

黑龙江大学

本科毕业生毕业论文

论文题目： 基于 3D 高斯溅射的实时辐射场渲染算法的设计与实现

学 院： 计算机与大数据学院

年 级： 2020 级

专 业： 软件工程

姓 名： 杨阳

学 号： 20201556

指导教师： 马吉权

2025 年 5 月 10 日

摘要

随着多模态医学影像融合、精准手术导航和自动驾驶感知等技术的发展，作为关键支撑的新视角合成（Novel View Synthesis, NVS）正成为计算机视觉研究热点。辐射场方法最近颠覆了使用多张照片或视频捕捉的场景的新新视图合成，但要实现较高的视觉质量仍需要训练和渲染成本高昂的神经网络，在近期的研究中较快的方法难免以牺牲速度的代价换取视觉质量。

3D Gaussian Splatting 方法引入了三个关键要素，最终能够在具有竞争力的训练时间下达到最先进的视觉质量，同时在 1080p 分辨率下实现质量高的实时（ ≥ 30 fps）新视图合成。

该方法首先基于 Structure-from-Motion（SfM）获取稀疏点云，在此基础上初始化生成三维高斯体元，这些高斯体元保留了有利于场景优化的连续体积辐射场的理想特性，同时避免了在空白区域的冗余表示；随后对高斯体元进行密度控制，同时优化各向异性协方差实现场景的准确表示；最后 3D Gaussian Splatting 方法开发了一种高效的可见性感知的支持各向异性溅射的算法，其技能加速训练又能实现实时渲染。此方法在多个数据集上展示了最先进的视觉质量和实时渲染效果。

关键词

新视角合成；三维高斯溅射；体积渲染；显式表示；实时渲染

Abstract

With the development of multimodal medical image fusion, precise surgical navigation, and autonomous driving perception, novel view synthesis (NVS), as a key support, is becoming a hot topic in computer vision research. Radiant field methods have recently revolutionized novel view synthesis of scenes captured using multiple photos or videos, but high visual quality still requires expensive neural networks to train and render. In recent studies, faster methods inevitably sacrifice speed for visual quality.

The 3D Gaussian Splatting method introduces three key elements that ultimately achieve state-of-the-art visual quality at competitive training time, while achieving high-quality real-time (≥ 30 fps) novel view synthesis at 1080p resolution.

The method first obtains sparse point clouds based on Structure-from-Motion (SfM), and then initializes and generates three-dimensional Gaussian voxels based on this. These Gaussian voxels retain the ideal characteristics of continuous volume radiation fields that are conducive to scene optimization, while avoiding redundant representation in blank areas; then the density of the Gaussian voxels is controlled, and the anisotropic covariance is optimized to achieve accurate representation of the scene; finally, the 3D Gaussian Splatting method develops an efficient visibility-aware anisotropic splatting-supporting algorithm, which can accelerate training and achieve real-time rendering. This method demonstrates state-of-the-art visual quality and real-time rendering effects on multiple datasets.

Key words

Novel View Synthesis; 3D Gaussian Splatting; Volume Rendering; Explicit Representation; Real-Time Rendering

目录

摘要	I
Abstract	II
前言	3
第一章 绪论	4
1.1 研究背景与意义	4
1.2 国内外研究现状	5
1.2.1 传统的场景重建与渲染	5
1.2.2 神经渲染和辐射场	6
1.2.3 基于点的渲染和辐射场	7
1.3 研究问题分析与本文主要研究内容	8
1.4 本文的组织架构	10
第二章 高效新视角合成的神经渲染理论基础	11
2.1 新视角合成模型基础	11
2.1.1 相机成像原理与坐标系转换	11
2.1.2 三维场景表达	12
2.2 神经辐射场 (NeRF) 基本原理	13
2.2.1 辐射场的定义与神经辐射场的引入	13
2.2.2 体积渲染机制	14
2.3 实时渲染技术的理论基础	15
第三章 基于 3D 高斯溅射的实时辐射场渲染	17
3.1 可微的 3D 高斯溅射	17
3.2 优化机制	18
3.2.1 球谐函数的引入	18
3.2.2 高斯的自适应控制	19
3.3 快速可微的光栅化器	21
第四章 实验与结果分析	22
4.1 实验准备	22

4.1.1 数据集选择	22
4.1.2 硬件环境与软件工具	22
4.1.3 评价指标	23
4.2 核心实验结果	24
4.2.1 视觉效果对比	25
4.2.2 速度与效率提升	26
4.3 消融实验分析	27
4.3.1 初始化策略对模型表现的影响	28
4.3.2 克隆与分裂策略的对比	28
4.3.3 各向异性协方差的重要性	29
结论	30
参考文献	31
致谢	33

前言

近年来, 高效的新视角合成方法多依赖于由离散数据插值构建的连续场景表示。例如 NeRF (Neural Radiance Field) [1] 通过体素网格、哈希网格或点基元存储场景特征并进行空间插值。InstantNGP [2] 利用哈希网格和较小的多层感知机 (MLP) 来表示密度和外观。Plenoxels [3] 使用稀疏体素网格插值连续密度场。然而这类方法为求解辐射场积分需要沿光线方向进行高密度的随机采样, 导致了渲染的计算成本高昂, 同时容易引入噪声, 难以满足实时交互的需求。例如, Mip-NeRF360 [4] 虽然在图像质量上达到了先进水平, 但训练时间长达 48 小时, 渲染速度仅为 0.071 fps, 无法实现实时渲染。而与之相反的, 离散的网格和点也经常出现在 3D 场景表示中, 因为它们非常适合基于 GPU/CUDA 的快速光栅化。

为了实现基于多张照片描述的场景的实时渲染并在效率上与业内最高效的方法平齐。本文引入了一种结合了连续场景表示与离散场景表示的优点的基于 3D 高斯散点的高效辐射场方法 (3D Gaussian Splatting) [5], 其可以通过优化 3D 高斯表示保持最佳的视觉质量和有竞争力的训练时间, 同时其基于图块的溅射方案能确保在先前发布的多个数据集上以最佳视觉质量实时渲染 1080p 分辨率的图像。

3D Gaussian Splatting 方法在最初以运动结构恢复 (SfM) 生成的稀疏点云来初始化 3D 高斯体元, 与大多数需要多视角立体数据的基于点的方案相比, 3DGS 仅使用 SfM 点作为输入即可获得高质量的结果, 甚至在 NeRF 合成数据集上即使使用随机点初始化 3D 高斯体元也能获得高质量效果。

初始化过后我们将优化 3D 高斯体元的各个属性——3D 位置、不透明度 α 、各向异性协方差和球谐函数 (SH) 系数。这个步骤将与自适应密度控制步骤交错进行, 即在优化的过程中我们会偶尔添加或删除 3D 高斯体元, 从而生成一个相当紧凑、非结构化且准确的场景表示 (在测试的场景中会出现 100 万到 500 万个高斯体元)。

在实时渲染方面 3D Gaussian Splatting 方法受基于图块的光栅化方法的启发, 开发了一种高效的可见性感知的支持各向异性溅射的算法, 同时实现快速准确的反向传播。

该方法在先前发布的数据集上都得到了接近或超越先前隐式辐射场方法最佳质量的结果, 最重要的是该方法首次实现了高质量的新视图合成实时渲染。

第一章 绪论

1.1 研究背景与意义

近期虚拟现实、增强现实、自动驾驶和医学图像导航等技术发展迅速，从有限的观测视角合成任意角度的真实图像成为一项关键需求。这就是新视角合成（Novel View Synthesis, NVS）任务，其核心目标是从一组已知视角下采集的图像中，推理出新视角下场景的视觉表现。新视角合成算法的核心流程一般涵盖了采集图像、提取特征、重建场景以及渲染视角等关键步骤，如图 1-1 所示。理想的 NVS 系统应具备高保真度、广视角范围和良好的实时响应能力。

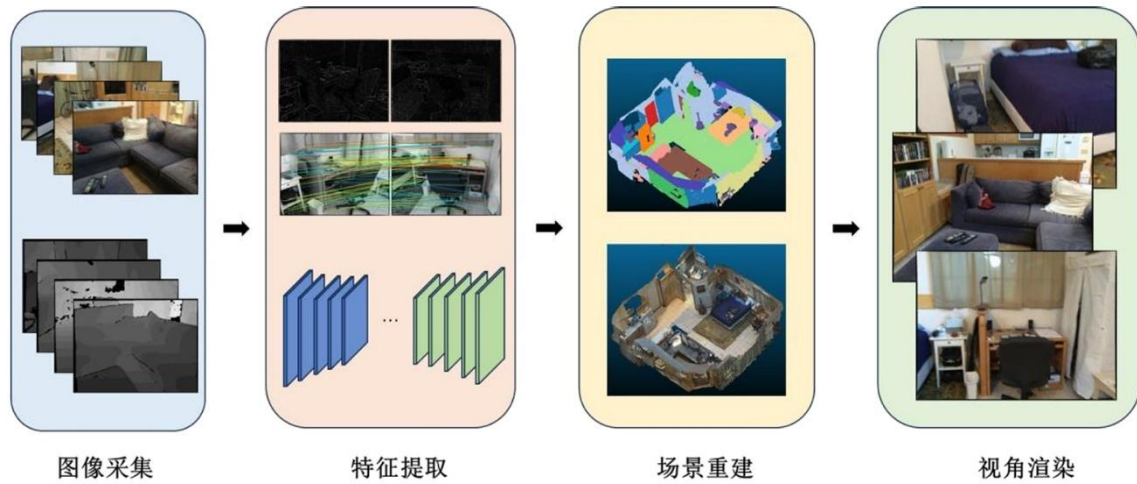


图 1-1 新视角合成算法的基本流程^[6]

