Supplemental to AAA-Gaussians: Anti-Aliased and Artifact-Free 3D Gaussian Rendering

Michael Steiner*[1] Thomas Köhler*[1] Lukas Radl[1]
Felix Windisch[1] Dieter Schmalstieg[1,2] Markus Steinberger[1,3]
[1]Graz University of Technology [2]University of Stuttgart [3]Huawei Technologies

A. Derivation of Amplitude Scaling Factor

Let

$$\mathbf{d} = \frac{\boldsymbol{\mu} - \mathbf{o}}{\|\boldsymbol{\mu} - \mathbf{o}\|}$$

be a unit vector in $\mathbb{R}^3$, where $\boldsymbol{\mu}$ is the mean of the Gaussian and $\mathbf{o}$ is the camera position in world space. We are interested in the area of the Gaussian's intersection with the plane perpendicular to $\mathbf{d}$.

Let

$$\mathbf{U} = \begin{pmatrix} | & | & | \\ \mathbf{d} & \mathbf{u}_2 & \mathbf{u}_3 \\ | & | & | \end{pmatrix} \in \mathbb{R}^{3\times 3}$$

be an orthonormal basis with $\mathbf{d}$ as the first basis vector. The orthogonal vectors $\mathbf{u}_2$ and $\mathbf{u}_3$ may be arbitrarily oriented around $\mathbf{d}$, since we are only interested in the size of the area. We can perform an orthogonal change of basis on $\boldsymbol{\Sigma}$:

$$\boldsymbol{\Sigma}' = \mathbf{U}^\top \boldsymbol{\Sigma} \mathbf{U}.$$

Note that this transformation preserves Eigenvalues, because $\mathbf{U}$ is orthogonal. $\boldsymbol{\Sigma}'$ can be decomposed such that

$$\boldsymbol{\Sigma}' = \begin{pmatrix} \sigma_{11} & \boldsymbol{\sigma}_{12}^\top \\ \boldsymbol{\sigma}_{12} & \boldsymbol{\Sigma}_{\perp}, \end{pmatrix}$$

where:

- $\sigma_{11} \in \mathbb{R}$ is the $(1,1)$ entry,
- $\boldsymbol{\sigma}_{12} \in \mathbb{R}^2$ is the off-diagonal block,
- $\boldsymbol{\Sigma}_\perp \in \mathbb{R}^{2\times 2}$ is an orthogonal projection of $\boldsymbol{\Sigma}$ (arbitrarily rotated around $\mathbf{d}$) onto the perpendicular subspace of $\mathbf{d}$.

The area of the projected Gaussian is then simply given by the determinant of $\boldsymbol{\Sigma}_\perp$. We can find the determinant by applying the Schur Complement to $\boldsymbol{\Sigma}'$:

$$|\boldsymbol{\Sigma}'| = |\boldsymbol{\Sigma}_\perp| \cdot \left(\sigma_{11} - \boldsymbol{\sigma}_{12}^\top \boldsymbol{\Sigma}_\perp^{-1} \boldsymbol{\sigma}_{12}\right) .$$

Because of the orthogonal change of basis, $|\boldsymbol{\Sigma}| = |\boldsymbol{\Sigma}'|$ and

$$|\boldsymbol{\Sigma}_\perp| = \frac{1}{\sigma_{11} - \boldsymbol{\sigma}_{12}^\top \boldsymbol{\Sigma}_\perp^{-1} \boldsymbol{\sigma}_{12}} |\boldsymbol{\Sigma}|. \tag{1}$$

[1]Both authors contributed equally to this work

Using the standard formula for the inverse of a block matrix, we can rewrite the reciprocal of the Schur complement as the $(1,1)$ entry of $\boldsymbol{\Sigma}'^{-1}$,

$$\mathbf{e}_1^\top \boldsymbol{\Sigma}'^{-1} \mathbf{e}_1 = \frac{1}{\sigma_{11} - \boldsymbol{\sigma}_{12}^\top \boldsymbol{\Sigma}_{22}^{-1} \boldsymbol{\sigma}_{12}} \tag{2}$$

with $\mathbf{e}_1 = (1,0,0)^\top$. Due to the orthogonality of $\mathbf{U}$,

$$\begin{aligned} \mathbf{e}_1^\top \boldsymbol{\Sigma}'^{-1} \mathbf{e}_1 &= \mathbf{e}_1^\top (\mathbf{U}^\top \boldsymbol{\Sigma} \mathbf{U})^{-1} \mathbf{e}_1 \\ &= \mathbf{e}_1^\top \mathbf{U} \boldsymbol{\Sigma}^{-1} \mathbf{U}^\top \mathbf{e}_1 \\ &= \mathbf{d}^\top \boldsymbol{\Sigma}^{-1} \mathbf{d}. \end{aligned} \tag{3}$$

