

超大规模地形场景的高性能漫游*

李 胜^{1,3+}, 冀俊峰^{1,4}, 刘学慧¹, 吴恩华^{1,2}

¹(中国科学院 软件研究所 计算机科学重点实验室,北京 100080)

²(澳门大学 科学技术学院 电脑及资讯科学系,澳门)

³(北京大学 信息科学技术学院,北京 100871)

⁴(国家信息中心,北京 100045)

High Performance Navigation of Very Large-Scale Terrain Environment

LI Sheng^{1,3+}, JI Jun-Feng^{1,4}, LIU Xue-Hui¹, WU En-Hua^{1,2}

¹(Laboratory of Computer Science, Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

²(Department of Computer and Information Science, University of Macau, Macao, China)

³(School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)

⁴(State Information Center, Beijing 100045, China)

+ Corresponding author: Phn: +86-10-62754878, E-mail: lisheng@graphics.pku.edu.cn, <http://www.pku.edu.cn>

Li S, Ji JF, Liu XH, Wu EH. High performance navigation of very large-scale terrain environment. *Journal of Software*, 2006,17(3):535–545. <http://www.jos.org.cn/1000-9825/17/535.htm>

Abstract: Interactive navigation of very large-scale terrain environment involves many difficulties because of its out-of-core geometric data and texture, which can not be wholly loaded into the internal memory. This paper puts forward a high performance technique for real-time walkthrough of the large-scale terrain environments. It performs geometric simplification within each terrain chunk during preprocessing and then generates a view-dependent continuous LOD of the whole scene with geomorphing at runtime. Both silhouette preserving and shading preserving criteria are satisfied by applying a new error metric-constrained normal cone to view-independent simplification. The generated models for rendering is more suitable to human's perception than those by traditional geometric simplification criterion. Furthermore, the pre-extracted potential silhouette of each chunk can be used to construct the incremental horizon dynamically. This can prevent the ineffectual data from loading into the main memory and reduce the out-of-core paging and the number of triangles feeding to the renderer, thus lead to significant saving in interactive rendering. This paper also makes use of the advantage of GPU to generate lighting and shadow for large-scale outdoor environment, and balances the tradeoff between GPU, CPU and I/O through multi-thread.

Key words: interactive walkthrough; very large-scale terrain; constrained normal cone; potential silhouette; incremental horizon

* Supported by the National Natural Science Foundation of China under Grant Nos.60103011, 60473105 (国家自然科学基金); the National Grand Fundamental Research 973 Program of China under Grant Nos.2002CB312102, 2004CB719403 (国家重点基础研究发展规划(973))

Received 2004-06-29; Accepted 2004-12-08

摘要: 超大规模地形场景包含大量的几何和纹理数据,无法一次性载入内存,并具有极高的复杂度,因而无法进行实时绘制.提出一种高性能的外存地形场景实时漫游技术.该方法使用离散层次细节技术并结合视点相关的动态连续层次细节选择和过渡.算法为地表的简化提出一种新的基于受限法向锥的误差计算方法,使得模型简化具有轮廓保持和光照保持特性.当生成网格包含三角形数目相当时,该方法比传统的基于几何误差的简化更加符合漫游时视觉的感知规律.场景简化过程中提取出的潜在轮廓特征可以通过巧妙地构建漫游时视线方向上的增量地平线来随时更新场景不同部分的可见性信息,并以此控制无用数据页面的载入和无效场景部分的绘制,提高绘制速度.漫游系统采用多线程技术,使 CPU、GPU、I/O 三者的效率得到充分发挥,并可实时生成具有光照和阴影效果的漫游图像.

关键词: 交互式漫游;超大规模地形场景;受限法向锥;潜在轮廓;增量地平线

中图法分类号: TP393 **文献标识码:** A

大规模地形场景的实时绘制在各种驾驶模拟仿真、虚拟战场以及三维游戏中有非常重要的应用.地表场景的实时漫游面临两大困难问题:原始几何模型庞大的数据量远远超过了系统的实时处理能力,仅仅覆盖数平方公里的模型就包含千万级甚至更多的多边形;与地表模型相对应的高精度纹理更加重了交互漫游的难度.有两种实时加速方法常被用来解决上述问题,一种方法是采用视点相关自适应层次细节技术(LOD)^[1-6]简化整个场景的复杂度,同时尽量减小生成图像的失真.它能够以 1~2 个量级的简化程度有效降低每一帧绘制多边形的数目.LOD 技术同样可以减少纹理的绘制数量,因为对于远处的大部分场景,使用低分辨率的纹理映射就足够了.另一种方法是采用可见性剔除技术,包括视域剔除、背向面剔除和遮挡剔除^[7-10].尤其是在进行漫游时,视点离地面较近,视线方向几乎水平,地形场景具有很长的景深.因此,大部分算法都是针对此种情况进行遮挡计算以剔除对最终成像无效的场景部分,优化场景组织,实现近似 output-sensitive 的绘制.

