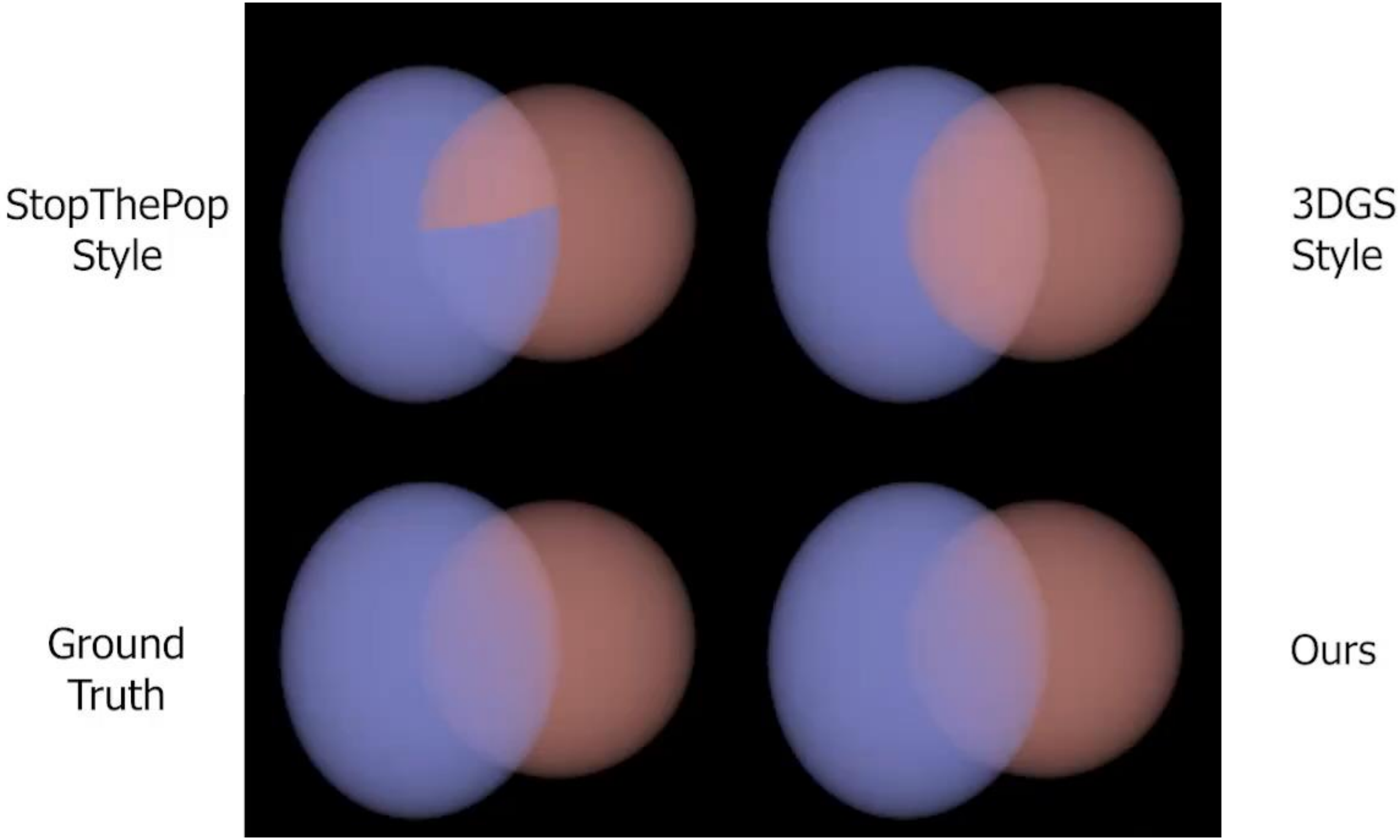
EVER: Exact Volumetric Ellipsoid Rendering for Real-time View Synthesis

汇报组:王之君 王俊逸 郎文翀



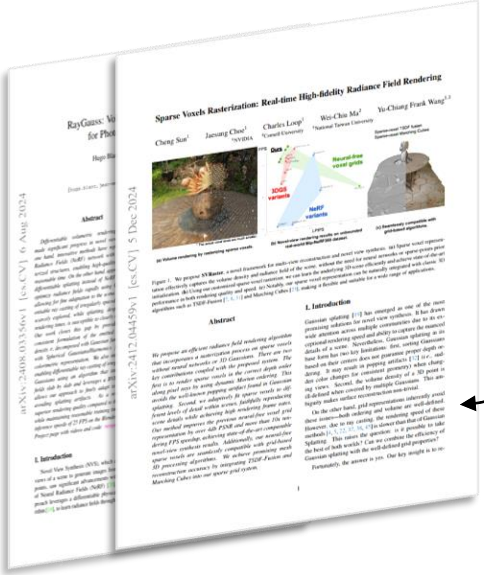
Content


NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

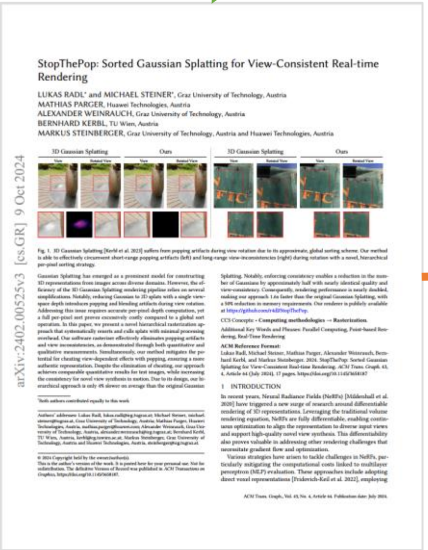
NeRF(20.08.03)


