3D-HGS: 3D Half-Gaussian Splatting *

Haolin Li Jinyang Liu Mario Sznaier Octavia Camps

{li.haoli, liu.jiyan, m.sznaier, o.camps}@northeastern.edu
Northeastern University
Boston, MA

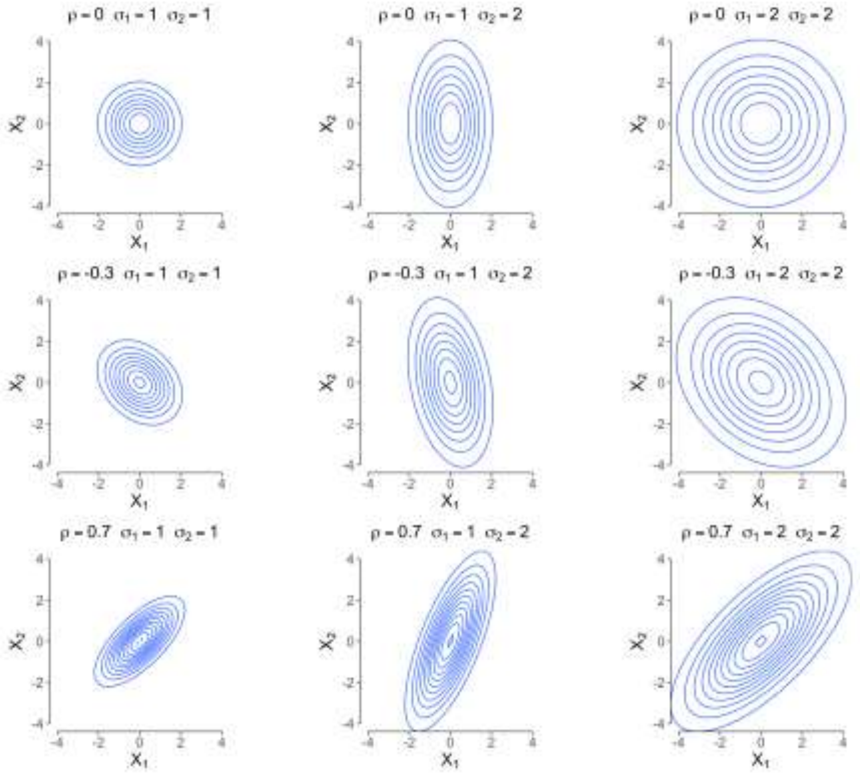
CVPR2025

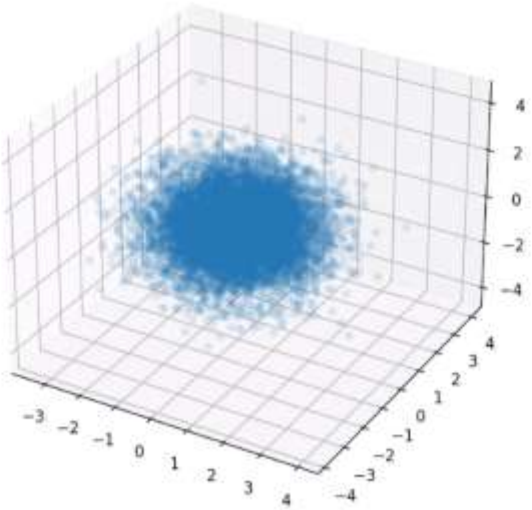
什么是3D高斯

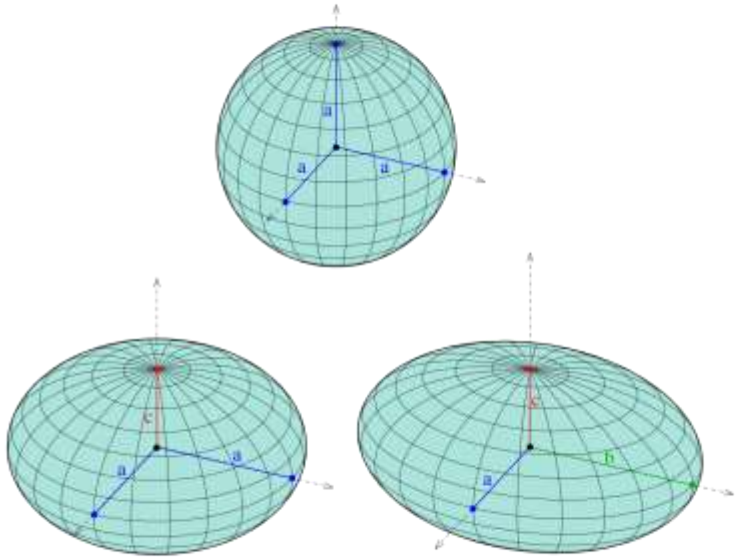
$$g(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)$$

