神经辐射场加速技术综述

Survey of Neural Radiance Field Acceleration Technologies

郑清芳/ZHENG Qingfang[1,2]
（1. 中兴通讯股份有限公司，中国 深圳 518057；
2. 移动网络和移动多媒体技术国家重点实验室，中国 深圳 518055）
(1. ZTE Corporation, Shenzhen 518057, China;
2. State Key Laboratory of Mobile Network and Mobile Multimedia Technology, Shenzhen 518055, China)

DOI：10.12142/ZTETJ.202302015
网络出版地址：https://kns.cnki.net/kcms/detail/34.1228.TN.20230411.1433.014.html
网络出版日期：2023-04-11
收稿日期：2023-02-26

摘要：神经辐射场（NeRF）技术可以从2D图像中学习场景的3D隐式模型，并合成出高清逼真新视角图像。该技术有着良好的应用前景，受到业界广泛关注。针对NeRF技术运算缓慢的问题，近两年业界研究者提出了各种加速技术。对现有加速技术进行了综述，分类梳理并分析了速度提升背后的技术机理和工程技巧，同时讨论了未来加速技术演进的方向。本研究有助于激发更高效算法的产生，从而推进NeRF技术在视觉内容生成及其他领域的应用。

关键词：NeRF；神经渲染；视点合成；体渲染

Abstract: The neural radiance field (NeRF) technology, which can learn the 3D implicit representation of a scene from a set of 2D images and synthesize high-resolution and photo-realistic images of novel views, has aroused extensive research interest due to its vast application potential. In order to solve NeRF's problem of slow running speed, various acceleration technologies have been proposed in recent two years. We review current acceleration technologies by categorizing and analyzing their technical mechanism and engineering skills. We also discuss directions for further acceleration. Our work will contribute to inspiring the invention of more efficient algorithms and promote NeRF's application in multiple fields including visual content generation and beyond.

Keywords: NeRF; neural rendering; view synthesis; volume rendering

过去10年，视频相关技术[1-5]领域最深刻的变革发生在内容分析方面。自从2012年AlexNet[6]在ImageNet大规模图像识别挑战（ILSVRC）竞赛中夺冠以来，基于深度学习的计算机视觉技术突飞猛进，将内容分析的准确率提升至前所未有的水平，并催生出巨大的市场应用规模。以人脸识别为代表的各项视频内容分析技术走出实验室，服务于千行百业。

未来10年，同样激动人心的突破有望发生在视觉内容生成方面。简单便捷地从2D视频/图像集中合成出崭新视角的视频/图像，甚至重建出物体及场景的3D模型，并且画质达到照片级的逼真度和清晰度，一直是内容生成方面的长期技术研究课题[7]。2020年美国加州大学伯克利分校的研究团队提出神经辐射场技术（NeRF）[8]。NeRF因其创新的方法及突出的效果，吸引了业界的广泛关注，成为视图合成/3D重建领域新的技术框架。自从发表后近两年的时间里，该NeRF论文被引用超过1 000次。同时业界研究者对NeRF技术进行了大量的改进，并将其应用领域扩展到视频编辑[9]、数据压缩[10]、虚拟人[11]、城市建模[12]、地图构建[13]等诸多方面。

NeRF技术的一个显著缺点是模型训练及图像渲染的速度极慢。在Nvidia的高端显卡上，训练一个场景的模型需耗时1~2 d，而从模型中渲染出一幅800×800分辨率的图像需耗时超过20 s。运算速度方面的不足阻碍了NeRF技术在实际应用中的部署。可喜的是，经过业界研究者近两年的努力，渲染速度提升超过10 000倍[14]，训练速度提升超过300倍[15]。

针对NeRF的各种加速技术，本文梳理并总结了速度提升的技术机理和工程技巧，并分析各项技术之间互相结合以达到复合加速效果的可能性，从而有助于激发更高效算法的产生，进一步推进NeRF技术在内容生成及其他领域的应用。

1 相关研究工作

文献[16]和文献[17]分别对2021年3月之前的NeRF相关技术做了综述。文献[16]针对促使NeRF出现的各种技术和NeRF出现后的各种改进性技术这两个主题，提供了注释性的参考文献，但不涉及对各技术的详细说明。文献[17]将相关技术大致分为两大类：第1类对NeRF表示方法的理论性

质和不足进行分析，并提出优化策略，包括对合成精度、绘制效率以及对模型泛用性的优化；第2类则以NeRF的框架为基础对算法进行扩展和延伸，使其能够解决更加复杂的问题。文献[16]和文献[17]促使更多的研究者对NeRF进行研究，但也因其成文时间较早，无法涵盖对2021年3月以后NeRF的许多重要进展的总结。

文献[18]综述了神经渲染技术的整体发展。神经渲染技术广义上是指所有利用神经网络产生新的视觉内容的技术，而NeRF仅是其中的一个子领域，侧重于合成出新的视角的视觉内容。文献[18]重点介绍了将经典渲染与可学习3D表示相结合的高级神经渲染方法，尽管提及了许多NeRF相关的文献，但本质上不是针对NeRF的综述。

在本文的撰写过程中，加拿大滑铁卢大学的研究者在Arxiv.org上展示了预印本[19]，全面介绍了过去两年业界提出的各种NeRF改进，以及NeRF技术在各种计算机视觉任务中的应用。与文献[19]不同，本文着眼于运算速度的提升，对各种加速技术进行分类，阐释技术背后的机理和工程技巧，展现NeRF发表以来的技术演进脉络，以期为相关研究者提供有益参考。