NeRF（Neural Radiance Fields）近年来率先提出以神经网络拟合连续体积场景的颜色与密度函数，精细的三维建模和高质量的新视角合成被实现^[1]。后续包括通过哈希编码加速的 Instant-NGP^[2]、处理大范围场景的 Mip-NeRF360^[4]、摆脱神经网络依赖的 Plenoxels^[3] 等在内的大量研究被引发。尽管这些以隐式神经体积表示为基础的方法在图像质量上表现优异，训练开销大、渲染速度慢等问题却仍是它们的瓶颈。

于是研究者们为突破性能限制，开始探索更高效的显式表示方式。3D Gaussian Splatting 被随之提出。它以高斯点云作为辐射场的核心建模单元，空间分布通过各向异性协方差描述，结合球谐函数表达颜色特征，瓦片化排序与前向传播并在 GPU 上的采用使其能够进行高效渲染^[5]。在 1080p 分辨率下能够实现每秒 135 帧的实时渲染能力的 3D Gaussian Splatting 方法，训练时间较 Mip-NeRF360 缩短近 60 倍，显式表示在 NVS

任务中的巨大潜力随之显现。

置身于动态场景或特定应用场景中，目前主流的新视角合成方法适应性仍有限。而自动驾驶、医学导航、机器人感知等领域，迫切需要具备高效率、高质量、可导性强等特性的显式渲染框。能够更好地平衡渲染质量与计算效率的 3D 高斯渲染，作为一种融合传统图形管线与现代学习优化策略的新兴方法，尤其适用于对时效性和精度要求并存的复杂任务。于是在高效新视角合成中的研究与实践我们聚焦于 3D Gaussian Splatting 方法，旨在为未来多场景、多任务的三维视觉系统提供更强健的理论支持与技术基础。

1.2 国内外研究现状

在本节中我们简要概述传统的场景重建工作，和基于神经网络或基于点的辐射场、渲染工作。

1.2.1 传统的场景重建与渲染

早期基于光场的新视图合成办法中，在 1996 年由 Gortler 等人提出的 Lumigraph 模型和 Levoy 等人提出的 Light Field Rendering 技术均通过密集采样构建四维函数描述光在空间中的传播来实现新视图合成。2001 年 Buehler 等人扩展了 Gortler 等人提出的 Lumigraph 模型，开发出一种不依赖规则相机阵列的新方法，支持从非结构化相机分布（如手持拍摄的图像）中重建视图。

2006 年，Snavely 等人最早将 SfM 带入大众视野，SfM 在相机校准期间估计稀疏点云，最初用于 3D 空间的简单可视化。它的出现开启了一个全新的领域，在这个领域中，合成新视图可以仅仅使用该场景的一组照片。随后 Schönberger 等人于 2016 年系统地优化了 SfM 管线各阶段，提出了高效稳健的重建方法，广泛用于后续的多视图立体视觉（MVS）系统。

令人印象深刻的全三维重建算法被 Goesele 等人于 2007 年提出后，多种多视图合成算法的开发被推动。这些方法都将输入图像重投影并融合到新视图的相机中，并使用几何体来引导这种重投影。虽然它们在许多情况的运行结果优异，但它们仍然无法完全恢复未重建区域，还有可能出现“过度重建”（即生成不存在的几何体）。近期的神经渲染算法优化了这种过度重建的问题。

点基元（point primitives）在传统渲染系统中也同样重要。Gross 与 Pfister 所著的

《点基图形学》(Point-Based Graphics)^[7]系统性总结了点为基本单位的建模与渲染方法。Zwicker 等人提出的表面 Splatting 技术^[8], 利用各向异性高斯分布对点云进行加权投影, 因点间间距不均导致的缝隙问题被减少, 渲染的连续性得到提升。Botsch 等人则针对现代 GPU 架构, 将 Surface Splatting 实现进行优化, 提升了点云渲染的性能与质量。不过此类方法大多缺乏显式的颜色建模能力, 且复杂的光照变化和材质属性不好被表达。

为了兼并传统算法的效率与灵活性, Merrill 和 Grimshaw^[9]在研究通用 GPU 流处理架构下的排序算法时, 提出了优化级联操作效率的策略。这一技术思路在后续如 3D Gaussian Splatting 等方法中被进一步继承和发展。

1.2.2 神经渲染和辐射场

传统基于几何的方法随着深度学习的迅猛发展, 在新视角合成任务中逐渐暴露出效率低、细节复原能力弱的瓶颈。研究者们开始引入神经网络对三维场景进行建模, 以 Neural Radiance Fields (NeRF) 为代表的一系列神经渲染方法随之诞生。NeRF 首次在网络中引入体渲染公式, 通过多层感知机 (MLP) 学习从相机光线投影到空间中的体密度和颜色分布函数, 隐式地表示场景^[1]。

NeRF 模型中三维空间中各点的颜色与密度分布由神经网络习得, 使用一组包含相机姿态的多视角图像进行训练, 同时借助多个角度的观测信息进行优化, 使模型能够有效推断场景中未被观测的视角下的图像内容。如下图 1-2 所示, NeRF 的训练流程可分为三个核心阶段: 首先, 从训练图像中获取一系列光线采样点, 记录其三维空间坐标 (x, y, z) 及观察方向 (θ, ϕ) , 形成五维输入向量; 随后, 将这些向量输入多层感知机 (MLP) 以预测对应位置的体密度 σ 和颜色值 RGB; 最后, 结合体渲染过程累积该光线上多个点的颜色贡献, 生成预测图像, 并通过与真实图像的误差计算损失函数, 利用反向传播优化网络参数, 从而提升模型对新视角的生成能力。

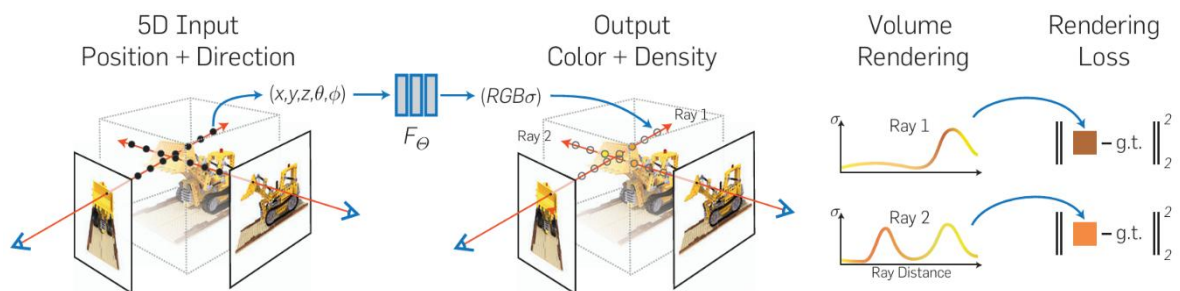


图 1-2 NeRF 训练流程^[1]

Müller 等人为加速训练与渲染效率提出了 Instant-NGP^[2]，高维位置编码被引入的多分辨率哈希编码替代，结合稀疏体素网格快速定位与特征插值，使其几分钟内即可完成小规模场景建模。神经网络在 Fridovich-Keil 与 Yu 进一步提出 Plenoxels 方法^[3]中被摒弃，颜色球谐系数（Spherical Harmonics）使用稀疏体素直接存储，保证了质量与计算效率。Chen 等人提出的 TensorRF^[10] 则将空间特征表示为四维张量，通过张量分解方法降低冗余特征维度，减少了存储开销。提出在体素网格的基础上引入视锥变换的 Wang 等人^[11]，对近远景采用不同分辨率采样策略，增强了模型对任意视角路径下场景细节的建模能力。

为解决传统 NeRF 结构在大尺度或无边界场景容易产生背景模糊、远景失真的问题，Barron 等人通过视锥采样与多分辨率卷积调整采样粒度提出了 Mip-NeRF 和 Mip-NeRF360^[4]，网络对场景尺度变化的鲁棒性得到了增强，背景区域变得更清晰。虽然该方法训练时间仍需 24 小时以上。后来 Chen 等人进一步提出了 MobileNeRF^[12]。该方法通过将 NeRF 表示转换为带有纹理映射的多边形网格，并利用传统的栅格化渲染与片段着色器网络进行颜色预测，实现了支持在移动端进行实时交互的轻量化、高帧率的图像渲染。

NeRF 应用场景从实验室逐渐转向现实世界后，Block-NeRF^[13] 将城市级别的三维场景划分为多个模块（Block），分别建模并通过 3D 配准融合连接区域，实现多视角无缝浏览。BungeeNeRF^[14] 则引入渐进式多尺度监督机制，在大范围复杂环境中同时保留全局结构与局部细节，满足交互式渲染的需求。

1.2.3 基于点的渲染和辐射场

离散且非结构化的集合样本（即点云）可以被基于点的方法高效渲染^[7]。通过对一组非结构化点集合进行光栅化并赋予固定尺寸实现渲染，是最简单的点样本渲染形式。渲染这些点的方法有两种：一种是直接用图形工具里已经提供的“点渲染”功能；另一种是自己在 GPU 上编程，把点转成图像，这种方法虽然复杂一点，但更灵活、更强大。

这种忠于原始数据的方法也容易产生孔洞和锯齿伪影，并且在图像上离散呈现。早期的高质量点渲染开创性地使用了“溅射”技术（splatting）来解决此问题，溅射将点元渲染为大于像素范围的图元（如圆盘、椭圆盘、椭球等）^[8]。

