3D Gaussian Splatting for Real-Time Radiance Field Rendering

SIGGRAPH 2023 Best Paper Award

Presenter: Shaofan Sun
2024.7.8

Directory

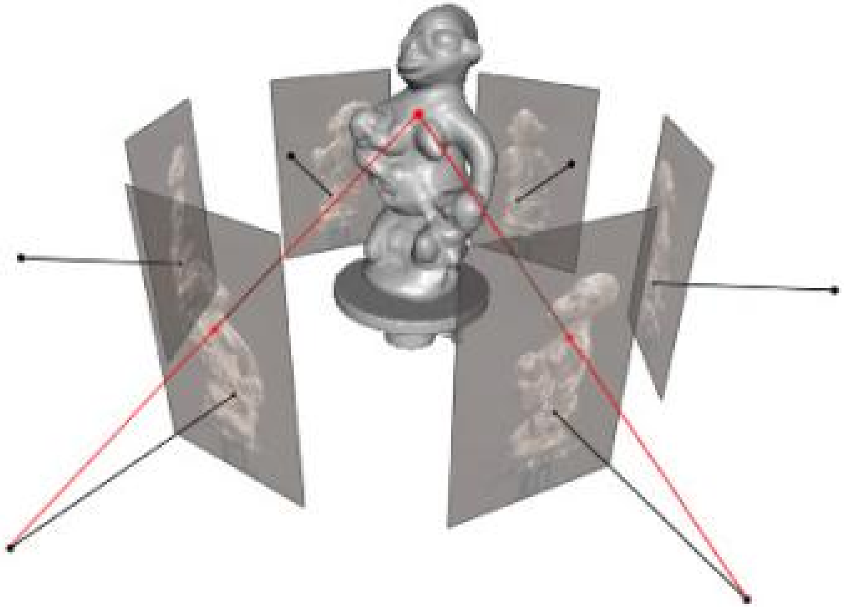
北京大学 PEKING UNIVERSITY

Background: SfM & MVS

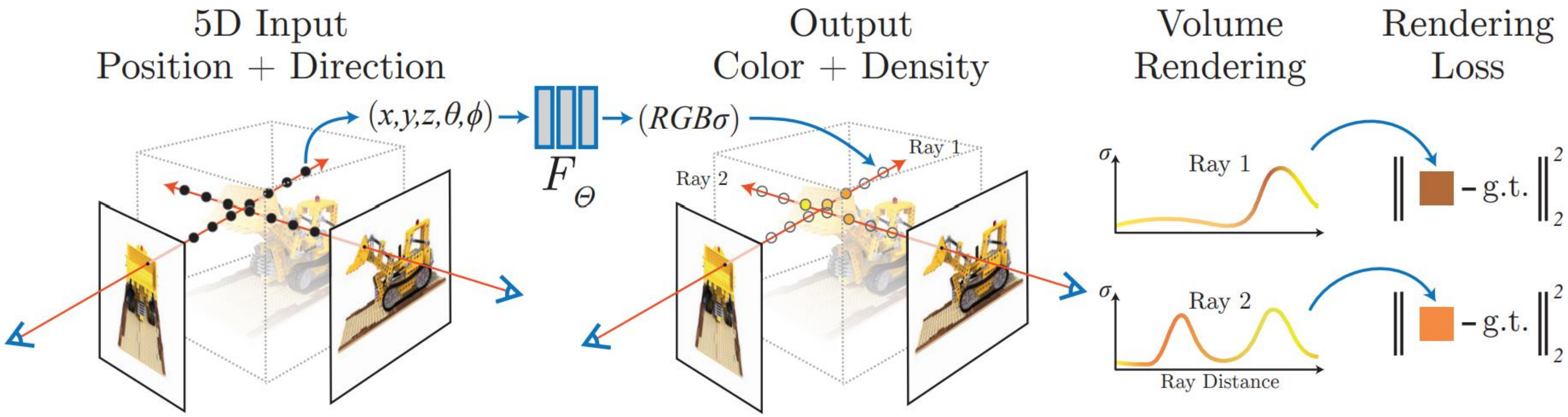
北京大學
PEKING UNIVERSITY

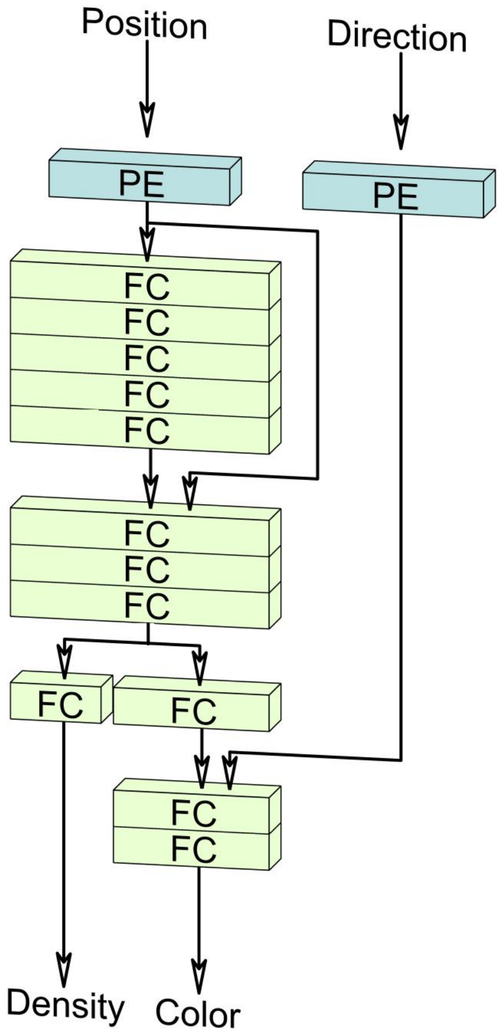
- Structure-from-Motion (SfM)
  - sparse reconstruction
  - estimate a sparse point cloud during camera calibration

- Multi-View Stereo (MVS)
  - dense reconstruction