无论是 LOD 技术还是遮挡剔除技术都是以 CPU 为核心的,其存在的缺陷是为了获得加速而所需的额外处理开销有时甚至会超过减少绘制多边形数目所带来的时间补偿.随着近年来图形硬件的迅猛发展,原有受 CPU 限制的图形处理流水线逐渐上升成为以具有特殊高效图形处理能力的 GPU 为中心的图形处理框架.这种趋势导致了面向 GPU 的批 LOD(aggregate LOD)算法^[11]的开发.此外,细节的增加和地表范围的扩大逐渐形成超大规模的海量数据,这些原始数据无法被一次性装入系统内存.因此,如何开发数据的内、外存高效调度算法,降低无用数据页面载入数目并相应减少无效场景部分的绘制,也是三维地形漫游系统面临的难题.

适应当前图形技术发展的新形势,本文提出一种高性能的外存外地形场景实时漫游技术.该方法是一种使用离散层次细节技术并结合视点相关的动态连续层次细节选择和几何过渡的批 LOD 方法.在预处理阶段使用视点无关的简化生成块网格,以受限法向锥作为其误差计算方法指导模型简化.它具有轮廓保持和光照保持特性.在简化得到的多边形数目相当时,该方法生成网格模型比传统的基于几何误差的简化更加符合漫游时人类视觉的感知规律.潜在轮廓伴随着块网格生成的过程得到.该轮廓特征被巧妙地用来组织场景漫游时的增量地平线,实施场景的可见性动态判定.最终以此可见性信息控制无用数据页面的载入和无效场景的绘制,降低绘制每一帧图像的开销,加速整个交互式场景漫游过程.我们采用多线程技术,使 CPU、GPU、I/O 三者的效率得到充分发挥.而增量地平线还可以提供视点潜在可见场景集合与光源潜在可见场景集合,用硬件支持的 shadow map 技术实时生成场景的具有明暗效果和阴影效果的漫游图像.

1 相关工作

大规模室外地形场景漫游一直是近年来的研究热点.以往的自适应层次细节绘制算法中最流行的是基于规则方形栅格(RSG)构建二叉树结构的层次表示形式.这种多分辨率表示具有压缩存储的优点,因为相邻顶点之间的连接关系隐含在规则栅格排列中.经典二叉树结构算法^[1,4-6]以及思想类似的二叉树结构算法^[2]都被用于生成视点相关的连续层次细节,其中还包含块误差计算^[4]、不同层次细节之间的 geomorphing 技术^[1]、三角带组织^[5]等具体内容的详细探讨.另一类重要的 LOD 算法建立于不规则三角形网格结构(TIN)基础上,主要包括 Multi-Triangulation 结构方法^[12]和视点相关递进网格(progressive mesh)方法^[3].在这类算法中,顶点位置的选择

有更大的自由度,生成网格的三角形数目少于基于规则栅格方法所生成的.但是,由于 TIN 数据结构组织高度复杂并且难以在其几何表示上贴金字塔层次纹理,通常的大规模地形可视化系统往往并不采用 TIN 类方法.

需要耗费大量的处理时间进行视点相关的三角化是自适应层次细节方法的主要缺点.为了减少运行时刻 CPU 进行动态三角化的负载并充分利用 GPU 上的显存,批 LOD 方法被提了出来.批 LOD 的思想就是把在预处理阶段进行三角化后,把邻接三角形集合组合成不同的组,称为三角形簇或者批三角形.运行时刻的 LOD 选择以三角形簇为单位在显存中进行,这样就可以尽量消除 CPU 造成的瓶颈. Levenberg^[11]提出 CABBT 算法,把连续几帧保持不变的批三角形存储到显存内,充分利用了帧与帧之间的连续性以及高性能显卡的优点.而 Cignoni 等人提出 BDAM 技术^[13],三角形簇采用 TIN 结构形式,并把三角形簇的几何层次表示与四叉树表示的纹理结合起来进行大规模地形场景的绘制. Ulrich^[14]采用类似的思想,生成块网格并进行连续的 LOD 控制.除此之外,批 LOD 方法的另外一个特点是易于与 out-of-core 的数据调度管理算法相结合. Lindstrom 专门提出一种有效的顶点布局方法(基于空间填充曲线^[15])组织整个外存模型的数据^[16],能利用顶点的局部性提高内存数据和外存数据之间的存储一致性,以减少内外存之间数据页面调度的次数.

