三维高斯溅射：实时辐射场渲染

3D Gaussian Splatting for Real-Time Radiance Field Rendering

Bernhard Kerbl · Georgios Kopanas · Thomas Leimkühler · George Drettakis

SIGGRAPH 2023 / ACM TOG 42(4)
arXiv:2308.04079v1 · 英文原版 14 页

原作：Bernhard Kerbl · Georgios Kopanas · Thomas Leimkühler · George Drettakis。3D Gaussian Splatting for Real-Time Radiance Field Rendering，SIGGRAPH 2023 / ACM TOG 42(4)。采用 arXiv:2308.04079v1 官方版本。arXiv页面的 view license 指向 Creative Commons Attribution 4.0 International（CC BY 4.0）。本中文译本为翻译与重新排版的演绎作品；原图、公式、表格及英文书目保留，中文译者为Codex辅助编译，2026-11-25。原作者未参与或背书本译本。许可：https://creativecommons.org/licenses/by/4.0/ 。

全文逐段精译 · 章节导航

1. 引言
2. 相关工作
3. 方法概览
4. 可微三维高斯溅射
5. 三维高斯优化与自适应密度控制
6. 高斯快速可微光栅化
7. 实现、结果与评估
8. 讨论与结论
参考文献（保留英文书目）
附录A 梯度计算细节
附录B 优化与增密算法
附录C 光栅化器细节
附录D 各场景误差指标

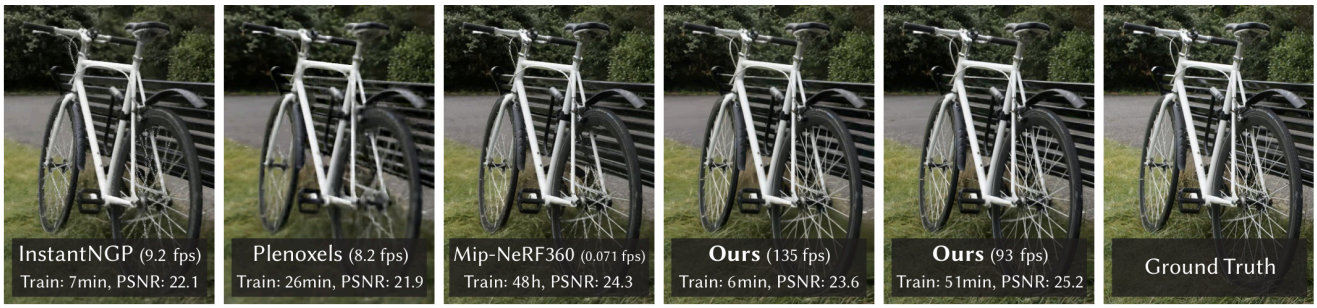

图1. 本方法以与此前最快方法相当的优化时间，实现辐射场实时渲染，质量达到此前最佳方法的水平。关键是新的三维高斯场景表示与实时可微渲染器，两者显著加速场景优化和新视角合成。训练时间与InstantNGP相当时，我们的质量也与之相近；而InstantNGP此时已达到其最高质量，我们继续训练51分钟后可达到SOTA，甚至略优于Mip-NeRF360。

原段引文：(Barron et al., 2022)；(Fridovich-Keil and Yu et al., 2022 ; Müller et al., 2022)；(Müller et al., 2022)

摘要

S001 · 英文底本 p.1

辐射场方法近来革新了由多张照片或视频采集场景的新视角合成。然而，获得高视觉质量仍需要训练和渲染成本高昂的神经网络，而近期更快的方法不可避免地以质量换速度。对于无界的完整场景（而非孤立物体）及1080p分辨率，现有方法尚不能达到实时显示速率。我们提出三个关键要素，在保持有竞争力的训练时间的同时，实现最先进的视觉质量，并在1080p下提供高质量实时（≥ 30 fps）新视角合成。第一，以相机标定产生的稀疏点为起点，用三维高斯表示场景，既保留连续体辐射场有利于场景优化的性质，又避免在空区域进行无效计算。第二，交替执行三维高斯的优化与密度控制，尤其优化各向异性协方差，从而准确表示场景。第三，开发快速、感知可见性的渲染算法，支持各向异性溅射，既加速训练又支持实时渲染。我们在多个成熟数据集上展示了最先进的视觉质量与实时渲染能力。

1. 引言

S002 · 英文底本 p.1

网格与点是最常见的三维场景表示，因为它们是显式的，适合快速GPU/CUDA光栅化。相反，近期的神经辐射场（NeRF）采用连续场景表示，通常通过体射线步进优化多层感知机（MLP），以合成已采集场景的新视角。迄今最高效的辐射场方案同样依赖连续表示，例如对体素、哈希网格或点中存储的值进行插值。这种连续性有助于优化，但渲染所需的随机采样成本高，且可能带来噪声。我们提出兼收两者优势的新方法：三维高斯表示使优化达到最先进（SOTA）视觉质量与有竞争力的训练时间；分块溅射方案则在多个已发表数据集上，以1080p分辨率实现SOTA质量的实时渲染（图1）。

原段引文：(Fridovich-Keil and Yu et al., 2022)；(Müller et al., 2022)；(Xu et al., 2022)；(Knapitsch et al., 2017 ; Hedman et al., 2018 ; Barron et al., 2022)

S003 · 英文底本 p.1

我们的目标是对多张照片采集的场景进行实时渲染，并使典型真实场景的表示优化速度达到此前最高效方法的水平。近期方法能够快速训练，但难以达到当前SOTA NeRF方法Mip-NeRF360的视觉质量，后者训练时间可长达48小时。较快但质量较低的辐射场方法，在某些场景可获得10–15帧/秒的交互式渲染，仍不能在高分辨率下达到实时。

原段引文：(Fridovich-Keil and Yu et al., 2022 ; Müller et al., 2022)；(Barron et al., 2022)

S004 · 英文底本 p.2

我们的方案包括三个主要组成部分。首先，引入三维高斯作为灵活而有表达力的场景表示。输入与此前NeRF类方法相同，即由运动恢复结构（SfM）标定的相机；三维高斯集合用SfM过程附带生成的稀疏点云初始化。多数点方法需要多视图立体（MVS）数据，而我们仅以SfM点为输入即可获得高质量结果。对于NeRF-synthetic数据集，即使随机初始化也能获得高质量。三维高斯是一种优秀选择：它既是可微体表示，又可高效投影到二维并以标准α混合进行光栅化，采用与NeRF等价的图像形成模型。第二部分是优化三维高斯的属性——三维位置、不透明度α、各向异性协方差和球谐（SH）系数——并交替进行自适应密度控制，在优化中添加及偶尔移除高斯。这样得到相对紧凑、无结构而准确的场景表示，所测试场景均使用100万–500万个高斯。第三部分是实时渲染方案：采用快速GPU排序，并借鉴近期工作中的分块光栅化。三维高斯表示使我们能够通过排序和α混合，执行遵守可见性顺序的各向异性溅射；同时跟踪任意所需数量的有序溅射遍历过程，实现快速准确的反向传播。

原段引文：(Snavely et al., 2006)；(Aliev et al., 2020 ; Kopanas et al., 2021 ; Rückert et al., 2022)；(Lassner and Zollhofer, 2021)

综上，我们作出以下贡献：

- S005 · 英文底本 p.2

  引入各向异性三维高斯，作为高质量、无结构的辐射场表示。

- S006 · 英文底本 p.2

  提出三维高斯属性优化方法，与自适应密度控制交替执行，为采集场景建立高质量表示。

- S007 · 英文底本 p.2

  提出面向GPU的快速可微渲染方法；该方法感知可见性，支持各向异性溅射与快速反向传播，实现高质量新视角合成。

S008 · 英文底本 p.2

在已发表数据集上的结果表明，我们能从多视图采集结果优化三维高斯，达到或超过此前质量最高的隐式辐射场方法。我们也能获得与最快方法相近的训练速度和质量，更重要的是，首次实现用于新视角合成的高质量实时渲染。

2. 相关工作

S009 · 英文底本 p.2

我们先简要回顾传统重建，再讨论点渲染和辐射场工作及其相似性。辐射场领域很广，因此仅聚焦直接相关研究；完整综述见Tewari等（2022）和Xie等（2022）。

原段引文：(Tewari et al., 2022 ; Xie et al., 2022)

2.1. 传统场景重建与渲染

S010 · 英文底本 p.2

最早的新视角合成基于光场，先要求密集采样，随后支持无结构采集。SfM使从照片集合合成新视角成为一个新的完整研究领域。相机标定时估计出的稀疏点云，最初仅用于简单的三维空间可视化。随后，多视图立体逐年发展出令人印象深刻的完整三维重建算法，推动了多种视角合成算法。这些方法均将输入图像重投影并混合到新视角，以几何指导重投影；在不少情况下效果优秀，却通常无法彻底弥补未重建区域，或MVS生成不存在几何所导致的“过重建”。近期神经渲染算法大幅减少此类伪影，避免将全部输入图像存入GPU的巨大成本，并在多数方面超过这些方法。

原段引文：(Gortler et al., 1996 ; Levoy and Hanrahan, 1996)；(Buehler et al., 2001)；(Snavely et al., 2006)；(Goesele et al., 2007)；(Eisemann et al., 2008 ; Chaurasia et al., 2013 ; Hedman et al., 2018 ; Kopanas et al., 2021)；(Tewari et al., 2022)

2.2. 神经渲染与辐射场

S011 · 英文底本 p.2

深度学习很早就用于新视角合成；卷积神经网络（CNN）被用于估计混合权重或构建纹理空间方案。多数这类方法的主要缺点是依赖MVS几何；此外，使用CNN生成最终图像往往导致时间闪烁。

