Quadratic Gaussian Splatting for Efficient and Detailed Surface Reconstruction

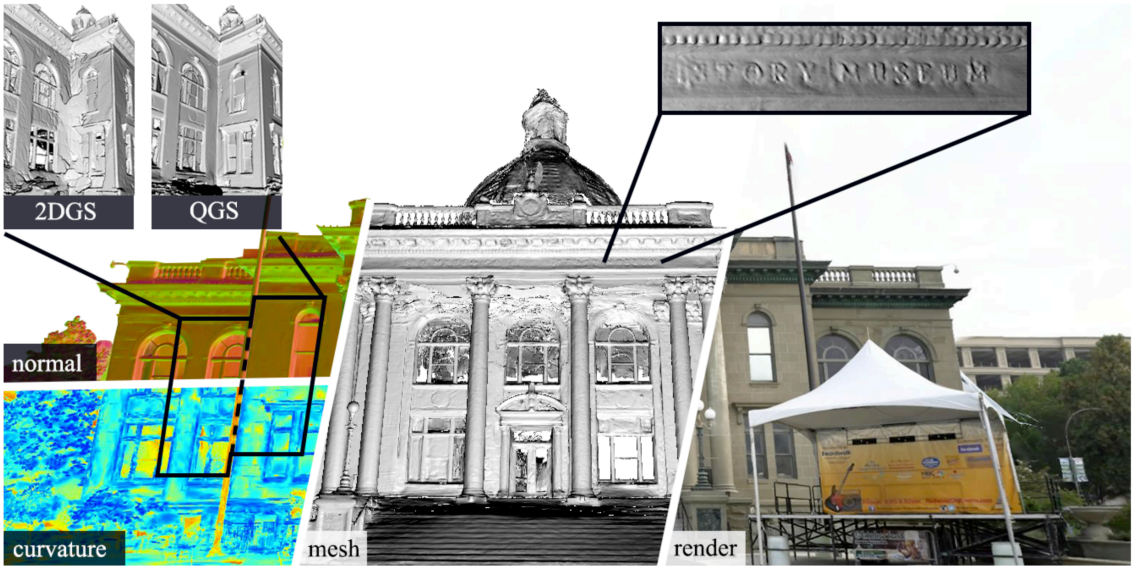


Figure 1. Quadratic Gaussian Splatting, a novel quadric-based approach for recovering high-precision real-world geometry from multi-view RGB images. In the Courthouse scene, our method achieves multi-view consistent normals, depth, and curvature, producing high-quality meshes and realistic images. Compared to 2DGS, QGS more accurately models the building's contours.

Introduction

表面重建（Surface Reconstruction）和真实感新视角合成（Realistic Novel View Synthesis, NVS）是计算机图形学和计算机视觉领域的重要任务。它们的目标是从不同视角捕获的图像中重建密集的几何结构，并渲染出真实感的图像。

最近，3D高斯散点（3D Gaussian Splatting, 3DGS）技术在渲染质量和速度上已经超越了基于神经辐射场（Neural Radiance Fields, NeRF）的方法。这些方法结合了传统的散点技术与端到端优化，从而取得了快速进展。

然而，传统的散点方法忽略了高斯在**z轴方向**的贡献，并使用了近似处理。这种方法虽然能在场景表面偏移的情况下渲染出高质量、多视图一致的纹理，但难以捕获精确的几何结构。

为了解决这些问题，本文提出了一种新型的基于二次曲面（Quadratic Surfaces）的高斯散点技术，称为**二次高斯散点（Quadratic Gaussian Splatting, QGS）**。相比传统方法，QGS通过引入更高阶的表面几何表示，显著提升了几何拟合能力。

本文方法的核心创新在于：

1. 定义了基于非欧几里得空间的高斯分布，使得原语能够更精确地拟合复杂的纹理。
2. 利用二次曲面的二阶拟合能力，在减少过度平滑的同时，输出包含法向一致性和曲率信息的高质量几何结构。

与以往基于平面或椭球体的散点方法相比，QGS在多视角一致性、曲面曲率建模和渲染质量方面均有显著提升。

以下是本文的主要贡献：

- 提出了一种基于二次曲面的高效可微分表示，显著提升了几何拟合能力。
- 首次在高斯散点方法中引入了基于测地距离的高斯分布建模，能够更好地拟合复杂的纹理。
- 通过更严格的深度排序和优化策略，QGS实现了当前最先进（SOTA）的几何重建效果和渲染质量。

通过在多个数据集上的实验验证，本文方法在几何重建和渲染质量方面均显著优于现有方法。

Related Work

本节从以下三个方面综述了与本文方法相关的研究工作：**新视角合成（Novel View Synthesis）**、**神经表面重建（Neural Surface Reconstruction）**和**高斯散点表面重建（Gaussian Splatting Surface Reconstruction）**。

Novel View Synthesis

新视角合成（NVS）近年来取得了快速发展，尤其是NeRF的引入。NeRF通过多层感知机（MLP）对场景几何和视角相关的外观进行编码，并通过体渲染优化，实现了高质量图像生成。

主要改进方法

1. **抗锯齿与采样改进**：
   - **Mip-NeRF**、**Mip-NeRF 360** 和 **Zip-NeRF** 通过引入新的采样策略解决了NeRF的锯齿问题。
2. **加速训练和渲染**：
   - **I-NGP** 和 **DVGO** 通过使用特征网格代替MLP，显著加快了NeRF的训练和渲染速度。

3D高斯散点的优势

最近，3D高斯散点（3DGS）因其在渲染效果和实时性上的优势逐渐受到关注：

- 例如，**Mip-Splatting** 引入了3D平滑滤波以消除渲染过程中的高频伪影。
- **StopThePop** 提出了针对瓦片（per-tile）和像素（per-pixel）的重新排序方法，以解决多视角几何不一致问题。

尽管如此，现有方法在准确几何表示和渲染一致性方面仍存在不足，这正是本文方法的改进重点。

Neural Surface Reconstruction

相比传统的多视角几何重建，基于神经体渲染的表面重建方法通常能生成更平滑、更完整的结果。

符号距离场改进

1. **NeuS** 和 **VolSDF** 引入了符号距离场（Signed Distance Fields, SDF），基于NeRF对场景几何进行描述。
2. **Geo-NeuS** 和 **NeuralWarp** 在重建过程中进一步融合了多视角一致性，提升了几何重建的精度。

局限性

- 神经渲染表面重建通常需要较长时间（几个小时）才能收敛。
- 由于依赖隐式场的结构化表示，获取和编辑场景几何并不直观。

这些问题突出了探索显式几何表示的重要性。

Gaussian Splatting Surface Reconstruction

随着3DGS技术的快速发展，基于高斯散点的方法在表面重建领域也取得了显著进展。

表面对齐的高斯散点

- **SuGaR** 通过在体密度场上引入正则化项，优化高斯椭球的表面拟合，利用泊松重建方法生成网格。
- **GSDF** 和 **NeuSG** 结合了3D高斯椭球和符号距离场，实现了高质量的几何重建与渲染。

局限性与优化

- **2DGS** 将3D高斯椭球简化为高斯盘，更贴近表面几何，但由于仅使用了一阶拟合，重建结果往往过于平滑。
- **MVG-Splatting** 通过引入多视角一致性约束，扩展了2DGS的适用范围。

本文的改进

本文提出的二次高斯散点（QGS）通过引入二次曲面，使高斯分布的拟合能力显著增强：

1. **二次曲面模型**：能够捕获局部几何的高曲率区域。
2. **基于测地距离的高斯分布**：更好地描述了非欧几里得空间中的复杂纹理。
3. **排序优化**：采用StopThePop的排序方法，改进了2DGS中出现的几何伪影。

本文方法进一步推进了高斯散点在表面重建领域的应用，取得了最先进的几何重建效果。

Method

本文方法基于二次曲面高斯分布，提出了一种可微分的表面表示和优化方法，以实现更高效、更精细的三维场景重建和新视角合成。本节分为三个小节进行详细说明：**预备知识**、**二次高斯散点（Quadratic Gaussian Splatting）**和**优化策略（Optimization）**。

