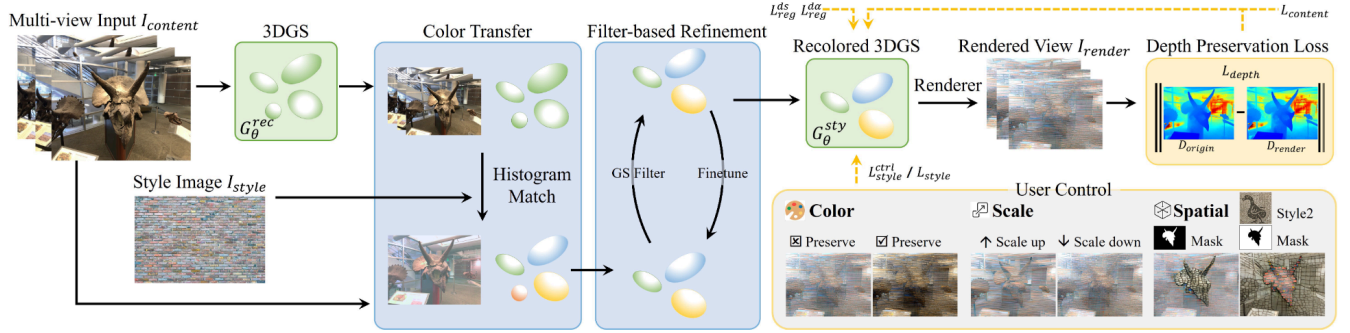


StylizedGS

Method



1 Style Transfer to 3D Gaussian Splatting

对内容图像和 3D 高斯溅射的颜色进行线性变换，以使其色彩统计（均值和协方差）与风格图像对齐。

该公式由两部分组成：

1. 线性变换表达式：

$$p_c^{re} = Ap_c + b$$

$$c^{re} = Ac + b$$

- p_c^{re} ：经过重新着色后的内容图像中的像素。
- p_c ：原始内容图像中的像素。
- c^{re} ：经过重新着色后的 3D Gaussians 的颜色参数。
- c ：原始 3D Gaussians 的颜色参数。
- $A \in \mathbb{R}^{3 \times 3}$ ：一个 3×3 的变换矩阵，对颜色线性缩放和旋转。
- $b \in \mathbb{R}^3$ ：一个 3×1 的平移向量，对颜色线性偏移。
- 这论文通过一个线性变换（矩阵乘法 A 和向量加法 b ）来调整内容图像像素和 3D Gaussians 颜色，以实现颜色风格的迁移。

[疑问]

在风格化过程中保留原始图像的纹理特征，以免被风格化内容覆盖，导致不精确。

2. 约束条件 (s.t.):

$$\text{s.t. } \mu_{p_c^{re}} = \mu_{p_s}, \Sigma_{p_c^{re}} = \Sigma_{p_s}$$

- s.t.: 是“subject to”的缩写，表示“在...条件下”或“满足...约束”。
- $\mu_{p_c^{re}}$ ：重新着色后的内容像素的平均颜色（RGB 三个通道的均值）。

- μ_{ps} : 表示**风格图像**像素的**平均颜色**。
- $\Sigma_{p_c^{re}}$: 表示重新着色后的内容像素的颜色协方差矩阵。协方差矩阵描述了颜色分量（R、G、B）之间的统计关系和分布的形状。
- Σ_{p_s} : 表示风格图像像素的颜色协方差矩阵。
- 这两个约束条件是色彩迁移的核心目标：通过求解 A 和 b ，使得内容图像（以及 3D Gaussians）重新着色后的颜色分布的均值和协方差与风格图像完全匹配。

2 Filter-based Refinement

2.1 目的:

消除原始 3DGS 重建中的“浮点物”(floaters, 即场景中的噪声高斯点)。

2.2 核心思想:

通过对 3DGS 模型进行颜色预处理, 并结合周期性过滤策略和重建损失优化, 实现 3D 场景的“净化”。

2.3 主要步骤:

1. 颜色预处理 (Color Pre-alignment)

- 将输入的多视角内容图像 $\mathcal{I}_{content}$ 的颜色分布, 通过线性变换和直方图匹配, 调整至与给定风格图像 \mathcal{I}_{style} 的颜色统计一致。
- 得到“重新着色的内容图像” $\mathcal{I}_{content}^{re}$ 。
- 同时, 初始 3DGS 模型的颜色参数 c 也会被相应地调整, 使其与新的颜色基调对齐。

2. 周期性过滤与优化 (Iterative Filtering & Optimization)