$$\Sigma = RSS^TR^T$$

μ R S O C







如何渲染高斯椭球

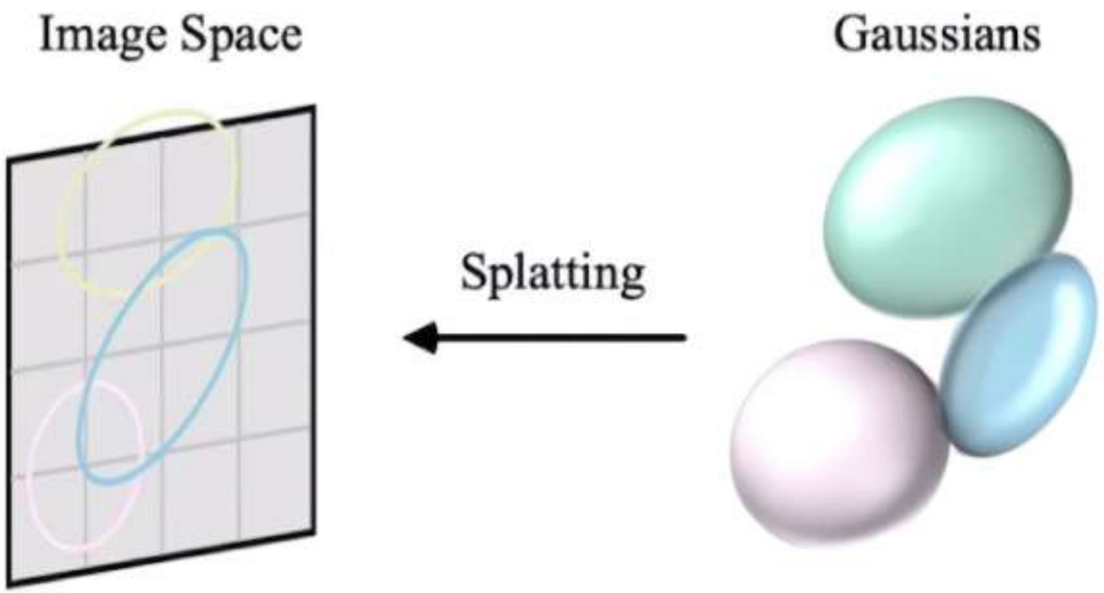
$$\int_{\mathbb{R}} G_{\mathbf{\Sigma}}(\mathbf{x}-\mu)dz = \hat{G}_{\hat{\mathbf{\Sigma}}}(\hat{\mathbf{x}}-\hat{\mu})$$

