

HAC: Hash-grid Assisted Context for 3D Gaussian Splatting Compression 论文复现

摘要

3DGS (3D Gaussian Splatting) 是一种有前景的新型视图合成框架, 以其快速的渲染速度和高保真度受到了广泛关注。然而, 对于一个场景, 需要 600MB-1.2GB 的容量去存储 3DGS 的高斯点和它们相关的属性, 因此需要有效的压缩技术或者紧致性表示技术来便于 3DGS 在实际工程应用中的存储与传输。Scaffold-GS 通过引入了锚点, 对相关高斯进行聚类 and 预测高斯球属性, 而不是直接存储高斯球的属性, 能有效压缩 3DGS。HAC 在 Scaffold-GS 的基础上, 进一步探索锚点之间的空间联系, 以进一步压缩 3DGS。为了方便日后的学习研究, 本文根据其开源代码进行复现, 并添加轨迹渲染功能以可视化形式展示 HAC 在率失真的优越性, 得到了和 HAC 论文相近的结果。

关键词: 3DGS; 锚点; 压缩; 率失真;

1 引言

本人的方向是 3DGS (3D Gaussian Splatting) 的紧致性表示, HAC 通过对率失真的研究, 在压缩率和失真之间权衡, 使得最终结果不仅在压缩性能有很好表现, 在保真度也有较高的视觉效果。HAC 可以被认为提供一种框架, 通过使用结构化的表示形式去表示非结构化, 混乱的点云形式, 从中能更好的发现其中冗余信息, 进行进一步压缩。复现这篇论文的代码对我理解和研究 3DGS 压缩和紧致性表示将会有很大的帮助。

2 相关工作

2.1 Nerf 及其压缩

NeRF (Neural Radiance Fields) [7] 是一种用于 3D 场景重建和视图合成的技术, 它通过一个单一的可学习隐式多层感知器 (MLP) 模型, 在三维空间中沿射线进行 α 合成, 将 RGB 值累积到一起, 从而表现场景的光照和物体的透明度特征并从不同视角生成场景的真实图像。然而, NeRF 的一个关键问题是需要对场景中的大量采样点进行密集查询, 同时 MLP 的大小也相对较大, 这使得其渲染速度较慢, 难以满足实时渲染的要求。

为了有效缓解存储开销, 研究者们提出了多种压缩技术, 旨在减少显式表示所需的存储空间。这些压缩方法可以分为两大类: 基于“值”的压缩方法和基于结构关系的压缩方法。基于“值”的方法主要通过减少数据冗余来减小存储需求, 例如剪枝 [5]、量化、码本和熵约束 [3] 等技术。通过这些方法, 模型的存储需求得以压缩, 同时尽量保留关键信息。另一方面, 基于

结构关系的方法则更注重利用场景数据中的内在结构和空间规律来消除冗余。这类方法包括小波分解、秩残差分解和空间预测等，通过对网格数据的结构化特性进行处理，进一步降低了存储量。是 CNC [1] 方法，通过巧妙地利用这些结构信息，取得了显著的压缩效果，在率失真（RD）性能上获得了显著提升。

2.2 3DGS 及其压缩

3DGS [4] 是一种创新的三维场景表示与渲染技术，它通过可学习的高斯分布直接显式地建模三维空间。这些高斯分布从运动结构恢复（SfM）初始化，并具备可优化的形状和外观参数，能够高效地投影到二维平面上，利用基于瓦片的光栅化实现快速且可微的渲染。与传统的 NeRF 等基于体渲染的方法相比，3DGS 完全避免了耗时的体积采样过程，同时保持了高保真度的视觉效果。然而，由于高斯分布及其相关属性需要大量存储空间，因此有效的压缩技术成为了一个关键问题。

与 NeRF 基于特征网格的表示方法不同，3DGS 中的 3D 高斯分布是稀疏且无序的，这使得建立有效的结构关系变得更加困难。因此，针对 3DGS 的压缩方法主要集中在模型参数的“值”上，采用了如剪枝、代码本和熵约束等技术，以减小存储空间。例如，剪枝方法通过去除不必要的高斯分布来减少模型复杂度，码本技术则通过量化高斯分布的参数来减少冗余信息，熵约束则通过优化信息编码的方式进一步压缩数据。这些方法能够显著降低存储需求，同时尽量保留场景的重要信息。此外，尽管大多数压缩方法侧重于单独压缩高斯分布的参数，少数研究开始探索高斯分布之间的关系。比如，Scaffold-GS [6] 方法通过引入锚点中心特征来减少参数数量，这种方法有助于进一步优化高斯分布的表示。在另一项研究中，Morgenstern [8] 等人提出通过降维技术将高斯分布压缩到有序的二维空间中，从而实现更高效的存储和计算。

