

中国科学技术大学



《3D Gaussian Splatting之综述》

暑期课程“计算机图形学前沿”读书报告

题 目 : 3D Gaussian Splatting之综述

上课时间 : 2024.7.8-2024.7.12

授课教师 : 刘利刚 等

姓 名 : 刘舟洋

学 号 : PB21000231

组 别 : 无

日 期 : 2024.8.12

3D Gaussian Splatting之综述

刘舟洋，PB21000231

中国科学技术大学少年班学院

摘要：3DGS技术最近在显式辐射场和计算机图形学领域中崭露头角。与主要使用隐式、基于坐标的模型来将空间坐标映射到像素值的主流神经辐射场方法不同，这种创新的方法利用数百万个可学习的三维高斯，代表着一个重要的转变。它能够通过高效的训练，将多视图图像转化为显式的3D高斯表示，并实现新视角的实时渲染。3DGS不仅具备实时渲染的能力，还引入了前所未有的编辑性，这使其成为下一代三维重建和表示的潜在变革者。本文对3DGS领域的最新发展和关键贡献进行了系统的概述。本文深入探讨了3DGS的基本原理和其兴起的驱动因素，为理解其重要性奠定基础。

关键词：3DGS；显式辐射场；实时渲染；图像识别；

A Overview of 3D Gaussian Splatting

Zhouyang Liu

University of Science and Technology of China

Abstract: 3D Gaussian Splatting has recently emerged as a transformative technique in the realm of explicit radiance field and computer graphics. This innovative approach, characterized by the utilization of millions of learnable 3D Gaussians, represents a significant departure from mainstream neural radiance field approaches, which predominantly use implicit, coordinate-based models to map spatial coordinates to pixel values. It can effectively transform multi-view images into explicit 3D Gaussian representations through efficient training, and achieve real-time rendering of novel views. 3DGS, with its explicit scene representation and differentiable rendering algorithm, not only promises real-time rendering capability but also introduces unprecedented levels of editability. This positions 3DGS as a potential game-changer for the next generation of 3D reconstruction and representation. In the present paper, I provide a systematic overview of the recent developments and critical contributions in the domain of 3D GS. I begin with a detailed exploration of the underlying principles and the driving forces behind the emergence of 3D GS, laying the groundwork for understanding its significance.

Key Words: 3D Gaussian Splatting, Explicit Radiance Field, Real-time Rendering, Scene Understanding;

1 引言

3D重建的历程早在深度学习兴起之前就已经开始，早期的努力主要集中在光场和基本的场景重建方法上。然而，这些早期尝试受限于对密集采样和结构化捕捉的依赖，在处理复杂场景和光照条件时面临显著挑战。随着SfM (Structure-from-Motion, 运动恢复结构) 和后续的MVS (Multi-view Stereo, 多视图立体) 算法的发展，3D重建有了更为稳固的框架。尽管取得了这些进展，这些方法在新视角合成方面仍然存在困难，并且难以与深度场景理解模型兼容。

NeRF (Neural Radiance Fields, 神经辐射场) 代表了这一进展中的一个飞跃。通过利用深度神经网络, NeRF实现了从空间坐标到颜色和密度的直接映射。NeRF的成功在于其能够创建连续的体积场景函数, 产生了前所未有的细节和逼真度。然而, 正如任何新兴技术一样, 这一实现也带来了成本: i) 计算强度。基于NeRF的方法计算强度高, 通常需要大量的训练时间和渲染资源, 特别是在处理高分辨率输出时。尽管已经有大量的研究工作试图加速这一过程, 以便在普通设备上实现其应用, 但要在消费级GPU上在1小时内快速训练NeRF并在手机和笔记本电脑等常见设备上以可交互帧率 (约30FPS) 渲染3D场景, 依然非常困难。ii) 可编辑性。对隐式表示的场景进行操作具有挑战性, 因为直接修改神经网络的权重与几何或外观属性的变化没有直观的联系。

在这种背景下, 3DGS (3D Gaussian Splatting) 应运而生, 它不仅仅是一个逐步的改进, 而是一次根本性的突破, 重新定义了场景重建和渲染的边界。虽然NeRF在创建逼真图像这一方面表现出色, 但对于某些对延迟高度敏感的应用 (如虚拟现实和自动驾驶) 来说, 更快、更高效的渲染方法的需求越来越高。3DGS通过引入一种先进的显式场景表示, 利用数百万个可学习的3D高斯来建模场景, 解决了这一需求。与基于坐标的隐式模型不同, 3DGS采用显式表示和高度并行的工作流程, 促进了更高效的计算和渲染。3DGS的创新在于其将可微管线与基于点的渲染技术的优点结合在一起。通过使用可学习的3D高斯来表示场景, 它保留了连续体积辐射场的强大拟合能力, 这对于高质量的图像合成至关重要, 同时避免了与NeRF方法相关的计算开销 (如计算量大的光线行进和空白空间中的不必要计算)。