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

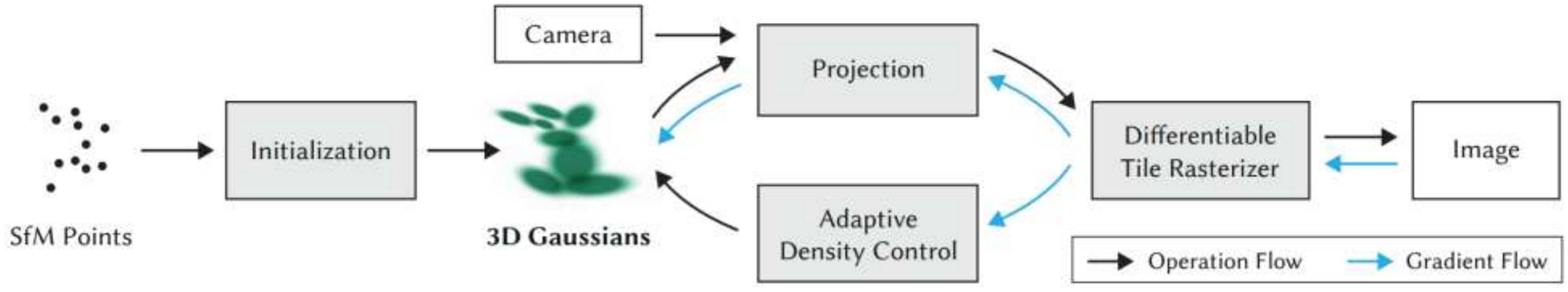
J 为雅各布矩阵，将非线性的三维到二维的投影变换转换为近似的投影变换

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

$$\alpha_i = \min\left(0.99, o_i \cdot \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{p_i})^T {\Sigma'}^{-1}(\mathbf{x}-\mathbf{p_i})\right)\right)$$

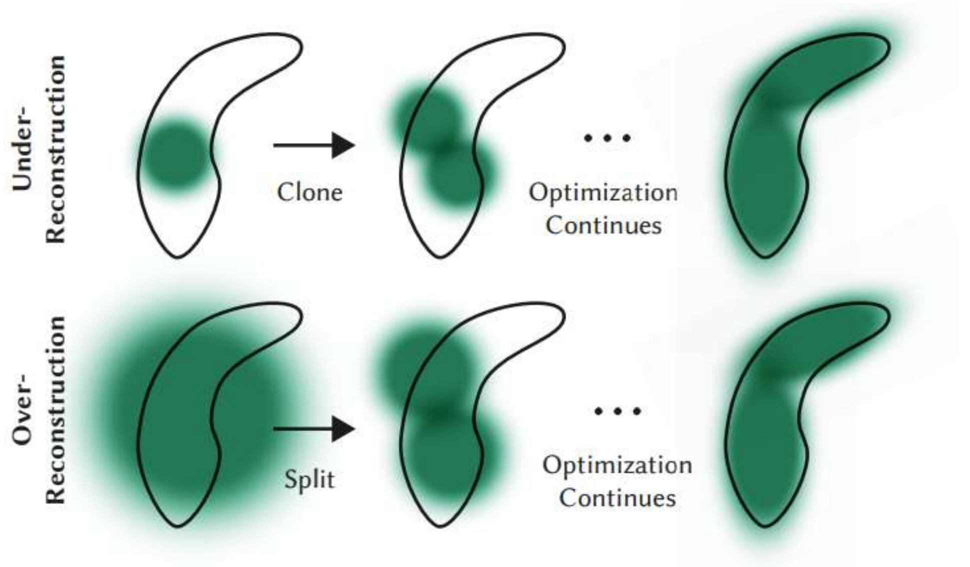


优化过程

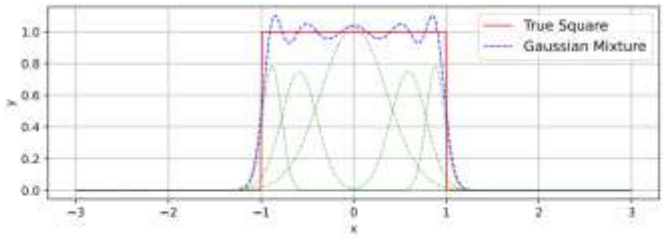


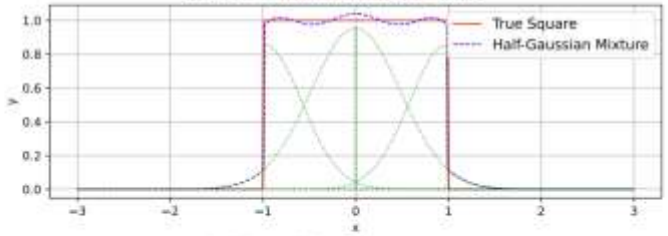
$$\mathcal{L} = (1-\lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{DSSIM}$$

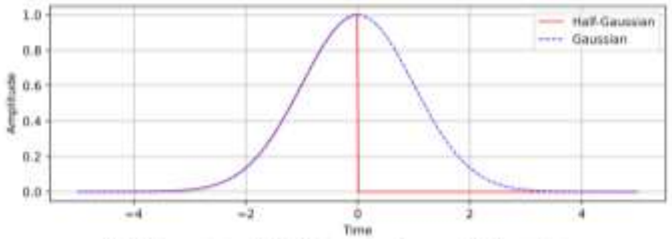
- $\mathcal{L}_1 = \|\mathbf{C}-\mathbf{C}_{gt}\|_1$：渲染颜色$\mathbf{C}$与ground truth $\mathbf{C}_{gt}$的绝对差。
- $\mathcal{L}_{DSSIM} = 1 - SSIM(\mathbf{C},\mathbf{C}_{gt})$：捕捉结构相似性，通常$\lambda = 0.2$。