可微分的点渲染技术 (differentiable point-based rendering) 也在近年来得到了发展, 研究借助卷积神经网络 (CNN) 进行渲染, 实现了快速甚至实时的新视图合成方法。但这些方法因为依赖多视图立体 (MVS), 在无纹理区域或细长结构等复杂情况中易导致过重建或欠重建, 继承了 MVS 的缺陷。

基于点的 α 混合渲染与 NeRF 式的体积渲染采用的图像生成模型本质上是相同的, 即图像颜色 C 由沿着一条射线进行的体积渲染积分计算得到。见下式 (1-1)。

$$C = \sum_{i=1}^N T_i (1 - \exp(-\sigma_i \delta_i)) c_i, T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j) \quad (1-1)$$

密度 σ 、透射率 T 和颜色 c 的样本沿射线以 δ_i 为间隔进行采样。这可以重写为公式 (1-2)。

$$C = \sum_{i=1}^N T_i \alpha_i c_i \quad (1-2)$$

其中

$$\alpha_i = (1 - \exp(-\sigma_i \delta_i)), T_i = \prod_{j=1}^{i-1} (1 - \alpha_j) \quad (1-3)$$

如式 (1-4), 非典型的基于神经点的方法靠混合与像素重叠的 N 个有序点来确定像素颜色。

$$C = \sum_{i \in N} \alpha_i c_i \prod_{j=1}^{i-1} (1 - \alpha_j) \quad (1-4)$$

其中 c_i 是每个点的颜色, α_i 是对协方差为 Σ 的二维高斯函数乘以学习到的每点不透明度得到的。

我们由公式 (1-2) 与 (1-4) 可看出其图像生成的模型相同, 但渲染算法不同。NeRF 是一种连续的隐式表示, 需要进行代价昂贵的随机采样获取公式 (1-2) 中的采样点, 导致了噪声和计算开销。而点是一种非结构化的离散表示, 并且可以通过优化不透明度和位置灵活地创建、销毁和移动几何体, 避免了完整体积表示的缺点。

1.3 研究问题分析与本文主要研究内容

我们基于图块和排序的渲染器受到了 Pulsar 实现的快速球体光栅化启发, 鉴于上述分析, 我们还希望排序后的溅射点依然保持传统的 α 混合以继承体积表示的优势, 于是光栅化步骤尊重可见性顺序。此外我们对像素中的所有溅射点进行梯度反向传播并光栅化各向异性的溅射点。基于此, 3D Gaussian Splatting 方法最终呈现了较高的视觉质量。Pulsar 和 ADOP 的渲染速度也激励着我们开发快速的渲染算法。

Neural Point Catacaustics 的漫反射点基渲染路径通过使用 MLP 克服了使用 CNN 进行渲染导致的时间不稳定性，但仍然需要 MVS 几何作为输入。最近的方法不需要 MVS 且使用 SH 表示方向，但他需要掩码进行初始化并且只能处理一个对象的场景。尽管对于小分辨率和低点数的情况此方法速度很快，但我们不知道如何将它扩展到典型的数据集场景^[4]。为了避免 MVS 对几何的需求我们使用更灵活的 3D 高斯表示场景，并由我们为投影高斯开发的基于图块的渲染算法实现了实时渲染。近期有一种在优化过程中采用了点修剪和密化技术的方法，使用点来表示具有径向基函数方法的辐射场，但由于其使用了体积光线步进，无法实现实时显示速率。

3D 高斯已被用于表示捕捉到的人体，最近它们已经与体积光线步进方法一起用于视觉任务，神经体积基元也被随之提出。这些方法专注于重建和渲染单个孤立对象（人体或面部）的特定情况，导致场景的深度复杂性较小。为此我们优化了各向异性协方差和交替密度控制，改进了深度排序渲染，从而使得 3D Gaussian Splatting 方法能够处理完整的复杂场景（包括室内外背景），并且具有较大的深度复杂性。

本文研究的方法由一组静态场景图像作为输入，通过 SfM 得到的相机标定信息生成稀疏点云，并由此创建一组 3D 高斯体元，每个高斯体元均由位置（均值）、协方差矩阵和不透明度（ α ）定义。场景的细节结构被高度各向异性的体积点紧凑地表示，方向性外观（颜色）通过球谐函数（SH）表示。

我们的算法通过交替进行对 3D 高斯体元的参数（即位置、协方差、 α 和 SH 系数）的优化和高斯密度的自适应控制来创建辐射场表示。支持各向异性点的 α 混合，并通过快速排序确保可见性顺序的图块式光栅化算法是保证效率的关键。我们的光栅化器还包括一个快速的反向传播过程，通过跟踪累计的 α 值，实现对所有点的梯度反向传播，无需限制可接收梯度的高斯点数量。如图（1-3）所示。

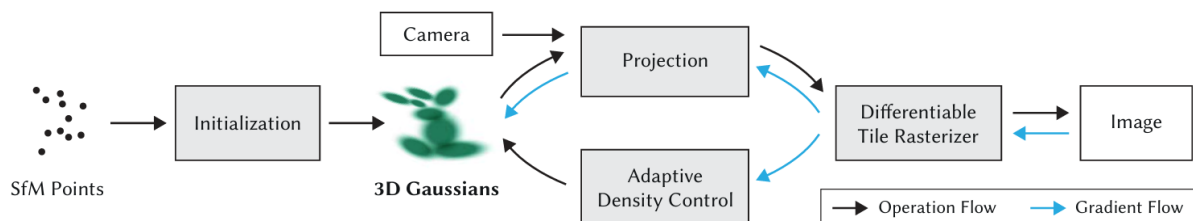


图 1-3 3D Gaussian Splatting 系统技术路线^[5]

1.4 本文的组织架构

本文各章节层层递进，围绕基于 3D 高斯溅射的实时辐射场渲染算法展开研究与阐述，具体内容如下：

第一章，绪论：阐述研究 3D 高斯溅射方法对提升新视角合成性能的意义。梳理传统视角合成算法和基于 NeRF 的视角合成算法的研究现状，分析当前新视角合成任务在显式建模、实时渲染和可导训练方面面临的关键问题，明确本文研究的必要性。

第二章，高效新视角合成的神经渲染理论基础：介绍相机成像原理和坐标系转换关系，以及常见的三维场景表达方法，为理解新视角合成模型提供基础。讲解神经辐射场（NeRF）的定义、引入过程和体积渲染机制。介绍实时渲染技术中 GPU 并行架构的优势和可见性排序等加速方法，为 3D 高斯溅射方法的设计提供技术参考。

第三章，基于 3D 高斯溅射的实时辐射场渲染：介绍 3D 高斯溅射方法的整体框架，说明利用 SfM 技术进行场景初始化和生成高斯点的过程，以及高斯点的基础参数，为后续优化和渲染提供初始条件。介绍高斯点的动态调整策略（Clone 和 Split 操作）和优化目标，旨在提升场景建模的精度和效率，平衡细节保留与效率。介绍基于图块的屏幕空间划分策略和瓦片排序渲染原理，通过将屏幕划分为瓦片来提高高斯点渲染和可见性排序的效率。介绍利用 GPU 并行计算和共享内存，以及采用 Radix Sort 快速排序算法的 GPU 加速技术。说明正向传播中颜色混合与透明处理，以及反向传播中梯度计算与参数更新的渲染流程优化方法，确保高效渲染和模型优化。

第四章，实验与结果分析：介绍选择 Mip - NeRF 360、Tanks and Temples、Deep Blending 等数据集的原因，以及实验所用的硬件环境、软件工具和评价指标。展示在多个数据集上与主流方法对比的视觉效果，以及训练时间、渲染帧率和显存使用等速度与效率方面的核心实验结果，验证 3D 高斯溅射方法的优势。通过消融实验分析初始化策略、克隆与分裂策略、各向异性协方差等因素对模型表现的影响，深入理解该方法的技术结构。

结论：总结本文对 3D 高斯溅射方法的复现、实现和验证工作，指出研究过程中发现的方法存在的不足，如遮挡剔除机制、训练资源开销和对动态场景的适应能力等问题，并对未来研究方向提出拓展建议，如内存优化、遮挡建模等，为后续研究提供参考。

第二章 高效新视角合成的神经渲染理论基础

2.1 新视角合成模型基础

合成相机新位姿处的渲染图像，是新视角合成算法的核心任务。这一过程依赖于场景的准确三维重建，以及模型对不同视角下空间结构与颜色变化规律的建模能力。前文已指出，想要实现高质量的视角合成，关键在于对场景几何、光照及观察机制的理解与建模。因此，本节将重点介绍与视角合成密切相关的基础知识，包括相机的成像模型、内外参的作用机制，以及世界坐标系、相机坐标系与图像平面之间的坐标变换关系。

2.1.1 相机成像原理与坐标系转换

相机的成像过程可以理想化为针孔模型，即小孔成像机制：如图 2-1 所示，外部世界中的光线经由透镜中心投射至图像平面，在二维传感器上形成图像。该过程实际涉及多个坐标系之间的几何映射关系，主要包括世界坐标系、相机坐标系和图像坐标系。