  - estimate pixel-level information after matching images



Background: Neural Radiance Field (NeRF)





$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt\,, \text{ where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$$

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i\,, \text{ where } T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j\delta_j\right)$$

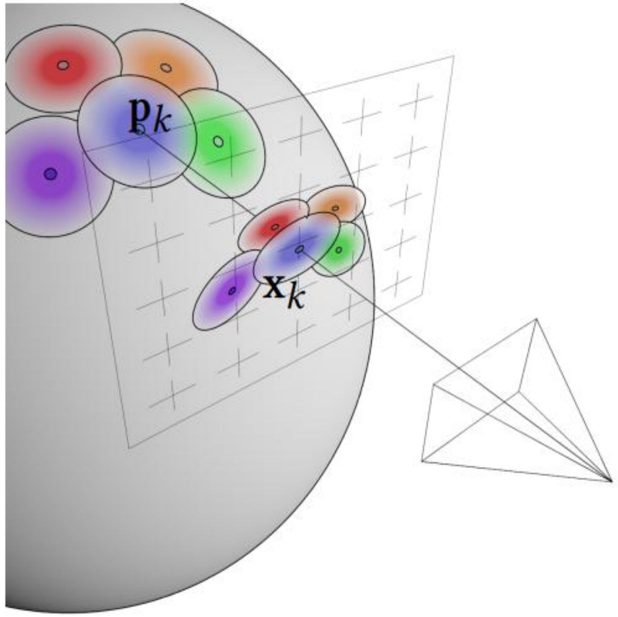
NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, Ben Mildenhall, et al., ECCV 2020