3.1 Preliminary

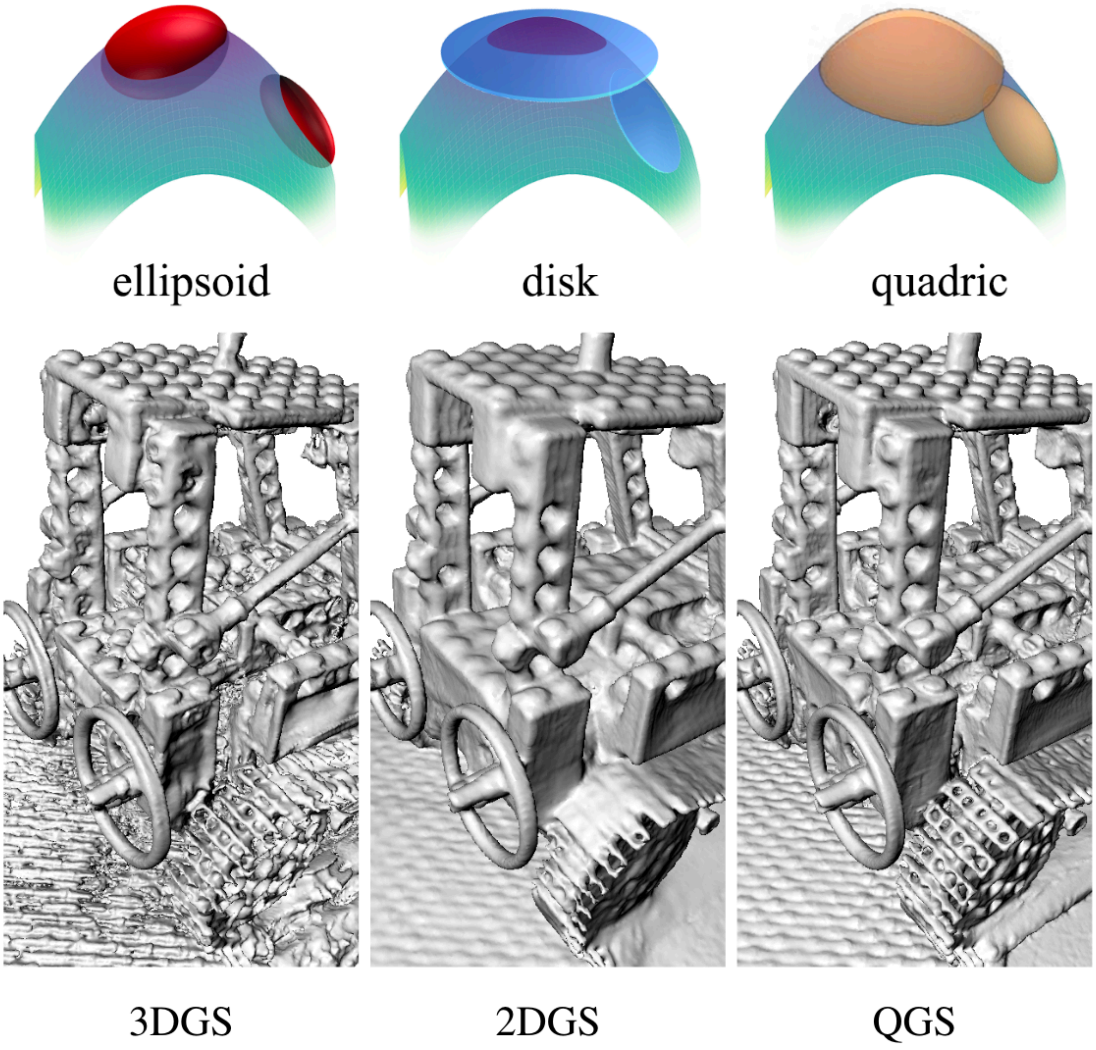


Figure 2. Demo and mesh reconstruction comparison on Mip-NeRF 360 dataset. 3D Gaussian ellipsoids struggle to fit precisely onto surfaces, resulting in coarse and incomplete meshes. Planes cannot fully capture high-curvature regions, leading to over-smoothed reconstructions. In contrast, quadric surfaces effectively capture geometric edge regions, achieving detailed reconstruction results.

本节介绍传统高斯散点方法的基本理论，包括高斯分布的定义及其在相机空间和射线空间的变换过程。

3D高斯分布定义

3D高斯散点将场景表示为高斯椭球的集合，每个椭球由其位置、尺度和方向控制，分布函数为：

$$G(x) = \exp\left(-\frac{1}{2}(x-p_k)^T\Sigma^{-1}(x-p_k)\right) \tag{1}$$

其中：

- x 是空间中的点坐标。
- p_k 是高斯分布的中心。
- 协方差矩阵 $\Sigma = RSS^TR^T$，其中：
  - R 为旋转矩阵，表示椭球的方向。
  - S 为对角矩阵，表示椭球的尺度。

投影到相机空间

高斯分布首先通过世界到相机的变换矩阵 W 映射到相机空间：

$$\Sigma' = JW\Sigma W^TJ^T \tag{2}$$

其中 J 是局部仿射变换矩阵，用于将相机坐标系中的视线方向与坐标轴对齐。

投影到射线空间

在正交投影下，丢弃 Σ' 的第三行和第三列，得到用于表示二维高斯的协方差矩阵 Σ_{2D}：

$$\Sigma_{2D} = \begin{bmatrix}\sigma_{11} & \sigma_{12}\\ \sigma_{21} & \sigma_{22}\end{bmatrix} \tag{3}$$

然后使用体渲染方法将二维高斯积分到图像上，得到像素点的颜色 $C(p)$：

$$C(p) = \sum_{i=0}^{N-1} G_{2D,i}(p)\alpha_i c_i \prod_{j=0}^{i-1}(1-G_{2D,j}(p)\alpha_j) \tag{4}$$

其中：

- α_i 是不透明度。
- c_i 是每个高斯原语的颜色。

3.2.1 Quadratic Gaussian Model

本节提出了QGS方法的核心，即基于二次曲面的高斯分布模型。二次曲面提供了更灵活的几何拟合能力，使得高斯分布能够捕捉复杂的纹理信息。以下是每个公式的详细解析。

二次曲面定义

方程形式

给定一个齐次坐标 $x = [x, y, z, 1]^T \in \mathbb{R}^4$，二次曲面可以表示为以下方程的解集：

$$f(x, y, z) = x^T Q x = 0 \tag{4}$$

公式解析

- x^T 是齐次坐标的转置。
- Q 是对称矩阵，定义了曲面的几何形状。

Q 的结构如下：

$$Q = \begin{bmatrix} A & B & C & D \\ B & E & F & G \\ C & F & H & I \\ D & G & I & J \end{bmatrix} \tag{5}$$

其中：

- 对角元素 A, E, H, J 控制二次项和常数项。
- 非对角元素 B, C, F, D, G, I 控制混合项和线性项。

标准形式化

通过相似对角化，可以将 Q 转换为标准形式：

$$Q = T^{-T} D T^{-1}, \quad D = \mathrm{diag}(d_{ii}) \tag{5}$$

公式解析

- T 是齐次变换矩阵，表示二次曲面的坐标变换，包括位置和方向信息。
- D 是对角矩阵，其对角元素 d_{ii} 确定曲面的具体形状。例如：
  - $d_{ii} = 1$ 定义椭球。
  - $d_{ii} = 0$ 定义平面。
  - $d_{ii} = -1$ 定义双曲面。

抛物面模型

为便于处理，本文专注于抛物面，其标准形式如下：

$$f(x, y, z) = \frac{d_{11}}{s_1^2} x^2 + \frac{d_{22}}{s_2^2} y^2 - \frac{d_{33}}{2 s_3} z = 0 \tag{6}$$

公式解析

- s_1, s_2, s_3 是缩放参数，控制抛物面的形状。
- d_{11}, d_{22}, d_{33} 决定抛物面的方向和弯曲特性。

为了实现更平滑的曲面类型过渡，本文引入带符号的缩放因子：

$$s(x, t) = \tanh(t) \exp(x) \tag{6}$$

这样，曲面方程变为：

$$f(x, y, z) = \frac{\mathrm{sign}(s_1)}{s_1^2} x^2 + \frac{\mathrm{sign}(s_2)}{s_2^2} y^2 - \frac{1}{s_3} z = 0 \tag{7}$$

公式解析

- $\mathrm{sign}(s_i)$ 确保缩放参数的正负号能够区分抛物面的凸凹形态。
- 当 $s_3 \to 0$ 时，抛物面退化为二维高斯盘（2DGS 的特例）。

高斯分布在二次曲面上的定义

为了定义高斯分布，本文使用测地距离作为度量，使高斯分布的能量集中于曲面。

抛物面显式方程为：

$$z = s_3 \left(\frac{\text{sign}(s_1)}{s_1^2} x^2 + \frac{\text{sign}(s_2)}{s_2^2} y^2 \right) \tag{8}$$

将其转换为柱坐标形式：

$$z(\theta, \rho) = a(\theta)\rho^2, \quad a(\theta) = s_3 \left(\frac{\text{sign}(s_1)\cos^2\theta}{s_1^2} + \frac{\text{sign}(s_2)\sin^2\theta}{s_2^2} \right) \tag{9}$$

公式解析

- ρ 是柱坐标中的径向距离。
- θ 是角度。
- $a(\theta)$ 是一个依赖角度的参数，控制抛物面的曲率。

测地距离 $l(a, \rho)$ 定义为：

$$l(a, \rho) = \int_0^\rho \sqrt{1 + (2at)^2} dt \tag{10}$$

其解析解为：

$$l(a, \rho) = \frac{\ln\left(\sqrt{u^2+1} + u\right) + u\sqrt{u^2+1}}{4a}, \quad u = 2a\rho \quad (10, \text{推导}$$

最终，二次曲面上的高斯分布定义为：

$$G(p) = \exp\left(-\frac{l(a, \rho)^2}{2\sigma^2} \right) \tag{13}$$

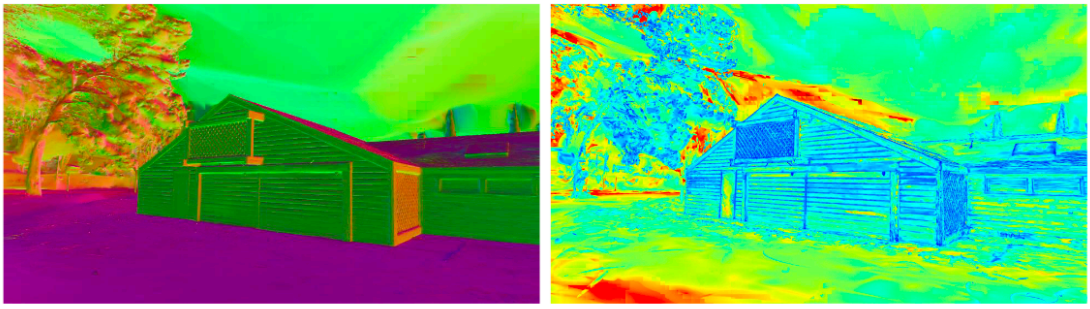
公式解析

- $G(p)$ 描述曲面上的高斯分布值。
- σ 是高斯分布的标准差，控制分布的扩散程度。
- 当 $a \to 0$（曲率接近于0）时，测地距离 l 退化为欧几里得距离，QGS退化为2DGS。

关键特性总结

1. **统一模型**：QGS能够表示椭球、平面和抛物面，是一种更广义的几何建模方法。
2. **曲面高斯分布**：通过测地距离集中高斯能量，捕获复杂的纹理信息。
3. **退化一致性**：在特定参数下，QGS能够退化为2DGS，保证与现有方法的兼容性。

3.2.2 Splatting



normal curvature

Figure 4. The normal map (left) and curvature map (right) of a scene. In the curvature map, blue indicates higher curvature and curved areas, while red indicates lower curvature and flatter areas.

