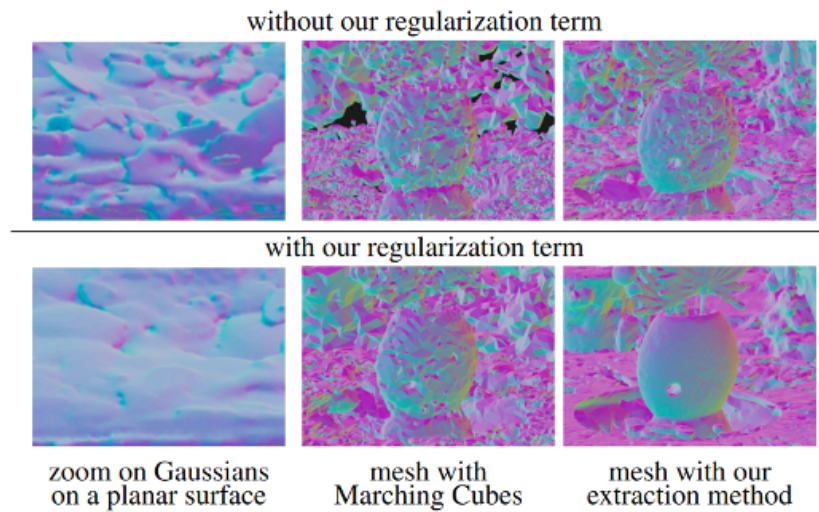


SuGaR

SuGaR: Surface-Aligned Gaussian Splatting for Efficient 3D Mesh Reconstruction and High-Quality Mesh Rendering (CVPR 2024)



优化后的高斯并没有通常呈现出有序的结构，并且与场景的实际表面相符得不好

- 对齐：新的正则项鼓励高斯更好地分布在表面上，定义密度函数 d ，导出点到最近表面的距离函数 $f(d)$ ，最小化该sdf函数和理想情况下的值。
- 提取：从密度函数估计水平集，更好地导出mesh
- 绑定：将高斯绑定在mesh表面进行优化，提高渲染质量

高斯下的密度函数与SDF距离函数：

① 密度函数：在任意空间位置 p 对高斯值及其 Alpha 混合系数进行加权求和计算 得到

$$d(p) = \sum_g \alpha_g \exp \left(-\frac{1}{2} (p - \mu_g)^T \Sigma_g^{-1} (p - \mu_g) \right), \quad (1) \quad C = \sum_{i \in N} c_i \alpha'_i \prod_{j=1}^{i-1} (1 - \alpha'_j), \quad \alpha'_i = \alpha_i \times \exp \left(-\frac{1}{2} (x' - \mu'_i)^T \Sigma_i^{-1} (x' - \mu'_i) \right),$$

② 理想紧贴表面情况下的密度函数 $\bar{d}(p)$ 近似表达推导：

假设高斯完全贴合平面，有以下性质：

➤ 最近的高斯占比最大（鼓励均匀稀疏分布），进行近似：

$$\alpha_{g^*} \exp \left(-\frac{1}{2} (p - \mu_{g^*})^T \Sigma_{g^*}^{-1} (p - \mu_{g^*}) \right) \quad g^* = \arg \min_g \{ (p - \mu_g)^T \Sigma_g^{-1} (p - \mu_g) \}$$

➤ 高斯近似为一个圆盘，理想下有个缩放因子为0，协方差中它贡献最多

$$(p - \mu_g)^T \Sigma_g^{-1} (p - \mu_g) \approx \frac{1}{s_g^2} \langle p - \mu_g, n_g \rangle^2$$

➤ 不透明度为0或1

$$\bar{d}(p) = \exp \left(-\frac{1}{2s_{g^*}^2} \langle p - \mu_{g^*}, n_{g^*} \rangle^2 \right)$$

高斯下的密度函数与SDF距离函数：

① 密度函数：

$$d(p) = \sum \alpha_g \exp \left(-\frac{1}{2} (p - \mu_g)^T \Sigma_g^{-1} (p - \mu_g) \right), \quad (1)$$

② 理想情况下的密度函数 $\bar{d}(p)$ ：

$$\bar{d}(p) = \exp \left(-\frac{1}{2s_{g*}^2} \langle p - \mu_{g*}, n_{g*} \rangle^2 \right). \quad (5)$$