在学习3DGS技术的过程中, 我阅读了相关文献, 并撰写了这份读书报告, 来总结我所学到的内容。3DGS是一项非常新的技术, 因此我特别关注了其基本原理, 以及自其引入以来在各个领域的多样化发展和应用。这份报告系统地整理了该领域的重要文献, 涵盖了3DGS的理论基础、显著的技术进展以及新兴的应用前景。虽然3DGS还处于初期且快速发展的阶段, 这份报告力求提供一个反映其当前状态和未来潜力的平衡视角。通过这份报告, 我希望能够梳理出该领域的主要研究趋势, 为进一步的学习和研究奠定基础。

2 背景

在本节中, 首先简要介绍了辐射场 (Radiance Fields) 的概念 (见第2.1节), 这是场景渲染中的一个关键概念。它概述了两种主要的辐射场表示形式: 隐式表示, 如NeRF, 它使用神经网络进行直接但计算量大的渲染; 以及显式表示, 如网格, 它采用离散结构以便更快速地访问, 但代价是更高的内存使用。第2.2节进一步建立了与相关领域 (如场景重建和渲染) 的联系。

2.1 辐射场

辐射场是3D空间中光分布的表达, 捕捉了环境中光与表面和材料的交互。神经场可表达为函数 $L : \mathbb{R}^5 \rightarrow \mathbb{R}^+$, 其中 $L(x, y, z, \theta, \phi)$ 将点 (x, y, z) 和球坐标下的方向 (θ, ϕ) 映射为非负辐射值。辐射场分为显示表示和隐式表示。

2.1.1 隐式辐射场

隐式辐射场在表示场景中的光分布时, 不需要显式地定义场景的几何结构。在深度学习时代, 它通常使用神经网络来学习一个连续的体积场景表示。在NeRF中, 通常使用多层感知器 (MLP) 这样的神经网络, 将一组空间坐标 (x, y, z) 和视角方向 (θ, ϕ) 映射到颜色和密度值。任何点处的辐射值并未显式存储, 而是在查询MLP时实时计算。因此, 该函数可以表示为:

$$L_{implicit}(x, y, z, \theta, \phi) = MLP(x, y, z, \theta, \phi) \quad (1)$$

这种格式允许对复杂场景进行可微分且紧凑的表示, 但往往需要付出较高的计算代价, 因为需要进行体积光线行进 (Volumetric Ray-Marching)。

2.1.2 显式辐射场

相比之下，显式辐射场直接以离散的空间结构（如体素网格或点集）表示光的分布。这种结构中的每个元素都存储其相应空间位置的辐射信息。这种方法允许更直接且通常更快速地访问辐射数据，但代价是更高的内存使用和潜在的较低分辨率。显式辐射场表示的一般形式可以表示为：

$$L_{\text{explicit}}(x, y, z, \theta, \phi) = \text{DataStructure}[(x, y, z)] \cdot f(\theta, \phi) \quad (2)$$

其中，DataStructure 可以是体积、点云等格式，而 $f(\theta, \phi)$ 是一个基于视角方向修改辐射的函数。

2.1.3 3DGS技术：结合两者优势

3DGS是一种显式辐射场技术，同时具有隐式辐射场的优势。具体来说，它通过利用可学习的3D高斯函数，提供了一种灵活且高效的表示方式。这些高斯函数在多视图图像的监督下进行优化，以准确表示场景。基于3D高斯的可微管线结合了神经网络优化的优点和显式、结构化的数据存储。这种混合方法旨在实现实时的高质量渲染，特别是在处理复杂场景和高分辨率输出时，所需的训练时间也更少。3D高斯表示的公式如下：

$$L_{\text{3DGS}}(x, y, z, \theta, \phi) = \sum_i G(x, y, z, \mu_i, \Sigma_i) \cdot c_i(\theta, \phi) \quad (3)$$

其中， G 是具有均值 μ_i 和协方差 Σ_i 的高斯函数， c 代表视角依赖的颜色。

2.2 背景和术语

有几种技术和研究领域与3DGS密切相关，以下部分将简要描述这些内容，以便更清晰地理解。

2.2.1 场景重建与渲染

粗略地说，场景重建（Scene Reconstruction）涉及从一系列图像或其他数据中创建一个场景的3D模型。渲染（Rendering）是一个更具体的术语，专注于将计算机可读的信息（例如场景中的3D对象）转换为基于像素的图像。早期的技术基于光场生成逼真的图像。SfM（Structure-from-Motion，运动恢复结构）和MVS（Multi-view Stereo，多视图立体）算法进一步推进了这一领域，通过图像序列估算3D结构。这些历史方法为更复杂的场景重建和渲染技术提供了坚实的基础。

