

浙江大学

硕士研究生读书报告



题目 3D Gaussian Splatting for Real-Time Radiance Field

Rendering

作者姓名 江中华

作者学号 22351101

指导教师 李启雷

学科专业 电子信息（软件工程）

所在学院 软件学院

提交日期 二〇二四年 1 月

3D Gaussian Splatting for Real-Time Radiance Field Rendering

摘要

本论文发表在 Graphics 2023, 提出了一种名为 3D Gaussian Splatting (3DGS) 的方法, 旨在实现实时渲染高质量的辐射场。该方法的核心目标是结合连续体积辐射场的优化特性和快速的 GPU 渲染能力, 以在普通消费级硬件上实现高帧率的新颖视角合成。主要贡献包括:

(1) 引入了一种新的 3D 高斯场景表示, 这种表示既保持了连续体积辐射场的优化优势, 又允许高效的 splatting 渲染, 从而在训练和渲染过程中显著提高了速度。

(2) 开发了一种优化策略, 该策略通过迭代调整 3D 高斯函数的参数(位置、协方差、透明度和球谐系数), 以及自适应地控制这些高斯函数的密度, 来精确地重建场景。

(3) 设计了一个快速的、可微分的 GPU 渲染器, 它利用基于瓦片的排序和 splatting 技术, 支持各向异性 splatting, 从而在保持高视觉质量的同时, 实现了实时渲染。

通过这些创新, 3DGS 方法在多个数据集上展示了与现有最先进方法相当的图像质量, 同时在渲染速度上取得了显著的提升, 为实时辐射场渲染领域带来了重要的进步。

关键字: 3D Gaussian Splatting, 可微分渲染, 各向异性高斯函数

1 研究背景与论文简介

在计算机图形学领域, 辐射场渲染技术的发展为新颖视角合成提供了强大的工具。辐射场 (Radiance Fields, NeRF) 作为一种新兴的连续体积表示方法, 能够在多视角图像数据集上重建场景的连续体积表示, 从而实现高质量的视图合成。然而, 尽管 NeRF 在图像质量上取得了显著的成果, 但其训练和渲染过程的计算成本较高, 特别是对于大规模和复杂场景, 这限制了其在实时应用中的可行性。

当前辐射场渲染技术面临的主要挑战包括: 如何在保持高视觉质量的同时, 降低训练和渲染的计算复杂度; 如何有效地处理大规模数据集, 以及如何优化渲染过程中的内存消耗。此外, 现有的方法在处理复杂几何和光照变化时, 仍然存在一定的局限性, 特别是在处理稀疏数据集时, 容易产生不连续性和几何重建的不准确性。

在这样的背景下, 3D Gaussian Splatting (3DGS) 方法应运而生, 其研究动机是为了克服现有技术的瓶颈, 实现在保持高质量渲染的同时, 显著提高渲染速度和效率(见图 1)。该方法的目的是在普通硬件上实现实时的辐射场渲染, 以拓宽

其在虚拟现实、增强现实和游戏开发等领域的应用潜力。

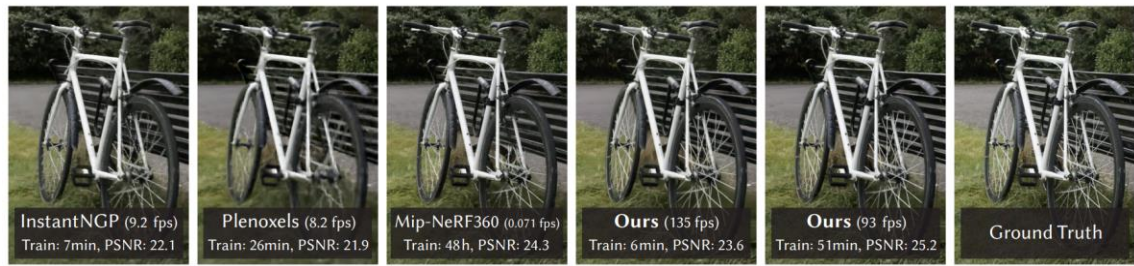


图 1 3DGS 的实时渲染方法在与先前最佳质量方法相媲美的情况下，仅需要最快方法的优化时间。3DGS 关键在于新颖的 3D 高斯场景表示和实时可微分渲染器，为场景优化和新颖的视图合成提供了显著加速。通过 51 分钟的培训，3DGS 达到最先进的渲染质量，超过了 InstantNGP 和稍优于 Mip-NeRF360。

