

Journal of Image
and Graphics

中国图象图形学报



ISSN1006-8961
CN11-3758/TB

2012
Vol.17 No.

4

中国科学院遥感应用研究所
中国图象图形学学会主办
北京应用物理与计算数学研究所

中国图象图形学报

Zhongguo Tuxiang Tuxing Xuebao

2012年4月 第17卷 第4期(总第192期)

目次

综述

图像分割中的模糊聚类方法 李旭超, 刘海宽, 王飞, 白春艳(447)

图像处理和编码

GPU 辅助的希尔伯特变换轮廓术 周波, 赵小敏, 王东平(459)

引入连续性强度和置信度因子的快速图像修复 李开宇, 孙玉刚(465)

自适应的有效非局部图像滤波 许光宇, 檀结庆, 钟金琴(471)

改进的 PMD 距离图像超分辨率重建算法 张旭东, 沈玉亮, 胡良梅, 陈菁菁(480)

压缩感知在 Micro-CT 图像超分辨重建中的应用 王丽艳, 韦志辉, 罗守华, 顾宁(487)

对偶四元数单片空间后方交会算法 姬亭, 盛庆红, 王惠南, 刘微微(494)

利用运动强度判据的高效自适应运动估计算法 郭晓珉, 姚睿, 刘智跃, 王友仁(504)

图像分析和识别

核空间散度阈值法 吴成茂(512)

近邻自适应局部尺度的谱聚类算法 孔万增, 孙昌思核, 张建海, 胡三清, 杨灿(523)

LUV 色彩空间中多层次化结构 Nyström 方法的自适应谱聚类算法 刘雅蓉, 汪西莉(530)

结合图像增强的心血管内超声中-外膜边缘检测 邱璇, 黄靖, 杨丰, 邢栋, 涂圣贤(537)

融合图像特征的一致点匹配方法及其应用 张久楼, 李春丽, 冯前进, 陈武凡, 阳维(546)

图像理解和计算机视觉

多蚁群动态协作优化的道路图像分割算法 林丽莉, 周文晖(553)

篮球比赛视频中持球队员行为预测	王千,夏利民,谭论正(560)
利用 Principal Warps 评估颅面几何相似度	朱新懿,耿国华,温超(568)

计算机图形学

图形处理器空间插值并行算法的实现	赵艳伟,程振林,董慧,方金云(575)
------------------------	---------------------

虚拟现实与增强现实

面向 GPU 的批 LOD 地形实时绘制	张兵强,张立民,张建廷(582)
----------------------------	------------------

遥感图像处理

光学遥感舰船目标识别方法	杜春,孙即祥,李智勇,滕书华(589)
自适应超完备字典学习的 SAR 图像降噪	杨萌,张弓(596)

第 18 届中国遥感大会征文通知	封 2
第 33 届亚洲遥感会议征文通知	封 2

中国图象图形学报

刊名题字: 宋 健

月刊(1996 年创刊)

第 17 卷 第 4 期

2012 年 4 月 16 日出版

主管单位 中国科学院
主 办 中国科学院遥感应用研究所
 中国图象图形学学会
 北京应用物理与计算数学研究所
主 编 李小文
编辑出版 《中国图象图形学报》编辑出版委员会
 北京 9718 信箱 邮编 100101
 电子信箱:jig@irsa. ac. cn
 电话:010-68407995 010-82614429
 网 址:www. cijg. cn
印刷装订 北京北林印刷厂
广告经营许可证 京朝工商广字第 0346 号
总 发 行 北京报刊发行局
订 购 全国各地邮局
国外发行 中国国际图书贸易总公司
 (中国国际书店)
 (北京 399 信箱 邮编 100044)

Superintended by Chinese Academy of Sciences
Sponsored by Institute of Remote Sensing Application,
 CAS China Society of Image and Graphics
 Institute of Applied Physics and Computational
 Mathematics
Chief editor LI Xiaowen
Editor, Publisher Editorial and Publishing Board
 of Journal of Image and Graphics
 (P. O. Box 9718, Beijing 100101, China)
 E-mail: jig@irsa. ac. cn
Distributed by Beijing Bureau for Distribution of Newspapers
 and Journals
Domestic All Local Post Offices in China
Foreign China International Book Trading Corporation
 (P. O. Box 399, Beijing 100044, China)
Printed by Beijing Beilin Printing House

ISSN 1006-8961 CN11-3758/TB CODE ZTTFXZ 国内邮发代号: 82-831 国外发行代号: M1406 国内定价: 45.00 元

第 18 届中国遥感大会征文通知

“第 18 届中国遥感大会”将于 2012 年 10 月 19 日-23 日在武汉召开。本届会议由中国遥感委员会主办,中国测绘学会摄影测量与遥感专业委员会和武汉大学承办。会议将围绕“遥感—全方位的社会服务”这一宗旨,以遥感学界院士与知名专家的特邀报告,分会场专题技术交流与技术讲座,重点项目研讨汇报、技术展览,新技术与新产品发布,专业委员会理事会等多种形式开展,同时举行“第 7 届中国青年遥感辩论会”和“第 2 届全国高分辨率遥感数据处理与应用研讨会”。

会议将全方位地展示遥感(RS)、全球定位系统(GPS)、地理信息系统(GIS)等方面的最新成果,为专家、学者和政府主管部门搭建联系纽带,为研发和用户提供技术交流平台,共同促进遥感科技的发展、遥感产业化的推进和大遥感体系的建立。

本届会议围绕大会主题将就遥感新理论、技术、方法和应用进行征文,范围包含但不限于以下方面:

- 1) 国家遥感中长期发展战略、国际遥感前沿与进展;
- 2) 航天、航空、低空、地面遥感技术及系统;
- 3) 光学、红外、高光谱及激光遥感技术;
- 4) 主、被动微波及雷达遥感技术;