2.2.2 神经渲染与辐射场

神经渲染（Neural Rendering）将深度学习与传统图形技术结合，以创建逼真的图像。早期的尝试使用卷积网络来估算混合权重或纹理空间解决方案。如第2.1.1节所述，辐射场（Radiance Fields）表示一个函数，描述了光线在空间中每个方向传播的数量。NeRF使用神经网络，通常是多层次感知器（MLP），来建模辐射场，从而实现细致且逼真的场景渲染。

2.2.3 体积表示与光线行进

体积表示（Volumetric Representations）不仅将物体和场景建模为Surface，还建模为充满材料或空白空间的Volume。这允许更准确地渲染雾、烟雾或半透明材料等现象。光线行进（Ray-Marching）是一种与体积表示结合使用的技术，通过追踪光线在Volume中的路径来渲染图像。NeRF采用了与体积光线行进相同的理念，并引入了重要性采样（Importance Sampling）和位置编码（Positional Encoding），以提高合成图像的质量。尽管提供了高质量的结果，但体积光线行进的计算量很大。这促使人们寻找更高效的方法，例如3DGS。

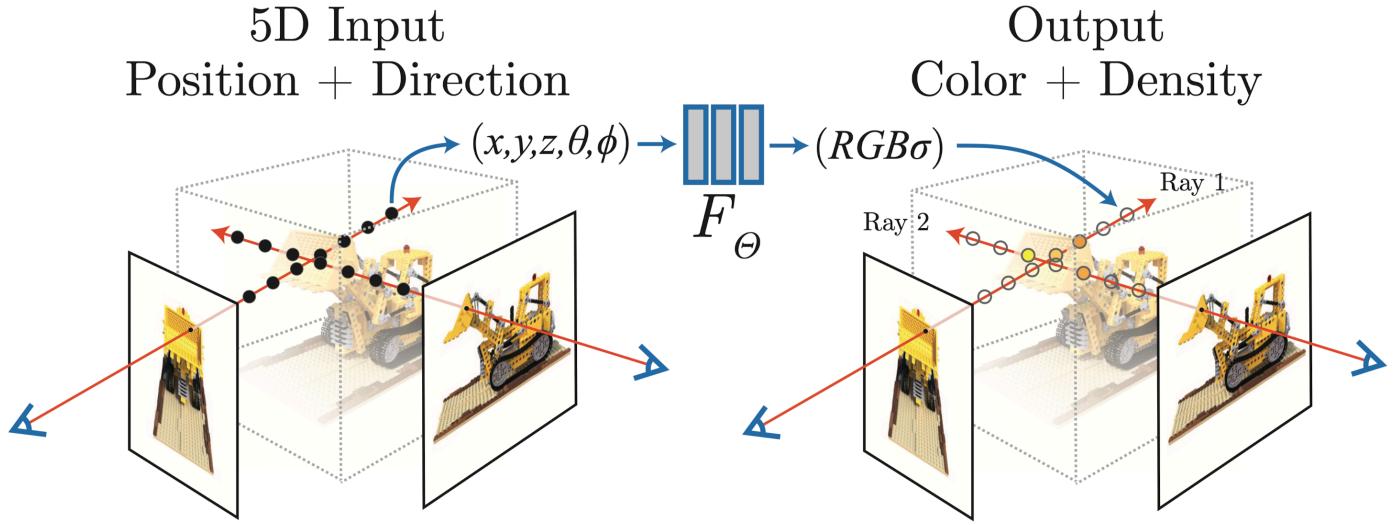


图 1 NeRF体渲染过程^[1]

2.2.4 基于点的渲染

基于点的渲染 (Point-based Rendering) 是一种使用点而不是传统多边形来可视化3D场景的技术。这种方法在渲染复杂的、非结构化的或稀疏的几何数据时特别有效。点可以通过额外的属性（如可学习的神经描述符）进行增强，并且可以高效地渲染。但这种方法也存在一些问题，如渲染中的孔洞或锯齿效应。3DGS通过使用各向异性高斯函数扩展了这一概念，使场景的表示更加连续和一致。更多实现细节将在第3节中进一步讨论。

3 3DGS

3DGS 在无需依赖深度神经网络的情况下，实现了实时高分辨率图像渲染的突破。本节旨在提供3DGS的基本见解。首先将在第3.1节详细解释3DGS如何在给定构建良好的3D高斯函数的情况下合成图像，即3DGS的前向过程。接着，第3.2节将介绍如何为给定的场景获得构建良好的3D高斯函数，即3DGS的优化过程。