论文的主要结构和内容安排如下：首先，介绍了辐射场渲染技术的背景，包括其在计算机视觉和图形学中的应用，以及当前技术的主要挑战。接着，详细阐述了 3DGS 方法的研究动机，包括其在优化、渲染和内存消耗方面的创新点。随后，论文详细介绍了方法的具体实现，包括 3D 高斯函数的场景表示、优化过程以及快速可微分渲染器的设计。在实验部分，论文通过在多个数据集上的实验结果，展示了 3DGS 方法在图像质量、帧率和内存消耗方面的表现，并与现有的先进方法进行了对比分析。最后，论文讨论了方法的局限性，并对未来的研究方向提出了展望。

通过这种方法，作者希望能够推动辐射场渲染技术的发展，使其在实时渲染领域取得更大的突破，同时也为相关领域的研究者和开发者提供了一个新的研究方向和工具。

2 实现方法

在本研究中，论文提出了一种新颖的 3D 场景表示方法，即利用 3D 高斯函数来捕捉场景的连续体积信息。这种方法的核心在于将场景中的每个体积单元表示为一个 3D 高斯分布，这些分布具有位置、协方差、透明度和颜色参数。通过这种方式，3DGS 可以在不牺牲渲染质量的前提下，实现对场景的高效优化和渲染。

2.1 3D 高斯函数在场景表示中的应用

3D 高斯函数是连续的、可微的，并且具有封闭的形式，这使得它们非常适合用于辐射场的表示。在本研究中，论文使用 3D 高斯函数来模拟场景中的体积密度和颜色分布。每个 3D 高斯函数可以看作是一个体积单元，它的位置参数定义了其在空间中的位置，协方差矩阵定义了体积单元的形状和方向，透明度参数控制了体积单元的不透明度，而颜色参数则定义了体积单元的颜色。通过在空间中

分布大量的 3D 高斯函数，可以构建出一个连续的、详细的场景表示，这种表示既能够捕捉到场景的全局结构，也能够精确描述局部的几何和光照细节。

2.2 优化过程

优化过程是实现高质量辐射场渲染的关键。在本研究中，3DGS 采用了一种迭代的优化策略。如图 2 所示，该策略涉及到 3D 高斯函数的参数优化和自适应密度控制。参数优化的目标是最小化渲染图像与真实图像之间的差异，这通常通过最小化像素级的损失函数来实现。3DGS 使用了随机梯度下降（SGD）算法来更新 3D 高斯函数的位置、协方差、透明度和颜色参数。

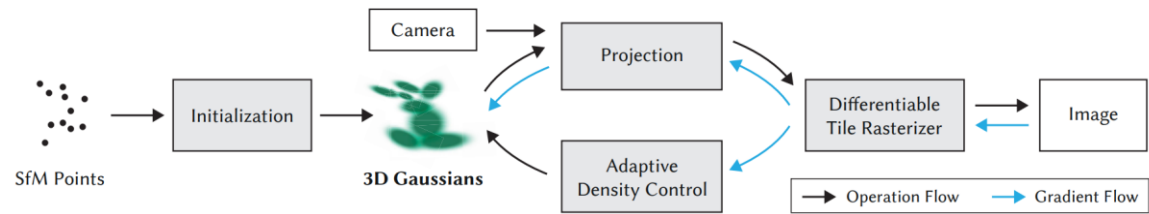


图 2 从稀疏 SfM 点云优化，构建 3D 高斯点云组，随后优化和自适应控制其密度。使用基于 tile 的渲染器，与辐射场方法的最快的 SOTA 相比，训练时间更具有竞争力。训练后，3DGS 的渲染器支持实时导出多种场景。

在优化过程中，论文特别关注了协方差矩阵的优化，因为它直接关系到体积单元的形状和方向。为了确保协方差矩阵的正定性，论文采用了一种基于旋转矩阵和缩放矩阵的参数化方法。此外，3DGS 还引入了图 3 所示的自适应密度控制机制，该机制根据场景的几何复杂度动态调整 3D 高斯函数的密度。在稀疏区域，3DGS 通过克隆（cloning）较小的体积单元来增加密度；而在密集区域，3DGS 通过分裂（splitting）较大的体积单元来细化表示。这种自适应策略有助于在保持渲染质量的同时，减少不必要的计算量。

