执行解码器的推理计算，从而实现在移动端上实时 NeRF 神经渲染. 但是 MobileNeRF 也存在很多问题，比如对训练数据集要求高(否则会出现大量空洞)、训练时间很长(要分为 3 个步骤进行训练)等.

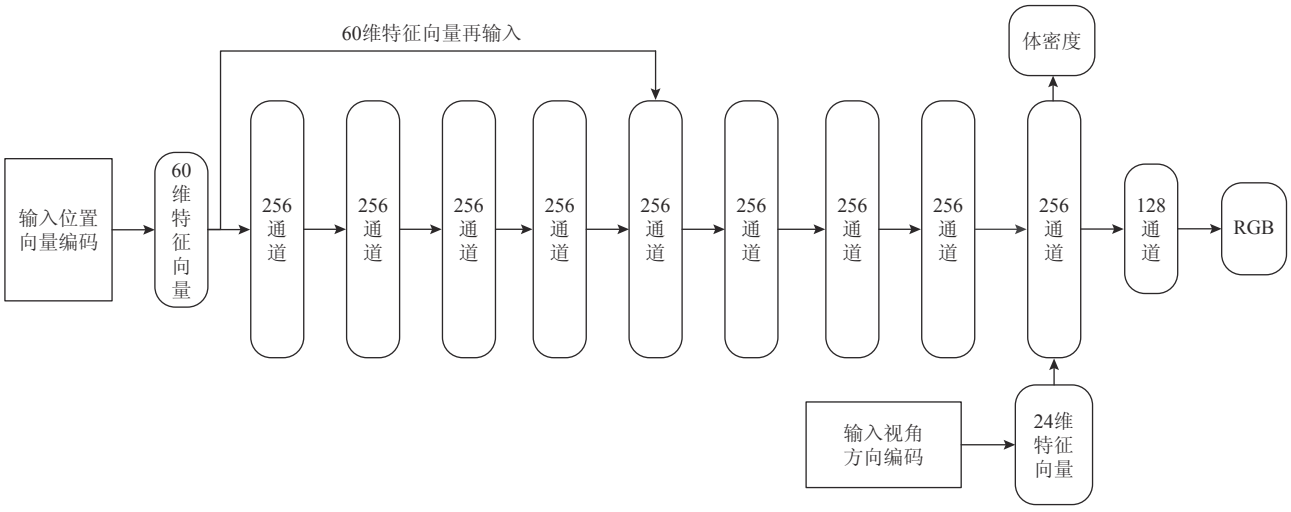


Fig. 9 FC network architecture of NeRF

图 9 NeRF 全连接网络架构

R2L[76]模型用神经光场 NeLT 代替 NeRF 来加快渲染速度，由于模型的输出直接是 RGB 且没有体密度学习和不透明度合成步骤，只需要对光线执行 1 次前向传递来预测像素颜色，这使得 R2L 在渲染方面比 NeRF 快得多. 但是 NeLF 相比与 NeRF 来说更难训练，所以文献 [76] 使用了一个 88 层深度且带残差层的 MLP 架构作为映射函数，这会带来较长的渲染延迟，此外 NeLT 还需要 NeRF 来生成伪数据集协助训练；MobileR2L [77] 基于 R2L 模型提出了一种面向移动端的架构优化，相比于 MobileNeRF 不用预存储中间数据，更适合移动端计算能力受限的硬件平台，但是在处理尺寸较大光线时会导致内存过大的问题.

综上所述，图 10 给出了神经逆渲染任务的流水线，对于每条光线上的每个采样点都要独立地执行该流水线，所以具有天然的高并行特性，可以被硬件加速. 神经渲染应用逆渲染过程的硬件加速需求为输入编码查询、光线步进、矩阵运算、卷积运算以及通用计算能力，相比于前向渲染具有更高的算力和存储需求，神经逆渲染应用所需的光线步进、编码查询、体渲染、神经网络计算量与渲染分辨率以及光线生成数量密切相关，获得高质量的图像会带来巨大的计算和访存开销，而实现更快的输入编码查询和 MLP 网络计算是硬件加速重点关注的对象.

3.2.3 瓶颈分析

从 3.2.1 节可以得到神经前向渲染应用和逆渲染应用的训练过程具有计算量大、访存带宽要求高的特点. 神经前向渲染任务相较与逆渲染应用的计算和访存需求较低，以 deep illumination[35] 为例，对于生成 3 通道 256×256 像素图像的 16 层神经网络，使用单块 NVIDIA P500 GPU 需要 3 h；而神经逆渲染任务以 NeRF[3] 类应用为例，一个场景使用单块 NVIDIA V100 GPU 需要 1~2 天. 所以对于神经渲染任务的训练过程，尤其是逆渲染类应用的训练需要极大的算力资源和存储资源做支撑，神经渲染的训练过程一般是离线完成的，通常在云端数据中心级硬件平台或服务器级硬件平台运行，设备端硬件平台的算力和存储能力难以支撑训练任务.

从 3.2.2 节可以得到对于神经前向渲染应用和逆渲染应用的渲染过程，神经前向渲染应用通常能实现高分辨率的实时渲染，而逆渲染应用则突显出性能瓶颈. Muhammad 等人[78] 在 NVIDIA RTX3090 GPU（35.58 TFLOPS@FP16）上对 NeRF, NSDF, GIA, NVR 四种算法，采用多分辨率哈希编码方式[51] 渲染一帧 1920 × 1 080 大小的图片分别需要 231 ms, 27.87 ms, 2.12 ms, 6.32 ms 的总运行时间，这对于实时渲染来说是不可接受的，其中嵌入网格哈希编码和 MLP 查询的子任务是神经渲染应用中耗时最久的性能瓶颈，二者时长加起来占总时长约 70%；Fu 等人[68] 评估了 NeRF 合成和 Deepvoxels 任务，实验结果表明在 NVIDIA RTX 2080Ti 桌面级 GPU 上分别需要 28 s 和 13 s 完成，而在 NVIDIA Jetson TX2 边缘级 GPU 上分别需要 1 000 s 和 380 s 完成，其中 MLP 查询和光线 Transformer 计算 2 个子任务的运行时间占比为 70%~90%. 所以嵌入网格哈希编码、MLP 查询和光线 Transformer 计算是期望硬件平台重点加速的对象.

3.3 平台现状

常见的图形应用硬件处理平台有以通用处理器、图形处理器和以张量处理器、神经处理器为代表的领域特定架构（domain specific architecture, DSA）[79]，当前这些硬件处理平台都对神经网络进行了支持，也可以适用于神经渲染应用. 本节分别介绍它们在神经网络加速方面的现状以及对神经渲染应用的适配性分析.

3.3.1 通用处理器（CPU）

CPU 作为应用最广泛的硬件处理平台，针对神经网络加速的硬件支持主要集中在 3 个方面：1）增加神经网络扩展指令集；2）增加神经网络专用数据类型；3）增加神经网络专用加速部件. 表 7 给出了不同 CPU 平台对神经网络的加速支持.