3.1 3DGS渲染方法

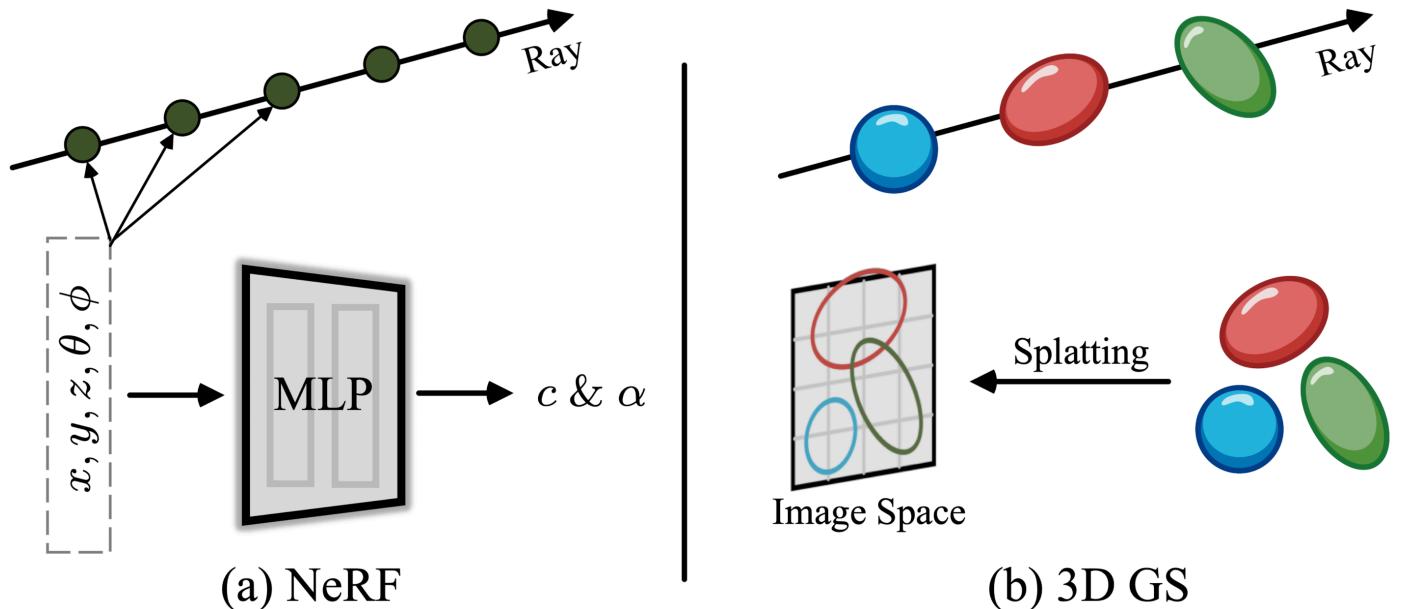


图 2 NeRF与3DGS对比^[3]

考虑一个由（数百万个）优化后的3D高斯函数表示的场景。目标是从指定的摄像机视角生成一幅图像。NeRF的渲染方式是体渲染，神经网络输出Camera Ray上不同采样点的RGB和体密度后，对该Camera Ray进行一个Accumulate，最终得到2D图像；3DGS在会过程中将建模好的3D高斯进行光栅化处理，将其投影到2D图像中，这个过程可以理解为向一个平地抛雪球，这个雪球被染上了不同的颜色，雪球落地之后会溅开，产生不同颜色的痕迹，多个雪球（3D高斯）丢到平地上之后，把不同的颜色痕迹按照深度等进行混合，就得到了最终的图像，这个渲染过程就叫做Splatting，这里的渲染技术是 α -blending。如图2所示，NeRF和3DGS的渲染过程可以被视为彼此的逆过程。表1从六个不同维度对比了NeRF与3DGS：

表 1 NeRF与3DGS对比

	NeRF	3DGS
Input	Camera Pose(5D)	Image Point Clouds from SfM
Output	Image	Image
Rendering	Volume Rendering	α -blending
Optimization	Gradient Descend	Gradient Descend
3D Information Expression	Implicit	Explicit
Time&Effect	12hour for Isolated Object	6min for Isolated Object

下面是一些关于3DGS的名词：

3.1.1 3D高斯

3D高斯由其中心位置 μ 、不透明度 α 、3D协方差矩阵 Σ 和颜色 c 构成。颜色 c 使用球谐函数来表示视角依赖的Appearance。所有这些属性都是可以通过反向传播学习和优化的。

3.1.2 视锥裁剪 (Frustum Culling)

