www.cagsbulletin.com

基于 GPU 的地形可视化加速算法研究

李朝奎¹⁾, 方 军^{1)*}, 肖克炎²⁾, 王 宁¹⁾, 周青蓝¹⁾, 隗含涛²⁾

1)湖南科技大学地理空间信息技术国家地方联合工程实验室,湖南湘潭 411201; 2)中国地质科学院矿产资源研究所,自然资源部成矿作用与资源评价重点实验室,北京 100037

摘 要:地形可视化是利用数字高程模型 DEM,采用计算机图形学和图像处理技术进行三维地形模拟显示。 该技术在深部矿产预测、矿产资源评价、虚拟现实、娱乐游戏、飞行模拟等诸多领域有着广泛的应用。随着 数据量的增大,三维地形可视化的实时、流畅视觉效果受到当前的计算机硬件技术水平限制。针对这一问题, 本文运用 ROAM 算法进行地形建模,利用 GPU 高速并行运算性能加速地形可视化建模速度,加速模型显示效 果。实验对比表明:当计算量比较小时,加速效果不显著;随着计算量的增大,计算效果越来越明显;当计算 量达到一定值时,加速效果达到一个稳定的加速趋势。研究结果为地形可视化及矿产资源评价等类似工作提 供了原创性可视化技术支撑。

关键词: 地形建模; ROAM 算法; GPU

中图分类号: P62 文献标志码: A doi: 10.3975/cagsb.2020.012202

Research on Acceleration Algorithm of Terrain Visualization Based on GPU

LI Chao-kui¹), FANG Jun¹)*, XIAO Ke-yan²), WANG Ning¹, ZHOU Qing-lan¹), WEI Han-tao²)

 National-Local Joint Engineering Laboratory of Geo-Spatial Information Technology, Hunan University of Science and Technology, Xiangtan, Hunan 411201;
MNR Key Laboratory of Metallogeny and Mineral Assessment, Institute of Mineral Resources, Chinese Academy of Geological Sciences, Beijing 100037

Abstract: Terrain visualization is a three-dimensional terrain simulation display using DEM, computer graphics and image processing technology. This technology is widely used in many fields, such as deep mineral prediction, mineral resource evaluation, virtual reality, entertainment games, flight simulation and so on. With the increase of data volume, the real-time and smooth visual effect of 3D terrain visualization is limited by the current level of computer hardware technology. In order to solve this problem, ROAM algorithm is used in terrain modeling, GPU high-speed parallel operation performance is used to accelerate the speed of terrain visualization modeling and the effect of model display. The experimental results show that: when the amount of calculation is small, the acceleration effect is not significant; with the increase of the amount of calculation, the calculation effect is more and more obvious; when the amount of calculation reaches a certain value, the acceleration effect reaches a stable acceleration trend. The research results provide original visualization technology support for similar work such as terrain visualization and mineral Resources evaluation.

Key words: terrain modeling; ROAM algorithm; GPU

1999年, NVIDIA 公司首次提出 GPU(Graphics Process Unit, 图形处理器)的概念, 自此, GPU 逐渐 地出现在大众的视野当中(Asirvatham and Hoppe, 2005)。GPU 是计算机显卡的一个核心部件。在最

初的设计中 GPU 仅被用来渲染图形图像,不能推 广到其他计算应用上,很大程度上限制了 GPU 的 发展。但是随着研究的深入,GPU 逐渐成为通用计 算处理器,完成图像处理中的大部分计算。GPU 不

本文由国家自然科学基金(编号: 41571374)和国家重点研发计划(编号: 2018YFB0504501; 2017YFB0503802)联合资助。

收稿日期: 2019-11-30; 改回日期: 2020-01-16; 网络首发日期: 2020-02-06。责任编辑: 张改侠。

第一作者简介:李朝奎,男,1967年生。博士,教授,博士生导师。主要从事三维地理信息理论方法与应用技术研究。E-mail: chkl_hn@163.com。 *通讯作者:方军,男,1985年生。博士后。主要从事地理信息理论方法及应用研究。E-mail: 280518056@qq.com。

仅比 CPU有更多的计算处理单元,而且自身的存储 器的传输速度相对于 CPU也具有明显的优势,因此 具有高速计算的能力(Brown and Severin, 2009)。随 着近年来数据量的增大,对于计算机的运算能力的 要求逐渐提高, GPU的应用领域亦愈来愈广泛(涂超, 2003)。