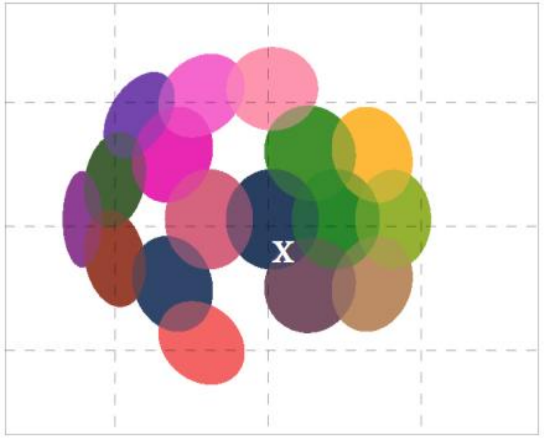
Background: EWA Point Splatting

- Points: elliptical Gaussians

- Pixel value: normalized sum

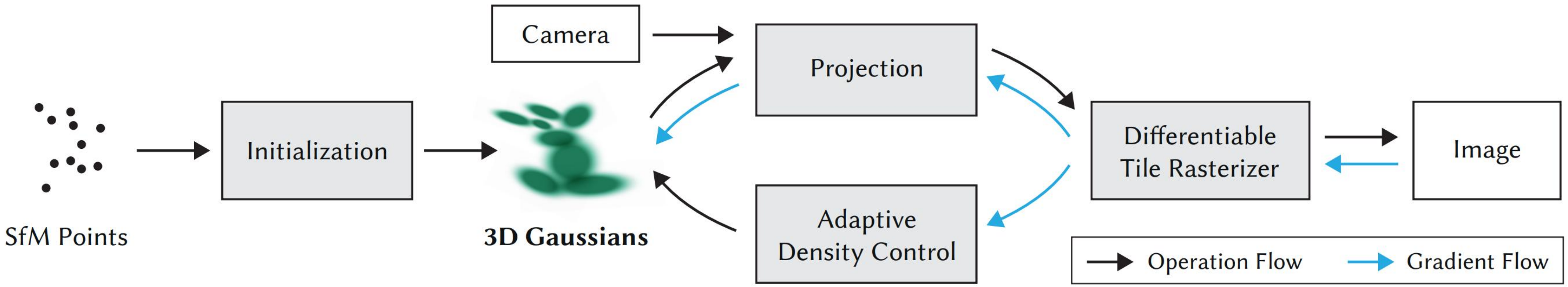
$$\mathbb{I}_{\mathbf{x}} = \frac{\sum_{k=0}^{N-1} \rho_k(\mathbf{x})\ \mathbf{w}_k}{\sum_{k=0}^{N-1} \rho_k(\mathbf{x})}$$





Differentiable surface splatting for point-based geometry processing, YiFan Wang, et al., ACM TOG 38, 6 (2019)

Method: Overview



- Create 3D Gaussians from sparse point cloud produced by SfM
- Create the radiance field representation via a sequence of optimization of 3D Gaussian parameters
- Allow α -blending of anisotropic splats with a tile-based rasterizer

Method: 3D Gaussian

北京大學 PEKING UNIVERSITY

- Model the geometry as a set of 3D Gaussians

$$G(x) = e^{-\frac{1}{2}(x)^T\Sigma^{-1}(x)}$$

- Project 3D Gaussians to 2D for rendering

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

  - W: transformation from object coordinates to camera coordinates
  - J: Jacobian of the affine approximation of the projective transformation

Method: 3D Gaussian

- Gaussian parameters to optimize
  - Positions p (mean)
  - Opacity α for α-blending of anisotropic splats
  - Covariance matrix Σ
  - Color c represented by Spherical Harmonics (SH) coefficients
- Loss function
  - Compare the resulting image to the training views
  - $\mathcal{L} = (1 - \lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{\text{D-SSIM}}$

Method: 3D Gaussian

- Optimize the covariance matrix Σ, but:
  - Σ has physical meaning only when it is positive semi-definite
  - Gradient descent for all parameters can create invalid matrices

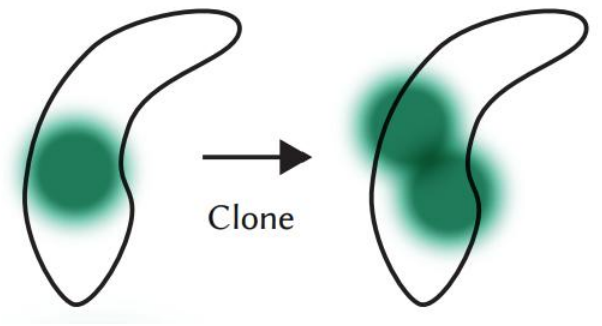
- Decompose Σ: $\Sigma = RSS^T R^T$
  - R: rotation matrix represented by a quaternion q
  - S: scaling matrix represented by a 3D vector s
  - Independently optimize both the factors