High-Resolution Image Synthesis with Latent Diffusion Models

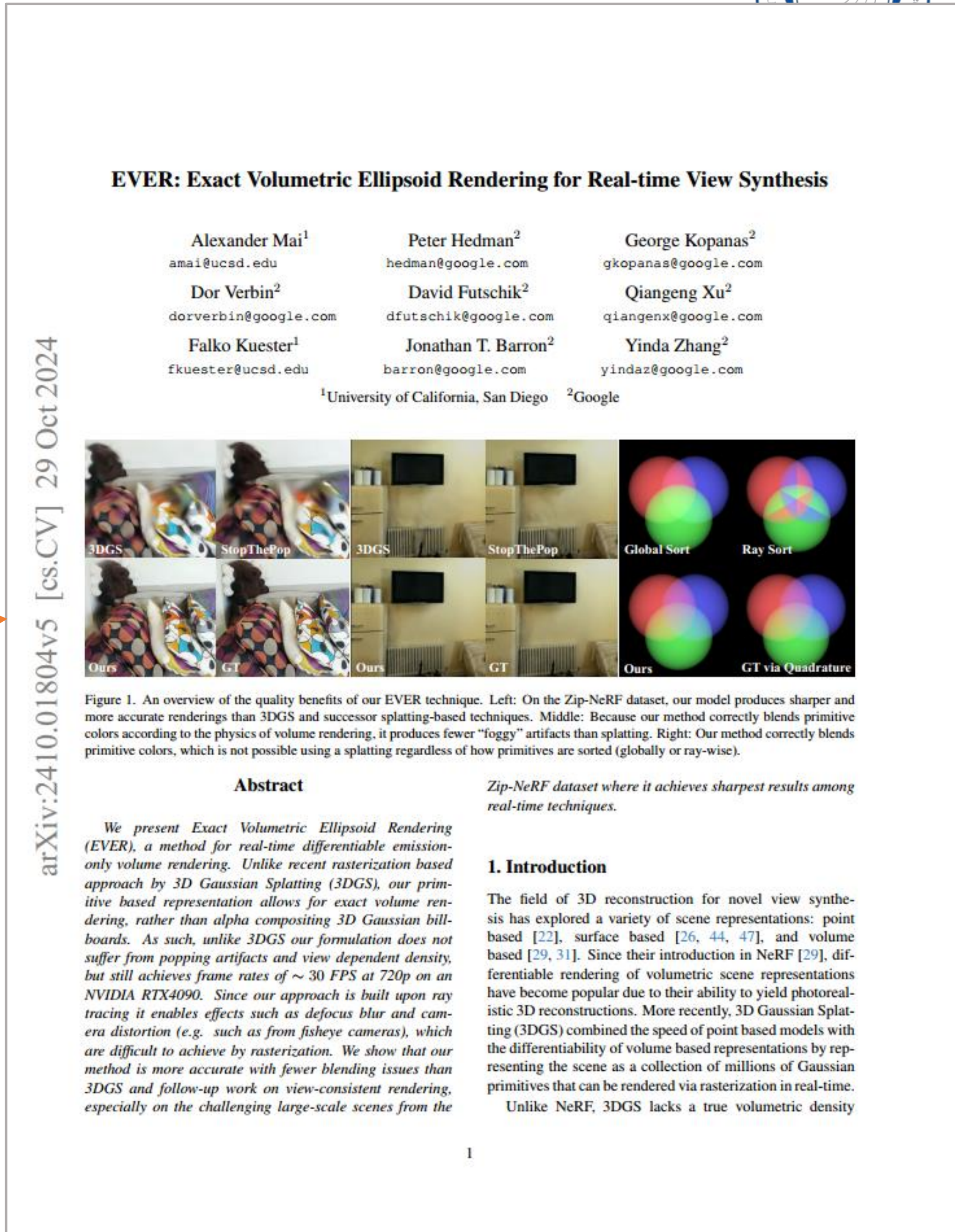
3D-GS(23.08.08)


Sparse Voxels Rasterization: Real-time High-fidelity Radiance Field Rendering

SVRaster/RayGS…(Now)


StopThePop: Sorted Gaussian Splatting for View-Consistent Real-time Rendering

Stop-The-Pop(24.02.01)


arXiv:2410.01804v5 [cs.CV] 29 Oct 2024

EVER: Exact Volumetric Ellipsoid Rendering for Real-time View Synthesis

Alexander Mai[1] amai@ucsd.edu
Peter Hedman[2] hedman@google.com
George Kopanas[2] gkopanas@google.com
Dor Verbin[2] dorverbin@google.com
David Futschik[2] dfutschik@google.com
Qiangeng Xu[2] qiangenx@google.com
Falko Kuester[1] fkuester@ucsd.edu
Jonathan T. Barron[2] barron@google.com
Yinda Zhang[2] yindaz@google.com

[1]University of California, San Diego [2]Google

Figure 1. An overview of the quality benefits of our EVER technique. Left: On the Zip-NeRF dataset, our model produces sharper and more accurate renderings than 3DGS and successor splatting-based techniques. Middle: Because our method correctly blends primitive colors according to the physics of volume rendering, it produces fewer "foggy" artifacts than splatting. Right: Our method correctly blends primitive colors, which is not possible using a splatting regardless of how primitives are sorted (globally or ray-wise).