可见性判定常被用来加速场景绘制,尤其是像大规模地形这样的长景深场景. Cohen-Or 等人^[17]综述了可见性技术的原理与应用.而针对室外地形场景, Stewart 提出了快速计算地表每个栅格点的地平线,并构建层次离散地平线,然后在运行时刻进行块地形剔除的算法^[9,18].基于 Tile-Voxel 的可见性预计算方法也被应用到室外地形场景中^[10].以上算法的共同不足之处在于:它们都需要进行大量的预处理,而预计算得到的可见性信息也需要占用大量的磁盘存储空间. Lloyd 等人提出了一种几乎不需要预处理量的地形场景动态地平线的追踪方法^[7],李胜等人也利用室外场景所具有的地理特征,开发了层叠式遮挡剔除以及可见性驱动模型连续层次细节的地形绘制加速算法^[8].但是这些可见性算法没有提供对超大规模 out-of-core 模型数据的支持,加速作用也仅仅限于减少三角形绘制数目上.

2 框架概述

整个高性能室外地形漫游系统如图 1 所示,主要由地表模型几何和纹理数据的预处理过程以及场景实时漫游调度与显示过程组成.

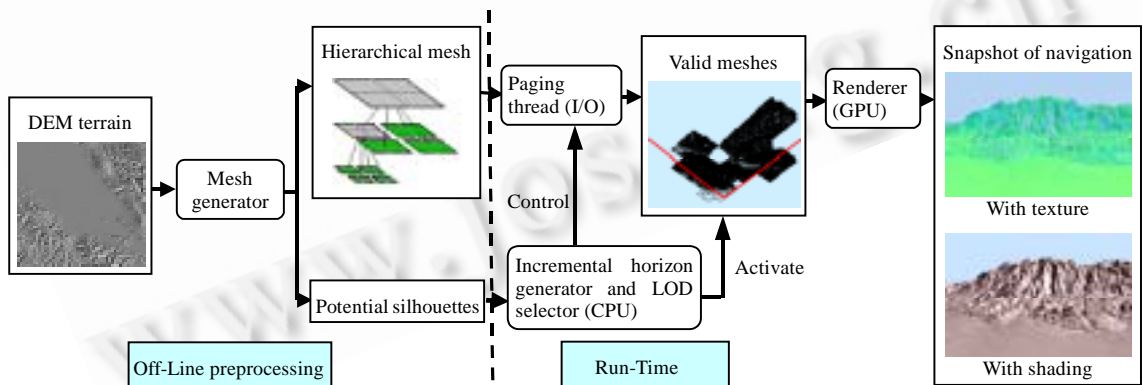


Fig.1 High performance terrain navigation system architecture
图 1 高性能室外地表漫游系统框架

(1) 离线预处理部分.地表层次模型表示为一棵由静态块网格为结点构成的规则四叉树.原始的外存模型数据经过网格生成器进行视点无关的层次简化,而简化所采用的误差计算标准即为受限法向锥.层次块网格都以三角带的形式存储表示.同时,在三角带的构建过程中,采样得到每个层次块的潜在轮廓,层次块网格作为四叉树的结点按照线性索引的方式存储在内存中.此外,在处理过程中还可以顺便计算得到整个场景的法向图,以便于漫游时利用其恢复被简化模型的表面细节,生成具有真实感光照效果的图像.

(2) 实时运行部分.在实时运行阶段,系统采用多线程机制,其中主线程的工作为计算块网格的 LOD 层次,

并利用预处理好的块潜在轮廓生成增量地平线,进行可见性剔除.子线程为模型数据页面调度线程,负责依照当前视点情况从外存中载入有效模型和纹理数据,子线程的数据块调度受主线程计算得到的可见性和 LOD 信息的控制.增量地平线生成器同时激活场景中可见层次细节结点,最后 GPU 将场景中的活动结点对应的有效数据进行绘制,并且可以根据用户交互指令生成具有真实感光照或阴影效果的漫游图像.

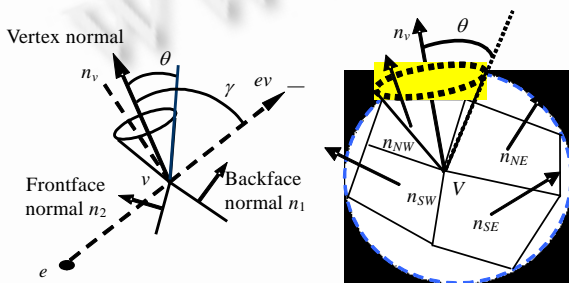
在运行时刻,整个系统框架中 CPU,GPU,I/O 各司其职,各负其责,相互协调,构成了一个高效、有机的整体.