- 5) 数字摄影测量与制图;
- 6) 高分辨率遥感数据处理与应用;
- 7) 地理空间数据处理技术与方法;
- 8) 地理国情监测(土地、农业、林业、矿产、环境、地质及水资源等);
- 9) 海洋、气象与全球变化;
- 10) 遥感、地理信息系统与导航定位系统(3S)集成与应用;
- 11) 智慧城市与数字地球;
- 12) 深空探测与行星测绘;
- 13) 教育、培训与社会公共事业。

征文采用在线方式投稿;

投稿要求:论文内容不涉密,且未在国内外学术刊物或正式学术会议上发表过;被录用的全文将收入大会论文集(送 ISTP 检索),并精选 70~90 篇口头报告论文编辑出版英文 SPIE 会议文集;大会将评选青年优秀论文(参加口头报告),论文将直接进入英文 SPIE 会议文集。

论文摘要截止日期为 2012 年 5 月 15 日,全文截稿日期为 2012 年 6 月 15 日。

会议相关信息,请查阅会议网址:<http://rsgis.whu.edu.cn/18ccrs/index.html>

“第 18 届中国遥感大会”组委会

第 33 届亚洲遥感会议征文通知

“第 33 届亚洲遥感会议”将由泰国地理信息和空间技术发展局(GISTDA)、科技部(MOST)和亚洲遥感协会(AARS)联合主办,于 2012 年 11 月 26-30 日,在泰国芭堤雅市宗滴恩酒店举行。这是亚洲遥感协会每年一届的系列学术会议。本届大会征文包括传感器与平台、算法和图像处理、GIS 与 Web GIS、全球导航卫星系统、灾害、自然资源、环境科学、教育和宣传、健康科学、制图、其他等方面。

会议重要日期:

- 论文摘要提交截止:2012 年 5 月 15 日;
 - 论文接收通知:2012 年 7 月 1 日;
 - 论文全文提交截止:2012 年 9 月 30 日;
 - 网上注册截止:2012 年 10 月 26 日;
 - 会议召开日期:2012 年 11 月 26-30 日。
- 会议还将组织学生专场和技术展览,其他信息请访问会议网站:<http://acrs2012.gistda.or.th>

与往年一样,中国遥感委员会仍将鼓励中国遥感科研人员和企事业单位参加会议,并组团参加学术交流和会议展览。

中国遥感委员会

Journal of Image and Graphics

(Monthly, Started in 1996)

Vol. 17 No. 4 April 2012

Contents

Review

The survey of fuzzy clustering method for image segmentation Li Xuchao, Liu Haikuan, Wang Fei, Bai Chunyan (447)

Image Processing and Coding

GPU assisted Hilbert transform profilometry Zhou Bo, Zhao Xiaomin, Wang Dongping (459)

Fast image inpainting algorithm introducing continuous strength and confidence factor Li Kaiyu, Sun Yugang (465)

Adaptive efficient non-local image filtering Xu Guangyu, Tan Jieqing, Zhong Jinqin (471)

Improved super-resolution reconstruction algorithm for PMD range image
..... Zhang Xudong, Shen Yuliang, Hu Liangmei, Chen Jingjing (480)

Image superreconstruction for Micro-CT based on compressed sensing Wang Liyan, Wei Zhihui, Luo Shouhua, Gu Ning (487)

Dual quaternion of space resection with single-image Ji Ting, Sheng Qinghong, Wang Huinan, Liu Weiwei (494)

Efficient adaptive motion estimation algorithm based on motion intensity Guo Xiaomin, Yao Rui, Liu Zhiyue, Wang Youren (504)

Image Analysis and Recognition

Divergence thresholding method in kernel space Wu Chengmao (512)

Spectral clustering based on neighboring adaptive local scale
..... Kong Wanzeng, Sun Changsihe, Zhang Jianhai, Hu Sanqing, Yang Can (523)

Adaptive spectral clustering algorithm based on Nyström method with multi-level structure in LUV color space
..... Liu Yarong, Wang Xili (530)

Image enhancement based media-adventitia border detection in intravascular ultrasound images
..... Qiu Xuan, Huang Jing, Yang Feng, Xing Dong, Tu Shengxian (537)

Coherent point drift registration combined with image feature and its application
..... Zhang Jiulou, Li Chunli, Feng Qianjin, Chen Wufan, Yang Wei (546)

Image Understanding and Computer Vision

Dynamic multi-colony ant cooperative optimization schemes for road image segmentation
..... Lin Lili, Zhou Wenhui (553)

Behavior prediction of ball carriers in basketball match videos Wang Qian, Xia Limin, Tan Lunzheng (560)

Estimate of craniofacial geometry shape similarity based on principal warps
..... Zhu Xinyi, Geng Guohua, Wen Chao (568)

Computer Graphics

Realization of GPU parallel spatial interpolation method
..... Zhao Yanwei, Cheng Zhenlin, Dong Hui, Fang Jinyun (575)

Virtual Reality and Augmented Reality

GPU-based real-time terrain rendering algorithm using batched LOD
..... Zhang Bingqiang, Zhang Limin, Zhang Jianting (582)

Remote Sensing Image Processing

Method for ship recognition using optical remote sensing data
..... Du Chun, Sun Jixiang, Li Zhiyong, Teng Shuhua (589)

SAR images de-speckling algorithm via an adaptive over-complete learning dictionary
..... Yang Meng, Zhang Gong (596)