2 NeRF技术简介

对于给定的三维场景，任意位置的外观取决于具体位置和观测角度。场景表现出的颜色与光照条件相关，导致从不同角度观察同一位置时颜色也会出现变化。NeRF是一个描述三维场景的函数$(r,g,b,\sigma)=F_\Theta(x,y,z,\theta,\varphi)$，其中$F_\Theta$用多层感知机（MLP）来具体表示。输入位置信息$(x,y,z)$和观测角度$(\theta,\varphi)$后，该函数输出该位置的体密度$\sigma$和在对应观测角度的颜色值$rgb$。在基于NeRF的场景表示基础上，可以采用经典体渲染方法渲染出不同视角的新图像。具体地，对于图像中任意像素，沿着观测角度的光线r采样N个点$X_i(i=1,\cdots,N)$，对每个采样点先根据F_Θ计算出σ_i和rgb_i，然后根据以下公式计算出最终的颜色值：

$$\hat{rgb}=\sum_{i=1}^{N} rgb_i w_i, \tag{1}$$

$$w_i=T_i\alpha_i, \tag{2}$$

$$T_i=\exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right), \tag{3}$$

$$\alpha_i=1-\exp(-\sigma_i\delta_i), \tag{4}$$

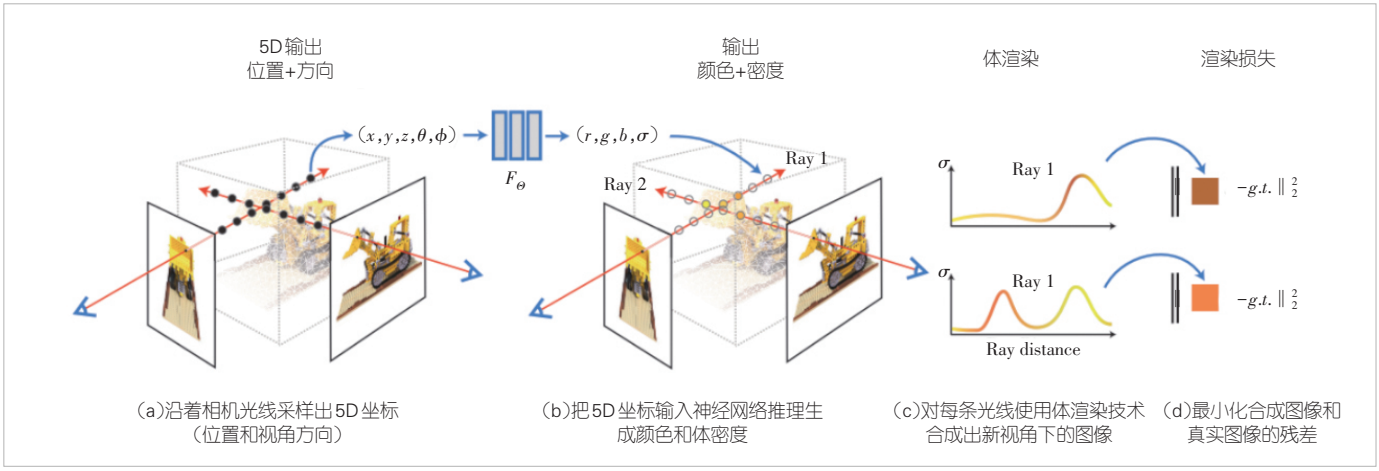
其中，δ_i代表在光线r上的采样间隔。

为了训练出F_Θ对应的MLP具体参数，对于给定的场景，采用不同位姿的摄像头拍摄得到n幅图像，利用梯度下降的方法，通过最小化预测图像I_p与真值图像I_c之间的误差对F_Θ进行拟合，即$\min\sum_{i=1}^{n}|I_p-I_c|^2$。

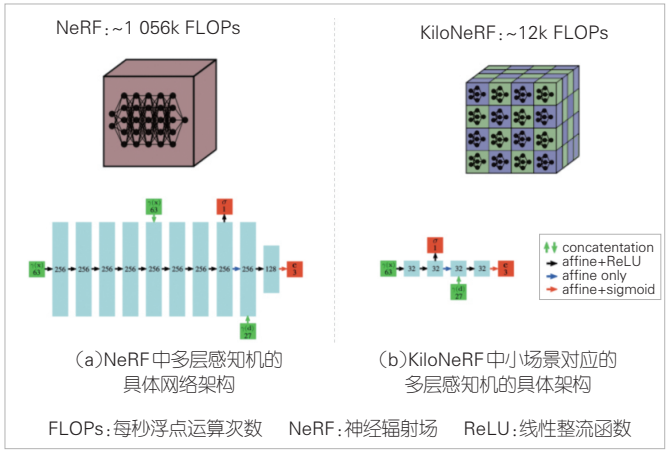
图1给出了NeRF算法的流程。在文献[8]中，作者为了得到含有更多高频信息的输出图像，对输入MLP的位置和视角参数进行了高阶编码操作$\gamma(.)$。另外，为了提升运行速度，作者采用先粗放后精细的策略提高采样的效率：首先在光线方向上均匀采样64个点，然后由这64点的密度值估计出密度分布函数，再对高密度的区域采样128点。

NeRF的运行速度十分缓慢，主要有两个原因：

1）巨大的计算量。例如：在渲染一幅分辨率为800×800的图像时，NeRF为了计算出每个像素的最终颜色值，需在光线行进的方向上采样192（64+128）个点，并进行256次MLP推理。这意味着渲染该图像总共需要800×800×256=163 840 000次MLP推理。MLP的网络架构如图2（a）所示。每次推理计算需要超过100万次浮点运算。总体而



▲图1 神经辐射场算法流程[8]



▲图2 NeRF与KiloNeRF的比较[21]

