Lecture 25 - Apr 17

Sunday, March 10, 2024 8:43 PM

3D Gaussian Splatting

References

M. Zwicker, H. Pfister, J. van Baar, and M. Gross (2001). Surface splatting. *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*

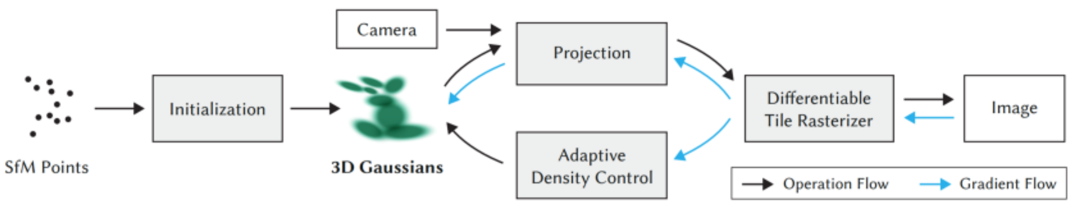
B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis (2023). 3D Gaussian Splatting for Real-time Radiance Field Rendering. *ACM Transactions on Graphics 42*(4).

K. Yurkova (2023). A Comprehensive Overview of Gaussian Splatting. https://towardsdatascience.com/a-comprehensive-overview-of-gaussian-splatting-e7d570081362

Slides

Tuesday, March 26, 2024 12:32 AM

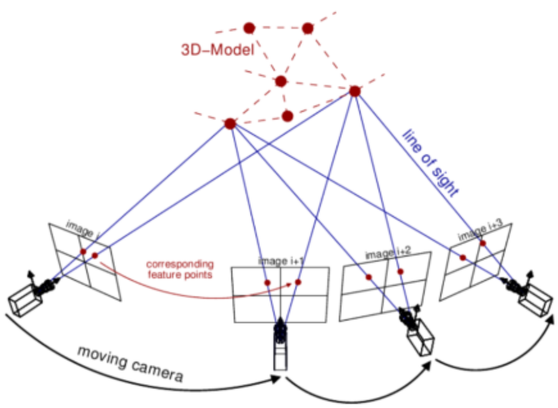
3D Gaussian Splatting (3DGS)



B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis (2023). 3D Gaussian Splatting for Real-time Radiance Field Rendering. *ACM Transactions on Graphics* 42(4).

Structure-from-Motion (SfM)

- We have multiple images of the scene from known camera positions
- Exploit the relative 2D pixel points in different view frames
- Reconstruct a 3D point cloud representation of the scene
  - COLMAP algorithm



J. Schonberger Lutz and J-M. Frahm (2016). Structure-from-Motion Revisited. *Conference on Computer Vision and Pattern Recognition (CVPR)*

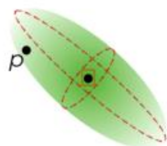
3D Gaussian Densities

- Points in the SfM point cloud are initialized as means of **3D Gaussian random variables**
- For each mean $\mu \in \mathbb{R}^3$ and covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$, the probability density is

$$f(p) \propto \exp\left(-\frac{1}{2}(p-\mu)^T\Sigma^{-1}(p-\mu)\right)$$

- Covariance matrix:

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_2^2 & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_3^2 \end{pmatrix}$$



J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan (2023). Dynamic 3D Gaussians: Tracking by Persistent Dynamic View Synthesis. *International Conference on 3D Vision.*

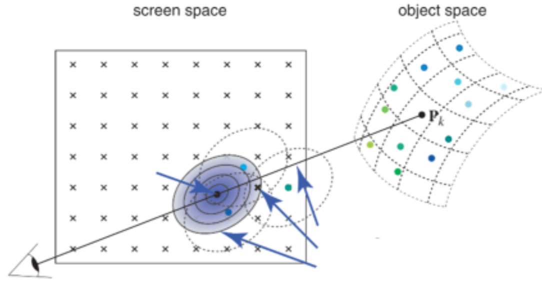
Projection

- We will project $f(p)$ onto the 2D camera frame we wish to render to get a "splat" $f^{2D}(p)$
- For intrinsic camera matrix K and extrinsic camera matrix W to get the 2D mean:

$$\begin{bmatrix}\mu^{2D} \\ 1\end{bmatrix} = \begin{bmatrix}\mu_x^{2D} \\ \mu_y^{2D} \\ 1\end{bmatrix} = z\begin{bmatrix}u/z \\ v/z \\ 1\end{bmatrix} = \begin{bmatrix}u \\ v \\ z\end{bmatrix} = KW\begin{bmatrix}\mu_x \\ \mu_y \\ \mu_z \\ 1\end{bmatrix}$$

- For a viewing transformation W, the covariance matrix is, using the the Jacobian, $J = \frac{\partial \mu^{2D}(\mu)}{\partial \mu}$:

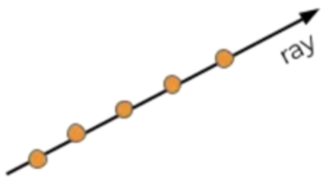
$$\Sigma^{2D} = JW\Sigma W^T J^T$$

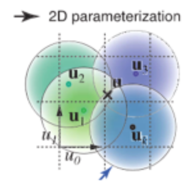


B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis (2023). 3D Gaussian Splatting for Real-time Radiance Field Rendering. *ACM Transactions on Graphics 42*(4).
M. Zwicker, H. Pfister, J. van Baar, and M. Gross (2001). Surface splatting. *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques.*

