

三维高斯溅射数据压缩技术的改进

淮卓桦, 裴树军*

(哈尔滨理工大学计算机科学与技术学院 哈尔滨 150080)
(Peisj@hrbust.edu.cn)

摘要: 三维高斯溅射(3D Gaussian splatting, 3DGS)是新视图合成领域中的新技术。针对3DGS技术合成的场景存储空间大的问题,提出一种高效的压缩方法。首先考虑随着迭代次数的增加逐步增强梯度的权重,通过修改原本的自适应密度控制策略,更精准地控制密度;然后利用敏感性感知向量聚类对高斯点的多个属性进行压缩,如颜色特征、协方差矩阵和不透明度参数;最后对极其紧凑且颜色、形状和不透明度参数相似度高的小高斯点进行合并,进一步优化存储空间。在多个数据集上的实验结果表明,与3DGS相比,所提方法在存储空间上平均减少97.64%,PSNR平均提升0.307 dB。

关键词: 新视图合成; 高斯溅射; 三维重建; 模型压缩

中图分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2024-00600

Improvement in Compression Technique for 3D Gaussian Splatting Data

Huai Zhuohua and Pei Shujun*

(School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080)

Abstract: 3D Gaussian Splatting (3DGS) is a new technology in the field of novel view synthesis. To address the problem of large storage space for scenes synthesized by 3DGS technology, an efficient compression method is proposed. Firstly, considering gradually increasing the weight of gradients as the number of iterations increased, the original adaptive density control strategy was modified to control density more accurately. Secondly, sensitivity-aware vector clustering was utilised to compress multiple attributes of Gaussian points, such as color features, covariance matrices, and opacity parameters. Finally, small Gaussian points that are extremely compact and have high similarity in color, shape, and opacity parameters were merged to further optimize storage space. Experimental results on multiple datasets show that compared with 3DGS, the proposed method reduces the storage space by an average of 97.64% and improves the PSNR by an average of 0.307 dB.

Key words: novel view synthesis; Gaussian splatting; 3D reconstruction; model compression

新视图合成(novel view synthesis, NVS)旨在从图片或视频中捕获的场景生成新的三维视图,该领域在过去几十年里取得了显著的进步。当前,传统的多视角立体方法^[1]逐步被深度学习所取代,尤其是神经辐射场(neural radiance fields, NeRF)^[2]方法。

NeRF及其改进方法通过使用小型神经网络,如多层感知机(multilayer perceptron, MLP)和体渲

染直接渲染出新视图场景,已被广泛应用于三维重建中。虽然NeRF本身已经展现了非常好的NVS质量,但其中的神经网络计算成本高、训练时间长,严重降低了NVS的效率。为了解决这一问题,三维高斯溅射(3D Gaussian splatting, 3DGS)^[3]应运而生。

3DGS是一种基于点的渲染方法,利用从运动恢复结构(structure from motion, SfM)中提取的稀

疏点云作为初始化,将每个点云扩展为一个具有形状、颜色和不透明度等参数的各向异性的三维高斯基元,并使用高度定制化的 CUDA 内核算法进行光栅化管道渲染.由于 3DGS 的参数是可微分的,因此在匹配图像的过程中通过可微分渲染进行优化.目前,3DGS 已被扩展到了其他领域,如动态建模^[4-7]、自动驾驶领域^[8]、机器人 SLAM^[9-11]、人工智能生成内容(artificial intelligence generated content, AIGC)^[12-14],以及医疗领域^[15-16]等;然而,其需要大量的三维高斯基元来保持渲染图像的高保真度,每个三维高斯基元有高达 62 个参数,占用了大量的存储空间,对于渲染效率和数据传输是极大的挑战.

为了压缩重建的场景,本文首先改进三维高斯基元的自适应密度控制策略,提出将迭代次数加入跨视图的梯度平均中,以放大迭代次数更高的三维高斯基元的梯度贡献;然后针对直接删除低不透明度高斯基元导致成像细节丧失的问题,本文通过仅删除一小部分极低不透明度的三维高斯基元,并计算剩余的三维高斯基元不透明度参数对图像的贡献度,得到不透明度敏感性评分,再利用矢量量化编码将评分较高的不透明度参数编码到紧凑的码本中;再利用 KNN 算法寻找存在大量小三维高斯基元的区域,在这些区域融合颜色、不透明度和形状相似度高的小三维高斯基元,进一步压缩存储空间;还提出不透明度参数压缩后专门用于渲染的 GPU 排序和光栅化的场景渲染器,可以实时地进行包含不透明度参数编码本索引的场景渲染;最终,结合改进的光栅化器,进行微调.

1 相关工作

1.1 NeRF 表示及压缩

2020 年, Mildenhall 等^[2]提出的 NeRF 是结合深度学习方法的三维重建技术,它使用体渲染通过 MLP 隐式地表达三维场景.在小型 MLP 网络中,输入一个 5D 向量 (\mathbf{X}, \mathbf{d}) ,输出依赖于方向的颜色 \mathbf{c} 和相应位置的密度 σ .自 NeRF 出现以来,已有大量的改进工作致力于研究提升不同场景的性能,如大型场景^[17-18]和动态场景^[19-20].

然而,NeRF 本身训练时间极长,存储空间占用大,为了改进这些缺点,多种改进方法应运而生. Xu 等^[21]提出的 Point-NeRF 使用特征点云作为体渲染的中间步骤,提高了 NeRF 本身的训练速

度; Hedman 等^[22]提出 SNeRG,将预计算的 NeRF 存储在稀疏体素网格上,其速度比原始 NeRF 提升近 3 000 倍.为了减少 NeRF 的存储消耗, Chen 等^[23]提出混合张量秩分解 TensorRF; Tang 等^[24]提出向量矩阵分解,将张量分解为紧凑的向量因子和矩阵因子; Deng 等^[25]提出 Re-NeRF,通过对体素网格进行剪枝,减少了 NeRF 模型的存储空间; Li 等^[26]提出矢量量化辐射场,用于压缩基于网格的辐射场; Takikawa 等^[27]使用 K 均值向量量化初始化编码本,提出可学习的索引.

尽管有许多关于 NeRF 的工作^[28-29]致力于提高训练速度和减少内存消耗,但是 NeRF 在大场景的实时渲染能力仍然是具有挑战的,体渲染和 MLP 本身成为了一种限制.

1.2 3DGS 表示及压缩

近年来,基于点的渲染方法已经被广泛应用于渲染三维场景,这是一种使用点而非传统的多边形网格渲染三维场景的技术.点渲染特别适用于复杂、非结构化或稀疏的几何数据,如点云.通常,点云是 SfM 的直接输出,但点渲染容易出现空洞,导致成像不连续.为了解决空洞问题,研究人员提出基于点的可微渲染, Point-NeRF 使用神经三维点云重建三维场景,并提出修剪和生长机制解决空洞和异常值.