Abstract

We present Exact Volumetric Ellipsoid Rendering (EVER), a method for real-time differentiable emission-only volume rendering. Unlike recent rasterization based approach by 3D Gaussian Splatting (3DGS), our primitive based representation allows for exact volume rendering, rather than alpha compositing 3D Gaussian billboards. As such, unlike 3DGS our formulation does not suffer from popping artifacts and view dependent density, but still achieves frame rates of ~ 30 FPS at 720p on an NVIDIA RTX4090. Since our approach is built upon ray tracing it enables effects such as defocus blur and camera distortion (e.g. such as from fisheye cameras), which are difficult to achieve by rasterization. We show that our method is more accurate with fewer blending issues than 3DGS and follow-up work on view-consistent rendering, especially on the challenging large-scale scenes from the Zip-NeRF dataset where it achieves sharpest results among real-time techniques.

1. Introduction

The field of 3D reconstruction for novel view synthesis has explored a variety of scene representations: point based [22], surface based [26, 44, 47], and volume based [29, 31]. Since their introduction in NeRF [29], differentiable rendering of volumetric scene representations have become popular due to their ability to yield photorealistic 3D reconstructions. More recently, 3D Gaussian Splatting (3DGS) combined the speed of point based models with the differentiability of volume based representations by representing the scene as a collection of millions of Gaussian primitives that can be rendered via rasterization in real-time.

Unlike NeRF, 3DGS lacks a true volumetric density

1

References & Division

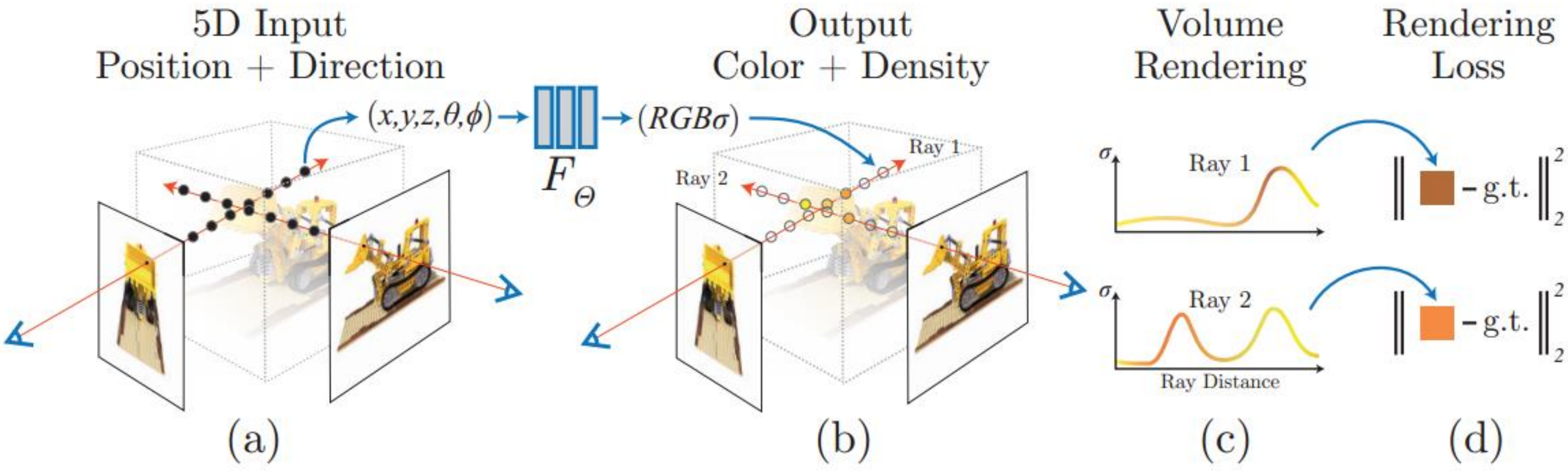
- **References**
  - 3D Representation
    - NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis
    - 3D Gaussian Splatting for Real-Time Radiance Field Rendering
    - PlenOctrees for Real-time Rendering of Neural Radiance Fields
  - View-Consistence
    - StopThePop: Sorted Gaussian Splatting for View-Consistent Real-time Rendering
    - EVER: Exact Volumetric Ellipsoid Rendering for Real-time View Synthesis
    - Sparse Voxels Rasterization: Real-time High-fidelity Radiance Field Rendering

Name	ID	Task
Wang Zhijun	SA23001118	Code Replication&Article Research
Wang Junyi	SA24010048	Code Replication&Article Research
Lang Wenchong	SA23229086	Code Replication& PPT

01 Preliminary : What is NeRF(Neural Radiance Field) and NVS?

- Image pixel-by-pixel inverse depth ray sampling query points are used to train the MLP network.
- The continuous radiation field MLP accepts 5 degrees of freedom input and output the color and density of each point.

