

# 三维高斯溅射研究进展

解君豪<sup>1)</sup>

<sup>1)</sup>(南京邮电大学计算机、软件工程、网络空间安全学院计算机科学与技术, 南京市210003)

**摘 要** 三维高斯喷溅(3DGS)最近在显式辐射场和计算机图形学领域展现了革命性的创新, 这种创新方法的主要特点为利用数百万个3D高斯函数, 这与传统的神经辐射场(NeRF)方法有很大的不同, 后者主要采用的方法为隐式的、基于坐标的模型来将空间坐标映射到像素值。而3DGS以其明确的场景表示和可微分的渲染算法, 不仅拥有实时渲染的能力, 而且引入了前所未有的可控制性和可编辑性。这说明3DGS在3D重建和表现领域的无限潜力, 在本文中, 我们提供对于三维领域的一个系统性概述。首先探讨3DGS出现背后的基本原理和驱动力, 为理解其重要性奠定基础。我们讨论的一个焦点是3DGS的实际适用性。通过这项调查, 旨在对3DGS提供一个全面的概述, 促进进一步探索辐射场表示。

**关键词** 3D高斯泼溅; 显示辐射场; 深度学习; 实时渲染

## A Survey on 3D Gaussian Splatting

junhao xie<sup>1)</sup>

<sup>1)</sup>(Department of school of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, 210000)

### Abstract

The recent revolutionary innovation of 3D Gaussian Splat (3DGS) in the fields of explicit radiation fields and computer graphics has distinctive features, utilizing millions of 3D Gaussian functions, which is very different from the traditional neural radiation field (NeRF) method, which mainly uses implicit, coordinate-based models to map spatial coordinates to pixel values. 3DGS, with its explicit scene representation and differentiable rendering algorithm, not only has real-time rendering capabilities but also introduces unprecedented controllability and editability. This indicates the infinite potential of 3DGS in the fields of 3D reconstruction and representation, and in this article, we provide a systematic overview of 3DGS. We first explore the fundamental principles and driving forces behind 3DGS, laying the foundation for understanding its importance. One of the focuses of our discussion is the practical applicability of 3DGS. Through this investigation, we aim to provide a comprehensive overview of 3DGS to facilitate further exploration of radiation field representation.

**Keywords** 3D Gaussian Splatting; Explicit Radiance Field; Deep Learning; Real-time Rendering

## 1 引言

辐射场合成的研究早在NeRF出现之前就已经开始,早期的研究主要集中在广场和基本的场景重建方法上<sup>[1]-[3]</sup>。然而,这些初始技术由于依赖于密集采样和结构化捕获,因此在处理复杂场景和光照条件时面临重大的挑战。随着运动结构恢复(SFM)和多视图立体(MVS)算法的出现,为3D场景重建提供了更加坚实的基础,也为更复杂的视图合成算法铺平了道路。神经辐射场的出现标志着计算机图形学和3D场景重建领域中一个重要的里程碑,彻底改变了我们进行新试图合成的方式。NeRF基于深度学习和计算机视觉,使我们能够从少量输入视图中渲染出高质量逼真场景,从而在图像合成领域中建立了一种新的范式。然而NeRF也面临着不足之处,尤其在于计算效率和可控性方面。而正式为了解决NeRF的不足之处,3DGS随之出现了,它不仅仅是一种改进,而是在场景表示和计算机图形学渲染方法上的根本性的改变,重新定义了表示和渲染的边界。其采用了显式表示和高度并行的工作流程,有利于更高效的计算和渲染,其创新性在于其独特的融合了可微分管道和基于点渲染技术的优点,通过用可学习的3D高斯函数表示场景,保留了连续体积辐射场的理想特性,这对于高质量图像合成至关重要。

## 2 背景

在本节中,我们首先要介绍两种主要的表示方式:隐式表示和显式表示。隐式表示,如NeRF所采用的,随便方式较为直接,但是在渲染过程中要求的计算成本较高。另一方面,显式表示,如网格结构所使用的,虽然提供更快数据访问,但是代价则是更高的内存使用。进一步地,建立了场景重建和渲染等相关领域的联系。

### 2.1 传统三维重建方法

传统三维重建由三个主要部分组成,首先是以SFM为代表的稀疏重建算法,这一步是为了获得图像序列的相机位姿,第二步是将第一步中估计出来的相机位姿作为输入,对图像中每一个像素做稠密重建,第三步式点云模型化(包括深度图拼接,深度图转网格,纹理贴图等),最后呈现出来的就是一个三维模型的样子。

