论文阅读报告：《4D Gaussian Splatting for Real-Time Dynamic Scene Rendering》

1. 论文摘要

表示和渲染动态场景一直是一项重要但具有挑战性的任务。特别是，要准确地建模复杂运动，通常很难保证高效率。为了在实现实时动态场景渲染的同时，还能享有高训练和存储效率，作者提出了 4D 高斯溅射（4D - GS）作为动态场景的整体表示，而不是对每个单独的帧应用 3D - GS。在 4D - GS 中，提出了一种同时包含 3D 高斯和 4D 神经体素的新颖显式表示。提出了一种受 HexPlane 启发的分解神经体素编码算法，以有效地从 4D 神经体素构建高斯特征，然后应用一个轻量级的多层感知器（MLP）来预测新时间戳下的高斯变形。达到了和先前技术水平相当或更好的质量。

2. 论文介绍

2.1 核心观点

- 提出 4D 高斯溅射框架，将 3D 高斯与 4D 神经体素相结合，通过高斯变形场网络建模高斯运动和形状变化。
- 设计多分辨率编码方法，利用空间 - 时间结构编码器连接附近 3D 高斯，构建丰富特征。
- 实现动态场景实时渲染，在高分辨率下具有较高帧率，渲染质量与现有方法相当或更优。

2.2 研究背景

动态场景渲染在 VR、AR 和电影制作等领域具有重要应用，但准确建模复杂运动并保证高效性是一个挑战。NeRF 通过隐式函数表示场景取得成功，但训练和渲染成本高。3D 高斯溅射（3D-GS）通过将场景表示为 3D 高斯，显著提高了渲染速度，但在处理动态场景时面临存储和建模复杂运动的问题。

2.3 研究目的

构建紧凑表示，同时保持训练和渲染效率，实现实时动态场景渲染。

2.4 研究方法

提出 4D 高斯溅射框架，包含 3D 高斯和高斯变形场网络。通过空间 - 时间结构编码器编码 3D 高斯的时空特征，多分辨率 HexPlane 模块分解 4D 神经体素，利用多头高斯变形解码器预测 3D 高斯的变形。采用 L1 颜色损失和总变差损失进行优化。

