3D-HGS: 3D Half-Gaussian Splatting

Supplementary Material

A. Detailed Derivations

A.1. 3D Half-Gaussian Kernel

In this section, we give a more detailed derivation for Eq. 8, and Eq. 9. We start with the density function of a 3D Half-Gaussian with covariance matrix Σ:

$$HG_{\mathbf{\Sigma}}(\mathbf{x}) = \begin{cases} e^{-\frac{1}{2}\mathbf{x}^T\mathbf{\Sigma}^{-1}\mathbf{x}} & \mathbf{n}^T\mathbf{x} \geq 0 \\ 0 & \mathbf{n}^T\mathbf{x} < 0 \end{cases} \tag{12}$$

To compute the integral of the non-zero Half Gaussian along the z direction, $\int_{\mathbf{n}^T\mathbf{x}\geq 0} HG_{\mathbf{\Sigma}}(\mathbf{x})dz$, we start by calculating the conditional distribution of z, given x and y and the marginal distribution of x and y:

$$f(z|x,y) = \exp\left(-\frac{1}{2}\left(\frac{z-\mu_{z|xy}}{\sigma_{zz|xy}}\right)^2\right) \tag{13}$$

$$f(x,y) = \exp\left(-\frac{1}{2}\begin{bmatrix}x\ y\end{bmatrix}\Sigma_{xx}^{-1}\begin{bmatrix}x\\y\end{bmatrix}\right) \tag{14}$$

where $\mu_{z|xy}$ and $\sigma_{zz|xy}$ are the mean and variance of the conditional distribution, respectively. Σ_{xx} is defined as the top left 2×2 covariance submatrix of Σ:

$$\Sigma = \begin{bmatrix}\Sigma_{xx} & \Sigma_{xz}\\ \Sigma_{zx} & \sigma_{zz}\end{bmatrix} \tag{15}$$

and σ_{zz} is the variance for z. Then $\mu_{z|xy}$ and $\sigma_{zz|xy}$ can be expressed as:

$$\mu_{z|xy} = \Sigma_{zx}\Sigma_{xx}^{-1}\begin{bmatrix}x\\y\end{bmatrix} \tag{16}$$

$$\sigma_{zz|xy} = \sigma_{zz} - \Sigma_{zx}\Sigma_{xx}^{-1}\Sigma_{xz} \tag{17}$$

Then, the integral of the 3D Half-Gaussian can be computed as:

$$\begin{aligned}\int_{\mathbf{n}^T\mathbf{x}\geq 0} HG_{\mathbf{\Sigma}}(\mathbf{x})dz &= f(x,y)\int_{-\frac{n_1x+n_2y}{n_3}}^{\infty} f(z|x,y)dz \\ &= \exp\left(-\frac{1}{2}\begin{bmatrix}x\ y\end{bmatrix}\Sigma_{xx}^{-1}\begin{bmatrix}x\\y\end{bmatrix}\right) \\ &\int_{-\frac{n_1x+n_2y}{n_3}}^{\infty}\exp\left(-\frac{1}{2}\left(\frac{z-\mu_{z|xy}}{\sigma_{zz|xy}}\right)^2\right)dz\end{aligned} \tag{18}$$

where $\exp\left(-\frac{1}{2}\begin{bmatrix}x\ y\end{bmatrix}\Sigma_{xx}^{-1}\begin{bmatrix}x\\y\end{bmatrix}\right)$ is the same as $\hat{G}_{\hat{\mathbf{\Sigma}}}(\hat{\mathbf{x}}-\hat{\mu})$. We apply the substitution $u = \frac{z-\mu_{z|xy}}{\sqrt{2}\sigma_{zz|xy}}$, then we get:

$$\begin{aligned}&\int_{\mathbf{n}^T\mathbf{x}\geq 0} HG_{\mathbf{\Sigma}}(\mathbf{x})dz \\ &\qquad = \hat{G}_{\hat{\mathbf{\Sigma}}}(\hat{\mathbf{x}}-\hat{\mu})\int_{\frac{-\frac{n_1x+n_2y}{n3}-\mu_{z|xy}}{\sqrt{2}\sigma_{zz|xy}}}^{\infty} e^{-u^2}du\end{aligned} \tag{19}$$