Method: Adaptive Control of Gaussians

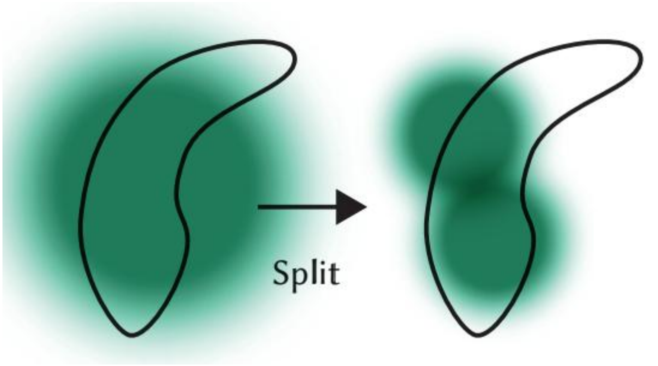
- Control the number and density (this “density” is not the σ in NeRF)
  - Focus on “under-reconstruction” and “over-reconstruction” regions
  - Densify and remove transparent Gaussians
- Set α close to zero periodically
- Remove “large” Gaussians

Method: Adaptive Control of Gaussians

- For under-reconstruction regions
  - Create a copy of the same size
  - Move in the direction of the positional gradient



- For over-reconstruction regions
  - split into two smaller Gaussians
  - Initialize position with original Gaussian as PDF



Method: Tile-based Rasterizer

- Pre-sort primitives instead of sorting per pixel
  - Split the screen into 16×16 tiles
  - Keep Gaussians with a 99% confidence interval intersecting the view frustum
  - Reject Gaussians at extreme positions
  - Instantiate each Gaussian according to the number of tiles they overlap
  - Assign each instance a key that combines view space depth and tile ID
  - sort Gaussians with fast GPU Radix sort

Method: Tile-based Rasterizer

- α -blending forward process:
  - Produce a list of sorted Gaussian instances for each tile
  - Accumulate color and α values front-to-back for each pixel
  - Stop when all pixels reach a target saturation of α
- Backward process:
  - Traverse the lists back-to-front
  - Each point stores the final accumulated opacity
  - Divide by each point's α to obtain the coefficients for gradients

Method: Detailed Implementation

北京大学 PEKING UNIVERSITY

- Use SGD for optimization
- Add custom CUDA kernels for some operations
- Sigmoid activation function for α
- Exponential activation function for the scale of the covariance
- “warm up” from lower resolution
- Optimize SH coefficients starting from zero-order component

Experiments: Results

- Real-world Scenes

Dataset	Mip-NeRF360					
Method|Metric	*SSIM*↑	*PSNR*↑	*LPIPS*↓	Train	FPS	Mem
Plenoxels	0.626	23.08	0.463	25m49s	6.79	2.1GB
INGP-Base	0.671	25.30	0.371	5m37s	11.7	13MB
INGP-Big	0.699	25.59	0.331	7m30s	9.43	48MB
M-NeRF360	0.792†	27.69†	0.237†	48h	0.06	8.6MB
Ours-7K	0.770	25.60	0.279	6m25s	160	523MB
Ours-30K	0.815	27.21	0.214	41m33s	134	734MB