■ 第一种正则： 密度函数

最小化 $|d(p) - \bar{d}(p)|$

■ 第二种正则：距离函数 $f(p)$

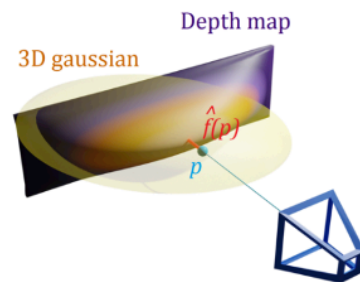
$$\bar{f}(p) = |\langle p - \mu_g, n \rangle| \quad \text{点到平面距离公式}$$

$$\bar{f}(p) = \pm s_{g*} \sqrt{-2 \log(\bar{d}(p))} \quad (6)$$

$$f(p) = \pm s_{g*} \sqrt{-2 \log(d(p))} \quad (7)$$

最小化 $|\hat{f}(p) - f(p)|$

$\bar{f}(p)$ 直接使用采样点到depth_map的距离来代替计算



距离函数的几何意义

$$\mathcal{R} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} |\hat{f}(p) - f(p)|,$$

$$\mathcal{R}_{\text{Norm}} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \left\| \frac{\nabla f(p)}{\|\nabla f(p)\|_2} - n_{g*} \right\|_2^2.$$

距离函数正则和法线正则

计算水平点以进行泊松重建mesh

① 对像素对应的3D点，沿射线方向的 $\pm 3\sigma$ 内采样

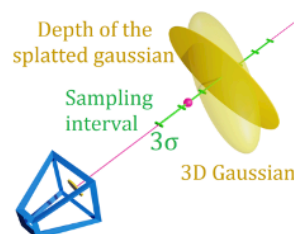
② 找到这些点中密度函数最接近给定阈值 λ 的区间左右点

$$d_i < \lambda < d_j$$

③ 插值得到更精细的点，密度函数等于阈值，此点作为水平集点

$$d(p + t^*v) = \lambda.$$

$$\text{法向: } \frac{\nabla d(\hat{p})}{\|\nabla d(\hat{p})\|_2}$$



泊松重建中水平集上的采样点

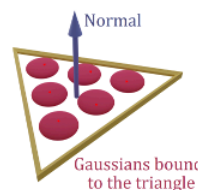
将高斯绑定在mesh的三角内，联合优化高斯和mesh

3D Gaussians 属性:

- 均值: sfm初始化
- 协方差矩阵: 平移和缩放
- 颜色: 球谐函数
- 不透明度

3D Gaussians 属性:

- 均值: 三角形内
- 协方差矩阵: 只在三角内缩放旋转
- 颜色: 不变
- 不透明度: 不变



绑定在三角内的高斯

1. 早期正则化方案的代表
调整高斯到表面上

2. 限制高斯的位置降低了渲染质量
没有最后的优化 渲染质量也不高

	Indoor scenes			Outdoor scenes			Average on all scenes		
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
No mesh (except SuGaR)									
Plenoxels [42]	24.83	0.766	0.426	22.02	0.542	0.465	23.62	0.670	0.443
INGP-Base [23]	28.65	0.840	0.281	23.47	0.571	0.416	26.43	0.725	0.339
INGP-Big [23]	29.14	0.863	0.242	23.57	0.602	0.375	26.75	0.751	0.299
Mip-NeRF360 [2]	31.58	0.914	0.182	25.79	0.746	0.247	29.09	0.842	0.210
3DGS [15]	30.41	0.920	0.189	26.40	0.805	0.173	28.69	0.870	0.182
R-SuGaR-15K (Ours)	29.43	0.910	0.216	24.40	0.699	0.301	27.27	0.820	0.253
With mesh									
Mobile-NeRF [6]	—	—	—	21.95	0.470	0.470	—	—	—
NeRFMeshing [26]	23.83	—	—	22.23	—	—	23.15	—	—
BakedSDF [39]	27.06	0.836	0.258	—	—	—	—	—	—
R-SuGaR-2K (Ours)	26.29	0.872	0.262	22.97	0.648	0.360	24.87	0.776	0.304
R-SuGaR-7K (Ours)	28.73	0.904	0.226	24.16	0.691	0.313	26.77	0.813	0.263
R-SuGaR-15K (Ours)	29.43	0.910	0.216	24.40	0.699	0.301	27.27	0.820	0.253

Extraction method	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
Marching Cubes [21]	23.91	0.703	0.392
Poisson (centers) [14]	23.76	0.756	0.340
Ours (Surface level 0.1)	24.62	0.765	0.313
Ours (Surface level 0.3)	24.87	0.776	0.304
Ours (Surface level 0.5)	24.91	0.777	0.304