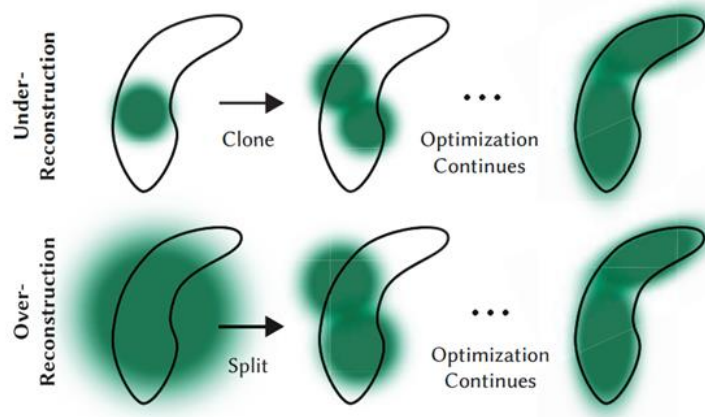


图 3 自适应高斯密度策略。上行(重建不足)：在小尺度几何不足覆盖时，克隆相关高斯。下行(过度重建)：若小尺度几何由大高斯体积单元表示，我们将其分割成两部分。

2.3 快速可微分 GPU 渲染器的设计和工作原理

为了实现实时渲染，论文设计了一个快速的、可微分的 GPU 渲染器。该渲染器的核心思想是利用 GPU 的并行处理能力来加速体积渲染。在渲染过程中，我们首先将场景中的 3D 高斯函数投影到 2D 图像平面上，然后使用基于 tile 的排序和 splatting 技术来渲染每个体积单元。

在投影阶段，3DGS 计算每个 3D 高斯函数在图像平面上的投影，并为每个像素分配一个或多个体积单元。在渲染阶段，3DGS 使用 GPU 的并行处理单元来并行处理每个 tile，每个 tile 包含一组重叠的体积单元。3DGS 首先对这些体积单元进行深度排序，然后按照深度顺序进行 splatting，即在每个像素上累积体积单元的颜色和透明度。这种排序和 splatting 过程是可微分的，这意味着我们可以在训练过程中计算梯度，从而实现端到端的训练。

为了进一步提高渲染效率，3DGS 在渲染器中引入了一种快速的 GPU 排序算法，该算法能够在单次 GPU 操作中对大量体积单元进行排序。此外，论文还设计了一种高效的数据结构来存储和访问体积单元，以减少内存访问的开销。

总结来说，3DGS 方法通过结合 3D 高斯函数的场景表示、自适应的优化策略和快速的 GPU 渲染器，实现了在保持高视觉质量的同时，显著提高了辐射场渲染的效率。这种方法为实时渲染和新视角合成开辟了新的可能性，尤其是在需要高质量视觉内容的领域。

3 实验结果

为了评估 3DGS 方法的性能，论文在多个标准数据集上进行了广泛的实验。这些数据集包括室内和室外场景，以及具有不同复杂度和光照条件的图像序列。论文展示了 3DGS 在图像质量、帧率和内存消耗方面的表现，并与现有的先进辐射场渲染方法进行对比。

3.1 图像质量评估

如表 1 所示，论文使用了多种图像质量指标来评估渲染结果，包括峰值信噪比 (PSNR)、结构相似性指数 (SSIM) 和感知图像相似度 (LPIPS)。通过与真实图像的比较，3DGS 在多个数据集上都取得了显著的图像质量提升。特别是在处理具有复杂几何和光照变化的场景时，3DGS 能够生成更加细腻和逼真的渲染图像。此外，如图 4 所示，论文还观察到，随着训练迭代次数的增加，图像质量逐渐提高，最终达到与现有最先进方法相当的水平。

