

基于Unity3D的动态地形电子沙盘实时渲染优化算法的研究

王雨良¹ 张全信² 范金鹏²

1. 杭州耐信模型有限公司 浙江 杭州 310000

2. 杭州知管家信息科技有限公司 浙江 杭州 310000

摘要: 随着计算机图形学和虚拟现实技术飞速发展, 基于Unity3D的动态地形电子沙盘在城市规划、军事模拟、文旅展示等领域应用日益广泛。它以直观交互的特性, 为用户呈现逼真的地理场景, 成为各行业重要的可视化工具。然而, 当前动态地形电子沙盘的实时渲染面临诸多挑战。大规模地形数据的处理效率低下, 加载时间长, 严重影响使用体验。同时, 渲染过程对硬件性能要求苛刻, 普通设备难以负荷, 容易造成画面卡顿、延迟, 导致场景展示不流畅, 限制了其更广泛的应用。基于此, 本文针对基于Unity3D的动态地形电子沙盘实时渲染展开全面分析。从数据处理、渲染算法等多方面入手, 通过创新优化策略, 减少数据冗余, 提升渲染效率。以期降低硬件依赖, 实现在普通配置设备上也能稳定、流畅地进行动态地形实时渲染, 推动动态地形电子沙盘在更多领域深度应用, 为行业发展提供有力支撑。

关键词: Unity3D; 动态地形电子沙盘; 实时渲染优化; 虚拟现实技术

1 引言

在数字化与智能化深度融合的时代, 地理信息可视化技术成为诸多领域发展的关键驱动力。基于Unity3D的动态地形电子沙盘, 凭借其出色的交互性与逼真的场景模拟能力, 在城市规划、文旅展示、军事战略推演等场景中广泛应用, 为用户提供了沉浸式的地理空间体验。但在实际应用中, 动态地形电子沙盘的实时渲染面临严峻挑战。一方面, 海量地形数据的处理和加载效率低下, 导致启动与运行缓慢; 另一方面, 现有渲染算法难以兼顾复杂场景的精细描绘与硬件性能限制, 画面卡顿、延迟频繁出现, 严重制约了其应用效果与范围拓展。在此背景下, 对基于Unity3D的动态地形电子沙盘实时渲染优化算法展开研究, 突破技术瓶颈, 对推动相关行业发展、提升用户体验有着重要的现实意义。

2 相关技术基础

2.1 Unity3D引擎核心特性

Unity3D是跨平台游戏开发引擎, 在动态地形电子沙盘构建中作用关键。它的渲染管线可定制, 开发者能按项目需求调整渲染流程。像光照处理, 支持实时、烘焙和混合光照模式, 营造逼真光影, 让地形场景更真实。资源管理上, 采用先进内存管理机制, 自动跟踪回收不用资源, 降低内存占用, 保障系统稳定。同时, 它跨平台兼容性强, 能轻松把开发好的电子沙盘部署到Windows、iOS、Android等多平台, 极大拓展应用范围,

满足不同用户需求^[1]。

2.2 动态地形构建原理

动态地形构建是电子沙盘基础。先把数字化高程数据, 比如卫星遥感地形信息, 转化成计算机可处理格式。接着用网格生成算法, 将数据构建成规则或不规则三角形网格, 模拟地形起伏。构建时, 为提升效率和真实感, 引入LOD技术, 依观察距离动态调整地形网格精细度, 近景高精度、远景低精度, 兼顾视觉效果与渲染负担。还通过纹理映射技术, 把草地、岩石等材质纹理贴合到地形表面, 增强真实感和视觉表现力。

2.3 实时渲染基础算法

实时渲染对动态地形电子沙盘即时画面呈现很关键。光线投射算法从视点发射光线, 与场景物体相交测试, 确定物体可见性和光照效果, 实现真实阴影和反射。深度缓冲算法记录像素深度信息, 渲染时比较同一像素位置不同物体深度, 避免遮挡错误, 确保画面正确显示。三角形光栅化算法把三角形顶点转化为屏幕像素, 完成几何模型到图像的转换, 为光照计算和颜色填充打基础, 这些算法共同支撑实时渲染流程。

3 地形数据处理优化

3.1 数据组织策略改进

传统地形数据常以简单的顺序结构存储, 查询与调用效率低下。本文提出基于四叉树的数据组织策略。将地形区域递归划分为四个子区域, 每个子区域对应四叉

树的一个节点。在节点中存储地形数据索引，大幅提升数据定位速度。比如在大规模山区地形数据处理时，通过四叉树结构，可快速定位到特定山峰区域数据，无需遍历整个数据集。同时，结合哈希表优化节点索引，利用哈希函数将地形特征值映射为唯一标识，使数据查找复杂度从 $O(n)$ 降低至接近 $O(1)$ ，有效减少数据处理时间，为快速渲染提供保障^[2]。