3DGS 使用三维高斯结合栅格化方法进行渲染,有效地解决了空洞问题;作为 NVS 的新方法,与原始 NeRF 相比,3DGS 在训练速度上取得了显著提升.

在解决空洞问题方面,3DGS 引入自适应密度控制策略增加三维高斯基元的数量.然而,对于决定哪些点需要克隆与分裂时,原始的自适应密度控制策略考虑得不够全面. AbsGS^[30]中指出,在包含高频细节的复杂场景的过度重建区域中,原始的自适应密度控制策略只使用视图空间中的平均梯度作为每个高斯是否需要致密的标准,渲染的图像模糊;而每个三维高斯基元的像素级梯度方向并不相同,会发生梯度冲突,导致梯度衰减,忽略了某个高斯的致密化:因此提出同向位置梯度,仅保留梯度的大小,不考虑方向,取每个梯度分量的绝对值进行累加,最终将综合的梯度作为高斯的最终梯度. Pixel-GS^[31]中指出,由于一部分大高斯覆盖的像素数量过多,使得最终的平均梯度太小,而这些点难以进行克隆和分裂,导致这些区域模型化效果不理想,因此提出使用每个三维高斯基元覆盖的像素数量作为该三维高斯基元梯度的

一个权重, 合理地促进这些高斯致密化.

然而, AbsGS 会导致三维高斯基元的平均梯度增大, 尽管提高了梯度阈值, 但仍然使得一些不必要的三维高斯基元分裂. 计算平均梯度时, Pixel-GS 虽然引入三维高斯基元覆盖的像素数量作为权重, 可以在一定程度上提高场景重建的准确性, 但这种方法在迭代过程中有缺陷; 特别是在每 100 次迭代的开始阶段, 当三维高斯基元覆盖的像素数量较多、其对应的梯度也相对较大时, 随着迭代的深入, 三维高斯基元经过优化后覆盖的像素数量可能减少, 相应的梯度也可能会降低. 这种变化本应使得该三维高斯基元不再需要进行克隆或分裂, 但由于像素数量的权重影响, 使得加权后的平均梯度可能依然高于设定的阈值, 从而错误地引起高斯基元的分裂. 在权重机制可能导致梯度变化过度敏感的情况下, 这种过度敏感可能引起三维高斯基元进行不必要的克隆或者分裂, 影响最终场景重建的效率和质量.

尽管 AbsGS 和 Pixel-GS 在渲染质量上取得了一些进步, 但是耗费了大量的存储空间, 一个中等大小的场景就占用了数百兆的存储空间.

为了解决原始的 3DGS 本身占用了极大的存储空间这一关键问题, 研究人员开发了各种压缩技术^[32]来优化 3DGS 的效率. Shi 等^[33]提出的一种分层递进的 3DGS 技术 LapisGS, 采用多层结构, 每层通过累积表示法逐步增加细节, 实现连续的渲染质量适应; 同时, 引入动态不透明度优化, 在训练过程中调整各层的贡献, 以保持视觉保真度; 最后利用占用图映射有效地管理高斯基元, 排除重要性较低的基元, 提高计算和存储效率. Navaneet 等^[34]提出 CompGS, 使用 K 均值聚类算法对高斯参数进行向量量化, 对不透明度进行正则化, 提升了存储效率. Lu 等^[35]提出 Scaffold-GS, 利用锚点分布三维高斯基元, 根据视角方向和距离动态地预测高斯基元属性, 极大地减少了冗余高斯基元. Chen 等^[36]利用锚点和哈希网格进行上下文建模, 提出哈希网格辅助上下文框架 HAC, 并使用一个自适应掩蔽策略消除无效的高斯基元和锚点, 减少了存储空间. Fan 等^[37]提出 LightGaussian, 可以识别并修剪对场景重建贡献不大的高斯, 有效地减少冗余的高斯基元; 同时, 提出一种混合 VecTree 量化方案, 降低了位宽表示. Fang 等^[38]提出 Mini-Splatting, 引入了密集化和简化算法, 通过重组高斯的空間位置, 实现了更均匀和有效的高斯分布.

Joo 等^[39]提出基于体积的掩码策略 Compact-GS, 去除冗余高斯点并且压缩高斯属性, 使用小型 MLP 表示颜色, 显著地减少了存储需求; 尽管该策略取得了良好的结果, 但是其颜色特征是使用小型 MLP 表示的, 并没有使用球谐函数 (spherical harmonics, SH), 导致无法直接进行三维可视化.

Niedermayr 等^[40]提出敏感值感知向量聚类算法 Compress-GS, 首先对颜色和高斯形状参数进行敏感性评分, 保留超过一定阈值的参数, 对低于这个阈值的参数使用 K 均值聚类初始代码本; 然后通过迭代优化代码本; 该模型通过压缩颜色特征参数和形状参数, 并结合熵编码技术, 显著地减少了数据的存储空间, 虽然熵编码后的数据 (存储为 npz 格式) 在体积上得到了极大的压缩, 但这些压缩后的数据并不能直接用于点云的可视化, 必须先转换回 ply 文件格式才能进行查看, 而 ply 文件的大小与原始的 3DGS 文件相比并未显著减少. Compress-GS 在压缩过程中有效地降低了数据的存储需求, 但在实际应用中, 如点云可视化, 文件的还原过程仍然会恢复大部分的空间占用.

2 本文方法

2.1 可微 3DGS

3DGS 从 SfM 生成的稀疏点云初始化, 用一组可微的各向异性的三维高斯基元表示三维场景信息. 每个三维高斯基元的形状 \mathbf{G} 可以表示为

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

其中, 每个点云作为三维高斯基元的中心位置 μ ; x 表示场景中的任意位置; Σ 表示协方差矩阵. 由于 Σ 必须是正定的, 如果用 Σ 直接参与优化, 难以保证 Σ 保持正定, 因此用旋转矩阵 \mathbf{R} 和缩放矩阵 \mathbf{S} 来参数化 Σ , 公式为

$$\Sigma = \mathbf{R}\mathbf{S}\mathbf{S}^T\mathbf{R}^T.$$

为了单独优化 \mathbf{R} 和 \mathbf{S} , 3DGS 将 \mathbf{R} 用一个四元数 \mathbf{q} 表示, \mathbf{S} 用一组三维向量 \mathbf{s} 表示.