言，整个过程需要超过100T次浮点计算。

2）低效的实现方式。在通常针对Nvidia图形处理器（GPU）优化的深度学习函数库中，MLP是逐层计算的。每一层用一个核函数来实现具体的计算，并向GPU全局显存写入计算结果，而下一层在计算时又需从全局显存读取该计算结果并将其作为本层的输入。在目前的GPU芯片架构中，全局显存的数据读写速度远小于计算速度。频繁的数据读写严重制约了GPU的实际工作性能。

3 加速技术介绍

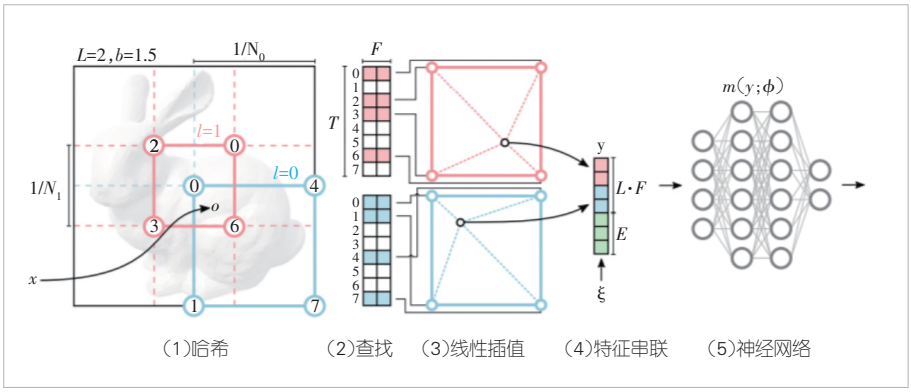
鉴于NeRF巨大的潜在应用前景，针对当前NeRF十分低效的运行速度，近两年来研究者们提出了一系列加速技术。由上一节的分析可知，对于给定分辨率的图像，NeRF的实际运行速度受到每道光线中的MLP的计算复杂度、采样点数量、硬件的技术特性等因素的综合影响。本文从采用小型化MLP、减少采样点数量、缓存中间计算结果以及充分利用硬件特性4个方面对现有技术进行分类和介绍。

3.1 采用小型化MLP

在NeRF的实现中，图像中每个像素的色彩值的计算都需上百次的MLP推理，因此减轻MLP的计算复杂度是非常有必要的。如果采用更小型化的MLP，不论是减少其深度还是宽度，都会导致模型表征能力的下降，损害最终输出图像的视觉质量。因而，此类方法的关键在于采用何种策略能够确保小型化MLP的最终输出不会影响最终的视觉质量。

DeRF[20]和KiloNeRF[21]采用了分而治之的策略：用更小的MLP来表示目标场景的一部分而非整个场景。DeRF把整个场景划分为不规则的互相独立的16个Voronoi单元。相比于NeRF中的MLP，每个Voronoi单元对应的MLP深度不变但宽度减半，因此计算量只有原来的1/4。在渲染时，由于每条光线上的每个采样点只计算与它相对应的MLP，因此整体速度可以达到原始NeRF的1.7倍。KiloNeRF采用沿坐标轴均匀分解场景的方法，最多可以把场景分为16×16×16=4 096个小场景。每个小场景对应的MLP的具体架构如图2（b）所示。该架构仅有4个隐含层且每层只有32个通道，其计算量为NeRF中MLP的1/87。相比于原始的NeRF，KiloNeRF的总体渲染速度可以达到3个数量级的加速倍速。为了保证视觉质量，在KiloNet训练过程中采用知识蒸馏的方式，使KiloNeRF的输出与NeRF的输出相一致。

Instant NeRF[22]的核心思路是：既然MLP的最终输出值取决于MLP自身的参数和输入的特征，那么小型化MLP表征能力的减弱可以通过增强输入特征的表征能力来弥补。Instant NeRF中的MLP由两个分别包含1个及2个隐含层且每层都为64个通道的小型MLP串联组成。不同于NeRF中的位置编码，Instant NeRF对输入参数采取多分辨哈希编码方式：输入参数在某个分辨率中经过哈希后对应一个特征向量，把输入参数在所有分辨率中对应的特征向量串联起来形成最终的特征向量。Instant NeRF不但加速了渲染过程，在Nvidia RTX 3090 GPU上能够以60 fps的速度输出1 920×1 080的图片，而且解决了NeRF模型训练慢的问题，将NeRF训练速度提高了60倍。实验结果表明，在最快的情况下，Instant NeRF模型的训练时间只需要5 s。图3以2D空间场景为例解释了Instant NeRF的计算过程。该计算过程包括5个步骤：1）对于给定的输入坐标x，在不同的分辨率中分别找到周围的体素；2）在哈希表中查询不同分辨率的体素所对应的特征向量；3）根据x在各自体素中的相对位置，插值计算



▲图3 Instant NeRF的计算过程[22]