#### 2.1.1 传统几何的三维重建方法

首先我们通过运动恢复结构SFM,其输入通常包含多视角图像以及摄像机内参数,我们通过摄像机将三维世界的物体转换成为像素,摄像机坐标系

下的一个三维点到像素之间的转换关系的建立是由摄像机内参数决定。该步骤的输出为指定场景的稀疏三维点云和摄像机的外参数。当我们明确了外参数,就能把世界坐标系里面的一个点,转换到当前的摄像机坐标系当中,继而再通过内参数转换到像素上,也就建立了世界坐标系的一个点和像素点的对应关系。其次我们在经过多视立体视觉,其输入为多视角的图像,以及前一个步骤所得到的摄像机的内外参数和稀疏的三维点云数据。而其输出为对三维稀疏点云的稠密化结果。继而我们再进行表面重建,其输入是上一步骤的稀疏点云数据,而输出结果为场景或者模型的三角网络。最后我们在进行纹理重建,其输入依旧是多视角图像、摄像机内外参数以及表面重建后的三维模型,而后得到我们的输出结果是带纹理的三维场景或模型。

### 2.2 辐射场

辐射场是三维空间中光分布的表征,它捕获了光如何与环境中的表面和材料相互作用<sup>[4]</sup>,数学上,一个辐射场可以描述为一个函数 $L: \mathbb{R}^5 \rightarrow \mathbb{R}^+$ ,其中 $L(x, y, z, \theta, \Phi)$ 映射空间中一个点 $(x, y, z)$ 和一个由球坐标 $(\theta, \Phi)$ 指定的方向,到一个非负的辐射值,辐射场可以通过隐式和显式表示封装,每个都具有场景表示和渲染的特定优势。

#### 2.2.1 隐式辐射场

隐式辐射场表示场景中的光分布,而不明确的定义场景的几何形状,最突出的例子式NeRF<sup>[5]</sup>,在NeRF中,MLP网络用于将一组空间坐标 $(x, y, z)$ 和观看方向 $(\theta, \Phi)$ 映射到颜色和密度值,任何点的亮度都不明确存储,而是通过查询神经网络实时计算,因此函数可以写成:

$$L_{\text{implicit}}(x, y, z, \theta, \Phi) = \text{NeuralNetwork}(x, y, z, \theta, \Phi) \quad (1)$$

这种格式允许复杂场景的可微分和紧凑的表示,尽管在渲染过程中,由于体积射线行进<sup>[6]</sup>,通常需要高计算符合。

#### 2.2.2 显式辐射场

相比之下,显式辐射场直接表示光在离散空间结构中的分布,例如体素网格或者一组点<sup>[7]</sup>,该结构中的每个元素存储其在空间中各自位置和辐射信息。这种方法允许更直接,更快速的访问辐射数据,但是由此造成的代价就是更高的内存使用和潜在的低分辨率。显式辐射场表示的一般形式可以写成:

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

其中DataStructure可以是点云或者网格, $f(\theta, \Phi)$ 是基于观看方向修改辐射的函数。

### 3 3D 高斯泼溅

在本节中，我们主要介绍3DGS的基本见解，其在实时、高分辨率图像渲染方面取得了突破性进展，无需依赖神经网络组件。我们将首先阐述3DGS如何在给定构造良好的3D高斯下来合成图像，即3DGS的正演过程。然后我们介绍如何构建良好的3D高斯变换，即三维图像的优化过程。最后，我们将从整体来概述整个3DGS的流程。以帮助大家对3DGS有一个宏观的理解。

#### 3.1 3D高斯函数的新视图合成

考虑到一个由数百万个优化的3D高斯函数表示的场景，目标是根据指定的摄像机姿态生成一幅图像，回想一下，NeRF通过计算要求较高的体积射线追踪来完成这个任务，对每个像素的3D空间点进行采样。这样的方法与高分辨率图像合成做比较，未能实现实时渲染速度<sup>[6]</sup>。而3DGS与之形成鲜明对比，3DGS首先将这些3D高斯分布投影到基于像素的图像平面上，这一过程被称作“splat” (图1)。然后3DGS对这些高斯分布进行分类，并计算每个像素的值。NeRF和3DGS的渲染可以看作是彼此的方向过程。接下来，我们从3D Gaussian的定义开始，它是3DGS中用于场景表示的最小元素，紧接其后，我们描述如何将这3D高斯函数用于可微渲染，最后，我们将介绍3DGS中使用的加速技术，这是快速渲染的关键所在。

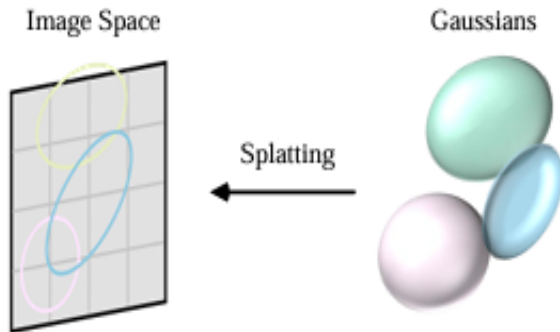


图1 将3D高斯投影到图像空间

##### 3.1.1 3D高斯函数的性质

一个3D高斯函数是以其中心位置 $\mu$ 、不透明度 $\alpha$ 、3D协方差矩阵 $\Sigma$ 和颜色 $C$ 为特征，其中 $C$ 通过球谐函数来表示。所有属性可以通过反向传播梯度进行学习和优化的。