3 本文方法

3.1 本文方法概述

HAC [2] 的模型框架如图 1 所示，图的上面部分是 Scaffold-GS 的框架图，通过锚点的属性来聚类 and 预测原始高斯球的属性，避免直接存储。图的中间部分是 HAC 提出的二进制可学习的哈希网格，通过锚点的位置信息在哈希网格查询可得到插值的哈希特征。图的下面部分是哈希特征通过 MLP 一方面输出高斯分布的均值和方差，用于对锚点属性的值分布进行建模，从而可以为算术编码计算每个量化属性值的概率；另一方面输出量化步长的修改值，设计自适应量化模块，将锚点属性值量化为合适的有限集。此外，采用了自适应偏移掩蔽模块（图左上角）来修剪多余的高斯和锚点。总体来说，HAC 希望通过哈希特征的辅助来最小化锚点属性的熵，从而在使用像算术编码这样的熵编码对锚点属性进行编码时促进比特减少。

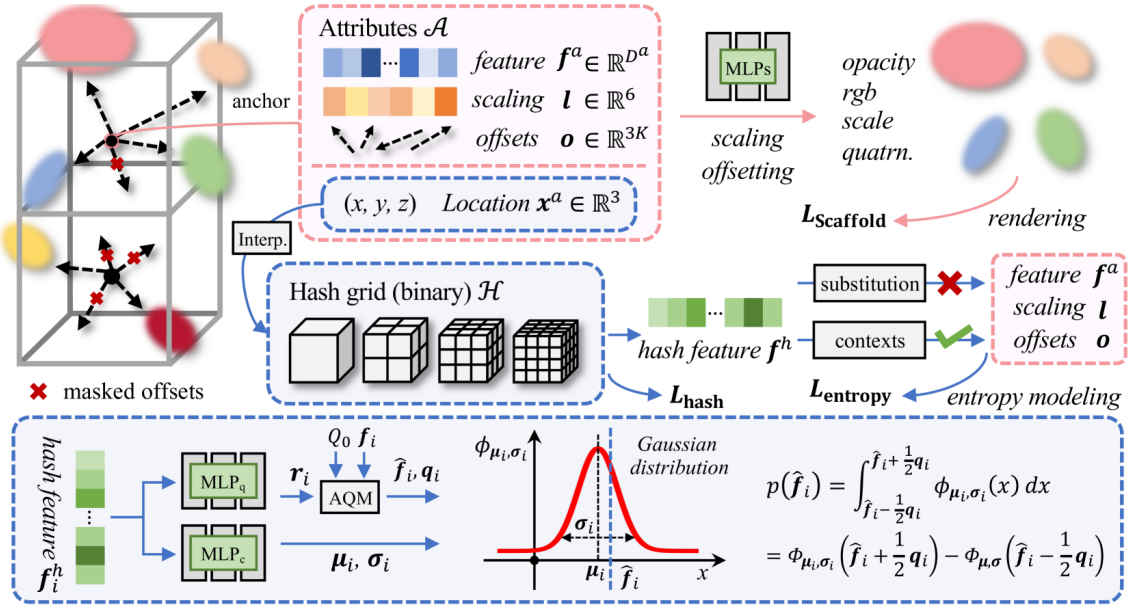


图 1. 方法示意图

3.2 自适应量化模块

因为锚点属性的取值范围各不相同，锚点属性的缩放参数和偏移量通常是很小的值，使用传统的四舍五入量化方法不合适，不能充分保留数据的细节信息，因此引入自适应量化模块（AQM）。AQM 通过输入哈希特征，经过得到量化调整值 r ，后通过对预设量化步长 Q_0 计算公式 $Q_0 \times (1 + \text{Tanh}(r))$ 得到最终量化步长 q 。

在训练阶段，AQM 通过在量化过程中添加噪声来模拟量化效果，同时确保梯度能够通过直通估计（Straight-Through Estimator, STE）进行反向传播。在测试阶段，AQM 则使用舍入操作来实现实际的量化。这种设计使得 AQM 能够在保持高保真度的同时，实现高效的量化处理。

3.3 高斯分布模块

由于在量化后我们需要对特征值进行编码，而这种编码需要通过概率模型来量化其信息量，因此在训练过程中，必须以可微分的方式估算每个量化特征的概率。每个属性之间是独立的，并为每个属性构建独立的高斯分布模型。在这种模型中，均值和标准差是通过一个简单的基于 MLP 的上下文模型从特征哈希中预测的。通过高斯分布的累积分布函数来计算量化特征的概率，具体计算公式如下：

$$\begin{aligned}
 p(\hat{f}_i) &= \int_{\hat{f}_i - \frac{1}{2}q_i}^{\hat{f}_i + \frac{1}{2}q_i} \phi_{\mu_i, \sigma_i}(x) dx \\
 &= \Phi_{\mu_i, \sigma_i}\left(\hat{f}_i + \frac{1}{2}q_i\right) - \Phi_{\mu_i, \sigma_i}\left(\hat{f}_i - \frac{1}{2}q_i\right)
 \end{aligned}$$