每个三维高斯基元中除了形状参数外, 还有外观参数颜色 \mathbf{C} 和不透明度 α , 其中, \mathbf{C} 由 SH 组成, 表示不同方向的颜色. 在渲染阶段, 3DGS 利用栅格化渲染, 混合重叠每个像素 \mathbf{p} 对应的 n 个有序 2D 高斯, 并使用 α 混合, 计算 \mathbf{p} 的最终 \mathbf{C} , 公式为

$$C(x') = \sum_{i=1}^n c_i \alpha_i \left(\prod_{j=1}^{i-1} (1 - \alpha_j) \right);$$

其中, x' 表示查询的像素位置; n 表示与查询的像素相关的有序 2D 高斯数量; c_i 表示解码 SH 后得到的对应视图方向的颜色。

在优化阶段, 通过不断优化 μ , s , q , SH 系数和 α , 使渲染的 2D 图像与训练图像匹配。

2.2 像素感知和次数感知的累积权重

在原始的 3DGS 中, 每 100 次迭代的自适应密度控制, 使用跨视图的平均梯度大小作为需要致密化的条件, 每个高斯点 i 在多个视图 k 下的平均梯度大小为

$$\frac{\sum_{k=1}^M \sqrt{\left(\frac{\partial L_k}{\partial v_{i,k,x}} \right)^2 + \left(\frac{\partial L_k}{\partial v_{i,k,y}} \right)^2}}{M} > \tau;$$

其中, L_k 表示视图 k 下的损失; $v_{i,k}$ 表示第 i 个高斯点在视图 k 下的平面坐标; M 表示高斯点在每 100 次迭代中参与的视图总数; τ 表示梯度阈值, 原始 3DGS 中设置为 0.000 2。

AbsGS 使用同向位置梯度, 仅保留梯度的大小, 不考虑方向, 取每个梯度分量的绝对值进行累加, 将综合的梯度作为该高斯的最终梯度, 公式为

$$\left\{ \begin{array}{l} \mathbf{grad}_{i,x} = \sum_{k=1}^M \left| \frac{\partial L_k}{\partial v_{i,k,x}} \right| \\ \mathbf{grad}_{i,y} = \sum_{k=1}^M \left| \frac{\partial L_k}{\partial v_{i,k,y}} \right| \\ \mathbf{grad}_i = \frac{\sqrt{\mathbf{grad}_{i,x}^2 + \mathbf{grad}_{i,y}^2}}{M} \end{array} \right.$$

Pixel-GS 在原始的自适应密度控制策略中加入了高斯基元覆盖的像素数量作为权重, 公式为

$$\frac{\sum_{k=1}^M \left(m_i^k \times \sqrt{\left(\frac{\partial L_k}{\partial v_{i,k,x}} \right)^2 + \left(\frac{\partial L_k}{\partial v_{i,k,y}} \right)^2} \right)}{\sum_{k=1}^M m_i^k};$$

其中, m_i^k 表示第 i 个高斯基元在视图 k 下, 三维高斯基元覆盖的像素数量。

本文认为, 每次致密化后, 随着迭代次数的增加, 三维高斯基元在不断优化, 越靠后的梯度应该占有更高的权重。因此, 本文提出一种逐步增强的权重参与梯度的累加, 来计算平均梯度, 公式为

$$\frac{1}{2} \times \left[\frac{1}{2} \times \frac{\sum_{k=1}^M \left(d_i \times m_i^k \times \sqrt{\left(\mathbf{grad}_{i,x} \right)^2 + \left(\mathbf{grad}_{i,y} \right)^2} \right)}{M \times \sum_{k=1}^M m_i^k} + \frac{\sum_{k=1}^M \sqrt{\left(\frac{\partial L_k}{\partial v_{i,k,x}} \right)^2 + \left(\frac{\partial L_k}{\partial v_{i,k,y}} \right)^2}}{M} \right];$$

其中, 由于每 100 次进行一次致密化, 则 d_i 表示这 100 次进行致密化的过程中的当前迭代次数。

2.3 不透明度梯度感知和向量聚类

在原始 3DGS 中, 每个三维高斯基元都有高达 62 个参数, 其中, 颜色特征有 48 个, 形状参数有 7 个, 因此, Compact-GS 和 Compress-GS 压缩了颜色特征和形状参数。

在原始的 3DGS 中, 每 100 次迭代, 会删除 α 低于 0.005 的三维高斯基元。本文通过分析训练完成后的 α , 在删除一定阈值 γ_α 的低 α 的三维高斯基元后, 图像质量并没有明显降低, 如表 1 所示。

表 1 删除不同 γ_α 的 α 后的图像质量比较

γ_α	PSNR/dB	大小/MB
0.005	20.698	157.927
0.01	20.683	148.813
0.05	20.452	117.364
0.1	19.634	95.635

Salomon 等^[41]的实验结果表明, 峰值信噪比 (peak signal to noise ratio, PSNR) 差值小于 0.5 时人眼是无法区分的, 因此本文选取 0.05 作为删除 α 的 γ_α 。对于 α 高于 γ_α 的三维高斯基元, 继续直接删除会导致成像的细节损失。受到 Compress-GS 的启发, 本文对 α 使用梯度感知和向量聚类, 在删除 α 低于 γ_α 的三维高斯基元的基础上, 使用梯度感知, 将 α 的敏感性 w 定义为

$$w(\alpha) = \frac{1}{\sum_{k=1}^N P_k} \sum_{k=1}^N \left| \frac{\partial E_k}{\partial \alpha} \right|;$$

其中, N 表示用于场景重建的训练集中的视图总数; P_k 表示视图 k 中的像素总数; E_k 表示视图 k 的总能量, 即所有像素上的 RGB 分量之和。为了衡量 E_k 对 α 变化的敏感性, 采用 E_k 相对于 α 的梯度进行表示。

对于 α 高于 γ_α , 并且 w 大于 0 的三维高斯基元, 采用 K 均值聚类算法进行处理。首先随机且均匀地初始化这些三维高斯基元的 α 编码本, 然后采用迭代优化的方式更新编码本, 达到更稳定的

结果.

2.4 聚类紧凑高斯点集

使用可视化工具渲染三维场景时, 本文发现, 在 Mip-NeRF 360^[42]的大场景的某些区域中, 存在许多颜色差异极小的小三维高斯基元, 这些小三维高斯基元彼此紧密地排列在一起, 形成高度密集的区域. 在这些高度密集的区域, 由于小三维高斯基元在 C , α 和形状上有高度的相似性, 因此可以通过将这些高密度、高相似性的小三维高斯基元合并成一个较大的三维高斯基元, 优化场景的表示, 这种合并方式不仅减少了数据的冗余性, 还能提高渲染效率.

基于上述思想, 本文提出一种基于局部近邻搜索和相似性度量的三维高斯基元合并算法, 用于减少数据的冗余性, 优化三维场景的表示, 首先使用最近邻搜索计算每个三维高斯基元的局部密度, 对于每个三维高斯基元 i , 将其 t 个最近的高斯点的平均距离 $l(i, j)$ 的倒数作为密度 D_i , 公式为

$$D_i = \frac{1}{\frac{1}{t} \sum_{j=1}^t l(i, j)};$$

本文中, t 取为 20.