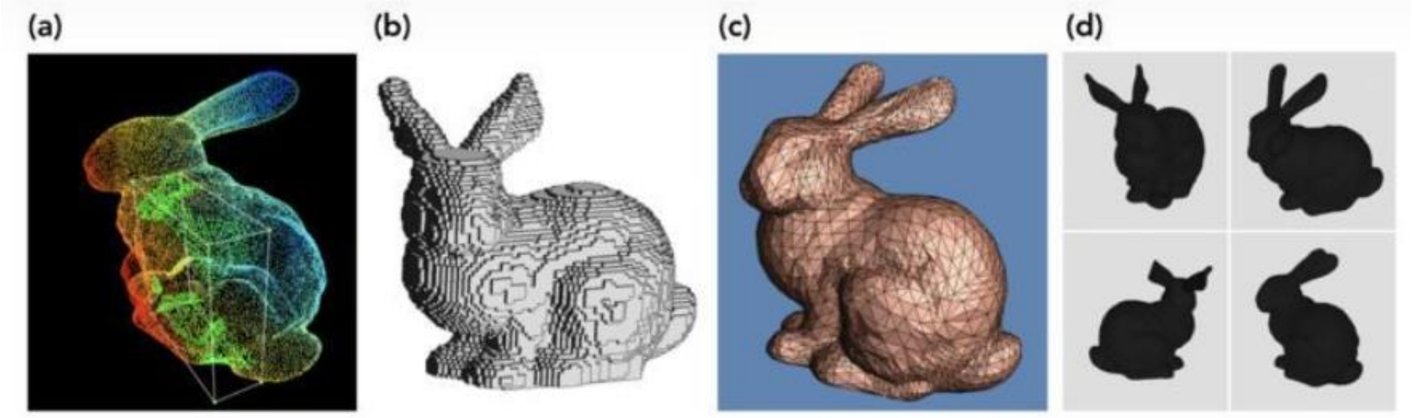
01 Preliminary : What is NeRF(Neural Radiance Field) and NVS?

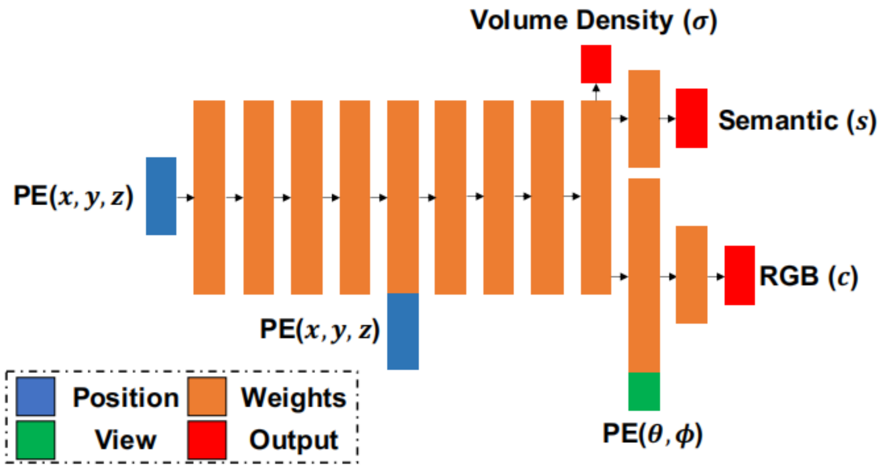


- Train Network
- 5DOF Input
- C,σ Output
- Volume Render

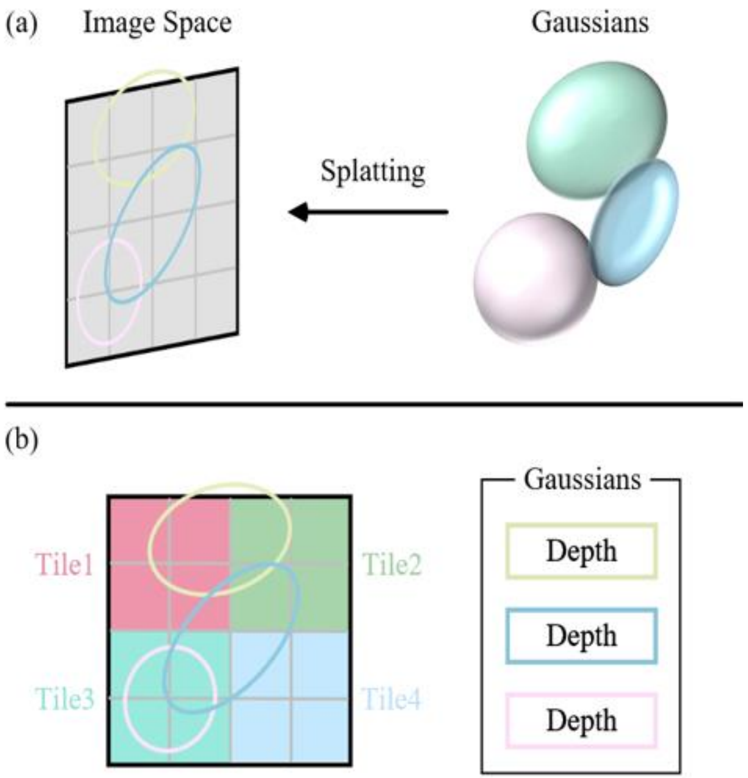
$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t),\mathbf{d})dt\,, \text{ where } T(t) = \exp\left(-\int_{t_n}^{t}\sigma(\mathbf{r}(s))ds\right). \quad (1)$$