Rendering Pixel Colors by Ray Tracing

NeRF	3D Gaussian Splatting
$C(p) = c_i\left(1 - e^{-\sigma_i \delta_i}\right) \prod_{j=1}^{i-1} e^{-\sigma_j \delta_j}$	$C(p) = c_i\left(1 - f_i^{2D}(p)\right) \prod_{j=1}^{i-1} f_j^{2D}(p)$

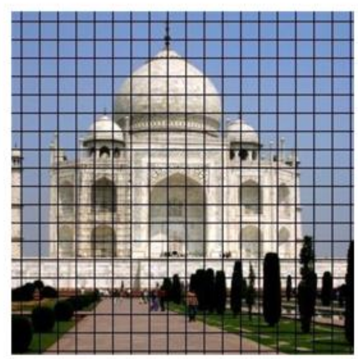




M. Zwicker, H. Pfister, J. van Baar, and M. Gross (2001). Surface splatting. *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques.*
K. Yurkova (2023). A Comprehensive Overview of Gaussian Splatting. https://towardsdatascience.com/a-comprehensive-overview-of-gaussian-splatting-e7d570081362

Tile-based Rasterizer

- Split the screen into 16x16 pixel tiles
- Project 3D Gaussians onto "splats" f^{2D} on the view frame
- Create a list of tiles each splat overlaps (if $f^{2D}(p) > 0.01$)
- GPU-accelerated radix sort: **per-tile ordered lists of splats**
- **Blend the splats back-to-front** with C(p) formula



Optimization

- **Initialization:** 3D Gaussians densities centered at SfM points with isotropic covariance (spherical with radius = mean distance to 3 nearest neighbors—no "holes" in the space)
- **Training:** Render views from ground-truth viewing angles, use stochastic gradient descent **(with some caveats)** to minimize the loss function **with respect to means and covariances**

$$L(\theta) = (1-\lambda)L_1(\theta) + \lambda L_{D-SSIM}(\theta)$$

  - All derivatives are computed explicitly resulting in fast gradient calculations

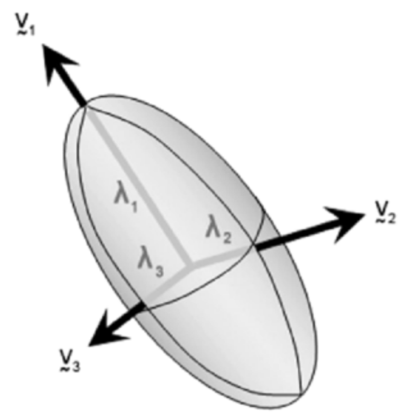
Covariance Decomposition and Optimization

- Covariance matrices must be **positive semi-definite**
- Optimizing their entries naively would violate this, so we use **eigendecomposition**

$$\Sigma = \mathrm{VLV}^{-1} = VSS^TV^T$$

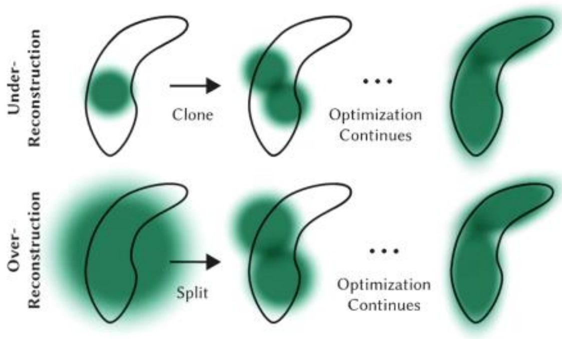
where V (orthogonal eigenvector matrix—**rotation**) and $S = \sqrt{L}$ (diagonal eigenvalue matrix—**scaling**)

- 3DGS **modifies V and S during optimization** to ensure covariance matrices are legitimate



G. Dougherty (2010). Image analysis in medical imaging: recent advances in selected examples. *Biomedical Imaging and Intervention Journal*, 6(3):e32.

Adaptive Density Control



Every 100 training iterations, add and remove some 3D Gaussian points:

- In regions missing geometric features, **clone "small" Gaussians** and move in the direction of $\nabla_{\mu} f(p)$
- In regions where Gaussians cover large spaces, **split "large" Gaussians** and divide Σ by $\phi = 1.6$
- **Remove Gaussians** with small $\alpha = 1 - f_i^{2D}(p)$, which are nearly transparent

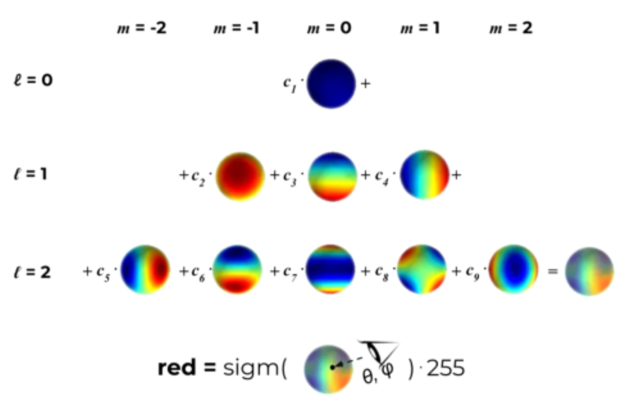
Spherical Harmonics for View-Dependent Colors

- Color **should be** dependent on viewing angle to represent reflections and lighting properly
- Spherical harmonics form an orthonormal basis of $L^2(S^2)$ and have an explicit structure:

$$Y_l^m(\theta,\phi) = (-1)^l \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}}\, e^{im\phi} P_l^m(\cos\theta)$$

  for $-l \le m \le l$

- Assume RGB are each linear combinations of the first l_{max} spherical harmonics



K. Yurkova (2023). A Comprehensive Overview of Gaussian Splatting. https://towardsdatascience.com/a-comprehensive-overview-of-gaussian-splatting-e7d570081362