原段引文：(Zhou et al., 2016 ; Flynn et al., 2016)；(Hedman et al., 2018)；(Thies et al., 2019 ; Riegler and Koltun, 2020)

S012 · 英文底本 p.2

Soft3D开创了用于新视角合成的体表示；随后出现深度学习与体射线步进结合的方法，以连续可微密度场表示几何。由于需要大量采样来查询体数据，体射线步进渲染的成本显著。NeRF引入重要性采样与位置编码提高质量，但大型MLP影响速度。NeRF成功后，大量后续工作通过正则化等策略改善质量和速度；当前新视角合成图像质量的SOTA是Mip-NeRF360。虽然质量卓越，其训练与渲染仍极耗时；我们能达到、在某些情况下超过这一质量，同时保持快速训练和实时渲染。

原段引文：(Penner and Zhang, 2017)；(Sitzmann et al., 2019 ; Henzler et al., 2019)；(Mildenhall et al., 2020)；(Barron et al., 2022)

S013 · 英文底本 p.2

最近的方法主要利用三类设计来加速训练或渲染：用空间数据结构存储（神经）特征，在体射线步进中插值；使用不同编码；以及调整MLP容量。具体包括不同空间离散化变体、码本和哈希表等编码，从而允许使用更小的MLP，甚至完全不使用神经网络。

原段引文：(Takikawa et al., 2021 ; Fridovich-Keil and Yu et al., 2022 ; Yu et al., 2021 ; Hedman et al., 2021 ; Chen et al., 2022a ; Garbin et al., 2021 ; Reiser et al., 2021 ; Chen et al., 2022b ; Wu et al., 2022)；(Takikawa et al., 2022)；(Müller et al., 2022)；(Fridovich-Keil and Yu et al., 2022 ; Sun et al., 2022)

S014 · 英文底本 p.3

其中最突出的是InstantNGP与Plenoxels：前者以哈希网格和占据网格加速计算，用较小MLP表示密度及外观；后者以稀疏体素网格插值连续密度场，可完全去掉神经网络。二者都依赖球谐：前者直接表示方向效应，后者编码颜色网络输入。尽管二者效果优秀，空空间的高效表示仍受场景或采集类型影响。用于加速的结构化网格在很大程度上限制了图像质量，每次射线步进需要查询大量样本又限制了渲染速度。我们采用无结构、显式且GPU友好的三维高斯，不使用神经组件即可获得更快渲染与更高质量。

原段引文：(Müller et al., 2022)；(Fridovich-Keil and Yu et al., 2022)

2.3. 点渲染与辐射场

S015 · 英文底本 p.3

点方法可高效渲染不相连、无结构的几何样本，即点云。最简单的点采样渲染以固定尺寸光栅化无结构点集，可利用图形API原生点类型或GPU并行软件光栅化。虽然忠于原始数据，点采样渲染却有孔洞、混叠和严格不连续的问题。高质量点渲染的奠基工作通过“溅射”范围大于像素的点图元来处理这些问题，例如圆盘、椭圆盘、椭球或表面元。

原段引文：(Gross and Pfister, 2011)；(Grossman and Dally, 1998)；(Sainz and Pajarola, 2004)；(Schütz et al., 2022 ; Laine and Karras, 2011)；(Zwicker et al., 2001b ; Botsch et al., 2005 ; Ren et al., 2002 ; Pfister et al., 2000)

S016 · 英文底本 p.3

近期研究关注可微点渲染：为点附加神经特征，再用CNN渲染，可以实现快速乃至实时视角合成。然而，它们仍依赖MVS提供初始几何，因此继承其伪影，尤其是无纹理、光亮区域或细结构等困难情况中的过重建或欠重建。

原段引文：(Yifan et al., 2019 ; Wiles et al., 2020)；(Rückert et al., 2022 ; Aliev et al., 2020)

S017 · 英文底本 p.3

基于点的α混合与NeRF式体渲染，本质上采用相同的图像形成模型。具体说，颜色C由沿射线的体渲染给出：（1）

$$C = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i \text{ with } T_i = \exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right),$$

，其中密度σ、透射率T与颜色$\mathbf{c}$以间隔δ_i沿射线采样。它可改写为（2）

$$C = \sum_{i=1}^{N} T_i\alpha_i\mathbf{c}_i,$$

，其中

$$\alpha_i = (1 - \exp(-\sigma_i\delta_i)) \text{ and } T_i = \prod_{j=1}^{i-1}(1 - \alpha_i).$$

。典型神经点方法通过混合覆盖该像素的$\mathcal{N}$个有序点，计算像素颜色C：（3）

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

。这里$\mathbf{c}_i$是各点颜色，α_i由协方差为Σ的二维高斯求值，再乘以可学习的逐点不透明度得到。

原段引文：(Kopanas et al., 2021 ; Kopanas et al., 2022)；(Yifan et al., 2019)

S018 · 英文底本 p.3

由式（2）和式（3）可清楚看出，两种图像形成模型相同，但渲染算法迥异。NeRF是隐式描述空/占据空间的连续表示，需要昂贵的随机采样寻找式（2）中的样本，因而带来噪声与计算成本。点则是无结构离散表示，其灵活性足以像NeRF那样创建、删除和移动几何。已有工作表明，优化不透明度与位置即可做到这一点，同时避免完整体表示的缺点。

原段引文：(Kopanas et al., 2021)

S019 · 英文底本 p.3

Pulsar实现了快速球体光栅化，启发了我们的分块排序渲染器。但依据上述分析，我们希望对有序溅射保留近似传统的α混合，以维持体表示的优势：与其顺序无关方法不同，我们的光栅化遵守可见性顺序。此外，我们对一个像素中的全部溅射反传梯度，并光栅化各向异性溅射；这些要素共同提升了视觉质量（7.3节）。上述不少方法用CNN渲染，会产生时间不稳定。尽管如此，Pulsar与ADOP的速度仍是我们开发快速渲染方案的动机。

原段引文：(Lassner and Zollhofer, 2021)；(Rückert et al., 2022)

S020 · 英文底本 p.3

Neural Point Catacaustics主要研究镜面效应，其漫反射点渲染分支通过MLP克服了时间不稳定，但仍要求MVS几何输入。该类最新方法无需MVS，也使用SH表示方向，却只能处理单物体场景，且初始化需要掩码。在低分辨率和少量点时虽快，能否扩展到典型数据集场景仍不清楚。我们用三维高斯构建更灵活的场景表示，无需MVS几何，并借助投影高斯的分块渲染算法实现实时渲染。

原段引文：(Kopanas et al., 2022)；(Zhang et al., 2022)；(Knapitsch et al., 2017 ; Hedman et al., 2018 ; Barron et al., 2022)

S021 · 英文底本 p.4

近期一种方法用径向基函数框架下的点表示辐射场，优化中也剪枝和增密点，但仍使用体射线步进，无法达到实时显示速率。

原段引文：(Xu et al., 2022)

S022 · 英文底本 p.4

在人类表演捕捉领域，三维高斯已用于表示采集的人体；近期又与体射线步进结合用于视觉任务。类似背景下还出现了神经体图元。这些研究启发我们选择三维高斯，但它们聚焦单个孤立物体（人体或脸）的重建与渲染，场景深度复杂度较低。相反，各向异性协方差优化、交替优化/密度控制与高效深度排序，使我们可以处理含背景的完整复杂室内外场景，包括深度复杂度很高的场景。

原段引文：(Stoll et al., 2011 ; Rhodin et al., 2015)；(Wang et al., 2023)；(Lombardi et al., 2021)

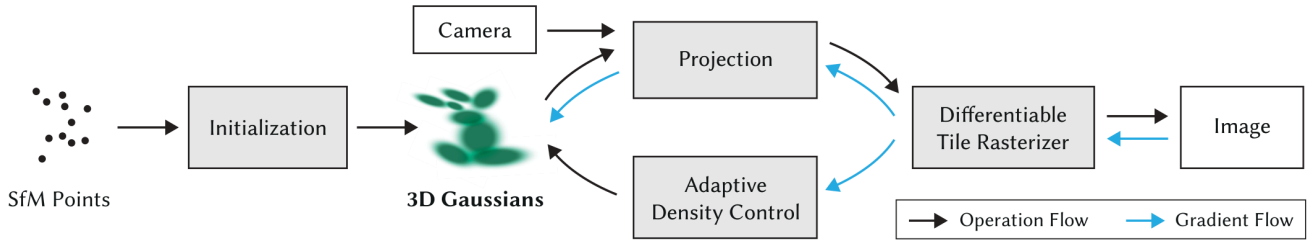

图2. 优化从稀疏SfM点云开始，建立三维高斯集合，随后优化并自适应控制其密度。优化中采用快速分块渲染器，训练时间与SOTA快速辐射场方法相当。训练完成后，渲染器支持多种场景的实时漫游。

3. 方法概览

S023 · 英文底本 p.4

输入是静态场景图像集合及对应的SfM标定相机，SfM同时附带产生稀疏点云。由这些点建立三维高斯集合（第4节），每个高斯由位置（均值）、协方差和不透明度α定义，支持灵活优化。所得三维表示相当紧凑，部分原因是高度各向异性的体溅射能紧凑表示精细结构。辐射场的方向外观（颜色）依照惯例用球谐表示。算法交替优化位置、协方差、α与SH系数，并自适应控制高斯密度，最终得到辐射场（第5节）。效率的关键是分块光栅化器（第6节）：借助快速排序，按可见性顺序执行各向异性溅射的α混合；跟踪累计α又支持快速反向传播，而且不限制可获得梯度的高斯数量。图2给出方法概览。

原段引文：(Schönberger and Frahm, 2016)；(Fridovich-Keil and Yu et al., 2022 ; Müller et al., 2022)