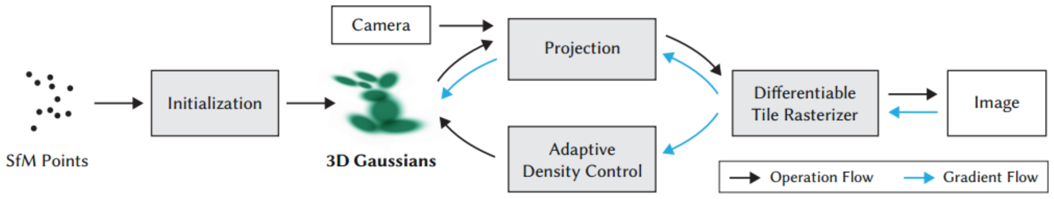
$$\mathcal{L} = \sum_{\mathbf{r}\in\mathcal{R}}\left[\left\|\hat{C}_c(\mathbf{r}) - C(\mathbf{r})\right\|_2^2 + \left\|\hat{C}_f(\mathbf{r}) - C(\mathbf{r})\right\|_2^2\right]$$





01 Premilinary: 3D Gaussian Splatting for Real-Time Radiance Field Rendering

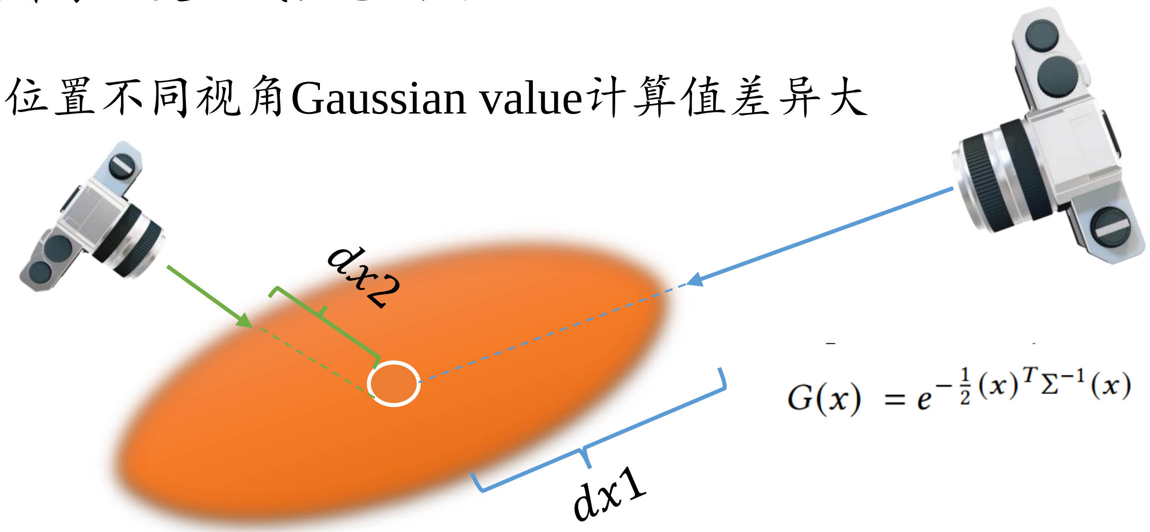


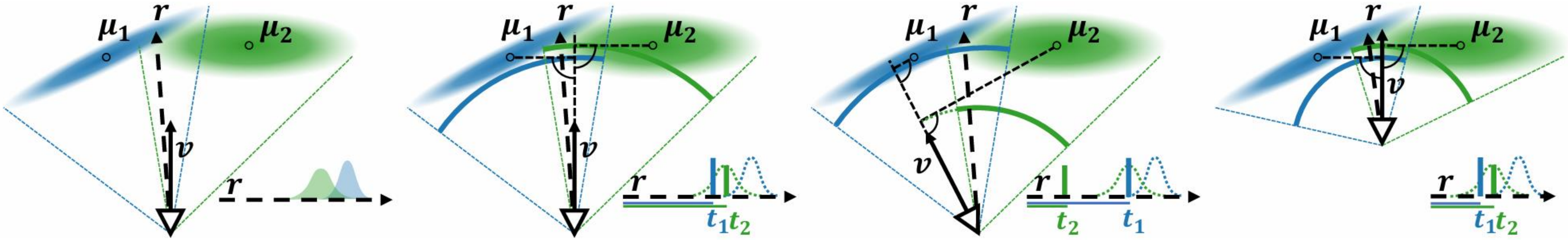


- Implicit vs Explicit , Volume Rendering vs Splatting
- Much more Fast(10x) in training and rendering!

02 Problem: View-Consistent Rendering Bad in 3D-GS

- 3D-GS视角不一致原因：
- 1.深度排序跳变造成颜色闪烁
- 2.相同位置不同视角Gaussian value计算值差异大





(a) **Ray Tracing排序** (b) **Splat方法排序** (c) **Splat视角旋转深度跳变** (d)