The final result of the integral of the 3D Half-Gaussian is:

$$\begin{aligned}&\int_{\mathbf{n}^T\mathbf{x}\geq 0} HG_{\mathbf{\Sigma}}(\mathbf{x})dz \\ &= \hat{G}_{\hat{\mathbf{\Sigma}}}(\hat{\mathbf{x}}-\hat{\mu})\frac{1}{2}\operatorname{erfc}\left(-\frac{\left(\begin{bmatrix}n_1\ n_2\end{bmatrix}/n_3 + \Sigma_{zx}\Sigma_{xx}^{-1}\right)\begin{bmatrix}x\\y\end{bmatrix}}{\sqrt{2\left(\sigma_{zz}-\Sigma_{zx}\Sigma_{xx}^{-1}\Sigma_{xz}\right)}}\right)\end{aligned} \tag{20}$$

A.2. Fourier transform of 1D Half-Gaussian

In this section, we present a detailed derivation of the Fourier transform for the 1D Half-Gaussian, which was utilized to generate Fig. 2d. We begin with the definition of the half-Gaussian:

$$f(x) = \begin{cases} e^{-\frac{x^2}{2\sigma^2}}, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{21}$$

The following provides the derivation of its Fourier transform:

$$g(k) = \int_{-\infty}^{\infty} HG(x)e^{-ikx}dx \tag{22}$$

$$= \int_0^{\infty} e^{-\frac{x^2}{2\sigma^2}}e^{-ikx}dx \tag{23}$$

$$= \int_0^{\infty} e^{\frac{-1}{2\sigma^2}(x^2+2i\sigma^2kx)}dx \tag{24}$$

$$= \int_0^{\infty} e^{\frac{-1}{2\sigma^2}((x+i\sigma^2k)^2-(i\sigma^2k)^2)}dx \tag{25}$$

$$= e^{-\frac{\sigma^2k^2}{2}}\int_0^{\infty} e^{-\frac{1}{2\sigma^2}(x+i\sigma^2k)^2}dx \tag{26}$$

Now we apply the substitution $u = x + i\sigma^2k$, where $du = dx$. Under this substitution, the integration limits transform. The integral becomes:

$$g(k) = e^{-\frac{\sigma^2k^2}{2}}\int_{i\sigma^2k}^{\infty} e^{-\frac{u^2}{2\sigma^2}}du \tag{27}$$

Table 4. **Quantitative comparison to previous methods on standard benchmarks. Competing metrics are directly taken from the respective papers for consistency.**

		3DGS	3DHGS	Scaffold-HGS	3DHGS-Mip-Splatting	3DHGS-MCMC
		PSNR↑/SSIM↑/LPIPS↓	PSNR↑/SSIM↑/LPIPS↓	PSNR↑/SSIM↑/LPIPS↓	PSNR↑/SSIM↑/LPIPS↓	PSNR↑/SSIM↑/LPIPS↓
MipNeRF360	Counter	28.70/0.905/0.204	29.84/0.920/0.177	29.52/0.914/0.183	30.11/0.925/0.166	30.24/0.928/0.161
	Stump	26.56/0.770/0.217	26.64/0.760/0.242	26.45/0.76/0.259	26.94/0.777/0.215	27.17/0.794/0.202
	Kitchen	30.80/0.927/0.127	32.22/0.936/0.113	31.67/0.931/0.113	32.56/0.938/0.108	32.72/0.940/0.108
	Bicycle	25.21/0.765/0.209	25.25/0.750/0.230	25.05/0.74/0.258	25.43/0.773/0.189	25.52/0.777/0.196
	Bonsai	32.20/0.946/0.183	33.52/0.950/0.180	32.75/0.947/0.176	33.55/0.952/0.161	34.33/0.958/0.159
	Room	31.40/0.918/0.223	32.52/0.931/0.193	32.21/0.929/0.184	32.80/0.937/0.175	33.25/0.940/0.169
	Garden	27.30/0.865/0.107	27.50/0.860/0.110	27.12/0.848/0.130	27.80/0.865/0.106	27.67/0.864/0.110
	Average	28.88/0.870/0.182	29.66/0.873/0.178	29.25/0.867/0.186	29.88/0.881/0.160	30.13/0.886/0.158
	Flowers	21.52/0.605/0.336	21.55/0.610/0.250	/	/	/
	Treehill	22.49/0.638/0.317	22.63/0.640/0.330	/	/	/
Tanks & Temples	Train	21.91/0.815/0.210	22.65/0.826/0.193	22.63/0.828/0.186	22.65/0.836/0.176	23.24/0.841/0.180
	Truck	25.30/0.88/0.152	26.25/0.887/0.138	26.17/0.889/0.138	26.42/0.894/0.115	26.91/0.902/0.107
	Average	23.60/0.847/0.181	24.45/0.857/0.169	24.42/0.859/0.162	24.53/0.865/0.145	25.08/0.841/0.144
Deep Blending	Dr Johnson	29.02/0.902/0.244	29.30/0.903/0.240	29.82/0.908/0.236	29.18/0.901/0.241	29.02/0.890/0.26
	Playroom	29.81/0.904/0.242	30.22/0.907/0.244	30.90/0.912/0.244	30.04/0.901/0.241	30.58/0.905/0.231
	Average	29.41/0.903/0.243	29.76/0.905/0.242	30.36/0.910/0.240	29.61/0.901/0.241	29.80/0.898/0.245