地形可视化在深部矿产预测、矿产资源评价、 航空仿真、飞行模拟、虚拟现实以及娱乐游戏方面 有着广泛的应用(刘庆元等, 2014)。为了使计算机在 处理复杂物体表面模型时保持实时的显示, 通常采 用 LOD(Level of Detail)模型(Hoppe, 1997)。LOD 模 型根据视点与模型之间的距离大小选择不同精度的 模型进行表示,离视点比较远,模型比较粗糙;离 视点越近,模型越精细(Li et al., 2016)。在众多的 LOD 算法中, ROAM(Real-time Optimally Adapting Meshes)算法因简单易懂,具有很好的可扩展性,能 够实现在连续的区域内从平面到最高级的优化, 还 能够进行动态地形的实时更新,在地形可视化领域 受到广泛关注(Duchaineau et al., 1997)。基于 GPU 的地形可视化加速算法研究结果为地形可视化及深 部矿产预测、矿产储量估算等类似工作提供原创性 可视化技术支持,具有相当的参考价值。

1 ROAM 算法

1.1 ROAM 算法原理

如图 1 所示, ROAM 算法是利用二叉树剖分方 法对三角形进行划分, 从而表达地表形态。当需要 精细的模型时, 进行细粒度分割, 等级较高; 当需 要粗糙模型时, 进行粗粒度分割, 等级较低(Zhang et al., 2013)。

1.2 裂缝问题

将地形进行划分成许多三角形进行地形模拟 时, 渲染后的模型会出现一些三角形裂缝。这是由



Fig. 1 ROAM mesh division principle

于相邻三角形划分的层次等级不同所导致的(刘局 科, 2015)。如图 2 所示,在 AD 边两侧出现了相邻 地形块不同等级的情况,AD 边右上侧由 \triangle AEC 和 \triangle CED 组成,AD 边左下侧由一个大三角形 \triangle ABD 组成,AD 边为它们的公共边。如果 $H_E \neq 1/2(H_A+H_D)$, 即点 E 没有落到 AD 边上,则 AD 边两侧 \triangle ABD 和 \triangle ACE、 \triangle CED 在 AD 边处无法重合,在渲染后,即 产生裂缝 \triangle AED。

1.3 裂缝消除

采用强制分割法进行裂缝消除。如图 3a, 要对 1 号三角形进行分割, 但当对 1 号三角形进行分割 后, 2 号三角形的分割层次比 1 号三角形分割后的层 次低, 有可能会出现裂缝现象, 因此对 2 号三角形 进行分割, 再对 1 号三角形进行分割。同理, 对 2 号三角形分割后, 2 三角形和 3 号三角形之间会出 现裂缝, 需对 3 号三角形进行分割, 进行递归分割 后, 直至分割的三角形与相邻三角形的分割层次差 不超过 1。最终结果如图 3d 所示。

2 实验与结果分析

2.1 GPU 编程模型

GPU 强大的并行计算能力和高速资源带宽解决 了 CPU 面对大数据计算能力不足的问题。但是传统 的 GPU 计算要求编程开发人员了解图形编程知识和



图 2 裂缝的产生 Fig. 2 Crack Generation



Fig. 3 Forced segmentation

原理,而且各个线程之间是无法进行通信的,这些因素极大地限制了 GPU 的发展(额尔敦达来,2006)。 针对此问题,NVIDIA 公司于 2007 年 6 月推出统一 计算设备架构(Compute Unified Device Architecture, CUDA)(王冬等,2009)。CUDA 的推出为广大的 GPU 爱好者提供了极大的方便。CUDA 是一种基于 GPU 的并行计算软件环境的集成,使用类 C 语言进 行编程开发。CUDA 构架提供了不同的存储器分区, 方便数据的存储和使用,也为各线程之间进行通信 提供了渠道。开发者则可以利用 C 语言知识和 CUDA 架构对 GPU进行并行计算开发,不需要掌握 复杂的图形学知识和学习新的计算机语言。 CUDA 的推出为 GPU 在各行各业中的应用奠定了 良好的基础。

CUDA 架构定义为:将 CPU 端称为主机端,也称为 host 端,GPU 端称为设备端,也叫做 device 端 (王冬等,2009)。目前,CUDA 常用的编程模型是 CPU+GPU 模式(杨金柱等,2010)。此模式如图 4 所示,主机端将数据传输到设备端,利用 GPU 的并行 运算能力将计算结果传输回主机端。其中主机端运行的主要是程序中串行的部分,数据并行的部分交 由设备端进行处理。