中图法分类号: TP391.9 文献标志码: A 文章编号: 1006-8961(2012)04-0582-07

论文引用格式: 张兵强, 张立民, 张建廷. 面向 GPU 的批 LOD 地形实时绘制[J]. 中国图象图形学报, 2012, 17(4): 582-588

面向 GPU 的批 LOD 地形实时绘制

张兵强, 张立民, 张建廷

海军航空工程学院电子信息工程系飞行仿真技术研究所, 烟台 264001

摘要: 为提高大规模地形实时渲染时的绘制效率, 提出一种使用地形分块作为处理单元的批 LOD 算法。在预处理阶段, 将多分辨率的地形数据划分成适于 GPU 批处理的分块, 使用四叉树进行分块的有效组织。在此基础上, 提出一种基于分块绘制的 LOD 误差标准, 简化层次选取的计算量, 通过增加“裙”和进行几何变形实现了层次间的有效过渡; 实时绘制过程中, 使用视锥裁剪减少进入图形硬件的数据量, 利用地形四叉树列表和预测机制实现地形数据的有效加载管理。实验结果表明, 本文算法能够充分发挥图形硬件的性能, 具有较高的地形实时渲染效率。

关键词: 地形绘制; LOD 误差标准; 四叉树; 多分辨率; 视锥裁剪

GPU-based real-time terrain rendering algorithm using batched LOD

Zhang Bingqiang, Zhang Limin, Zhang Jianting

*Institute of Flight Simulation Technology, Department of Electronic and Information Engineering,
Naval Aeronautical and Astronautical University, Yantai 264001, China*

Abstract: To make the real-time rendering of large scale terrain more efficient, a batched LOD algorithm is proposed in which terrain blocks are handled as the processing unit. In the preprocessing stage, multi-resolution terrain data is partitioned into regular blocks with predefined sizes. These blocks are organized in the form of quadtree. Based on this form, an error metrics for LOD selection is designed in order to simplify the computation of level decision. Then by the means of adding skirts and geometry morphing, different levels are transited without a visual distortion or popping. When terrain is being rendered, frustum culling is used to reduce data transmitted to graphics hardware. To efficiently implement data loading and datasets management, terrain quadtree lists and a prediction strategy are introduced. From the result of final simulation, it can be seen that the algorithm can fully harness the power of current graphics hardware and reach higher rendering rates.

Key words: terrain rendering; error metrics for LOD; quadtree; multi-resolution; view frustum culling

0 引言

大规模地形数字高程模型 (DEM) 的可视化在 GIS、虚拟现实、飞行仿真和军事演练等方面应用广泛。不断发展的遥感技术推生出覆盖范围更广、分辨率更高的 DEM 数据, 然而大规模、高分辨率地形数据往往超过了现代图形硬件的处理能力, 而且, 三

角形过密会在视点较远时, 使同一像素对应多个三角形, 视角的微小扰动会引起像素所对应三角形的剧烈变化, 从而该像素点的颜色不稳, 产生走样现象^[1]。

为了对大规模地形进行实时渲染, 当前研究主要集中在各种视点相关的细节层次模型 (LOD) 算法上。早期 LOD 算法使用 CPU 进行多分辨率网格的生成, 能够实现最优化的地形数据表示, 但是需要

收稿日期: 2011-06-10; 修回日期: 2011-10-08

第一作者简介: 张兵强 (1981—), 男, 讲师, 海军航空工程学院通信与信息系统专业博士研究生, 主要研究方向为战场综合环境仿真、虚拟现实。E-mail: zbqwh1981@163.com

大量的 CPU 时间,并且增加了 CPU 与图形硬件的数据通信量。

当前 GPU 的数据处理速度远快于 CPU,为提高系统整体绘制效率,LOD 算法发展趋势是充分利用 GPU 的批量绘制能力,减少 CPU 与 GPU 通信带宽需求。因此,地形渲染不再是逐个多边形或顶点进行绘制,而是对多个三角形组成的三角形簇进行批量绘制,这类方法就是面向 GPU 的批 LOD 算法^[2-6]。而现代 GPU 着色器的可编程性带来了高效的地形绘制算法,例如,使用片段着色器实现地形的光线投射计算^[7],利用顶点着色器实现顶点高程纹理拾取^[8],使用几何着色器实现基于 Tilelets 的视锥体二次裁剪^[9]等,这些算法将部分优化计算移植到 GPU 上执行,大大提高了算法效率。

本文实现了一种高效的基于 GPU 的 LOD 地形实时绘制算法,将地形数据预处理为适合 GPU 批处理的分块,采用四叉树对地形数据进行有效组织,提出一种基于分块绘制的 LOD 误差标准,并且对地形绘制过程中的层次过渡及数据实时调度算法进行了研究。

1 地形数据组织

在本文地形绘制算法中,地形数据以规则网格高度场的形式给出,在进行实时绘制之前,对地形原始数据进行预处理,生成多分辨率数据模型。

首先,各层数据组织顺序为从第 0 层到第 L-1 层,第 0 层为最高的分辨率层,其他各层数据分辨率依次减为 1/4。将所有数据组织成金字塔形状,并引入四叉树结构进行数据组织,如图 1 所示。对每一层的数据点进行记录,使用三元坐标 (i, j, k) 表示数据金字塔中的点,其中 k ∈ {0, 1, ..., L-1} 表示数据所在的细节层次, i, j ∈ {0, 1, ..., 2^k - 1} 表示顶点在数据网格中的位置。