Combining Eqns. (1,2,3) results in

$$|\boldsymbol{\Sigma}_\perp| = |\boldsymbol{\Sigma}| \cdot \mathbf{d}^\top \boldsymbol{\Sigma}^{-1} \mathbf{d} \,,$$

and in turn our perpendicular scaling factor is equal to:

$$\sqrt{\frac{|\boldsymbol{\Sigma}_\perp|}{|\boldsymbol{\Sigma}_\perp + k\mathbf{I}|}} = \sqrt{\frac{|\boldsymbol{\Sigma}| \cdot \mathbf{d}^\top \boldsymbol{\Sigma}^{-1} \mathbf{d}}{|\boldsymbol{\Sigma}| \cdot \mathbf{d}^\top \left(\boldsymbol{\Sigma} + k\mathbf{I}\right)^{-1} \mathbf{d}}} .$$

B. Derivation of Bounds

We perform the plane fitting in view space with planes $\boldsymbol{\pi}_\theta = (\cos(\theta), 0, -\sin(\theta), 0)^\top, \boldsymbol{\pi}_\phi = (0, \cos(\phi), -\sin(\phi), 0)^\top$, and their transformed counterparts in Gaussian space $\boldsymbol{\pi}'_\theta, \boldsymbol{\pi}'_\phi$:

$$\boldsymbol{\pi}'_\theta = \mathbf{T}_{\text{view}}^\top \boldsymbol{\pi}_\theta = \cos(\theta)\mathbf{T}_{(\text{view},1)} - \sin(\theta)\mathbf{T}_{(\text{view},3)}, \tag{4}$$

$$\boldsymbol{\pi}'_\phi = \mathbf{T}_{\text{view}}^\top \boldsymbol{\pi}_\phi = \cos(\phi)\mathbf{T}_{(\text{view},2)} - \sin(\phi)\mathbf{T}_{(\text{view},3)}, \tag{5}$$

with $\mathbf{T}_{\text{view}} = \mathbf{VT}$ being the transformation matrix from Gaussian space to view space via view-matrix $\mathbf{V}$, and $\mathbf{T}_{(\text{view},i)}$ denoting the i-th row of $\mathbf{T}_{\text{view}}$. For simplicity, we will refer to $\mathbf{T}_{(\text{view},i)}$ as $\mathbf{T}_i$ in the following derivation.

Following Sigg *et al.* [2], the touching condition to the cutoff ellipsoid in Gaussian space for these planes is

$$\boldsymbol{\pi}'^\top \mathbf{Q} \boldsymbol{\pi}' = 0 \tag{6}$$

with $\mathbf{Q} \in \mathbb{R}^{4\times4}$ being a diagonal matrix, which is defined as $\mathbf{Q} = \text{diag}(\boldsymbol{t}), \boldsymbol{t} = (\tau_\rho, \tau_\rho, \tau_\rho, -1)^\top$. For $\boldsymbol{\pi}'_\theta$, this simplifies to

$$
\begin{aligned}
&(\cos(\theta)\mathbf{T}_1 - \sin(\theta)\mathbf{T}_3)^\top \mathbf{Q}(\cos(\theta)\mathbf{T}_1 - \sin(\theta)\mathbf{T}_3) \\
&= \cos(\theta)^2 \mathbf{T}_1^\top \mathbf{Q}\mathbf{T}_1 - 2\sin(\theta)\cos(\theta)\mathbf{T}_1^\top \mathbf{Q}\mathbf{T}_3 + \sin(\theta)^2 \mathbf{T}_3^\top \mathbf{Q}\mathbf{T}_3 \\
&= \tan(\theta)^2 \mathbf{T}_1^\top \mathbf{Q}\mathbf{T}_1 - 2\tan(\theta)\mathbf{T}_1^\top \mathbf{Q}\mathbf{T}_3 + \mathbf{T}_3^\top \mathbf{Q}\mathbf{T}_3 \\
&= \tan(\theta)^2 \langle \boldsymbol{t}, \mathbf{T}_1 \odot \mathbf{T}_1 \rangle - 2\tan(\theta)\langle \boldsymbol{t}, \mathbf{T}_1 \odot \mathbf{T}_3 \rangle + \langle \boldsymbol{t}, \mathbf{T}_3 \odot \mathbf{T}_3 \rangle.
\end{aligned}
$$