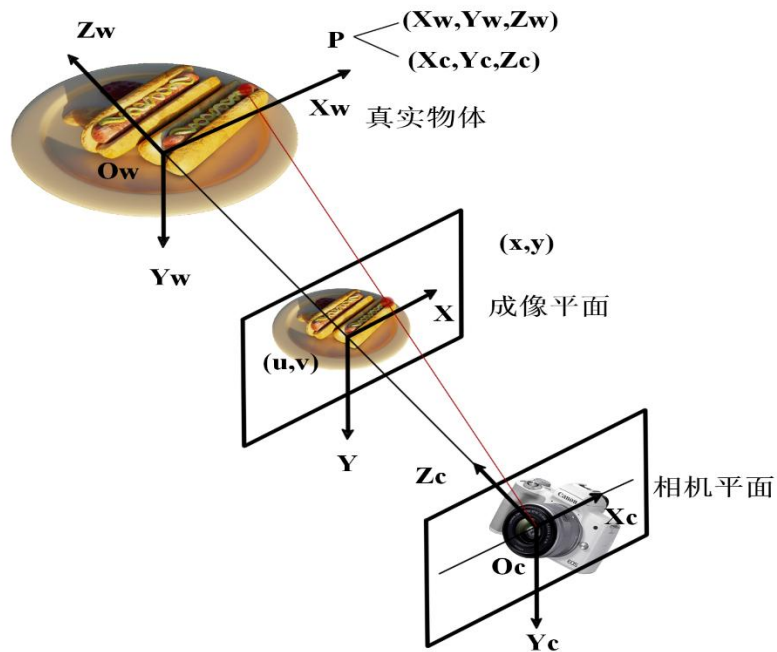


图 2-1 相机成像模型

一、世界坐标系（World Coordinate System）。这是用于描述真实三维空间中物体位置的全局参考坐标系，通常记为 (X_w, Y_w, Z_w) 。场景中所有点的三维坐标最初都以此系为基础表达。

二、相机坐标系 (Camera Coordinate System)。以摄像机镜头的光心 O_c 作为原点, Z 轴指向相机前方的观察方向, 它是图像采样和光线投射的起始参考系, 坐标表示为 (X_c, Y_c, Z_c) 。从世界坐标到相机坐标的变换通过相机的外参矩阵 (也称为相机位姿矩阵) 实现, 其由三维旋转矩阵 R 与平移向量 t 构成的齐次矩阵 $T_{w \rightarrow c}$ 如公式 (2-1) 所示。

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (2-1)$$

三、图像坐标系 (Image Coordinate System)。该坐标系用于描述成像平面上的像素位置, 通常表示为 (x, y) 。其与相机坐标系的映射关系通过相机内参矩阵 K 实现, 包含焦距 (f_x, f_y) 和主点坐标 (u_0, v_0) , 见公式 (2-2)。

$$Z_c \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \quad (2-2)$$

将上述两个变换过程联立, 即可得到从三维空间点 (X_w, Y_w, Z_w) 到图像像素坐标 (x, y) 的投影映射关系, 见公式 (2-3)。

$$Z_c \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K \cdot [R | t] \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (2-3)$$

相机中心、三维空间点与成像点三者共线, 这一几何约束构成了光线追踪或视角合成中射线计算的基础。对每个像素点而言, 其在世界空间中对应一条光线, 射线可表达为: $r(t) = O_c + t \cdot d$ 。其中 O_c 是相机中心位置, d 是方向向量, t 为尺度因子。该表达式在体积渲染等视角合成框架中用于构建从相机出发穿过像素至三维场景的采样路径, 是后续建模和渲染模块的重要输入。

2.1.2 三维场景表达

计算机图形学领域种三维物体建模有多种有效表示方式。如图 2-2 所示, 主流的三维场景表达手段包括体素网格 (Voxel)、点云 (Point Cloud)、多边形网格 (Mesh)、符号距离函数 (Signed Distance Function, SDF) 以及神经辐射场 (Neural Radiance Field, NeRF)。其中, 体素、点云与网格属于显式表示方法, 能够直接提供空间中的几何位置信息; 而 SDF 与 NeRF 属于隐式表示, 依赖特定函数结构间接建模几何及外观特性。


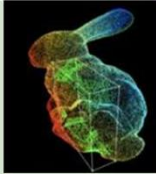
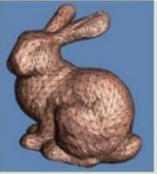
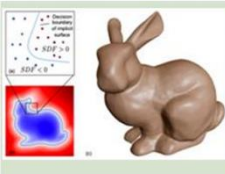
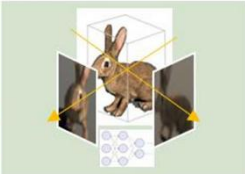
三维表达	体素	点云	网格	符号距离函数	神经辐射场
表示图像					
说明	立方体表达场景	通过大量的点来表达场景	使用顶点和平面表达场景	通过符号距离函数表示场景到表面的距离	通过神经辐射场表示场景每个点的颜色和密度

图 2-2 常见的三维表达方式^[6]

NeRF 作为近年来广泛研究的隐式方法，借助多层感知机（MLP）学习颜色与密度场，具备出色的表达能力。相较之下，3D Gaussian Splatting（3DGS）采用了以点云为核心的显式建模形式，不仅便于构建高精度几何，还天然支持 GPU 并行渲染操作，具备更优的运行效率。

2.2 神经辐射场（NeRF）基本原理

随着深度学习在图像生成与三维重建等任务中的广泛应用，基于神经网络的隐式表示方法在新视角合成领域逐渐取代传统显式建模手段，成为研究的热点。其中，神经辐射场（Neural Radiance Field, NeRF）作为代表性方法之一，首次提出以多层感知机对三维空间中的颜色与密度分布进行建模，并结合体积渲染理论，实现高保真视角合成。本节将从辐射场的基本概念出发，介绍 NeRF 的核心数学建模思想与体积渲染机制，为理解与对比后续的 3D Gaussian Splatting 方法基础。

2.2.1 辐射场的定义与神经辐射场的引入

辐射场（Radiance Field）用于描述三维场景中每个空间位置的光线属性，包括该位置的颜色、光强度、密度分布和其他光学特性，是一种数学模型或表示方法。其通过对场景中各个位置的建模，表达光线在这些位置的传播和与物质的相互作用，从而为图像渲染、光照模拟和场景重建等应用提供基础数据。

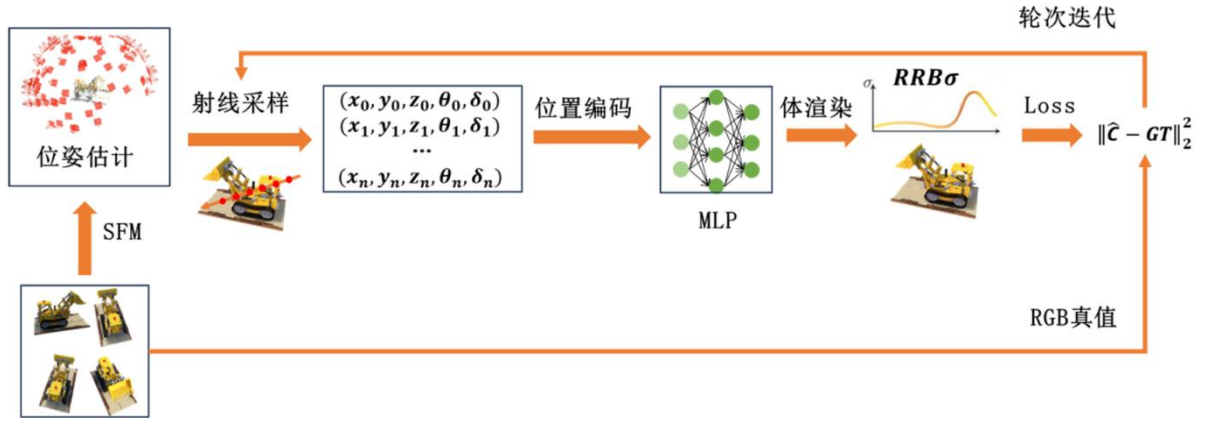


图 2-3 NeRF 算法框架^[6]

神经辐射场（NeRF）^[1]采用了多层感知器（MLP）来模型化场景。图 2-3 呈现了基于神经辐射场（NeRF）的视角合成算法的全面概述。NeRF 首先根据相机和场景的相对位置，利用射线在场景中的路径进行采样，然后通过体积渲染公式来计算每个样本点对最终图像的贡献。NeRF 的核心优势在于，它能够通过位置编码和重要性采样来改进传统的渲染过程，从而实现更高质量的图像合成。

2.2.2 体积渲染机制

体积渲染通过模拟光线在三维介质中传播过程来生成图像，广泛应用于医学成像、科学可视化以及神经渲染等领域。与传统的表面渲染方法不同，体积渲染关注的是整个空间中体素（Voxel）或连续介质的颜色与密度分布，而不仅仅是物体的表面轮廓。

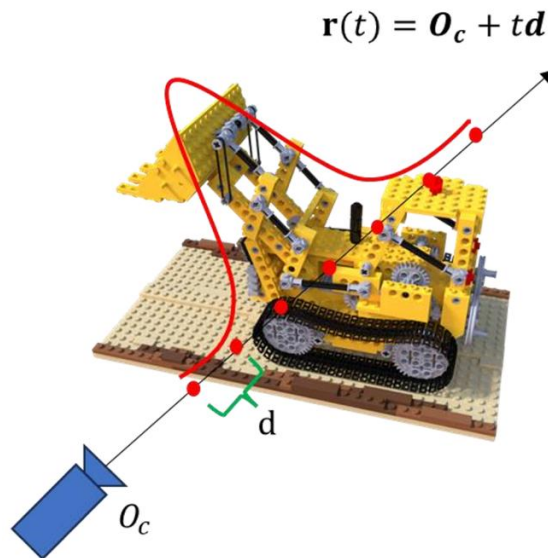


图 2-4 体积渲染^[6]