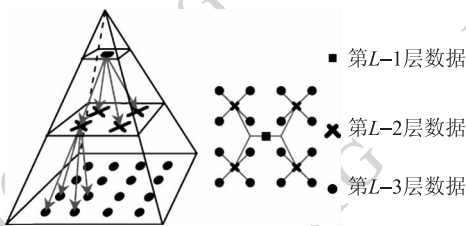


图 1 多分辨率金字塔数据示意图

Fig. 1 Multi-resolution pyramid representation

在地形低分辨率生成过程中,需要对高度图进行重采样,而使用传统基于点采样或者箱式滤波方法会使某些高海拔点或地形细节丢失,因此需要使用一种能够保存数据特征的滤波方法,本文采用文献[10]中提到的 Neville 滤波方法。常用的 Neville 滤波模型有 2 阶和 4 阶,滤波权重模型如图 2 所示。

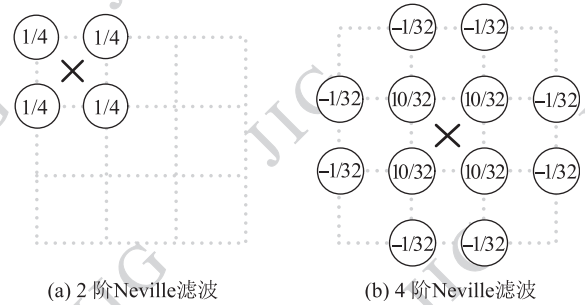


图 2 Neville 滤波器

Fig. 2 Neville filter

对于边界点,向外扩充一个单元的采样点,扩充采样点的高程值等于与其相邻的高程值,为得到较好的绘制效果,采用 4 阶 Neville 滤波器模型,则坐标 (i, j, k) 高程值可按下式计算

$$H_{i,j,k} = \sum_{m=2i-1}^{2i+2} \sum_{n=2j-1}^{2j+2} C_{m,n} H_{m,n,k-1} \quad (1)$$

式中, $H_{i,j,k}$ 代表坐标 (i, j, k) 处的高程值, $C_{m,n}$ 为图 2 (b) 所示的权重值。使用 Neville 滤波后的效果如图 3 所示。

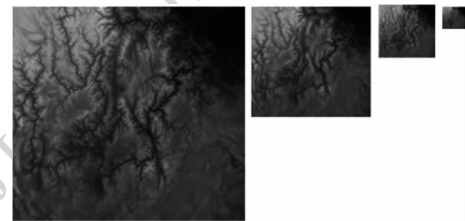


图 3 高度图滤波处理效果图

Fig. 3 Height map generated after filtering

在生成地形数据金字塔后,从地形数据的第 0 层最精细层开始,逐层地将地形网格划分为固定大小的 M × M 的地形分块。在实时绘制时,以地形分块为处理的最小单元,以便降低 CPU 的运算,并充分利用 GPU 的批量三角形绘制能力,同一层相邻地形分块之间边界处重叠一个单位的采样点,如图 4 所示。

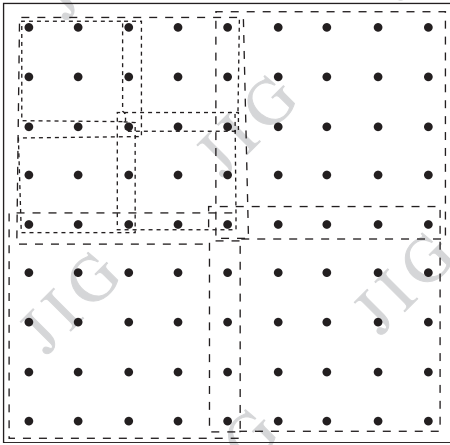


图4 地形分块示意图

Fig. 4 Illustration of terrain blocks

2 LOD 误差标准

大地形绘制时,需要根据当前视点位置信息选取合适的分辨率地形数据进行绘制。分辨率的选取标准是 LOD 误差标准。目前较为常用的有基于视点的屏幕误差方法^[11],基于首先四叉树的精确误差计算方法^[12],批量动态自适应网格误差计算方法^[13],考虑了面片法向量变化的受限法向锥误差计算方法^[14]。本文使用屏幕空间误差作为网格层次选择的标准。

由于数据处理单元为地形分块,因此在这里误差计算不采用逐顶点的方式。为简化起见,假设细节层次 l 中的每一个地形分块的几何空间误差 ε_l 为固定值,考虑到地形高度场采样的等间隔性,设在 l 层地形数据的网格间距为 g_l ,则引入误差因子 μ 限定 $\varepsilon_l = \mu g_l$, μ 为预设常量值,用于进行地形整体细节调整。即在第 l 层,各地形分块与原始地形数据的高程偏差为 ε_l ,由于地形数据网格间距 $g_{l+1} = 2g_l$,所以 $\varepsilon_l = 2^l \varepsilon_0$ 。

不失一般性,设视线的方向与地形水平面平行,则几何空间误差 δ_c 到屏幕空间误差 η 的变换为^[11]

$$\eta = \delta_c \frac{H}{2d \left| \tan\left(\frac{fov}{2}\right) \right|} \quad (2)$$

式中, H 为视口对应的屏幕像素高度, d 是离视点的距离, fov 为视场角。对于地形绘制,设屏幕误差的阈值为 τ ,对于点 P , P 到视点的距离为 d_p ,则点 P 对应的空间误差 $\varepsilon(P)$ 可表达为

$$\varepsilon(P) = \frac{2}{h} \tan\left(\frac{fov}{2}\right) \tau d_p \quad (3)$$

若令 $\alpha = \frac{2}{h} \tan\left(\frac{fov}{2}\right) \tau$, 则式(3)变为 $\varepsilon(P) = \alpha d_p$ 。对地形块测试时,可以通过测试其包围体得到离视点最近的点 P' ,通过 $d_{p'}$ 计算该地形块的空间误差 $\varepsilon(P')$,由 $\varepsilon(P')$ 决定该地形块使用的细节层次,按下式进行层次选择

$$\varepsilon_l \leq \varepsilon(P') \leq \varepsilon_{l+1} \quad (4)$$

由于 $\varepsilon_l = 2^l \varepsilon_0$,所以式(4)变化为 $2^l \leq \frac{\varepsilon(P')}{\varepsilon_0} \leq 2^{l+1}$,