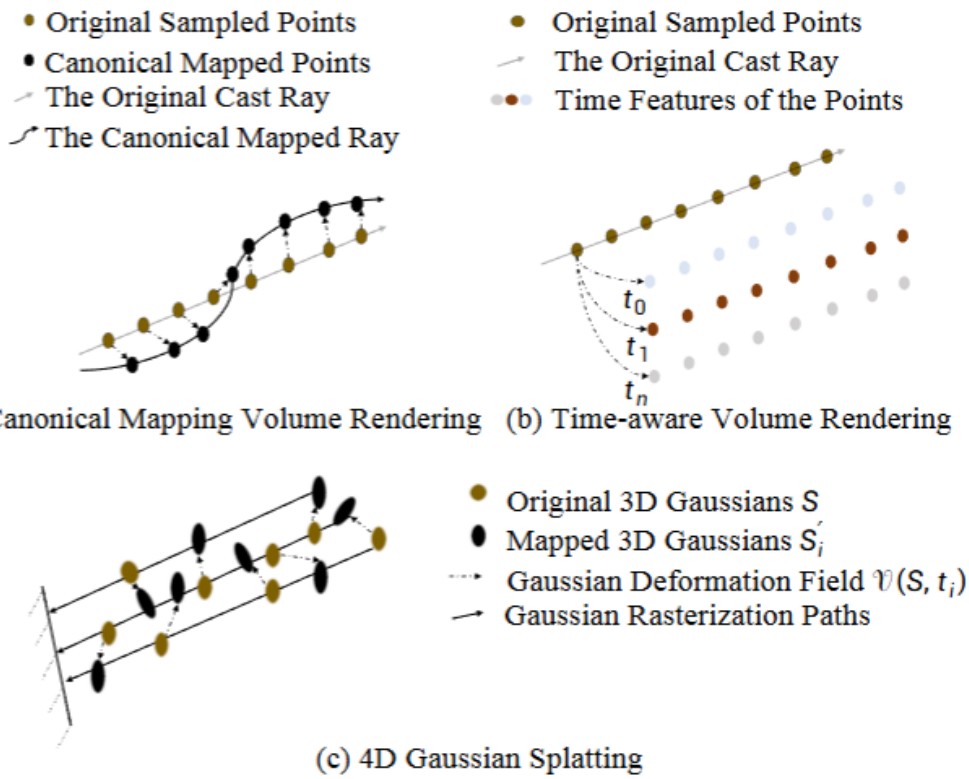
2. 方法

2.1 动态场景表示与渲染方法比较

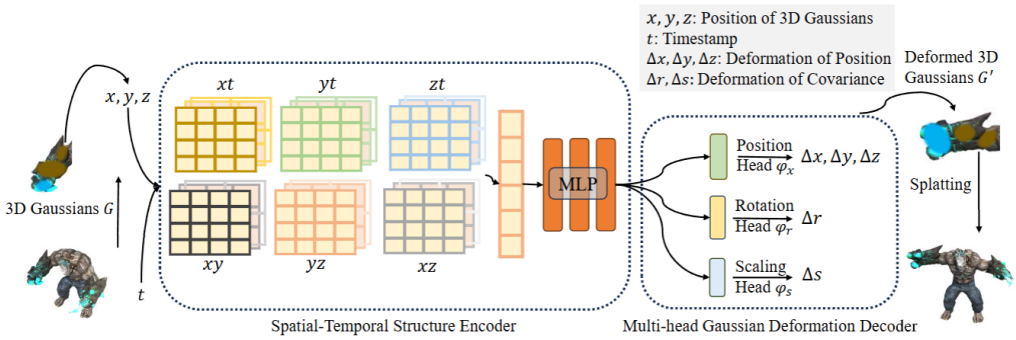
- 动态 NeRF 方法包括规范映射体绘制和时间感知体绘制，前者通过变形网络将采样点映射到规范空间，后者直接计算每个点的特征。这些方法在训练速度和实时渲染方面仍面临挑战，尤其是单目输入场景。
- 基于点云的神经渲染算法中，3D-GS 通过将场景表示为 3D 高斯实现实时渲染，但在处理动态场景时，存储成本会随时间序列增加。本文提出的 4D-GS 方法通过紧凑网络建模 3D 高斯运动，提高了训练效率和实时渲染能力。

2.2 4D 高斯溅射框架

- 给定视图矩阵和时间戳，4D 高斯溅射框架通过高斯变形场网络将原始 3D 高斯转换为变形后的 3D 高斯，然后通过微分溅射进行渲染。



- 高斯变形场网络包括空间 - 时间结构编码器和多头高斯变形解码器。空间 - 时间结构编码器利用多分辨率 HexPlane 和微小 MLP 编码 3D 高斯的时空特征，多头高斯变形解码器通过单独的 MLP 计算 3D 高斯的位置、旋转和缩放变形。



2.3 优化过程

- 3D 高斯初始化采用 SfM 点初始化，先优化 3D 高斯 3000 次迭代进行预热，然后用 3D 高斯渲染图像。
- 损失函数采用 L1 颜色损失和总变差损失，监督训练过程。

2.4 损失函数

与其他重建方法类似，作者使用 L1 颜色损失来监督训练过程。还应用了基于网格的总变分损失.

$$\mathcal{L} = \hat{I} - I + \mathcal{L}_{tv}.$$

3. 实验结果

3.1 数据集

作者主要基于 PyTorch 框架，并在单个 RTX 3090 GPU 中进行测试，并通过3D-GS [9]中概述的配置微调了相关优化参数。

3.1.1 合成数据集

作者主要使用 DNeRF [11]引入的合成数据集来评估模型的性能。这些数据集是针对单眼设置而设计的，但值得注意的是每个时间戳的相机姿势接近随机生成。

这些数据集中的每个场景都包含动态帧，数量从 50 到 200 不等。

3.1.2 真实世界数据集

作者使用 HyperNeRF[13]和 Neu3D[14]提供的数据集作为基准数据集来评估模型在现实场景中的性能。Nerfies 数据集是使用一台或两台相机在简单的相机运动后捕获的，而 Neu3D 的数据集是使用 15 到 20 个静态相机捕获的，涉及较长的周期和复杂的相机运动。作者使用 SfM 从 Neu3D 数据集中每个视频的第一帧计算出的点以及 HyperNeRF 中随机选择的 200 帧。

3.1 实验设置与结果分析

- 实验基于 PyTorch 框架，在 RTX 3090 GPU 上进行。使用合成数据集和真实数据集评估模型性能，包括峰值信噪比（PSNR）、结构相似性指数（SSIM）、LPIPS、帧率（FPS）、训练时间和存储等指标。
- 在合成数据集上，4D-GS 与其他方法对比，在渲染质量和速度上表现出色，如 PSNR 达到 34.05 dB，800×800 分辨率下 FPS 为 82。在真实数据集上，也能实现较高质量的渲染和较快的收敛速度。

Model	PSNR(dB)↑	SSIM↑	LPIPS↓	Time↓	FPS ↑	Storage (MB)↓
TiNeuVox-B [6]	32.67	0.97	0.04	28 mins	1.5	48
KPlanes [8]	31.61	0.97	-	52 mins	0.97	418
HexPlane-Slim [4]	31.04	0.97	0.04	11m 30s	2.5	38
3D-GS [14]	23.19	0.93	0.08	10 mins	170	10
FFDNeRF [12]	32.68	0.97	0.04	-	< 1	440
MSTH [37]	31.34	0.98	0.02	6 mins	-	-
Ours	34.05	0.98	0.02	20 mins	82	18

Table 2. Quantitative results on HyperNeRF's [25] vrig dataset. Rendering resolution is set to 960×540.