本节详细介绍了二次曲面与射线交点的求解方法，以及高斯分布值的计算过程。为保证渲染时的多视角一致性，本文对交点的深度排序和累积权重计算进行了优化。

射线与二次曲面交点求解

射线方程

在渲染过程中，射线由相机中心 $\hat{o}$ 和方向 $\hat{d}$ 定义，表示为：

$$p(t) = \hat{o} + t\hat{d}, \quad t \in \mathbb{R} \tag{24}$$

公式解析

- $\hat{o}$：相机中心位置。

- $\hat{d}$：射线方向的单位向量。
- t：射线参数，表示点 p 在射线上的位置。

代入曲面方程

将射线方程代入二次曲面方程 $f(x) = x^T Q x = 0$，得到关于 t 的二次方程：

$$At^2 + Bt + C = 0, \tag{25}$$

其中：

- $A = \hat{d}^T Q \hat{d}$
- $B = 2\hat{o}^T Q \hat{d}$
- $C = \hat{o}^T Q \hat{o}$

公式解析

- A, B, C 分别由射线方向 $\hat{d}$ 和相机位置 $\hat{o}$ 决定。
- 二次方程的解 t_n 和 t_f 表示射线与二次曲面的两个交点。

求解交点

二次方程的根可以通过公式法求解：

$$t = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \tag{26}$$

数值稳定性处理

- 当 $|A| \to 0$ 时，方程退化为线性形式 $Bt + C = 0$。
- 为避免浮点运算中的精度丢失，当判别式 $B^2 - 4AC < 0$ 时，判定射线未与曲面相交。

深度范围

实际渲染时，选择满足 $t > 0$ 且位于视锥范围内的交点。

高斯分布的累积计算

对于射线在曲面上的投影，本文使用体渲染方法计算高斯分布值和颜色权重。射线穿过的所有高斯原语按照深度 t 排序，并累积计算颜色值 $C(p)$：

$$C(p) = \sum_{i=0}^{N-1} G_{2D,i}(p)\alpha_i c_i \prod_{j=0}^{i-1}(1 - G_{2D,j}(p)\alpha_j) \tag{27}$$

公式解析

- $C(p)$：像素点 p 的最终颜色。
- N：射线穿过的高斯原语数量。
- $G_{2D,i}(p)$：高斯分布的二维投影值。
- α_i：不透明度，表示高斯原语的遮挡程度。
- c_i：高斯原语的颜色。

排序优化

传统深度排序

标准的体渲染方法按深度 t 从近到远排序高斯原语，然后依次累积计算颜色值。然而，二维高斯分布 $G_{2D,i}(p)$ 的中心与真实深度可能不完全一致，导致几何不一致问题。

瓦片排序

为改进排序策略，本文采用瓦片排序（Tile Sorting）：

1. 将图像划分为 16×16 的瓦片。
2. 对每个瓦片中所有射线的高斯原语按照深度进行初步排序。

瓦片排序的优点是能够在局部范围内提高排序效率

像素重新排序

在体渲染时，为减少几何伪影，进一步对每个像素的高斯原语重新排序。具体方法如下：

1. 缓存每个像素的局部排序结果。
2. 根据透过率调整原语的深度顺序

曲面的法向计算

曲面上的法向量可以通过对二次曲面方程 $f(x, y, z)$ 求梯度计算：

$$\hat{n} = \nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right) = \left(\frac{2\text{sign}(s_1)x}{s_1^2}, \frac{2\text{sign}(s_2)y}{s_2^2}, -\frac{1}{s_3} \right) \quad (28)$$

公式解析

- $\hat{n}$：曲面上的法向量，用于描述曲面在某点的朝向。

- s_1, s_2, s_3：缩放因子，影响法向量的大小和方向。

法向量是几何拟合和光照计算中的关键因素，其准确性直接影响渲染质量。

关键特性总结

1. **二次曲面交点求解**：

   - 使用解析方法高效计算交点，结合数值稳定性处理，保证了渲染的精确性。

2. **排序优化**：

   - 瓦片排序和像素重新排序显著减少了几何伪影，提高了多视角一致性。

3. **高斯分布累积**：

   - 通过深度排序和累积计算，生成了高质量的渲染结果。

3.3 Optimization

本节介绍了本文方法的优化目标函数与策略。

损失函数

1. **深度失真损失（Depth Distortion Loss）**：约束曲面深度的一致性：

$$L_d = \sum_{i=0}^{N-1} \sum_{j=0}^{i-1} \omega_i \omega_j (t_i - t_j)^2 \quad (7)$$

2. **法向一致性损失（Normal Consistency Loss）**：约束曲面法向与实际表面的对齐：

$$L_n = \sum_i \omega_i (1 - \hat{n}_i^T N) \quad (8)$$

3. **总损失函数**：结合上述损失函数优化：

$$L = L_c + \lambda_d L_d + \lambda_n L_n \tag{9}$$

其中 L_c 为光度损失，λ_d 和 λ_n 为权重。

优化过程

优化过程中，本文通过稀疏点云和姿态图像初始化高斯原语，并逐步更新其位置、尺度和旋转参数。

3.2. Quadratic Gaussian splatting

To enhance the geometric fitting capability of the surface representation, we present differentiable Quadratic Gaussian Splatting, as shown in Fig 1. We will first introduce the Quadric Gaussian Model, then discuss the splatting design for quadrics, and finally explain the optimization process.

3.2.1 Quadratic Gaussian Model

Quadratic Model. Given a homogeneous coordinate $\mathbf{x} = [x, y, z, 1]^T \in \mathbb{R}^4$, a quadric surface can be defined as the solution set to the following equation:

$$\begin{aligned} f(x,y,z) &= Ax^2 + 2Bxy + 2Cxz + 2Dx + Ey^2 \\ &\quad + 2Fyz + 2Gy + Hz^2 + 2Iz + J \\ &= \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} A & B & C & D \\ B & E & F & G \\ C & F & H & I \\ D & G & I & J \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \\ &= \mathbf{x}^T \mathbf{Q} \mathbf{x} = 0 \end{aligned} \tag{4}$$

Similar to [26], we apply congruent diagonalization to transform the above equation into its canonical form:

$$\begin{aligned} \mathbf{Q} = \mathbf{T}^{-T}\mathbf{D}\mathbf{T}^{-1}, \text{with } \mathbf{D} \text{ diagonal}, d_{ii} \in \{0, \pm 1\} \\ \mathbf{T} = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{w} & \mathbf{c} \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{5}$$

Here, $\mathbf{c}$ denotes the position of the quadric, and $[\mathbf{u}, \mathbf{v}, \mathbf{w}]$ can be decomposed into $\mathbf{RS}$, where $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$ and $\mathbf{S} = \text{diag}(s_1, s_2, s_3)$, which denotes the orientation and scale of the quadric in the object space. The matrix $\mathbf{D}$ defines the surface shape: $\mathbf{D} = \text{diag}(1, 1, 1, 1)$ yields an ellipsoid, while $\mathbf{D} = \text{diag}(1, 0, 0, 0)$ produces a plane.

To compute the Gaussian weight at any surface point, we first define a measure on the surface to establish the Gaussian distribution. To concentrate the Gaussian energy on the surface, we use geodesic length [27] as the metric: the geodesic distance between two surface points is the shortest path along the surface, as illustrated by the red line in Fig 3.

However, not all geodesics have closed-form solutions. When $\mathbf{Q}$ is an ellipsoid or hyperboloid, computing the geodesic length typically requires numerical methods. Therefore, we focus only on the case of a paraboloid:

$$\begin{aligned} f(x,y,z) &= \mathbf{x}^T \begin{bmatrix} \mathbf{R} & \mathbf{c} \\ 0 & 1 \end{bmatrix}^{-T} \bar{D} \begin{bmatrix} \mathbf{R} & \mathbf{c} \\ 0 & 1 \end{bmatrix}^{-1} \mathbf{x} \\ &= \hat{\mathbf{x}}^T \begin{bmatrix} \frac{d_{11}}{s_1^2} & 0 & 0 & 0 \\ 0 & \frac{d_{22}}{s_2^2} & 0 & 0 \\ 0 & 0 & 0 & -\frac{d_{33}}{2s_3} \\ 0 & 0 & -\frac{d_{33}}{2s_3} & 0 \end{bmatrix} \hat{\mathbf{x}} \\ &= \frac{d_{11}}{s_1^2}\hat{x}^2 + \frac{d_{22}}{s_2^2}\hat{y}^2 - \frac{d_{33}}{s_3}\hat{z} = 0 \end{aligned} \tag{6}$$

Here and henceforth, we use $\hat{\cdot}$ to denote Gaussian local coordinate. $d_{ii} \in \{0, \pm 1\}$ determines whether the paraboloid is elliptic, hyperbolic, or planar. However, since d_{ii} is discrete, the primitive cannot transition smoothly between elliptic and hyperbolic paraboloids. To resolve this, we introduce a signed scale for a differentiable transition between paraboloid types.

$$f(\hat{x}, \hat{y}, \hat{z}) = \frac{\text{sign}(s_1)}{s_1^2}\hat{x}^2 + \frac{\text{sign}(s_2)}{s_2^2}\hat{y}^2 - \frac{1}{s_3}\hat{z} = 0 \tag{7}$$

In vanilla 3DGS [16], the scale is obtained through the $\exp$ activation function, i.e., $s(x) = \exp(x)$. To introduce a sign, we add another variable t to control the sign, i.e., $s(x, t) = \tanh(t)\exp(x)$.

