

Indoor Gaussian Splatting Paper Report

Xinze LI

2024.7.5

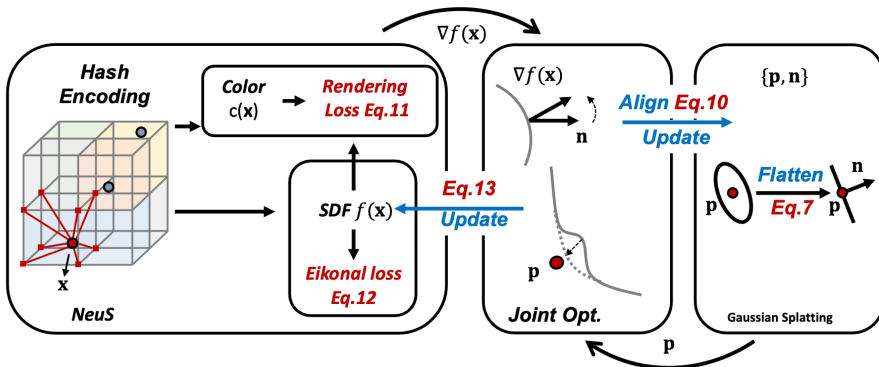
Paper List

- ▶ NeuSG: Neural Implicit Surface Reconstruction with 3D Gaussian Splatting Guidance
- ▶ GaussianPro: 3D Gaussian Splatting with Progressive Propagation
- ▶ GaussianRoom: Improving 3D Gaussian Splatting with SDF Guidance and Monocular Cues for Indoor Scene Reconstruction
- ▶ Mip-Splatting: Alias-free 3D Gaussian Splatting

NeuSG 主要贡献

- ▶ 联合 NeuS + Gaussian Splatting: 利用 GS 生成 Point Cloud 来优化 NeuS, 又利用 NeuS 预测的法线来改进 GS, 以生成更好的 Point Cloud.
- ▶ 引入了两种正则化器, 确保从极细的 3D 高斯分布生成的点云紧密贴合表面。这些正则化器包括将每个 3D 高斯分布的最小缩放因子正则化到接近零, 并使这些高斯分布的法线与表面垂直对齐。(这我就不得不评论了, GaussianPro 也有完全一样的过程, 但是比 NeuSG 晚)

Architecture



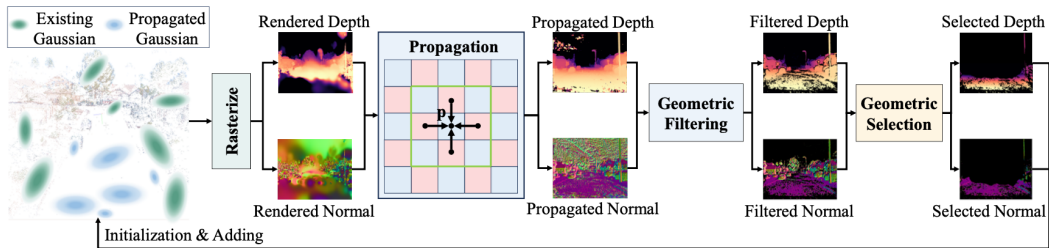
比较核心的点

- ▶ 利用几个 Loss 来完成 NeuS 和 GS 的相互优化过程
- ▶ 压缩 GS 的最小轴到 0 来使得 GS 更加广泛分布在平面的表面上，而不是平面内。

GaussianPro 主要贡献

- ▶ 可以生成更加紧凑和准确的 Gaussians 尤其是在低纹理区域的引导 densification 的 propagation 策略
- ▶ 添加了一个 Planar （平面） Loss 来优化 Gaussians.