4. 可微三维高斯溅射

S024 · 英文底本 p.4

目标是从不带法线的稀疏SfM点出发，优化出支持高质量新视角合成的场景表示。为此，需要一种兼具可微体表示性质、无结构性和显式性的图元，以实现极快渲染。我们选择可微的三维高斯，它易于投影为二维溅射，并通过快速α混合完成渲染。

S025 · 英文底本 p.4

我们的表示与此前把二维点看作带法线小平面圆盘的方法相似。但SfM点极稀疏，法线很难估计；从这些高噪声估计继续优化法线同样困难。因此，我们以无需法线的三维高斯集合表示几何。高斯在世界空间由完整三维协方差矩阵Σ定义，以点（均值）μ为中心：（4）

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

。混合时再将此高斯乘以α。

原段引文：(Yifan et al., 2019 ; Kopanas et al., 2021)；(Zwicker et al., 2001a)

S026 · 英文底本 p.4

渲染时需把三维高斯投影到二维。Zwicker等（2001a）说明了如何投影到图像空间。给定视图变换W，相机坐标中的协方差Σ'为：（5）

$\Sigma' = JW\,\Sigma\,W^T J^T$

，其中J是投影变换仿射近似的雅可比矩阵。该工作还指出，去掉Σ'的第三行、第三列，即得到2×2方差矩阵，其结构和性质与从带法线平面点出发所得矩阵相同。

原段引文：（ 2001a ）；（ Kopanas et al., 2021 ）

S027 · 英文底本 p.4

直接优化协方差矩阵Σ似乎是得到辐射场三维高斯的自然方案。但协方差只有半正定时才有物理意义。我们对所有参数采用梯度下降，难以施加这一约束，更新和梯度很容易产生无效协方差矩阵。

S028 · 英文底本 p.4

因此，我们选择了更直观而表达能力等价的优化表示。三维高斯的协方差Σ类似于描述椭球形态。给定缩放矩阵S和旋转矩阵R，对应协方差Σ可写为：（6）

$\Sigma = RSS^T R^T$

。

S029 · 英文底本 p.4

为独立优化这两个因素，分别保存三维缩放向量s与表示旋转的四元数q。二者容易转换成矩阵并组合；需要归一化q以得到有效单位四元数。

S030 · 英文底本 p.4

为避免训练中自动微分带来的显著开销，我们显式推导所有参数的梯度。精确导数计算见附录A。

S031 · 英文底本 p.4

这种适于优化的各向异性协方差表示，使三维高斯能适应采集场景中不同形状的几何，形成相当紧凑的表示。图3展示了相应实例。



图3. 最右列把优化后的三维高斯缩小60%以作可视化，清楚展示各向异性形状如何紧凑表示复杂几何。左列为实际渲染图像。

5. 三维高斯优化与自适应密度控制

S032 · 英文底本 p.4

核心优化步骤创建密集的三维高斯集合，准确表示场景以进行自由视角合成。除位置p、α与协方差Σ外，还优化表示各高斯颜色c的SH系数，以正确捕获视角相关外观。参数优化与高斯密度控制交替执行，以更好地表示场景。

5.1. 优化

S033 · 英文底本 p.5

优化不断渲染并把结果与采集数据集中的训练视图比较。三维到二维投影存在歧义，几何不可避免会放错位置。因此优化必须能够创建几何，也能删除或移动放错的几何。三维高斯协方差参数的质量对紧凑性至关重要：大面积均匀区域可用少数大型各向异性高斯表示。

S034 · 英文底本 p.5

我们采用随机梯度下降技术，充分利用标准GPU加速框架，并按近期实践为部分操作添加自定义CUDA核。尤其是快速光栅化（第6节）决定优化效率，因为它是主要计算瓶颈。

原段引文：（ Fridovich-Keil and Yu et al., 2022 ; Sun et al., 2022 ）

S035 · 英文底本 p.5

对α使用sigmoid激活，把它约束在$[0-1)$范围并获得平滑梯度；出于类似原因，对协方差尺度使用指数激活。

S036 · 英文底本 p.5

初始协方差估计为各向同性高斯，其各轴取最近三个点的平均距离。采用类似Plenoxels的标准指数衰减调度，但仅作用于位置。损失由$\mathcal{L}_1$与D-SSIM项组合：

原段引文：(Fridovich-Keil and Yu et al., 2022)

S037 · 英文底本 p.1

(7)

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

S038 · 英文底本 p.5

所有实验均采用$\lambda = 0.2$。学习调度及其他细节见7.1节。

5.2. 高斯自适应控制

S039 · 英文底本 p.5

从SfM初始稀疏点集出发，自适应控制高斯数量及其单位体积密度，使稀疏高斯逐渐变为更密集、参数正确而更适合场景的集合。这里“高斯的密度”不要与NeRF文献的密度σ混淆。优化预热后（7.1节），每100次迭代增密一次，并删除几乎透明、即α低于阈值ϵ_α的高斯。

S040 · 英文底本 p.5

自适应控制需要填补空区域，既关注缺失几何特征的“欠重建”区域，也关注高斯覆盖场景大面积的“过重建”区域。我们观察到二者的视图空间位置梯度都较大。直观上，这可能因为这些区域尚未重建好，优化正尝试移动高斯加以纠正。

S041 · 英文底本 p.5

这两类情况都适合增密，因此对视图空间位置梯度平均幅值高于阈值τ_{pos}的高斯进行增密；实验把阈值设为0.0002。

S042 · 英文底本 p.5

下面给出该过程的细节，图4作了说明。

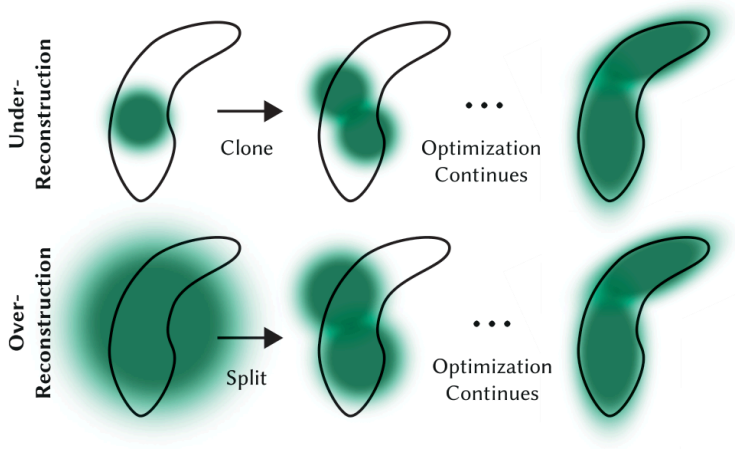

图4. 自适应高斯增密方案。上排（欠重建）：小尺度几何（黑轮廓）覆盖不足时，克隆相应高斯。下排（过重建）：若小尺度几何由一个大型溅射表示，则将其拆为两个。

S043 · 英文底本 p.5

对欠重建区域的小高斯，需要覆盖新建几何；更合适的方法是克隆：生成同尺寸副本，并沿位置梯度方向移动。

S044 · 英文底本 p.5

对高方差区域的大高斯，需要拆成更小高斯。我们以两个新高斯替换它，将尺度除以实验确定的因子$\phi = 1.6$；新位置通过把原三维高斯当作概率密度函数采样来初始化。

S045 · 英文底本 p.5

第一种情况需要同时增加系统总体积与高斯数量，第二种则增加数量而保持总体积。与其他体表示一样，优化可能陷入输入相机附近的漂浮物，导致高斯密度无依据地增加。缓和数量增长的有效方法是每$N = 3000$次迭代把α设为接近零。随后优化只在需要的位置增大α，使上述裁剪策略可以移除α低于ϵ_α的高斯。高斯会缩小、长大并大量重叠；我们还定期删除世界空间过大或视图空间足迹过大的高斯，以总体控制数量。模型中的高斯始终是欧氏空间图元；不同于其他方法，我们不需要为空间压缩、扭曲或投影设计特殊策略来处理远处或大型高斯。

原段引文：(Barron et al., 2022 ; Fridovich-Keil and Yu et al., 2022)

6. 高斯快速可微光栅化

S046 · 英文底本 p.6

我们的目标是整体渲染快、排序快，以支持包括各向异性溅射的近似α混合，并避免此前工作对可接收梯度的溅射数量施加硬限制。

原段引文：(Lassner and Zollhofer, 2021)

S047 · 英文底本 p.6

为此，我们受近期软件光栅化启发，设计分块高斯溅射光栅化器，一次为整幅图像预排序图元，避免此前α混合方案中昂贵的逐像素排序。快速光栅化器以很少的额外内存，对任意数量的混合高斯进行高效反传，每像素只需常数开销。整条管线完全可微，结合第4节的二维投影，可像此前二维溅射方法一样光栅化各向异性溅射。

原段引文：(Lassner and Zollhofer, 2021)；(Kopanas et al., 2021 ; Kopanas et al., 2022)；(Kopanas et al., 2021)

S048 · 英文底本 p.6

首先将屏幕划为16×16像素块，再依据视锥和各块裁剪三维高斯；仅保留99%置信区间与视锥相交的高斯。另外，用保护带直接拒绝极端位置的高斯，即均值接近近裁剪面、又远离视锥者，因为其投影二维协方差计算不稳定。随后按每个高斯重叠的块数复制实例，为每个实例分配结合视图空间深度与块ID的键；用一次快速GPU基数排序完成排序。此后不再逐像素重排，而是按初始顺序混合，因此某些配置中的α混合只是近似。但随着溅射接近单像素大小，近似误差变得可忽略。此选择显著提升训练与渲染性能，且在收敛场景中未见可见伪影。

原段引文：(Merrill and Grimshaw, 2010)