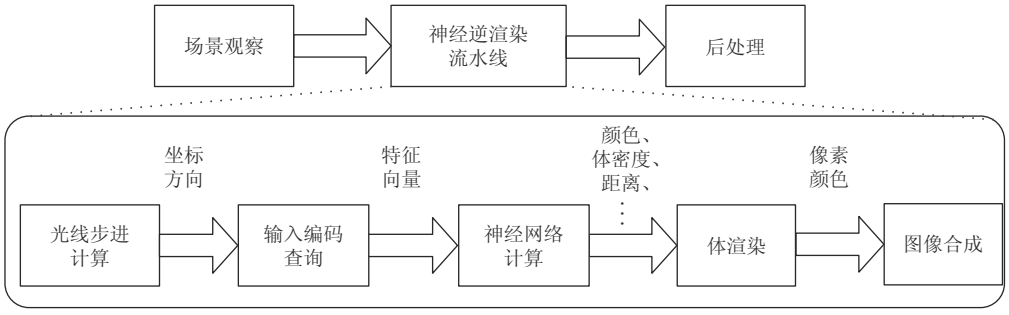


Fig. 10 Pipeline of neural inverse rendering

图 10 神经逆渲染流水线

Table 7 Acceleration Support of CPU for Neural Network

表 7 CPU 对神经网络的加速支持

平台	指令集	数据类型	硬件单元	面向场景
IBM Power10[80]	Power-ISA 3.1 MMA 指令扩展	FP16	MMA 运算单元	推理/训练
Tesla DOJO[81]	64 B 专用指令集	BF16 CFP8	向量/矩阵协处理单元 Scatter/Gather 单元	推理/训练
AMD EPYC	Scatter/Gather 指令 BF16 指令 VNN（向量神经网络）指令	BF16 FP16 INT8	Mul MAC 单元	推理/训练
NVIDIA Grace	BF16 指令 Scatter/Gather 指令	BF16 INT8	MatMul 单元	推理/训练
Intel Xeon 4th Gen	TMUL 指令 VNN 指令	BF16 INT8	AMX 单元	推理/训练

IBM Power10 处理器[80]在最新的 Power-ISA 3.1 版本引入了一组新的矩阵乘法辅助(MMA)指令和配套的 MMA 计算单元. 它直接对小矩阵实现数值线性代数运算, 同时支持半精度数据运算指令. 这些指令可以加速计算密集型内核, 如矩阵乘法、卷积和离散傅里叶变换, 以及包括神经网络在内的人工智能工作负载. 在面对神经网络工作负载时, 增加 MMA 单元的 Power10 处理器相比于 Power9 可以在单核级提高 2.6 倍的能效.

DOJO[81]介绍了特斯拉面向人工智能训练的 DOJO 高通量处理器的微架构, DOJO 处理节点中通用处理器核心的指令宽度为 64 b, 而向量/矩阵协处理器的指令宽度为 64 B, 协处理器指令通过标量调度模块接口发送给协处理器的向量调度模块进行处理; 通用处理器核心的访存通路宽度为 8 B×2, 而向量/矩阵协处理器的访存通路宽度为 64 B×3, 通用处理器核心和协处理器使用 SRAM 空间而不是数据缓存进行数据交互. 所以在 DOJO 中通用处理器核心只负责通用的地址计算和逻辑计算, 而神经网络工作负载计算任务则都由协处理器完成.

AMD 和 Intel 这 2 大厂商分别在其最新的 EPYC 霄龙处理器 Xeon 4th Gen 处理器中增加了用于加速神经网络负载计算的向量神经网络(vector neural network, VNN)指令集和对应的硬件加速单元, 以及用于加速神经网络负载访存的 Scatter/Gather 访存指令扩展. 此外, AMD, Intel, ARM 都在其最新的 CPU 中增加了对 BF16 数据类型的支持.

无论是服务器级还是桌面级/移动端的主流 CPU 平台, 都已对神经网络应用进行了体系结构级的优化加速, CPU 平台增加神经网络计算单元及配套指令集具有 3 个好处: 1)对原有体系结构的其他部分没有影响, 对 CPU 微体系结构的影响最小; 2)CPU 可以通过门控时钟或动态开关方式降低神经网络单元的功耗影响; 3)可以继承原有的编程模式, 并具有一定的通用性.

3.3.2 图形处理器（GPU）

GPU 或通用图形处理器(GPGPU)拥有远超 CPU 的计算资源和存储资源以及强大的浮点计算能力, 它们的架构非常适合处理大规模高并行的计算密集型任务, 自 1998 年 GPU 诞生以来已经经历了快速的发展, 已经从处理图形任务的专用平台发展为涵盖高性能通用计算、图形处理计算和深度学习计算多种领域的硬件处理平台. 当前 GPU/GPGPU 是部署神经网络应用最多的硬件平台, 许多深度学习的框架都基于 GPU 平台接口. 针对神经网络加速的硬件支持主要集中在 4 个方面: 1)适合神经网络计算的芯片体系结构; 2)增加神经网络专用数据类型; 3)增加神经网络专用加速部件; 4)增加神经网络专用访存单元. 表 8 给出了不同 GPU/GPGPU 平台对神经网络的加速支持.

Table 8 Acceleration Support of GPU/GPGPU for Neural Network and Ray Process

表 8 GPU/GPGPU 对神经网络和光线处理的加速支持

平台	峰值算力/FLOPS	数据类型	硬件加速单元	访存加速单元	光线加速单元	面向场景
NVIDIA H100[82]	500T@TF32 1000T@FP16 1000T@BF16 2000T@FP8	TF32 FP16 BF16 FP8	张量核心	张量访存加速单元 TMA		训练/推理/高性能
AMD MI250[83]	383T@FP16 383T@BF16 383@INT8	FP16 BF16	矩阵核心			训练/推理/高性能
Intel Ponte Vecchio XMX[84]	419T@TF32 839T@FP16 839T@BF16 1678@INT8	TF32 FP16 BF16	矩阵引擎	控制流处理预取单元 CSP		训练/推理/高性能
Biren BR100[85]	512T@TF32 1024T@BF16 2048T@INT8	TF32 BF16	向量单元	张量数据存储加速单元 TDA		训练/推理
NVIDIA GeForce RTX 4090[86]	660.6T@FP8 191T@ray	FP8	张量核心		光线追踪单元	推理/图形
AMD RADEON™ RX 6950 XT[87]	47.31T@FP16	FP16	矩阵核心		光线追踪加速器	推理/图形

NVIDIA 公司从 Kepler 架构开始对其 GPU 体系结构进行了大量的改动, 从面向图形处理专用架构向通用处理计算架构转变, 并在 Volta 架构中首次引入了张量核心(tensor core)来加速神经网络任务, 每个张量核心可以完成如图 11 所示的 4×4×4 矩阵乘累加(multiply-accumulate, MAC)混合精度运算, 在其最

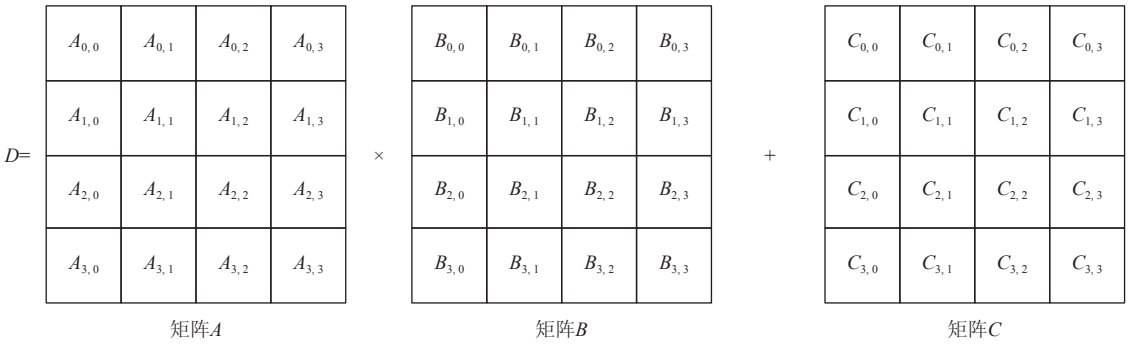


Fig. 11 MAC operation in tensor core

图 11 张量核心乘累加运算

新的 H100 GPU 中，半精度浮点 FP16 的峰值性能已经达到了 1000 TFLOPS. 此外 NVIDIA 公司还陆续在其 GPU 中加入 TF32, BF16, FP8 等神经网络训练/推理的专用数据类型以及加入用于张量数据异步读写加速的张量访存加速单元 TMA, 进一步加速神经网络任务，这些技术变革也引来众多产业界公司和学术界研究人员的跟随. AMD 公司也在其 GPGPU 系列芯片的体系结构中加入了矩阵核心和 FP16/BF16 数据类型来支持神经网络任务加速；Intel 公司在即将发布的 Ponte Vecchio XMX 芯片架构中加入了矩阵引擎和控制流处理预取单元来支持神经网络任务加速.