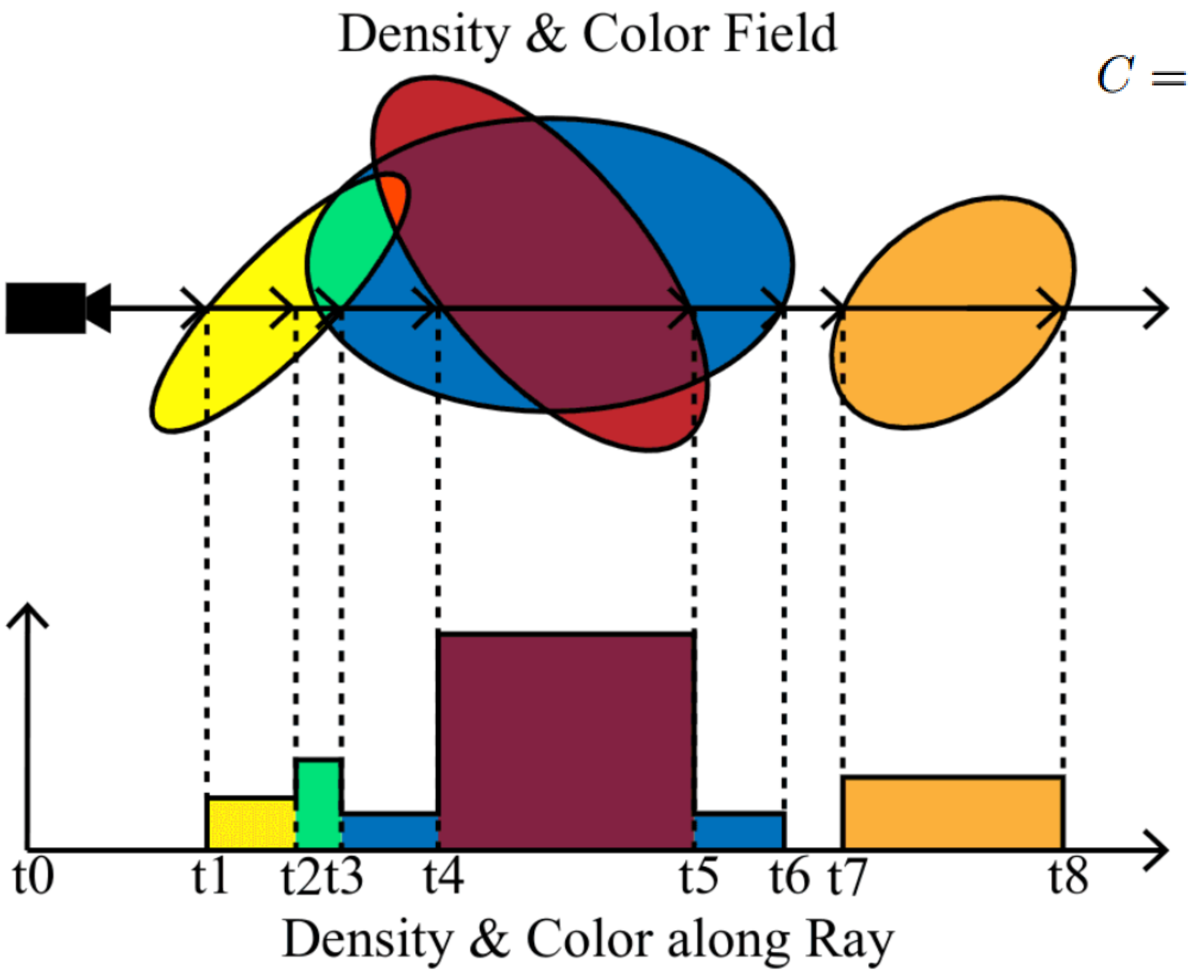
03 EVER: Exact Volumetric Ellipsoid Rendering for Real-time View Synthesis

- 3D-GS视角不一致对策：
- 1.深度排序替换为Ray Tracing求椭圆交点根据交点先后顺序排序
- 2.3D-GS(概率渐变椭球)替换为Density一致的椭球
- 3.颜色渲染公式Opacity*Gs替换为[t0,t1]区间内的Density积分

$$C = \sum_{i \in \mathcal{N}} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j),$$

$$C = \int_0^{\infty} c_{\mathbf{r}}(t) \sigma_{\mathbf{r}}(t) \exp\left(-\int_0^t \sigma_{\mathbf{r}}(s)\, ds\right) dt$$

03 EVER: Exact Volumetric Ellipsoid Rendering for Real-time View Synthesis



$$C = \sum_{i=1}^{N} \mathbf{c}_i (1 - \exp(-\sigma_i \Delta t_i)) \prod_{j=1}^{i-1} \exp(-\sigma_j \Delta t_j)$$