在设备端上所执行的程序称为 kernel 函数,也 叫做内核函数。如图 5 所示,用线程网格(Grid)的形



式来表达 kernel 函数。每个 Grid 可以包括多个线程 块(block),每个 block 包括多个线程(thread),由此 组成大量的线程进行并行计算。而每个 grid 中分为 多少 block, 每个 block 又包含多少 thread, 这些都 是由开发者在编程时所设定。并且分为不同的 block, 不同的 thread, 运算的效率是不同的, 这是由数据 指令传达方式所决定的。理论上,数据量足够大的 情况下,分割的线程数越多,并行运算效率越快, 但由于硬件本身的限制因素, block 的数量不能超过 65 535 个, 在每一个 block 中设置的线程个数不能 超过 512 个。Thread 是 GPU 执行的最小单位, 在 CUDA 构架下,并行计算分为两种形式,一种是粗 粒度并行计算,另一种是细粒度并行计算。粗粒度 并行计算是指块与块之间的并行计算、细粒度并行 计算是指每个块内多线程的并行计算。在任务执行 过程中,最重要的是对每个线程的准确定位,对于 一维的 block, 线程 threadID 就是 threadIdx.x, 对于 二维的 Block(T_x, T_y), 线程的 ThreadID 为 (ThreadIdx.x + ThreadIdx.y × T_x), 对于三维的 Block(T_x , T_y , T_z)线程的 ThreadID 为(ThreadIdx.x + ThreadIdx. $y \times T_x$ + ThreadIdx. $z \times T_x \times T_y$).

2.2 实验与结果分析

实验的硬件条件为 4 GB 内存, Inter Core i5-3210M 处理器, NVIDIA GeForce GT 620M 显卡, Windows7 操作系统。利用 Visual C++ 6.0 进行编写, 使用 OPENGL 进行渲染。

2.2.1 裂缝修复实验

实验数据采用 2¹⁰×2¹⁰ 的大小的高程图进行实验, 根据 ROAM 算法原理, 对未消除地形裂缝的



图 5 并行计算模式 Fig. 5 Parallel computing mode

ROAM 算法及使用强制分割算法消除地形裂缝的两种方法进行实验分析。图 6 为未经过裂缝消除的地形模型渲染(图 6a)和利用强制分割算法消除地形裂缝后(图 6b)的渲染效果对比图。

实验结果对比可以发现:不进行裂缝消除的 ROAM 算法, 渲染后的地形模型效果不理想, 出现 许多三角形裂缝; 经过强制分割算法进行绘制的模 型, 渲染效果比较好, 没有裂缝的产生, 视觉效果 比较好。



图 6 裂缝修复前后对比图 Fig. 6 Contrast diagram of fracture repair



图 7 不同层次模型对比图 Fig. 7 Contrast diagram of different level model

2.2.2 GPU 地形模型构建实验

实验数据大小为 1025×1025 个采样点和 1024×1024 个像素, 原始数据是 png 格式, 通过 photoshop 将其转换为 raw 格式。利用 GPU 的高性 能并行特性对地形模型进行模拟, 并使用同样数据, 但不采用 GPU 的实验进行对比, 分析在不同层次 等级下, 两种实验效率对比。对两种不同层次细节 模型进行展示, 对比两种细节层次模型下三角网模 型、光照模型、纹理模型的三种状态。

图 7a、c、e 为精细模型的三角网模式、光照模 式和纹理模式下的效果图。图 7b、d、f 为粗糙模型 的三角网模式、光照模式和纹理模式下的效果图。 三角网密集的区域为视点所在的位置,离视点越近, 其三角形分割的等级越高,符合人眼视觉的特点。 对比这两种不同层次下的效果图,可以发现:精细 模型剖分的三角网面数比粗糙模型更多。从渲染效 果对比图中可以看出,精细模型可以表达地形更多 的细节特征。

为研究 GPU 高性能并行计算加速效果,对 8 个 不同等级下的地形模型进行对比试验。图 8 展示了 8 种不同层次细节模型下的地形模拟三角网。



对不同层次细节模型下的三角网进行统计,

图 8 8 种不同层次模型效果图 Fig. 8 Eight different level model effect diagram

表 1 不同层次模型加速比 Table 1 Acceleration ratio of different level model

三角形数量	CPU OK 耗时 /ms	GPU OK 耗时 /ms	加速比
	71115	71115	
2 088.00	484.27	20.05	24.15
4 376.00	1 076.83	33.64	32.01
5 488.00	1 341.76	36.16	37.11
9 712.00	2 309.66	58.13	39.73
21 840.00	5 335.40	113.44	47.03
36 079.00	8 790.80	163.17	53.87
72 150.00	17 441.69	318.63	55.61
144 406.00	33 619.29	579.38	58.02



图 9 加速比折线图 Fig. 9 Discounted graph of acceleration ratio

并进行性能测试。实验结果表明: GPU 能极大地加速 模型的构建。表1展示了利用 GPU 和利用 CPU 进行 模型构建的加速比。图 9 展示了加速比的折线图。 分析可以发现,当构建的模型层次较低,三角形面 片数较少时, GPU 具有一定的加速效果。但随着模型 精度的提高,计算量的加大, GPU 的计算效果也越来 越好;当加速到一定阈值时,加速比趋于平稳阶段。