##### 3.1.2 视锥体剔除

在给定的摄像机姿态的情况下，该步骤可以确定那些3D高斯函数分布在摄像机的视锥体之

外。通过此步骤，后续步骤中，不在给定视图中的3D高斯函数将不会参与计算，从而节省计算资源。

##### 3.1.3 泼溅

在该步骤中，3D高斯将会投影至2D图像中进行渲染。给定视角变换矩阵 $W$ 和三维协方差矩阵 $\Sigma$ ，计算投影二位协方差矩阵 $\Sigma'$ ，其计算公式如下：

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

其中 $J$ 是投影变换的仿射近似的雅各比矩阵<sup>[8]</sup>

##### 3.1.4 像素渲染

在深入研究利用不同的技术来提高并行计算的3DGS之前，首先对最简单的形式进行阐述，更便于我们理解其工作机制。给定像素点 $x$ 的位置，它到所有重叠高斯点的距离，即这些高斯点的深度，可以通过查看变换 $W$ 得到，形成高斯点的分布列表 $N$ 。然后采用 $\alpha$ 渲染计算该像素的最终颜色：

$$C = \sum_{i \in N} c_i \alpha'_i \prod_{j=1}^{i-1} (1 - \alpha'_j) \quad (4)$$

其中 $c_i$ 为学习到的颜色，最终的不透明度 $\alpha'_i$ 为学习到的不透明度 $\alpha_i$ 与高斯值相乘得结果：

$$\alpha'_i = \alpha_i \times \exp\left(-\frac{1}{2}(x' - \mu'_i)^T \Sigma_i'^{-1} (x' - \mu'_i)\right) \quad (5)$$

其中 $x'$ 和 $\mu'_i$ 是投影空间中得坐标，考虑到生成所需要的排列列表很难进行并行化，因此有一定可能性上述得渲染过程将比NeRF得效率更低，事实上，渲染速度的确会受到显著的影响。因此，为了实现实时渲染，3DGS做出了一些让步，以使得可以使用并行计算。

##### 3.1.5 Tiles

为了避免计算每个像素的高斯分布，3DGS将精度从像素级降低到了补丁级，具体来说，3DGS最初将图像分成多个不重叠的小块，在原始论文中被称为“tiles”，图2提供了tiles的示例。

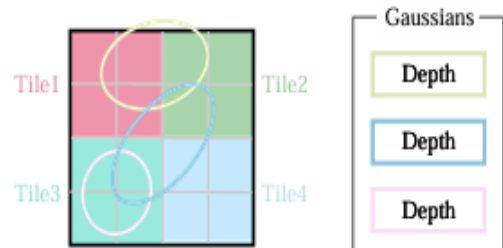


图2 tiles示例

每个贴图包含 $16 \times 16$ 个像素。3DGS进一步确定哪些贴图与这些投影的高斯分布相交，假设一个投影的高斯分布可能会覆盖几个图块，一个逻辑方

法是复制这些高斯分布，为每个拷贝分配一个标识符(tile id)。

### 3.1.6 并行渲染

对tile进行复制后，3DGS将各自的tile id与每个高斯的视图转换获得的深度图结合起来。这将产生一个未排序的字节列表，上面的位表示tile id，下面的位表示深度。通过这样做，排序列表可以直接用于渲染(即 $\alpha$ 渲染)。图3、图4提供了这些概念的可视化展示。值得强调的是，渲染的每个贴图和像素是独立发生的，这使得这个过程非常适合并行计算，另外一个额外的好处是，每个贴图的像素可以访问一个公共内存，并保持一个统一的读取序列，从而提高了 $\alpha$ 渲染的并行执行效率，在原论文的具体实现中，该框架将tile和像素的处理分别视为类似于CUDA编程架构中的块和线程。