The remaining integral can be evaluated using the complex error function erfi(z), resulting in the final expression:

$$g(k) = e^{-\frac{\sigma^2 k^2}{2}} \sqrt{\frac{\pi}{2}} \left(\sigma - i\sigma \,\text{erfi}\left(\frac{k\sigma}{\sqrt{2}} \right) \right) \tag{28}$$

This provides the Fourier transform of the half-Gaussian function.

B. Implementation Details

B.1. Splatting for Splitting Plane

For simplicity, we define the splitting plane as intersecting the 3D ellipsoid at the three-sigma boundary of the original Gaussian and oriented orthogonal to the normal direction. To approximate the splitting plane, we use an external ellipsoid that circumscribes the original 3D ellipsoid, with a significantly shortened third axis aligned parallel to the normal of the splitting plane.

During rendering, the precomputed 2D projection of the splitting plane facilitates efficient determination of the inner boundary of the Gaussian. Combined with the outer boundary derived from the original Gaussian, this enables accurate partitioning into two halves and precise identification of the valid effect areas for each half-Gaussian. This approach achieves rendering speeds that are comparable to or even exceed those of standard Gaussian splatting.

Additionally, we provide a detailed procedure to compute the covariance matrix for the smaller circumscribing ellipsoid based on the covariance matrix of the original Gaussian and the normal vector of the cutting plane in Algorithm 1.

Algorithm 1 Covariance Matrix Transformation

$\mathbf{\Sigma}$: The covariance matrix for the 3D gaussian
$\mathbf{n}$: The normal for the cutting plane
$\mathbf{\Sigma}$: the covariance for the ellipsoid include the cutting plane

Normalize the normal vector: $\mathbf{n} \leftarrow \mathbf{n}/\|\mathbf{n}\|$
if $n_x = 0$ and $n_y = 0$ **then**
  Set $\mathbf{v}_1 = (1, 0, 0)$ and $\mathbf{v}_2 = (0, 1, 0)$
else
  $\mathbf{v}_1 \leftarrow (n_y, -n_x, 0)/\|(n_y, -n_x, 0)\|$
  $\mathbf{v}_2 \leftarrow \mathbf{n} \times \mathbf{v}_1/\|\mathbf{n} \times \mathbf{v}_1\|$
end if
basis $\leftarrow$ concatenate$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{n})$
cov $\leftarrow$ basis$^T \cdot \Sigma \cdot$ basis
$a' \leftarrow \text{cov}[0,0] - \frac{\text{cov}[0,2]^2}{\text{cov}[2,2]}$
$b' \leftarrow \text{cov}[0,1] - \frac{\text{cov}[0,2]\text{cov}[1,2]}{\text{cov}[2,2]}$
$c' \leftarrow \text{cov}[1,1] - \frac{\text{cov}[1,2]\text{cov}[1,2]}{\text{cov}[2,2]}$
$\sigma \leftarrow \begin{bmatrix} a' & b' & 0 \\ b' & c' & 0 \\ 0 & 0 & \min(a', c')/100 \end{bmatrix}$
$\sigma \leftarrow$ basis $\cdot \sigma \cdot$ basis^T

B.2. Training Settings

During training, we introduce a new parameter, the normal vector, and assign it a dedicated learning rate of 0.003. Unlike original Gaussian Splatting, each Gaussian in our framework has two opacities. The initial learning rates for opacities are consistent with those used in Gaussian Splatting. In 3D-HGS, the learning rates for both opacity and the normal vector are progressively reduced by a factor of 1.4

every 5000 iterations.