3 结论

利用 GPU 高性能并行计算功能,将其与 ROAM 算法相结合进行地形可视化研究,为矿产资 源评价、三维地质建模等工作提供基础分析数据。 实验及结果表明:

(1)运用强制分割方法对 ROAM 算法所产生的 裂缝进行修复,能够消除地形建模过程中产生的地 形裂缝。

(2)利用 GPU 进行加速实验时,当计算量比较 小时,加速效果不显著;随着计算量的增大,计算 效果越来越明显;当计算量达到一定值时,加速效 果达到一个稳定的趋势。

(3)研究结论为开展地形可视化、深部矿产预测、矿产资源评价等同类计算及渲染工作时合理运用 GUP 性能提供参考。

(4)研究结果为提高三维精准空间分析效果,优 化三维模型可视化等提供原创性技术支持。

Acknowledgements:

This study was supported by National Natural Science Foundation of China (No. 41571374), and National Key Research & Development Program of China (Nos. 2018YFB0504501 and 2017YFB0503802).

参考文献:

- 额尔敦达来. 2006. 基于 GPU 的大规模地形场景绘制关键技术 研究[D]. 长沙: 国防科学技术大学.
- 刘局科. 2015. 基于 GRID/TIN 混合结构的地形场景数据组织方 法研究[D]. 南京:南京师范大学.
- 刘庆元, 易柳城, 刘莉. 2014. 基于 diamond-square 算法的数字 地形模型构建与三维可视化研究[J]. 测绘工程, 23(2):1-4.
- 涂超. 2003. ROAM 算法原理及其应用研究[J]. 辽宁工程技术大 学学报, 22(2): 176-179.
- 王冬, 张豫南, 李瀚飞. 2009. 一种基于 GPU 的瓦片块四叉树地 形绘制算法[J]. 系统仿真学报, (S1): 106-109.
- 杨金柱,赵大哲,栗伟. 2010. 基于 GPU 的体绘制算法研究[J]. 电子学报, 38(B02): 202-206.

References:

- ASIRVATHAM A, HOPPE H. 2005. Terrain Rendering using GPU-Based Geometry Clipmaps[J]. Gpu Gems, 2(2): 27-46.
- BROWN R J, SEVERIN K P. 2009. Otolith chemistry analyses indicate that water Sr: Ca is the primary factor influencing otolith Sr:Ca for freshwater and diadromous fish but not for marine fish[J]. Journal Canadien Des Sciences Halieutiques Et Aquatiques, 66(10): 1790-1808.
- DA Lai. 2006. Research on GPU-Based Large-Scale Terrain Rendering[D]. Changsha: National University of Defense Technology(in Chinese with English abstract).
- DUCHAINEAU M, WOLINSKY M, SIGETI D E, MILLER M C. 1997. ROAMing terrain: Real-time Optimally Adapting Meshes[J]. Office of Scientific & Technical Information Technical Reports, Proceedings IEEE Visualization 97: 81-88.
- LI Feng-Chun, WANG Run-qiu, JIANG Xian-yi, YANG Jian. 2016. Real-time visualization of large three-dimensional seismic geometry on terrain[J]. Geophysical & Geochemical Exploration, 40(5): 1030-1034.
- HOPPE H. 1997. View-dependent refinement of progressive meshes[C]//Conference on Computer Graphics and Interactive Techniques. ACM Press/Addison-Wesley Publishing Co.: 189-198.
- LIU Ju-ke. 2015. Research on the Method of Terrain Scene Data Organization Based On GRID/TIN Hybrid Structure[D]. Nanjing: Nanjing Normal University(in Chinese with English abstract).
- LIU Qing-yuan, YI Liu-cheng, LIU Li. 2014. Digital terrain model construction and 3D visualization based on diamond square algorithm[J]. Engineering of Surveying and Mapping, 23(2): 1-4(in Chinese with English abstract).
- TU Chao. 2003. Research on theory and application of ROAM method[J]. Journal of Liaoning Technical University, 22(2): 176-179(in Chinese with English abstract).
- WANG Dong, ZHANG Yu-nan, LI Han-fei. 2009. Tiled Block Quadtree Terrain Rendering Algorithm based on GPU[J]. Journal of System Simulation, (S1): 106-109(in Chinese with English abstract).
- YANG Jin-zhu, ZHAO Da-zhe, LI Wei. 2010. The Research Volume Rendering Algorithm Based on GPU[J]. Acta Electronica Sinica, 38(B02): 202-206(in Chinese with English abstract).
- ZHANG Zhi-feng, ZHANG Na. 2013. A LOD algorithm based on out-of-core for large scale terrain rendering[C]//International Conference on Mechatronic Sciences, Electric Engineering and Computer. IEEE: 2168-2171.