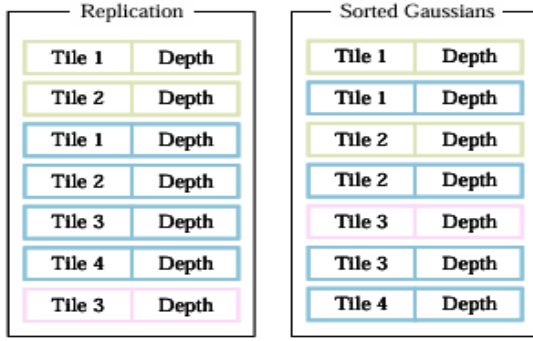


图3 复制和列表的示例

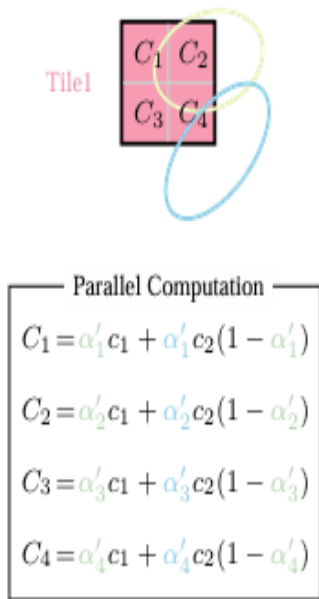


图4 并行渲染

简而言之，3DGS在前向处理阶段引入了几个近似，用于提高计算效率，同时保持高标准的图像合成质量。

### 3.2 三维高斯泼溅的优化

3DGS的核心是一个优化程序，旨在构建一个丰富的3D高斯集合，准确的捕获场景的本质，从而促进自由视点渲染。一方面，3D高斯函数的属性应该通过可微渲染来优化，以适应给定场景的纹理，另外一方面，可以很好的表示给定场景的3D高斯函数的数量是事前为止的，一个途径是让神经网络自动学习三位高斯的密度。我们接下来将介绍如何优化每个高斯函数的性质，以及如何控制高斯函数的密度。

#### 3.2.1 参数优化

损失函数：一旦图像合成完成，将会计算渲染图像和真实图像之间的差值，视为损失：

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{D-SSIM} \quad (6)$$

其中 $\lambda$ 是一个权重因子， $\mathcal{L}_1$ 损失和 $D-SSIM$ 是标准度量，3DGS的损失和NeRF的损失略有不同。NeRF由于昂贵的射线行进，通常在像素水平计算损失，而不是在图像水平。参数更新：三维高斯函数的大多数特性可以通过反向传播直接进行优化。但值得注意的是，直接优化协方差矩阵 $\Sigma$ 可能会导致非半正定矩阵，这将不符合协方差矩阵相关的物理解释。3DGS选择优化四元数 $q$ 和3D向量 $S$ ， $q$ 和 $S$ 分别代表旋转和缩放，这种方法允许协方差矩阵重构如下：

$$\Sigma = RSS^T R^T \quad (7)$$

其中 $R$ 和 $S$ 分别代表由 $q$ 和 $S$ 导出的旋转和缩放矩阵，有一个复杂的计算图来获得不透明度 $\alpha$ ，即 $q, S \rightarrow \Sigma, \Sigma \rightarrow \Sigma', \Sigma' \rightarrow \alpha$ 。为了避免自动微分的代价，3DGS导出 $q$ 和 $S$ 的梯度，以便在优化时可以直接计算他们。

#### 3.2.2 密度控制

初始化：3DGS从SFM或者随机初始化的稀疏点初始集开始。然后，采用点密度化和剪枝来控制三维高斯密度。点密度化：在点密度化阶段，3DGS自适应增加高斯密度，以用于更好的捕捉场景的细节，这个特征特别集中在缺少几何特征的区域或者高斯分布过于分散的区域。经过一定的迭代次数过后，密度会形成，目标时具有较大视图空间位置的梯度(即高于指定的阈值)的高斯函数，它包括在重建不足的区域克隆小高斯或在重建过度的区域分裂大高斯。对于克隆，创建一个高斯函数的副本并向位置梯度移动。对于分裂，一个大的



高斯函数被两个较小的高斯函数取代, 通过一个特定的因子减少它们的规模, 这一步寻求三维空间中高斯分布的最佳分布表示, 从而提高重建的整体质量。点修剪: 点修剪阶段包括去除多余的或者影响较小的高斯分布, 这在某种程度上可以看作是一个正则化过程, 这一步是通过消除几乎透明( $\alpha$  低于指定阈值)的高斯函数和那些在世界空间或视图空间中过大的高斯函数来执行的。此外, 为了防止输入摄像机附近高斯密度的不合理增加, 在一定迭代次数过后, 高斯函数的 $\alpha$ 值被设置成接近于零, 这允许必要的高斯密度的可控增加, 同时允许剔除冗余高斯密度, 这个过程不仅有助于节省计算资源, 而且还确保模型中的高斯函数在场景的表示中保持精确和有效。