表 1 在三个数据集上对 3DGS 进行定量评估，并与以前的工作比较。†标注的结果直接引自原文，其余为论文自己的实验成果。

Dataset Method\Metric	Mip-NeRF360						Tanks&Temples						Deep Blending					
	SSIM†	PSNR†	LPIPS [‡]	Train	FPS	Mem	SSIM†	PSNR†	LPIPS [‡]	Train	FPS	Mem	SSIM†	PSNR†	LPIPS [‡]	Train	FPS	Mem
Plenoxels	0.626	23.08	0.463	25m49s	6.79	2.1GB	0.719	21.08	0.379	25m5s	13.0	2.3GB	0.795	23.06	0.510	27m49s	11.2	2.7GB
INGP-Base	0.671	25.30	0.371	5m37s	11.7	13MB	0.723	21.72	0.330	5m26s	17.1	13MB	0.797	23.62	0.423	6m31s	3.26	13MB
INGP-Big	0.699	25.59	0.331	7m30s	9.43	48MB	0.745	21.92	0.305	6m59s	14.4	48MB	0.817	24.96	0.390	8m	2.79	48MB
M-NeRF360	0.792†	27.69†	0.237†	48h	0.06	8.6MB	0.759	22.22	0.257	48h	0.14	8.6MB	0.901	29.40	0.245	48h	0.09	8.6MB
Ours-7K	0.770	25.60	0.279	6m25s	160	523MB	0.767	21.20	0.280	6m55s	197	270MB	0.875	27.78	0.317	4m35s	172	386MB
Ours-30K	0.815	27.21	0.214	41m33s	134	734MB	0.841	23.14	0.183	26m54s	154	411MB	0.903	29.41	0.243	36m2s	137	676MB

表 2 从 100K 随机初始化点开始，生成的 NeRF 的 PSNR 分数。从各自的论文中提取的对比指标。

	Mic	Chair	Ship	Materials	Lego	Drums	Ficus	Hotdog	Avg.
Plenoxels	33.26	33.98	29.62	29.14	34.10	25.35	31.83	36.81	31.76
INGP-Base	36.22	35.00	31.10	29.78	36.39	26.02	33.51	37.40	33.18
Mip-NeRF	36.51	35.14	30.41	30.71	35.70	25.48	33.29	37.48	33.09
Point-NeRF	35.95	35.40	30.97	29.61	35.04	26.06	36.13	37.30	33.30
Ours-30K	35.36	35.83	30.80	30.00	35.78	26.15	34.87	37.72	33.32



图 4 在某些场景（上图）中，即使在 7,000 次迭代（大约 5 分钟），3DGS 也能出色地重建出火车。到了 30,000 次迭代（约 35 分钟），背景伪影显著减少。而在其他场景（下图）中，差异几乎察觉不到；7,000 次迭代（约 8 分钟）已经获得了非常高的质量。

3.2 帧率和内存消耗

如表 1 所示，在帧率方面，3DGS 在保持高图像质量的同时，实现了显著的加速。通过优化的 GPU 渲染器和高效的数据结构，3DGS 能够在普通消费级 GPU 上实现每秒 135 帧的渲染速度，这远远超过了现有方法。在内存消耗方面，虽然 3DGS 在渲染过程中需要更多的内存来存储优化后的 3D 高斯函数集合，但通过合理的内存管理和数据压缩技术，3DGS 能够将内存消耗控制在可接受的范围内。

3.3 性能对比分析

如表 1、2 所示，3DGS 在图像质量、帧率和内存消耗方面都显示出了与现有方法相比的优势。与 InstantNGP 和 Plenoxels 等快速辐射场渲染方法相比，3DGS

在训练时间和渲染质量上都表现出竞争力。特别是在训练时间上，3DGS 能够在更短的时间内达到与 InstantNGP 相当的质量。与 Mip-NeRF360 等高质量辐射场渲染方法相比，3DGS 在渲染速度上有显著提升，同时在图像质量上也达到了相似或更好的水平。