3 场景多分辨率模型的构建

整个地表场景的多分辨率结构由规则层次四叉树表示.四叉树每个结点包含的块网格以三角带的形式组织,可以直接利用硬件绘制命令调用进行绘制.块网格的简化方法对地表漫游时的视觉效果十分重要,因此我们提出以受限法向锥计算顶点误差并实施层次选择的方法,以便更好地保持地表场景的特征.同时,为了剔除不可见的场景区域,对每层次地形块采样得到其潜在轮廓,在实时漫游中递进生成当前视点的增量地平线.

3.1 基于受限法向锥(constrained normal cone)的误差计算

在场景的多分辨率表示生成过程中,误差的计算是关键.以往视点无关的简化算法几乎都采用了空间几何误差这种局部误差计算方法作为简化标准^[2,4,5,16],同时依照地形场景中每个顶点的依赖关系图,几何误差值可以进行传递.由于传统方法中一个顶点的误差往往由区域内具有顶点依赖关系的父亲顶点和顶点自身决定,并不能真实反映模型表面的局部形状和变化,以这样的误差计算方式进行简化有失严谨.



(a) Normal cone
(a) 法向锥示意图
(b) Computation of constrained normal cone in 3D space
(b) 受限法向锥在 3D 空间中的计算
Fig.2
图 2

简化图像的保真度并不是由几何标准来度量的,而是由实际视觉感知决定.轮廓信息往往能够勾勒出物体的外形,为人的视觉对物体的感知提供非常重要的信息.而在轮廓部分产生的扭曲变形也容易误导人对空间中的三维物体形状的感知.在室外场景漫游时,群山遮挡掩映,绵延不断,不同深度的轮廓信息对于漫游时的观察尤其重要.法向锥(normal cone)是一种检测轮廓的有效方法^[19].一个结点所表示的表面包含轮廓信息,当且仅当相对于当前视点,它的法向锥既包含了前向网格表面的法线方向,也包含了后向网格表面的法线方向,如图 2(a)所示.

在地表场景漫游中,人的视点在场景中可以随意移动,但视线方向基本都是水平的或者与水平方向呈较小的夹角,在此前提下,可以使用法向锥角代表一个顶点所在的微面成为轮廓的概率,具有较大锥角的顶点所在区域成为轮廓的概率也较大.为了保持轮廓,那些法向锥角的顶点越大越应该具有更高的误差值而不致被简化掉.因此,我们采用受限法向锥作为模型简化的误差标准.受限法向锥的计算方法如图 2(b)所示,对地形场景中的每一栅格点 V 所代表的由 4 块相邻面片所组成的表面的中心,其 4 块邻域表面的单位法向量分别为 $n_{NW}, n_{NE}, n_{SW}, n_{SE}$, 则 n_v 为其主法向, θ 是法向锥的半锥角.顶点 V 所在表面的法向锥的主法向 $n_v = [n_{vx}, n_{vy}, n_{vz}]$ 和半锥角计算公式如下^[20]:

$$\begin{cases} n_{vx} = \frac{1}{2} (\max_i n_{ix} + \min_i n_{ix}) \\ n_{vy} = \frac{1}{2} (\max_i n_{iy} + \min_i n_{iy}), \cos(\theta) = \min_i (n_i \cdot \bar{n}_v), \bar{n}_v \text{ 为 } n_v \text{ 的单位矢量表示} \\ n_{vz} = \frac{1}{2} (\max_i n_{iz} + \min_i n_{iz}) \end{cases}$$

其中 n_i 分别表示 $n_{NW}, n_{NE}, n_{SW}, n_{SE}$ 4 种情况.由于凹陷表面通常不会成为轮廓,所以采用受限法向锥的半锥角作为

真正的误差值.受限法向锥半锥角的计算在法向锥基础上添加如下限制,使凹陷表面误差值最小,并保证凸表面具有较大误差值(n_z 和 n_x 分别为 z 坐标轴和 x 坐标轴的法线矢量):

$$\begin{aligned} (n_{NW}, n_{NE}, n_z) > 0 \text{ and } (n_{SW}, n_{SE}, n_z) > 0 &\Rightarrow \theta = 0, \\ (n_{NW}, n_{SW}, n_x) > 0 \text{ and } (n_{NE}, n_{SE}, n_x) > 0 &\Rightarrow \theta = 0, \end{aligned}$$