给定指定的摄像机视角，这一步确定哪些3D高斯位于摄像机视锥外。通过这样做，视野外的3D高斯将不参与后续计算，从而节省计算资源。

3.1.3 Splatting

在这一步中，3D空间中的3D高斯（椭球）被投影到2D图像空间中（椭圆）以进行渲染。给定视图变换矩阵 W 和3D协方差矩阵 Σ ，投影后的2D协方差矩阵 Σ' 可以通过以下公式计算：

$$\Sigma' = JW\Sigma W^T J^T \quad (4)$$

其中， J 是仿射变换的雅可比矩阵（Jacobian）。

Important

原因较复杂，见：[\[3D Gaussian Splatting中的数学推导\]](#)

3.1.4 按像素渲染

在深入探讨3DGS的最终版本（利用了多种技术来并行计算）之前，先看一个较为简单的形式。给定一个像素的位置 x ，可以通过视图变换 W 计算该像素与所有重叠高斯的距离，即这些高斯的深度（Depth），形成一个排序后的高斯列表 \mathcal{N} 。然后，使用Alpha Compositing来计算该像素的最终颜色：

$$C = \sum_{n=1}^{|\mathcal{N}|} c_n \alpha'_n \prod_{j=1}^{n-1} (1 - \alpha'_j) \quad (5)$$

其中， c_n 是学习到的颜色。最终的不透明度 α'_n 是学习到的不透明度 α_n 与高斯的乘积，定义如下：

$$\alpha'_n = \alpha_n \cdot e^{-\frac{1}{2}(x' - \mu'_n)^T \Sigma_n'^{-1} (x' - \mu'_n)} \quad (6)$$

其中， x' 和 μ'_n 是投影空间中的坐标。

可以看出，上面的渲染过程需要对所有像素作排序，难以实现并行化，渲染速度也会大幅降低。对此，3DGS进行了若干让步，以适应并行计算。

3.1.5 瓦片 (Tiles)

为了避免每个像素计算高斯的开销，3DGS将精度从像素级别转移到Tiles级别的细节。具体来说，3DGS最初将图像划分为多个不重叠的Tiles。图3b提供了Tiles的示意图。每个Tile包含 16×16 像素。对某个投影来说，首先会确定哪些Tiles与其相交。若一个高斯的投影覆盖了多个Tiles，则复制它，并为每个副本分配一个标识符，即Tile ID。最后，高斯会根据Tile ID排序，然后渲染，即可获得Tile内的所有像素。

Tile中的所有像素访问存储在共享内存中的同一个有序高斯列表进行渲染。当系统依次处理每个高斯时，Tile中的每个像素都会根据距离（即（6）中的指数项）评估该高斯的贡献。因此，通过遍历高斯列表一次即可完成Tile的渲染。