To further improve training efficiency, we adjust the thresholds for opacity during the densification and pruning stages. Specifically, we increase the opacity threshold to 0.01 and reset the opacity value for half-Gaussians to 0.02. These adjustments help to effectively eliminate dust and noise from the scene, leading to cleaner and more accurate representations.

C. Additional Experiments

C.1. Detailed Results

We evaluated the performance of 3D-HGS and other 3D Gaussian architectures, including Scaffold, mip-splatting, and MCMC, using our 3D half-Gaussian kernel and compared to the original 3D Gaussian Splatting. Table 1 in the main paper summarizes the average results derived from Table 4. Our method consistently outperforms the original 3D Gaussian Splatting across all metrics. Each average reported in Table 4 was computed across all scenes within the dataset.

C.2. Splitting Plane Visualization

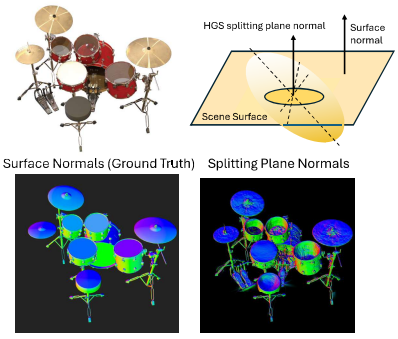

Figure 10. **Splitting Plane Visualization.** The normals to the splitting planes are parallel to the corresponding 3D surface normals. Please note that the tops of three of the drums are transparent.

In order to visualize the locations of the splitting planes, we provide side by side the scene surface normals and the splitting plane normals for the drums scene in Fig. 10. The figure shows that the normals of the splitting planes generally align well with the normals of the scene surfaces, indicating that the half ellipsoids effectively model the sharp transitions between the object surfaces and the empty space. This is also supported by Fig. 5 in the main paper, which shows that a large number of Gaussians have one of their halves transparent.

C.3. Training time vs PSNR

As shown in Fig. 11, for the same training time, 3DHGS using a similar or smaller number of Gaussians achieves better PSNR than using 3DGS.

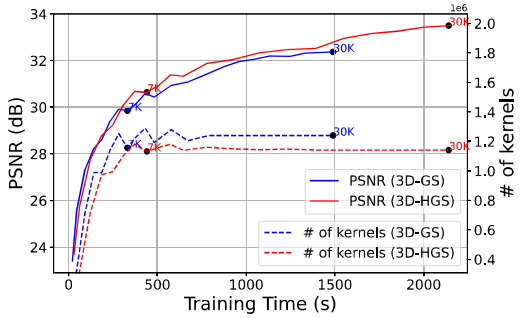

Figure 11. **Training Time:** Performance (PSNR↑) and memory (number of kernels↓) versus training time for the Bonsai scene. Numbers on the curves indicate number of iterations.

C.4. Rendering Speed Ablation Study

Table 5. **Ablation study** on Rendering Speed on a single RTX-3090. A 3DGHS kernel consists of its two half-Gaussians.

Dataset	# of Kernels		3DHGS FPS / PSNR	
	3DGS	3DHGS	w/o Eff. Rasterizer	with Eff. Rasterizer
Mip-NeRF 360	3.22 M	2.88 M	76 / 29.56	125 / 29.66
Tank & Template	1.81 M	1.81 M	95 / 24.49	160 / 24.45
Deep Blending	3.00 M	2.66 M	82 / 29.76	126 / 29.76

We performed an ablation study on the proposed efficient 3D half-Gaussian Splatting Rasterizer (Sec.3.4). Table 5 reports the rendering speed in FPS and the performance in PSNR for our method, 3DHGS, with and without the efficient Rasterizer. The results highlight a significant improvement in rendering speed with our proposed Rasterizer. Additionally, we provide the average number of kernels used by 3DGS and 3DHGS for reference.