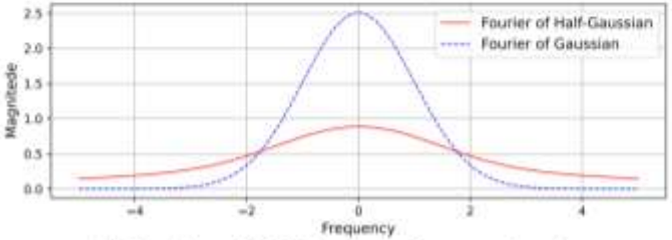


在拟合高频分布时的缺陷,引入半高斯


(a) Five Gaussians fitting a square

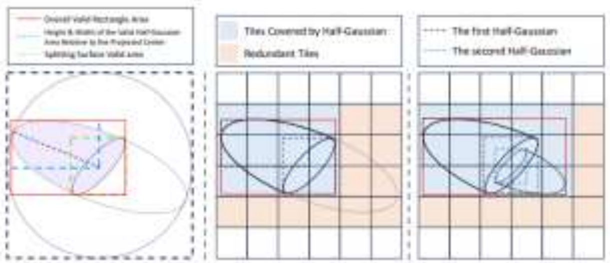

(b) Three HGs fitting a square


(c) Gaussian and Half-Gaussian in spatial domain


(d) Gaussian and Half-Gaussian in Frequency domain

半高斯的形式化定义

半高斯



$$HG_{\mathbf{\Sigma}}(\mathbf{x}-\mu) = \begin{cases} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})} & \mathbf{n}^T(\mathbf{x}-\mu) \geq 0 \\ 0 & \mathbf{n}^T(\mathbf{x}-\mu) < 0 \end{cases}$$

$$\int_{\mathbf{n}^T(\mathbf{x}-\mu)\geq 0} HG_{\mathbf{\Sigma}}(\mathbf{x}-\mu)dz = \frac{1}{2}I(x,y)\hat{G}_{\hat{\mathbf{\Sigma}}}(\hat{\mathbf{x}}-\hat{\mu})$$

where

$$I(x,y) = \operatorname{erfc}\left(-\frac{(n_1x+n_2y)/n_3+\mu_{z|xy}}{\sqrt{2}\sigma_{z|xy}}\right)$$

$$C = \sum_{i\in\mathcal{N}} c_i H\hat{G}_i(\hat{\mathbf{x}}-\hat{\mu})\prod_{j=1}^{i-1}\left(1-H\hat{G}_j(\hat{\mathbf{x}}-\hat{\mu})\right)$$

$$H\hat{G}(\hat{\mathbf{x}}-\hat{\mu}) = \frac{1}{2}\left\{2\alpha_2+(\alpha_1-\alpha_2)\,I(x,y)\right\}\hat{G}_{\hat{\mathbf{\Sigma}}}(\hat{\mathbf{x}}-\hat{\mu})$$