如图 2-4 所示，一条光线从相机的光心 O_c 出发，沿着方向向量 d 进入三维场景，并以固定步长进行均匀采样。每个采样点的位置由参数 t 控制，表示沿光线方向的距离。根据体积渲染的原理，该过程需对每个采样点处由神经网络估计得到的密度值 σ_i 进行积分，以模拟光线在介质中的衰减和传播。为实现离散化计算，引入透明度累积函数 T_i ，表示从光源起点至第 i 个采样点之间未被遮挡的光线透射率，其计算可由式（2-4）给出：

$$T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j) \quad (2-4)$$

在获取每个采样点的透射率后，便可进一步结合其颜色预测值 c_i ，通过体积渲染公式求出该条光线对应像素点的最终颜色值。该过程综合了各采样点对颜色的贡献，并由公式（2-5）表示：

$$C = \sum_{i=1}^N T_i (1 - \exp(-\sigma_i \delta_i)) c_i \quad (2-5)$$

上式中 σ_i 表示第 i 个样本点的密度， T_i 为第 i 个样本点的透射度（即光线在传播过程中经过的透明度）， c_i 为该样本点的颜色， δ_i 表示每个采样点之间的步长。此公式通过累积每个采样点对光线传输的贡献来计算最终的颜色值。

2.3 实时渲染技术的理论基础

实现高质量图像合成的同时满足实时性需求，对底层硬件架构与图形管线的支持提出了更高要求。本节将从计算平台的角度出发，介绍支持高效神经渲染的 GPU 架构特点，并进一步探讨在图像空间中常用的加速策略与可见性排序方法。

相较于传统中央处理器（CPU），图形处理器（GPU）凭借其高度并行的体系结构，已成为支持大规模神经渲染任务的核心计算平台。

GPU 采用“多核大吞吐”的设计思路，具备大量可同时执行的线程，可在同一时刻并行处理成千上万条计算路径。这一特点尤其适用于图像生成中典型的像素级并行任务，例如射线发射、采样计算、特征查询与颜色累积等操作。在神经辐射场（NeRF）和 3D Gaussian Splatting（3DGS）等方法中，每条射线都需要在三维空间中进行多次采样，并执行密集的数学运算；而每个像素的颜色计算也是独立的，这种强并行性使得 GPU 成为渲染过程的理想计算平台^{[1][5]}。

以 Instant-NGP 为例，该方法通过在 GPU 上实现稀疏体素遍历与多分辨率哈希编码查询，显著减少了射线步进所需的计算量，实现了毫秒级别的图像生成能力^[2]。类

似地，在 3DGS 方法中，所有的高斯体元信息（包括位置、协方差、颜色和密度）都可以在 GPU 上预加载并并行处理，结合光栅化机制完成高效投影、排序与混合，最终实现对 1080p 图像的实时渲染输出^[5]。

现代 GPU 还提供了丰富的并行编程接口，如 CUDA、OpenCL 以及图形渲染管线中通用的 OpenGL 和 Vulkan。这些工具都支持用户直接访问 GPU 内存与线程结构，从而进一步提高了神经图形系统的可扩展性与效率。

第三章 基于 3D 高斯溅射的实时辐射场渲染

3.1 可微的 3D 高斯溅射

由于我们的目标是从一个没有法线的稀疏点云开始, 优化得到一种能够进行高质量的新视角合成的场景表示方法, 我们需要一种继承了可微体积表示特性由具有非结构性和明确性特点的原始数据来实现快速渲染, 于是公式 (3-1) 中可微的 3D 高斯分布成为了优选。

我们的表示方法与先前使用 2D 点的某些方法类似, 去假设每个点是带有法线的小平面圆形。但由于 SfM 点非常稀疏, 估算法线很困难, 所以我们将几何体建模成一组不需要法线的 3D 高斯分布, 每一个 3D 高斯分布都由一个完整的 3D 协方差矩阵 Σ 定义, 这个矩阵在世界坐标系中, 并以均值 μ 为中心。

$$G(x) = e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \quad (3-1)$$

在我们的渲染中, 会先将 3D 高斯投影到 2D 图像平面上, 然后每个高斯分布都会与一个用于混合的不透明系数 α 相乘。具体的, 给定一个视图变换矩阵 W 后, 可以将高斯分布的协方差矩阵 Σ 转换到相机坐标系中, 得到新的协方差矩阵 Σ' , 用于投影后的渲染。

$$\Sigma' = J \Sigma W^T J^T \quad (3-2)$$

在公式 (3-2) 中, J 是投影变换的仿射近似的雅可比矩阵。我们可以选择直接优化协方差矩阵 Σ 获得表示辐射场的 3D 高斯分布, 但是具有物理意义的协方差矩阵需要是半正定的, 使用梯度下降的方法生成有效的协方差矩阵却很难。因此我们选择了另一种更加直观的方式进行优化。3D 高斯的协方差矩阵 Σ 类似于描述一个椭球体的参数。如式 (3-3), 给定一个缩放矩阵 S 和旋转矩阵 R , 我们可以找到相应的 Σ :

$$\Sigma = R S S^T R^T \quad (3-3)$$

为了方便优化, 我们将它们分别存储: 用一个三维向量 s 表示缩放, 一个四元数 q 表示旋转。这两个值转化为各自的矩阵和互相组合都十分方便。



图 3-1 紧凑表示的 3D 高斯^[7]

如图 3-1 所示，这种各向异性协方差的表示使我们能够让 3D 高斯分布在场景中的不同形状的几何结构中都生成一个相对紧凑的表示。

3.2 优化机制

我们反复迭代进行优化：每次先渲染一张图像，然后把渲染结果和数据集中对应的视角图像作比较。我们的优化不仅要能够创建新的几何表示，也要能够删除或移动那些位置错误的几何表示，因为现实场景从三维投影到二维，本身就存在歧义。一些面积较大且均匀的区域，其实可以只用少数的几个大号的各向异性高斯来表示，所以每个 3D 高斯分布的协方差参数的好坏对最终表示的紧凑程度影响非常大。

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{\text{D-SSIM}} \quad (3-4)$$

我们用 sigmoid 激活函数将不透明度 α 的值限制在 $[0,1)$ 范围内，协方差的缩放尺度使用指数激活函数。一开始我们用一个各向同性的高斯模型，将其轴长设置为当前点到最近三个点之间距离的平均值来初始化协方差估计矩阵。如式 (3-4) 所示，损失函数采用了 L1 损失和 D-SSIM 项的结合，并在所有测试中 $\lambda = 0.2$ 。

3.2.1 球谐函数的引入

球谐函数是定义在球面上的一组基函数，一个真实的物体可以使用一组基函数表示其近似的形状，其可视化结果可以参考图 3-2。阶数代表着正交基的数量，越高的基数

代表着越好的拟合效果。球谐函数能够实现紧凑的球面坐标编码，具有良好的旋转不变性、各向异性，能够表达不同观察方向下的颜色特征，在 PlenOctrees 这篇工作中首次被引入视角合成领域。3DGS 沿用了这种编码方式，默认使用 4 阶的球谐函数，总计 16 个球谐系数，而 RGB 三个通道各需要一组球谐基，对于每个 3D 高斯而言就具有 48 个颜色系数。