Gaussian Distribution on Quadric. We will now describe how to define a Gaussian distribution on a paraboloid. First, we need to define a measure on the quadric. Paraboloid (Equation 7) could be expressed in explicit form:

$$\hat{z}(\hat{x}, \hat{y}) = s_3\left(\frac{\text{sign}(s_1)}{s_1^2}\hat{x}^2 + \frac{\text{sign}(s_2)}{s_2^2}\hat{y}^2\right) \tag{8}$$

We convert it isometrically into cylindrical coordinates, i.e. $\hat{x} = \rho\cos\theta, \hat{y} = \rho\sin\theta$. And we rewrite Equation 8 as:

$$\begin{aligned} \hat{z}(\theta, \rho) &= s_3\left(\frac{\text{sign}(s_1)\cos^2\theta}{s_1^2} + \frac{\text{sign}(s_2)\sin^2\theta}{s_2^2}\right)\rho^2 \\ &= a(\theta)\rho^2 \end{aligned} \tag{9}$$

Due to the paraboloid's symmetry, for any point $\hat{\mathbf{p}}_0 = (\rho_0, \theta_0, \hat{z}(\theta_0, \rho_0))$, the intersection line of the plane $\theta = \theta_0$ with the paraboloid: $\hat{z}(\theta_0, \rho), \rho \in (0, \rho_0)$, is the geodesic to the origin. Then the geodesic distance is the arc length l of this curve, as shown in Fig 3.

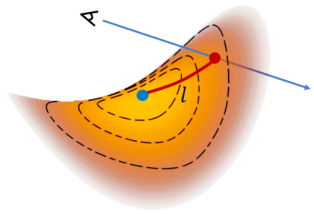


Figure 3. Illustration of a Gaussian on a quadric surface. The blue dot marks the Gaussian centroid, the red dot indicates the ray-splat intersection, and the red line represents the geodesic.

$$\begin{aligned} l(a, \rho_0) &= \int_0^{\rho_0} \sqrt{1 + (2at)^2}\mathrm{d}t \\ &= \frac{\ln(\sqrt{u^2+1} + u) + u\sqrt{u^2+1}}{4a} \\ &\text{where } u = 2a\rho_0 \end{aligned} \tag{10}$$

For the derivation of the integral, please refer to the supplementary materials. We then define the mean of the 2D

Gaussian distribution on the surface at the origin of the quadric surface, with (s_1, s_2) representing the principal axis variances of the Gaussian. Since the contours of the 2D Gaussian distribution form an ellipse:

$$\frac{\rho^2 \cos^2 \theta}{s_1^2} + \frac{\rho^2 \sin^2 \theta}{s_2^2} = 1 \tag{11}$$

Given a point (θ_0, ρ_0) on the ellipse, ρ_0 represents the standard deviation of the 2D Gaussian distribution in the θ_0 direction.

$$\sigma(\theta_0) = \rho_0 = \frac{s_1 s_2}{\sqrt{(s_2 \cos \theta_0)^2 + (s_1 \sin \theta_0)^2}} \tag{12}$$

Thus, for any point $\hat{\mathbf{p}}_0$ on the surface, we can define the corresponding Gaussian function value as:

$$G(\hat{\mathbf{p}}_0(\theta_0, \rho_0)) = \exp(-\frac{(l(a(\theta_0), \rho_0))^2}{2(\sigma(\theta_0))^2}) \tag{13}$$

Notably, when $|s_3| \to 0$, the paraboloid becomes equivalent to a disk. Furthermore, as $x \to 0$, we have $\sqrt{1+x} \to 1$ and $\ln(1+x) \sim x$. Thus, as $|s_3| \to 0$, we get $a \to 0$ and $l \to \rho_0$ by Equation 10, meaning the geodesic distance becomes equivalent to the Euclidean distance. This indicates that 2DGS can be regarded as a specific degenerate case of QGS, whose more generalized nature allows it to fit high-curvature regions effectively.

3.2.2 Splatting

Although Sigg et al. [26] derived the Ray-Quadric Intersection, QGS integrates Gaussian distribution, necessitating a redefinition of the intersection for multi-view consistency.

Ray-splat Intersection. Let the camera center in the Gaussian local space be denoted as $\hat{\mathbf{o}} \in \mathbb{R}^{3\times1}$ and the ray direction as $\hat{\mathbf{d}} \in \mathbb{R}^{3\times1}$. A point on the ray can be defined as $\hat{\mathbf{p}} = \hat{\mathbf{o}} + t\hat{\mathbf{d}}$. By substituting $\hat{\mathbf{p}}$ into the Equation 8, we solve a quadratic equation to obtain two intersection points: the nearer point $\hat{\mathbf{p}}_n = (\hat{x}_n, \hat{y}_n, \hat{z}_n)$ and the farther point $\hat{\mathbf{p}}_f = (\hat{x}_f, \hat{y}_f, \hat{z}_f)$, with $t_n \leq t_f$. To ensure multi-view consistency in QGS, we select one of these two points: First, if the geodesic distance of $\hat{\mathbf{p}}_n$ is within $3\sigma(\theta_n)$, we choose $\hat{\mathbf{p}}_n$. If not, we check if $\hat{\mathbf{p}}_f$ is within $3\sigma(\theta_f)$. If so, we select $\hat{\mathbf{p}}_f$. If neither condition is met, we assume no intersection between the ray $\hat{\mathbf{p}}(t)$ and the primitive. The derivation is provided in the supplementary material.

Normal and Curvature. Similar to 2DGS [12], QGS is a surface-based representation that naturally possesses multi-view consistent geometric properties, making it straightforward to compute surface normals. Given any point $\hat{\mathbf{p}}_0 = (\hat{x}_0, \hat{y}_0, \hat{z}(\hat{x}_0, \hat{y}_0))$ on the surface, we can take the partial derivatives of the Equation 7, yielding:

$$\hat{\mathbf{n}}_0(\hat{\mathbf{p}}_0) = \left(\frac{2\text{sign}(s_1)}{s_1^2}\hat{x}_0, \frac{2\text{sign}(s_2)}{s_2^2}\hat{y}_0, -\frac{1}{s_3}\right) \tag{14}$$

As QGS provides a second-order fit, it naturally outputs second-order geometric information like curvature, which describes surface bending. For each QGS primitive, the Gaussian curvature at the ray-splat intersection can be computed analytically, with detailed derivation available in the supplementary material. Let $\lambda_1 = \text{sign}(s_1) \cdot s_3/s_1^2$ and $\lambda_2 = \text{sign}(s_2) \cdot s_3/s_2^2$. The curvature at point $\hat{\mathbf{p}}_0$ is:

$$\hat{K}_0(\hat{\mathbf{p}}_0) = \frac{4\lambda_1\lambda_2}{(1 + 4\lambda_1^2\hat{x}_0^2 + 4\lambda_2^2\hat{y}_0^2)^2} \tag{15}$$

We render the normal map $N(\mathbf{p})$ and curvature map $K(\mathbf{p})$ by Equation 16 for a given viewpoint using alpha-blending, as shown in the Fig 4.

$$\begin{aligned} N(u,v) &= \sum_{i=0}^{N-1} G_i\alpha_i\mathbf{n}_i \prod_{j=0}^{i-1}(1 - G_j\alpha_j) \\ K(u,v) &= \sum_{i=0}^{N-1} G_i\alpha_i K_i \prod_{j=0}^{i-1}(1 - G_j\alpha_j) \end{aligned} \tag{16}$$

Per-tile Sorting and Per-pixel Resorting. In Equation 16, the index i is determined by sorting primitives from near to far. Most GS-like methods sort by splat centroid depth rather than ray-splat intersection depth, which, as noted by StopThePop [25], causes popping artifacts. Additionally, we observed that centroid sorting introduces streak artifacts in geometry reconstruction, as shown in Fig 8. To address this, we apply StopThePop's Per-tile Sorting and Per-pixel Resorting [25] to our QGS. In short, we first compute the ray closest to the quadric center in each tile to estimate intersection depth for rough sorting. During volume rendering, we use a buffer array to locally resort quadrics based on the intersection depth of each pixel. Further details are provided in the supplementary material.

$$\mathcal{L}_d = \sum_{i=0}^{N-1} \sum_{j=0}^{i-1} \omega_i \omega_j (t_i - t_j)^2 \tag{17}$$

Here, $\omega_i = \bar{\alpha}_i T_i$ denotes the alpha-blending weight of the i-th Gaussian, and t_i represents the depth at the ray-splat intersection. i, j index the Gaussians contributing to the ray. Additionally, following GOF's recommendation, we freeze the gradient of the depth distortion loss through ω_i, optimizing only the depth t_i.

Normal Consistency. 2DGS [12] introduces the normal consistency loss to ensure that all primitives on a ray are locally aligned with the actual surface.

$$\mathcal{L}_n = \sum_i \omega_i (1 - \mathbf{n}_i^T \mathbf{N}) \tag{18}$$

Here, $\mathbf{n}_i$ represents the splat normal facing the camera, while $\mathbf{N}$ is computed by differentiating the depth point $\mathbf{p}$ from neighboring pixels.

$$\mathbf{N}(u,v) = \frac{\nabla_u \mathbf{p} \times \nabla_v \mathbf{p}}{|\nabla_u \mathbf{p} \times \nabla_v \mathbf{p}|} \tag{19}$$

However, neighboring pixels may not satisfy the local planar assumption, especially in regions with significant depth variation. Using differential normal consistency loss in edge areas can introduce errors, contributing to the over-smoothing observed in 2DGS. PGSR [4] and MVG-splatting [19] have both noted this issue and use image edges to approximate geometric edges, applying additional processing at these edges. We have found that, in most scenes, image edges do not fully correspond to geometric edges, especially in uniformly lit areas. Thus, we utilize the curvature map, which more accurately corresponds to geometric edges and is both efficiently and uniquely generated by QGS, to guide normal supervision.

$$\begin{aligned} \lambda_K(K(u,v)) &= 1 - \text{sigmoid}(\ln(|K(u,v)| + \varepsilon) \\ \mathcal{L}_{Kn}(u,v) &= \lambda_K(K(u,v))\mathcal{L}_n(u,v) \end{aligned} \tag{20}$$