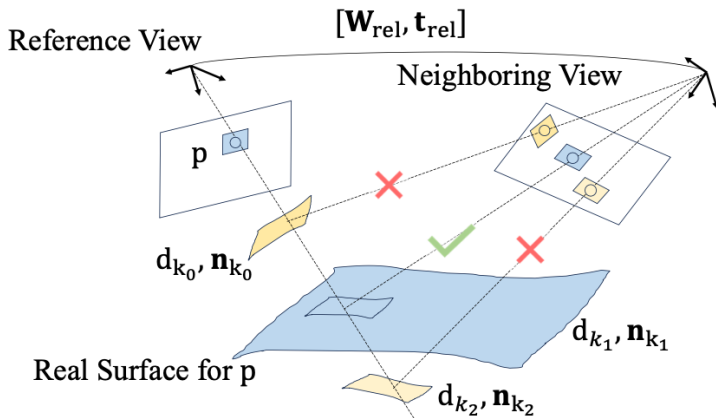
GaussianPro Architecture



流程

1. 从 3DGS 中渲染深度图和法线图。
2. 使用 patch-matching 技术，迭代地对渲染的深度和法线进行传播操作，生成新的深度和法线值（称为传播深度和传播法线）。
3. 使用**几何一致性**过滤掉不可靠的传播深度和传播法线，得到过滤后的深度和过滤后的法线。
4. 识别那些渲染的深度和法线与过滤后的深度和法线显著偏离的区域，表明现有的高斯分布可能未能准确捕捉几何形状，因此需要更多的高斯分布。
5. 将这些区域的像素投影到 3D 空间中，使用过滤后的深度和法线初始化新的高斯分布。

Patch-matching 技术



Patch-matching 技术

1. 对于像素 p ，在传播过程中，执行 p 与每个平面候选之间的单应性变换¹。
2. 单应性变换会得到相邻视图中可能对应的像素。
3. 在 p 与其可能配对像素之间，选择表现出最高颜色一致性的平面候选作为解决方案。
4. 选择的平面候选用于更新像素 p 的深度和法线。

¹单应性：Wikipedia

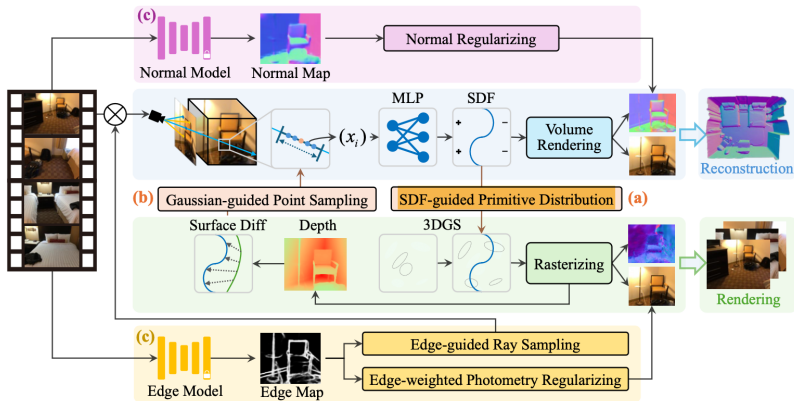
Geometric Filtering and Selection

- ▶ 通过多视角几何一致性检查 (Schönberger et.al, 2016), 过滤掉传播结果中的不准确深度和法线, 得到过滤后的深度和法线图。
- ▶ 计算过滤后深度与渲染深度之间的绝对相对差异。
- ▶ 对于绝对相对差异大于阈值 σ 的区域, 将这些区域的像素投影回 3D 空间。
- ▶ 使用与 3D 高斯喷洒相同的初始化方法, 将这些像素初始化为 3D 高斯分布。
- ▶ 将这些新的高斯分布添加到现有的高斯分布中, 进行进一步优化。