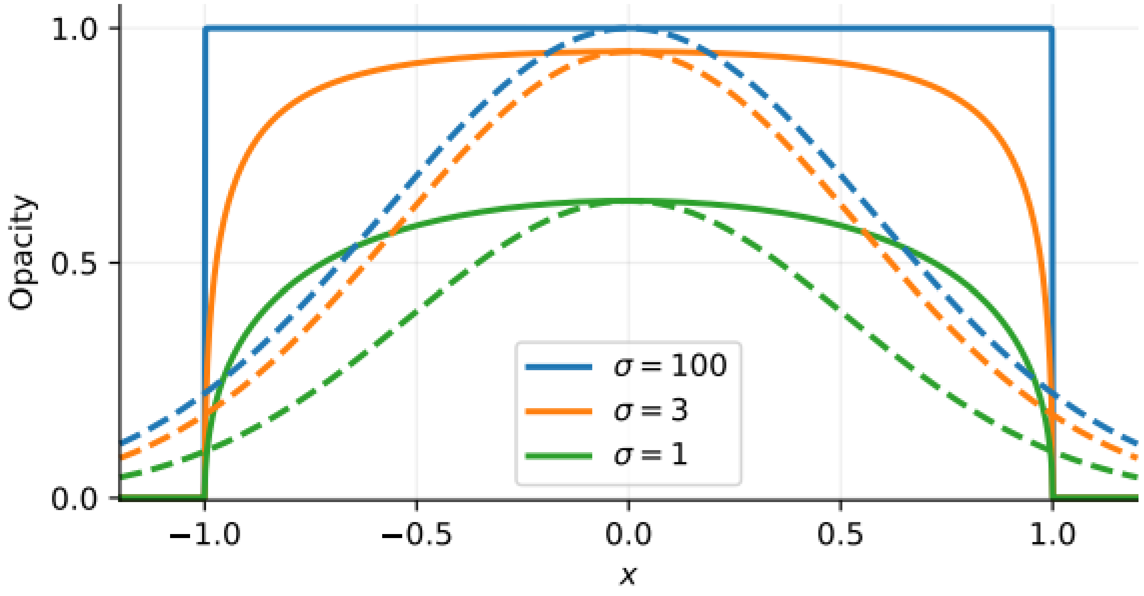
$$\sigma(t) = \sum_{i=1}^{N} \sigma_i \mathbb{1}[t_i \leq t < t_{i+1}],$$

$$\mathbf{c}(t) = \sum_{i=1}^{N} \mathbf{c}_i \mathbb{1}[t_i \leq t < t_{i+1}],$$

$$\sigma_i = \sum_{k=1}^{i} \Delta\sigma_k, \quad \mathbf{c}_i = \frac{1}{\sigma_i} \sum_{k=1}^{i} \Delta\sigma_k \Delta\mathbf{c}_k,$$

- 核心算法演示：
- 1.椭圆交点 2.获取每一个交点线段内的累积Density 3.NeRF风格体渲染公式计算像素颜色

03 EVER: Exact Volumetric Ellipsoid Rendering for Real-time View Synthesis



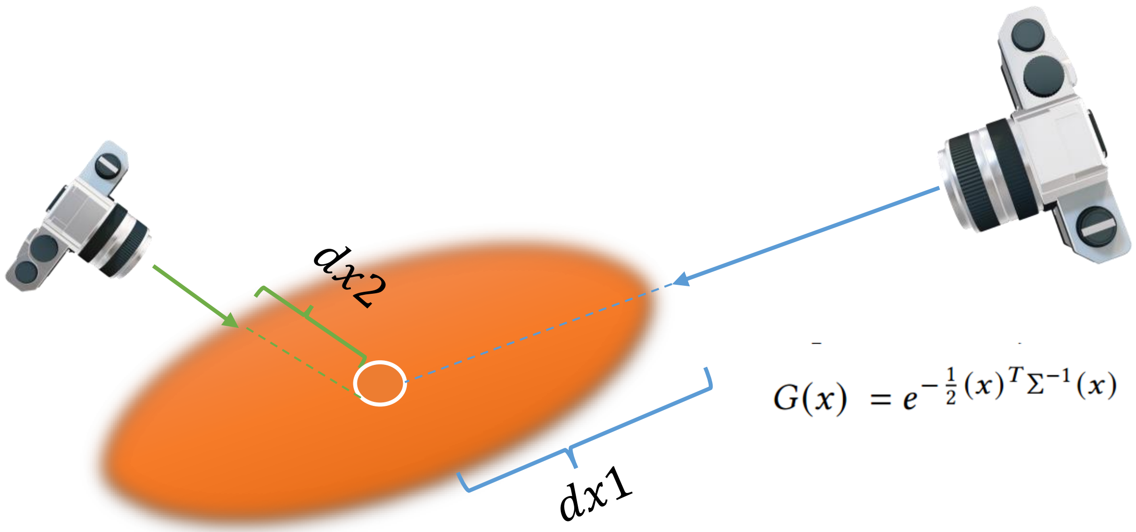
$$C = \sum_{i \in N} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j),$$

NeRF

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

3DGS

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



- Alpha计算格式不同
- EVER替换回NeRF Density Field天然解决了各向异性图元视角不一致性的问题
- Density σ，Alpha α， Opacity o关系？
- 思考:NeRF为什么没有3DGS的问题？

03 EVER: Exact Volumetric Ellipsoid Rendering for Real-time View Synthesis

	Mip-NeRF360				Zip-NeRF			
	PSNR ↑	SSIM ↑	LPIPS ↓	FPS ↑	PSNR ↑	SSIM ↑	LPIPS ↓	FPS ↑
3DGS [22]	27.48	.816	.257	224	25.84	.842	.418	559
StopThePop [34]	27.33	.816	.251	180	25.92	.819	.411	403
3DGRT [30]	27.20	.818	.248	78†	-	-	-	-
SMERF [15]	27.99	.818	.238	204†	27.28	.830	.389	217†
Our model	27.51	.825	.233	36	26.60	.845	.368	24
ZipNeRF [4]	28.54	.828	.219	0.5†	27.37	.836	.364	0.5†

- 注意：虽然图像性能指标提升不多，但是视觉效果提升明显，主要表现在大场景繁多高斯重叠情况

04 Experiment:

05 Thinking: Forward vs Backward Render / Implicit vs Explicit / 3DGS vs Ellipsoid

Project	Quality	Velocity	Primitives	Rendering Method	Representation
EVER	2	2	Ellipsoid	Backward	Explicit
3DGS	3	1	GS	Forward	Explicit
ZIPNeRF	1	3	NetWork	Backward	Implicit