Final Loss. Finally, we input a sparse point cloud and posed images to optimize QGS with the following loss function:

$$\mathcal{L} = \mathcal{L}_c + \lambda_d \mathcal{L}_d + \lambda_n \mathcal{L}_{Kn} \tag{21}$$

Supplementary Material

本文附录包含以下内容：

1. 投影边界框的计算方法。
2. 瓦片排序与像素重新排序的详细说明。
3. 射线与二次曲面交点的求解与数值处理。
4. 测地距离公式积分的详细推导。

5. 高斯曲率的推导。
6. 附加定性结果展示。

1. 投影边界框的计算方法

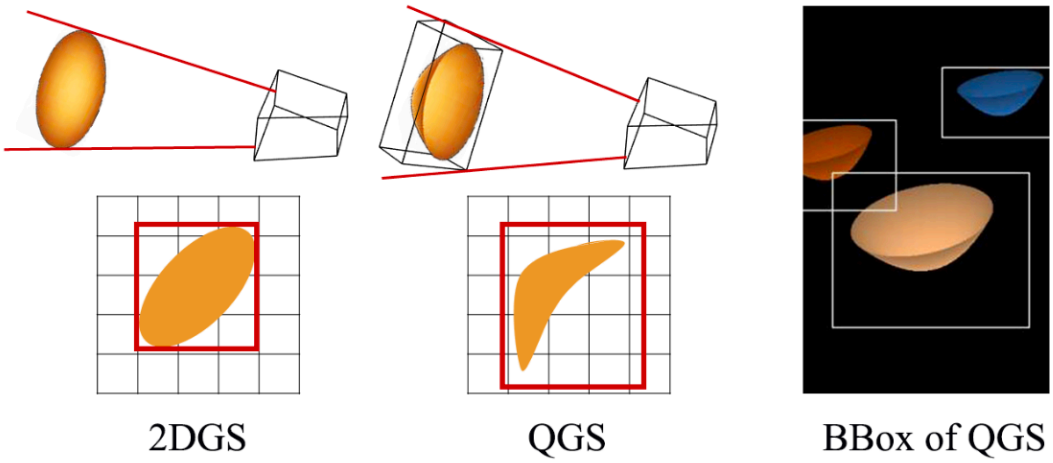


Figure 9. Comparison diagram of the bounding boxes between 2DGS and QGS.

对于2DGS/3DGS，原语为凸形表示，其投影边界框的计算较为简单。然而，QGS包含凹形情况，且测地距离的引入使得精确计算更复杂。

为简化处理，本文使用二次曲面的外接矩形框作为3D边界框，并将其顶点投影到图像平面，计算2D轴对齐边界框（AABB）。为此，本文对测地函数 $l(a,\rho)$ 进行了近似：

$$l(a,\rho) \approx \hat{l}(a,\rho) = 4|a|\rho^2 + 6\rho \tag{10}$$

基于主要轴（$\theta = 0$ 和 $\theta = \frac{\pi}{2}$），计算边界框的长半径 ρ_l 和短半径 ρ_s，从而确定其边界。

2. 瓦片排序与像素重新排序

问题描述：
大多数基于高斯散点的几何重建方法（包括2DGS）使用高斯质心深度排序，这可能导致几何不一致。

本文方法：

- **瓦片排序**：以16×16像素为瓦片，使用离曲面中心最近射线的交点深度进行初步排序。
- **像素重新排序**：在体渲染时，通过缓存每个像素的局部排序结果，对原语重新排序以消除块状伪影。

具体公式如下：
渲染过程中，像素的体渲染值 $\hat{X}$ 表示为：

$$\hat{X} = \sum_{i=0}^{N-1} \alpha_i T_i X_i \tag{11}$$

其中：

- α_i 为不透明度。
- $T_i = \prod_{j=0}^{i-1}(1-\alpha_j)$ 为累积透过率。

反向传播中，将排序修改为由近到远的顺序，以保证一致性。

---

3. 射线与二次曲面交点的求解与数值处理

给定射线方程：

$$p(t) = \hat{o} + t\hat{d} \tag{12}$$

代入二次曲面方程得到：

$$At^2 + Bt + C = 0 \tag{13}$$

通过求解该二次方程，得到两个交点 t_n 和 t_f：

$$t = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \tag{14}$$

数值稳定性处理：

- 当 $|A| < 1e-6$ 时，忽略二次项，简化为线性方程 $Bt + C = 0$，从而避免浮点溢出。

---

4. 测地距离公式积分的推导

测地距离公式为：

$$l(a, \rho_0) = \int_0^{\rho_0} \sqrt{1 + (2at)^2} dt \tag{15}$$

通过变量代换 $u = 2at$，公式变为：

$$l(a, \rho_0) = \frac{1}{2a} \int_0^{2a\rho_0} \sqrt{1 + u^2} du \tag{16}$$

进一步通过分部积分和标准公式推导，最终得到：

$$l(a, \rho_0) = \frac{\sqrt{1 + (2a\rho_0)^2} + \ln\left(2a\rho_0 + \sqrt{1 + (2a\rho_0)^2}\right)}{4a} \tag{17}$$

---

5. 高斯曲率的推导

设二次曲面显式方程为：

$$z = \lambda_1 x^2 + \lambda_2 y^2 \tag{18}$$

曲面的第一基本形式为：

$$E = 1 + 4\lambda_1^2 x^2, \quad F = 0, \quad G = 1 + 4\lambda_2^2 y^2 \tag{19}$$

第二基本形式为：

$$L = 2\lambda_1, \quad M = 0, \quad N = 2\lambda_2 \tag{20}$$

高斯曲率公式为：

$$K = \frac{LN - M^2}{EG - F^2} \tag{21}$$

代入上述值，得到：

$$K = \frac{4\lambda_1\lambda_2}{(1 + 4\lambda_1^2 x^2 + 4\lambda_2^2 y^2)^2} \tag{22}$$

---

6. 附加定性结果展示

本文在附录中展示了QGS方法在多个数据集（包括DTU、Tanks and Temples、Mip-NeRF 360等）上的更多定性结果，验证了方法在细节捕获和几何一致性方面的优越性能。

例如：

- 在城市场景中，QGS能够重建更多建筑细节。
- 在曲率较高的场景中，QGS相比2DGS表现出了更强的拟合能力。

Quadratic Gaussian Splatting for Efficient and Detailed Surface Reconstruction

Supplementary Material

In this supplementary material, we provide detailed explanations of the following: (1) Calculation of the projected bounding box of primitives. (2) Details of per-tile and per-pixel sorting. (3) Solving the ray-primitive intersection equation and numerical handling. (4) Derivation of the geodesic distance formula integration. (5) Derivation of Gaussian curvature. (6) We present more qualitative results. Additionally, we include a video (QGS.mp4) that briefly explains the main workflow of QGS and showcases comparative results.

Details of Bounding Box Calculation.

In 2DGS/3DGS [12, 16], Gaussian primitives are enclosed convex representations, making it straightforward and convenient to compute their bounding box on the image, as shown in the first column of Fig 9.

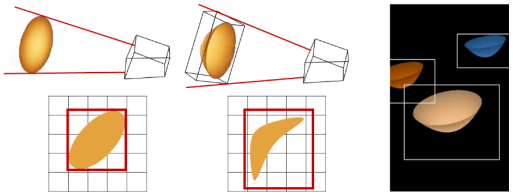


Figure 9. Comparison diagram of the bounding boxes between 2DGS and QGS.

However, QGS includes concave cases, and geodesic distance complicates analytical determination of the Gaussian primitive's rendered portion during preprocessing. Specifically, the geodesic function Equation 10 in the main text lacks an elementary inverse, so computing the equipotential $\rho(\theta_0) = \{l^{-1}(l_0) : l_0 = \sigma(\theta_0)\}$, requires numerical or approximate solutions. Even with an analytical solution for $\rho(\theta_0) = l^{-1}(\sigma(\theta_0))$, the Gaussian distribution in non-Euclidean space means equipotential lines may not fully enclose the primitive, requiring extra boundary calculations. To simplify, we use the quadric's circumscribing rectangular box as the 3D bounding box. Its 8 vertices are projected onto the image plane, and their 2D AABB is used as the 2D bounding box, as shown in Fig 9, columns two and three. Specifically, we first scale the geodesic function to a quadratic polynomial function:

$$l(a,\rho) \approx \hat{l}(a,\rho) = \frac{4|a|\rho^2 + 6\rho}{7} \tag{22}$$

Figure 10. Comparison diagram of the approximate function versus the geodesic function. The red line represents the approximate function, while the black line represents the geodesic function.

Given the direction θ_0, we obtain the variance of the Gaussian distribution $\sigma_0(\theta_0)$. The root of the equation $\hat{l}(a,\rho) - \sigma_0 = 0$ in this direction can be calculated as:

$$\rho_{0,1} = \frac{-6 \pm \sqrt{36 + 112|a|\sigma_0}}{8|a|} \tag{23}$$

The comparison between the approximate function and the geodesic distance function is shown in Fig 10, demonstrating that the two functions nearly overlap.

Since the quadratic curve passes through the origin, we take its positive root, representing the horizontal distance from the primitive's vertex at a geodesic distance of σ_0. We compute this for the major and minor axes, i.e. $\theta = 0$ and $\theta = \pi/2$, yielding ρ_l and ρ_s, with circumscribing box radii $(\rho_l, \rho_s, a\max(\rho_l^2, \rho_s^2))$. We then project the eight vertices of the bounding box onto the image and take the minimum 2D bounding box of the resulting polygon. Due to approximations, gaps may appear between the bounding box and the primitive as shown in the third column of Fig 9, affecting rendering speed but not quality.

Details of Per-tile Sorting and Per-pixel Resorting

2DGS [12] suggests the surface lies at the median intersection point where opacity reaches 0.5. However, for surface-based representations like 2DGS and QGS, the Gaussian distribution is concentrated on the surface, making the alpha-blending order more sensitive. Most reconstruction methods [4, 19, 35, 36], including 2DGS, sort by Gaussian centroid depth, causing geometric inconsistencies, as shown in Fig 8 of the main text.