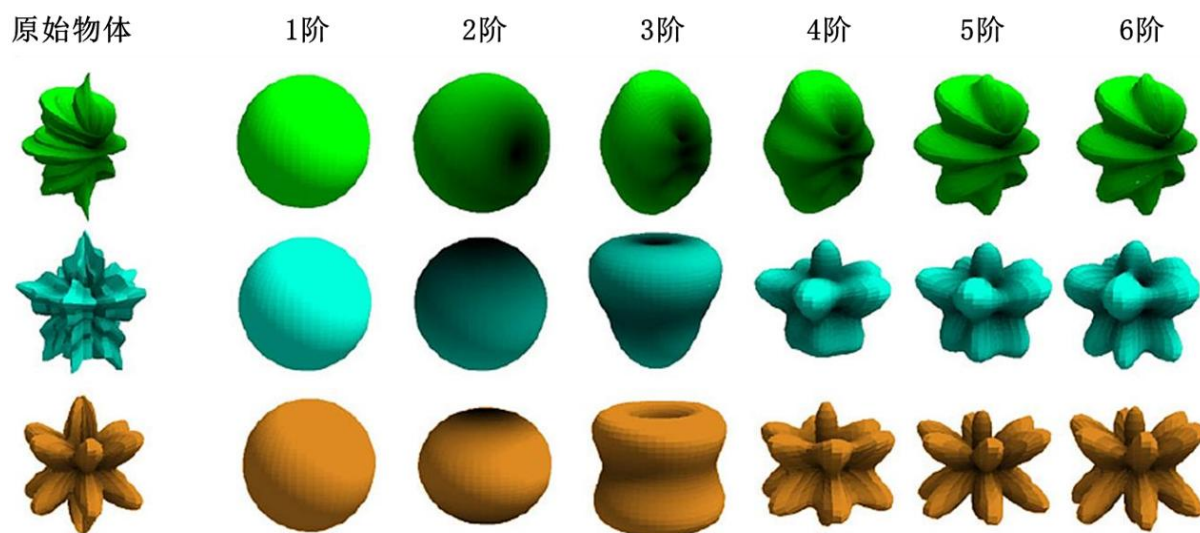


图 3-2 球谐函数示意图

场景空间中任意点在给定角度下观察到颜色都可以用这一组 k 阶的球谐基表示。

3.2.2 高斯的自适应控制

在 SfM 初始化得到一组稀疏高斯点后，我们通过逐步的调整将其变成一组更密集、更能准确表示场景的参数更加合理的高斯点，如图 3-3 所示。

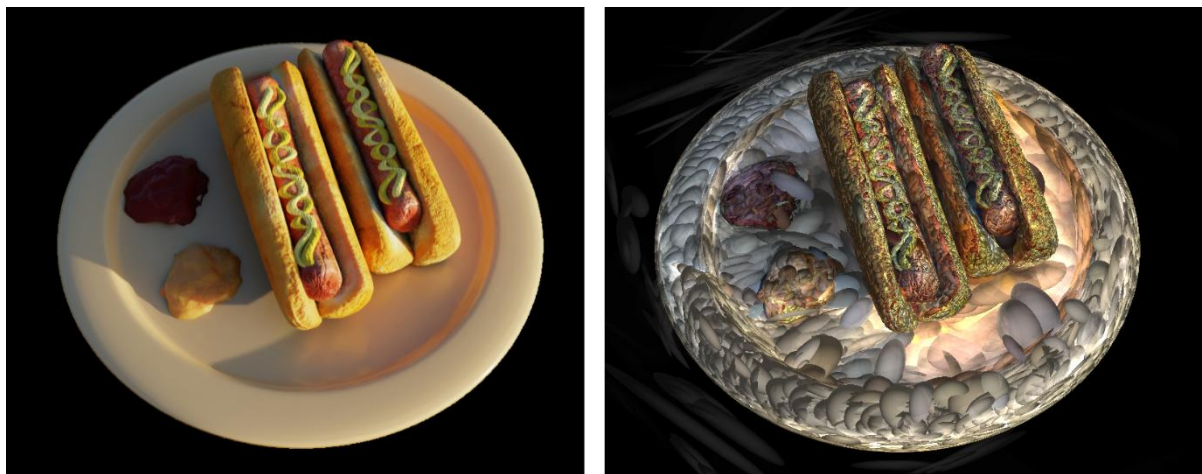


图 3-3 真实训练图像与密集的高斯椭球表面

在优化的初始化阶段完成后我们每隔 100 此迭代就会把透明度 α 小于某个阈值的高斯点删掉。自适应密度控制的过程主要用来填补场景中空缺的区域，主要放在两个地方：一是有几何信息缺失的区域（欠重建），二是一些高斯覆盖了很大一片区域的地方（过重建）。这两种区域在图像空间中位置的梯度都很大，且都适合“加密”。我们为此设置了一个触发加密的梯度阈值 τ_{pos} 。

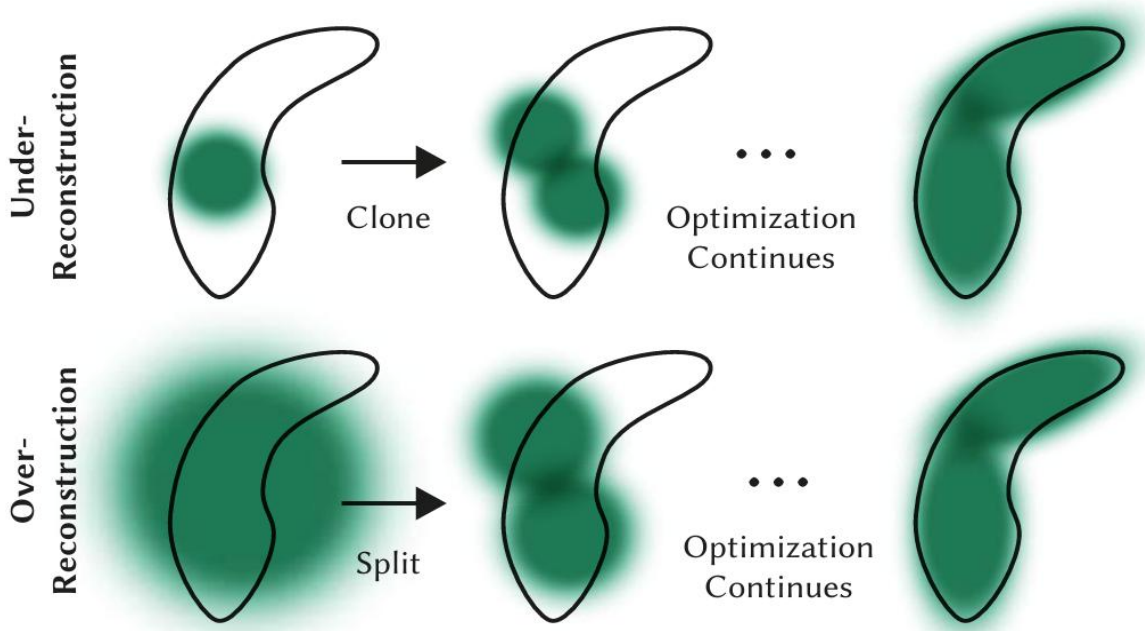


图 3-3 Clone 与 Split 示意图^[5]

如图 3-3 所示，对于小的高斯，我们在出现欠重建的地方需要生成新的几何，于是复制出一个一模一样的高斯，之后沿着梯度方向稍微移动补上空缺；对于大的高斯，如果它位于变化很大的区域我们就将它一分为二，之后缩小尺寸变成两个更小的高斯。我们会把原始高斯当作一个采样概率分布由此初始化两个新高斯的位置，并把它们的尺寸除以一个系数 $\phi = 1.6$ ，这个值是我们实验得出来的。

当然，我们的方法也可能会像其它体积表示方法一样遇到“悬浮点”的问题，一些高斯点一直停在靠近输入相机的位置。我们为此设计了一个机制，即每经过 3000 次迭代就把所有高斯的 α 值暂时清空（设为 0），然后让优化过程自动把“该留下的”高斯重新激活（ α 重新升上来），而不需要的高斯就会被前面提到的删点机制清理掉。高斯点在这个过程中可能会变小、变大，甚至重叠很多，我们也会定期清理那些在世界空间中太大或者在图像中“覆盖面积太大”的高斯点。

而且我们的方法一直保持高斯点是欧几里得空间中的基本元素，不像一些方法^{[4][3]}

那样需要对空间进行压缩、变形,或者做一些特别的投影来处理远处或体积大的高斯点。

3.3 快速可微的光栅化器

我们参考球形神经渲染方法设计了一种基于图块的高斯 `splat` 光栅化器,它可以实现快速的整体渲染和排序以支持近似的 α 混合(包括各向异性 `splats`),并避免以往方法中对可接收梯度的 `splats` 数量的硬性限制。这样的方法一次性对整个图像的原语进行预排序,从而避免了以往每像素排序的开销。

算法首先将屏幕划分为 16×16 的图块,并且仅保留与视锥体相交的 99%置信区间内的高斯。随后实例化所有的高斯,为每个实例赋予一个包括“图块的标识符”和“距离相机视图的深度”的键值,基于这些键值对这些高斯使用快速的 GPU 基数排序算法进行排序。排序结束后我们通过识别第一个和最后一个深度排序条目为每一个图块生成一个列表,这些条目会映射到给定的图块上。

在光栅化过程中我们为每一个图块都启动一个线程块,每个线程块先将高斯包加载到共享内存种,对于给定的像素,随后通过从前到后的遍历列表来累积颜色和 α 值。我们会定期查询图块中的线程,当我们在像素中达到目标的 α 饱和度时,相应的线程停止。

在反向传播过程中,我们必须恢复前向传播中每个像素的完整混合点序列。我们再次遍历每个图块的列表以避免动态内存管理的的开销,可以重用前向传播中的高斯排序数组和图块范围。从后往前的遍历更方便计算梯度,遍历从影响图块中任何像素的最后一个点开始,点加载到共享内存中再次协作地进行。只有当像素的深度小于或等于前向传播中最后一个对其颜色有贡献的点的深度时,才开始代价高昂的重叠测试和点处理。计算梯度需要在原始混合过程中每一步的累积不透明度值。我们通过仅在前向传播结束时存储每个点的最终累积不透明度 α 来恢复这些中间的 α 值。即每个点在前向过程中存储最终的累积不透明度 α ,我们在从后向前遍历时将其除以每个点的 α 就可以获得计算梯度所需的系数。

这样的方法能够处理大量混合高斯,并且在训练和渲染过程中具有较低的内存消耗。

第四章 实验与结果分析

4.1 实验准备

4.1.1 数据集选择

为验证 3D Gaussian Splatting 方法在不同类型场景中的性能表现, 实验选用了三个具有代表性的数据集, 包括真实世界室内外环境、结构多样的城市景观以及合成数据场景。

一、Mip-NeRF 360 数据集为目前神经辐射场方法中的最高质量基准之一, 包含多个 360 度拍摄的户外与室内场景, 覆盖复杂的光照与不规则几何结构, 适合评估方法在无界场景下的泛化能力与细节保留能力。

二、Tanks and Temples 数据集则提供了多视角真实扫描数据, 具有良好的几何结构与光照一致性, 常被用于传统与神经渲染方法的定量对比。本文选用了其中的 Truck 和 Train 场景进行实验。

三、Deep Blending 数据集被用于检验方法在室内小尺度场景下的细节建模与透明过渡能力。该数据集包含多个从真实相机采集的高分辨率场景图像, 视角密集, 纹理与光照信息丰富。由于其本身为神经渲染任务设计, 具备高度配准的图像与准确相机参数, 能够充分发挥 3DGS 所依赖的高斯空间表达优势。

4.1.2 硬件环境与软件工具

本研究在 AutoDL 平台上完成了 3D Gaussian Splatting 方法的复现与测试, 所使用的云服务器配置如下:

- 主机环境: AutoDL 内蒙 B 区 / 198 机
- 操作系统镜像: graphdeco-inria/gaussian-splatting/3D-gaussian-splatting-quick_start:v1;
- GPU: NVIDIA RTX 4090 24GB (1 卡);
- CPU: 16 核 Intel (R) Xeon (R) Gold 6430 虚拟处理器;
- 内存: 120 GB;