- 显示表达与隐式表达主要取决于图元的表示是否显示
- 渲染方式决定渲染速度，Forward速度更快更直接，Backward排序更准确
- 3DGS与EVER图元并不相同，3DGS为类椭球概率分布，EVER为均匀密度椭球体，定义本质区别
- EVER结合3DGS显示表达与Backward准确排序方式结合两者优势，质量与速度兼并

06 More Step: Sparse Voxels Rasterization: Real-time High-fidelity Radiance Field Rendering

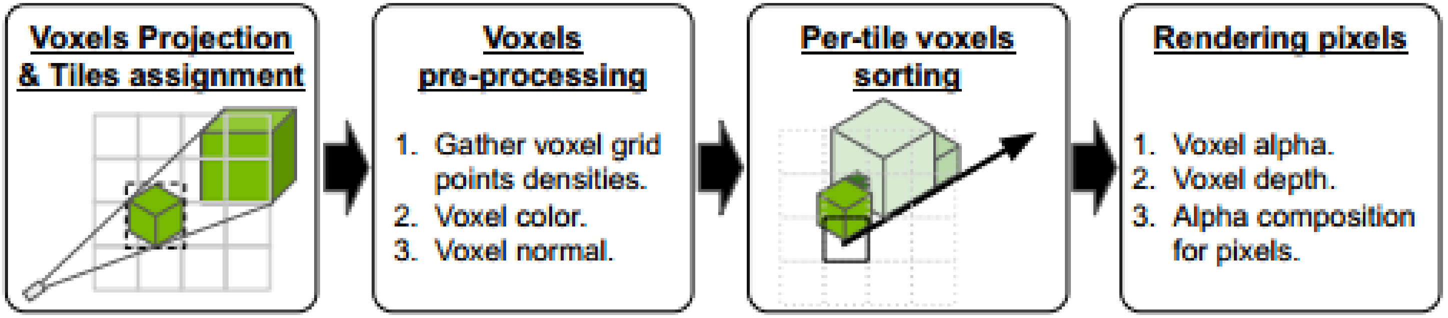
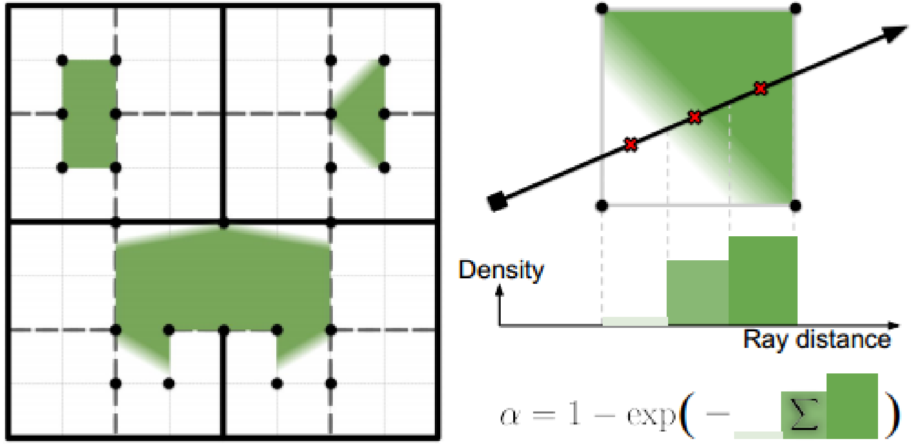
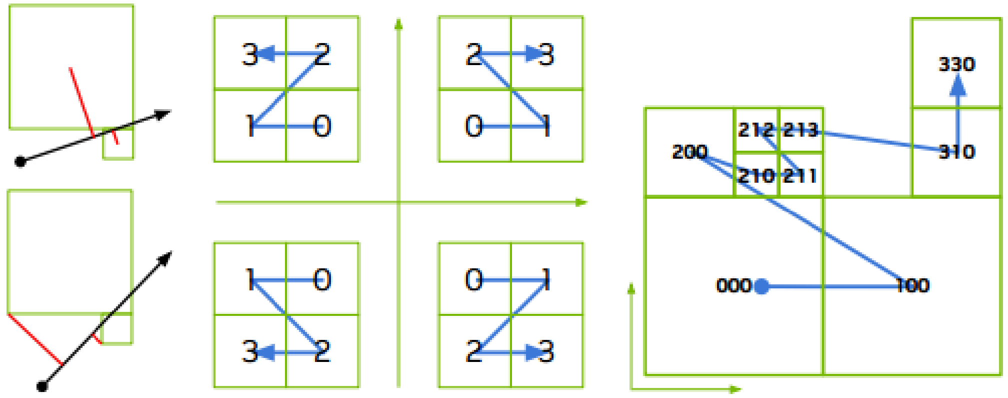


Figure 3. **Rasterization procedure.** Refer to Sec. 3.1.2 for details.

- 图元修改为规则体素网格
- Density Field借助插值
- 渲染方法Backward+体渲染
- 排序求交点AABB+Morton编码
- 优势：更加迅速，与Fusion完美结合，规则结构适用于树结构





Inspired By Plenoctrees

END & THANKS

汇报组:王之君，王俊逸，郎文翀 2024.12.30