To address this, we introduce StopThePop's per-tile sorting and per-pixel resorting [25] for more precise ordering. Specifically, for each 16×16 pixel tile, we compute the intersection depth using the ray from the pixel closest to the projected vertex of the quadric surface and apply this depth to all 256 pixels for tile-based global sorting. As shown in Fig 8 in the main text, this method effectively removes

streak-like inconsistencies but introduces minor blocky artifacts due to approximating with a single ray per tile. Then we adopt StopThePop's per-pixel local resorting to reorder the Gaussians along each ray, to eliminate blocky inconsistencies. Specifically, after calculating each Gaussian's depth, normal, and other properties, we do not use them for alpha-blending immediately. Instead, we store them in an 8-length buffer array, and once the buffer is full, we select the closest Gaussian for alpha-blending. In original 3DGS, forward and backward computations use different traversal orders, causing the buffering mechanism to produce inconsistent rendering. Thus, following StopThePop's strategy, we perform gradient computation in a near-to-far order, which requires modifying the original gradient formulas. For volumetric rendering maps, we can uniformly express them as:

$$\hat{X} = \sum_{i=0}^{N-1} \bar{\alpha}_i T_i X_i, \quad \text{where } \bar{\alpha}_i = \alpha_i g_i, T_i = \prod_{j=0}^{i-1}(1-\bar{\alpha}_j)$$

Where X_i represents properties such as the color, depth, normal, or curvature of the primitives. The gradient computation in a back-to-front order is given as follows:

$$\begin{aligned}\frac{\partial \hat{X}}{\partial \bar{\alpha}_i} &= X_i T_i - \frac{\sum_{j=i+1}^{N} X_j \bar{\alpha}_j T_j}{1-\bar{\alpha}_i} \\ &= (X_i - \frac{\sum_{j=i+1}^{N} X_j \bar{\alpha}_j T_j}{T_{i+1}}) T_i\end{aligned}$$

We rewrite it as front-to-back form:

$$\frac{\partial \hat{X}}{\partial \bar{\alpha}_i} = (X_i - \frac{\hat{X} - \sum_{j=0}^{i} X_j \bar{\alpha}_j T_j}{T_{i+1}}) T_i$$

Ensuring that the backward computation follows the same order as the forward computation. Fortunately, the gradient computation order does not affect the distortion loss, meaning no additional derivation is needed for the distortion loss.

Details of Ray-splat Intersection

Given the camera center $\hat{\mathbf{o}} = [\hat{o}^1, \hat{o}^2, \hat{o}^3]^T$ and ray direction $\hat{\mathbf{r}} = [\hat{r}^1, \hat{r}^2, \hat{r}^3]^T$ in the Gaussian coordinate system, the ray can be expressed as:

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{o}^1 + t\hat{r}^1 \\ \hat{o}^2 + t\hat{r}^2 \\ \hat{o}^3 + t\hat{r}^3 \end{bmatrix} \tag{24}$$

Here, t denotes the depth. By solving the ray equation 24 together with the quadric surface equation 7 in the main text, we find two intersection points:

$$\begin{aligned}&t^2[\frac{\text{sign}(s_1)}{s_1^2}(\hat{r}^1)^2 + \frac{\text{sign}(s_2)}{s_2^2}(\hat{r}^2)^2] + \\ &t[\frac{\text{sign}(s_1)}{s_1^2}2\hat{o}^1\hat{r}^1 + \frac{\text{sign}(s_2)}{s_2^2}2\hat{o}^2\hat{r}^2 - \frac{1}{s_3}\hat{r}^3] + \\ &[\frac{\text{sign}(s_1)}{s_1^2}(\hat{o}^1)^2 + \frac{\text{sign}(s_2)}{s_2^2}(\hat{o}^2)^2 - \frac{1}{s_3}\hat{o}^3] = 0 \\ &At^2 + Bt + C = 0 \\ &t_n = \frac{-B - \text{sign}(A)\cdot\sqrt{B^2 - 4AC}}{2A}, \\ &t_f = \frac{-B + \text{sign}(A)\cdot\sqrt{B^2 - 4AC}}{2A}\end{aligned} \tag{25}$$

However, Equation 25 encounters numerical issues when $|A|$ is very small, particularly during backpropagation, as a small denominator can cause floating-point overflow, resulting in invalid values such as NaN or Inf. Upon examination, we observe that when A is especially small, it corresponds to cases where the quadric surface is nearly flat or when the ray is almost perpendicular to the $s_1 \times s_2$ plane. In such cases, we can ignore the quadratic term, reducing Equation 25 to $Bt + C = 0$, or $t = -\frac{C}{B}$. Geometrically, this solution represents the depth of the intersection between the ray and the tangent plane determined by the point where the perpendicular line from the camera center meets the quadric surface, as indicated by the thick blue line n in Fig 11.

Let $\mathbf{M} = s_1 \times s_2$ be the horizontal plane in the Gaussian local coordinate system. Let $\mathbf{p}_o = [\hat{o}_1, \hat{o}_2, s_3(\frac{\hat{o}_1^2}{s_1^2} + \frac{\hat{o}_2^2}{s_2^2})]^T$ represent the intersection of the perpendicular line from the camera center to $\mathbf{M}$ and the quadric surface. The normal of the tangent plane $\mathbf{N}$ at the intersection point can be expressed as:

$$\hat{\mathbf{n}}(\mathbf{p}_o) = [\frac{2\text{sign}(s_1)\hat{o}_1}{s_1^2}, \frac{2\text{sign}(s_2)\hat{o}_2}{s_2^2}, -\frac{1}{s_3}]^T$$

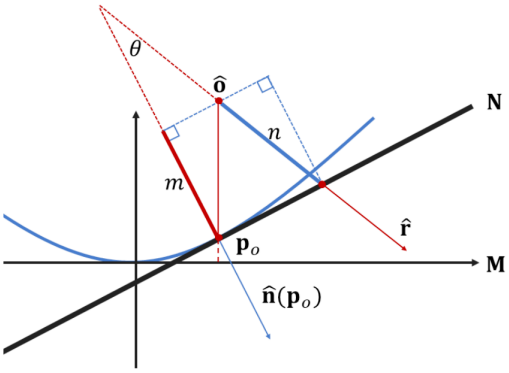


Figure 11. Illustration of approximate intersections. The blue thick line represents the depth of the approximate intersection.

The projection of $(\mathbf{p}_o - \hat{\mathbf{o}})$ onto the normal direction $\mathbf{n}$ is:

$$m = \frac{\hat{\mathbf{n}}(\mathbf{p}_o) \cdot (\mathbf{p}_o - \hat{\mathbf{o}})}{||\hat{\mathbf{n}}(\mathbf{p}_o)||}$$

As indicated by the thick red line segment in the Fig 11. On the other hand, the cosine of the angle between the normal vector and the viewing direction can be expressed as:

$$\cos\theta = \frac{\hat{\mathbf{r}} \cdot \hat{\mathbf{n}}(\mathbf{p}_o)}{||\hat{\mathbf{r}}|| \cdot ||\hat{\mathbf{n}}(\mathbf{p}_o)||}$$

Noting that $||\hat{\mathbf{r}}|| = 1$, the length of the line from the camera center along the viewing direction to the intersection with the tangent plane is given by:

$$\begin{aligned} n &= \frac{m}{\cos\theta} = \frac{\hat{\mathbf{n}}(\mathbf{p}_o) \cdot (\mathbf{p}_o - \hat{\mathbf{o}})}{\hat{\mathbf{r}} \cdot \hat{\mathbf{n}}(\mathbf{p}_o)} \\ &= -\frac{C}{B} \end{aligned}$$

It is evident that when the surface is nearly flat or the viewing direction is almost perpendicular to the horizontal plane, the approximate solution nearly coincides with the exact solution. Therefore, for cases where $|A| < 1\mathrm{e}-6$, we approximate the solution using the method described above.

Derivation of Geodesic Arc Length

For the geodesic arc length formula:

$$l(a, \rho_0) = \int \sqrt{1 + (2at)^2} dt \tag{26}$$

Let $u = 2at, x = \arctan(u)$, then Equation 26 becomes:

$$\begin{aligned} l(a, \rho_0) &= \int \frac{\sqrt{1+u^2}}{2a} du \\ &= \frac{1}{2a} \int \sqrt{1 + \tan^2(x)} d\tan(x) \\ &= \frac{1}{2a} \int \sec(x) d\tan(x) \\ &= \frac{1}{2a} \int \sec^3(x) dx \end{aligned} \tag{27}$$

Equation 27 can be solved using integration by parts:

$$\begin{aligned} \int \sec^3(x) dx &= \tan(x)\sec(x) - \int \tan(x) d\cos(x) \\ &= \tan(x)\sec(x) - \int \sec^3(x) dx \\ &\quad + \int \sec(x) dx \end{aligned}$$

Using $\int \sec(x) dx = \ln|\sec(x) + \tan(x)| + C$, we have:

$$\begin{aligned} \int \sec^3(x) dx &= \frac{\tan(x)\sec(x) + \ln|\sec(x) + \tan(x)|}{2} \\ &= (u\sqrt{1+u^2} + \ln(u + \sqrt{1+u^2}))/2 \\ \Rightarrow l(a, \rho_0) &= \frac{u\sqrt{1+u^2} + \ln(u + \sqrt{1+u^2})}{4a} \end{aligned}$$

Details of Curvature

Here, we compute the Gaussian curvature analytically using a standard differential geometry approach [27]. Given the intersection point $\hat{\mathbf{p}}_0 = [\hat{x}_0, \hat{y}_0, \hat{z}_0]^T$, we simplify Equation 7 as $\hat{z} = \lambda_1 \hat{x}^2 + \lambda_2 \hat{y}^2$. The partial derivatives at $\hat{\mathbf{p}}_0$ are:

$$\begin{aligned} x_u &= (1, 0, 2\lambda_1 \hat{x}_0) \\ x_v &= (0, 1, 2\lambda_2 \hat{y}_0) \end{aligned}$$