- 硬盘：系统盘 30 GB，数据盘共 450 GB（50 GB 免费 + 400 GB 付费）；
- CUDA 与驱动版本：CUDA ≤ 12.6 ，NVIDIA 驱动版本 560.35.03。

在软件方面，3DGS 原始实现采用 Python 与 PyTorch 框架构建，同时使用了自定义的 CUDA 内核完成关键的光栅化任务，并集成了开源的 SIBR 交互式可视化工具用于结果展示与评估，排序操作采用了 NVIDIA CUB 库中高度优化的 Radix Sort 算法。

4.1.3 评价指标

本文采用了当前神经图形学领域中常用的三种图像质量评价指标：峰值信噪比（PSNR）、结构相似性指数（SSIM）以及感知图像相似度指标（LPIPS）。其中，PSNR 主要用于度量渲染图像与真实图像在像素级的误差，数值越高表示重建质量越高。SSIM 强调图像在亮度、对比度和结构等维度上的一致性，其值越接近于 1，说明图像的整体结构越接近真实图像。LPIPS 指标则是以深度神经网络作为感知模型，评估两幅图像之间的视觉感知差异，值越小表示图像越相似，越接近人眼主观判断。

本实验严格遵循原论文中的评估协议，即使用与 Mip-NeRF 360 相同的视角划分方法，将每第八帧图像作为测试集，并在测试视角下分别计算三项指标的平均值作为模型性能的最终评价依据。

4.2 核心实验结果

图 4-1 展示了我们的方法与先前方法的比较，以及来自保留测试视图的相应地面真实图像。场景从上到下依次为：来自 Mip-NeRF360 数据集的自行车、花园、树桩、柜台和房间；来自 Deep Blending 数据集 [Hedman et al. 2018] 的 Playroom 和 DrJohnson；以及来自 Tanks&Temples 的卡车和火车。箭头/插图突出显示了质量上不明显的差异。

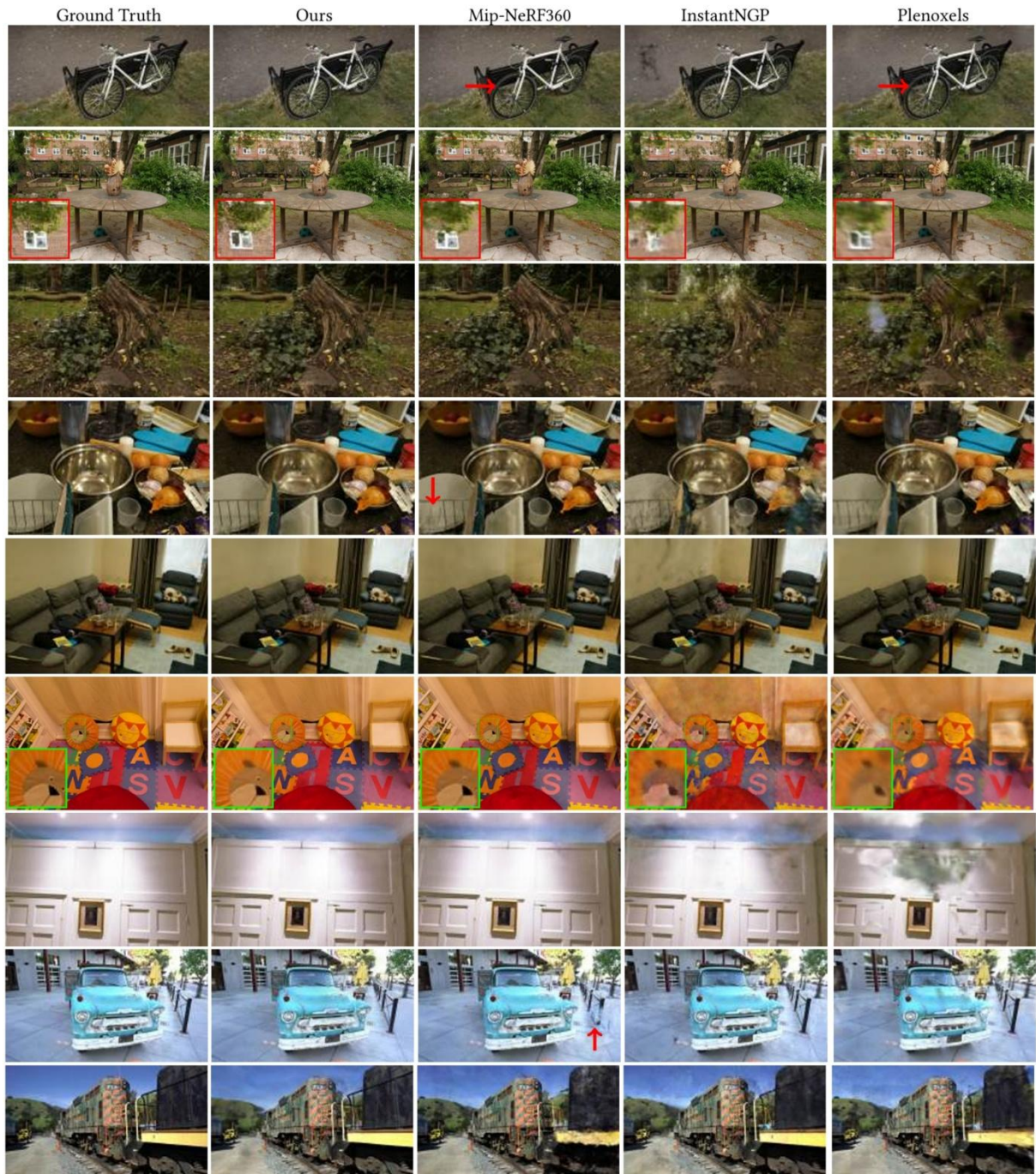


图 4-1 实验视觉效果对比^[7]

4.2.1 视觉效果对比

为评估 3D Gaussian Splatting 方法在新视角图像合成中的视觉表现，本文在多个公开数据集上与当前主流神经辐射场方法进行了对比，包括 Plenoxels、Instant-NGP（基本版与扩展版）、Mip-NeRF 360 以及 Point-NeRF 等。对比指标包括 SSIM、PSNR、LPIPS 三类图像质量指标，并结合训练时间、渲染帧率（FPS）与显存使用，展示了模型在视觉质量与效率之间的平衡能力。

在 Mip-NeRF 360 数据集上，Ours-30K 实现了 PSNR 27.21、SSIM 0.815 和 LPIPS 0.214 的表现，仅略低于 Mip-NeRF360（PSNR 27.96），但训练时间仅需 41 分钟，推理帧率高达 134 fps，而后者需耗时 48 小时训练，推理帧率仅为 0.06 fps，表明本方法在大幅提升渲染效率的同时，依然保持了极具竞争力的视觉质量（见表 4-1）

表 4-1 Mip-NeRF360 数据集下各模型的表现^[7]

	SSIM	PSNR	LPIPS	Train	FPS	Mem
Plenoxels	0.626	23.08	0.463	25m49s	6.79	2.1GB
INGP-Base	0.671	25.30	0.371	5m37s	11.7	13MB
INGP-Big	0.699	25.59	0.331	7m30s	9.43	48MB
M-NeRF360	0.792	27.96	0.237	48h	0.06	8.6MB
Ours-7K	0.770	25.60	0.279	6m25s	160	523MB
Ours-30K	0.815	27.21	0.214	41m33s	134	734MB

在 Tanks and Temples 数据集上，Ours-30K 的表现同样优异，以 PSNR 23.14、SSIM 0.841 和 LPIPS 0.183 显著优于 Mip-NeRF360 和 Instant-NGP 等方法，同时实现 154 fps 的渲染速度，充分体现出方法在结构复杂的户外场景中仍具有良好的细节恢复与性能优势（见表 4-2）。

表 4-2 Tanks&Temples 数据集下各模型的表现^[7]

	SSIM	PSNR	LPIPS	Train	FPS	Mem
Plenoxels	0.719	21.08	0.379	25m5s	13.0	2.3GB
INGP-Base	0.723	21.72	0.330	5m36s	17.1	13MB
INGP-Big	0.745	21.92	0.305	6m59s	14.4	48MB
M-NeRF360	0.759	22.22	0.257	48h	0.14	8.6MB
Ours-7K	0.767	21.20	0.280	6m55s	197	270MB
Ours-30K	0.841	23.14	0.183	26m54s	154	411MB

在 Deep Blending 数据集上, Ours-30K 的综合表现甚至略优于 Mip-NeRF360, 以 PSNR 29.41、SSIM 0.903 和 LPIPS 0.243 与其持平甚至稍优, 训练时间缩短至 36 分钟, 渲染速度达 137 fps。相比之下, Plenoxels 训练耗时接近半小时却仅获得 PSNR 23.06, LPIPS 达到 0.510, 视觉质量明显逊色 (见表 4-3)

表 4-3 Deep Blending 数据集下各模型的表现^[7]

	SSIM	PSNR	LPIPS	Train	FPS	Mem
Plenoxels	0.795	23.06	0.510	27m49s	11.2	2.7GB
INGP-Base	0.797	23.62	0.423	6m31s	3.26	13MB
INGP-Big	0.817	24.96	0.390	8m	2.79	48MB
M-NeRF360	0.901	29.40	0.245	48h	0.09	8.6MB
Ours-7K	0.875	27.78	0.317	4m35s	172	386MB
Ours-30K	0.903	29.41	0.243	36m2s	137	676MB