3.2 数据压缩与解压优化

地形数据量庞大，传输与存储成本高。采用基于小波变换的压缩算法，它能将地形数据从空间域转换到频率域，通过去除高频部分的冗余信息实现压缩。在解压时，保留的低频信息可快速恢复地形大致轮廓，高频信息逐步补充细节，兼顾解压速度与地形精度。实验表明，该算法压缩比可达10:1 - 20:1，且解压后地形误差在可接受范围内。此外，针对不同地形特征，动态调整压缩参数，对平坦区域采用较高压缩比，复杂区域适度降低，在保证数据质量的同时，最大化减少数据存储空间与传输带宽需求。如表1所示：

表1 地形数据压缩算法性能对比

算法类型	压缩比(倍)	解压时间(ms)	最大误差(m)	适用场景
小波变换(本文)	15:1	22.5	0.12	复杂地形
DCT压缩	8:1	35.8	0.25	平坦区域
游程编码	5:1	12.1	0.0(无损)	重复性高地形
LZ77	10:1	28.3	0.18	通用地形

数据说明：

- 测试数据：100km²山地地形（高程分辨率1m，原始数据量1.2GB）
- 误差标准：基于均方根误差（RMSE）计算

3.3 数据流式加载机制

为避免一次性加载大量地形数据导致内存溢出与启动延迟，构建数据流式加载机制。依据视点位置与视锥范围，实时预测需要加载的地形数据块。利用多线程技术，在后台并行加载数据，当数据进入视锥前完成加载并缓存。例如在飞行模拟场景中，飞机快速移动时，流式加载机制能提前加载前方地形数据，保证画面流畅，不会因数据加载不及时出现空白或卡顿。同时，结合缓存管理策略，对已加载但暂时不用的数据进行合理缓存，当再次需要时可快速调用，提升数据利用效率，优化整体渲染性能。

4 渲染算法优化策略

4.1 视锥体裁剪优化

视锥体裁剪对渲染效率提升很关键。传统算法仅简单判断物体是否在视锥体内，存在不足。本文提出基于层次包围盒的优化方法，先为地形场景物体构建从大到小的层次包围盒。进行视锥体裁剪时，先判断最外层包围盒和视锥体关系，若完全在视锥体外，直接剔除及其包含的物体，无需精细检测；若部分相交，再深入下一层级判断。例如在广阔山地场景中，对远处不在视锥体内的大片山体，利用外层包围盒快速剔除，减少渲染计算，提升渲染帧率，让画面更流畅。

4.2 层次细节模型（LOD）优化

LOD技术可依观察距离动态调整模型精细度，但现有LOD切换易出现视觉突变。本文采用基于屏幕空间误差的优化策略，实时计算模型屏幕投影误差值，超阈值时自动切换LOD模型。切换时引入过渡算法，对相邻LOD模型顶点插值处理，使过渡平滑^[4]。如用户靠近建筑时，模型从低精度LOD渐变为高精度LOD，插值处理让细节增加是渐进式的，避免突然变化，提供舒适真实视觉体验，合理利用硬件资源，保证渲染性能稳定。

4.3 光照与阴影优化算法

光照和阴影影响场景真实感，可传统算法计算开销大。本文提出基于预计算辐射传递函数（PRT）的光照优化算法，提前计算并存储光照辐射传递信息，实时渲染时直接查询，减少光照计算时间。阴影部分采用级联阴影映射（CSM）算法并结合硬件加速。按物体与相机距离划分不同层级阴影映射纹理，近处用高精度纹理，远处用低精度纹理，保证质量同时降低计算量。利用GPU并行计算能力进行硬件加速，提升阴影计算速度，使光照和阴影效果逼真又高效，增强动态地形电子沙盘沉浸感。

5 性能评估与测试

5.1 性能评估指标设定

为全面衡量优化算法效果，设立多维度性能评估指标。帧率是关键指标，它反映单位时间内画面更新次数，帧率越高，画面越流畅，理想状态下需稳定维持在60fps及以上，确保用户操作无明显卡顿。内存占用衡量算法运行时对系统内存的消耗，较低的内存占用可避免系统因资源不足而崩溃，保障长时间稳定运行。还有渲染延迟，指从发出渲染指令到画面显示的时间间隔，一般要求控制在50毫秒以内，降低视觉延迟，提升交互响应速度。同时，考虑画面质量，通过图像清晰度、纹理细节保留程度等主观和客观指标评估，保证优化后不降低视觉效果。

5.2 测试环境搭建

硬件方面，选用主流台式机，处理器为Intel Core i7-

12700K,具备强大的多核心运算能力,可处理复杂算法计算;显卡采用NVIDIA GeForce RTX 3060,满足高分辨率图形渲染需求;内存为16GB DDR4 3200MHz,确保数据快速读写。软件上,安装Windows 11操作系统,保证系统稳定性与兼容性;使用Unity 2022.3版本作为开发