在传统 GPU 体系结构方面，NVIDIA 公司重点关注神经图形学、元宇宙以及数字人等应用的加速，在其最新的 GeForce RTX 4090 芯片中协同使用光线追踪单元 RTCore、张量核心和通用计算核心来实现基于深度学习的 3 维场景增强以及本文 2.2.3 节中所介绍的 DLSS 技术，AMD 公司也在其高端 GPU 系列芯片中加入了矩阵核心和光线追踪加速器来支持相关领域应用. 除此之外，GPU 中的特殊功能单元 (special-function unit, SFU) 可以加速超越函数（如倒数、平方根、幂函数、对数、三角函数）的计算，由于 SFU 硬件加速本质是采用查表结合 2 次插值的方法，所以也能实现对哈希编码查表算法的支持.

3.3.3 特定领域架构（DSA）

随着深度学习技术的不断发展，上层应用场景对底层硬件算力的需求呈爆炸式增长，Hennessy 等人[79] 提出通过软硬件协同设计的 DSA 作为应对方案. DSA 是一种更加倾向以硬件为中心，针对特定问题领域设计定制的体系结构. DSA 能为该领域提供显著的性能（和能效）增益，它和 CPU 相比更贴近应用程序的需求，在加速特定应用程序时可以实现更好的性能；与此同时 DSA 具有一定的可编程性，所以它和专用集成电路（application specific integrated circuit, ASIC）相比具有更好的灵活性和更广的功能覆盖范围.

神经处理单元 NPU 或张量处理单元 TPU 是面向神经网络的 DSA 架构，当前已经出现了许多 NPU/TPU 硬件平台，Reuther 等人[88] 收集和总结了目前已公开公布具有峰值性能和功耗数字的商用神经网络加速器，表 9 列举了部分 NPU/TPU 平台对神经网络的加速支持.

Table 9 Acceleration Support of NPU/TPU for Neural Network

表 9 NPU/TPU 对神经网络的加速支持

平台	支持网络类型	数据类型	加速结构	面向场景
TPU v4[89]	MLP CNN RNN Transformer	BF16 INT8	脉动阵列	训练/推理
TPUv4i[90]	MLP CNN RNN Transformer	BF16 INT8	脉动阵列	推理
寒武纪 MLU370[91]	MLP CNN RNN Transformer	FP32 FP16 BF16 INT16/8/4	点积树	训练/推理
百度昆仑芯 2 代[92]	CNN Transformer GEMM	FP16 INT8	乘累加阵列	训练/推理
ARM Ethos-U55[93]	CNN RNN	INT8	乘累加单元	推理

NPU/TPU 是针对神经网络的特定架构，它们主要面向深度神经网络和卷积神经网络，支持神经网络计算所需的有限算子和数据类型. 面向训练的 NPU/TPU 芯片通常具有较高的计算能力和丰富的访存资源，可以高效支持张量计算加速；面向推理的 NPU/TPU 芯片通常计算能力和访存带宽受限，只有少量用于激活函数处理等非线性操作，其他计算能力相对较为欠缺.

神经图形处理单元 NePU 是面向 NeRF 类任务的 DSA 架构，当前仍处于起步研究阶段. ICARUS[94] 是面向 NeRF 渲染的特定架构. 图 12 给出了 ICARUS

的总体架构, ICARUS 和 CPU、内存一起组成渲染系统, 该系统执行 NeRF 渲染过程的步骤为: 1)将网络模型、编码频率和位置方向等输入存储在内存中, 在运行时由 CPU 控制输入给 ICARUS; 2)将输入数据通过片上总线分发到对应的目标全光核心(plenoptic core); 3)网络模型、位置和方向数据被加载到 ICARUS 中进行 NeRF 处理(从位置编码单元 PEU 到 MLP 引擎再到体渲染单元 VRU), 1 组光线(包含这条光线所有的步进点)由同一个全光核心处理; 4)通过数据总线流出最终渲染的像素颜色. 文献 [94] 通过将 MLP 计算的 FC 层的操作转换为近似可重构多重常数乘法(reconfigurable multiple constant multiplications, RMCMs) 来提高硬件加速效率, 因此这相比于传统的乘累加计算降低了约 1/3 的硬件复杂性(包括计算量和参数容量), 这使得整个 NeRF 流水线可以在全光核心内部完成, 所有权值参数都放在全光核心内的 SRAM 中不需要和 DRAM 交换数据. ICARUS 在 40 nm 工艺节点下的芯片面积为 16.5 mm^2, 工作频率达到 400 MHz, 渲染一帧 800×800 分辨率的图像(光线步进采样点数为 192)需要 45.75 s, 功耗为 282.8 mW. ICARUS 架构取得了一定的创新, 但距离 NeRF 实时渲染任务还具有较大的差距.

神经图形处理簇 NGPC[78] 在传统 GPU 架构中加入神经场处理器(neural fields processor, NFP)来加速神经图形应用. 图 13 中给出了 NGPC 的总体架构, 将包含 NFP 的 NGPC 作为和图形处理簇(GPC)并列的硬件单元嵌入在 GPU 架构中实现片内异构计算, NFP 由编码引擎和 MLP 引擎组成, 分别加速输入编码和 MLP 计算, 而对于哈希编码查询计算需要 GPC 单元进行处理. 经过实验得出包含 8 个、16 个和 32 个 NFP 单元的 NGPC, 在 NeRF, NSDF, GIA, NVR 这 4 种应用下相比于 RTX3090 平台执行可以分别获得 12.94 倍、20.85 倍、33.73 倍的性能提升. 该文作者指出 NGPC 可以实现 NeRF 应用 30 FPS 的 4 K 像素超高清分辨率渲染(但没有明确提到 NFP 的数量), 在 7 nm 工艺下的估算结果为包含 1 个 NFP 单元的 NGPC-8 GPU 的面积开销增加了约 4.52%, 功耗开销增加了约 2.75%. NGPC 提供了一种基于 GPU 架构实现神经渲染异构加速的思路, 好处是可以利用 GPU 原有的通用计算资源, 如光线步进、超越函数等, 缺点是会进一步增加硬件开销, 并且针对神经渲染专用场景, 可能会造成较为严重的暗硅现象.

Li 等人[67] 设计了 Instant-3D 加速器用于 NeRF 类的 3 维重建神经渲染任务. Instant-3D 加速器如图 14 所示, 它主要由 MLP 单元、网格核心和 I/O 接口 3 部分组成, 该文作者将 Instant-3D 算法中的嵌入网格哈希查询和 MLP 查询 2 个步骤映射到 Instant-3D 加速器上进行硬件流水加速处理, 同时通过分析算法的访存行为设计了一个前向传播读映射器来 FRM 合并多个访存读请求和一个反向传播更新合并器 BUM 将多个网格更新合并为一个更新, 从而提高片上 SRAM 阵列的利用率, 最小化 SRAM 的写次数和功