3.4 损失函数定义

$$\text{LOSS} = L_{\text{Scaffold}} + \lambda_e \frac{1}{N(D^a + 6 + 3K)} (L_{\text{entropy}} + L_{\text{hash}}) + \lambda_m L_m$$

$L_{Scaffold}$ 代表渲染损失，它包括两个保真度惩罚项和一个用于缩放因子正则化的项，第二部分则是对可控比特消耗的估计，包括两个部分： $L_{entropy}$ 是锚点属性的比特消耗损失，而 L_{hash} 则与哈希网络的比特消耗相关。这部分损失有助于控制模型在压缩过程中的比特率，同时保持较好的渲染质量。 L_m 是掩码损失（masking loss），用于正则化自适应偏移掩码模块，以进一步稳定和优化模型的训练。 λ_e 和 λ_m 是超参数，用于在不同损失项之间进行权衡，以平衡渲染质量和比特消耗的控制，同时保证模型在训练过程中的稳定性。

4 复现细节

4.1 与已有开源代码对比

本次复现中，我直接使用了 HAC 的渲染模块的 cuda 代码，在参考官方源码的基础上，根据自己的学习和理解重写了 pytorch 版本的代码，代码更加简单易懂，便于后续的学习和改进，并且添加了对鱼眼镜头照片的功能支持，添加了轨迹化渲染和生成演示视频的功能支持。

4.2 实验环境搭建

1. 根据 requirements.txt 中的创建 conda 环境，其中的 pytorch 版本需要根据服务器的 cuda 版本调整
2. 安装 ffmpeg 用于对拍摄视频进行提取照片和最后根据渲染图片合成视频
3. 安装 colmap 对提取的照片进行提取 sfm 数据集
4. 运行 train.py 并配置相关参数即可

5 实验结果分析

拍摄了会议室的视频，并使用 ffmpeg 和 colmap 构建输入数据集，输入模型训练，训练结果如图 2 所示，训练 30k 后测试的 psnr 为 28.57908。对比使用 3DGS 训练后场景的存储大小，如图 3, 4，可以看出确实与论文宣传的一样，能够对一个场景的存储容量压缩超过 75x 倍。选取其中一个视角下的图片，如图 5, 6, 7 对比渲染图片和真实图片，压缩之后确实有一个精度损失。根据添加的 3D 圆球轨迹渲染功能，生成一段演示视频可在辅助材料查看。