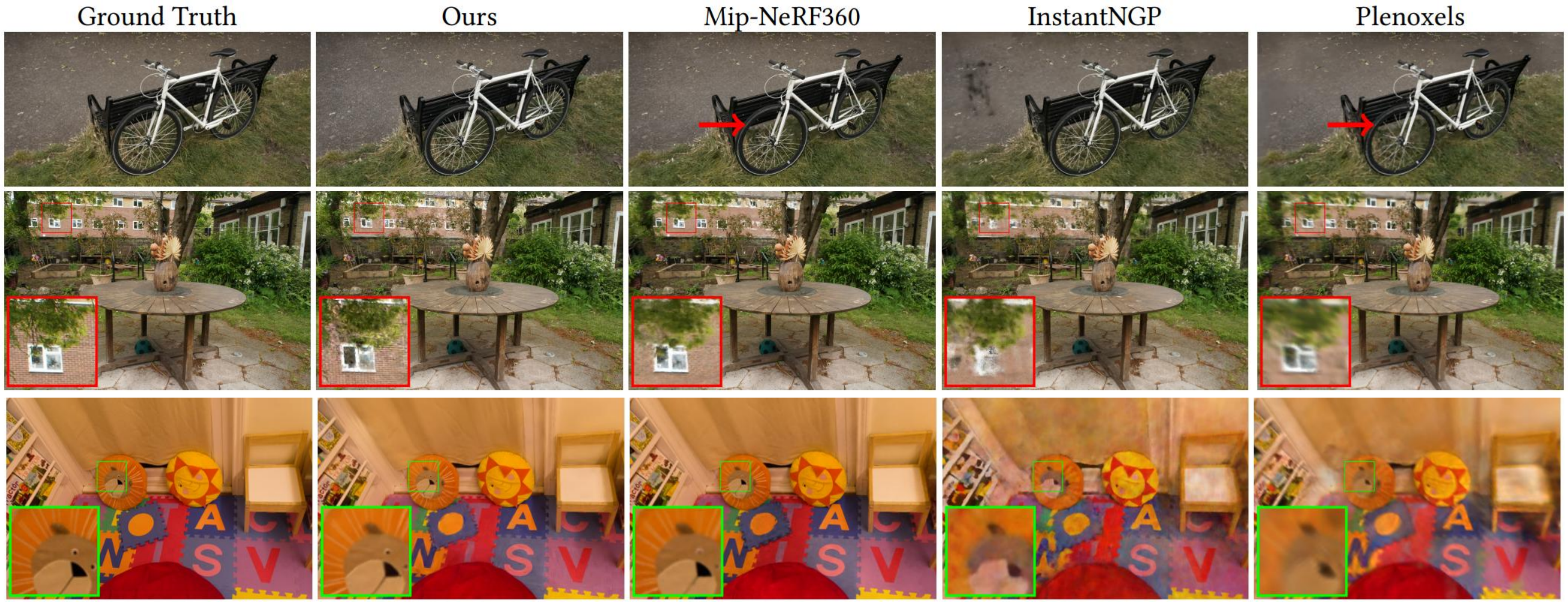
Experiments: Results

- Real-world Scenes

Tanks&Temples						Deep Blending					
SSIM↑	*PSNR*↑	*LPIPS*↓	Train	FPS	Mem	*SSIM*↑	*PSNR*↑	*LPIPS*↓	Train	FPS	Mem
0.719	21.08	0.379	25m5s	13.0	2.3GB	0.795	23.06	0.510	27m49s	11.2	2.7GB
0.723	21.72	0.330	5m26s	17.1	13MB	0.797	23.62	0.423	6m31s	3.26	13MB
0.745	21.92	0.305	6m59s	14.4	48MB	0.817	24.96	0.390	8m	2.79	48MB
0.759	22.22	0.257	48h	0.14	8.6MB	0.901	29.40	0.245	48h	0.09	8.6MB
0.767	21.20	0.280	6m55s	197	270MB	0.875	27.78	0.317	4m35s	172	386MB
0.841	23.14	0.183	26m54s	154	411MB	0.903	29.41	0.243	36m2s	137	676MB

Experiments: Results

- Real-world Scenes



Experiments: Results

北京大学 PEKING UNIVERSITY

- Synthetic Bounded Scenes

	Mic	Chair	Ship	Materials	Lego	Drums	Ficus	Hotdog	Avg.
Plenoxels	33.26	33.98	29.62	29.14	34.10	25.35	31.83	36.81	31.76
INGP-Base	36.22	35.00	31.10	29.78	36.39	26.02	33.51	37.40	33.18
Mip-NeRF	36.51	35.14	30.41	30.71	35.70	25.48	33.29	37.48	33.09
Point-NeRF	35.95	35.40	30.97	29.61	35.04	26.06	36.13	37.30	33.30
Ours-30K	35.36	35.83	30.80	30.00	35.78	26.15	34.87	37.72	33.32