### 3.3 3DGS的整体流程概述

本章节将会对3DGS的整体算法执行流程进行概述, 从宏观上叙述整体数据流程以及优化流程。

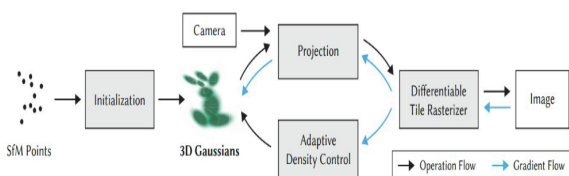


图5 3DGS流程图

我们首先从基于SfM得到点云数据进行初始化3D高斯, 每个三维点初始化为一个高斯椭球体。再获取给定摄像机的内、外参数, 之后将3维椭球体泼溅到2D图像上, 之后再通过 $\alpha$ 渲染进行光栅化渲染, 当得到渲染图像后, 与标答图像计算损失。之后进行梯度回传, 上分支回传梯度用于更新3D高斯椭球体的属性, 不断矫正3D高斯球的参数。下分支, 用于对3D高斯球进行克隆和分裂, 用于矫正每个高斯球对物体描述不清晰。若当前的3D高斯函数较小, 不足以渲染整个物体时, 将会对该3D高斯函数进行复制, 用于更完整的覆盖物体, 若当前的3D高斯函数较大时, 超出物体的覆盖范围时, 将分裂当前的3D高斯函数, 以确保更准备的渲染物体, 如图6所示。

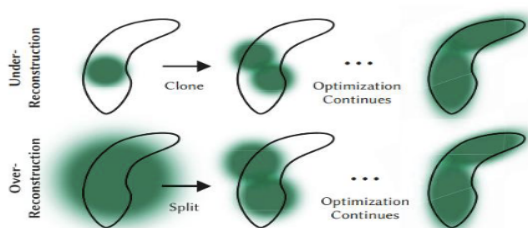


图6 复制和分裂过程

## 4 应用领域和具体任务

在本节中, 我们将进一步深入了解3DGS所应用的具体任务和各个不同的领域, 他在不同的领域中, 给我们带来了超乎想象的变革力量。因此, 我们将深度探讨3DGS正在产生重大影响的各个应用领域。如机器人、场景重建和表示、人工智能生成内容, 自动驾驶等学科。3DGS的应用展示了其多功能性和革命性潜力。在这里, 我们将概述一些值得关注的领域, 表明自己对这些领域的一些见解。

### 4.1 同步定位和地图创建(SLAM)

SLAM是机器人和自主系统的核心计算问题。它涉及到机器人或者设备在未知的环境中了解其位置信息的挑战, 同时绘制环境布局<sup>[9]</sup>。SLAM在各种应用中都至关重要, 这些应用包括但不限于自动驾驶汽车、增强现实和机器人导航等。SLAM的核心是创建一个未知领域的地图, 并实时确定设备在该地图上的位置。因此SLAM对计算密集型场景表示技术提出了巨大的挑战。而3DGS方法因其提供连续表面建模、降低内存要求、改进噪声和异常值处理、增强孔填充和场景修复以及3D网格重建的灵活分辨率而脱颖而出<sup>[10]</sup>

3DGS作为一种创新的场景表示方法进入SLAM领域。传统的SLAM系统通常使用点或者体素网格来表示环境<sup>[11]</sup>。相比之下, 3DGS则利用各向异性高斯来更好的表示环境。这种表示方式具体以下几种益处: 1) 高效性: 三位高斯密度是自适应控制的, 以便于更紧凑的表示空间数据, 减少计算负荷。2) 准确性: 各向异性高斯允许更详细和更准确的环境建模, 特别是在复杂或动态变化场景中有益。3) 适应性: 3DGS可以适应各种规模和复杂的环境, 使其适用于不同的SLAM应用。一些创新研究已经在SLAM中使用了3DGS<sup>[12]-[14]</sup>, 展现出3DGS的巨大潜力和多功能性。例如GS SLAM<sup>[15]</sup>在SLAM系统中引入了3D Gaussians用于场景表示。该方法注重效率和精度的平衡, 显著的改善了RGS-D映射中的地图优化和重绘制过程。在将神经渲染与SLAM相结合时, photo-SLAM<sup>[16]</sup>利用3DGS进行高效、高质量绘图。它展示了在逼真的地图绘制和渲染速度方面的显著改进, 特别是在便携式设备上。3DGS在动态城市场景重建方面也显示出了令人印象深刻的结果<sup>[17]</sup>。