S049 · 英文底本 p.6

排序后，识别向各块溅射的首末深度有序条目，为每块生成列表。光栅化时每块启动一个线程块，协作把成包高斯载入共享内存；对给定像素从前向后遍历列表，累加颜色和α，使数据加载/共享与处理均获得最大并行收益。当像素的α达到目标饱和度，该线程停止；定期查询块内线程，全部像素饱和（即α趋于1）后终止整块。排序细节与光栅化高层概览见附录C。

S050 · 英文底本 p.6

光栅化中，α饱和是唯一停止条件。不同于此前工作，我们不限制获得梯度更新的混合图元数，因此可准确学习深度复杂度任意且可变的场景，无需场景专用超参数。反向传播必须恢复前向中每像素的完整混合点序列。一种方案是为每像素在全局内存存储任意长列表；为避免动态内存管理开销，我们重新遍历每块列表，复用前向已排序高斯数组与块范围，并从后向前遍历以便计算梯度。

原段引文：(Kopanas et al., 2021)

S051 · 英文底本 p.6

从最后一个对块内任意像素有影响的点开始遍历，仍协作加载点到共享内存。只有当点深度不大于前向中最后贡献颜色的点深度时，像素才开始昂贵的重叠测试和点处理。第4节梯度计算需要原混合过程每一步的累计不透明度；无需保存并遍历不断缩小的不透明度显式列表，只保存前向末尾总累计值即可恢复中间值。具体而言，原文描述在前向保存最终累计不透明度α，反向从后向前遍历时除以各点的α，得到梯度所需系数。

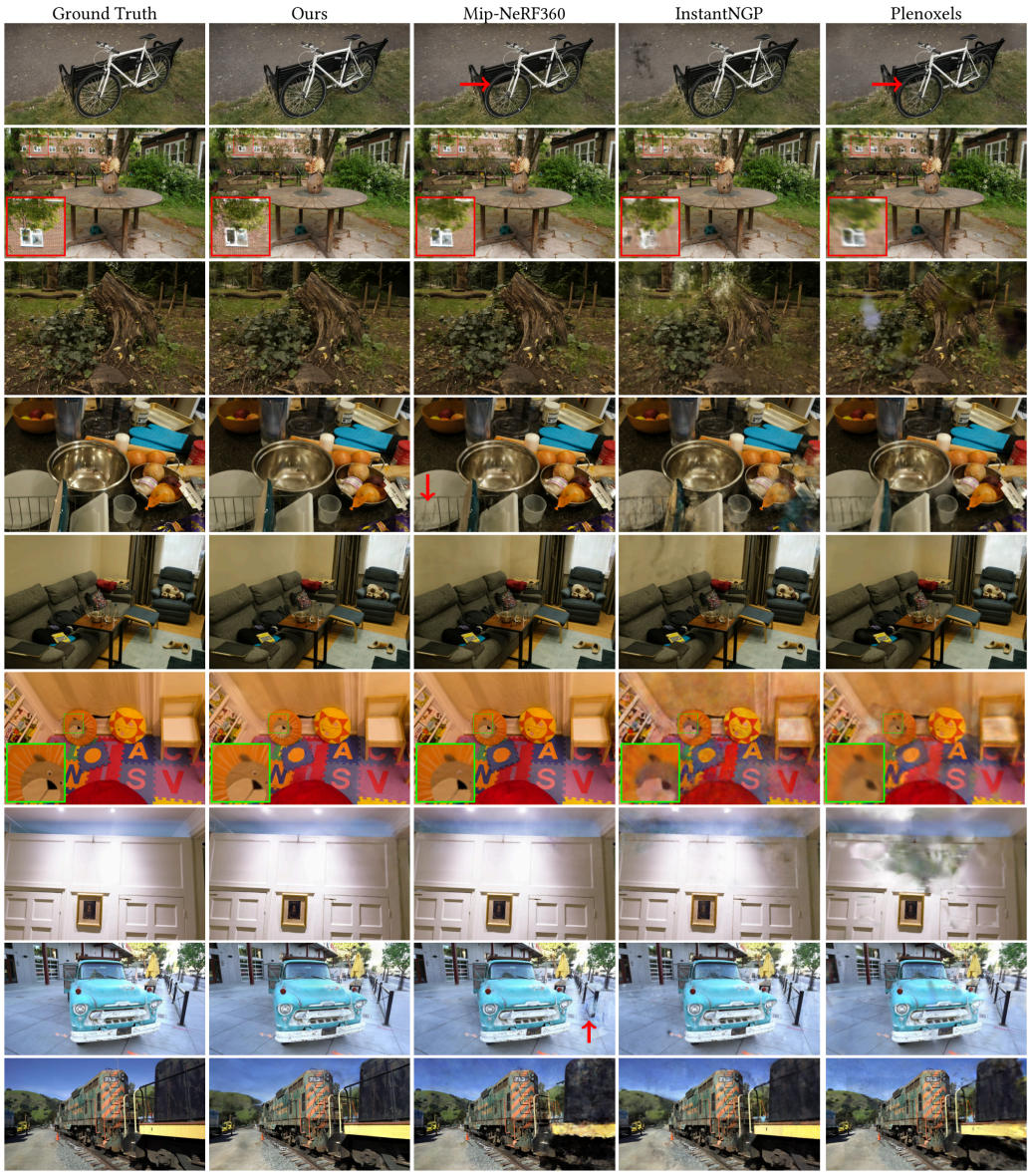

图5. 本方法、此前方法与保留测试视图真值的比较。自上而下为Mip-NeRF360的Bicycle、Garden、Stump、Counter、Room；Deep Blending的Playroom、DrJohnson；Tanks&Temples的Truck、Train。箭头及局部放大图突出不易察觉的质量差异。

原段引文：(Hedman et al., 2018)

表1. 在三个数据集上与此前工作的定量比较。带剑号 † 的结果直接取自原论文，其余均来自我们的实验。

Dataset	Mip-NeRF360						Tanks&Temples						Deep Blending					
Method|Metric	$SSIM^{\uparrow}$	$PSNR^{\uparrow}$	$LPIPS^{\downarrow}$	Train	FPS	Mem	$SSIM^{\uparrow}$	$PSNR^{\uparrow}$	$LPIPS^{\downarrow}$	Train	FPS	Mem	$SSIM^{\uparrow}$	$PSNR^{\uparrow}$	$LPIPS^{\downarrow}$	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	$0.792^{\dagger}$	$27.69^{\dagger}$	$0.237^{\dagger}$	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



图6. 某些场景（上）仅7K次迭代（该场景 ~ 5分钟）就很好地捕获了火车；30K次迭代（ ~ 35分钟）后背景伪影显著减少。其他场景（下）差异几乎不可见，7K次迭代（ ~ 8分钟）已具有很高质量。

7. 实现、结果与评估

S052 · 英文底本 p.6

下面讨论实现细节，展示与此前工作的结果比较、算法评估及消融研究。

表2. Synthetic NeRF的PSNR；从100K个随机初始化点开始。对比指标取自相应论文。

	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

7.1. 实现

S053 · 英文底本 p.8

采用Python与PyTorch实现方法；光栅化使用扩展此前方法的自定义CUDA核，快速基数排序使用NVIDIA CUB例程。我们还用开源SIBR建立交互查看器，并用它测量帧率。源码与全部数据公开于：https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/ 。

原段引文：(Kopanas et al., 2021)；(Merrill and Grimshaw, 2010)；(Bonopera et al., 2020)

优化细节

S054 · 英文底本 p.8

为提高稳定性，用较低分辨率预热计算：以小四倍的图像分辨率开始，在第250和500次迭代后各上采样一次。

S055 · 英文底本 p.8

SH系数优化对角度信息不足很敏感。典型NeRF式采集在整个半球围绕中心物体拍照，优化效果良好；若采集缺失某些角度（例如场景角落或由内向外采集），SH零阶分量，即基础/漫反射颜色，可能被优化为完全错误的值。为此，先仅优化零阶分量，此后每1000次迭代加入一个SH频带，直至包含全部四个频带。

原段引文：(Hedman et al., 2016)

表3. 消融实验PSNR。本实验手动将各场景高分辨率输入图像下采样至其他实验约定的渲染分辨率，以减少随机伪影（例如Mip-NeRF360预下采样输入中的JPEG压缩伪影）。

	Truck-5K	Garden-5K	Bicycle-5K	Truck-30K	Garden-30K	Bicycle-30K	Average-5K	Average-30K
Limited-BW	14.66	22.07	20.77	13.84	22.88	20.87	19.16	19.19
Random Init	16.75	20.90	19.86	18.02	22.19	21.05	19.17	20.42
No-Split	18.31	23.98	22.21	20.59	26.11	25.02	21.50	23.90
No-SH	22.36	25.22	22.88	24.39	26.59	25.08	23.48	25.35
No-Clone	22.29	25.61	22.15	24.82	27.47	25.46	23.35	25.91
Isotropic	22.40	25.49	22.81	23.89	27.00	24.81	23.56	25.23
Full	22.71	25.82	23.18	24.81	27.70	25.65	23.90	26.05

7.2. 结果与评估

结果

S056 · 英文底本 p.8

我们在已发表数据集中的13个真实场景与合成Blender数据集上测试。具体使用当前NeRF质量SOTA Mip-NeRF360的全部场景、Tanks&Temples两个场景及Hedman等提供的两个场景。采集方式各异，覆盖有界室内及大型无界室外环境。所有实验使用同一超参数配置。除下述Mip-NeRF360外，结果均在A6000 GPU上运行。

原段引文：(Mildenhall et al., 2020)；(Barron et al., 2022)；(2017)；(Hedman et al., 2018)