最后, 在 NeRF 合成数据集的多个标准场景中, Ours-30K 也实现了稳定的高质量表现, 平均 PSNR 达 33.32, 略高于 Mip-NeRF (33.09) 与 Instant-NGP (33.18), 优于 Plenoxels (31.76), 并在 Hotdog、Lego、Ficus 等细节复杂场景中表现突出 (见表 4-4)。

表 4-4 在合成 NeRF 数据集上的 PSNR 分数^[7]

	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	29.78s	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

4.2.2 速度与效率提升

在训练时间方面, 3DGS 利用稀疏点初始化与密度自适应控制策略, 显著减少了冗余计算, 仅需 7 分钟即可达到 PSNR 为 23.6 dB 的结果; 延长训练至 51 分钟时, PSNR 可达 25.2 dB, 超越 Mip-NeRF360 所用的 48 小时训练结果。

在实时渲染效率上, 得益于高斯体素的显式表达和基于 GPU 的前向光栅化流程,

3DGS 在 1080p 分辨率下可稳定实现 ≥ 135 fps 的实时显示性能。而作为对比，Plenoxels 和 Instant-NGP 在相同分辨率下分别仅能达到 8.2 fps 与 9.2 fps，而 Mip-NeRF360 更是远低于实时标准，仅为 0.071 fps。

通过对比本文方法还展现出良好的内存管理特性。在不同规模下训练的模型中，7K 迭代版本仅使用 523MB 显存即可实现流畅渲染；而扩展至 30K 迭代时，虽然模型容量上升至 734MB，但仍在常规消费级 GPU 处理能力范围内，相比 Mip-NeRF360 所需复杂多级 MLP 网络结构显著节省内存资源。

4.3 消融实验分析

本文在 Garden、Bicycle、Trunk 等典型场景下，分别在 5K 和 30K 迭代次数设置下进行系统性消融实验，并对不同配置下模型输出图像的 PSNR 进行统计（见表 4-5 至表 4-7）。消融因素包括初始化策略、密度调控机制（Clone 与 Split）、颜色建模方式（SH）、协方差表示方法等，下面将分别进行分析。

表 4-5 5K 下消融实验的 PSNR 分数^[7]

	Trunk-5K	Garden-5K	Bicycle-5K
Limited-BW	14.66	22.07	20.77
Random Init	16.75	20.90	19.86
No-Split	18.31	23.98	22.21
No-SH	22.36	25.22	22.88
No-Clone	22.29	25.61	22.15
Isotropic	22.40	25.49	22.81
Full	22.71	25.82	23.18

表 4-6 30K 下消融实验的 PSNR 分数^[7]

	Trunk-30K	Garden-30K	Bicycle-30K
Limited-BW	13.84	22.88	20.87
Random Init	18.02	22.19	21.05
No-Split	20.59	26.11	25.02
No-SH	24.39	26.59	25.08
No-Clone	24.82	27.47	25.46
Isotropic	23.89	27.00	24.81
Full	24.81	27.70	25.65

表 4-7 消融实验的 averages PSNR 分数^[7]

	Trunk-30K	Bicycle-30K
Limited-BW	19.16	19.19
Random Init	19.17	20.42
No-Split	21.50	23.90
No-SH	23.48	25.35
No-Clone	23.35	25.91
Isotropic	23.56	25.23
Full	23.90	26.05

4.3.1 初始化策略对模型表现的影响

我们对比了使用结构光束法（SfM）初始化与随机初始化（Random Init）的模型性能差异。结果显示，随机初始化会显著降低训练起点的几何一致性与密度分布准确性，导致模型初期难以收敛，最终精度也明显下降。例如在 Garden-30K 场景下，随机初始化方案的 PSNR 仅为 22.19，而使用 SfM 初始化的完整模型则达到 27.70，提升超过 5.5 dB（见表 5-6）。此外，随机初始化还易导致背景伪影（floaters）产生，降低整体画面一致性。因此可以认为，稀疏结构先验对于点集建模的收敛性和几何约束具有重要意义。

4.3.2 克隆与分裂策略的对比

为了检验密度自适应模块中 Clone（克隆）与 Split（分裂）操作的实际作用，本文分别移除两者，构造 “No-Clone” 与 “No-Split” 模型进行比较。实验结果表明，分裂操作对高斯体元覆盖能力具有关键意义，禁用分裂会导致渲染区域中细节缺失严重。如在 Trunk-30K 场景下，No-Split 模型仅实现 PSNR 20.59，相较完整模型下降了 4.2 dB。而克隆操作主要用于增强细长结构与边缘区域的几何密度，对画面细节保留尤为关键。例如在 Bicycle-30K 场景中，No-Clone 模型比完整版本低 0.59 dB（见表 5-6）。两者结合使用时可实现最高保真度，说明 Clone 与 Split 是相辅相成的密度调整机制，缺一不可。

4.3.3 各向异性协方差的重要性

为了评估协方差矩阵中各向异性建模（Anisotropic Covariance）的效果，实验中引入了 “Isotropic” 版本，即仅对高斯体元的三个轴采用统一缩放，不再单独优化协方差主轴方向。实验表明，去除各向异性后模型的拟合能力明显减弱。在 Garden-30K 场景下，Isotropic 模型 PSNR 为 27.00，低于完整版本的 27.70；而在 Trunk-30K 场景中，下降幅度更大（23.89 vs 24.81），说明在不规则场景中，各向异性有助于提升贴合性和空间覆盖精度。此外，该模型也在视觉上表现出更模糊的边缘和弱纹理区域，印证了协方差主轴优化对于建模复杂几何结构的重要性。

结论

文章研究基于 3D 高斯溅射的实时辐射场渲染算法的设计与实现，对当前神经图形学领域的代表性方法——3D Gaussian Splatting 进行复现并实现训练-渲染-评估，验证其在真实场景与合成数据集中的应用效果。

论文先梳理了包括体积渲染原理、光线投射机制、GPU 并行渲染等核心知识在内的神经辐射场的基础理论，而后基于原始论文内容搭建了涵盖三维高斯点的初始化、密度优化、自适应调整与可见性感知渲染等关键模块的实现框架，并在 AutoDL 云平台上完成了实际部署与测试，成功在多个公开数据集上复现了 3DGS 方法的图像质量和渲染性能，完成了对比实验与消融分析。

综上所述，本文在完成核心算法复现的基础上，构建了完整的神经渲染流程，验证了 3D Gaussian Splatting 在高质量新视角合成中的性能表现，并对其实现细节与系统瓶颈进行了深入分析。该研究不仅提升了作者对神经图形渲染技术的理解与实现能力，也为今后从事相关研究与工程实践打下了良好基础。

参考文献

- [1] MILDENHALL B, SRINIVASAN P P, TANCIK M, BARRON J T, RAMAMOORTHY R, NG R. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis[J]. Communications of the ACM, 2021, 65(1): 99-106.
- [2] MULLER S, EVANS A, SCHIED C, et al. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding[EB/OL]. arXiv, 2022-01-16[2025-05-11].
- [3] Fridovich-Keil S, Yu A, Tancik M, Chen Q, Recht B, Kanazawa A. Plenoxels: Radiance Fields without Neural Networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New York: IEEE, 2022: 5491–5500.
- [4] BARRON J T, MILDENHALL B, SARLIN P, et al. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, LA, USA: IEEE, 2022: 5470–5479.
- [5] KERBL B, KOPANAS G, LEIMKÜHLER T, et al. 3D Gaussian Splatting for Real-Time Radiance Field Rendering[C]//ACM SIGGRAPH 2023 Conference Proceedings. New York: ACM, 2023: 1-15.
- [6] 余隆吉. 基于 3D 高斯溅射的低空遥感图像视角合成算法研究[D]. 武汉: 华中科技大学, 2024.
- [7] GROSS M, PFISTER H. Point-Based Graphics[M]. Amsterdam: Elsevier, 2011.
- [8] BOTSCH M, HORNUNG A, ZWICKER M, et al. High-Quality Surface Splatting on Today's GPUs[C]// Proceedings of the Second Eurographics/IEEE VGTC Conference on Point-Based Graphics. Aire-la-Ville: Eurographics Association, 2005: 17–24.
- [9] MERRILL D, GRIMSHAW A. Revisiting Sorting for GPGPU Stream Architectures[C]//Proceedings of the 19th International Conference on Parallel Architectures and Compilation Techniques (PACT). New York: IEEE, 2010: 545–546.
- [10] CHEN A, XU Z, WANG X, et al. TensorRF: Tensorial Radiance Fields [EB/OL] . [2022-11-29] [2025-05-11]. Available: <https://arxiv.org/abs/2203.09517>. DOI: 10.48550/arXiv.2203.09517.
- [11] WANG C, ZUO X, WANG S, et al. F2-NeRF: Fast Neural Radiance Field Training with Free Camera Trajectories[J]. arXiv preprint arXiv:2303.13030, 2023.
- [12] CHEN A, XU Z, ZHAO G, et al. MobileNeRF: Exploiting the Polygon Rasterization Pipeline for Efficient Neural Field Rendering on Mobile Architectures[J]. arXiv preprint arXiv:2208.00277, 2022.
- [13] TANCIK M, MILDENHALL B, FRIDOVICH-KEIL S, et al. Block-NeRF: Scalable Large Scene

Neural View Synthesis[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New York: IEEE, 2022: 8248–8258.

[14] XIANGLI Y, LIN Y, DAI B, et al. BungeeNeRF: Progressive Neural Radiance Field for Extreme Multi-scale Scene Rendering[J]. arXiv preprint arXiv:2212.05735, 2022.

致谢

感谢指导老师耐心与包容，感谢父母的支持与陪伴，论文工作多有不足，未来希望做得更好。