By solving this quadratic equation w.r.t. $\tan(\theta)$ (and similarly $\tan(\phi)$), we find solutions for θ, ϕ:

$$
\theta_{1,2} = \tan^{-1}\left(\frac{s_{1,3} \pm \sqrt{s_{1,3}^2 - s_{1,1}s_{3,3}}}{s_{3,3}}\right), \tag{7}
$$

$$
\phi_{1,2} = \tan^{-1}\left(\frac{s_{2,3} \pm \sqrt{s_{2,3}^2 - s_{2,2}s_{3,3}}}{s_{3,3}}\right), \tag{8}
$$

with $s_{i,j} = \langle \mathbf{t}, \mathbf{T}_i \odot \mathbf{T}_j \rangle$. This closely relates to the bounds computed by Hahlbohm *et al*. [1], but allows for the analysis and bounding of angles before transforming them to the screen, instead of directly receiving screen bounds.

C. Multi-Resolution Evaluation Images

We show an example view of our multi-resolution evaluation in Fig. 1. While Ours is able to retain good image quality at all resolution levels, and correctly dilates and smooths content. In contrast, MCMC and our method without the anti-aliasing 3D smoothing filter exhibit considerable aliasing: content becomes too thick on lower resolution and too thin on higher resolution. This also shows in the inset PSNR values.

D. Per Scene Image Metrics

We provide per-scene image metrics for all our evaluated scenes in Tab. 1.

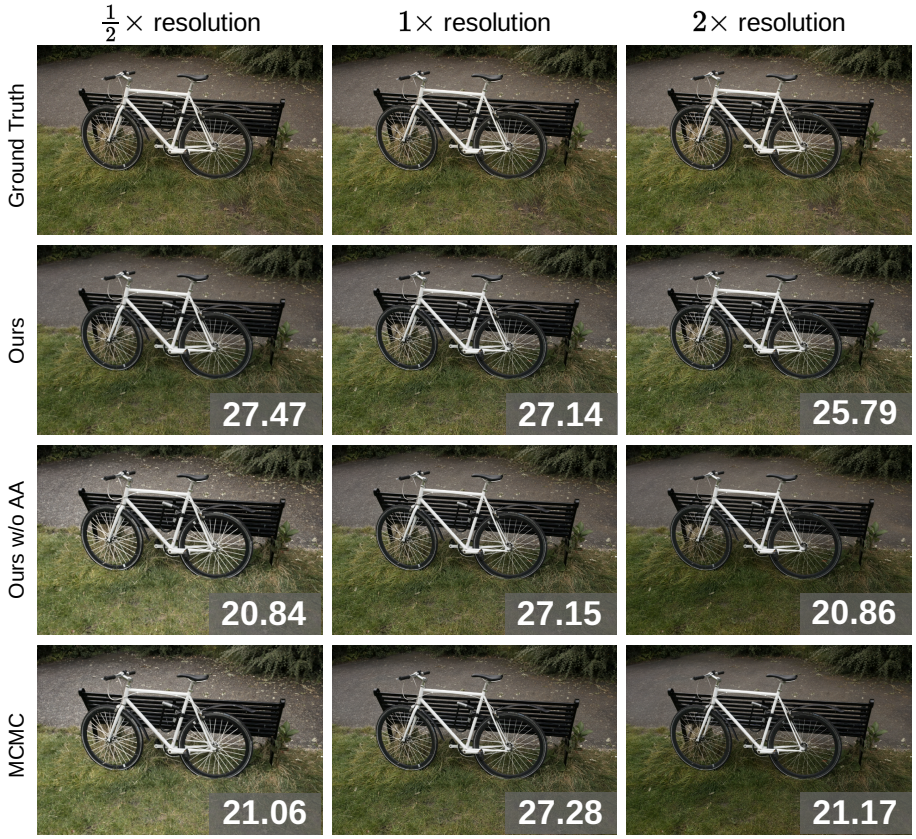

Figure 1. A single view of our multi-resolution evaluation on the Mip-Nerf 360 bicycle scene, with inset PSNR values.