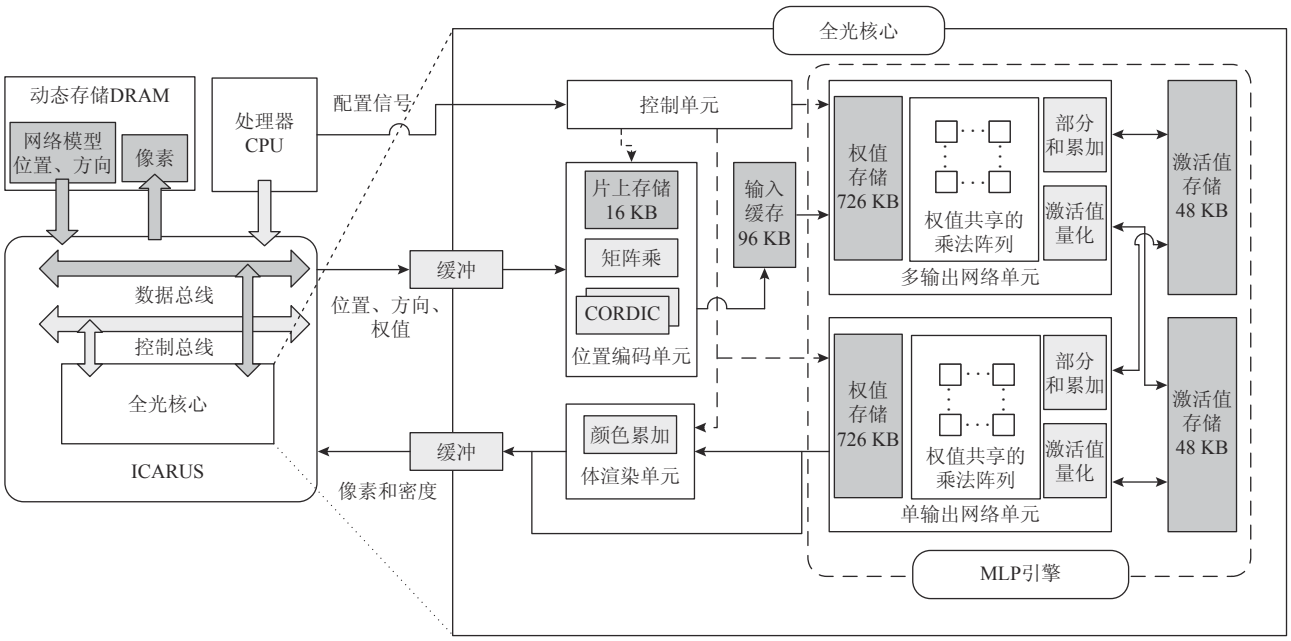


Fig. 12 Overall architecture of ICARUS

图 12 ICARUS 总体架构

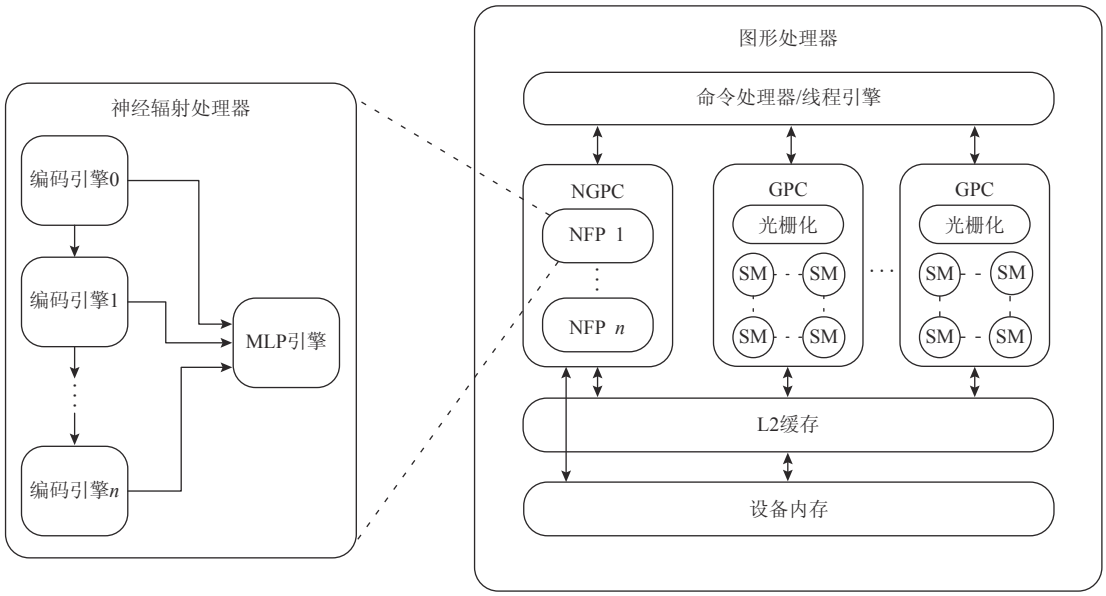


Fig. 13 Overall architecture of NGPC

图 13 NGPC 总体架构

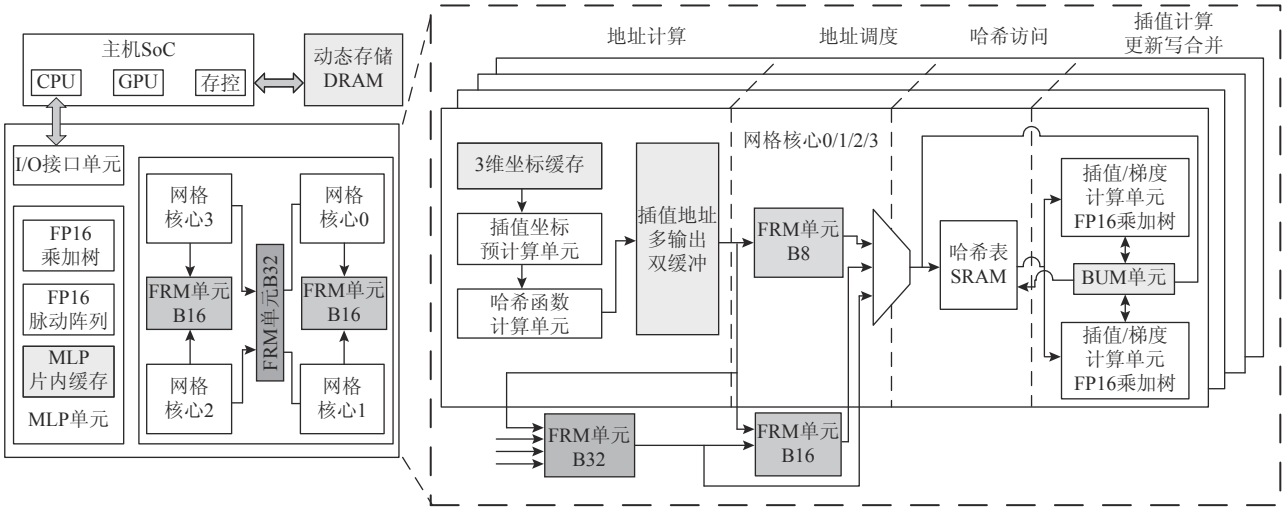


Fig. 14 Overall architecture of Instant-3D accelerator

图 14 Instant-3D 加速器总体架构

耗以及支持 Instant-3D 算法所需的不同网格大小. Fu 等人[68] 设计了 Gen-NeRF 加速器用于实时可泛化的 NeRF 类任务. Gen-NeRF 加速器的总体架构如图 15 所示，它主要由软件定义的渲染引擎、工作负载调度器和片上存储组成. 利用对极几何的推论来加速目标工作负载和设计渲染引擎架构，从而优化软件到硬件的数据流映射，能最大限度地提高光线之间的数据重用.

3.4 设计挑战

综上所述，当前硬件平台在处理神经渲染应用时仍存在一定设计挑战：

1）主流 CPU 平台虽然已针对神经网络所需的卷积、矩阵-矩阵乘等算子进行了针对性的设计和优化，但是对于神经渲染任务，CPU 平台受到神经网络算力和通用算力的相对限制更适合用于神经前向渲染任务或逆渲染中推理部分任务；此外由于快速查表（激活函数、哈希编码）能力的欠缺，CPU 平台完成端到端的高清实时神经逆渲染任务具有很大的挑战.