出 x 在不同分辨中的特征向量；4）将在各分辨率上的特征向量串联，形成最终的特征向量；5）将特征向量输入神经网络进行推理计算

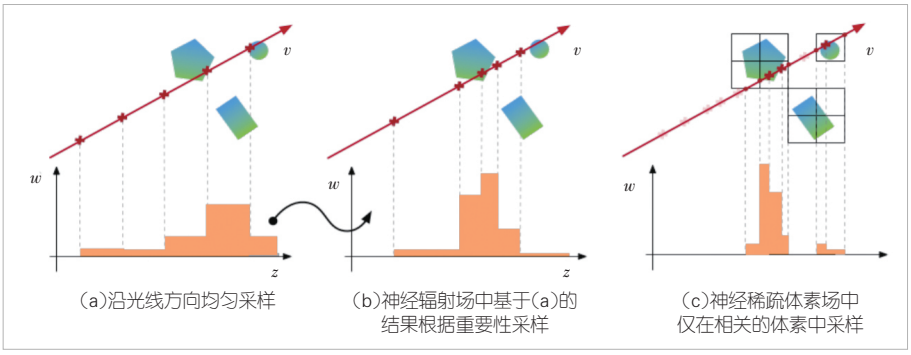
3.2 减少采样点数量

尽管 NeRF 已经采用了层次化的采样策略来避免对整条光线进行密集采样，但是仍然需要固定的 192 个采样点。事实上，由于目标场景通常无法完全充满整个三维空间，必然有某些采样点落在目标场景之外。另外，某些采样点在视角方向上被完全遮挡，使得这些采样点对最终的计算结果并无帮助。因此，更合理的采样策略应该可以避免把计算资源浪费在这些采样点上。

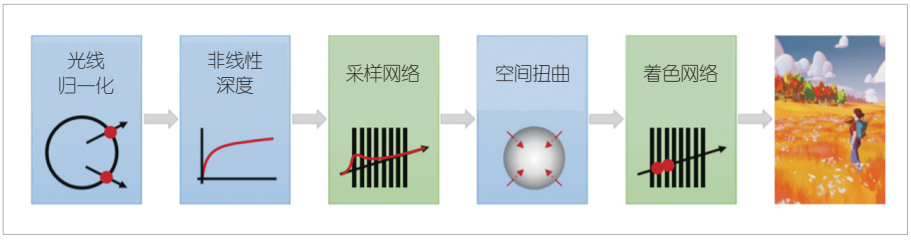
文献[23]引入了一种用于快速和高质量自由视角渲染的新神经场景表示方法：神经稀疏体素场（NSVF）。NSVF 定义了一组由稀疏体素八叉树组织的体素有界隐式场，以对每个体素中的局部属性进行建模，并为体素的每个顶点分配一个特征。体素内部具体位置的特征通过对体素 8 个顶点处的特征进行插值计算。在渲染过程中，需要对每条光线进行轴对齐边界框相交（AABB）测试，即比较从光线原点到体素的 6 个边界平面中的距离，检查光线是否与体素相交。对于不相交的空体素，可以直接跳过，从而实现 10 倍以上的渲染加速。图 4 比较了 NSVF 与 NeRF 的不同采样策略。因为 NSVF 渲染过程是完全可微的，所以可以通过将渲染的输出结果与一组目标图像进行比较，然后进行反向传播来实现端到端优化。监督训练 NSVF 的过程采用了渐进式的策略，使得不包含场景信息的稀疏体素会被修剪掉，以允许网络专注于具有场景内容的体积区域的隐函数学习。文献[23]中的实验表明，只需 1 万~10 万个稀疏体素就能够实现复杂场景的逼真渲染。

尽管 TermiNeRF[24]、NeuSample[25]、DONeRF[26]的具体做法不同，但背后的思想却是类似的：在训练的过程中，联合训练 NeRF 和一个采样神经网络，而在渲染过程中，仅需对每条光线推理一次采样神经网络，即可得到所需的全部采样点位置。以加速效果最好的 DONeRF 为例，为了在不影响图像质量的前提下大幅减少每条光线所需的采样点数量，文献[26]的作者引入真实深度信息，只考虑物体表面周围的重要采样点。DONeRF 由一个着色网络和一个采样网络组成。其中，着色网络使用类似 NeRF 的光线行进累积法来输出颜色值，而采样网络则通过将空间沿光线离散化并预测沿光线的采样概率，来预测每条光线上的多个潜在采样对象。为了消除输入的模糊性，光线被转换到一个统一的空间中。作者使用非线性采样来追踪接近的区域，并在采样网络和着色网络之间，对局部采样进行扭曲，以使着色网络的高频预测被引导到前景上。图 5 展示了 DONeRF 的计算过程。实验结果表明，DONeRF 只用 4 个采样点就取得了与 NeRF 相似的图像质量，渲染速度可实现 20 ~ 48 倍的提升。

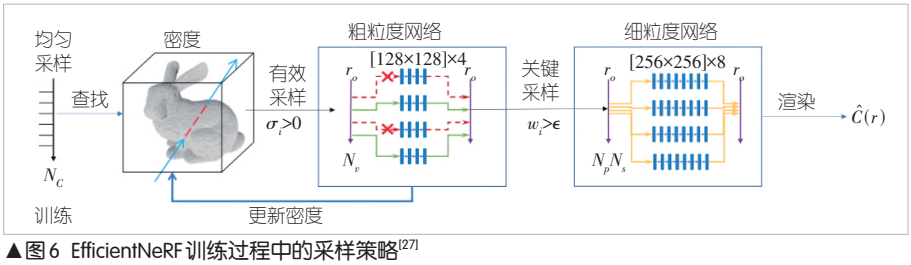
如图 6 所示，EfficientNeRF[27]在训练时采用了与 NeRF 相似的先粗放后精细的采样策略，但是在粗放采样阶段只计算体密度 $\sigma > 0$ 的有效样本，在精细采样阶段只计算 $w > 0.000\,1$ 的关键样本以及与其临近的另外 4 个关键样本，整体的训练时间减少了 88%。每个位置对应的 σ 值被初始化为非零值并存储在 V_σ 中，在后续的每次训练迭代中根据 $V_\sigma^i =$



▲图 4 神经稀疏体素场和神经辐射场的不同采样策略对比[23]



▲图 5 DONeRF 的计算过程[26]