本文使用密度阈值 γ_d 选择密度值较高的三维高斯基元, 在这些三维高斯基元的邻域内进行搜索, 找到距离小于阈值 γ_1 的邻居. 对于每个邻居 j , 计算其与高斯基元 i 的相似性, 相似性的条件定义为

$$f(i, j) = (\|\alpha_i - \alpha_j\| \leq \delta) \cap (\|C_i - C_j\| \leq \delta) \cap (\|S_i - S_j\| \leq \delta) \cap (\|R_i - R_j\| \leq \delta);$$

其中, δ 表示相似性阈值.

将最终满足条件的邻居, 合并成一个新的三维高斯基元, 用它们的平均值作为新三维高斯基元的参数; 合并完成后, 删除原来的三维高斯基元.

2.5 微调与渲染

在 Compress-GS 的微调中, 硬件光栅化器在渲染过程中有效地处理了经过压缩的 C 和形状参数, 但该光栅化器并不适用于本文提出的压缩 α 后的渲染, 导致渲染结果出现许多噪点. 为了解决这一限制, 本文对光栅化器进行有针对性的修改, 使其能够更好地支持 α 的压缩与编码本索引的使用.

本文提出一种改进后的硬件光栅化器, 专门优化对 α 编码本索引的处理, 确保在微调阶段, 渲染时能够准确地反映压缩后的 α 的数据. 通过改进, 不仅提高硬件光栅化器在处理 α 时的准确

性, 还确保在复杂场景下, 所有特征参数(包括 C , 形状和 α)都能在压缩后得到一致且高质量的渲染效果.

3 实验结果与分析

3.1 数据集

在 14 个场景下对本文方法进行实验, 涵盖了室内和室外、有边界和无边界场景. 这些场景来源于 NeRF Synthetic 数据集^[2]的 8 个场景、Mip-NeRF 360 数据集^[42]的 2 个场景、Tanks&Temples 数据集^[43]的 2 个场景和 Hedman 等^[44]提出的数据集(简称为 Deep Blending 数据集)的 2 个场景. 对比模型包括 3DGS^[3], Compact-GS^[39], Compress-GS^[40], Pixel-GS^[31], LightGaussian^[37]和 Mini-Splatting^[38]. 所有实验均在 RTX3090 GPU 上进行.

3.2 参数设置

本文在批处理 α 的聚类时, 使用的 α 码本大小为 1 024, γ_α 为 6×10^{-6} , 对 α 编码本进行 800 次更新步骤, 批量大小为 2^{18} . 在高斯合并阶段, 使用的 γ_d 为 200, γ_1 为 0.2, δ 为 0.1.

本文方法的总迭代次数为 12k, 其中, 前 7k 为致密化阶段, 致密化后进行 α 删除, 同时进行合并大高斯, 然后对高斯参数进行编码; 后 5k 为微调阶段.

3.3 对比实验

选取 Tanks&Temples 数据集^[43]中的 train 场景, 在 3DGS^[3], Pixel-GS^[31]和本文提出的权重的模型中进行实验, 前 600 次迭代的平均梯度对比, 如图 1 所示.

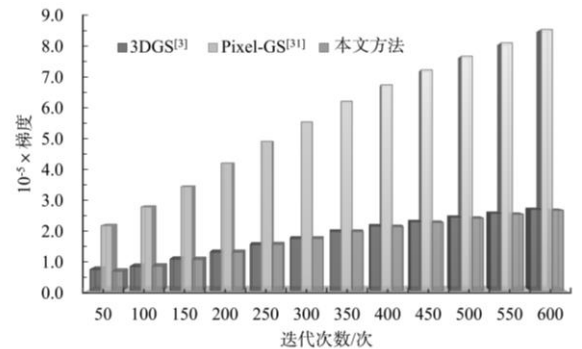


图 1 不同模型在 train 场景前 600 次迭代平均梯度对比. 在此基础上经过完全训练后, 其性能结果如表 2 所示.

表 2 在 train 场景上 3 种模型的性能比较

模型	PSNR/dB	大小/MB
3DGS ^[3]	20.698	157.927

Pixel-GS ^[31]	20.766	514.336
本文方法	20.490	90.766

从表 2 可以看出:

(1) 与 3DGS^[3]相比, 存储空间减少 42.53%, PSNR 仅降低 0.208 dB.

(2) 与 Pixel-GS^[31]相比, 存储空间减少 82.35%, PSNR 仅降低 0.276 dB.

对于所有场景, 本文使用未压缩前的文件大小(简称为 ply-Size), 压缩后的文件大小(简称为

npz-Size), Compact-GS^[39]压缩后的文件大小(简称为 pth-Size), PSNR, 结构相似性指数(structural similarity index, SSIM)和感知图像质量(learned perceptual image patch similarity, LPIPS)^[45]这 6 个指标进行评估. 在 Mip-NeRF 360 数据集^[42]和 Tanks&Temples 数据集^[43]上, 对 7 种模型的平均性能进行实验, 结果如表 3 所示. 在 NeRF Synthetic 数据集^[2]和 Deep Blending 数据集^[44]上, 对 7 种模型的平均性能进行实验, 结果如表 4 所示.

表 3 在 Mip-NeRF 360 和 Tanks&Temples 数据集上 7 种模型的平均性能对比

模型	Mip-NeRF 360 ^[42]					Tanks&Temples ^[43]				
	ply-Size/MB	PSNR/dB	SSIM	LPIPS	耗时/min	ply-Size/MB	PSNR/dB	SSIM	LPIPS	耗时/min
3DGS ^[3]	509.476	21.054	0.621	0.393	4:34	298.399	22.725	0.797	0.258	2:35
Compact-GS ^[39] (pth-Size)	91.095 (48.097)	20.581	0.595	0.421	5:49	65.859 (35.824)	22.053	0.770	0.294	2:50
Compress-GS ^[40] (npz-Size)	506.032 (20.825)	21.434	0.633	0.385	10:12	295.826 (14.678)	23.917	0.829	0.230	4:32
Pixel-GS ^[31]	956.210	21.678	0.659	0.367	5:41	693.413	22.855	0.807	0.239	3:36
本文方法-12k (npz-Size)	91.566 (7.206)	20.156	0.555	0.472	10:54	99.819 (8.672)	23.371	0.828	0.259	5:54
LightGaussian ^[37] (npz-Size)	178.400 (51.850)	22.545	0.688	0.335	56:54	98.459 (28.796)	24.861	0.841	0.222	54:35
Mini-Splatting ^[38] (npz-Size)	147.287 (35.522)	22.858	0.700	0.352	25:16	71.233 (16.577)	25.050	0.868	0.172	13:38
本文方法-30k (npz-Size)	141.731 (10.778)	21.198	0.618	0.412	21:35	125.442 (10.700)	25.225	0.862	0.192	14:58