The first fundamental form is:

$$\begin{aligned} E &= < x_u, x_u > = 1 + 4\lambda_1^2 \hat{x}_0^2 \\ F &= < x_u, x_v > = 4\lambda_1 \lambda_2 \hat{x}_0 \hat{y}_0 \\ G &= < x_v, x_v > = 1 + 4\lambda_2^2 \hat{y}_0^2 \end{aligned}$$

The second fundamental form is:

$$\begin{aligned} n &= \frac{x_u \times x_v}{||x_u \times x_v||} = \frac{(-2\lambda_1 \hat{x}_0, -2\lambda_2 \hat{y}_0, 1)}{\sqrt{1 + 4\lambda_1^2 \hat{x}_0^2 + 4\lambda_2^2 \hat{y}_0^2}} \\ x_{uu} &= (0, 0, 2\lambda_1) \\ x_{uv} &= (0, 0, 0) \\ x_{vv} &= (0, 0, 2\lambda_2) \\ L &= < n, x_{uu} > = \frac{2\lambda_1}{\sqrt{1 + 4\lambda_1^2 \hat{x}_0^2 + 4\lambda_2^2 \hat{y}_0^2}} \\ M &= < n, x_{uv} > = 0 \\ N &= < n, x_{vv} > = \frac{2\lambda_2}{\sqrt{1 + 4\lambda_1^2 \hat{x}_0^2 + 4\lambda_2^2 \hat{y}_0^2}} \end{aligned}$$

Finally, the Gaussian curvature can be computed as:

$$K = \frac{LN - M^2}{EG - F^2} = \frac{\frac{4\lambda_1\lambda_2}{1+4\lambda_1^2\hat{x}_0^2+4\lambda_2^2\hat{y}_0^2}}{1 + 4\lambda_1^2 \hat{x}_0^2 + 4\lambda_2^2 \hat{y}_0^2}$$

Addtional Results. In this section, we present additional qualitative results of QGS in both indoor and outdoor scenarios. Fig 12 compares QGS and 2DGS on the DTU dataset, where QGS demonstrates sharper contours. QGS also achieves accurate reconstructions in indoor and outdoor scenes, as shown in Fig 13 with results from the TNT and Mip-NeRF 360 datasets. Rendering results for all three datasets are shown in Fig 14. Finally, we qualitatively tested our method on an urban scene captured from aerial view. As shown in Fig 15, QGS captures more building details than 2DGS [12] in aerial views. Additionally, most methods using the default 3DGS configuration failed to reconstruct the Courthouse scene and the urban scene, both containing nearly 1,000 images. For these two scenes, we simply doubled the training parameters, setting the total iterations to 60,000, densifying every 800 iterations from the 4,000th to the 48,000th iteration, and resetting Gaussian opacity every 6,000 iterations. For other scenes, we used the default 3DGS configuration.

Experiment

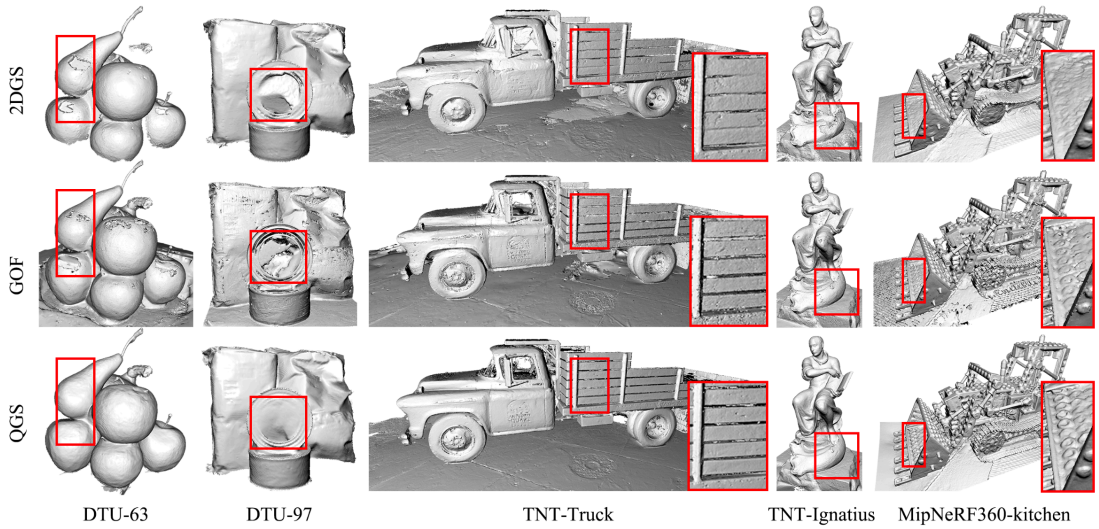


Figure 5. Qualitative geometric reconstruction comparisons on three datasets (DTU, Tanks and Temples, and Mip-NeRF 360) against 2DGS and GOF, covering indoor, outdoor, object-centric, and forward-facing scenes. QGS shows obvious improvements in both detail and smoothness compared to 2DGS and GOF.

CD (mm)↓	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean	Time
NeuS [30]	1.00	1.37	0.93	0.43	1.10	0.65	0.57	1.48	1.09	0.83	0.52	1.20	0.35	0.49	0.54	0.84	>12h
VolSDF [32]	1.14	1.26	0.81	0.49	1.25	0.70	0.72	1.29	1.18	0.70	0.66	1.08	0.42	0.61	0.55	0.86	>12h
Neuralangelo [18]	0.37	0.72	0.35	0.35	0.87	0.54	0.53	1.29	0.97	0.73	0.47	0.74	0.32	0.41	0.43	0.61	>128h
SuGaR[11]	1.47	1.33	1.13	0.61	2.25	1.71	1.15	1.63	1.62	1.07	0.79	2.45	0.98	0.88	0.79	1.33	1h
2DGS [12]	0.48	0.91	0.39	0.39	1.01	0.83	0.81	1.36	1.27	0.76	0.70	1.40	0.40	0.76	0.52	0.80	0.32h
GOF [35]	0.50	0.82	0.37	0.37	1.12	0.74	0.73	1.18	1.29	0.68	0.77	0.90	0.42	0.66	0.49	0.74	1h
Ours	0.46	0.81	0.40	0.42	1.07	0.80	0.79	1.25	0.95	0.67	0.62	1.27	0.40	0.60	0.49	0.73	0.83h

Table 1. Quantitative Chamfer Distance comparison of QGS with GS-like and NeRF-like methods on the DTU dataset. Compared to the planar representation in 2DGS, the surface-based QGS shows a clear improvement in geometric reconstruction quality. Additionally, QGS is an order of magnitude faster than NeRF-like methods.

F1-Score ↑	NeuS	Geo-NeuS	Neuralangelo	2DGS	GOF	Ours
Barn	0.29	0.33	0.70	0.41	0.51	0.43
Caterpillar	0.29	0.26	0.36	0.24	0.41	0.31
Courthouse	0.17	0.12	0.28	0.16	0.28	0.26
Ignatius	0.83	0.72	0.89	0.52	0.68	0.79
Meetingroom	0.24	0.20	0.32	0.17	0.28	0.25
Truck	0.45	0.45	0.48	0.45	0.58	0.60
Mean	0.38	0.35	0.50	0.33	0.46	0.44
Time	>24h	>24h	>127h	0.57h	2h	2h

Table 2. Quantitative F1-Score comparison of QGS with GS-like and NeRF-like methods on the TNT dataset. QGS shows notable gains over 2DGS and is competitive with prior SOTA methods.

本文通过多个数据集上的实验，验证了QGS方法在几何重建和渲染质量方面的性能。实验主要包括以下内容：

1. 实验设置。

2. 方法对比。
3. 消融实验。

4.1 Implementation Details

渲染器实现

QGS的实现基于3DGS框架，使用自定义CUDA核函数扩展了渲染器，输出了深度失真图、深度图、法向图和曲率图。

优化细节：

- 由于抛物面的非凸特性和测地距离函数的非逆性，本文采用近似方法计算图像边界框。
- 使用StopThePop的瓦片排序（Per-tile Sorting）和像素重新排序（Per-pixel Resorting）来处理复杂的曲面交集。

实验硬件：

- 所有实验在单张A6000 GPU上进行。

参数设置：

- 为保证与其他方法比较时高斯原语数量相当，设置梯度阈值为0.3，密度阈值为0.001。
- 在TSDF融合中，体素大小为0.004，截断阈值为0.002。

4.2 Comparison

本文与多种当前最先进（SOTA）方法在DTU、Tanks and Temples（TNT）和Mip-NeRF 360数据集上进行了几何和渲染评估。

几何评估

数据集：DTU

- 衡量指标：Chamfer Distance（CD，单位：mm）。

- 结果：QGS在显式方法（如2DGS、GOF）和隐式方法（如NeuS、VolSDF）中表现出显著的几何重建优势。

表1中列出了QGS与其他方法在DTU数据集上的CD比较：

Table 1: Chamfer Distance Comparison (DTU) (23)

Method	**CD ↓**	**Time**
NeuS	0.84	>12h
VolSDF	0.86	>12h
Neuralangelo	0.61	>128h
2DGS	0.80	0.32h
GOF	0.74	1h
QGS (Ours)	0.73	0.83h

数据集：TNT

- 衡量指标：F1-Score。
- 结果：在大型场景中（如Courthouse和Meetingroom），QGS捕获了更多几何细节，表现优于2DGS和GOF。

表2中列出了QGS与其他方法在TNT数据集上的F1-Score比较：

Table 2: F1-Score Comparison (TNT) (24)

Scene	**2DGS**	**GOF**	**QGS (Ours)**
Barn	0.41	0.51	0.43
Courthouse	0.16	0.28	0.26
Ignatius	0.52	0.68	0.79
Meetingroom	0.17	0.28	0.25