于是得到

$$l = \left\lfloor \log_2\left(\frac{\varepsilon(P')}{\varepsilon_0}\right) \right\rfloor \quad (5)$$

式中, $\lfloor \cdot \rfloor$ 表示向下取整。当计算的几何误差小于 ε_0 时,取最精细层 $l=0$ 进行绘制,而当 $l > L-1$ 时,取最粗糙层 $l=L-1$ 。综上,可以得到 LOD 的细节层次为

$$l = \begin{cases} 0 & \varepsilon(P') \leq \varepsilon_0 \\ \left\lfloor \log_2\left(\frac{\varepsilon(P')}{\varepsilon_0}\right) \right\rfloor & \varepsilon_0 < \varepsilon(P') < 2^{L-1} \varepsilon_0 \\ L-1 & \varepsilon(P') \geq 2^{L-1} \varepsilon_0 \end{cases} \quad (6)$$

3 层次过渡优化

3.1 裂缝消除

绘制地形时,当相邻地形分块处于不同的细节层次时,会在边界处产生 T 顶点,造成 T 裂缝失真。若采用顶点平滑过渡或者边界模板的方法,则需要判断相邻地形块的情况,需要较大的 GPU 运算量来确定裂缝的处理方案。

为了便于 GPU 上实现裂缝的消除,采用了添加垂直“裙”对裂缝进行遮盖的方法^[5]。在每个地形块的边界围上一圈从顶点垂直到水平面上的“裙”面,使得这些面遮挡了裂缝口,从而消除裂缝。每个块都有自己的“裙”,不需要考虑当前块与其他块之间的边界连接、层次差别、拓扑关系等,具有很强的块独立性,是面向 GPU 地形渲染算法的一个很好选择,如图 5 所示。

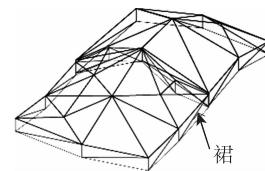


图5 相邻分块“裙”示意图

Fig. 5 Illustration of skirts around block borders

增加“裙”消除裂缝时,难点在于如何确定“裙”的高度,在相邻块的边界处产生裂缝的主要原因是使用不同的细节层次造成的,即相邻层边界处存在几何误差。因此,可以利用几何误差确定“裙”的高度。下面对“裙”的高度进行计算说明。

在 l 层的每个地形分块含有 $M \times M$ 地形采样点,且由上一小节知道网格几何空间间距为 g_l ,因此,地形分块空间覆盖范围为 $(M-1)g_l = (M-1)\varepsilon_l/\mu$ 。在以下的计算推导中只考虑地形的 2 维水平方向。在第 l 层,根据式(3)可以设 $\varepsilon_l = \alpha d_l$,即 $d_l = \frac{\varepsilon_l}{\alpha}$,则由式(3)知,对任意的该层数据采样点 P , d_p 为视点到点 P 的距离,有 $d_l \leq d_p \leq d_{l+1}$,并且有 $d_{l+1} = \frac{1}{\alpha}\varepsilon_{l+1} = \frac{1}{\alpha}2\varepsilon_l = 2d_l$ 。

假设两相邻地形块细节层次分别是 l 和 $l + \Delta l$,两地形块各自边界点距离最小值为 d_1 和 d_2 ,因此, d_1 和 d_2 的取值范围分别是 $[d_l, d_{l+1}]$ 和 $[d_{l+\Delta l}, d_{l+\Delta l+1}]$,则有

$$|d_1 - d_2| \geq d_{l+\Delta l} - d_{l+1} \geq (2^{\Delta l} - 2)d_l \quad (7)$$

又两地形块相交于一条网格边,所以根据三角形不等式有

$$|d_1 - d_2| \leq (M-1)g_l \quad (8)$$

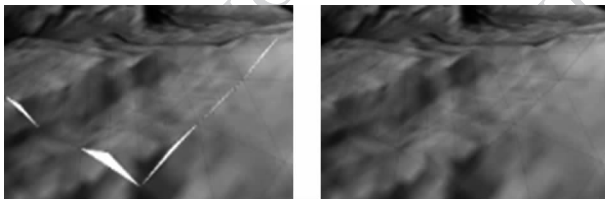
由式(7)(8)得

$$\Delta l \leq \log_2 \left[\frac{(M-1)\alpha}{\mu} + 2 \right] \quad (9)$$

由此可以得到地形位置相邻层之间最大层次差别 $\Delta l_{\max} = \log_2 \left[\frac{(M-1)\alpha}{\mu} + 2 \right]$ 。所以在相邻两层之间,交界处裂缝最大的几何空间距离为

$$S_h = \varepsilon_l + \varepsilon_{l+\Delta l_{\max}} = (2^{\Delta l_{\max}} + 1)\varepsilon_l \quad (10)$$

在确定边界“裙”的高度时,可以采用式(10)的 S_h 。对于给定位置的视点,相邻地形块的细节层次相差通常不超过 1,因此可以取“裙”的高度为 $3\varepsilon_l$ 。使用“裙”的绘制效果如图 6 所示。



(a) 未使用“裙” (b) 使用“裙”

图 6 使用“裙”前后绘制效果图

Fig. 6 Terrain rendered with and without skirts

3.2 几何变形

为消除同一地形区域绘制时层次细节变化引起的突跃现象,需要对数据进行几何变形,使层次平滑过渡,这属于地形平滑可视化中的时间连续性问题^[15]。从粗糙层到精细层只需要在粗糙层两相邻采样点之间添加精细层对应的点,反之移除相应的顶点,如图 7 所示。第 n 层的顶点 V_n 与相邻另一层中对应位置 V_{n-1} 进行相互过渡转换。