S057 · 英文底本 p.8

补充材料给出部分场景的渲染视频路径，其中包含远离输入照片的视角。

真实场景

S058 · 英文底本 p.8

以当前质量SOTA Mip-NeRF360作为质量基准，同时比较两种近期快速NeRF：InstantNGP与Plenoxels。

原段引文：(Barron et al., 2021)；(Müller et al., 2022)；(Fridovich-Keil and Yu et al., 2022)

S059 · 英文底本 p.8

按照Mip-NeRF360建议，每8张照片取1张作测试，保证误差比较一致且有意义；使用文献常见的PSNR、L-PIPS和SSIM（表1）。除Mip-NeRF360在其数据集上的数字为避免当前SOTA歧义而直接取自原论文，其余均为运行各作者代码所得。图中图像使用我们自己运行的Mip-NeRF360，其数值见附录D。另报告平均训练时间、渲染速度与优化参数存储内存。InstantNGP包含运行35K次迭代的基础配置（Base）及作者建议的稍大网络（Big）；本方法报告7K与30K两种配置。图6显示两种配置的视觉差异，许多情况下7K次迭代质量已很好。

S060 · 英文底本 p.9

各数据集的训练时间不同，分别报告；图像分辨率也随数据集变化。项目网站提供所有方法、所有场景用于统计的全部测试视图渲染。所有渲染保持原始输入分辨率。

S061 · 英文底本 p.9

表中，完全收敛模型的质量与SOTA Mip-NeRF360相当，有时略优。同等硬件下，其平均训练时间为48小时，而我们为35–45分钟；其渲染耗时10秒/帧。此处48小时来自4卡A100节点运行12小时折算的单GPU时间，且A100比A6000更快。我们训练5–10分钟即达到InstantNGP与Plenoxels相近质量，继续训练还能达到SOTA，而其他快速方法不能。Tanks&Temples中，在近似训练时间（本方法 ~ 7分钟）下，质量与基础InstantNGP相近。

S062 · 英文底本 p.9

图5比较保留测试视图，本方法使用30K次迭代。某些情况下，Mip-NeRF360仍有我们避免的伪影，例如Bicycle与Stump的植被模糊、Room的墙面模糊。补充视频与网页提供远距离路径比较；本方法往往能在远处保留覆盖良好区域的细节，此前方法并不总能做到。

合成有界场景

S063 · 英文底本 p.9

除真实场景外，还在合成Blender数据集上评估。其视角覆盖全面、场景尺寸有限且相机参数精确，因此即使随机初始化也能达到SOTA。训练从包围场景的体积内100K个均匀随机高斯开始，方法会快速自动剪枝到约6–10K个有意义高斯；30K次迭代后，每场景约有200–500K个高斯。为兼容既有比较，表2使用白背景报告PSNR。示例见图10左数第二幅与补充材料。训练后合成场景的渲染速度为180–300 FPS。

原段引文：(Mildenhall et al., 2020)

紧凑性

S064 · 英文底本 p.9

相较此前显式表示，各向异性高斯可用更少参数描述复杂形状。我们与Zhang等的高度紧凑点模型比较：从其利用前景掩码空间雕刻所得初始点云出发，优化至达到其报告PSNR，通常需2–4分钟。使用约四分之一点数即超过其指标，平均模型3.8 MB，而对方为9 MB。此实验与其相似，仅使用两个阶次的球谐。

原段引文：(Zhang et al., 2022)

7.3. 消融研究

S065 · 英文底本 p.9

我们隔离各项贡献与算法选择，评估SfM初始化、增密策略、各向异性协方差、允许无限数量溅射接收梯度，以及球谐的影响。定量结果汇总于表3。

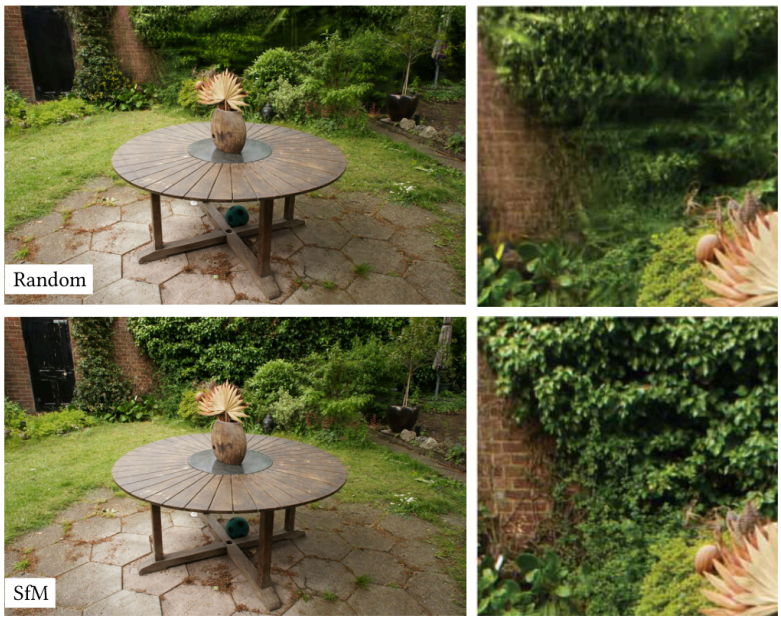

图7. SfM点初始化有帮助。上：随机点云初始化；下：SfM点初始化。

SfM初始化

S066 · 英文底本 p.9

为评估SfM点云初始化的重要性，在边长为输入相机包围盒范围三倍的立方体内均匀采样。即使不用SfM点，本方法仍表现尚可、不会完全失败，主要在背景退化（图7）；训练视图覆盖不佳区域，随机初始化产生更多无法被优化移除的漂浮物。合成NeRF没有此现象，因为它没有背景，且输入相机约束充分。

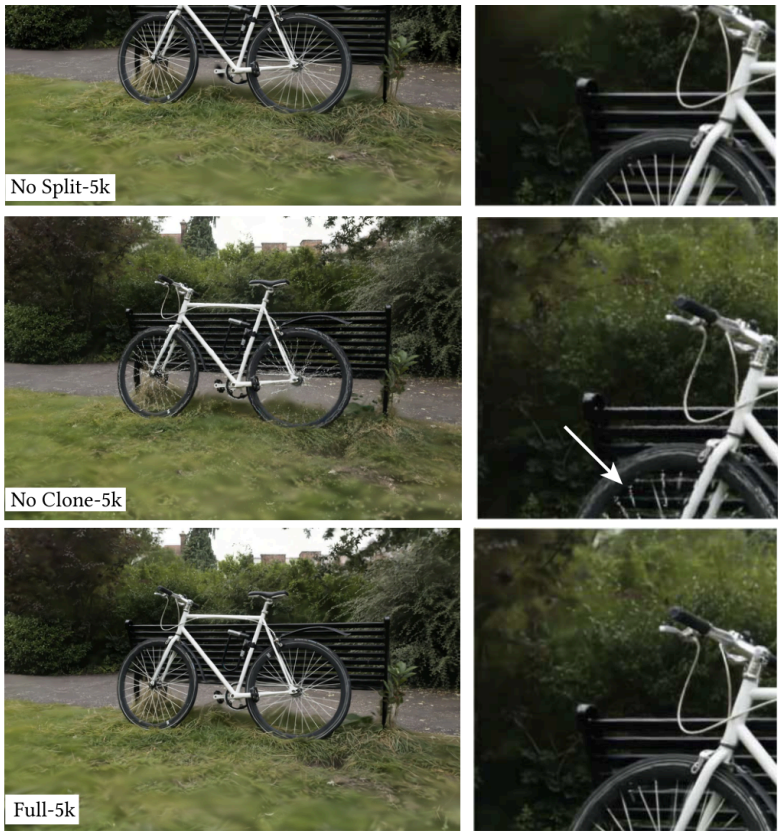

图8. 对“克隆”与“拆分”两种增密策略的消融（第5节）。

增密

S067 · 英文底本 p.9

分别关闭第5节的克隆与拆分策略，其他部分保持不变。结果表明，拆分大高斯对良好重建背景十分重要（图8）；对小高斯执行克隆而非拆分，可获得更好、更快的收敛，尤其在包含细结构的场景。



图9. 限制获得梯度的点数会显著影响视觉质量。左：只允许10个高斯接收梯度；右：完整方法。

梯度溅射不限制深度复杂度

S068 · 英文底本 p.10

依据Pulsar的建议，测试能否在最前面的N个点之后跳过梯度计算，以提速且不损质量。取N=10，已是Pulsar默认值的两倍，但严重梯度近似仍导致优化不稳定。Truck场景PSNR下降11 dB（表3，Limited-BW）；Garden视觉结果见图9。

原段引文：(Lassner and Zollhofer, 2021)

各向异性协方差

S069 · 英文底本 p.10

优化完整三维高斯协方差矩阵是一项重要选择。消融中去掉各向异性，仅优化一个标量，统一控制三个轴的半径。图10显示，各向异性显著提升高斯与表面的对齐能力，使相同点数获得高得多的渲染质量。