① Note

像素和Tiles的计算流程是独立的，可以并行进行。

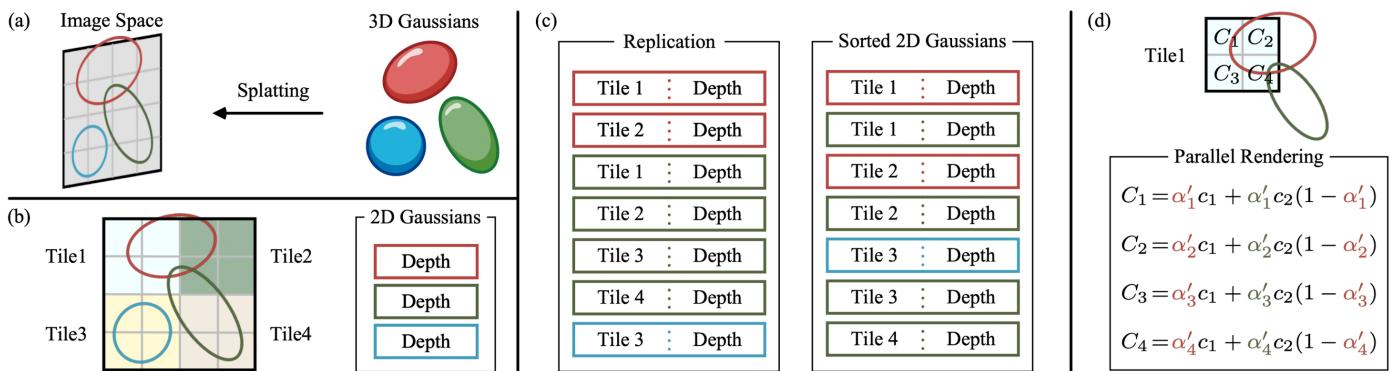


图 3 3DGS前向过程图示

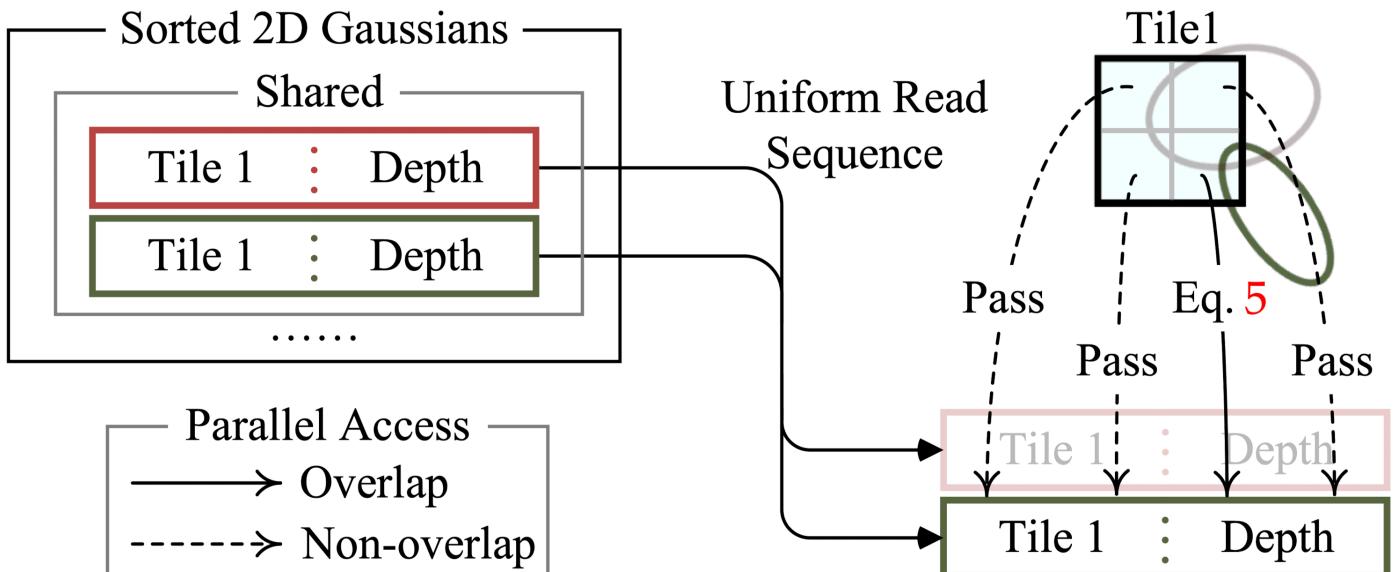


图 4 Tiles像素级并行渲染图示

3.1.6 并行渲染

在复制之后，3DGS将每个高斯的Tile ID与通过视图变换获得的Depth值相结合。这会产生一个未排序的字节列表，其中高位表示Tile ID，低位表示Depth。通过这种方式，可以直接利用排序后的列表进行渲染（即Alpha Compositing）。图3c和图3d展示了其视觉示意图。值得强调的是，Tiles和像素的渲染过程是独立的，非常适合并行计算。此外，每个Tile的像素可以访问一个共享的内存，并保持统一的读取顺序（图4），从而以更高的效率并行执行Alpha Compositing。

① Note

[2] 指出，3DGS对Tiles和像素的处理与CUDA编程架构中的块和线程类似。

3.2 3DGS优化

3DGS的核心即为优化。因为其需要构建大量3D高斯函数，并使其能够准确捕捉场景的本质，从而实现任意视角的渲染。一方面，3D高斯的属性需要通过可微渲染进行优化，以适应给定场景的纹理；另一方面，能够很好地表示给定场景的3D高斯数量是不确定的。一种比较有前景的方法是让神经网络自动学习3D高斯的密度。本文将在第3.2.1节介绍如何优化每个高斯的属性，并在第3.2.2节介绍如何控制高斯的密度。这两个过程在优化过程中交替进行。由于优化过程中存在许多手动设置的超参数，为简化起见，本文省略了大多数超参数的符号表示。