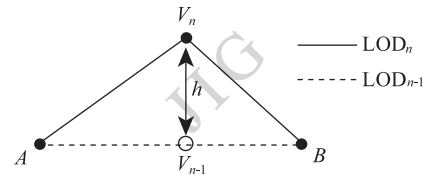


图 7 几何变形过渡示意图

Fig. 7 Illustration of level transition using geometry morphing

点的过渡变化方向只是在高度方向上,是 1 维的,因此只需要一个过渡因子。假设过渡因子为 f , $f \in [0, 1]$,则在顶点着色器中,对顶点 V 进行变形的公式为

$$V_{pos} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} - f \begin{bmatrix} 0 \\ 0 \\ h \end{bmatrix} \quad (11)$$

式中, h 为过渡点的最终高程值与当前高程值 z 的差值; $f \in [0, 1]$, f 根据视点离地形分块的距离 d 逐渐变化; d_{new} 为要变化到的层次对应的距离值; d_{old} 为当前的层次距离值,则 f 的计算式如下:

$$f = 1.0 - \frac{d - d_{\text{old}}}{d_{\text{new}} - d_{\text{old}}} \quad (12)$$

在上面的方法中,只有视点移动时才进行几何过渡,减少了视点移动过程的视觉失真。

4 数据更新

4.1 视锥体裁剪

视锥体裁剪是在 CPU 上以地形分块为单位进行的,利用四叉树的空间数据结构,按层次进行裁剪,对含有包围体的每个节点和视锥进行相交测试。测试从根节点开始,若某个节点在视锥体之外,则该节点及其子节点不被处理;若节点包围体与视锥相交,则继续对该节点的子节点进行测试,直到最终子节点;若节点包围体在视锥内部,则该节点的所有终子节点将被绘制,不需要再进行子节点的测试。地形裁剪效果如图 8 所示,其中围绕视锥体的方块效

果是因为绘制基本单元为地形分块。

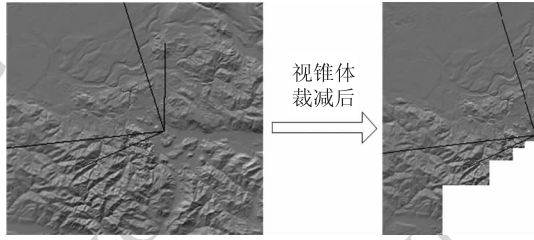


图 8 地形裁剪后效果图

Fig. 8 Terrain view after frustum culling

4.2 数据实时加载处理

地形数据加载过程中,为提高效率,外存大量数据应当进行有选择的加载。选择的依据是:根据当前视点位置,选择位于视锥体内的地形分块。

首先按照坐标划分成一定尺寸的不同区域,每个区域含有固定数量的地形分块,然后使用地形四叉树(TQ)进行数据组织。每个区域的地形高度数据以多分辨率金字塔的形式存储。整个地形高度场数据采用多个 TQ 来组织表示,对所有的 TQ 使用一个列表来组织,称为地形四叉树列表(TQL)。

将每个地形数据分块的信息保存在一个分片对象中(TO),在每个 TO 中除了存储地形分片的地理信息、细节层次和尺寸外,还保存指向分片数据的指针和记录分片在内存中被调用情况。将所有的 TO 存入一个分片向量(TV),使数据能够更好地被组织与访问。在 TQL 中的每一个四叉树节点都对应 TV 中的一个分片对象,TQ 的每个层次对应地形的一个细节层次,TQ 根节点与一个分片对象对应,代表一块地形区域,具有最小分辨率,每个父节点对应 4 个子节点。

在每一帧绘制时,遍历 TQL 四叉树,确定 TQL 中位于视锥内部的分片,当视点变化时,分片集与地形区域同时发生变化。由于地形区域使用不同分辨率,因此需要对进入视锥内的地形分块进行优先级设定,以取得最佳的分辨率表示。在视点移动的过程中,同一地形区域往往要经历由远到近再远的过程,因此,对位于视锥体内的地形区域,内存中应该加载和该区域对应的多个分辨率地形分块,再根据前述的 LOD 误差标准进行分辨率选择。

设 Q_M 是地形四叉树的节点队列, T_M 是当前帧位于视锥体内的节点队列,为了防止加载的数据量

超出内存容量,限制加载到系统内存的地形数据总量为 M ,对地形四叉树列表进行访问,确定在视锥体内的地形方法伪代码如下:

```

sort(TQL) //按离视点距离由近及远对 TQ 进行排列
for all TQi in TQL
  if(TQi is in frustum)
    enqueue( $Q_M$ , root(TQi)) //将 TQi 根节点插入  $Q_M$  队列中
if( $T_M$  is not empty)
   $T_M = empty$ 
while size( $Q_M$ ) < M and  $Q_M$  is not empty do
  dequeue( $q$  in  $Q_M$ ) //从  $Q_M$  队列中取出  $q$  节点放入  $T_M$ 
  enqueue( $T_M$ ,  $q$ )
  for all children  $c_j$  of  $q$ 
    if( $c_j$  intersects the frustum)
      enqueue( $Q_M$ ,  $c_j$ )