### 4.2 自动驾驶

在自动驾驶领域, 3DGS主要应用于大规模驾驶场景的动态重构和组合SLAM应用。

#### 4.2.1 自动驾驶场景重建

驾驶场景重构是一项具有挑战性的任务，涉及大规模场景重构、动态对象重构、静态对象重构、高斯混合重构等多个技术领域。驾驶场景重建的重建过程<sup>[18]-[20]</sup>被区分为静态背景重建和动态目标重建。Driving Gaussian<sup>[18]</sup>旨在利用多传感器数据重构自动驾驶中的大规模动态场景。在静态背景下，Driving Gaussian在不同深度箱下引入增量静态三维高斯函数，以减轻远处街景造成的尺度混淆。对于动态对象，Driving Gaussian引入动态高斯图来构建多个目标(其属性包括位置、局部到师姐坐标变换矩阵、方向等)之间的关系，共同重构整个自动驾驶场景。Street Gaussians<sup>[19]</sup>采用了类似的方法，关键的区别在于重建背景和前景引入了语义属性。此外Street Gaussians使用傅里叶变换来有效地表示3DGS的SH时间变换。在前人的基础上，HUGS<sup>[20]</sup>结合了独轮车模型和正确的速度建模，以协助在物理约束下进行动态重建，与之前的动态3DGS<sup>[21]</sup>工作类似，HUGS也采用光流监督，并结合渲染RGB损失，语义损失和独轮车模型损失，提高了动态重建精度。此外3DGS已被应用于多模态时空校准任务<sup>[22]</sup>。通过利用LiDAR点云作为高斯位置的参考，3DGS-Calib构建了连续的场景表示，并在所有传感器之间执行几何和光度一致性，与基于NeRF的方法相比，显著减少了训练时间，实现了准确和稳健的校准。

#### 4.2.2 自动驾驶应用

自动驾驶旨在让车辆在没有人干预的情况下导航和操作。这些车辆配备了一套传感器，包括相机、LiDAR、雷达，并结合了先进的算法、机器学习模型和强大的计算能力。中心目标是感知环境，因此需要自动驾驶汽车需要感知和解读周围环境，才能安全行驶，这包括实时重构驾驶场景，准确识别静态和动态物体，并了解他们的空间关系和运动，在完成上述自动驾驶场景重建后，汽车仍需要对其识别到的场景进行实时应对。

### 4.3 AIGC

AIGC是指由人工智能系统自主创建或显著改变的数字内容，特别是在计算机视觉、自然语言处理和机器学习领域。AIGC的特点是能够模拟、扩展或增强人工生成的内容，实现从逼真图像合成到动态叙事创作的应用。AIGC的意义在于其在各个领域的变革潜力，包括娱乐、教育和技术发展。它是不断发展的数字内容创作格局中的一个关键元素，为传统方法提供了可扩展、可定制且通常更高效的替代方案。3DGS的这种明确特性有助于实现

实时渲染功能以及前所未有的控制和编辑水平，使其与AIGC应用程序高度相关。3DGS的显式场景表示和可微分渲染算法完全符合AIGC生成高保真、实时和可编辑内容的要求，这对虚拟现实、交互式媒体等领域的应用至关重要。

## 5 性能比较

在本节中，我们将进一步深入讨论集中3DGS算法的性能，提供对3DGS的更多了解。3DGS在许多任务中的不同应用，加上每个任务的定制算法设计，使得在单个任务或数据集集中对3DGS算法进行统一比较变得不切实际，因此，我们在3DGS领域中选择几个具有代表性的任务进行性能评估。

### 5.1 定位任务

定位任务涉及在环境中确定机器人或者设备的精确位置和方向，通常使用传感器数据。

数据集：Replica数据集<sup>[23]</sup>是18个十分详细的三维场景的集合，旨在推进各种机器学习和计算机视觉领域的研究。这些场景不仅视觉逼真，而且还提供全面的数据，包括密度网格，高质量HDR纹理，以及每个元素的详细语义信息，根据[24]，我们使用三个关于房间的序列和五个关于办公室的序列进行评估。

基准算法：为了进行性能比较，我们涉及了4篇论文，这些论文将3D Gaussians引入到了他们的系统中(Gaussian-SLAM<sup>[12]</sup>, GSS-LAM<sup>[13]</sup>, SplaTAM<sup>[14]</sup>, GS-SLAM<sup>[15]</sup>)，以及七种典型的SLAM方法(F2F<sup>[24]</sup>, iMAP<sup>[25]</sup>, Vox-Fusion<sup>[26]</sup>, NICE-SLAM<sup>[27]</sup>, ESLAM<sup>[28]</sup>, Point-SLAM<sup>[29]</sup>, Co-SLAM<sup>[30]</sup>)

评估标准：绝对轨迹误差和均方根误差是评估SLAM系统常用的度量标准<sup>[31]</sup>。测量机器人或者设备整个轨迹上估计位置和真是位置之间的欧氏距离的均方根。

结果：如表一、二所示，展示了几种算法的绝对轨迹误差评估结果。最近基于3DGS的定位算法比现有的SLAM系统又明显的优势。

### 5.2 渲染静态场景

静态场景渲染主要是指将一个三维场景通过计算和渲染技术表现为一个静态画面，渲染的重点是将计算机可读信息(如场景中的3D物体)转换为基于像素的图像，渲染目标是生成高质量的单帧图像。在此我们将对渲染结果的质量进行评估。