表 4 在 NeRF Synthetic 和 Deep Blending 数据集上 7 种模型的平均性能对比

模型	NeRF Synthetic ^[2]					Deep Blending ^[44]				
	ply-Size/MB	PSNR/dB	SSIM	LPIPS	耗时/min	ply-Size/MB	PSNR/dB	SSIM	LPIPS	耗时/min
3DGS ^[3]	48.842	33.038	0.969	0.038	1:14	402.440	30.501	0.895	0.271	4:23
Compact-GS ^[39] (pth-Size)	10.547 (6.975)	30.804	0.956	0.051	1:40	72.453 (48.121)	29.069	0.882	0.318	4:57
Compress-GS ^[40] (npz-Size)	48.193 (3.007)	33.358	0.973	0.035	3:50	399.611 (17.394)	32.114	0.918	0.264	9:03
Pixel-GS ^[31]	81.931	33.407	0.971	0.036	2:20	612.447	30.066	0.892	0.296	5:37
本文方法-12k (npz-Size)	11.911 (1.467)	33.078	0.966	0.045	4:39	123.328 (8.476)	31.942	0.909	0.045	9:49
LightGaussian ^[37] (npz-Size)	13.957 (4.192)	34.331	0.966	0.039	24:15	118.153 (34.197)	32.365	0.918	0.230	60:02
Mini-Splatting ^[38] (npz-Size)	31.585 (7.027)	34.908	0.967	0.030	11:10	128.944 (27.798)	34.782	0.938	0.229	19:15
本文方法-30k (npz-Size)	15.364 (1.868)	34.882	0.965	0.035	7:31	204.873 (12.759)	34.824	0.941	0.227	23:40

为了确保实验设计的公平性与结果的准确性, 对 3DGS^[3], Compact-GS^[39], Compress-GS^[40], Pixel-GS^[31], LightGaussian^[37]和 Mini-Splatting^[38]这 6 种模型进行了重新训练, 所有模型默认的训练周期均设定为 12k 步. 其中, 3DGS^[3], Compact-GS^[39]和 Pixel-GS^[31]的基础训练均为 7k 步; Compress-GS^[40]则接续 3DGS^[3]的输出进行进一步训练, 添加了 5k 步的微调, 使总训练步数达到 12k 步; 本文也是在更改权重算法后, 进行 7k 步的训练, 编码后再进行 5k 步的微调, 该连续性的设计既保证从 3DGS^[3]到 Compress-GS^[40]的技术衔接, 也确保了不同模型之间比较的一致性和科学性.

对 LightGaussian^[37]和 Mini-Splatting^[38]进行实验时, 基于其设计的复杂性, 依照原文献的建议, 进行了 30k 步的训练, 确保这 2 种模型在评估过程中达到最佳表现. 这种方法论的选择体现了实验设计的适应性, 旨在通过合适的训练长度, 展现各模型在最佳状态下的性能差异.

从对 7 种模型的实验中可以看出:

(1) 与 3DGS^[3]相比, 本文方法实现了显著的压缩效果, 存储空间平均减少 97.64%; 从图像质量的角度看, 本文方法的 PSNR 平均提升 0.307 dB.

(2) 与 Compact-GS^[39]相比, 本文方法虽然在 ply 文件格式的数据压缩效率方面未能超越, 但该方法提供了直接的三维可视化能力, 这是其显著的优势; 更重要的是, 从图像重建质量的角度看, 本文方法的 PSNR 平均提升 1.510 dB.

(3) 与 Compress-GS^[40]相比, 尽管本文方法的 PSNR 平均下降 0.569 dB, 但在压缩效率方面取得了显著成果; 应用熵编码之前, 本文方法已在 ply 文件格式的存储空间中平均减少 73.15%, 显著地减少了存储需求; 应用熵编码后, npz 文件格式的存储空间进一步平均减少 52.20%.

(4) 与 Pixel-GS^[31]相比, 本文方法实现了显著的压缩效果, 存储空间平均减少 98.71%; 从图像质量的角度看, 本文方法的 PSNR 平均提升 0.135 dB.

(5) 在迭代 30k 步中, 与 LightGaussian^[37]相比, 本文方法的 PSNR 平均提升 0.507 dB; 尽管 ply 文件格式的平均文件大小增加 19.610 MB, 但在 npz

文件格式的存储空间中平均减少 65.05%; 在时间效率上, 本文方法所需的处理时间显著低于 LightGaussian^[37].

(6) 与 Mini-Splatting^[38]相比, 本文方法的 PSNR 平均降低 0.367 dB; 在压缩效率方面, 对于全部场景的 ply 文件格式, 文件大小平均增加 27.090 MB; 尽管整体效果略有不足, 但是在 NeRF Synthetic 数据集^[2]这种小场景中本文方法的表现更为优异, 时间效率更高, 并且在 ply 文件格式的存储空间中平均减少 51.36%, 而 PSNR 仅降低 0.026 dB, 几乎没有差异. 本文方法在所有数据集最终压缩的 npz 文件格式中压缩效率更高, 存储空间平均减少 58.16%.

3DGS^[3], Compact-GS^[39], Compress-GS^[40], Pixel-GS^[31], LightGaussian^[37], Mini-Splatting^[38]和本文方法这 7 种模型在 NeRF Synthetic 数据集^[2]的压缩可视化结果如图 2 所示.

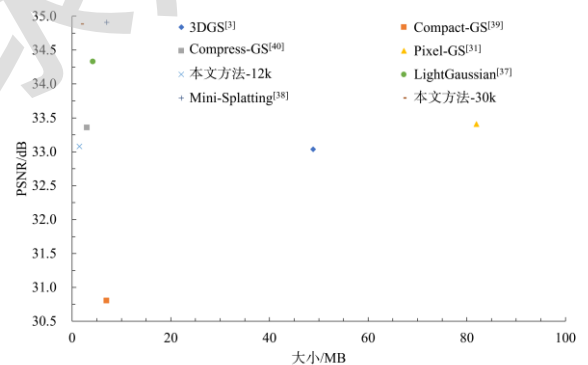


图 2 在 NeRF Synthetic 数据集中 7 种模型的压缩可视化对比

NeRF Synthetic 数据集^[2]为小场景, Mip-NeRF 360 数据集^[42]、Tanks&Temples 数据集^[43]和 Deep Blending 数据集^[44]为中小型场景, 在 3DGS^[3], Compress-GS^[40], Pixel-GS^[31], LightGaussian^[37], Mini-Splatting^[38]和本文方法进行实验, 渲染后图像的可视化结果, 以及 PSNR 与最终文件大小的对比如图 3 和图 4 所示.