```

在得到 T_M 后,从外存中加载相应的地形分块数据,对位于 T_M 中但没有驻留在内存中的地形分块进行加载。设 R_M 为当前在内存中的地形分块集,则 R_M 可以分为两部分:一部分为 $R_M^E = T_M \cap R_M$,表示属于 T_M 但是又在当前系统内存中的地形分块集合;另一部分为 $R_M^F = R_M - T_M$,表示驻留在内存中但不属于 T_M 的地形分块集合。由于要求 $|R_M^E| + |R_M^F| \leq M$,为了能够加载不在内存中但属于 T_M 的地形分块数据,要求将 R_M^F 中某些数据分块从内存卸载掉。卸载或保留的依据是地形分块在后续帧中被绘制的概率。

一般来说,在视点附近的地形较远处地形更容易被重用,同样,离视锥中心线近的分块比与视线成大角度的分块更容易被重用。设 u_i 是视点到地形分块 T_i 中心的单位向量, V 是视线方向单位向量, d_i 是视点到分块 T_i 中心点的距离,则可以得到每一个地形分块的优先级因子表达式

$$w_i = \frac{u_i \cdot V}{d_i} \quad (13)$$

通过为 R_M^F 中的分块建立优先级,确定被卸载的分块,卸载分块后从外存中加载 $R_M - T_M$ 分块集。

视点的运动速度是变化的,当视点运动较慢时, T_M 中的数据可能全部在 R_M 中,即 $T_M \subset R_M$,此时不需要从外存加载数据。这时,为了提高加载效率,使用一种预加载策略,即通过当前帧和前一帧的视点信息预测下一帧视点位置。假设当前视点位置为 P_1 ,前一帧视点位置为 P_0 ,则一般可以假定当前视点运动方向为 $(P_1 - P_0) / \|P_1 - P_0\|$ 。假设两帧之间的时间间隔为 t ,则视点运动速度大小为

$\|P_1 - P_0\|/t$, 所以视点移动速度为

$$v = \frac{P_1 - P_0}{\|P_1 - P_0\|} \frac{\|P_1 - P_0\|}{t} = \frac{P_1 - P_0}{t} = (P_1 - P_0)f \quad (14)$$

式中, f 表示帧速, 则下一帧视点位置 P_2 预测为

$$P_2 = P_1 + vt = P_1 + (P_1 - P_0) = 2P_1 - P_0 \quad (15)$$

通过预测的视点位置, 预加载下一帧视锥内的地形数据, 预卸载优先级低的分块, 这样提高了系统内存中地形数据的加载效率。

5 实验结果

对本文算法进行仿真实验, 编程环境为 VC++ 2008, 使用 C++、OpenGL 和 GLSL 语言进行程序实现, 所使用的计算机硬件配置为 Intel(R) Core(TM) 2 DUO E8400 @ 3.0 GHz CPU, 2 GB RAM, NVIDIA GeForce 9600 GT 显卡。

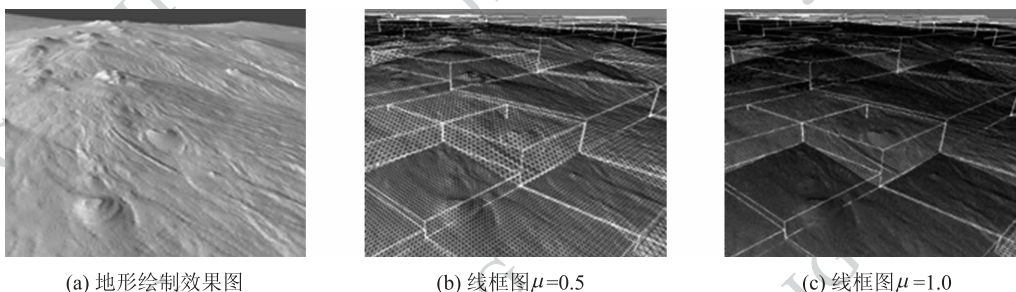


图9 地形绘制效果图及线框图

Fig.9 Terrain view and corresponding wireframe modes

为进一步验证本文算法的绘制效率和实时性能, 选取 Geometry Clipmaps 算法与本文算法进行对比。预设程序视点的漫游路径, 屏幕显示窗口大小为 1024×768 , 计算并记录实时绘制的帧率, 得到图 8 所示的结果。帧率的快速起伏变化主要是由地形数据的更新引起的。从图 10 中可以看出, 本文算法的绘制帧率在 180 帧/s 以上, 而 GC 算法的仅达到 130 帧/s; 使用本文算法的数据加载和预测策略, 数据更新对绘制帧率的影响远小于 GC 算法, 并且能够得到比较稳定的帧率。此外, 分块批处理方法以及本文提出的 LOD 误差标准在较大程度上提高了整体绘制速率, 大大减少了绘制一帧所需的时间。

算法的测试数据为 Hawaii 地形高度图数据, 数据分辨率为 10 m, 数据对应的网格大小为 13825×16769 , 覆盖地形区域大小为 $138 \text{ km} \times 168 \text{ km}$, 预处理 LOD 层共生成 9 层网格, 占用存储空间为 463 MB。该数据集数据量较大, 能够说明算法对大规模地形数据的处理效果, 进一步增加原始高程数据量只会增加硬盘存储量, 对本文算法而言, 不会影响内存以及 CPU 和 GPU 的性能表现。

得到的绘制效果如图 9 所示。使用预先计算的法向图, 并添加光照效果进行绘制, 得到地形效果图(图 9(a)), 图 9(b)(c) 为相应的线框图, 为说明本文算法效果, 对绘制的地形分块添加包围体。通过调整 μ 值得到不同的绘制精细模型, 图 9(b) 中 $\mu = 0.5$ 时, 所选取 LOD 级别为 2, 图 9(c) 中 $\mu = 1.0$, LOD 级别为 1。随着 μ 值增加, LOD 绘制选取细节层次的分辨率会提高, 这样, 借助 μ 值可以对地形绘制整体细节进行调整。从图 9(b)(c) 中可以看出, 算法进行了有效的 LOD 计算与分辨率层次的选取。