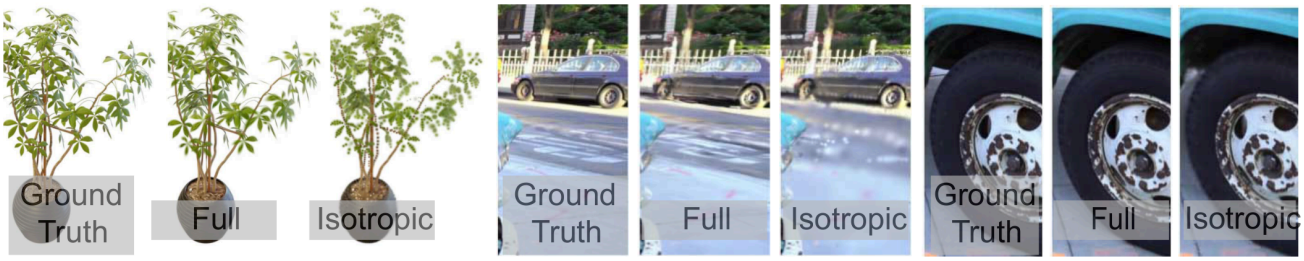

图10. 分别禁用与启用高斯各向异性训练场景。各向异性体溅射可表示精细结构，明显影响视觉质量。为便于展示，Ficus两种配置都限制为最多5K个高斯。

球谐

S070 · 英文底本 p.10

最后，球谐可补偿视角相关效应，因此提高总体PSNR（表3）。

7.4. 局限

S071 · 英文底本 p.10

本方法仍有限制。观测不足区域会出现伪影，其他方法也有困难（例如图11的Mip-NeRF360）。尽管各向异性高斯有上述优点，仍可能形成细长伪影或斑块状高斯（图12），此前方法在这些情况也很困难。

S072 · 英文底本 p.10

优化产生大型高斯时偶尔会出现突变伪影，常发生在外观随视角变化的区域。一个原因是光栅化器用保护带直接拒绝高斯，更有原则的裁剪可缓解；另一个原因是简单可见性算法可能让高斯突然交换深度/混合顺序，可通过抗混叠处理，留待未来研究。目前优化未使用正则化；加入正则化有望改善未见区域与突变伪影。

S073 · 英文底本 p.10

虽然完整评估使用相同超参数，早期实验显示，极大场景（例如城市数据）可能需要降低位置学习率才能收敛。

S074 · 英文底本 p.11

相对此前点方法我们很紧凑，但内存显著高于NeRF方案。未优化原型训练大场景时，GPU峰值可超过20 GB；如InstantNGP那样细致底层实现优化逻辑，可显著降低。渲染时需存下完整模型，大场景为数百MB；光栅化器另需30–500 MB，取决于场景尺寸与分辨率。本方法仍有很多压缩机会；点云压缩研究成熟，值得探索如何适配此表示。

原段引文：(De Queiroz and Chou, 2016)

see how such approaches could be adapted to our representation.



图11. 失败伪影比较。Mip-NeRF360出现“漂浮物”与颗粒感（左、前景）；本方法出现粗大的各向异性高斯，造成低细节外观（右、背景）。场景为Train。



图12. 与训练视图重叠很少的视角中，本方法可能产生伪影（右）；Mip-NeRF360也会出现伪影（左）。场景为DrJohnson。

8. 讨论与结论

S075 · 英文底本 p.11

我们提出首个真正能在广泛场景与采集风格下实现实时、高质量辐射场渲染的方法，训练时间与此前最快方法竞争。

S076 · 英文底本 p.11

三维高斯图元既保留用于优化的体渲染性质，又直接支持快速溅射光栅化。结果表明，与广泛接受的看法相反，连续表示并非快速、高质量辐射场训练的严格必要条件。

S077 · 英文底本 p.11

为便于他人使用，我们以PyTorch构建方案，大部分（～80%）训练时间花在Python代码上，仅光栅化是优化CUDA核。若像InstantNGP那样把其余优化全部移植至CUDA，预期可进一步显著提速，适用于性能至关重要的应用。

原段引文：（Müller et al., 2022）

S078 · 英文底本 p.11

我们也展示了实时渲染原则、GPU能力与软件光栅化管线架构速度的重要性。这些选择同时决定训练与实时渲染性能，使其在性能上优于此前体射线步进。

S079 · 英文底本 p.11

探索能否由这些高斯重建场景网格很有意义。网格应用广泛，因此有实际价值；同时还能帮助理解本方法在体表示与表面表示连续谱中的确切位置。

S080 · 英文底本 p.11

总之，我们给出首个辐射场实时渲染方案：质量匹敌此前最优但昂贵的方法，训练时间则与现有最快方案相当。

致谢

本研究由ERC高级项目FUNGRAPH No.788065资助（http://fungraph.inria.fr）。感谢Adobe的慷慨捐赠、Université Côte d’Azur的OPAL基础设施和GENCI–IDRIS高性能计算资源（2022-AD011013409）。感谢匿名评审的宝贵意见，P. Hedman和A. Tewari校读早期稿件，以及T. Müller、A. Yu和S. Fridovich-Keil协助比较。

参考文献（保留英文书目）

- **Aliev et al. (2020)** Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. 2020. Neural Point-Based Graphics. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII*. 696–712.
- **Barron et al. (2021)** Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. 2021. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5855–5864.
- **Barron et al. (2022)** Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2022. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. *CVPR* (2022).
- **Bonopera et al. (2020)** Sebastien Bonopera, Jerome Esnault, Siddhant Prakash, Simon Rodriguez, Theo Thonat, Mehdi Benadel, Gaurav Chaurasia, Julien Philip, and George Drettakis. 2020. sibr: A System for Image Based Rendering. https://gitlab.inria.fr/sibr/sibr_core
- **Botsch et al. (2005)** Mario Botsch, Alexander Hornung, Matthias Zwicker, and Leif Kobbelt. 2005. High-Quality Surface Splatting on Today's GPUs. In *Proceedings of the Second Eurographics / IEEE VGTC Conference on Point-Based Graphics* (New York, USA) *(SPBG'05)*. Eurographics Association, Goslar, DEU, 17–24.
- **Buehler et al. (2001)** Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. 2001. Unstructured lumigraph rendering. In *Proc. SIGGRAPH*.
- **Chaurasia et al. (2013)** Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. 2013. Depth synthesis and local warps for plausible image-based navigation. *ACM Transactions on Graphics (TOG)* 32, 3 (2013), 1–12.
- **Chen et al. (2022b)** Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. 2022b. TensoRF: Tensorial Radiance Fields. In *European Conference on Computer Vision (ECCV)*.
- **Chen et al. (2022a)** Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. 2022a. MobileNeRF: Exploiting the Polygon Rasterization Pipeline for Efficient Neural Field Rendering on Mobile Architectures. *arXiv preprint arXiv:2208.00277* (2022).
- **De Queiroz and Chou (2016)** Ricardo L De Queiroz and Philip A Chou. 2016. Compression of 3D point clouds using a region-adaptive hierarchical transform. *IEEE Transactions on Image Processing* 25, 8 (2016), 3947–3956.
- **Eisemann et al. (2008)** Martin Eisemann, Bert De Decker, Marcus Magnor, Philippe Bekaert, Edilson De Aguiar, Naveed Ahmed, Christian Theobalt, and Anita Sellent. 2008. Floating textures. In *Computer graphics forum*, Vol. 27. Wiley Online Library, 409–418.
- **Flynn et al. (2016)** John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. 2016. Deepstereo: Learning to predict new views from the world's imagery. In *CVPR*.
- **Fridovich-Keil and Yu et al. (2022)** Fridovich-Keil and Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. 2022. Plenoxels: Radiance Fields without Neural Networks. In *CVPR*.
- **Garbin et al. (2021)** Stephan J. Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. 2021. FastNeRF: High-Fidelity Neural Rendering at 200FPS. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 14346–14355.
- **Goesele et al. (2007)** Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M Seitz. 2007. Multi-view stereo for community photo collections. In *ICCV*.
- **Gortler et al. (1996)** Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. 1996. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 43–54.
- **Gross and Pfister (2011)** Markus Gross and Hanspeter (Eds) Pfister. 2011. *Point-based graphics*. Elsevier.
- **Grossman and Dally (1998)** Jeff P. Grossman and William J. Dally. 1998. Point Sample Rendering. In *Rendering Techniques*.
- **Hedman et al. (2018)** Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. 2018. Deep blending for free-viewpoint image-based rendering. *ACM Trans. on Graphics (TOG)* 37, 6 (2018).
- **Hedman et al. (2016)** Peter Hedman, Tobias Ritschel, George Drettakis, and Gabriel Brostow. 2016. Scalable Inside-Out Image-Based Rendering. *ACM Transactions on Graphics (SIGGRAPH Asia Conference Proceedings)* 35, 6 (December 2016). http://www-sop.inria.fr/reves/Basilic/2016/HRDB16
- **Hedman et al. (2021)** Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. 2021. Baking Neural Radiance Fields for Real-Time View Synthesis. *ICCV* (2021).
- **Henzler et al. (2019)** Philipp Henzler, Niloy J Mitra, and Tobias Ritschel. 2019. Escaping plato's cave: 3d shape from adversarial rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9984–9993.
- **Knapitsch et al. (2017)** Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. 2017. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)* 36, 4 (2017), 1–13.
- **Kopanas et al. (2022)** Georgios Kopanas, Thomas Leimkühler, Gilles Rainer, Clément Jambon, and George Drettakis. 2022. Neural Point Catacaustics for Novel-View Synthesis of Reflections. *ACM Transactions on Graphics (SIGGRAPH Asia Conference Proceedings)* 41, 6 (2022), 201. http://www-sop.inria.fr/reves/Basilic/2022/KLRJD22