数据集：我们使用和定位算法相同的数据集。但仅对房间数据集进行测试，得到算法的渲染质量。

表1 定位算法房间性能比较

Method	3DGS	Room0	Room1	Room2
[24]		1.64	1.92	2.80
[25]		3.12	2.54	2.31
[26]		1.37	4.70	1.47
[27]		0.97	1.31	1.07
[28]		0.71	0.70	0.52
[29]		0.61	0.41	0.37
[30]		0.70	0.95	1.35
[12]	✓	3.35	8.74	3.13
[13]	✓	0.47	0.43	0.31
[14]	✓	0.31	0.40	0.29
[15]	✓	0.48	0.53	0.33

表2 定位算法办公室性能比较

Office0	Office1	Office2	Office3	Office4
2.48	0.80	4.55	2.64	2.27
1.69	1.03	3.99	4.05	1.93
8.48	2.04	2.58	1.11	2.94
0.88	1.00	1.06	1.10	1.13
0.57	0.55	0.58	0.72	0.63
0.38	0.48	0.54	0.69	0.72
0.59	0.55	2.03	1.56	0.72
1.11	0.81	0.78	1.08	7.21
0.70	0.57	0.31	0.31	3.20
0.47	0.27	0.29	0.32	0.55
0.52	0.41	0.59	0.46	0.70

基准算法：为了评估比较性能，我们引用了四篇最近的论文，这些论文将3DGS引入了他们的系统中(Gaussian-SLAM<sup>[12]</sup>, GSS-LAM<sup>[13]</sup>, SplaTAM<sup>[14]</sup>, GS-SLAM<sup>[15]</sup>)，针对这四篇论文所提出的方法评估其对房间数据集渲染画面的质量

评估标准：峰值信噪比(PSNR)、相似结构度(SSIM)、图像感知相似度(LPIPS)被用于评价渲染结果的质量。

结果：表三展示了基于3DGS系统生成结果的渲染质量。

## 6 未来和展望

最近3DGS的工作日渐涌现,但仍然存在一些我们认为应该克服的开放挑战, 3DGS领域的蓬勃发展促成了许多相关下游任务和技术出现,从而导致了该领域内出现了许多仍然需要克服的挑战,以及导致了该领域内日益增加的复杂性和混乱, 因此我

表3 静态渲染房间性能比较

Method	Metric	Room0	Room1	Room2
[12]	PSNR	34.31	37.28	38.18
	SSIM	0.99	0.99	0.99
	LPIPS	0.08	0.07	0.07
[13]	PSNR	34.83	36.43	37.49
	SSIM	0.95	0.96	0.96
	LPIPS	0.07	0.08	0.07
[14]	PSNR	32.86	33.89	35.25
	SSIM	0.98	0.97	0.98
	LPIPS	0.07	0.10	0.08
[15]	PSNR	31.56	32.86	32.59
	SSIM	0.97	0.97	0.97
	LPIPS	0.09	0.07	0.09

们仍存在许多需要解决的问题。

### 6.1 3DGS的数据高效性

从有限的点生成新视图和重建场景具有十分重要的意义,特别是因为他们具有以最小的输入增强现实主义和用户体验感的潜力,最近又需要工作正在从事这样的研究,因此这一领域仍然有进一步探索的迫切需要,此外, 3DGS的一个值得注意的问题是,在观测数据不足的情况下,稀疏的数据会导致不准确的重建, 因此, 在稀疏区域开发新的数据插值方式或者集成方法,仍然是未来研究的一个有希望的领域。

### 6.2 更具体的下游任务

目前3DGS算法的速度与质量已经可以取得了较高的一个水准, 因此在3D领域的很多工作可以在基于3DGS所渲染的场景下完成, 包括但不限于目标检测、语义分割、开放词汇分割等。而当下也有很多研究正在不断探索3DGS在3D场景分割领域的挑战。

### 6.3 更先进的渲染算法

目前3DGS的渲染算法是简单的,可以进一步优化,例如, 简单的可见性算法可能导致高斯函数深度/混合顺序的剧烈切换, 由此可以看出,实现更先进的渲染算法将会是未来研究的一个重要机会,这些算法的改进应该旨在更准确的模拟给定场景中光线和材料属性之间复杂的相互作用。一种很有前途