GaussianRoom 主要贡献

- ▶ GaussianPro + NeuSG 的改进版
- ▶ We propose GaussianRoom, a novel unified framework incorporating neural SDF with 3DGS. An SDF-guided primitive distribution strategy is proposed to guide the densification and pruning of Gaussians. At the same time, the geometry represented by Gaussians could improve the efficiency of the SDF field by piloting its point sampling. (不是不想翻译, 是因为这个跟 NeuSG 做的工作一模一样, 所以真没话说)
- ▶ Edge-aware regularization term 来提升重建和渲染效果。并且引入 monocular normal priors in optimization to provide the geometric cues for textureless regions. (越看越眼熟, 是不是跟 GaussianPro 一模一样)

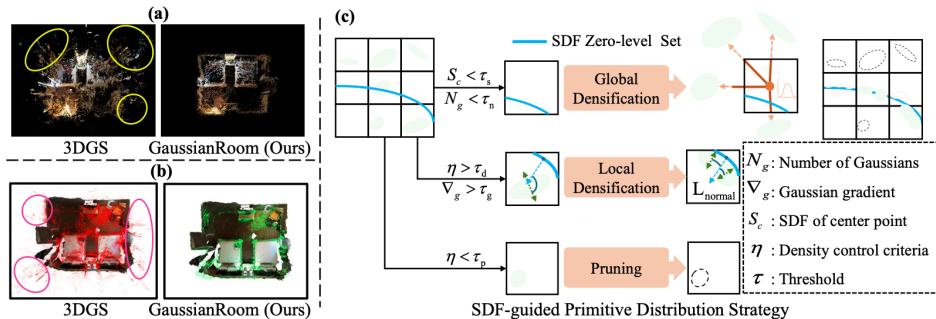
GaussianRoom Architecture



GaussianRoom 流程

1. 利用 SDF 中的几何信息约束高斯基元，以确保它们的空间分布与场景表面对齐。
2. 使用高斯的光栅化深度提供粗略的几何信息，缩小采样范围，从而加速神经 SDF 的优化过程。
3. 引入单眼法线先验和边缘先验，处理室内无纹理区域和细小结构。

SDF Guide



SDF Guide

- ▶ 一是在空间位置上，部署高斯基元以实现具有低 SDF 值的全局加密，这些位置缺乏高斯基元；
- ▶ 二是指导现有高斯基元的加密和修剪过程。

SDF-guided Global Densification

- ▶ 将场景空间划分为 n^3 个立方网格，并计算每个网格中心处的 SDF 值。
- ▶ 如果该值低于阈值 $S_c < \tau_c$ ，则表示该网格靠近场景表面。随后，我们枚举每个网格内现有的高斯基元。
- ▶ 对于高斯基元数量不足的情况 $N_g < \tau_n$ ，我们选择网格中心点的 K 个高斯邻居，并在网格内生成 K 个新的高斯基元。这些新生成的高斯基元的初始属性从由 K 个邻近高斯的均值和方差定义的正态分布中进行采样。(文章的实验采用 $K = 10$)

SDF-guided Densification and Pruning

► 定义:

$$\eta = \exp\left(-\frac{S^2}{\lambda_\sigma \sigma^2}\right)$$

其中, σ 为 opacity of the Gaussian primitive, λ_σ 是一个超参。

► 过程参考前页图片

Gaussian-guided Point Sampling for Neural SDF

- ▶ The rendered depth value D of pixel $\mathbf{p}$ rendered is defined as follows:

$$D(\mathbf{p}) = \sum_{i \in N} d_i \sigma_i \prod_{j=1}^{i-1} (1 - \sigma_j)$$

其中 N 为在像素 $\mathbf{p}$ 出光线 $\mathbf{r}$ 遇到的 Gaussians 数量, d_i 为 3D 高斯函数的深度, σ_i 为投影高斯基元的不透明度。

- ▶ 一旦 $D(\mathbf{p})$ 确定, 定义采样范围 $\mathbf{r}$ 为 (这里有点歧义, 不过应该是用词的区别):

$$[\mathbf{o} + D(\mathbf{p}) - \gamma|S| \cdot \mathbf{v}, \mathbf{o} + D(\mathbf{p}) + \gamma|S| \cdot \mathbf{v}]$$