▲图 6 EfficientNeRF 训练过程中的采样策略[27]

$(1-\beta)V_{\sigma}^{i-1}+\beta\sigma(x)$进行更新。其中，$\beta\in(0,1)$是控制更新率的参数，$\sigma(x)$是本次迭代中得到的体密度值。$w$根据公式（2）—（4）计算。

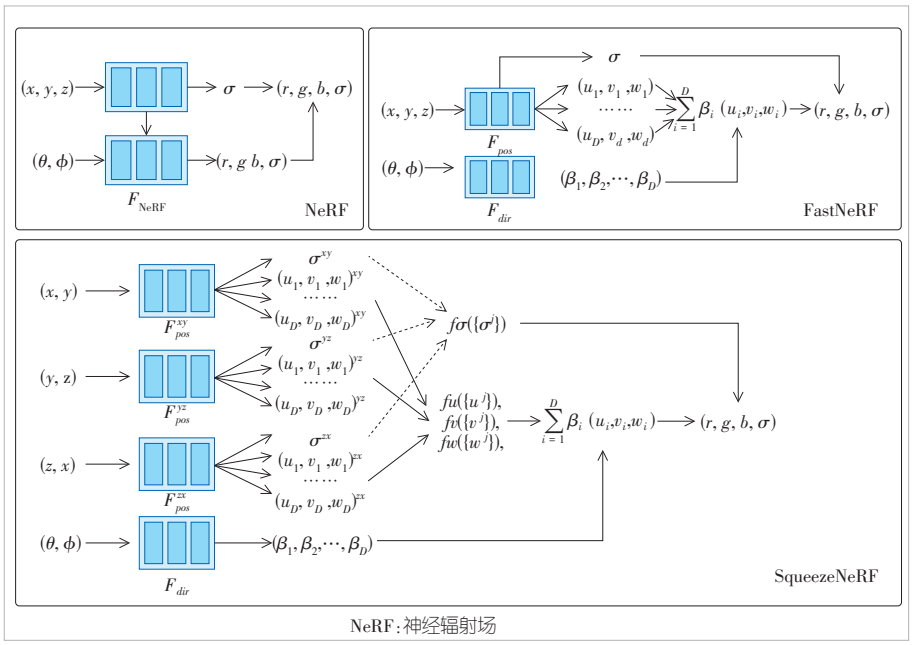
3.3 缓存中间计算结果

通过预先计算并将计算结果缓存起来，在后续使用时直接获取该结果，是常见的加速方法。NeRF本质上是将5维的输入参数（3维的位置参数+2维的视角参数）映射到4维向量的函数$(r,g,b,\sigma)=F_{\Theta}(x,y,z,\theta,\varphi)$。对此，最简单方法是直接将5维输入空间离散化，并将对应的计算结果全部缓存。但该方法因为所需的内存容量太大而不具可行性：即使假设输入参数每个维度的分辨率都是512，所需的内存也超过150 TB。

如图7所示，FastNeRF[28]和SqueezeNeRF[29]都利用了函数分解的思想。FastNeRF将NeRF分解为两个函数：第1个函数$(\sigma,u,v,w)=F_{\text{pos}}(x,y,z)$将位置参数映射到体密度$\sigma$和辐射图$(u,v,w)$；第2个函数$\beta=F_{\text{dir}}(\theta,\varphi)$则将视角参数映射到与辐射图对应的权重向量$\beta$，其中$u$、$v$、$w$、$\beta$都是D维向量。$F_{\text{dir}}$和$F_{\text{pos}}$的全部结果被预先计算并缓存。渲染时先根据位置和视角参数从缓存中分别查询得到权重向量和深度辐射图，然后通过计算二者的内积即可获得颜色值：$(r,g,b)=\sum_{i=1}^{D}\beta_i(u_i,v_i,w_i)$。该计算量远小于NeRF中的一次MLP推理，因此FastNeRF的渲染速度比NeRF提升了约3 000倍。假设k和l分别表示位置和视角方向的分辨率，缓存F_{pos}和F_{dir}所需的内存空间复杂度分别为$O(Dk^3)$和$O(Dl^2)$，且$O(Dk^3)$和$O(Dl^2)$远小于缓存NeRF所需的内存空间复杂度$O(k^3l^2)$。在通常设置中，$k=l=1\,024$，$D=8$，FastNeRF所需的最大缓存空间大约为54 GB。为了能够运行在内存更小的嵌入式设备上，如图7所示，SqueezeNeRF在FastNeRF的基础上将函数F_{pos}进一步分解为3个函数并缓存其结果，所需内存空间的复杂度也从$O(Dk^3)$降到$O(Dk^2)$。

在文献[30]中，作者首次提出NeRF-SH模型$(\sigma,k)=F_{\text{SH}}(x,y,z)$，将空间位置映射为体密度$\sigma$和球谐系数$k=(k_l^m)_{l:0\leqslant l\leqslant l_{\max}}^{m:-l\leqslant m\leqslant l}$。NeRF-SH的训练过程、渲染过程与NeRF类似。作者采用稀疏八叉树表示3维场景，在叶子节点上存储各体素位置对应的体密度σ和球谐系数k。该位置在方向(θ,φ)的颜色值为$(r,g,b)=S(\sum_{l=0}^{m}\sum_{m=-l}^{l}k_l^mY_l^m(\theta,\varphi))$，其中，$Y_l^m(\theta,\varphi)$是与方向$(\theta,\varphi)$相对应的球谐系数，$S(x)=(1+\exp(-x))^{-1}$。与NeRF相比，PlenOctree技术不仅可以将渲染速度提升3 000多倍，还能将训练速度提升约4倍。在一般的场景中，PlenOctree需要的内存空间通常不超过5 GB。EfficientNeRF[27]采用不同的数据结构来组织缓存数据，用一个两层的树NerfTree代替PlenOctree中的稀疏八叉树，实现了更快的缓存查询速度。