- **Kopanas et al. (2021)** Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. 2021. Point-Based Neural Rendering with Per-View Optimization. *Computer Graphics Forum* 40, 4 (2021), 29–43. https://doi.org/10.1111/cgf.14339
- **Laine and Karras (2011)** Samuli Laine and Tero Karras. 2011. High-performance software rasterization on GPUs. In *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics*. 79–88.
- **Lassner and Zollhofer (2021)** Christoph Lassner and Michael Zollhofer. 2021. Pulsar: Efficient Sphere-Based Neural Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1440–1449.
- **Levoy and Hanrahan (1996)** Marc Levoy and Pat Hanrahan. 1996. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 31–42.
- **Lombardi et al. (2021)** Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. 2021. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–13.
- **Merrill and Grimshaw (2010)** Duane G Merrill and Andrew S Grimshaw. 2010. Revisiting sorting for GPGPU stream architectures. In *Proceedings of the 19th international conference on Parallel architectures and compilation techniques*. 545–546.
- **Mildenhall et al. (2020)** Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.
- **Müller et al. (2022)** Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Trans. Graph.* 41, 4, Article 102 (July 2022), 15 pages. https://doi.org/10.1145/3528223.3530127
- **Penner and Zhang (2017)** Eric Penner and Li Zhang. 2017. Soft 3D reconstruction for view synthesis. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–11.
- **Pfister et al. (2000)** Hanspeter Pfister, Matthias Zwicker, Jeroen van Baar, and Markus Gross. 2000. Surfels: Surface Elements as Rendering Primitives. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*. ACM Press/Addison-Wesley Publishing Co., USA, 335–342. https://doi.org/10.1145/344779.344936
- **Reiser et al. (2021)** Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. 2021. KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs. In *International Conference on Computer Vision (ICCV)*.
- **Ren et al. (2002)** Liu Ren, Hanspeter Pfister, and Matthias Zwicker. 2002. Object Space EWA Surface Splatting: A Hardware Accelerated Approach to High Quality Point Rendering. *Computer Graphics Forum* 21 (2002).
- **Rhodin et al. (2015)** Helge Rhodin, Nadia Robertini, Christian Richardt, Hans-Peter Seidel, and Christian Theobalt. 2015. A versatile scene model with differentiable visibility applied to generative pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*. 765–773.
- **Riegler and Koltun (2020)** Gernot Riegler and Vladlen Koltun. 2020. Free view synthesis. In *European Conference on Computer Vision*. Springer, 623–640.
- **Rückert et al. (2022)** Darius Rückert, Linus Franke, and Marc Stamminger. 2022. ADOP: Approximate Differentiable One-Pixel Point Rendering. *ACM Trans. Graph.* 41, 4, Article 99 (jul 2022), 14 pages. https://doi.org/10.1145/3528223.3530122
- **Sainz and Pajarola (2004)** Miguel Sainz and Renato Pajarola. 2004. Point-based rendering techniques. *Computers and Graphics* 28, 6 (2004), 869–879. https://doi.org/10.1016/j.cag.2004.08.014
- **Schönberger and Frahm (2016)** Johannes Lutz Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- **Schütz et al. (2022)** Markus Schütz, Bernhard Kerbl, and Michael Wimmer. 2022. Software Rasterization of 2 Billion Points in Real Time. *Proc. ACM Comput. Graph. Interact. Tech.* 5, 3, Article 24 (jul 2022), 17 pages. https://doi.org/10.1145/3543863
- **Sitzmann et al. (2019)** Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. 2019. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2437–2446.
- **Snavely et al. (2006)** Noah Snavely, Steven M Seitz, and Richard Szeliski. 2006. Photo tourism: exploring photo collections in 3D. In *Proc. SIGGRAPH*.
- **Stoll et al. (2011)** Carsten Stoll, Nils Hasler, Juergen Gall, Hans-Peter Seidel, and Christian Theobalt. 2011. Fast articulated motion tracking using a sums of gaussians body model. In *2011 International Conference on Computer Vision*. IEEE, 951–958.
- **Sun et al. (2022)** Cheng Sun, Min Sun, and Hwann-Tzong Chen. 2022. Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction. In *CVPR*.
- **Takikawa et al. (2022)** Towaki Takikawa, Alex Evans, Jonathan Tremblay, Thomas Müller, Morgan McGuire, Alec Jacobson, and Sanja Fidler. 2022. Variable bitrate neural fields. In *ACM SIGGRAPH 2022 Conference Proceedings*. 1–9.
- **Takikawa et al. (2021)** Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. 2021. Neural Geometric Level of Detail: Real-time Rendering with Implicit 3D Shapes. (2021).
- **Tewari et al. (2022)** Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, W Yifan, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, et al. 2022. Advances in neural rendering. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 703–735.
- **Thies et al. (2019)** Justus Thies, Michael Zollhöfer, and Matthias Nießner. 2019. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–12.

- **Wang et al. (2023)** Angtian Wang, Peng Wang, Jian Sun, Adam Kortylewski, and Alan Yuille. 2023. VoGE: A Differentiable Volume Renderer using Gaussian Ellipsoids for Analysis-by-Synthesis. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=AdPJb9cud_Y
- **Wiles et al. (2020)** Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. 2020. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7467–7477.
- **Wu et al. (2022)** Xiuchao Wu, Jiamin Xu, Zihan Zhu, Hujun Bao, Qixing Huang, James Tompkin, and Weiwei Xu. 2022. Scalable Neural Indoor Scene Rendering. *ACM Transactions on Graphics (TOG)* (2022).
- **Xie et al. (2022)** Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. 2022. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 641–676.
- **Xu et al. (2022)** Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. 2022. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5438–5448.
- **Yifan et al. (2019)** Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. 2019. Differentiable surface splatting for point-based geometry processing. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–14.
- **Yu et al. (2021)** Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. 2021. PlenOctrees for Real-time Rendering of Neural Radiance Fields. In *ICCV*.
- **Zhang et al. (2022)** Qiang Zhang, Seung-Hwan Baek, Szymon Rusinkiewicz, and Felix Heide. 2022. Differentiable Point-Based Radiance Fields for Efficient View Synthesis. In *SIGGRAPH Asia 2022 Conference Papers* (Daegu, Republic of Korea) *(SA '22)*. Association for Computing Machinery, New York, NY, USA, Article 7, 12 pages. https://doi.org/10.1145/3550469.3555413
- **Zhou et al. (2016)** Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. 2016. View synthesis by appearance flow. In *European conference on computer vision*. Springer, 286–301.
- **Zwicker et al. (2001a)** Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. 2001a. EWA volume splatting. In *Proceedings Visualization, 2001. VIS'01*. IEEE, 29–538.
- **Zwicker et al. (2001b)** Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus Gross. 2001b. Surface Splatting. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. Association for Computing Machinery, New York, NY, USA, 371–378. https://doi.org/10.1145/383259.383300

附录A 梯度计算细节

S081 · 英文底本 p.13

回顾记号：$\Sigma/\Sigma^{'}$分别是世界/视图空间高斯协方差，q为旋转，s为缩放，W为视图变换，J为投影变换仿射近似的雅可比。用链式法则得到对缩放、旋转的导数：（8）

$$\frac{d\Sigma^{'}}{ds}=\frac{d\Sigma^{'}}{d\Sigma}\frac{d\Sigma}{ds}$$

，以及（9）

$$\frac{d\Sigma^{'}}{dq}=\frac{d\Sigma^{'}}{d\Sigma}\frac{d\Sigma}{dq}$$

。令$U=JW$，并令$\Sigma^{'}$为$U\Sigma U^{T}$左上角对称2×2矩阵，可简化式（5）；用下标表示矩阵元素，得到偏导$\frac{\partial\Sigma^{'}}{\partial\Sigma_{ij}}=\begin{pmatrix} U_{1,i}U_{1,j} & U_{1,i}U_{2,j} \\ U_{1,j}U_{2,i} & U_{2,i}U_{2,j} \end{pmatrix}$。

S082 · 英文底本 p.13

接着求$\frac{d\Sigma}{ds}$和$\frac{d\Sigma}{dq}$。因$\Sigma=RSS^{T}R^{T}$，可计算$M=RS$并改写$\Sigma=MM^{T}$，于是有$\frac{d\Sigma}{ds}=\frac{d\Sigma}{dM}\frac{dM}{ds}$和$\frac{d\Sigma}{dq}=\frac{d\Sigma}{dM}\frac{dM}{dq}$。协方差$\Sigma$及其梯度对称，公共第一部分可紧凑写为$\frac{d\Sigma}{dM}=2M^{T}$。缩放还满足$\frac{\partial M_{i,j}}{\partial s_k}=\begin{cases} R_{i,k} & \text{if j = k} \\ 0 & \text{otherwise} \end{cases}$。推导旋转梯度时，回顾由单位四元数$q$（实部$q_r$、虚部$q_i,q_j,q_k$）转为旋转矩阵$R$的公式：（10）

$$R(q)=2\begin{pmatrix} \frac{1}{2}-(q_j^2+q_k^2) & (q_iq_j-q_rq_k) & (q_iq_k+q_rq_j) \\ (q_iq_j+q_rq_k) & \frac{1}{2}-(q_i^2+q_k^2) & (q_jq_k-q_rq_i) \\ (q_iq_k-q_rq_j) & (q_jq_k+q_rq_i) & \frac{1}{2}-(q_i^2+q_j^2) \end{pmatrix}$$

。由此得到q各分量的梯度：（11）$\frac{\partial M}{\partial q_r}=2\begin{pmatrix}0 & -s_y q_k & s_z q_j\\ s_x q_k & 0 & -s_z q_i\\ -s_x q_j & s_y q_i & 0\end{pmatrix}$，$\frac{\partial M}{\partial q_i}=2\begin{pmatrix}0 & s_y q_j & s_z q_k\\ s_x q_j & -2s_y q_i & -s_z q_r\\ s_x q_k & s_y q_r & -2s_z q_i\end{pmatrix}$ $\frac{\partial M}{\partial q_j}=2\begin{pmatrix}-2s_x q_j & s_y q_i & s_z q_r\\ s_x q_i & 0 & s_z q_k\\ -s_x q_r & s_y q_k & -2s_z q_j\end{pmatrix}$，$\frac{\partial M}{\partial q_k}=2\begin{pmatrix}-2s_x q_k & -s_y q_r & s_z q_i\\ s_x q_r & -2s_y q_k & s_z q_j\\ s_x q_i & s_y q_j & 0\end{pmatrix}$。四元数归一化的梯度可直接求出。