2）当前如 AMD 和 NVIDIA 等主流厂商将旗下 GPU 芯片分为 GPGPU 和传统 GPU 平台 2 条分支路线，分别面向神经网络/高性能应用和传统图形应用. GPGPU 平台通常具备超高神经网络算力、较高的访存带宽和片上存储容量，但删减了传统 GPU 中光栅化、渲染输出和光线追踪等单元，GPGPU 平台可以

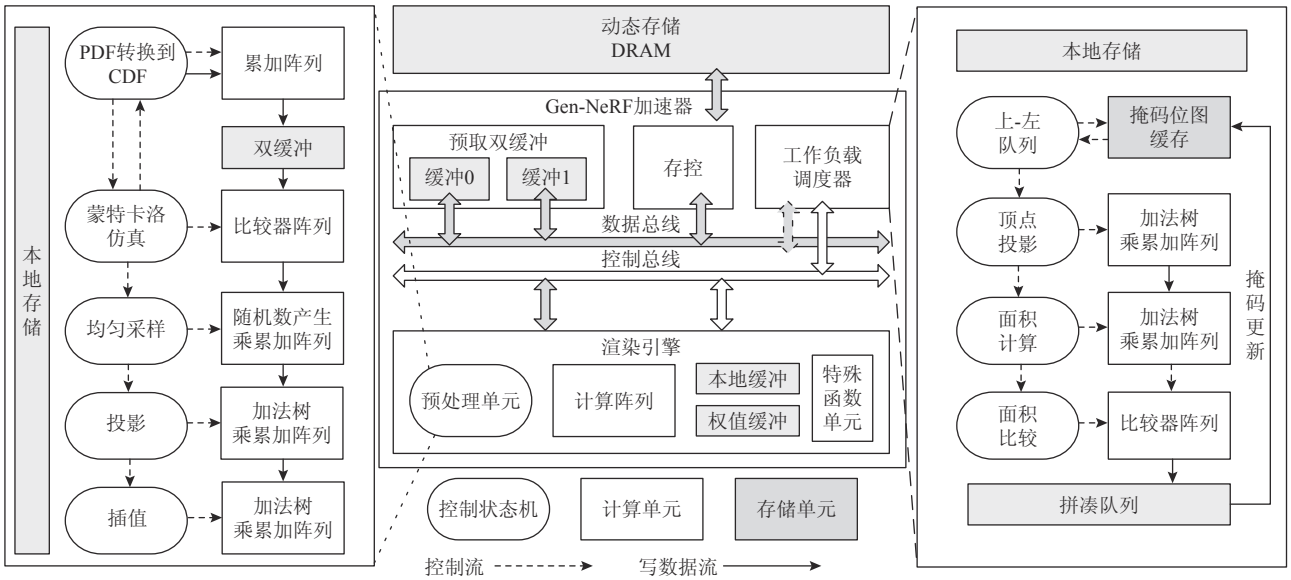


Fig. 15 Overall architecture of Gen-NeRF accelerator

图 15 Gen-NeRF 加速器总体架构

较好地加速神经网络计算和体渲染计算，能较好地匹配神经渲染任务的训练需求，但对于光线步进、哈希编码查表等专用算力的支持稍有欠缺；而传统 GPU 平台在传统光栅化、光线追踪单元的基础上增加了可观的神经网络算力和较高的访存性能，能较好地适配神经前向渲染任务需求，但是无法满足神经逆渲染任务的加速需求. 此外这 2 条产品分支的发展路线也增加了厂商的研发成本和系统搭建的复杂性.

3）对于 DSA 平台，NPU/TPU 是针对神经网络的特定架构，它们主要面向深度神经网络和卷积神经网络，大部分硬件只支持神经网络计算所需的算子和数据类型，而片内的计算资源主要用于张量计算加速，有少量用于激活函数处理等非线性操作，通用计算能力相对较为欠缺. 所以对于神经渲染应用来说，由于对光线步进、哈希编码、体渲染计算等算力的欠缺以及所支持算子与数据类型的限制，这使得 NPU/TPU 硬件平台可能只能支持少数神经渲染任务. 而几种 NePU 架构的研究主要面向 NeRF 类神经渲染任务且取得了一定程度的架构突破，但它们普遍作为 CPU 和 GPU 的硬件补充加速平台实现，具有一定的场景局限性，例如 ICARUS[94] 和 Instant-3D 加速器[67] 需要配合主机 SoC 使用，NGPC[78] 则增加了 GPU 内任务调度和数据流映射的复杂性.

4 研究发展趋势

本节主要从神经渲染应用和硬件加速架构 2 个方面对神经渲染系统的研究发展方向进行展望.

4.1 神经渲染应用

经过几年的发展，神经渲染在场景表示、实时全局光照、新视图合成、重光照以及空间照明方面已经显示出令人印象深刻的能力，但同时也存在泛化性、可扩展性、多模态合成等挑战，神经渲染应用对 AR/VR 领域的发展起到了关键的作用. 对于传统图形渲染流水线中的各个子模块，例如曲面细分或光栅化，相信完全可以由神经渲染来取代，实现端到端的神经前向渲染流水线，开发基于神经渲染的图形应用程序. 对于逆渲染应用，当前神经渲染的大部分工作仅关注简单物体和相对简单的复合场景，基于单物体开发的场景编辑、重光照等方法不能扩展到大规模场景[22]、复杂环境场景[95] 或运动场景[96] 的神经表示仍然面临挑战；对于新视图合成任务目前仍然依赖于大规模多视图数据集或只能针对某个特殊物体来进行训练，提升跨场景或少量视图下的神经体素表示的通用性也是重要挑战；对于神经渲染的多模态学习应用，意味着要同时处理语义、文本、声音和视觉信号，如何渲染动态交互以及声貌匹配的数字人是极大的挑战；对于神经渲染的消费级需求，如何划分应用在云端和移动端的功能及交互模式，实现轻量化和低成本部署是主要的挑战.

4.2 神经渲染处理器

综上所述，由于神经渲染任务的高可并行性和特殊性，神经渲染任务与传统图形任务和神经网络任务的差异性，以及当前硬件平台处理神经渲染任

务的问题和不足，所以研究神经渲染处理器（neural rendering processing unit, NRPU）来加速神经渲染应用是重要发展趋势.

4.2.1 协同设计

软硬件协同设计是体系结构设计的核心方法，如何将系统任务进行软硬件功能划分来简化硬件实现，如何针对软件算法瓶颈设计硬件加速单元来提升应用性能，以及如何探索找到二者之间的平衡点，是设计人员所追求的目标. 以 TPU 为代表的神经网络系统已经给我们提供了许多软硬件协同设计的启发和经验教训. 对于未来神经渲染系统的软硬件协同设计，需要注意 4 个问题：

1）设计目标为神经渲染应用、基于神经渲染流水线的图形编辑全栈工具链和神经渲染处理器. 从而实现现有图形处理器和图形编辑工具的革新，为虚拟现实和增强现实、影视创作和数字娱乐、人工智能和元宇宙等未来重点发展领域提供有效支持.

2）优化方法应该由部署场景决定. 例如 Instant-3D[67] 和 GEN-NeRF[68] 都是基于 AR/VR 设备的 NeRF 类应用的协同设计，由于 AR/VR 设备是计算、访存和功耗受限的平台，所以它们的协同设计目标是在尽量保证算法精度或在可接受范围内精度降低的前提下减少算法和硬件的计算或存储开销.

3）进行合理的子功能软硬件划分. 例如将 NeRF 类神经渲染任务中哈希查表计算、MLP 计算和体渲染计算等子功能由硬件进行加速，将 RenderNet 类神经渲染任务中神经网络计算或投影计算等子功能由硬件进行加速；而对于坐标预处理、位置编码和光线采样点判断等子功能由软件进行处理.