根据图形学知识，物体的颜色可以表示为漫反射色与镜面反射色两者之和，其中镜面反射色与相机观测角度有关。文献[31]定义一个函数$(\sigma,c_{\text{diffuse}},v_{\text{specular}})=F(x,y,z)$，场景中的每个位置都对应着体密度$\sigma$、漫反射色$c_{\text{diffuse}}$，以及与镜面反射色有关的4维特征向量$v_{\text{specular}}$。这些数据经预先计算后被缓存在稀疏神经辐射网格（SNeRG）中。在渲染时，每条光线上采样点通过查询缓存直接获取对应的σ、c_{diffuse}和v_{specular}，这些采样点的c_{diffuse}和v_{specular}分别根据σ值加权累积得到C_{diffuse}和V_{specular}。V_{specular}只需经过一次MLP推理就可得到累积的镜面反射色C_{specular}，则最终像素的颜色为$rgb=C_{\text{diffuse}}+C_{\text{specular}}$。利用该技术在AMD Radeon Pro 5500M GPU上渲染Synthetic 360°图像时，速度可达到84 fps。SNeRG中的数组内容可以通过便携式网络图形（PNG）、联合图像专家组（JPEG）等算法被压缩到平均90 MB以内。



NeRF：神经辐射场

▲图7 NeRF、FastNeRF、SqueezeNeRR 3种神经网络架构的对比

3.4 充分利用硬件特性

在实际部署中，算法总是运行在特定芯片上的。提升算法的运行速度通常意味着必须高效地利用芯片中的并行处理能力和内存/缓存资源。

在Nvidia GPU上，Instant NeRF和KiloNeRF取得显著加速效果的重要原因之一在于：小型化的MLP能够更充分利用GPU芯片的技术特性。例如，与NeRF中的MLP相比，KiloNeRF中MLP的理论计算量只有1/87，而实际加速效果却高出1 000倍。在常规的统一计算设备架构（CUDA）深度学习函数库中，MLP逐层用一个核函数计算，需要在GPU的全局显存中读写中间计算结果。对于计算小型化MLP，数据访问的时间远大于数据计算的时间。KiloNeRF和Instant NeRF都使用CUDA重新编写一个核函数，并在其中完成MLP的所有计算，从而省去中间计算结果的数据搬运操作，减少频繁启停核函数的时间开销。

为了在移动设备上部署NeRF，Google推出MobileNeRF[14]。MobileNeRF充分利用了标准GPU光栅化管道的并行性。测试结果表明，在输出图像视觉质量相当的前提下，MobileNeRF能够比SNeRG快10倍，相当于比原始NeRF快了10 000倍以上。为了适配GPU的光栅化管道，MobileNeRF采用与原始NeRF不同的训练过程和表征方法，用带有纹理的多边形来表征每个场景模型。其中，多边形大致沿着场景表面排布，纹理图中存储特征向量和离散的不透明度。渲染时MobileNeRF先利用带Z-buffering的经典多边形光栅化管道为每个像素生成特征向量，然后将特征向量传递给OpenGL着色语言（GLSL）片段着色器，并在其中运行小型化MLP，生成每个像素的色彩值。此外，MobileNeRF的GPU显存利用率也高于SNeRG，在运行过程中前者占用的GPU显存约为后者的1/5。

基于现场可编程逻辑门阵列（FPGA），上海科技大学开发了首个针对NeRF渲染算法的定制化芯片ICARUS[32]。ICARUS的架构由定制的全光核组成，其中每个全光核集成了位置编码单元（PEU）、MLP引擎和体渲染单元（VRU）。当采用40 nm互补金属氧化物半导体（CMOS）工艺且工作在300 MHz时，单个全光核仅占7.59 mm^2面积，功耗为309.8 mW，能效比GPU高146倍。ICARUS的高效性能主要得益于以下3个方面：

1）使用经过量化的定点数模型，尤其对于对复杂度最高的MLP计算，使用移位累加等近似算法。

2）全光核内部完成NeRF的全部计算过程。当芯片加载经过训练的NeRF网络模型参数后，只要输入观察位置与视角，即可输出对应像素的最终色彩值，无须在片外存储中间计算结果，从而消除了各运算单元内部、单元之间的数据存储和搬运操作。

3）每个像素的计算过程和结果完全独立，控制逻辑大大简化，可以方便地通过增加全光核数量来实现并行加速。

4 总结与讨论

NeRF技术可以从不同视角的2D图像集中学习并建立3D场景的隐含模型，并渲染出崭新视角的图像。不仅如此，新图像的视觉效果能够达到非常逼真的程度。自从2020年第1篇关于NeRF的论文发表以来，NeRF技术为视角合成乃至3D重建领域带来新的研究思路。在两年左右的时间里，该技术引起了业界广泛关注，并得到了突飞猛进的发展。在未来，NeRF技术将为视觉内容生成领域带来巨大变革，如同当前深度卷积网络技术为视觉内容分析领域带来的变革一样，在虚拟现实（VR）/增强现实（AR）及未来的元宇宙时代起到关键作用。

为了解决由NeRF技术运行速度缓慢导致的实际部署难的问题，研究者们已经提出各种加速技术。本文介绍了NeRF的技术原理，并分析该技术运行缓慢的原因：在获得每个像素的最终颜色值时，整体运行速度取决于MLP的计算复杂度、每道光线沿线的采样点数量等综合因素。本文相应地从采用小型化MLP、减少采样点数量、缓存中间计算结果以及充分利用硬件特性4个方面对现有技术进行综述，介绍了各技术的加速原理和实现方法，希望可以帮助相关研究者快速了解本领域的技术现状及演进脉络。