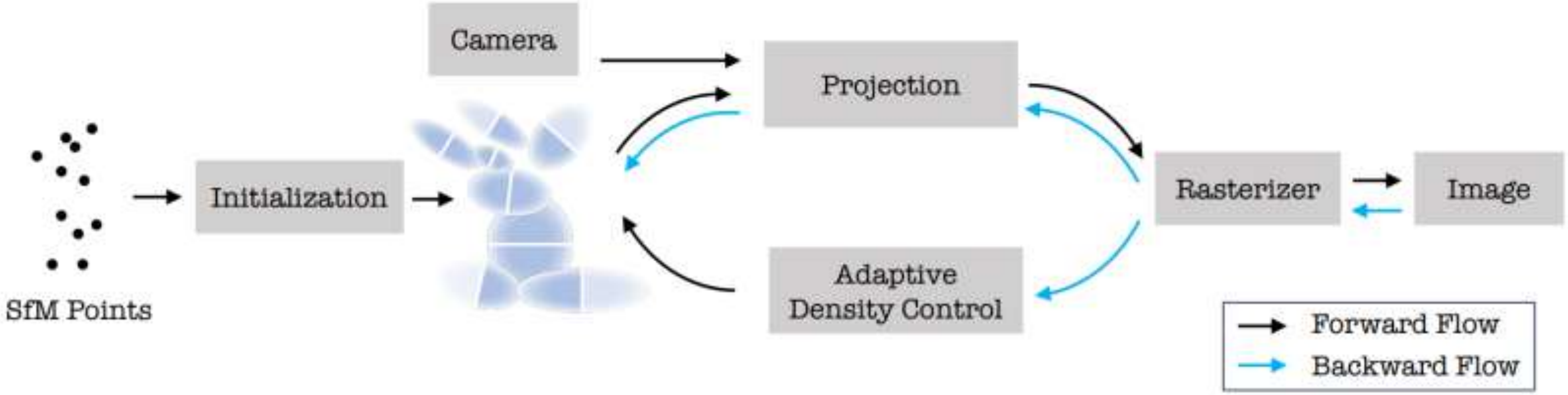
三维高斯

$$g(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)$$

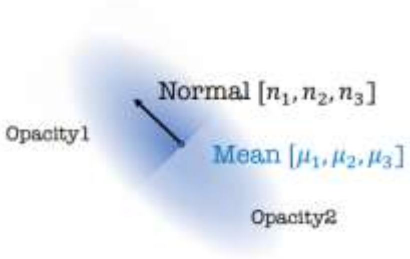
$$\int_{\mathbb{R}} G_{\mathbf{\Sigma}}(\mathbf{x}-\mu)dz = \hat{G}_{\hat{\mathbf{\Sigma}}}(\hat{\mathbf{x}}-\hat{\mu})$$

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

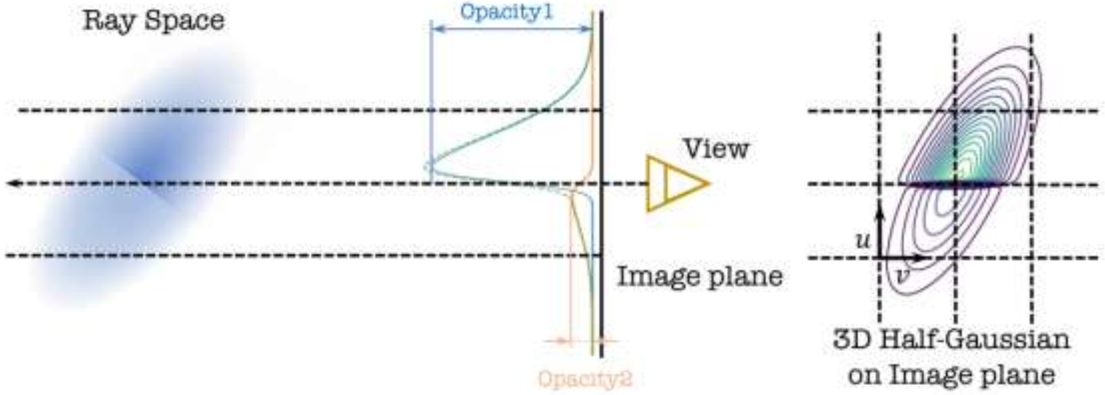
$$\alpha_i = \min(0.99, o_i\cdot\exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{p_i})^TV_i^{-1}(\mathbf{x}-\mathbf{p_i})\right))$$