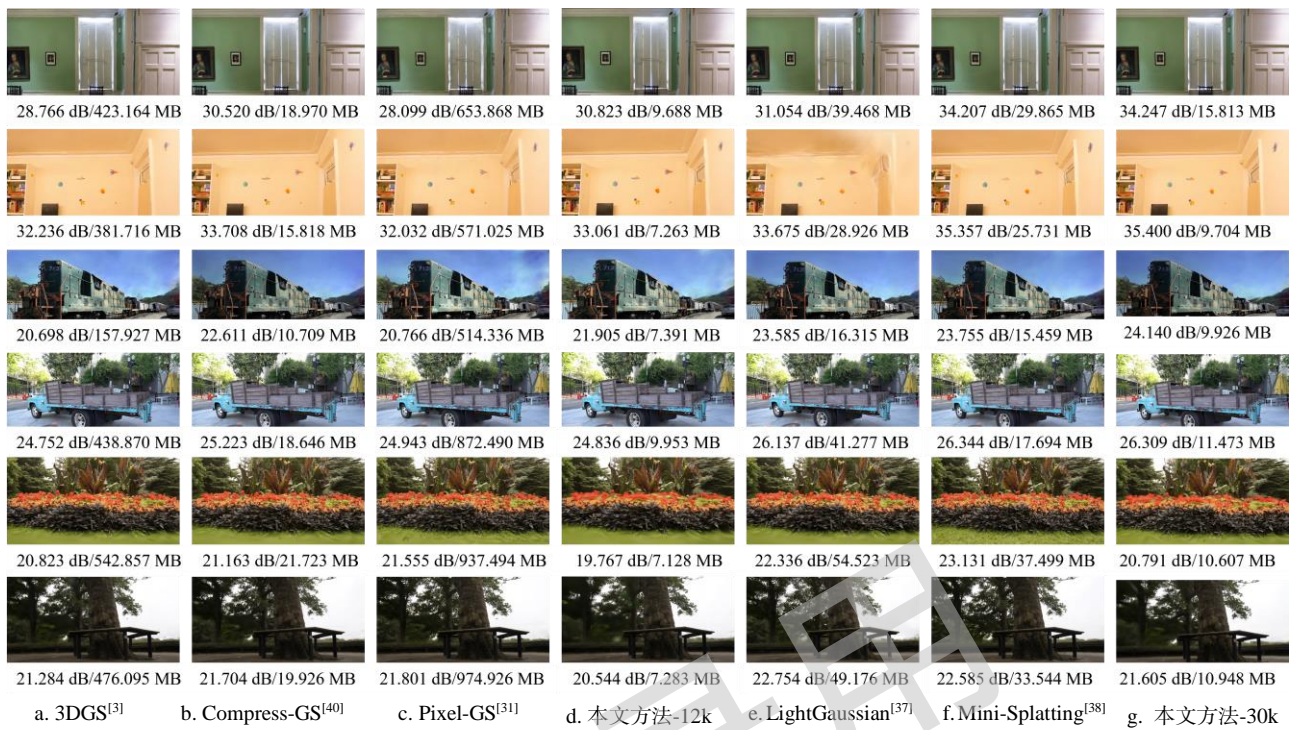


图 3 中大型场景在 6 种模型的可视化结果

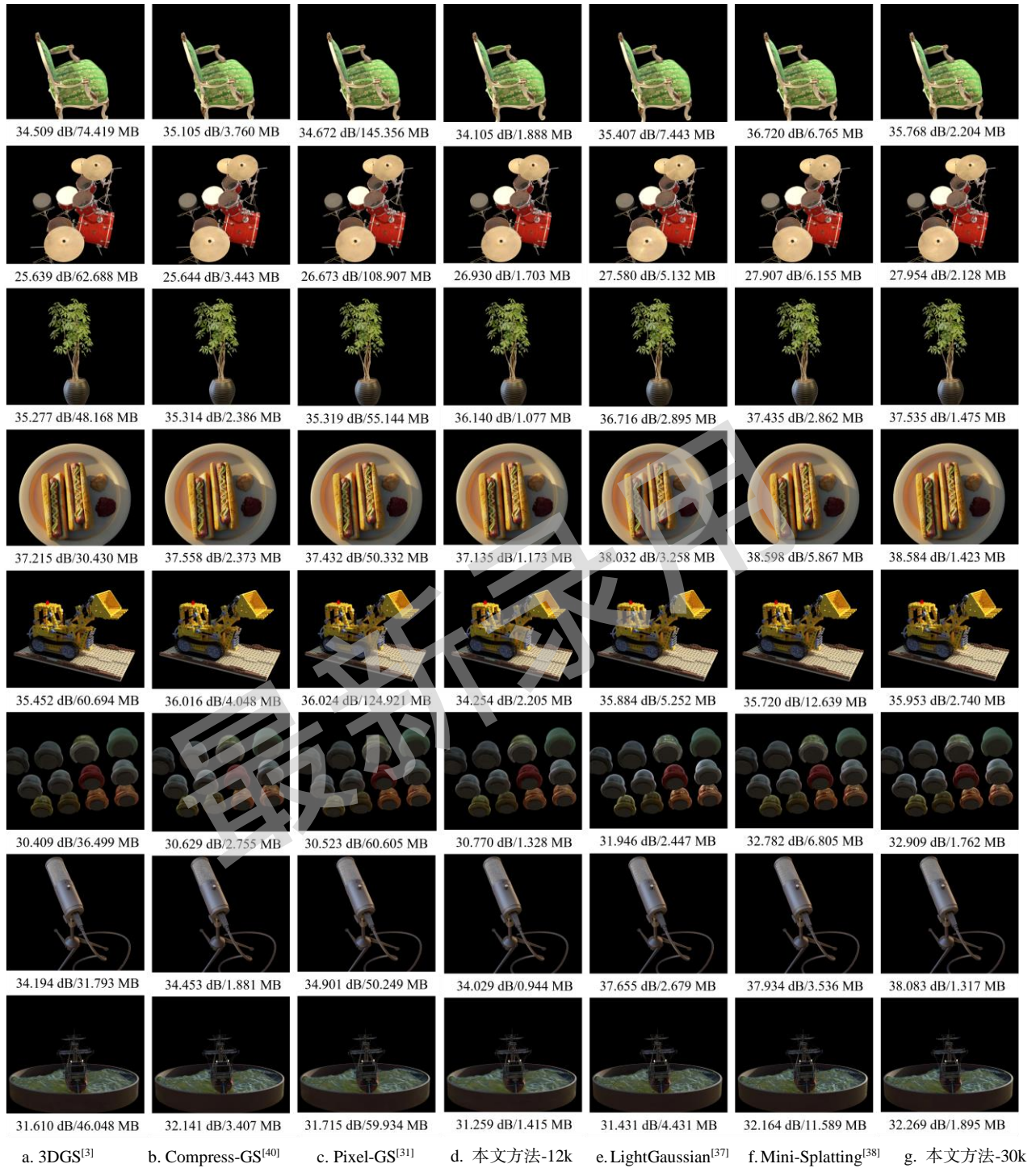


图4 小场景在6种模型的可视化结果

3.4 消融实验

为了验证本文方法的有效性,以 NeRF Synthetic 数据集^[2]上 8 个场景的平均值和 Tanks&Temples 数据集^[43]上 2 个场景的平均值为例,在 3DGS^[3]和 Compress-GS^[40]的基础上设计消融实验,结果如表 5 所示.其中,3DGS^[3]是原始的模型,不包含任何压缩机制;Compress-GS^[40]是在 3DGS^[3]的基础上,对 C 和形状参数进行压缩,并