附录B 优化与增密算法

S083 · 英文底本 p.13

优化与增密算法汇总于算法1。

算法1. 优化与增密。w、h为训练图像宽度和高度。

$M \leftarrow$ SfM点 ▷ 位置
$S, C, A \leftarrow$ InitAttributes() ▷ 协方差、颜色、不透明度
$i \leftarrow 0$ ▷ 迭代计数
while 尚未收敛 **do**
  $V, \hat{I} \leftarrow$ SampleTrainingView() ▷ 相机 V 及图像
  $I \leftarrow$ Rasterize(M, S, C, A, V) ▷ 算法 2
  $L \leftarrow Loss(I, \hat{I})$ ▷ 损失
  M, S, C, $A \leftarrow$ Adam(∇L) ▷ 反传与更新
  if IsRefinementIteration(i) **then**
   for all Gaussians (μ, Σ, c, α) **in** (M, S, C, A) **do**
    if $\alpha < \epsilon$ or IsTooLarge(μ, Σ) **then** ▷ 剪枝
     RemoveGaussian()
    end if
    if $\nabla_p L > \tau_p$ **then** ▷ 增密
     if $\|S\| > \tau_S$ **then** ▷ 过重建
      SplitGaussian(μ, Σ, c, α)
     else ▷ **欠重建**
      CloneGaussian(μ, Σ, c, α)
     end if
    end if
   end for
  end if
  $i \leftarrow i + 1$
end while

附录C 光栅化器细节

排序

S084 · 英文底本 p.13

设计假设有大量小溅射，因此每帧开始仅用一次基数排序。屏幕划为16×16像素块，对每个溅射在其重叠的各16×16块中建立实例，构造逐块列表。待处理高斯数适度增加，但更简单的控制流与优化GPU基数排序的高并行性可摊薄代价。每实例分配最多64位键，低32位编码投影深度，高位编码块索引；索引的实际大小取决于当前分辨率的块数。因此一次基数排序即可并行解决全部溅射深度顺序。排序后，通过识别相同块ID在有序数组中的首末位置，生成各块高斯列表；每个64位元素启动一线程，比较其高32位与两个邻居，完成并行操作。相较Pulsar，完全去掉串行图元处理步骤，并得到更紧凑的前向逐块遍历列表。算法2给出高层流程。

原段引文：（ Merrill and Grimshaw, 2010 ）；（ Lassner and Zollhofer, 2021 ）

算法2. 三维高斯GPU软件光栅化。w、h为待光栅化图像宽、高；M、S为世界空间高斯均值、协方差；C、A为高斯颜色、不透明度；V为当前相机视图配置。

```
function Rasterize(w, h, M, S, C, A, V)
  CullGaussian(p, V)  ▷ 视锥裁剪
  M',S' ← ScreenspaceGaussians(M, S, V)  ▷ 变换
  T ← CreateTiles(w, h)
  L, K ← DuplicateWithKeys(M', T)  ▷ 索引与排序键
  SortByKeys(K, L)  ▷ 全局排序
  R ← IdentifyTileRanges(T, K)
  I ← 0  ▷ 初始化画布
  for all 图块 t in I do
   for all 像素 i in t do
     r ← GetTileRange(R, t)
     I[i] ← BlendInOrder(i, L, r, K, M', S', C, A)
   end for
  end forreturn I
end function
```

数值稳定性

S085 · 英文底本 p.14

反向传播反复用前向累计不透明度除以各高斯的α，重建梯度所需中间值。朴素实现易数值不稳定，例如除零。为此，前后向均跳过$\alpha < \epsilon$的混合更新，取ϵ为$\frac{1}{255}$；同时将α上限钳制为0.99。此外，前向纳入高斯前先计算加入后累计不透明度，并在超过0.9999之前停止从前向后的混合。

附录D 各场景误差指标

S086 · 英文底本 p.14

表4–9列出所有比较方法和真实场景的各项误差指标。同时列出直接引用的Mip-NeRF360数字，以及生成本文图像所用自行运行结果；后者在完整Mip-NeRF360上的平均PSNR为27.58、SSIM为0.790、LPIPS为0.240。

表4. Mip-NeRF360各场景SSIM。†表示取自原论文。

	bicycle	flowers	garden	stump	treehill	room	counter	kitchen	bonsai
Plenoxels	0.496	0.431	0.6063	0.523	0.509	0.8417	0.759	0.648	0.814
INGP-Base	0.491	0.450	0.649	0.574	0.518	0.855	0.798	0.818	0.890
INGP-Big	0.512	0.486	0.701	0.594	0.542	0.871	0.817	0.858	0.906
Mip-NeRF360†	0.685	0.583	0.813	0.744	0.632	0.913	0.894	0.920	**0.941**
Mip-NeRF360	0.685	0.584	0.809	0.745	0.631	0.910	0.892	0.917	0.938
Ours-7k	0.675	0.525	0.836	0.728	0.598	0.884	0.873	0.900	0.910
Ours-30k	**0.771**	**0.605**	**0.868**	**0.775**	**0.638**	**0.914**	**0.905**	**0.922**	0.938

表5. Mip-NeRF360各场景PSNR。†表示取自原论文。

	bicycle	flowers	garden	stump	treehill	room	counter	kitchen	bonsai
Plenoxels	21.912	20.097	23.4947	20.661	22.248	27.594	23.624	23.420	24.669
INGP-Base	22.193	20.348	24.599	23.626	22.364	29.269	26.439	28.548	30.337
INGP-Big	22.171	20.652	25.069	23.466	22.373	29.690	26.691	29.479	30.685
Mip-NeRF360†	24.37	**21.73**	26.98	26.40	22.87	**31.63**	**29.55**	**32.23**	**33.46**
Mip-NeRF360	24.305	21.649	26.875	26.175	**22.929**	31.467	29.447	31.989	33.397
Ours-7k	23.604	20.515	26.245	25.709	22.085	28.139	26.705	28.546	28.850
Ours-30k	**25.246**	21.520	**27.410**	**26.550**	22.490	30.632	28.700	30.317	31.980

表6. Mip-NeRF360各场景LPIPS。†表示取自原论文。

	bicycle	flowers	garden	stump	treehill	room	counter	kitchen	bonsai
Plenoxels	0.506	0.521	0.3864	0.503	0.540	0.4186	0.441	0.447	0.398
INGP-Base	0.487	0.481	0.312	0.450	0.489	0.301	0.342	0.254	0.227
INGP-Big	0.446	0.441	0.257	0.421	0.450	0.261	0.306	0.195	0.205
Mip-NeRF360†	0.301	0.344	0.170	0.261	0.339	**0.211**	**0.204**	**0.127**	**0.176**
Mip-NeRF360	0.305	0.346	0.171	0.265	0.347	0.213	0.207	0.128	0.179
Ours-7k	0.318	0.417	0.153	0.287	0.404	0.272	0.254	0.161	0.244
Ours-30k	**0.205**	**0.336**	**0.103**	**0.210**	**0.317**	0.220	**0.204**	0.129	0.205

表7. Tanks&Temples与Deep Blending各场景SSIM。

	Truck	Train	Dr Johnson	Playroom
Plenoxels	0.774	0.663	0.787	0.802
INGP-Base	0.779	0.666	0.839	0.754
INGP-Big	0.800	0.689	0.854	0.779
Mip-NeRF360	0.857	0.660	**0.901**	0.900
Ours-7k	0.840	0.694	0.853	0.896
Ours-30k	**0.879**	**0.802**	0.899	**0.906**

表8. Tanks&Temples与Deep Blending各场景PSNR。

	Truck	Train	Dr Johnson	Playroom
Plenoxels	23.221	18.927	23.142	22.980
INGP-Base	23.260	20.170	27.750	19.483
INGP-Big	23.383	20.456	28.257	21.665
Mip-NeRF360	24.912	19.523	**29.140**	29.657
Ours-7k	23.506	18.892	26.306	29.245
Ours-30k	**25.187**	**21.097**	28.766	**30.044**

表9. Tanks&Temples与Deep Blending各场景LPIPS。

	Truck	Train	Dr Johnson	Playroom
Plenoxels	0.335	0.422	0.521	0.499
INGP-Base	0.274	0.386	0.381	0.465
INGP-Big	0.249	0.360	0.352	0.428
Mip-NeRF360	0.159	0.354	**0.237**	0.252
Ours-7k	0.209	0.350	0.343	0.291
Ours-30k	**0.148**	**0.218**	0.244	**0.241**

统一术语

Gaussian splatting	高斯溅射
radiance field	辐射场
Structure-from-Motion / SfM	运动恢复结构
spherical harmonics / SH	球谐
covariance	协方差
alpha blending	alpha混合
ray-splat intersection	射线–溅射求交
normal consistency	法线一致性
depth distortion	深度畸变

本机工程映射（译后独立阅读札记）

三维高斯是场景重建与渲染表示，不等于闭环动力学模型或动作策略。S9中45°视角变化后的59/64成功来自已标定、已配准的世界系点云与插槽几何；不能据此推出仅重建高PSNR高斯便获得SE(3)鲁棒性。2DGS深度/法线一致性可改善表面点和SDF构建的输入，但仍需误差标定、碰撞裕量与动态更新。

S7的单步生成质量与S3的闭环接触恢复问题，分别属于生成积分误差和分布偏移；3DGS实时光栅化只解决观测表示及渲染成本。S8的824.5 FPS计时从显存原始事件开始，不包含采集与主机上传；事件体素、光流和涡量不能直接替换静态RGB多视图重建假设。工程衔接应分别报告重建误差、标定误差、推理延迟与闭环成功率。

依据：V:/_astra_sprint_hub/logs/S3、S7、S8、S9/final_message.md（2026-11-25只读核对）。