(a) The training and rendering pipeline for 3D Half-Gaussian



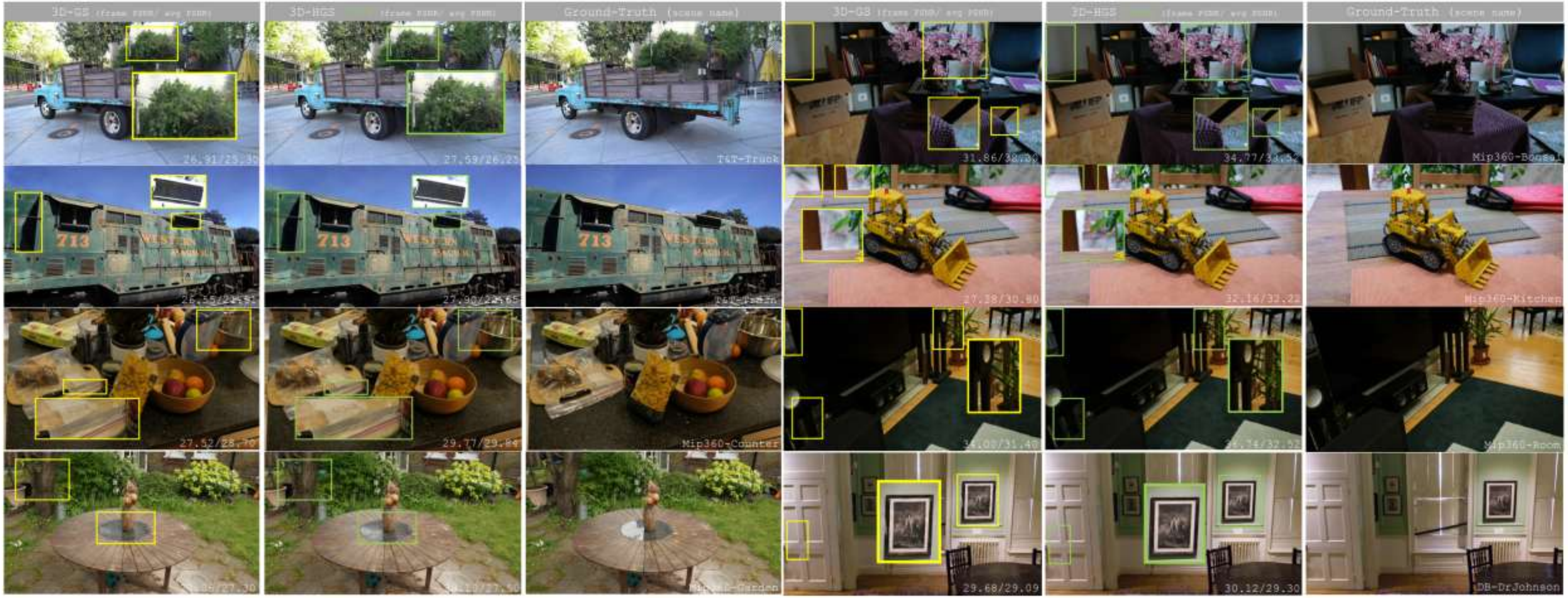
(b) 3D Half-Gaussian pair



(c) Mapping the Half-Gaussian to image plane in the ray space

Figure 4. Illustration of the 3D-HG kernel, and the mapping of a pair of 3D Half-Gaussians to a 2D image.