3.2.1 参数优化

3.2.1.1 损失函数（Loss Function）

一旦图像合成完成，就可以测量渲染图像与真实值之间的差异。所有可学习的参数通过使用 ℓ_1 和D-SSIM损失函数的随机梯度下降进行优化：

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{D-SSIM} \quad (7)$$

其中， $\lambda \in [0, 1]$ 是一个加权因子。

① Note

3DGS的损失函数与NeRF的损失函数略有不同：由于光线行进的计算开销较大，NeRF通常在像素级别计算损失，而不是在图像级别计算损失。

3.2.1.2 参数更新（Parameter Update）

大多数3D高斯的属性可以通过反向传播直接优化。需要注意的是，直接优化协方差矩阵 Σ 可能会导致其变成非半正定矩阵，这不符合协方差矩阵通常的物理解释。为了解决这个问题，3DGS选择优化四元数 q 和3D向量 s 。 q 和 s 分别表示旋转和缩放。这种方法允许协方差矩阵 Σ 通过以下公式重建：

$$\Sigma = RSS^T R^T \quad (8)$$

其中， R 和 S 分别表示从 q 和 s 导出的旋转和缩放矩阵。获得不透明度 α 涉及到复杂的计算图⁷，即 q 和 s 生成 Σ ， Σ 生成 Σ' ，而 Σ' 生成 α 。为了避免自动微分的计算成本，3DGS直接导出了 q 和 s 的梯度，以便在优化过程中直接计算它们。

3.2.2 密度控制

3.2.2.1 初始 (Initialization)

3DGS从运动恢复结构 (SfM) 的初始稀疏点集或随机初始化开始。需要注意的是，良好的初始化对于收敛性和重建质量是至关重要的。

3.2.2.2 点密化 (Point Densification)

在点密化阶段，3DGS自适应地增加高斯的密度，以更好地捕捉场景的细节。该过程主要针对几何特征缺失的区域或高斯分布过于稀疏的区域。Densification过程将在固定间隔内进行（即经过一定数量的训练迭代后），重点处理那些视角空间位置梯度较大的高斯（即高于特定阈值的高斯）。这包括在重建不足的区域克隆小高斯，或者在重建过度的区域分裂大高斯。对于克隆，高斯将被复制并移动到位置梯度的方向；对于分裂，大的高斯将被替换为两个较小的高斯，并按特定比例缩小它们的尺度。此步骤是为了寻求高斯在3D空间中的最佳分布和表示，以提高整体重建质量。