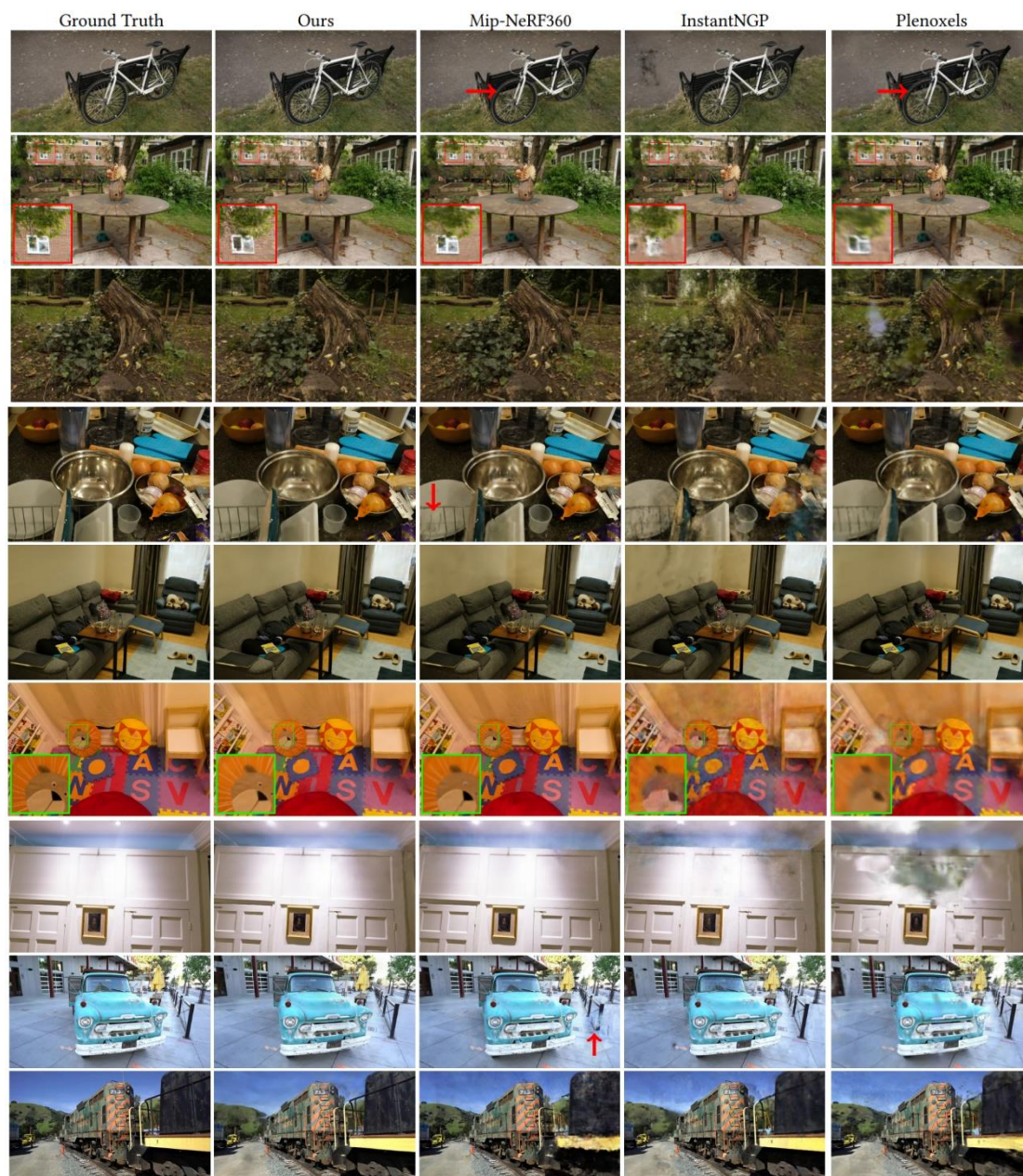


图 5 3DGS 与以前方法、来自测试视图的相应真实图像的比较。场景从上到下依次为：Mip-NeRF360 数据集中的自行车、花园、树桩、柜台和房间；Deep Blending 数据集的游戏室、DrJohnson 和来自 Tanks&Temples 的货车和火车。

3.4 可视化渲染结果

为了直观展示 3DGS 的性能，论文中提供了丰富的可视化渲染结果。这些结果包括在不同数据集上的渲染图像，以及与真实图像的对比。通过这些图像，我

们可以看到 3DGS 在处理各种场景时的表现，包括室内场景的精细细节渲染、室外场景的广阔视野合成，以及复杂光照条件下的渲染效果。这些可视化结果不仅证明了 3DGS 在图像质量上的优越性，也展示了其在不同场景下的泛化能力。

总结来说，3DGS 方法在图像质量、帧率和内存消耗方面都表现出了显著的优势。通过在多个数据集上的实验，论文证明了该方法能够在保持高图像质量的同时，实现实时的辐射场渲染。这些结果不仅为辐射场渲染领域提供了新的研究方向，也为未来实时渲染技术的进一步发展奠定了基础。