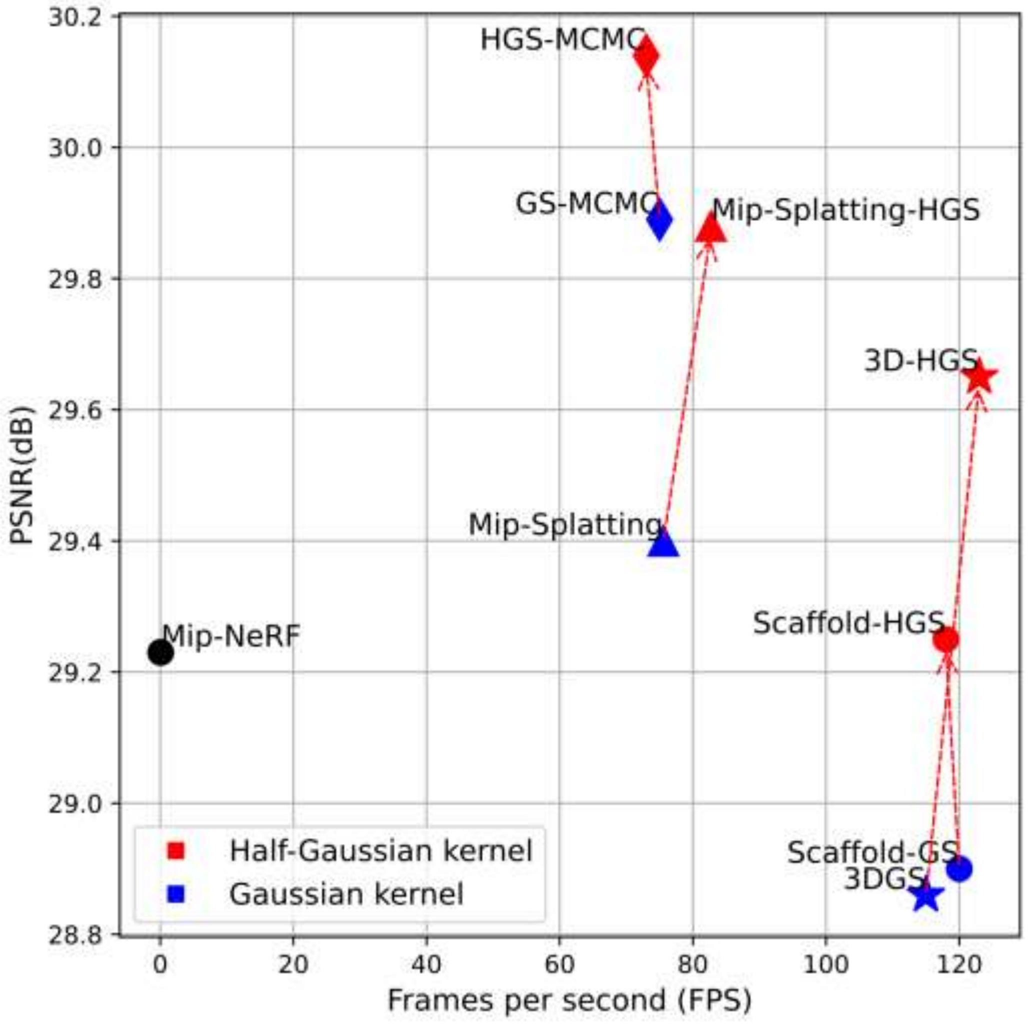
实验结果


3D-GS
3D-HGS
Ground-Truth
T&T-Truck
Mip360-Kitchen
Mip360-Room
29.68/29.09
30.12/29.30
DB-DrJohnson