4）硬件架构设计应充分考虑算法成熟度和硬件设计周期，保留一定程度的设计余量与灵活性. Jouppi 等人[90] 在总结 TPU 设计 10 条最重要的经验教训时提出由于深度神经网络的内存和计算每年增长 1.5 倍，所以架构师应该提供动态余量以便 DSA 在整个生命周期内都能发挥作用，当前 Transformer 类应用迅速取代大量的 LSTM 也显示了 DSA 设计中灵活性的重要性.

4.2.2 场景划分

针对未来神经渲染系统可能的部署场景，我们将神经渲染处理器设计按云端和设备端分别进行讨论：

1）训渲一体 NRPU 是面向云端场景，部署在数据中心的神经渲染处理器（加速卡），神经渲染应用在云端完成对应神经渲染网络训练和渲染或完成渲染中的重计算负载任务，它可以独立完成渲染任务或将推流以及特定编码文件通过网络传输到端侧设备平台完成最终渲染.

2）渲染专用 NRPU 是面向设备端场景，部署在移动设备或 AR/VR 设备中的神经渲染处理器（搭建片上系统），神经渲染应用在端侧完成图像渲染，渲染专用 NRPU 可能接收来自云端的推流或特定编码文件，经过专用硬件加速合成图像并渲染输出.

4.2.3 架构设计

训渲一体 NRPU 需要和 GPGPU/高端 NPU 一样具备强大的神经网络计算能力和访存性能，能够有效地支持神经渲染网络的训练；同时它还具备强大的例如超越函数处理、哈希编码查表处理、光线步进处理等算子的神经渲染专用计算能力. 如何设计编程模型和体系结构以满足神经渲染应用的训练和渲染数据流，如何提高神经渲染硬件加速单元的并行度同时设计性能匹配的访存和数据传输通路以及片上存储空间，在晶体管容量和工艺的约束下将是巨大的挑战.

渲染专用 NRPU 则需要根据具体神经渲染算法确定硬件所需要支持的算子单元、峰值计算性能、存储容量和带宽，同时对 NRPU 和片上系统内其他硬件单元（如 CPU、ISP、编解码单元等）竞争访存带宽和网络传输带宽进行数据流量和性能分析. 如何将神经渲染流水线高效映射到片上系统中提高片内数据重用性以降低访存带宽需求；如何在保证高质量实时神经渲染的前提下降低硬件运行功耗和设计成本，这在功耗、面积和价格限制的设备端平台是巨大的挑战.

5 总　　结

随着深度神经网络的迅速兴起和发展，神经渲染作为深度学习和计算机图形学的融合方法应运而生. 神经渲染是一种基于深度学习的图像和视频生成方法，它将深度神经网络模型和计算机图形学的物理知识相结合，从而获得可控和逼真的场景模型，实现对例如光照、相机参数、姿态等场景属性的控制. 神经渲染已经在过去几年中取得了快速进展，它的应用范围从利用深度神经网络增强前向渲染效果到逆渲染中的新视图合成、形状和材料编辑、重照明和人化身生成等. 本文介绍了神经渲染在前向渲染、逆渲染和后处理应用中具有代表性的研究成果，详细分析了神经渲染应用对硬件加速的共性需求，最后按场景讨论了神经渲染处理器架构的设计挑战.

作者贡献声明: 严忻恺查阅总结资料及撰写论文;霍宇驰在算法部分提出指导意见并修改论文;鲍虎军提出总体指导意见并修改论文.

参 考 文 献

[1] Eslami S, Rezende D, Besse F, et al. Neural scene representation and rendering[J]. Science 2018, 360(6394): 1204–1210

[2] Tewari A, Fried O, Thies J, et al. State of the art on neural rendering[J]. Journal of the European Association for Computer Graphics, 2020, 39(2): 701−727

[3] Mildenhall B, Srinivasan P, Tancik M, et al. NeRF: Representing scenes as neural radiance fields for view synthesis[J]. Communications of the ACM, 2022, 65(1): 99−106

[4] Tewari A, Thies J, Mildenhall B, et al. Advances in neural rendering[J]. Journal of the European Association for Computer Graphics, 2022, 41(2): 41−74

[5] Wang Qi, Zhong Zhihua, Huo Yuchi, et al. State of the art on deep learning-enhanced rendering methods[J]. Machine Intelligence Research, 2023, 20(6): 799−821

[6] Kajiya J. The rendering equation[C]//Proc of the 13th Annual Conf on Computer Graphics and Interactive Techniques. New York: ACM, 1986, 20(4): 143–150

[7] NVIDIA. RTX technology [EB/OL]. 2023 [2023-04-22]. https://developer.nvidia.com/rtx/ray-tracing

[8] Trina W. Truly global illumination: Ray tracing for the masses [EB/OL]. 2023[2023-04-22].https://blog.imaginationtech.com/truly-global-illumination-ray-tracing-for-the-masses

[9] Hornik K, Stinchcombe M, White H. et al. Multilayer feedforward networks are universal approximators[J]. Neural Networks, 1989, 2(5): 359−366

[10] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[C]//Proc of the 27th Neural Information Processing Systems. New York: ACM, 2014: 2672−2680

[11] Zhang R, Isola P, Efros A , et al. The unreasonable effectiveness of deep features as a perceptual metric[C]//Proc of the 36th IEEE Conf on Computer Vision & Pattern Recognition. Piscataway, NJ: IEEE, 2018: 586−595

[12] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. arXiv preprint, arXiv: 1706.03762, 2017

[13] Curless B, Levoy M. A volumetric method for building complex models from range images[C]//Proc of the 23rd SIGGRAPH. New York: ACM, 1996: 303−312

[14] Greger G, Shirley P, Hubbard P, et al. The irradiance volume[J]. IEEE Computer Graphics and Applications, 1998, 18(2): 32−43

[15] Aliev K, Ulyanov D, Lempitsky V, et al. Neural point-based graphics[J]. arXiv preprint, arXiv: 1906.08240, 2019

[16] Wiles O, Gkioxari G, Szeliski R, et al. Synsin: End-to-end view synthesis from a single image[C]//Proc of the 38th IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2020: 7465−7475

[17] Lassner C, Zollhofer M. Pulsar: Efficient sphere-based neural rendering[C]//Proc of the 39th IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2021: 1440−1449

[18] Neff T, Stadlbauer P, Parger M, et al. Point-based neural rendering with per-view optimization[J]. Journal of the European Association for Computer Graphics, 2021, 40(4): 40−54

[19] Park J, Florence P, Straub J, et al. DeepSDF: Learning continuous signed distance functions for shape representation[C]//Proc of the 37th IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2019: 165−174

[20] Chen Zhiqin, Zhang Hao. Learning implicit fields for generative shape modeling[C]//Proc of the 37th IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2019: 5932−5941

[21] Zhu Jingsen, Huo Yuchi, Qi Ye, et al. I2-SDF: Intrinsic indoor scene reconstruction and editing via raytracing in neural SDFs[C]//Proc of the 41st IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2023: 12489−12498

[22] Martin-Brualla R, Radwan N, Sajjadi M , et al. NeRF in the wild: Neural radiance fields for unconstrained photo collections[C]//Proc of the 39th IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2021: 7206−7215

[23] Niemeyer M, Geiger A. GIRAFFE: Representing scenes as compositional generative neural feature fields[C]//Proc of the 39th IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2021: 11448−11459

[24] Pumarola A, Corona E, Pons-Moll G, et al. D-NeRF: Neural radiance fields for dynamic scenes[C]//Proc of the 39th IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2021: 10313−10322

[25] Srinivasan P, Deng Boyang, Zhang Xiuming, et al. NeRV: Neural reflectance and visibility fields for relighting and view synthesis[C]// Proc of the 39th IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2021: 7491−7500