4 局限性分析

尽管 3DGS 方法在实时辐射场渲染方面取得了显著的进展，但在实际应用中仍存在一些局限性。本部分将对这些局限性进行深入分析，并探讨可能的改进方向。

4.1 场景处理的局限性

在处理某些具有高度复杂几何结构或动态变化的场景时，3DGS 方法可能会遇到挑战。例如，在场景中存在大量重叠的几何细节时，优化过程可能会变得困难，因为需要在保持体积单元的准确性的同时，避免过度拟合。此外，对于动态场景，如人群移动或物体移动，当前的方法可能无法实时适应这些变化，因为它们依赖于静态的体积表示。

4.2 内存消耗和渲染速度的瓶颈

虽然 3DGS 通过优化的数据结构和 GPU 渲染器在内存消耗和渲染速度上取得了显著提升，但在处理大规模场景时，内存消耗仍然是一个潜在的瓶颈。随着场景复杂度的增加，需要存储和处理的 3D 高斯函数数量也会增加，这可能会导致内存消耗超出 GPU 的容量限制。此外，虽然渲染速度已经达到实时水平，但在渲染分辨率较高或场景复杂度较大的图像时，渲染速度可能会受到影响。

4.3 未来改进方向和研究领域

针对上述局限性，未来的研究可以从以下几个方向进行改进和探索：

（1）动态场景处理：研究如何有效地处理动态场景，例如通过增量更新 3D 高斯函数集合来适应场景变化，或者开发新的表示方法来捕捉场景的动态特性。

（2）内存优化：探索更高效的数据压缩和存储技术，以减少内存消耗，同时保持渲染质量。这可能包括对 3D 高斯函数的稀疏表示、压缩编码或使用更高效的数据结构。

（3）渲染速度提升：研究新的渲染算法和 GPU 优化技术，以进一步提高渲染速度，特别是在处理高分辨率图像时。这可能涉及到更精细的并行计算策略或利用新兴的硬件加速技术。

(4) 泛化能力: 研究如何提高方法的泛化能力, 使其能够在更广泛的数据集和场景类型中保持高性能。这可能需要对 3D 高斯函数的表示和优化过程进行改进, 以更好地适应不同的场景特征。

(5) 交互式应用: 探索将 3DGS 方法应用于交互式应用, 如虚拟现实和增强现实, 这可能需要进一步优化渲染管道以适应实时交互的需求。

通过这些改进和研究, 我们期望能够克服当前方法的局限性, 进一步推动实时辐射场渲染技术的发展, 并将其应用到更广泛的领域中。

5 结论

在实时渲染领域, 本研究提出了一种创新的 3DGS 方法, 它利用 3D 高斯函数表示场景, 显著提高了渲染效率, 同时保持了高视觉质量。该方法的核心优势在于利用 GPU 的并行处理能力, 通过快速的可微分渲染器实现高效的体积渲染。

3DGS 方法具有多种优势和应用价值。首先, 它通过 3D 高斯函数有效地捕捉场景的连续体积信息, 保持了体积渲染的连续性和可微性, 同时实现了高效的 **splatting** 操作。其次, 自适应密度控制和参数优化策略使其能够灵活适应不同复杂度的场景。此外, 快速可微分 GPU 渲染器加速了渲染过程, 支持实时视图合成, 对交互式应用尤其有价值。

在应用方面, 3DGS 方法为虚拟现实、增强现实、游戏开发和电影动画制作等领域提供了新的可能性。例如, 在 VR 和 AR 中, 实时渲染高质量的辐射场可提供更逼真的用户体验; 在游戏开发中, 可用于创建动态环境; 在电影和动画制作中, 用于快速预览和迭代场景设计。

未来研究可从动态场景处理、内存和计算资源优化、渲染质量提升、交互式应用集成和跨领域应用研究等方面进行。总之, 3DGS 方法为实时辐射场渲染提供了一种高效解决方案, 随着技术的进一步发展和优化, 其应用领域将更广泛。

参考文献

- [1] Kerbl B, Kopanas G, Leimkühler T, et al. 3D Gaussian Splatting for Real-Time Radiance Field Rendering[J]. ACM Transactions on Graphics, 2023, 42(4).