另外，NeRF相关技术仍在快速发展中，同时实际应用场景仍需要更加高效的加速技术。展望未来的技术发展，我们认为应重点关注以下几个研究方向：

1） 复合加速效果

必须指出的是，尽管本文从技术原理的角度做了正交分类，并在各分类中列举了代表性的工作，但所提及的诸多具体技术都综合利用了多种加速原理。例如，EfficientNeRF在训练阶段减少采样点数量，而在渲染阶段缓存计算结果；Instant NeRF和KiloNeRF都采用小型化MLP，并针对特定GPU架构优化MLP的推理速度。我们推测，通过结合额外的加速原理，现有的方法可以实现更高的加速倍数，例如：KiloNeRF可以进一步与DONeRF相结合，减少采样点数量，进一步提高渲染速度；Instant NeRF可以在训练阶段结合EfficientNeRF中的采样策略，并采用DS-NeRF[33]中在损失函数里增加深度信息约束的做法，来加快训练过程收敛。对此，我们希望本文的分析能够启发感兴趣的研究者设计出更加高效的算法。

2）训练加速和渲染加速

NeRF技术的特点是：针对每一个静态场景都需要训练一个模型，然后从模型中渲染出所需的图像。鉴于原始NeRF的训练渲染过程都十分缓慢，为了在实际应用中使用NeRF技术，加速训练过程和渲染过程都十分必要。前述加速方法中有些只适用于渲染过程，甚至是以牺牲训练速度为代价的，例如：在KiloNeRF和MobileNeRF之类的采用小型化MLP的方法中，为了保证最终模型的输出质量，需要先训练出NeRF中的MLP模型，再通过知识蒸馏的方式，训练出更小型化的MLP。

在诸如电商货品展示的应用场景中，可以通过两阶段的过程来综合利用上述两类加速方法：先离线使用某种加速方法训练出NeRF模型，并进一步转换成更高效的表达形式，然后在线展示过程中采用另外一种加速方法渲染出图片。而对于诸如3D视频通信的端到端实时应用而言，往往需要同时加速模型训练和渲染过程。在本文提及的方法中，PlenOctree、EfficientNeRF和Instant-NeRF能同时加速训练过程和渲染过程，但训练过程的加速比远小于渲染过程，训练速度远小于30 fps。

3）专用加速芯片

算法与芯片的发展总是相辅相成、互相促进的。当某种算法被广泛采用时，通常研究者会为之设计专用的加速芯片，在性能、成本、功耗等方面实现更佳的匹配，从而进一步推广算法的应用。在关于AlexNet的论文发表之后大约两年的时间，中科院计算所设计出第一款深度卷积网络加速原型芯片DIANNAO[34]。该芯片将速度提升近120倍，从此拉开波澜壮阔的人工智能（AI）计算芯片产业化序幕；在有关NeRF的论文发表之后大约两年的时间，上海科技大学设计出第一款NeRF渲染加速芯片ICARUS，使能效提升近140倍。ICARUS是否会同样在芯片产业风起云涌？现有GPU的技术特性并不完全适配神经渲染的计算流程。类似Mobile-NeRF的技术通过复杂的转化过程后，可以更加高效地利用现有GPU的并行能力，从而能够运行在移动设备中的嵌入式GPU上。我们十分期待业界共同努力，持续创新，研发出神经渲染专用加速芯片产品，并创造出巨大的市场应用空间，使得在各种设备上便捷、快速、经济地渲染出逼真高清的视觉内容成为现实。

参考文献

[1] 张嘉琪, 雷萌, 马思伟. AVS3视频编码关键技术及应用 [J]. 中兴通讯技术, 2021, 27(1): 10–16. DOI:10.12142/ZTETJ.202101004

[2] 杨文哲, 徐迈, 白琳. 视频质量增强模型加速算法 [J]. 中兴通讯技术, 2021, 27(1): 21–26. DOI:10.12142/ZTETJ.202101006

[3] 高宸, 李勇, 金德鹏. 基于图神经网络的视频推荐系统 [J]. 中兴通讯技术, 2021, 27(1): 27–32. DOI:10.12142/ZTETJ.202101007

[4] 吕达, 郑清芳. 构建智能实时网络, 使能5G视频业务繁荣 [J]. 中兴通讯技术, 2021, 27(1): 60–67. DOI:10.12142/ZTETJ.202101013

[5] GAO N Z, YU Y F, HUA X H, et al. A content-aware bitrate selection method using multi-step prediction for 360-degree video streaming [J]. ZTE Communications, 2022, 20(4): 96–109. DOI: 10.12142/ZTECOM.202204012

[6] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks [J]. Communications of the ACM, 2017, 60(6): 84–90. DOI: 10.1145/3065386

[7] CHAN S C, SHUM H Y, NG K T. Image-based rendering and synthesis [J]. IEEE signal processing magazine, 2007, 24(99): 22–33. DOI: 10.1109/msp.2007.4317461

[8] MILDENHALL B, SRINIVASAN P P, TANCIK M, et al. NeRF: representing scenes as neural radiance fields for view synthesis [M]//Computer Vision – ECCV 2020. Cham: Springer International Publishing, 2020: 405–421. DOI: 10.1007/978-3-030-58452-8_24

[9] ZHANG J K, LIU X H, YE X Y, et al. Editable free-viewpoint video using a layered neural representation [J]. ACM transactions on graphics, 2021, 40 (4): 1 – 18. DOI: 10.1145/3450626.3459756