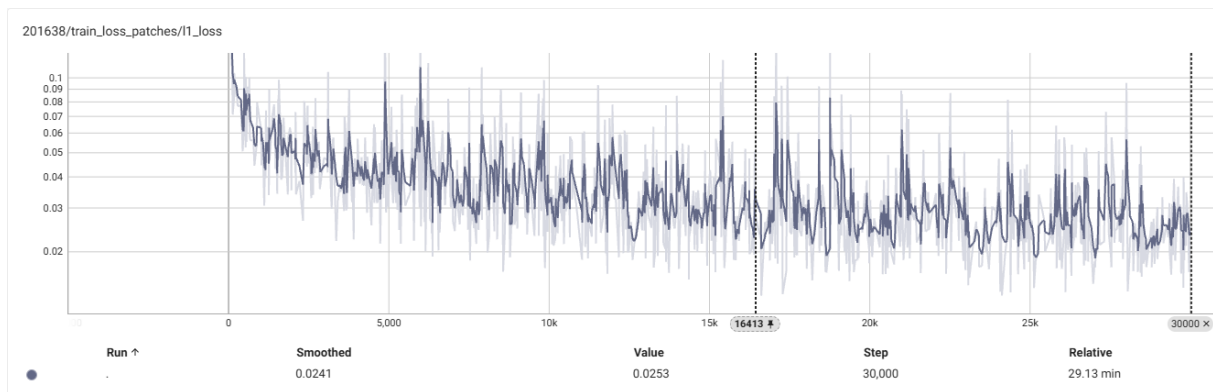


图 2. 实验结果示意图



图 3. 3DGS 存储大小

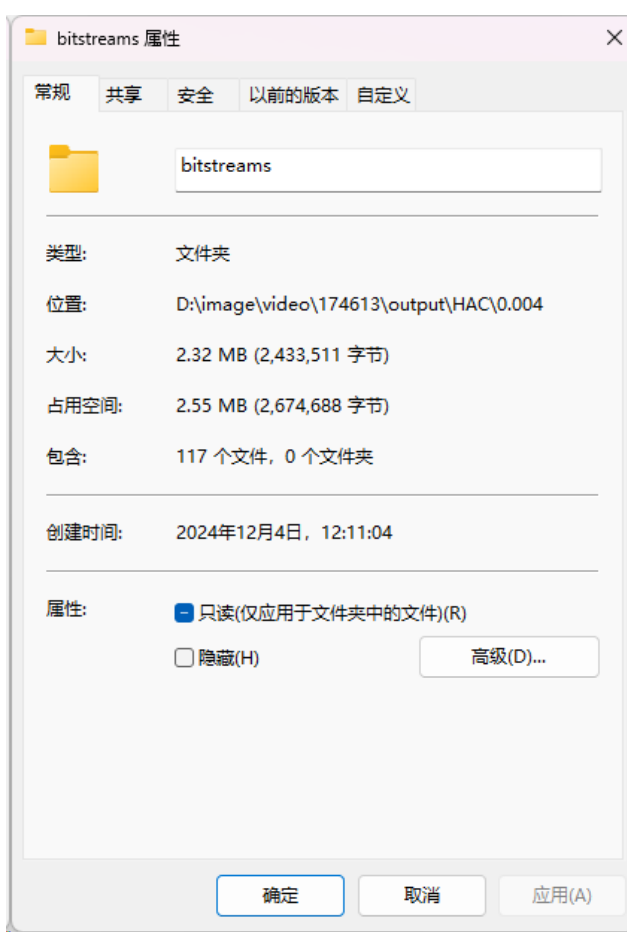


图 4. HAC 存储大小

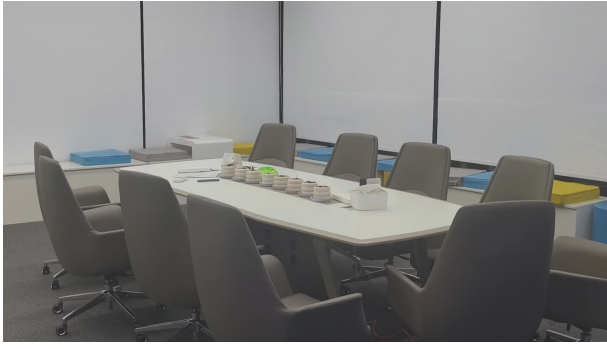


图 5. gt

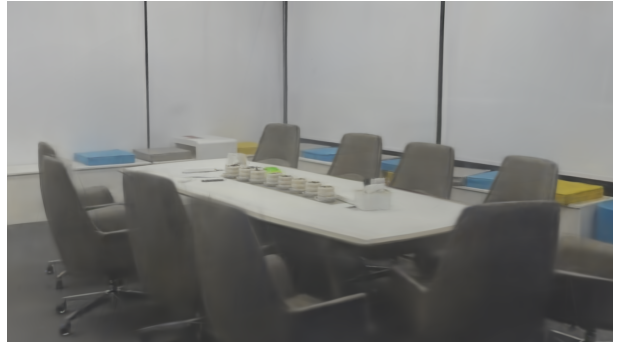


图 6. render



图 7. error

6 总结与展望

本文对 HAC 进行了详细的介绍和复现，在参考官方代码和使用 HAC 的渲染模块的情况下实现了和 HAC 宣称的效果，并且添加了对特殊轨迹的渲染和视频合成功能，能更好的展示 HAC 在压缩下，也能表现较好的保真度。由于时间和能力有限未能取得很大的改进，但是在这个过程中我系统前面的学习和理解了 HAC 的原理和实现，为之后的学习和研究打下良好的基础。未来，我准备将 3D 空间中的锚点信息，映射到 2D 图像中，并利用 2D 图像的先进压缩技术对 3DGS 进行进一步的压缩和研究。

参考文献

- [1] Yihang Chen, Qianyi Wu, Mehrtash Harandi, and Jianfei Cai. How far can we compress instant-ngp-based nerf? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

- [2] Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. Hac: Hash-grid assisted context for 3d gaussian splatting compression. In *European Conference on Computer Vision*, 2024.
- [3] Sharath Girish, Abhinav Shrivastava, and Kamal Gupta. Shacira: Scalable hash-grid compression for implicit neural representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17513–17524, October 2023.
- [4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.
- [5] Lingzhi Li, Zhen Shen, Zhongshu Wang, Li Shen, and Liefeng Bo. Compressing volumetric radiance fields to 1 mb. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4222–4231, 2023.
- [6] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024.
- [7] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.
- [8] Wieland Morgenstern, Florian Barthel, Anna Hilsmann, and Peter Eisert. Compact 3d scene representation via self-organizing gaussian grids. In *Computer Vision – ECCV 2024*, pages 18–34, Cham, 2025. Springer Nature Switzerland.