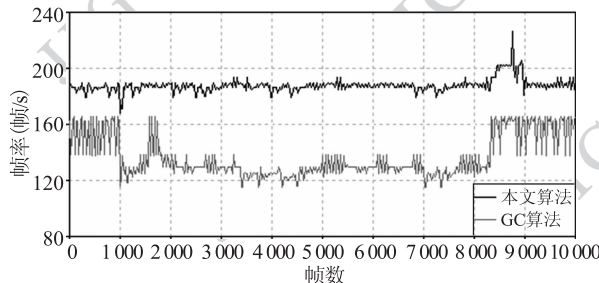


图10 地形实时绘制帧率变化曲线图

Fig.10 Recorded frame rates in terrain rendering

表 1 对两种算法的系统 CPU/GPU 使用率以及平均绘制三角形数量进行了比较。从表中可以看出, 本文算法 GPU 使用率较高, 绘制三角形数量(在

距地表 1 090 m, 俯仰角向下 14° 时的测量数据) 在设定误差因子 $\mu = 0.5$ 时明显多于 GC 算法。在本文算法中, 占用 CPU 时间的主要因素是实时的数据管理调度, 这种方法在没有明显增加 CPU 使用率的条件下, 增强了数据绘制的稳定性和三角形数量。

表 1 算法性能比较

Table 1 System performance of algorithms

	GC 算法	本文算法
平均 GPU 使用率/%	49	73
CPU 使用率/%	≤ 23	≤ 25
每帧绘制三角形数/ M	1.548	2.539
每帧绘制时间/ms	5.8 ~ 8.3	4.2 ~ 5.7

6 结 论

设计的基于 GPU 的 LOD 地形绘制算法充分利用了当前 GPU 的批处理能力, 以地形分块作为处理单元, 所采用的 LOD 算法使得进行多分辨率绘制时仅需较少的计算量, 使用“裙”进行层次间裂缝消除, 采用层次顶点过渡来防止同区域细节层次变化的突跃现象, 方法简单高效, 适用于大规模的、具有海量数据的地形漫游系统。而采用 TQL 的实时数据管理加载策略和预测加载方案能有效增强绘制的稳定性。从实验结果可知, 算法增加了 GPU 的使用率, 提高了绘制帧率。

参考文献 (References)

- [1] Williams L. Pyramidal parametrics [J]. Computer Graphics, 1983, 17(3): 1-11.
- [2] Pajarola R, Gobbetti E. Survey on semi-regular multiresolution models for interactive terrain rendering [J]. The Visual Computer, 2007, 23(8): 583-605.
- [3] de Boer W H. Fast Terrain Rendering Using Geometrical MipMapping [EB/OL]. (2000-10) [2011-05-15]. [http://](http://www.flipcode.com/tutorials/geomipmaps.pdf)

www.flipcode.com/tutorials/geomipmaps.pdf.

- [4] Levenber G J. Fast view dependent level of detail rendering using cached geometry [C] // Proceedings of IEEE Visualization 2002. Boston: IEEE Computer Society Press, 2002: 259-266.
- [5] Ulrich T. Rendering massive terrains using chunked level of detail control [C] // Proceedings of SIGGRAPH 2002 Volume Course Notes 35. San Antonio, Texas: ACM Press, 2002.
- [6] Losasso F, Hoppe H. Geometry clipmaps: Terrain rendering using nested regular grids [J]. ACM Transactions on Graphics, 2004, 23(3): 769-776.
- [7] Dick C, Krüger J, Westermann R. GPU ray-casting for scalable terrain rendering [C] // Proceedings of the Eurographics Areas Papers. Munich: ACM Press, 2009.
- [8] Bösch J, Goswami P, Pajarola R. RASTeR: Simple and efficient terrain rendering on the GPU [C] // Proceedings of Eurographics Areas Papers, 2009: 35-42.
- [9] Bhattacharjee S, Patidar S, Narayanan P J. Real-time rendering and manipulation of large terrains [C] // Proceedings of Indian Graphics & Image Processing. India: IEEE Computer Society Press, 2008: 551-559.
- [10] Kovacevic J, Sweldens W. Wavelet families of increasing order in arbitrary dimensions [J]. IEEE Transactions on Image Processing, 2000, 9(3): 480-496.
- [11] Lindstrom P, Koller D, Ribarsky W. Real-time, continuous level of detail rendering of height fields [C] // Proceedings of Siggraph 1996. New Orleans: ACM Press, 1996: 109-118.
- [12] Pajarola R. Large scale terrain visualization using the restricted quadtree triangulation [C] // Proceedings of Visualization 1998. North Carolina: IEEE Press, 1998: 19-26.
- [13] Cignoni P, Ganovelli F, Gobbetti E. BDAM: batched dynamic adaptive meshes for high performance terrain visualization [J]. Computer Graphics Forum, 2003, 22(3): 505-514.
- [14] Li S, Ji J F, Liu X H, et al. High performance navigation of very large-scale terrain environment [J]. Journal of Software, 2006, 17(3): 535-545. [李胜, 冀俊峰, 刘学慧, 等. 超大规模地形场景的高性能漫游 [J]. 软件学报, 2006, 17(3): 535-545.]
- [15] Zhang Y Y, Jiang H Z, Han J W. Smooth terrain rendering algorithm based on GPU [J]. Journal of University of Electronic Science and Technology of China, 2009, 38(6): 1006-1010. [张燕燕, 姜洪洲, 韩俊伟. 基于 GPU 的平滑地形可视化算法 [J]. 电子科技大学学报, 2009, 38(6): 1006-1010.]