Model	PSNR(dB)↑	MS-SSIM↑	Times↓	FPS↑	Storage(MB)↓
Nerfies [24]	22.2	0.803	$\sim$ hours	< 1	-
HyperNeRF [25]	22.4	0.814	32 hours	< 1	-
TiNeuVox-B [6]	24.3	0.836	30 mins	1	48
3D-GS [14]	19.7	0.680	40 mins	55	52
FFDNeRF [12]	24.2	0.842	-	0.05	440
Ours	25.2	0.845	1 hour	34	61

Model	PSNR(dB)↑	D-SSIM↓	LPIPS↓	Time ↓	FPS↑	Storage (MB)↓
NeRFPlayer [35]	30.69	0.034	0.111	6 hours	0.045	-
HyperReel [2]	31.10	0.036	0.096	9 hours	2.0	360
HexPlane-all* [4]	31.70	0.014	0.075	12 hours	0.2	250
KPlanes [8]	31.63	-	-	1.8 hours	0.3	309
Im4D [18]	32.58	-	0.208	28 mins	$\sim$5	93
MSTH [37]	32.37	0.015	0.056	20 mins	2(15‡)	135
Ours	31.15	0.016	0.049	40 mins	30	90

为评估新视图合成的质量，作者对几种领域内的先进方法进行了基准测试，相关结果汇总在表1中。结果表明，尽管现有的动态混合表示方法能够生成高质量的渲染结果，但通常在渲染速度上存在不足。由于缺乏对动态运动部分的有效建模，这些方法在动态场景的重建上表现出一定局限性。相比之下，本文提出的方法在合成数据集上的表现尤为出色，不仅实现了最高的渲染质量和极快的渲染速度，同时显著降低了存储需求和收敛时间。

此外，表2和表3展示了在真实世界数据集上的实验结果。可以看出，部分NeRF方法存在收敛速度缓慢的问题，而基于网格的NeRF方法在捕捉复杂物体细节时表现乏力。相比之下，本文提出的方法在真实数据集中表现出色，兼具较高的渲染质量、快速的收敛能力以及室内场景下的卓越自由视图渲染速度。尽管Im4D在渲染质量方面优于4D-GS，但其对多摄像头设置的依赖限制了其在单目场景中的应用。此外，其他方法也在自由视图渲染速度和存储效率上存在局限性。

3.2 消融研究

- 空间 - 时间结构编码器中，HexPlane 编码器能保留 3D 高斯的时空信息，提高渲染质量，去除该模块会导致模型无法准确建模复杂变形。
- 高斯变形解码器中，位置、旋转和缩放变形的建模对准确拟合动态场景细节至关重要，缺少任何一项都会影响渲染质量。
- 3D 高斯初始化对模型收敛有重要影响，适当的初始化可使模型更好地学习动态部分，避免数值误差，提高渲染质量。

Table 4. Ablation studies on synthetic datasets using our proposed methods.

Model	PSNR(dB)↑	SSIM↑	LPIPS↓	Time↓	FPS↑	Storage (MB)↓
Ours w/o HexPlane $R_l(i,j)$	27.05	0.95	0.05	10 mins	140	12
Ours w/o initialization	31.91	0.97	0.03	19 mins	79	18
Ours w/o ϕ_x	26.67	0.95	0.07	20 mins	82	17
Ours w/o ϕ_r	33.08	0.98	0.03	20 mins	83	17
Ours w/o ϕ_s	33.02	0.98	0.03	20 mins	82	17
Ours	34.05	0.98	0.02	20 mins	82	18

2.6 讨论

- 多分辨率 HexPlane 高斯编码器通过在体素平面中编码 3D 高斯特征，提高了训练和渲染速度及质量，可视化显示其能有效捕捉场景结构和运动信息。
- 3D 高斯可用于跟踪任务，在单目设置下能以较低存储呈现跟踪对象的运动轨迹。
- 4D 高斯可实现不同场景的组合，通过预测变形后的 3D 高斯并进行微分渲染，能将多个场景合成到同一空间。
- 渲染速度受渲染分辨率、3D 高斯数量和高斯变形场网络能力等因素影响，需在这些因素间平衡以实现实时渲染。

2.7 局限性

- 4D-GS 在处理大运动、无背景点和不精确相机姿态场景时优化困难。
- 在单目设置下，难以分离静态和动态高斯部分的关节运动，需额外监督。
- 对于城市规模重建，由于大量 3D 高斯对高斯变形场的查询开销大，需设计更紧凑算法

结论与展望

本文提出的4D Gaussian Splatting方法成功解决了动态场景高效建模与实时渲染的问题，为动态场景表示提供了一种新的可能。未来可以进一步优化模型以应对更大范围的场景动态，以及探索多相机设置下的优化方法。