的方法可能是涉及将传统计算机图形学的既定原则同化和适应到3DGS的具体环境中。

### 参 考 文 献

- [1] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Seminal Graphics Papers: Pushing the Boundaries*, Volume 2, 2023, pp. 453–464.
- [2] M. Levoy and P. Hanrahan, "Light field rendering," in *Seminal Graphics Papers: Pushing the Boundaries*, Volume 2, 2023, pp. 441–452.
- [3] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, "Unstructured lumigraph rendering," in *Seminal Graphics Papers: Pushing the Boundaries*, Volume 2, 2023, pp. 497–504.
- [4] Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. Sitzmann, and S. Sridhar, "Neural fields in visual computing and beyond," in *Computer Graphics Forum*, vol. 41, no. 2. Wiley Online Library, 2022, pp. 641–676.
- [5] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2020, pp. 405–421.
- [6] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Trans. Graphics*, vol. 42, no. 4, 2023.
- [7] C. Sun, M. Sun, and H.-T. Chen, "Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5459–5469.
- [8] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross, "Ewa volume splatting," in *Proceedings Visualization*, 2001. VIS'01. IEEE, 2001, pp. 29–538.
- [9] J. A. Placed, J. Strader, H. Carrillo, N. Atanasov, V. Indelman, L. Carlone, and J. A. Castellanos, "A survey on active simultaneous localization and mapping: State of the art and new frontiers," *IEEE Transactions on Robotics*, 2023.
- [10] F. Tosi, Y. Zhang, Z. Gong, E. Sandström, S. Mattoccia, M. R. Oswald, and M. Poggi, "How nerfs and 3d gaussian splatting are reshaping slam: a survey," *arXiv preprint arXiv:2402.13255*, 2024.
- [11] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.
- [12] V. Yugay, Y. Li, T. Gevers, and M. R. Oswald, "Gaussian-slam: Photo-realistic dense slam with gaussian splatting," *arXiv preprint arXiv:2312.10070*, 2023.
- [13] C. Yan, D. Qu, D. Wang, D. Xu, Z. Wang, B. Zhao, and X. Li, "Gs-slam: Dense visual slam with 3d gaussian splatting," *arXiv preprint arXiv:2311.11700*, 2023.
- [14] N. Keetha, J. Karhade, K. M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan, and J. Luiten, "Splatam: Splat, track & map 3d gaussians for dense rgb-d slam," *arXiv preprint arXiv:2312.02126*, 2023.
- [15] C. Yan, D. Qu, D. Wang, D. Xu, Z. Wang, B. Zhao, and X. Li, "Gs-slam: Dense visual slam with 3d gaussian splatting," *arXiv preprint arXiv:2311.11700*, 2023.
- [16] H. Huang, L. Li, H. Cheng, and S.-K. Yeung, "Photo-slam: Real-time simultaneous localization and photorealistic map ping for monocular, stereo, and rgb-d cameras," *arXiv preprint arXiv:2311.16728*, 2023.
- [17] Y. Yan, H. Lin, C. Zhou, W. Wang, H. Sun, K. Zhan, X. Lang, X. Zhou, and S. Peng, "Street gaussians for modeling dynamic urban scenes," *arXiv preprint arXiv:2401.01339*, 2024.
- [18] X. Zhou, Z. Lin, X. Shan, Y. Wang, D. Sun, and M.-H. Yang, "Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes," *arXiv preprint arXiv:2312.07920*, 2023.
- [19] Y. Yan, H. Lin, C. Zhou, W. Wang, H. Sun, K. Zhan, X. Lang, X. Zhou, and S. Peng, "Street gaussians for modeling dynamic urban scenes," *arXiv preprint arXiv:2401.01339*, 2024.
- [20] H. Zhou, J. Shao, L. Xu, D. Bai, W. Qiu, B. Liu, Y. Wang, A. Geiger, and Y. Liao, "Hugs: Holistic urban 3d scene understanding via gaussian splatting," *arXiv preprint arXiv:2403.12722*, 2024.
- [21] K. Katsumata, D. M. Vo, and H. Nakayama, "An efficient 3d gaussian representation for monocular/multi-view dynamic scenes," *arXiv preprint arXiv:2311.12897*, 2023.
- [22] Q. Herau, M. Bennehar, A. Moreau, N. Piasco, L. Roldao, D. Tsishkou, C. Migniot, P. Vasseur, and C. Demonceaux, "3dgsalib: 3d gaussian splatting for multimodal spatiotemporal calibration," *arXiv preprint arXiv:2403.11577*, 2024.
- [23] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma et al., "The replica dataset: A digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019.
- [24] J. Park, Q.-Y. Zhou, and V. Koltun, "Colored point cloud registration revisited," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 143–152.
- [25] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "imap: Implicit map ping and positioning in real-time," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6229–6238.
- [26] X. Yang, H. Li, H. Zhai, Y. Ming, Y. Liu, and G. Zhang, "Vox fusion: Dense tracking and mapping with voxel-based neural implicit representation," in *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2022, pp. 499–507.
- [27] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12786–12796.



- 
- [28] M. M. Johari, C. Carta, and F. Fleuret, “Eslam: Efficient dense slam system based on hybrid representation of signed distance fields,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17408–17419.
- [29] E. Sandström, Y. Li, L. Van Gool, and M. R. Oswald, “Point slam: Dense neural point cloud-based slam,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 18433–18444.
- [30] H. Wang, J. Wang, and L. Agapito, “Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13293–13302.
- [31] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of rgb-d slam systems,” in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 573–580.