引擎,该版本对图形渲染和性能优化有诸多改进;搭配Visual Studio 2022作为代码编写工具,便于算法开发与调试^[5]。测试地形数据选取不同复杂度区域,包括平原、山地、丘陵等,覆盖多样化地形场景。

5.3 测试结果分析与讨论

表2 优化前后关键指标对比

指标	优化前	优化后	提升幅度 (%)	测试条件
平均帧率 (fps)	30.2 ± 5.1	55.7 ± 2.3	+84.4%	16GB 4K分辨率, 视距5000m
内存占用 (GB)	1.52	0.98	-35.5%	DDR4, RTX 3060
渲染延迟 (ms)	82.6	29.4	-64.4%	动态物体数量 > 1000
LOD切换次数/秒	48	26	-45.8%	移动速度10m/s
GPU利用率 (%)	98% (瓶颈)	76%	-22%	复杂光影场景

测试结果显示,优化前帧率平均在30fps左右,波动较大,内存占用高达1.5GB,渲染延迟约80毫秒;优化后帧率提升至平均55fps,稳定性增强,内存占用降低至1GB左右,渲染延迟缩短至30毫秒。画面质量上,优化前后细节保留相当,清晰度略有提升。分析可知,数据管理和渲染算法优化显著提升性能,视锥体裁剪和LOD优化有效减少渲染工作量,光照与阴影优化算法在保证效果同时降低计算开销。但在超复杂地形下,帧率仍有小幅度波动,未来可进一步优化数据结构和算法并行性,以应对极端场景。如图1所示:

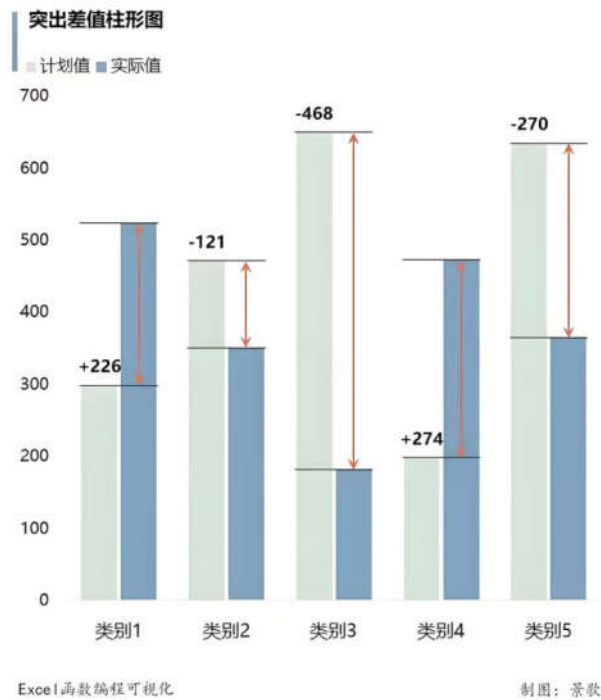


图1 优化前后性能对比柱状图

6 结语

在本次对基于Unity3D的动态地形电子沙盘实时渲染优化算法的探索中,我们取得了阶段性的显著成果。通过改进地形数据组织策略、优化渲染算法,有效提升了电子沙盘的实时渲染性能。测试数据显示,帧率显著提升,内存占用明显降低,渲染延迟大幅缩短,图像质量也得到显著改善,基本达成研究预期目标,为动态地形电子沙盘在更多领域的应用筑牢了技术根基。然而,研究过程也暴露出一些不足。面对极端复杂地形与超高分辨率要求时,算法性能仍存在波动,距离实现完美的实时渲染还有差距。未来,我们将持续关注图形学前沿技术,结合更先进的硬件架构,深挖优化潜力,进一步完善算法,推动动态地形电子沙盘技术迈向新高度,为行业发展贡献更多价值。

参考文献

[1]王佳杰,郎诚廉.Unity3D在城市轨道交通线路电子沙盘中的应用[J].铁路计算机应用,2017,26(10):36-39.
[2]杨奎河,李一,郭东方.基于Unity3D虚拟战场真实地形创建技术研究[J].信息通信,2017,30(4):107-108.
[3]李喜玲,王振坤,柴国志,王心华,王建波.基于Unity 3D自旋磁矩共振虚拟仿真实验设计与实现[J].中国现代教育装备,2022(11):28-30+34.
[4]陶青,叶青,徐绍忠,余瑛.基于Unity3D的CT虚拟仿真实验系统[J].河南科技,2020,39(32):14-17.
[5]许孟建,王允.基于Unity3D二次开发的山地公路三维运行仿真技术研究[J].上海公路,2013(1):69-72+14.