其中 $\mathbf{o}$ 为相机中心, $\mathbf{v}$ 为像素 $\mathbf{p}$ 的视线方向。 S 为 SDF 值, γ 为采样范围的超参。

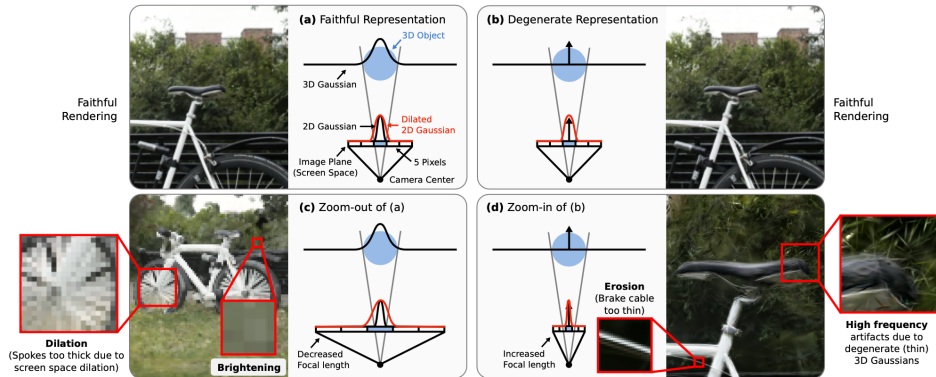
Monocular Cues constrained Optimization

- ▶ 鉴于我没有看 NeuS, 只看了 NeuSG, 合理怀疑这部分就是来自 NeuS
- ▶ Edge-aware 3DGS Optimization 来自 Reference 第 27

Mip-Splatting 主要贡献

- ▶ 为 3DGS 引入了一个 3D 平滑滤波器，有效地规范了 3D 高斯基元的最大频率，解决了先前方法在分布外渲染中观察到的伪影问题
- ▶ 用 2D Mip 滤波器替换了 2D 膨胀滤波器，以解决锯齿和膨胀伪影问题

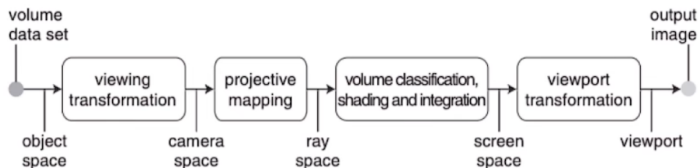
问题描述



问题描述

- ▶ 当改变焦距或相机距离时，会观察到明显的伪影。
- ▶ 这种现象的根源可以归因于缺乏 3D 频率约束以及使用 2D 膨胀滤波器。

CG 中转换的一些知识



- ▶ 观测变换：Eg. 世界坐标系 $\rightarrow$ 相机坐标系；仍然是一个仿射变换： $Wx + b$
- ▶ 投影变换 = 单应性变换（直线还是直线）；Eg. 从相机坐标系 $\rightarrow$ 任意 2D 空间
- ▶ 视口变换：世界坐标系中要显示的区域（通常在观察坐标系内定义）称为窗口。窗口映射到显示器（设备）上的区域称为视口（或称视区）。

NeuSG + GaussianPro + GaussianRoom 的数学路线

1. 定义 3DGS
2. 定义如何“坍缩”GS 的一个轴为 0 到平面表面
 - ▶ 坐标系转换：世界坐标系和相机坐标系的转换；相邻视图坐标系的转换
 - ▶ “坍缩”过程
3. 花里胡哨的 Loss

定义 3DGS

- ▶ 此部分写作最精彩的是 Mip-Splatting，花里胡哨程度堪称富丽堂皇。以下给出比较精简的写法。
- ▶ Each Gaussian is defined by a covariance matrix Σ , and a center point p , which is the mean of the Gaussian. The 3D Gaussian distribution can be represented as:

$$G(\mathbf{x}) = \exp\left\{-\frac{1}{2}(\mathbf{x} - \mathbf{p})^\top \Sigma^{-1}(\mathbf{x} - \mathbf{p})\right\}$$