渲染评估

数据集：Mip-NeRF 360

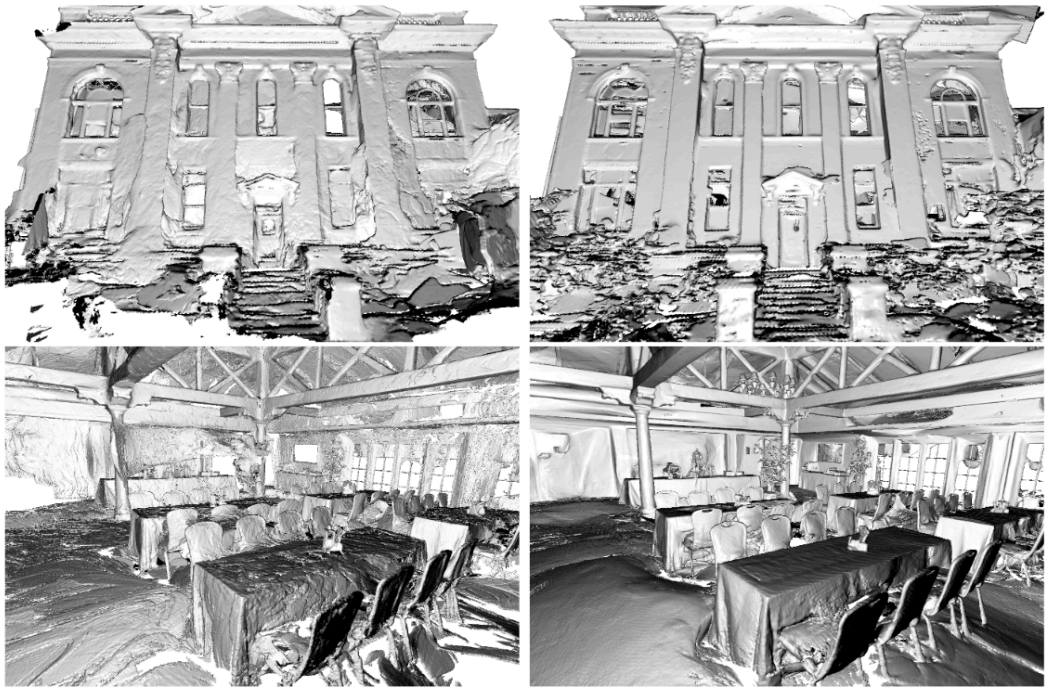
- 衡量指标：PSNR、SSIM、LPIPS。
- 结果：QGS在室内场景中的渲染质量接近SOTA方法，但在视角稀疏或纹理低的室外场景中略有不足。

表3中列出了QGS与其他方法在Mip-NeRF 360数据集上的渲染质量比较：

Table 3: Rendering Quality (Mip-NeRF 360) (25)

Metric	2DGS	GOF	QGS (Ours)
PSNR (Indoor)	30.39	30.80	30.76
PSNR (Outdoor)	24.33	24.76	24.08
LPIPS (Indoor)	0.182	0.167	0.166

4.3 Ablation

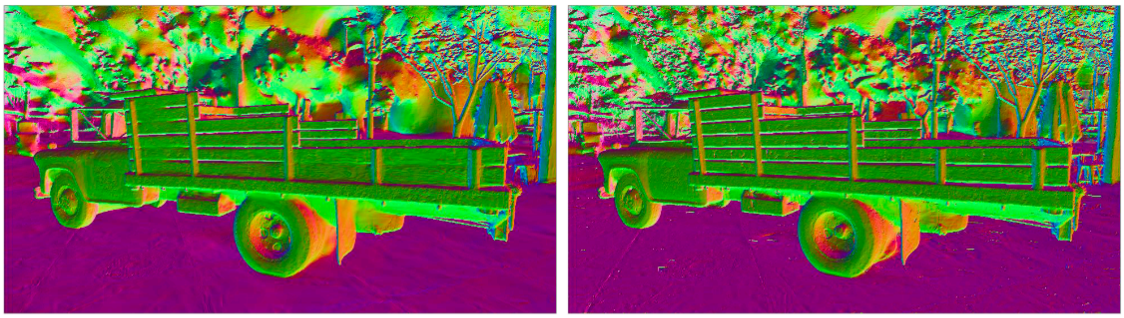


2DGS QGS

Figure 6. Mesh reconstruction comparison between QGS and 2DGS in large-scale scenes. The first row shows the Courthouse scene, with both using median depth, and the second row shows the Meetingroom scene, using a 0.5 weighted combination of median and expected depth.

	Indoor scenes			Outdoor scenes		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRF	26.84	0.790	0.370	21.46	0.458	0.515
Deep Blending	26.40	0.844	0.261	21.54	0.524	0.364
i-NGP	29.15	0.880	0.216	22.90	0.566	0.371
Mip-NeRF360	31.72	0.917	0.180	24.47	0.691	0.283
3DGS	30.99	0.926	0.199	24.24	0.705	0.283
SuGar	29.44	0.911	0.216	22.76	0.631	0.349
2DGS	30.39	0.924	0.182	24.33	0.709	0.284
GOF	30.80	0.928	0.167	24.76	0.742	0.225
QGS	30.76	0.926	0.166	24.08	0.699	0.273

Table 3. Quantitative comparison of appearance between QGS, GS-like, and NeRF-like methods on the Mip-NeRF 360 dataset.



w/o λ_K w/ λ_K

Figure 7. Comparison of curvature-guided normal consistency versus without curvature guidance. Curvature-guided supervision effectively reduces over-smoothing and recovers more details.

F1-Score ↑	Truck			Ignatius		
	Precision↑	Recall↑	F1-score↑	Precision↑	Recall↑	F1-score↑
w/o λ_K	0.63	0.51	0.57	0.74	0.75	0.75
w/ λ_K	**0.65**	**0.55**	**0.60**	**0.78**	**0.80**	**0.79**

Table 4. Quantitative F1-Score comparison on the TNT dataset with and without curvature-guided normal supervision.

w/o TS and PR　　w/ TS　　w/ TS and PR

Figure 8. Normal comparison with and without per-tile sorting. Centroid depth sorting causes noticeable streak-like inconsistencies, also seen in 2DGS.

CD (mm)↓	24	37	97	122
w/o TS and PR	0.48	0.85	1.04	0.55
w/ TS	**0.46**	**0.80**	0.96	0.52
w/ TS + PR	**0.46**	0.81	**0.95**	**0.49**

Table 5. Quantitative Chamfer Distance comparison on the DTU dataset with and without per-tile sorting.

为了验证QGS的关键设计，本文进行了消融实验。

曲率引导的法向一致性监督

- 使用曲率引导法向监督（Curvature-Guided Normal Consistency Loss），对曲率较高的区域减少监督权重。
- 结果：减少了过度平滑，恢复了更多细节。

表4展示了有无曲率引导法向监督的几何质量对比：

Table 4: Effect of Curvature-Guided Supervision (26)

Metric	Truck (F1-Score)	Ignatius (F1-Score)
w/o λ_K	0.57	0.75
w/ λ_K	0.60	0.79

瓦片排序与像素重新排序

- 为解决高斯质心深度排序带来的伪影，本文采用了StopThePop的排序策略。
- 结果：显著减少了条纹伪影和几何不一致。

表5展示了排序策略对Chamfer Distance的影响：

$$\text{Table 5: Effect of Sorting Strategies (DTU)} \tag{27}$$

Scene	w/o Sorting	w/ Tile Sorting	w/ Tile + Pixel Sorting
24	0.48	0.46	0.46
97	1.04	0.96	0.95

总结

QGS在多个数据集上展示了优异的几何重建和渲染性能：

- 在几何重建中，QGS捕获了更丰富的细节，并在高曲率区域表现出强大的拟合能力。
- 在渲染质量上，QGS在室内场景表现优异，但在稀疏视角下可能过拟合，未来可通过曲率正则化改进。

Conclusion

本文提出了一种基于二次曲面表示的高斯散点方法——**Quadratic Gaussian Splatting (QGS)**，以解决现有方法在几何拟合能力和渲染质量上的局限性。通过定义非欧几里得空间中的高斯分布并引入二阶几何特征，QGS在复杂场景的几何重建和多视角一致性方面取得了显著提升。

本文主要贡献

1. **创新的几何表示**：
   - 提出了一种基于二次曲面的可微分表示，能够更精准地拟合高曲率区域，显著提升几何拟合能力。
   - 高斯分布被重新定义为基于测地距离的模型，能够有效捕捉复杂纹理信息。

2. **高效的渲染与排序策略**：
    - 采用了更严格的深度排序策略（瓦片排序与像素重新排序），有效减少了条纹伪影和几何不一致问题。
    - 结合StopThePop的策略优化了渲染流程，提升了渲染质量。
3. **更高的几何与渲染性能**：
    - 在多个数据集（DTU、Tanks and Temples、Mip-NeRF 360）上，QGS在几何重建中达到了当前最先进（SOTA）的性能。
    - 渲染结果在室内场景中表现出高质量与细节保留能力。

局限性与未来工作

局限性

- **速度劣势**：由于QGS需要计算二次曲面的非凸边界框，其渲染速度相较于传统高斯散点方法稍慢。
- **稀疏视角下的表现**：在视角稀疏或低纹理区域，QGS可能出现过拟合问题。

未来改进方向

1. **正则化优化**：
    - 通过增加曲率正则化项，进一步优化QGS在稀疏视角下的拟合性能。
2. **大规模场景支持**：
    - 改善算法的扩展性，使其能够更高效地处理大规模场景。
3. **与其他方法的结合**：
    - QGS的二次曲面模型可与现有的表面重建方法（如Neuralangelo、2DGS）结合，以进一步提升性能。

总结

本文研究表明，通过引入二次曲面和测地距离建模，QGS在几何拟合能力、曲率建模和渲染一致性方面均实现了突破性进展。这种创新的表面表示不仅在当前的高斯散点方法中表现出显著优势，也为未来的表面重建研究提供了新方向。