Speedy-Splat: Fast 3D Gaussian Splatting with Sparse Pixels and Sparse Primitives

Alex Hanson, Allen Tu, Geng Lin, Vasu Singla, Matthias Zwicker, Tom Goldstein

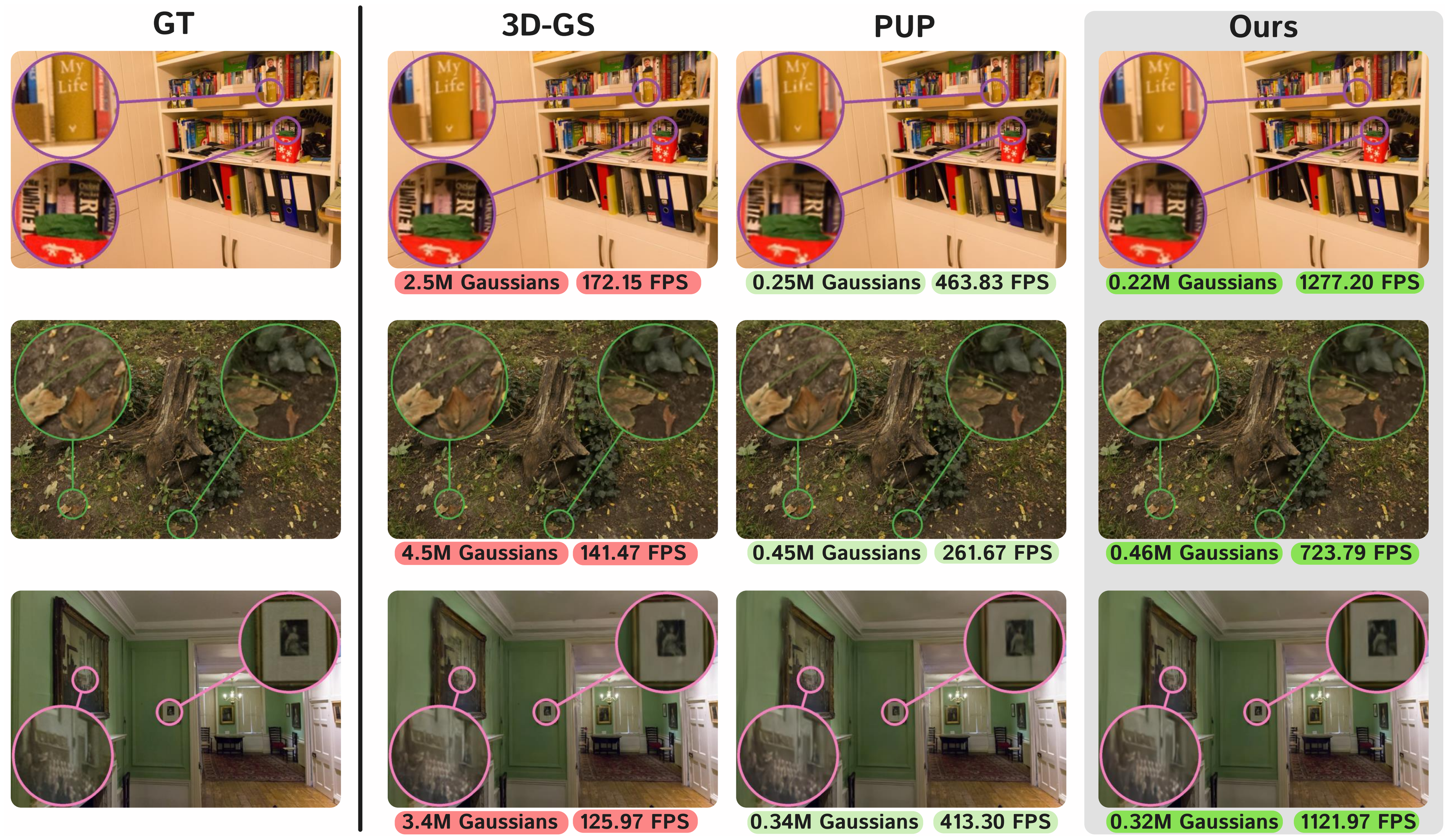
speedysplat.github.io

Background and Motivation

How can we accelerate the rendering speed of 3D Gaussian Splatting (3D-GS) by over 6 ×?