- ▶ For optimization purposes, the covariance matrix Σ is expressed as the product of a **scaling matrix** S and a **rotation matrix** R :

$$\Sigma = R S S^\top R^\top$$

定义 3DGS

- ▶ 对 Σ 的拆分实质上是把对轴的拉伸操作和旋转操作分别拆成了两个矩阵 S 和 R 。
- ▶ 注意到 scaling matrix S 是一个 Diagonal matrix, 其中包含着三维的缩放因子 s 。rotation matrix R 是一个 orthogonal matrix, 可以用一个四元数 $\mathbf{q} \in \mathbb{R}^4$ 来表示。

定义 3DGS

- 为了渲染新视角，下面将在 camera planes 上应用 Splatting 技术。利用视角变换矩阵 W (实质上就是坐标系的变换) 和仿射逼近的投影变换的雅可比矩阵 J ，变换后的 Σ 可以表达为：

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

- 至于为什么要引入雅可比矩阵，这里给出一种浅显的解释。注意到对于协方差矩阵来说，不能直接使用投影变换。因为原本的相机视角是一个透视投影，现在要到正交投影上，这是一个非仿射变换。
- 雅可比矩阵：假设某函数从 $\mathbf{f}: \mathbb{R}^n \rightarrow \mathbb{R}^m$ ，从 $\mathbf{x} \in \mathbb{R}^n$ 映射到向量 $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^m$ ，其雅可比矩阵是一 $m \times n$ 的矩阵，换句话讲也就是从 $\mathbb{R}^n$ 到 $\mathbb{R}^m$ 的线性映射，其重要意义在于它表现了一个多变数向量函数的最佳线性逼近。因此，雅可比矩阵类似于单变数函数的导数。

定义 3DGS

- ▶ 对于给定的像素，来自多个高斯函数的颜色和不透明度加权。重叠点的颜色混合为：

$$\mathbf{C} = \sum_{i \in N} \mathbf{c}_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j)$$

其中 $\mathbf{c}_i, \alpha_i$ 为一个点的颜色和不透明度。

Neural Implicit SDF

- ▶ SDF 提供了一种隐式表示几何表面的方法。
- ▶ 定义 zero-level set 为：

$$\{\mathbf{x} \in \mathbb{R}^3 \mid f_g(\mathbf{x}) = 0\}$$

其中, $f_g(\mathbf{x})$ is the SDF value from an MLP $f_g(\cdot)$.

- ▶ 利用逻辑函数将 SDF 替换进不透明度 α_i 中：

$$\alpha_i = \max\left(\frac{\Phi_s(f(\mathbf{x}_i)) - \Phi_s(f(\mathbf{x}_{i+1}))}{\Phi_s(f(\mathbf{x}_i))}, 0\right)$$

其中, Φ 为 sigmoid 函数。

Neural Implicit SDF

- ▶ 根据 volume rendering methodology, 像素 p 的预测颜色由沿着光线 $\mathbf{r}$ 上的采样点的加权颜色的累积和给出:

$$\hat{C}(\mathbf{p}) = \sum_{i=1}^N T_i \alpha_i \mathbf{c}_i, \quad T_i = \exp\left(-\sum_{j=1}^{i-1} \alpha_j \delta_j\right)$$

其中 T_i 表示累积透射率, N 表示沿光线 $\mathbf{r}$ 的采样点数量。

- ▶ 类似地, 法向量可以表示为:

$$\hat{\mathbf{N}}(p) = \sum_{i=1}^N T_i \alpha_i \hat{\mathbf{n}}_i$$

其中 $\hat{\mathbf{n}}_i = \nabla f_n(t_i)$