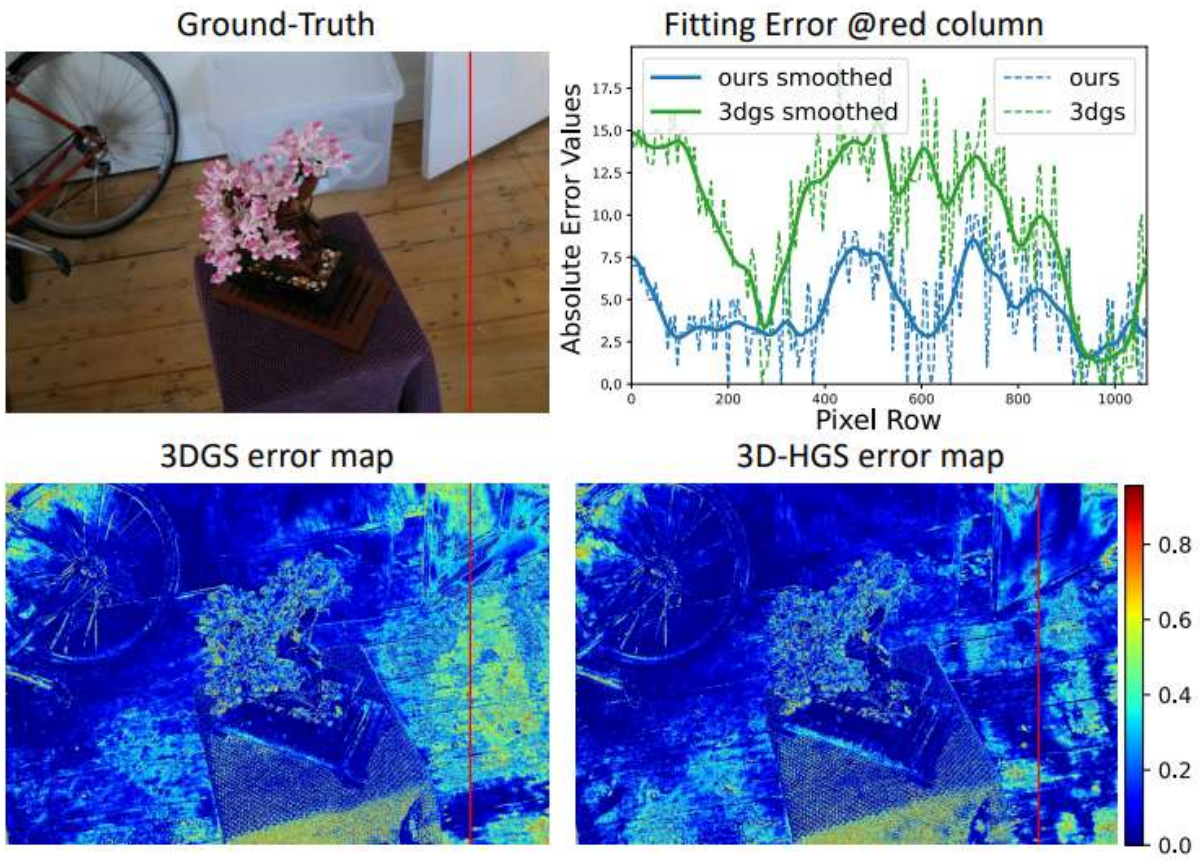
实验结果

Dataset	Mip-NeRF360			Tanks&Temples			Deep Blending		
Method	PSNR ↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Mip-NeRF [1]	29.23	0.844	0.207	22.22	0.759	0.257	29.40	0.901	0.245
2D-GS[8]	28.98	0.867	0.185	23.43	0.845	0.181	29.70	0.902	0.250
Fre-GS [30]	27.85	0.826	0.209	23.96	0.841	0.183	29.93	0.904	0.240
GES [6]	28.69	0.857	0.206	23.35	0.836	0.198	29.68	0.901	0.252
3D-GS [9]	28.88	0.870	0.182	23.60	0.847	0.181	29.41	0.903	0.243
3D-**HGS** (**Ours**)	29.66 **+0.78**	0.873	0.178	24.45 **+0.85**	0.857	0.169	29.76 **+0.35**	0.905	0.242
Scaffold-GS [15]	28.95	0.848	0.220	23.96	0.853	0.177	30.21	0.906	0.254
Scaffold-**HGS**(**Ours**)	29.25 **+0.30**	0.867	0.186	24.42 **+0.46**	0.859	0.162	30.36 **+0.15**	0.910	0.240
Mip-Splatting [29]	29.39	0.880	0.162	23.75	0.857	0.157	29.46	0.903	0.243
Mip-Splatting-**HGS**(**Ours**)	29.88 **+0.49**	0.881	0.160	24.53 **+0.78**	0.865	0.145	29.61 **+0.15**	0.901	0.241
GS-MCMC [10]	29.89	0.900	0.190	24.29	0.860	0.190	29.67	0.890	0.320
HGS-MCMC(**Ours**)	30.13 **+0.24**	0.886	0.158	25.08 **+0.77**	0.841	0.144	29.80 **+0.13**	0.898	0.245

实验结果



HGS-MCMC
GS-MCMC
Mip-Splatting-HGS
3D-HGS
Mip-Splatting
Mip-NeRF
Scaffold-HGS
Scaffold-GS
3DGS
Half-Gaussian kernel
Gaussian kernel
PSNR(dB)
Frames per second (FPS)
30.2
30.0
29.8
29.6
29.4
29.2
29.0
28.8
0
20
40
60
80
100
120



Ground-Truth
Fitting Error @red column
ours smoothed
3dgs smoothed
ours
3dgs
Absolute Error Values
Pixel Row
3DGS error map
3D-HGS error map
0.8
0.6
0.4
0.2
0.0

实验结果

Dataset	Mip-NeRF360		Tanks&Temples		Deep Blending	
Method	FPS ↑	Mem ↓	FPS ↑	Mem ↓	FPS ↑	Mem ↓
3D-GS [9]	115	762	149	429	104	668
3D-**HGS** (**Ours**)	125	694	160	437	126	641
Scaffold-GS [15]	120	173	120	77	129	55
Scaffold-**HGS**(**Ours**)	118	180	115	84	136	53
GS-MCMC [10]	75	732	133	438	90	969
HGS-MCMC(**Ours**)	72	743	139	445	92	980
Mip-Splatting [10]	76	970	117	569	91	843
Mip-Splatting-**HGS**(**Ours**)	83	883	121	566	102	808

谢谢！