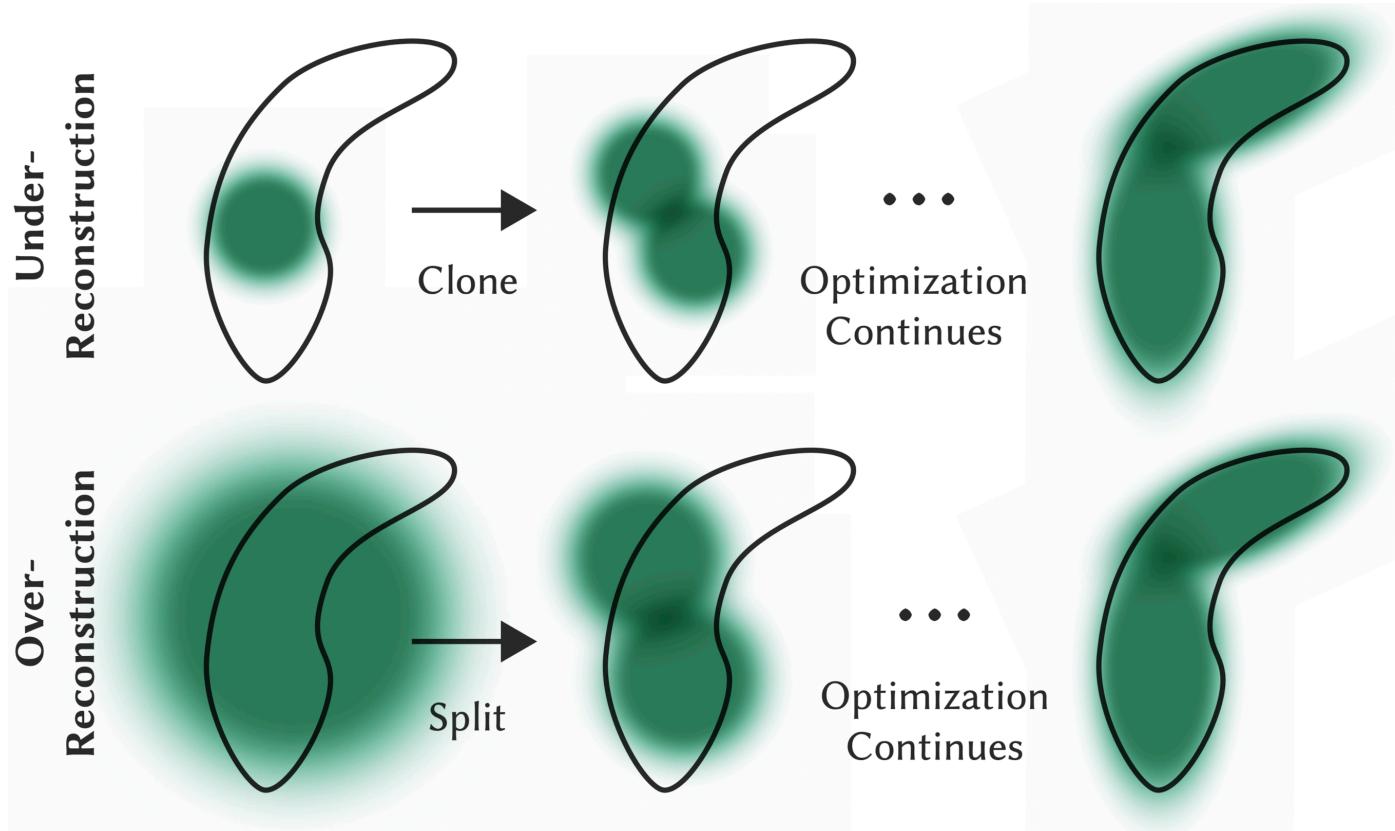


图 5 Densification过程^[2]

3.2.2.3 点剪枝 (Point Pruning)

Point Pruning阶段涉及移除多余或影响较小的高斯，可以视为一种正则化过程。该过程通过消除几乎透明 (α 值低于指定阈值) 或在世界空间或视角空间中过大的高斯来实现。此外，为防止输入相机附近高斯密度的不合理增加，在经过一定次数的迭代后，将高斯的 α 值设置为接近零。这使得必要高斯的密度能够得到控制性增加，同时剔除冗余的高斯。此过程不仅有助于节省计算资源，还确保了模型中的高斯在场景表示中保持精确和有效。

4 结语

在本次读书报告中，我详细解释了3D Gaussian Splatting (3DGS) 技术的整个过程，从其基本原理到具体的实现步骤，涵盖了其在渲染和优化中的关键环节。通过这次深入的学习，我不仅加深了对3DGS的理解，也对其在实际应用中的潜力有了更深刻的认识。

展望未来，我计划进一步深入研究3DGS的数学推理部分。3DGS中涉及的许多数学概念和推导，尤其是涉及到高斯函数的优化过程和协方差矩阵的处理，是理解这项技术核心原理的关键。因此，我希望通过系统地学习相关的数学知识，能够更加透彻地理解3DGS的工作机制，并为未来的算法改进奠定基础。

此外，我也计划更细致地研究3D GS的具体代码优化部分。虽然3DGS已经展示了其在实时渲染方面的巨大潜力，但在实际应用中仍存在进一步优化的空间。通过研究现有的代码实现，我希望能够找出可能的性能瓶颈，并探索如何通过算法优化或硬件加速来提升其运行效率。这不仅有助于提高3DGS的实际应用价值，也将为我在计算机图形学领域的进一步研究打下坚实的基础。

总而言之，这篇读书报告不仅是对3D GS技术的系统总结，也是我未来研究方向的明确规划。我将继续以此为基础，深入研究3DGS的数学推理和代码优化，通过不断的学习和实践，推动这一领域的发展，并为相关技术的进步贡献自己的力量。通过这些努力，我相信自己能够在计算机图形学领域取得更大的进展，并为未来的科研和技术创新打下坚实的基础。

参考文献：

- [1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, R. Ng, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," in *ECCV*, 2020, pp. 1-25.
- [2] B. Kerbl, G. Kopanas, T. Leimkuhler, and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," in *ACM Transactions on Graphics*, vol. 42, no. 4, 2023, pp. 1-14.
- [3] G. Chen, W. Wang, "A Survey on 3D Gaussian Splatting," *arXiv preprint arXiv:2401.03890*, 2024.
- [4] B. Fei, J. Xu, R. Zhang, Q. Zhou, W. Yang, Y. He, "3D Gaussian Splatting as New Era: A Survey," in *IEEE Transactions on Visualization and Computer Graphics*, 2024, pp. 1-20.
- [5] T. Wu, Y. Yuan, L. Zhang, J. Yang, Y. Cao, L. Yan, L. Gao, "Recent Advances in 3D Gaussian Splatting," *arXiv preprint arXiv:2403.11134*, 2024.
- [6] A. Dalal, D. Hagen, K. G. Robbersmyr, K. M. Knautsgård, "Gaussian Splatting: 3D Reconstruction and Novel View Synthesis, a Review," *arXiv preprint arXiv:2405.03417*, 2024.
- [7] Yanqi Bao, Tianyu Ding, Jing Huo, Yaoli Liu, Yuxin Li, Wenbin Li, Yang Gao, Jiebo Luo, "3D Gaussian Splatting: Survey, Technologies, Challenges, and Opportunities," *arXiv preprint arXiv:2407.17418*, 2024.