使用熵编码存储;本文方法-1 是在 3DGS^[3]的基础上修改自适应密度控制策略;本文方法-2 是在 Compress-GS^[40]的基础上,删除低 α 的高斯基元,然后对剩余的 α 进行敏感性感知和向量聚类,并将本文方法-1 的结果进行压缩的模型;本文方法-3 是在本文方法-2 的基础上,使用局部邻近搜索,在密集度高的区域,将综合相似度高的小高斯合并成一个大高斯,进一步压缩的模型.

表 5 在 NeRF Synthetic 和 Tanks&Temples 数据集上的消融实验结果

模型	NeRF Synthetic ^[2]					Tanks&Temples ^[43]				
	ply-Size/MB	PSNR/dB	SSIM	LPIPS	耗时/min	ply-Size/MB	PSNR/dB	SSIM	LPIPS	耗时/min
3DGS ^[3]	48.842	33.038	0.969	0.038	1:14	298.399	22.725	0.797	0.258	2:35
Compress-GS ^[40] (npz-Size)	48.193 (3.007)	33.358	0.973	0.035	3:50	295.826 (14.678)	23.917	0.829	0.230	4:32
本文方法-1	17.087	32.470	0.963	0.049	1:14	125.299	22.295	0.784	0.278	2:31
本文方法-2 (npz-Size)	12.416 (1.511)	33.085	0.966	0.045	3:07	101.822 (8.682)	23.427	0.811	0.257	5:06
本文方法-3 (npz-Size)	11.911 (1.467)	33.078	0.966	0.045	4:39	99.819 (8.672)	23.371	0.828	0.259	5:54

从表 5 可以看出:

(1) 在 NeRF Synthetic 数据集^[2]的场景中,与 3DGS^[3]相比,本文方法-1 的存储空间平均减少 65.02%, PSNR 平均降低 0.568 dB, 本文方法-2 的 PSNR 平均提升 0.047 dB; 与 Compress-GS^[40]相比,本文方法-2 的 ply 文件格式的存储空间平均减少 74.24%, npz 文件格式的存储空间平均减少 49.75%, PSNR 平均降低 0.273 dB.

(2) 在 Tanks&Temples 数据集^[43]的场景中,与 3DGS^[3]相比,本文方法-1 的存储空间平均减少 58.01%, PSNR 平均降低 0.430 dB, 本文方法-2 的 PSNR 平均提升 0.702 dB; 与 Compress-GS^[40]相比,本文方法-2 的 ply 文件格式的存储空间平均减少 65.58%, npz 文件格式的存储空间平均减少 40.85%, PSNR 平均降低 0.490 dB.

(3) 在本文方法-2 的基础上,本文方法-3 的合并进一步减小存储空间的同时, PSNR 几乎没有差异.

3.5 局限性

尽管本文方法在 NeRF Synthetic 数据集^[2]小场景中表现出色,但在中大型场景中处理 ply 文件格式时的效果不是最佳,这是由于中大型场景的结构复杂性显著,其中包含多层次的细节和广泛的空间分布,因此在不牺牲图像重建质量的前提下实现 ply 文件的极致压缩成为一项挑战. 对这些复杂场景进行有效的数据压缩,需要对场景的空间属性和结构特征有深刻的理解和精确的处理策略. 未来,应当重点关注如何结合中大型场景的深度信息,对高斯致密化过程进行精细控制,达到更高的数据压缩效率,同时保持图像质量.

4 结语

本文提出一种改进的对 3DGS 的数据进行压缩的方法,通过修改自适应密度控制策略,删除极低 α 的三维高斯基元并压缩剩余的三维高斯基元的 α , 以及在高密度区域合并高相似性的小三维高斯基元,极大降低了重建场景占用的存储空间. 实验结果表明,在大多数场景中,与 3DGS 模型相比,本文方法在不牺牲性能的情况下,存储空间平均减少 97.64%, PSNR 平均提升 0.307 dB, 为进一步压缩 3DGS 提供了新的解决思路. 如何使用更小的存储空间得到更高的更快的渲染质量,将是一个值得深入研究的问题,将有助于 3DGS 在移动设备上的应用.

参考文献(References):

- [1] Goesele M, Snavely N, Curless B, *et al.* Multi-view stereo for community photo collections[C] //Proceedings of the 11th IEEE International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2007: 1-8
- [2] Mildenhall B, Srinivasan P P, Tancik M, *et al.* NeRF: representing scenes as neural radiance fields for view synthesis[J]. Communications of the ACM, 2021, 65(1): 99-106
- [3] Kerbl B, Kopanas G, Leimkuehler T, *et al.* 3D Gaussian splatting for real-time radiance field rendering[J]. ACM Transactions on Graphics, 2023, 42(4): Article No.139
- [4] Duan Y X, Wei F Y, Dai Q Y, *et al.* 4D-rotor Gaussian splatting: towards efficient novel view synthesis for dynamic scenes[C] //Proceedings of the ACM SIGGRAPH Conference Papers. New York: ACM Press, 2024: Article No.87
- [5] Sun J K, Jiao H, Li G Y, *et al.* 3DGSStream: on-the-fly training of 3D Gaussians for efficient streaming of photo-realistic free-viewpoint videos[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2024: 20675-20685
- [6] Wu G J, Yi T R, Fang J M, *et al.* 4D Gaussian splatting for real-time dynamic scene rendering[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern

- Recognition. Los Alamitos: IEEE Computer Society Press, 2024: 20310-20320
- [7] Yu H, Julin J, Milacski Z Á, *et al.* CoGS: controllable Gaussian splatting[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2024: 21624-21633
- [8] Zhou X Y, Lin Z W, Shan X J, *et al.* DrivingGaussian: composite Gaussian splatting for surrounding dynamic autonomous driving scenes[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2024: 21634-21643
- [9] Deng T C, Chen Y H, Zhang L Y, *et al.* Compact 3D Gaussian splatting for dense visual slam[OL]. [2024-09-28]. <https://arxiv.org/abs/2403.11247>
- [10] Hu J R, Chen X H, Feng B Y, *et al.* CG-SLAM: efficient dense RGB-D SLAM in a consistent uncertainty-aware 3D Gaussian field[C] //Proceedings of the 18th European Conference on Computer Vision. Heidelberg: Springer, 2025: 93-112
- [11] Yan C, Qu D L, Xu D, *et al.* GS-SLAM: dense visual SLAM with 3D Gaussian splatting[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2024: 19595-19604
- [12] Huang Y H, Sun Y T, Yang Z Y, *et al.* SC-GS: sparse-controlled Gaussian splatting for editable dynamic scenes[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2024: 4220-4230
- [13] Chen Z L, Wang F, Wang Y K, *et al.* Text-to-3D using Gaussian splatting[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2024: 21401-21412
- [14] Zielonka W, Bagautdinov T, Saito S, *et al.* Drivable 3D Gaussian avatars[OL]. [2024-09-28]. <https://arxiv.org/abs/2311.08581>
- [15] Wu T, Yuan Y J, Zhang L X, *et al.* Recent advances in 3D Gaussian splatting[J]. *Computational Visual Media*, 2024, 10(4): 613-642
- [16] Cotton R J, Peyton C. Dynamic Gaussian splatting from markerless motion capture reconstruct infants movements[C] //Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops. Los Alamitos: IEEE Computer Society Press, 2024: 60-68
- [17] Mildenhall B, Hedman P, Martin-Brualla R, *et al.* NeRF in the dark: high dynamic range view synthesis from noisy raw images[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2022: 16190-16199
- [18] Tancik M, Casser V, Yan X C, *et al.* Block-NeRF: scalable large scene neural view synthesis[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2022: 8248-8258
- [19] Gao C, Saraf A, Kopf J, *et al.* Dynamic view synthesis from dynamic monocular video[C] //Proceedings of the IEEE/CVF International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2021: 5712-5721
- [20] Li T Y, Slavcheva M, Zollhoefer M, *et al.* Neural 3D video synthesis from multi-view video[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2022: 5521-5531
- [21] Xu Q G, Xu Z X, Philip J, *et al.* Point-NeRF: point-based neural radiance fields[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2022: 5438-5448
- [22] Hedman P, Srinivasan P P, Mildenhall B, *et al.* Baking neural radiance fields for real-time view synthesis[C] //Proceedings of the IEEE/CVF International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2021: 5875-5884
- [23] Chen A P, Xu Z X, Geiger A, *et al.* TensorRF: tensorial radiance fields[C] //Proceedings of the 17th European Conference on Computer Vision. Heidelberg: Springer, 2022: 333-350
- [24] Tang J X, Chen X K, Wang J B, *et al.* Compressible-composable NeRF via rank-residual decomposition[C] //Proceedings of the 36th International Conference on Neural Information Processing Systems. Red Hook: Curran Associates Inc, 2022: Article No.1076
- [25] Deng C L, Tartaglione E. Compressing explicit voxel grid representations: fast NeRFs become also small[C] //Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. Los Alamitos: IEEE Computer Society Press, 2023: 1236-1245
- [26] Li L Z, Shen Z, Wang Z S, *et al.* Compressing volumetric radiance fields to 1 MB[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2023: 4222-4231
- [27] Takikawa T, Evans A, Tremblay J, *et al.* Variable bitrate neural fields[C] //Proceedings of the ACM SIGGRAPH Conference Proceedings. New York: ACM Press, 2022: Article No.41
- [28] Chan E R, Lin C Z, Chan M A, *et al.* Efficient geometry-aware 3D generative adversarial networks[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2022: 16123-16133
- [29] Müller T, Evans A, Schied C, *et al.* Instant neural graphics primitives with a multiresolution hash encoding[J]. *ACM Transactions on Graphics*, 2022, 41(4): Article No.102
- [30] Ye Z X, Li W Y, Liu S D, *et al.* AbsGS: recovering fine details in 3D Gaussian splatting[C] //Proceedings of the 32nd ACM International Conference on Multimedia. New York: ACM Press, 2024: 1053-1061
- [31] Zhang Z, Hu W B, Lao Y X, *et al.* Pixel-GS: density control with pixel-aware gradient for 3D Gaussian splatting[C] //Proceedings of the 18th European Conference on Computer Vision. Heidelberg: Springer, 2025: 326-342
- [32] Bagdasarian M T, Knoll P, Li Y H, *et al.* 3DGS.zip: a survey on 3D Gaussian splatting compression methods[OL]. [2024-09-28]. <https://arxiv.org/abs/2407.09510>
- [33] Shi Y, Gasparini S, Morin G, *et al.* LapisGS: layered progressive 3D Gaussian splatting for adaptive streaming[OL]. [2024-09-28]. <https://arxiv.org/abs/2408.14823>
- [34] Navaneet K L, Meibodi K P, Koohpayegani S A, *et al.* CompGS: smaller and faster Gaussian splatting with vector quantization[OL]. [2024-09-28]. <https://arxiv.org/abs/2311.18159>
- [35] Lu T, Yu M L, Xu L N, *et al.* Scaffold-GS: structured 3D Gaussians for view-adaptive rendering[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2024: 20654-20664
- [36] Chen Y H, Wu Q Y, Lin W Y, *et al.* HAC: hash-grid assisted context for 3D Gaussian splatting compression[C] //Proceedings of the 18th European Conference on Computer Vision. Heidelberg: Springer, 2025: 422-438
- [37] Fan Z W, Wang K, Wen K, *et al.* LightGaussian: unbounded 3D Gaussian compression with 15x reduction and 200+ FPS[OL]. [2024-09-28]. <https://arxiv.org/abs/2311.17245>
- [38] Fang G C, Wang B. Mini-Splatting: representing scenes with a constrained number of Gaussians[C] //Proceedings of the 18th European Conference on Computer Vision. Heidelberg: Springer, 2024: 165-181
- [39] Lee J C, Rho D, Sun X Y, *et al.* Compact 3D Gaussian representation for radiance field[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2024: 21719-21728
- [40] Niedermayr S, Stumpfegger J, Westermann R. Compressed 3D Gaussian splatting for accelerated novel view synthesis[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2024: 10349-10358
- [41] Salomon D, Motta G. Handbook of data compression[M]. London: Springer, 2010
- [42] Barron J T, Mildenhall B, Verbin D, *et al.* Mip-NeRF 360: un-

- bounded anti-aliased neural radiance fields[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2022: 5470-5479
- [43] Knapitsch A, Park J, Zhou Q Y, *et al.* Tanks and temples: benchmarking large-scale scene reconstruction[J]. ACM Transactions on Graphics, 2017, 36(4): Article No.78
- [44] Hedman P, Philip J, Price T, *et al.* Deep blending for free-viewpoint image-based rendering[J]. ACM Transactions on Graphics, 2018, 37(6): Article No.257
- [45] Zhang R, Isola P, Efros A A, *et al.* The unreasonable effectiveness of deep features as a perceptual metric[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2018: 586-595

最新录用