[26] Chen Jianchuan, Yi Wentao, Ma Liqian, et al. GM-NeRF: Learning generalizable model-based neural radiance fields from multi-view images[C]//Proc of the 41st IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2023: 20648−20658

[27] Bao Chong, Zhang Yinda, Yang Bangbang, et al. SINE: Semantic-driven image-based NeRF editing with prior-guided editing field[C] // Proc of the 41st IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2023: 20919−20929

[28] Nguyen-Phuoc T, Li Chuan, Balaban S, et al. RenderNet: A deep convolutional network for differentiable rendering from 3D shapes[C]//Proc of the 36th Neural Information Processing Systems. New York: ACM, 2018: 7902−7912

[29] Rematas K, Ferrari V. Neural voxel renderer: Learning an accurate and controllable rendering tool[C]//Proc of the 38th IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2020: 5416−5426

[30] Aliev K, Ulyanov D, Lempitsky V. Neural point-based graphics[J]. arXiv preprint, arXiv: 1906.08240, 2019

[31] Dai Peng, Zhang Yinda, Li Zhuwen, et al. Neural point cloud rendering via multi-plane projection[C]//Proc of the 37th IEEE Conf

on Computer Vision and Pattern Recognition, Piscataway, NJ: IEEE, 2019: 7830–7839

[32] Sanzenbacher P, Mescheder L, Geiger A. Learning neural light transport[J]. arXiv preprint, arXiv: 2006.03427, 2020

[33] Niemeyer M, Mescheder L, Oechsle M, et al. Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision[C]//Proc of the 38th IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2020: 3501−3512

[34] Yariv L, Gu Jiatao, Kasten Y, et al. Volume rendering of neural implicit surfaces[J]. arXiv preprint, arXiv: 2106.12052, 2021

[35] Thomas M, Forbes A. Deep illumination: Approximating dynamic global illumination with generative adversarial network[J]. arXiv preprint, arXiv: 1710.09834, 2017

[36] T Müller, Rousselle F, J Novák, et al. Neural control variates[J]. ACM Transactions on Graphics, 2020, 39(6): 1−19

[37] Suppan C, Chalmers A, Zhao Junhong, et al. Neural screen space rendering of direct illumination[J/OL]. Journal of the European Association for Computer Graphics, 2021 [2023-04-29].https://diglib.eg.org/bitstream/handle/10.2312/pg20211385/037-042.pdf

[38] Nalbach O, Arabadzhiyska E, Mehta D, et al. Deep shading: Convolutional neural networks for screen space shading[J]. Journal of the Computer Graphics Forum, 2017, 36(4): 65−78

[39] Ren Lei, Song Ying. AOGAN: A generative adversarial network for screen space ambient occlusion[J]. Computational Visual Media, 2022, 8(3): 483−494

[40] Oechsle M, Mescheder L, Niemeyer M, et al. Texture fields: Learning texture representations in function space[C] //Proc of the 37th IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2019: 4530−4539

[41] Sitzmann V, Thies J, Heide F, et al. Deepvoxels: Learning persistent 3D feature embeddings[C]//Proc of the 37th IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2019: 2432−2441

[42] Lehrmann A, Schwartz G, Saragih J, et al. Neural volumes: Learning dynamic renderable volumes from images[J]. ACM Transactions on Graphics, 2019, 38(4): 1−14

[43] Lindell D, Martel J, Wetzstein G. AutoInt: Automatic integration for fast neural volume rendering[C]//Proc of the 39th IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2021: 14551−14560

[44] Liu Lingjie, Gu Jiatao, Lin K, et al. Neural sparse voxel fields[C]//Proc of the 38th Neural Information Processing Systems. New York: ACM, 2020: 15651−15663

[45] Reiser C, Peng Songyou, Liao Yiyi, et al. KiloNeRF: Speeding up neural radiance fields with thousands of tiny MLPs[C]//Proc of the 18th IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2021: 14315−14325

[46] Hedman P, Srinivasan P, Mildenhall B, et al. Baking neural radiance fields for real-time view synthesis[C]//Proc of the 18th IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2021: 5855−5864

[47] Garbin S, Kowalski M, Johnson M, et al. FastNeRF: High-fidelity neural rendering at 200FPS[C]//Proc of the 18th IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2021: 14326−14335

[48] Johari M, Lepoittevin Y, Fleuret F. GeoNeRF: Generalizing NeRF with geometry prior[C]//Proc of the 40th IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2022: 18344−18347

[49] Yu A, Li Ruilong, Tancik M, et al. PlenOctrees for real-time rendering of neural radiance fields[C]//Proc of the 18th IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2021: 5732−5741

[50] Sitzmann V, Rezchikov S, Freeman W, et al. Light field networks: Neural scene representations with single-evaluation rendering[J]. arXiv preprint, arXiv: 2106.02634, 2021

[51] Müller T, Evans A, Schied C, et al. Instant neural graphics primitives with a multiresolution Hash encoding[J]. ACM Transactions on Graphics, 2022, 41(4): 1−15

[52] Li Zhenqi, Niklaus S, Snavely N, et al. Neural scene flow fields for space-time view synthesis of dynamic scenes[C]//Proc of the 39th IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2021: 6498−6508

[53] Yu A, Ye V, Tancik M, et al. pixelNeRF: Neural radiance fields from one or few images[C]//Proc of the 39th IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2021: 4576−4585

[54] Wang Qianqian, Wang Zhicheng, Genova K, et al. IBRNet: Learning multi-view image-based rendering[C]//Proc of the 39th IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2021: 4688−4697

[55] Reizenstein J, Shapovalov R, Henzler P, et al. Common objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction[C]//Proc of the 18th IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2021: 10881−10891

[56] Sitzmann V, Zollhfer M, Wetzstein G. Scene representation networks: Continuous 3D-structure-aware neural scene representations[J]. arXiv preprint, arXiv: 1906.01618, 2019

[57] Bi Sai, Xu Zexiang, Srinivasan P, et al. Neural reflectance fields for appearance acquisition[J]. arXiv preprint, arXiv: 2008.03824, 2020

[58] Zhang Xiuming, Fanello S, Tsai Y, et al. Neural light transport for relighting and view synthesis[J]. arXiv preprint, arXiv: 2008.03806, 2020

[59] Sun Tiancheng, Lin K , Bi Sai, et al. NeLF: Neural light-transport field for portrait view synthesis and relighting[J]. arXiv preprint, arXiv: 2107.12351, 2021

[60] Meijering E, Zuiderveld K. Image reconstruction by convolution with symmetrical piecewise *n*th-order polynomial kernels[J]. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 1999, 8((2):): 192−201

[61] Keys R. Cubic convolution interpolation for digital image processing[J]. IEEE Transactions on Acoustics, Speech, and Signal Processing, 2003, 29(6): 1153−1160

[62] NVIDIA. Nvidia DLSS 3[EB/OL]. 2023 [2023-03-25].https://www.nvidia.cn/geforce/news/dlss3-ai-powered-neural-graphics-innovations

[63] Zhang Guozhen, Zhu Yuhan, Wang Haonan, et al. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation[C]//Proc of the 41st IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2023: 5682−5692

[64] Bi Sai, Sunkavalli K, Perazzi F, et al. Deep cg2real: Synthetic-to-real translation via image disentanglement[C]//Proc of the 17th IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2019: 2730−2739

[65] Zhu Junyan, Park T, Isola P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks[C]//Proc of the 16th IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2017: 2242−2251

[66] Hadadan S, Chen S, Zwicker M. Neural radiosity[J]. ACM Transactions on Graphics, 2021, 40(6): 1−11

[67] Li Sixu, Li Chaojian, Zhu Wenbo, et al. Instant-3D: Instant neural radiance field training towards on-device AR/VR 3D reconstruction[J]. arXiv preprint, arXiv: 2304.12467, 2023