- 在短迭代周期内 (例如 200 次迭代) 对 3DGS 模型进行微调, 并在此过程中周期性地进
行浮点物过滤。
- **优化:** 使用 $\mathcal{I}_{content}^{re}$ 作为监督信号, 通过最小化以下重建损失 \mathcal{L}_{rec} 来优化 3DGS 的参数。这确保了模型在去除浮点物后, 依然能够准确地渲染出带有新颜色基调的场景。
- **过滤:** 每隔固定的迭代次数 (例如 100 次), 根据 3D 高斯的尺寸和不透明度进行筛选:
 - 移除尺寸在前 $k\%$ 的高斯 (通常代表过大或不必要的扩散)。
 - 移除不透明度在最低 $k\%$ 的高斯 (通常代表稀疏或不重要的噪声)。
 - 论文中设置的 $k\%$ 经验值为: 不透明度 $k = 5\%$, 尺寸 $k = 8\%$ 。

重建损失 (\mathcal{L}_{rec}):

$$\mathcal{L}_{rec} = (1 - \lambda_{rec})\mathcal{L}_1(\mathcal{I}_{content}^{re}, \mathcal{I}_{render}) + \lambda_{rec}\mathcal{L}_{D-SSIM}$$

- $\mathcal{I}_{content}^{re}$: 重新着色后的内容图像(Ground Truth)。
- \mathcal{I}_{render} : 3DGS 模型当前渲染的图像。
- \mathcal{L}_1 : 像素值绝对差。
- \mathcal{L}_{D-SSIM} : 结构相似性。

- λ_{rec} : 超参。

输出:

一个经过“净化”和颜色预对齐的 3DGS 模型，为后续进行精细的图案和笔触风格迁移（第二阶段）奠定高质量基础。

3 Stylization

将参考风格图像的详细风格特征迁移到 3DGS 表示的 3D 场景中，同时确保内容和几何结构的保留。

3.1 目标:

将 2D 风格图像 I_{style} 的详细风格特征迁移到 3D 场景，生成风格化的 3D 模型 $_{sty}$ ，同时保持原始场景的内容结构。

3.2 方法概览:

1. 利用预训练的卷积神经网络（如 VGG-Net）提取特征。
2. 引入 Nearest Neighbor Feature Matching Loss (NNFM) 来捕捉高频风格细节。
3. 结合内容损失、Depth Preservation Loss 和其他正则化项来维持场景的结构和几何一致性

3.3 损失函数:

- 风格损失 (Style Loss)

论文采用了 [Nearest Neighbor Feature Matching Loss](#)。

它通过最小化渲染图像特征图与其在风格特征图中最近邻之间的余弦距离来衡量风格相似性。

公式:

$$_{st}((,),_{st}(,)) = 1 - D((,),_{st}(,))$$

其中:

- F_{render} : 渲染图像 I_{render} 的 VGG 特征图
- F_{style} : 风格图像 I_{style} 的 VGG 特征图
- (i, j) : F_{render} 中像素点的坐标
- (i^*, j^*) : F_{style} 中与 $F_{\text{render}}(i, j)$ 最相似的像素点坐标，通过以下方式找到:
$$(i^*, j^*) = \arg\min_{i', j'} D(F_{\text{render}}(i, j), F_{\text{style}}(i', j'))$$
- $D(a, b)$: 向量 a 和 b 之间的余弦距离
- NN : 特征图中像素点的总数，用于归一化

- 内容损失 (Content Loss)

为了在风格化过程中保留原始场景的内容结构，引入了内容损失。

它衡量渲染图像的特征图与原始内容图像（经过颜色匹配后的 I_{tt} ）特征图之间的均方距离。

公式：

$$L_{tt} = \frac{1}{H \times W} \sum_{i,j} \|F_{tt}(i,j) - F(I)(i,j)\|^2$$

其中：

- F_{tt} ：原始内容图像 I_{tt} 的 VGG 特征图
- $F(I)$ ：渲染图像 I 的 VGG 特征图
- $H \times W$ ：渲染图像的高度和宽度
- 深度保持损失 (Depth Preservation Loss)

为了在优化 3DGS 几何参数时防止场景几何结构发生显著变化，引入了 **Depth Preservation Loss**。

它通过最小化渲染深度图 D 与原始深度图 D 之间的距离，来确保几何的一致性。

公式：

$$L_t = \frac{1}{H \times W} \sum_{i,j} \|D_t(i,j) - D(i,j)\|^2$$

其中：

- D ：通过 3DGS 的 alpha-blending 方法在颜色迁移阶段生成的原始深度图
- D_t ：风格化过程中渲染的深度图
- 正则化项 (Regularization Terms)

为了对 3DGS 参数的变化进行约束，以保持场景的稳定性和细节，论文引入了针对高斯尺度 (s) 和不透明度 (α) 变化的正则化项。

公式：

$$L_s = \frac{1}{M} \sum_i \|s_i - s_0\|^2$$

$$L_{\alpha}^{\text{reg}} = \frac{1}{M} \sum_i \|\alpha_i - \alpha_0\|^2$$

其中：
 s ：高斯尺度参数的变化
 α ：高斯不透明度参数的变化
 M ：高斯点的总数

- 总变分损失 (Total Variation Loss)