Table 1. Per-scene image metrics for all methods on all evaluated scenes.

Dataset	Mip-NeRF 360 Outdoor					Mip-NeRF 360 Indoor				Deep Blending		Tanks & Temples	
Scene	Bicycle	Flowers	Garden	Stump	Treehill	Bonsai	Counter	Kitchen	Room	DrJ	Playroom	Train	Truck
	PSNR$^{\uparrow}$												
3DGS	25.19	21.53	27.30	26.62	22.46	32.11	28.97	31.33	31.48	29.05	29.97	22.05	25.41
StopThePop	25.22	21.54	27.23	26.70	22.44	31.98	28.60	31.18	30.84	29.45	30.40	21.49	24.96
Mip-Splatting	25.32	21.64	27.48	26.58	22.58	32.13	29.00	31.34	31.78	29.15	30.17	22.16	25.48
MCMC	25.69	22.01	27.87	27.36	22.94	32.65	29.38	32.09	32.25	29.52	29.93	22.83	26.45
Ours	25.74	22.13	27.50	27.24	23.03	32.32	29.24	31.88	31.45	29.90	31.07	21.73	25.43
Taming 3DGS	25.47	21.87	27.76	27.05	22.91	32.47	29.06	31.76	32.09	29.51	30.24	22.25	25.88
Hybrid Transparency	25.31	21.35	27.22	26.85	22.37	31.55	28.40	31.02	30.45	-	-	-	-
	SSIM$^{\uparrow}$												
3DGS	0.764	0.605	0.864	0.772	0.633	0.941	0.907	0.926	0.919	0.900	0.905	0.814	0.880
StopThePop	0.768	0.605	0.864	0.776	0.635	0.941	0.905	0.926	0.918	0.906	0.910	0.810	0.882
Mip-Splatting	0.768	0.608	0.869	0.773	0.638	0.942	0.909	0.928	0.920	0.902	0.909	0.818	0.886
MCMC	0.799	0.645	0.878	0.811	0.659	0.948	0.917	0.934	0.930	0.904	0.908	0.843	0.900
Ours	0.801	0.648	0.876	0.813	0.662	0.948	0.917	0.934	0.928	0.910	0.916	0.833	0.900
Taming 3DGS	0.780	0.614	0.873	0.788	0.646	0.944	0.910	0.931	0.924	0.909	0.911	0.819	0.892
Hybrid Transparency	0.785	0.631	0.867	0.793	0.639	0.941	0.902	0.923	0.920	-	-	-	-
	LPIPS$^{\downarrow}$												
3DGS	0.210	0.335	0.107	0.214	0.326	0.200	0.198	0.125	0.216	0.240	0.234	0.205	0.144
StopThePop	0.204	0.332	0.106	0.208	0.316	0.199	0.197	0.125	0.214	0.231	0.231	0.202	0.140
Mip-Splatting	0.212	0.339	0.108	0.216	0.326	0.204	0.200	0.126	0.218	0.243	0.243	0.205	0.147
MCMC	0.168	0.284	0.094	0.171	0.272	0.191	0.185	0.121	0.198	0.234	0.233	0.183	0.112
Ours	0.171	0.285	0.099	0.171	0.275	0.189	0.183	0.121	0.197	0.223	0.220	0.184	0.106
Taming 3DGS	0.192	0.332	0.100	0.196	0.313	0.201	0.198	0.122	0.210	0.234	0.235	0.208	0.128
Hybrid Transparency	0.178	0.282	0.106	0.193	0.275	0.189	0.197	0.128	0.205	-	-	-	-

References

[1] Florian Hahlbohm, Fabian Friederichs, Tim Weyrich, Linus Franke, Moritz Kappel, Susana Castillo, Marc Stamminger, Martin Eisemann, and Marcus Magnor. Efficient Perspective-Correct 3D Gaussian Splatting Using Hybrid Transparency, 2024. 2

[2] Christian Sigg, Tim Weyrich, Mario Botsch, and Markus Gross. GPU-Based Ray-Casting of Quadratic Surfaces. In *Symposium on Point-Based Graphics*. The Eurographics Association, 2006. 1