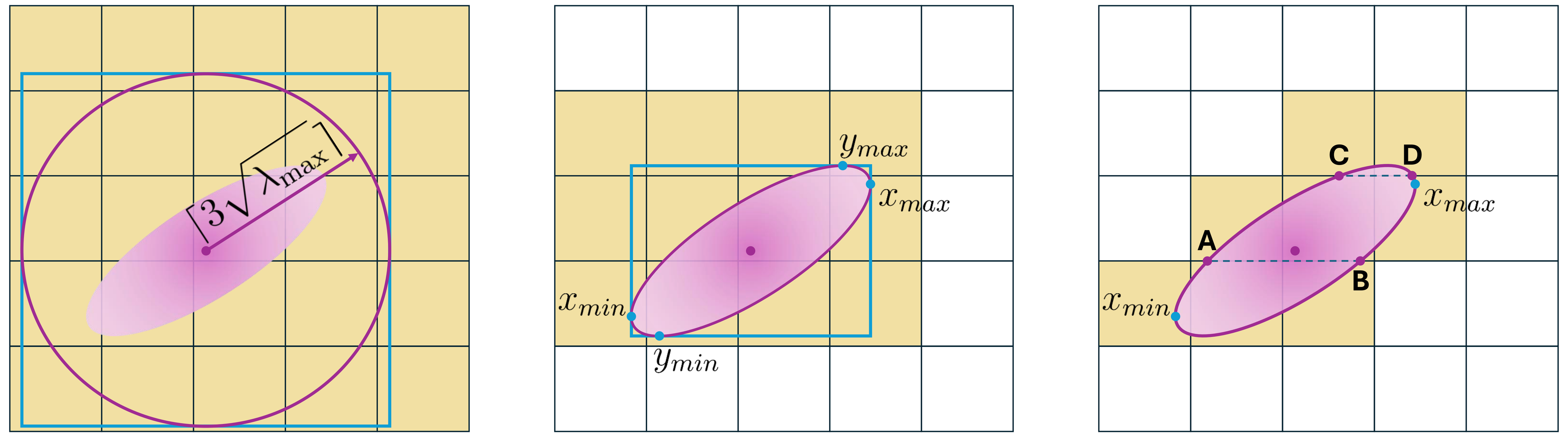
Rendering speed is primarily determined by two factors:

1. The **number of Gaussians allocated to each pixel**, and
2. The **total number of Gaussians** in the scene.



Method

Localization



(a) 3D Gaussian Splatting (b) SnugBox (c) AccuTile

Our localization algorithms reduce the number of Gaussians per pixel.

(a) 3D Gaussian Splatting overestimates Gaussian-to-tile intersections.
(b) Our **SnugBox** method finds the axis-aligned tight bounding box of the Gaussian and corresponding rectangular tile extent in constant time.
(c) Our **AccuTile** method extends SnugBox to quickly compute exact Gaussian-to-tile intersections.

SnugBox and AccuTile are **lossless** – they do not change the rendered image.

Pruning

Our pruning method reduces the total number of Gaussians by ~90%.

We compute a pruning score $\tilde{U}_i$ for each Gaussian $\mathcal{G}_i$ as a second order approximation of the L_2 reconstruction error:

$$\tilde{U}_i = \sum_{\phi \in \mathcal{P}_{gt}} \left(\nabla_{g_i} I_{\mathcal{G}}(\phi)\right)^2 \approx \sum_{\phi \in \mathcal{P}_{gt}} \nabla_{\mathcal{G}_i} I_{\mathcal{G}}(\phi) \nabla_{\mathcal{G}_i} I_{\mathcal{G}}(\phi)^T \approx \nabla^2_{\mathcal{G}_i} L_2,$$

where $\mathcal{P}_{gt}$ is the set of all training poses, $I_{\mathcal{G}}(\phi)$ is the rendered view for pose ϕ, and g_i is the value of the projected Gaussian in $I_{\mathcal{G}}(\phi)$.

We use this score to prune the scene during training via two modalities:

1. **Soft Pruning,** performed during the densification stage, and
2. **Hard Pruning,** performed after the densification stage.

Results

When compared to original 3D-GS, **Speedy-Splat** achieves
6.5× FPS, 10× compression, and 45% faster training.

Our **lossless** methods **boost FPS by 2×** for free.

Method	FPS↑	Comp.↑	Train↑	PSNR↑	SSIM↑	LPIPS↓
3D-GS	*1.00×*	*1.00×*	*1.00×*	27.55	0.814	0.222
EAGLES	1.51×	3.68×	1.37×	26.94	0.800	0.250
ELMGS	2.69×	5.00×	-	27.00	0.779	0.286
PUP	2.55×	8.65×	-	26.83	0.792	0.268
Mini-Splat	3.20×	6.84×	1.26×	27.34	0.822	0.217
Ours (Lossless)	1.99×	0.99×	1.10×	27.57	0.814	0.221
Ours (Full)	6.51×	10.6×	1.45×	26.94	0.782	0.296

Acknowledgements

This work was made possible by the IARPA WRIVA Program, the ONR MURI program, and DAPRA TIAMAT. Commercial support was provided by Capital One Bank, the Amazon Research Award program, and Open Philanthropy. Further support was provided by the National Science Foundation (IIS-2212182), and by the NSF TRAILS Institute (2229885). Zwicker was additionally supported by the National Science Foundation (IIS-2126407).