L_t 用于平滑渲染图像，减少图像中的噪声和锯齿，提高视觉质量。

- 总损失函数 (Total Loss Function)

公式：

$$L_{\text{total}} = \lambda_{\text{st}} L_{\text{st}} + \lambda_{\text{tt}} L_{\text{tt}} + \lambda_t L_t + \lambda_s L_s + \lambda_{\alpha} L_{\alpha} + \lambda_{\text{tv}} L_{\text{tv}}$$

其中：

- λ : 对应损失项的权重系数，用于平衡不同损失项的重要性。

4 Perceptual Control

4.1 Color Control

目标与功能：

允许用户在风格化过程中**独立调整场景的色彩**，提供以下灵活控制选项：

- 保留原始场景的颜色
- 完全采用参考风格图像的颜色
- 将某个图像的图案风格与另一个图像的颜色风格自由组合

实现方法：

核心思想：在 **YIQ 颜色空间** 中进行解耦操作。

YIQ 颜色空间将图像分解为：

- **Y 通道**：亮度 (Luminance)，主要携带结构、纹理、边缘、高频细节信息
- **I 和 Q 通道**：色度 (Chrominance)，主要携带颜色信息

仅在亮度 (Y) 通道上计算风格损失，从而让模型学习并迁移**颜色无关的风格模式**。

公式：

$$L_{st} = L_{st} \left((I), (I_{st}) \right)$$

其中：

- L_{st} —— 专为颜色控制设计的风格损失项
- $L_{st}(\cdot, \cdot)$ —— 核心风格损失函数，即 Nearest Neighbor Feature Matching Loss
- (\cdot) —— 特征提取器，通常为预训练的 VGG 网络（某层或多层特征图）
- I —— 当前渲染图像的**亮度通道** (Y channel)
- I_{st} —— 参考风格图像的**亮度通道** (Y channel)

4.2 Scale Control

允许用户调整风格化效果中基础风格元素的尺寸，例如笔触的粗细、图案的密度、纹理的颗粒大小等。

核心思想是通过选择用于计算风格损失的 VGG 网络层及其权重，来间接控制感受野的大小，从而实现风格图案尺寸的灵活变化。

尺度风格损失公式：

$$l_s^{st} = \sum_{l \in b_s} l \cdot st(l(I), l(I_{st}))(1)$$

参数解释：

- l_s^{st} ：用于实现 Scale Control 的风格损失项，由多个 VGG 网络层损失的加权和构成。
- b_s ：选定的 VGG-16 网络中的某个块（block）。通过选择不同的块，可以捕获不同尺度的特征。
- $l \in b_s$ ：对指定块 b_s 内所有层 l 进行求和。
- l ：第 l 层对应的权重。通过调整权重分布，可以强调或减弱特定感受野尺度的风格特征对最终结果的影响。
- $st(l(I), l(I_{st}))$ ：在第 l 层计算的 Nearest Neighbor Feature Matching Loss。
- $l(I)$ ：从渲染图像 I 中提取的第 l 层特征图。
- $l(I_{st})$ ：从风格图像 I_{st} 中提取的第 l 层特征图。

通过选择不同的 VGG 块并调节各层权重，用户可以实现从极细腻的高频细节到较为粗犷的大尺度风格结构的连续控制。

4.3 Spatial Control

目标： 让用户可以指定场景中不同区域或对象应用不同风格。

实现方式：

通过生成空间掩码（masks）定义需要风格化的区域，可通过

- 点交互 + SAM + 掩码跟踪（跨视图一致性）
 - 文本提示 + LangSAM
- 两种方式生成多视图一致的掩码。

在优化过程中引入专门的空间风格损失：

$$l_{st}^{st} = \sum_r r \cdot st\left((M_r^c \cdot I_r), (M_r^s \cdot I_{st}^r)\right)$$

参数解释：

- l_{st}^{st} ：空间风格损失，确保只在用户指定区域内进行风格匹配与迁移。
- r ：用户指定的区域对索引（第 r 对内容-风格区域匹配）。
- r ：第 r 个区域对的权重，用于平衡不同区域的重要性。
- $st(,)$ ：核心风格损失函数，即 Nearest Neighbor Feature Matching Loss（见论文公式(4)）。
- (\cdot) ：VGG 网络特征提取器。
- M_r^c ：作用于渲染图像的二值掩码，标记第 r 个待风格化的内容区域（1=保留，0=屏蔽）。
- M_r^s ：作用于风格图像的二值掩码，标记第 r 个用于迁移的风格参考区域。
- \cdot ：逐元素乘法（Hadamard product），实现区域选择。
- I_r ：当前渲染视图中与第 r 个区域对应的图像内容。

- I_{st}^r : 风格参考图像中与第 r 个区域对应的部分。