[68] Fu Yonggan, Ye Zhifan, Yuan Jiayi, et al. GEN-NeRF: Efficient and generalizable neural radiance fields via algorithm-hardware co-design[J]. arXiv preprint, arXiv: 2304.11842, 2023

[69] Wang Peng, Liu Yuan, Chen Zhaoxi, et al. F^2-NeRF: Fast neural radiance field training with free camera trajectories[J]. arXiv preprint, arXiv: 2303.15951, 2023

[70] Bai Jiayang, Huang Letian, Gong Wen, et al. Self-NeRF: A self-training pipeline for few-shot neural radiance fields[J]. arXiv preprint, arXiv: 2303.05775, 2023

[71] Zhou Feiyan, Jin Linpeng, Dong Jun. Review of convolutional neural network[J]. Chinese Journal of Computers, 2017, 40(6): 1229−1251 (in Chinese)
(周飞燕, 金林鹏, 董军. 卷积神经网络研究综述[J]. 计算机学报, 2017, 40(6): 1229−1251)

[72] Wang Endong, Yan Ruidong, Guo Zhenhua, et al. A survey of distributed training system and its optimization algorithms[J]. Chinese Journal of Computers, 2024, 47(1): 1−28 (in Chinese)
(王恩东, 闫瑞栋, 郭振华, 等. 分布式训练系统及其优化算法综述[J]. 计算机学报, 2014, 47(1): 1−28)

[73] Zhang Haoyu, Wang Tianbao, Li Mengze, et al. Comprehensive review of visual-language-oriented multimodal pre-training methods[J]. Journal of Image and Graphics, 2022, 27(9): 2652−2682 (in Chinese)
(张浩宇, 王天保, 李孟择, 等. 视觉语言多模态预训练综述[J]. 中国图象图形学报, 2022, 27(9): 2652−2682)

[74] Yin Jiong, Zhang Zedong, Gao Yuhan, et al. Survey on visual language pre-training[J]. Journal of Software, 2023, 34(5): 2000−2023 (in Chinese)
(殷炯, 张哲东, 高宇涵, 等. 视觉语言预训练综述[J]. 软件学报, 2023, 34(5): 2000−2023)

[75] Chen Zhiqin, Thomas F, Peter H, et al. MobileNeRF: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures[J]. arXiv preprint, arXiv: 2208.00277, 2023

[76] Wang Huan, Ren Jian, Huang Zeng , et al. R2L: Distilling neural radiance field to neural light field for efficient novel view synthesis[J]. arXiv preprint, arXiv: 2203.17261, 2022

[77] Cao Junli, Wang Huan, Pavlo C, et al. Real-time neural light field on mobile devices[C]//Proc of the 19th IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2023: 8328−8337

[78] Muhammad H, Ramakrishna K, Tobias Z, et al. Hardware acceleration of neural graphics[C/OL]//Proc of the 50th Int Symp on Computer Architecture. New York: ACM, 2023 [2023-04-28].https://dl.acm.org/doi/10.1145/3579371.3589085

[79] Hennessy J, Patterson D. A new golden age for computer architecture[J]. Communications of the ACM, 2019, 62(2): 48−60

[80] Brian W, Dung Q, Moreira E, et al. Energy efficiency boost in the AI-infused POWER10 processor[C]//Proc of the 48th Int Symp on Computer Architecture. New York: ACM, 2021: 29−42

[81] Talpes E, Williams D, Sarma D. DOJO: The microarchitecture of Tesla's exa-scale computer[C/OL]//Proc of the 34th IEEE Hot Chips Symposium. Piscataway, NJ: IEEE, 2022 [2023-04-21].https://ieeexplore.ieee.org/document/9895534

[82] Choqiette J. NVIDIA hopper GPU: Scaling performance[C]//Proc of the 34th IEEE Hot Chips Symposium. Piscataway, NJ: IEEE, 2022 [2023-04-21].https://ieeexplore.ieee.org/document/9895592

[83] Smith A, James N. AMD instinct™ MI200 series accelerator and node architectures[C]//Proc of the 34th IEEE Hot Chips Symp. Piscataway, NJ: IEEE, 2022 [2023-04-21].https://ieeexplore.ieee.org/document/9895477

[84] Jiang Hong. Intel's Ponte Vecchio GPU : Architecture, systems & software[C]//Proc of the 34th IEEE Hot Chips Symp. Piscataway, NJ: IEEE, 2022 [2023-04-21].https://ieeexplore.ieee.org/document/9895631

[85] Hong M, Xu Lingjie. Biren BR100 GPGPU: Accelerating datacenter scale AI computing [C]//Proc of the 34th IEEE Hot Chips Symp. Piscataway, NJ: IEEE, 2022 [2023-04-21].https://ieeexplore.ieee.org/document/9895604

[86] NVIDIA. NVIDIA ADA GPU ARCHITECTURE [EB/OL]. 2023 [2023-04-24].http://images.nvidia.cn/aem-dam/Solutions/geforce/ada/nvidia-ada-gpu-architecture.pdf

[87] AMD. AMD compare graphics specifications [EB/OL]. 2023 [2023-04-24].https://www.amd.com/zh-hans/products/specifications/compare/graphics/11836

[88] Reuther A, Michaleas P, Jones M, et al. AI and ML accelerator survey and trends[J]. arXiv preprint, arXiv: 2210.04055, 2022

[89] Norman P, George K, Kurian G, et al. TPU v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings[J]. arXiv preprint, arXiv: 2304.01433, 2023

[90] Jouppi N, Yoon D, Ashcraft M, et al. Ten lessons from three generations shaped Google's TPUv4i : Industrial product[C/OL]//Proc of the 48th Int Symp on Computer Architecture. New York: ACM, 2021 [2023-04-28].https://dl.acm.org/doi/10.1109/ISCA52012.2021.00010

[91] Cambricon. Cambricon MLU370 chip [EB/OL]. 2023 [2023-04-21]. https://www.cambricon.com/index.php?m=content&c=index&a=lists

&catid=360

[92] Ouyang Jian, Du Xueliang, Ma Yin, et al. 3.3 kunlun: A 14nm high-performance AI processor for diversified workloads[C]//Proc of the 68th IEEE Int Solid-State Circuits Conf. Piscataway, NJ: IEEE, 2021: 50−51

[93] Skillman A, Edso T. A technical overview of cortex-m55 and Ethos-U55: Arm's most capable processors for endpoint AI[C]//Proc of the 32nd IEEE Hot Chips Symp. Piscataway, NJ: IEEE, 2020 [2023-04-22].https://ieeexplore.ieee.org/document/9220415

[94] Rao Chaolin, Yu Huangjie, Wan Haochuan, et al. ICARUS: A specialized architecture for neural radiance fields rendering[J]. arXiv preprint, arXiv: 2203.01414, 2022

[95] Matthew T, Vincent C. Block-NeRF: Scalable large scene neural view synthesis[C]//Proc of the 19th IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2023: 8238−8248

[96] Dogyoon L, Minhyeok L, Chajin S, et al. DP-NeRF: Deblurred neural radiance field with physical scene priors[C]//Proc of the 19th IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2023: 12386−12396



Yan Xinkai, born in 1989. PhD candidate, lecturer. His main research interests include computer architecture and AI & graphics chip design.

严忻恺, 1989 年生. 博士研究生, 讲师. 主要研究方向为计算机体系结构、智能图形芯片设计.



Huo Yuchi, born in 1987. PhD, professor. His main research interests include computer graphics and computational optics.

霍宇驰, 1987 年生. 博士, 研究员. 主要研究方向为计算机图形学、计算光学.



Bao Hujun, born in 1966. PhD, professor. His main research interests include computer graphics and AR/VR.

鲍虎军, 1966 年生. 博士, 教授. 主要研究方向为计算机图形、增强现实/虚拟现实.