[10] ISIK B. Neural 3D scene compression via model compression [EB/OL]. [2023-02-25]. https://arxiv.org/abs/2105.03120

[11] WENG C, CURLESS B, SRINIVASAN P P, et al. HumanNeRF: free-viewpoint rendering of moving people from monocular video [C]// Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2022: 16189–16199. DOI: 10.1109/CVPR52688.2022.01573

[12] TANCIK M, CASSER V, YAN X C, et al. Block-NeRF: scalable large scene neural view synthesis [EB/OL]. [2023-02-25]. https://arxiv. org/abs/2202.05263

[13] ZHU Z H, PENG S Y, LARSSON V, et al. NICE-SLAM: neural implicit scalable encoding for SLAM [EB/OL]. [2023-02-25]. https://arxiv.org/abs/2112.12130

[14] CHEN Z Q, FUNKHOUSER T, HEDMAN P, et al. MobileNeRF: exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures [EB/OL]. [2023-02-25]. https://arxiv. org/abs/2208.00277

[15] ZHANG Q, BAEK S H, RUSINKIEWICZ S, et al. Differentiable point-based radiance fields for efficient view synthesis [EB/OL]. [2023-02-25]. https://arxiv.org/abs/2205.14330

[16] DELLAERT F, YEN-CHEN L. Neural volume rendering: NeRF and beyond [EB/OL]. [2023-02-25]. https://arxiv.org/abs/2101.05204

[17] 常远, 盖孟. 基于神经辐射场的视点合成算法综述 [J]. 图学学报, 2021, 42(3): 376–384. DOI: 10.11996/JG.j.2095-302X.2021030376

[18] TEWARI A, THIES J, MILDENHALL B, et al. Advances in neural rendering [EB/OL]. [2023-02-25]. https://doi.org/10.48550/arXiv.2111.05849

[19] GAO K, GAO Y N, HE H J, et al. NeRF: neural radiance field in 3D vision, A comprehensive review [EB/OL]. [2023-02-25]. https://arxiv. org/abs/2210.00379

[20] REBAIN D, JIANG W, YAZDANI S, et al. DeRF: decomposed radiance fields [EB/OL]. [2023-02-25]. https://arxiv.org/abs/2011.12490

[21] REISER C, PENG S Y, LIAO Y Y, et al. KiloNeRF: speeding up neural radiance fields with thousands of tiny MLPs [C]//Proceedings of 2021 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2022: 14315–14325. DOI: 10.1109/ICCV48922.2021.01407

[22] 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. DOI: 10.1145/3528223.3530127

[23] LIU L J, GU J T, LIN K Z, et al. Neural sparse voxel fields [EB/OL]. [2023-02-25]. https://arxiv.org/abs/2007.11571

[24] PIALA M, CLARK R. TermiNeRF: ray termination prediction for efficient neural rendering [C]//Proceedings of 2021 International Conference on 3D Vision (3DV). IEEE, 2022: 1106–1114. DOI: 10.1109/3DV53792.2021.00118

[25] FANG J M, XIE L X, WANG X G, et al. NeuSample: neural sample field for efficient view synthesis [EB/OL]. [2023-02-25]. https://arxiv. org/abs/2111.15552

[26] NEFF T, STADLBAUER P, PARGER M, et al. DONeRF: towards real-time rendering of compact neural radiance fields using depth oracle networks [J]. Computer graphics forum, 2021, 40(4): 45–59. DOI: 10.1111/cgf.14340

[27] HU T, LIU S, CHEN Y L, et al. EfficientNeRF – efficient neural radiance fields [C]//Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2022: 12892–12901. DOI: 10.1109/CVPR52688.2022.01256
[28] GARBIN S J, KOWALSKI M, JOHNSON M, et al. FastNeRF: high–fidelity neural rendering at 200FPS [C]//Proceedings of 2021 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2022: 14326–14335. DOI: 10.1109/ICCV48922.2021.01408
[29] WADHWANI K, KOJIMA T. SqueezeNeRF: further factorized FastNeRF for memory–efficient inference [EB/OL]. [2023–02–25]. https://arxiv.org/abs/2204.02585
[30] YU A, LI R L, TANCIK M, et al. PlenOctrees for real–time rendering of neural radiance fields [C]//Proceedings of 2021 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2022: 5732–5741. DOI: 10.1109/ICCV48922.2021.00570
[31] HEDMAN P, SRINIVASAN P P, MILDENHALL B, et al. Baking neural radiance fields for real–time view synthesis [EB/OL]. [2023–02–25]. https://arxiv.org/abs/2103.14645
[32] RAO C L, YU H J, WAN H C, et al. ICARUS: a specialized architecture for neural radiance fields rendering [EB/OL]. [2023–02–25]. https://arxiv.org/abs/2203.01414
[33] DENG K L, LIU A, ZHU J Y, et al. Depth–supervised NeRF: fewer views and faster training for free [EB/OL]. [2023–02–25]. https://arxiv.org/abs/2107.02791
[34] CHEN T S, DU Z D, SUN N H, et al. DianNao: a small–footprint high–throughput accelerator for ubiquitous machine–learning [C]//Proceedings of the 19th international conference on Architectural support for programming languages and operating systems. ACM, 2014: 42(1): 269–284. DOI: 10.1145/2541940.2541967

作 者 简 介

郑清芳，中兴通讯股份有限公司云视频首席科学家；主要从事视频领域的技术规划及研发工作，主要研究兴趣包括沉浸式视频内容生成、低时延视频传输、视频编解码、视频智能分析等。