PSNR Score

Experiments: Ablations

- Initialization from SfM

Experiments: Ablations

- Densification

- Split: better for background reconstruction

- Clone: better for thin structures

Experiments: Ablations

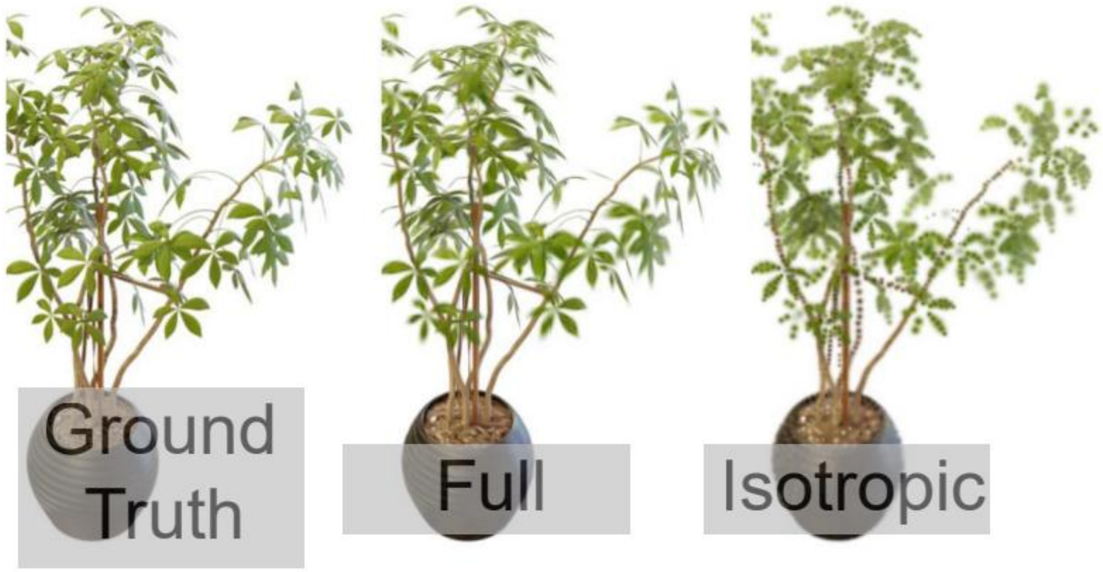
- Unlimited depth complexity of splats with gradients

limit 10 Gaussians

full version

Experiments: Ablations

- Anisotropic Covariance



Experiments: Ablations

北京大學 PEKING UNIVERSITY

- PSNR Scores

	Truck-5K	Garden-5K	Bicycle-5K	Truck-30K	Garden-30K	Bicycle-30K	Average-5K	Average-30K
Limited-BW	14.66	22.07	20.77	13.84	22.88	20.87	19.16	19.19
Random Init	16.75	20.90	19.86	18.02	22.19	21.05	19.17	20.42
No-Split	18.31	23.98	22.21	20.59	26.11	25.02	21.50	23.90
No-SH	22.36	25.22	22.88	24.39	26.59	25.08	23.48	25.35
No-Clone	22.29	25.61	22.15	24.82	27.47	25.46	23.35	25.91
Isotropic	22.40	25.49	22.81	23.89	27.00	24.81	23.56	25.23
Full	22.71	25.82	23.18	24.81	27.70	25.65	23.90	26.05

- More results: https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

Conclusion

- Contribution
  - Real-time, high-quality radiance field rendering
- Limitations
  - Artifacts
  - Memory consumption
- Future work
  - Culling approach, antialiasing and regularization
  - Adapt compression techniques for point clouds
  - Perform mesh reconstructions

Thanks for listening!