计算出每个顶点的局部误差后,根据层次结构中具有依赖关系的顶点之间的误差传递原则,每个顶点 V 的最终误差的计算公式为

$$e_V = \max \left\{ \theta, \max_{v_i \in D_V} \{e_{v_i}\} \right\},$$

D_V 为顶点 V 的所有子孙顶点的集合.该误差传递计算保证了具有依赖关系的顶点之间误差的单调性.相应地,设 δ 表示层次块网格简化的误差限,则每个层次的误差阈值计算如下:

$$\begin{cases} \delta(0) = \max \theta, \delta(d) = \min \theta \\ \delta(i) = f(i, \max \theta, \min \theta) (0 < i < d) \end{cases},$$

其中 $\max \theta, \min \theta$ 分别表示第 0 层最粗糙分辨率块网格的误差阈值和第 d 层最精细块网格的误差阈值, d 为由用户指定的四叉树最大深度,而 f 为 i 的单调递减函数.采用基于受限法向锥的误差计算方法进行简化,生成的网络具有轮廓保持的特性.图 3(a)采用几何误差标准简化后的场景包含了 25 275 个三角形,而图 3(b)包含了 24 677 个三角形.图 3(b)中的三角形数目虽然少,但是生成的图像保真度高.可见,当简化得到数目基本相当的三角形时,我们的简化方法与传统方法相比所表现的轮廓信息更加丰富.多分辨率模型不同层次之间顶点的变化会导致相应三角形的表面法向发生变化,对于漫反射光照,shading 变化与法向锥半锥角 θ 成正比,通过算法法向锥可以度量出漫反射 shading 的可能的偏差.简化具有较大 θ 角的表面会造成更多的 shading artifacts.而使用半锥角作为误差,使得具有较大半锥角值的顶点比半锥角小的顶点具有更高的优先级而不被简化删除掉,不容易造成 shading 失真,因此,我们的简化准则也具有 shading 保持的特性,适合高空俯瞰地形场景的情形.

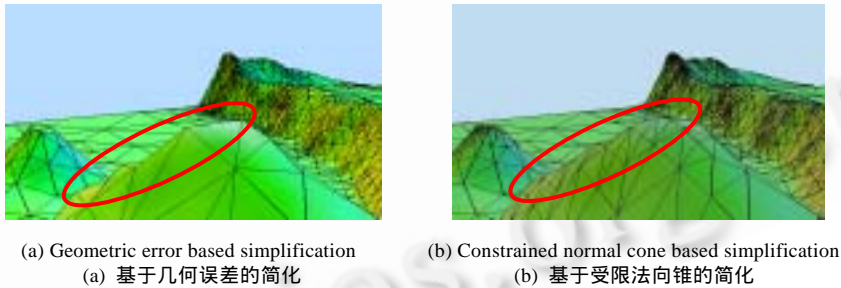


Fig.3
图 3

3.2 潜在轮廓(potential silhouette)

每个内部顶点计算出的法向锥半锥角值可以在预处理阶段用于构建出地形块的三维潜在轮廓.潜在轮廓能大致勾勒出当观察视线方向与 x 坐标轴平行以及与 z 坐标轴平行时的场景轮廓边缘.使用三维线段勾勒出三维场景的轮廓,使得三维线段的透视投影逼近三维表面进行二维透视投影后得到的轮廓线,在地表场景中则为地平线.构建过程首先要对块网格内的点进行采样,采样无须单独进行,在进行块网格简化并构建三角带的过程中,会遍历每个大于简化阈值的顶点,采样可作为遍历这些顶点的辅助过程执行.采样顶点的优先级 $Priority_{(x,z)} \propto height_{(x,z)} \cdot \theta_{(x,z)}$,其中 $height_{(x,z)}$ 为顶点的高程值,而 $\theta_{(x,z)}$ 则为顶点的法向锥半锥角.当 $Priority$ 大于指定阈值时,记录该顶点.然后把采样点集合按照 x 方向排序,并使用启发式搜索算法进行搜索得到二次采样的顺序顶点集合组成的线段.用这些线段逼近视线沿 z 轴平行方向观察时该部分地形场景的轮廓.同样,把采样点集合按照 z 方向排序,并采用搜索算法进行二次采样.所得顺序顶点集组成的连续线段近似视线沿 x 轴平行方向观察时的轮廓.而任意方向视线的二维轮廓线则可以通过对 x, z 方向的两个潜在轮廓透视投影后融合得到.

4 增量地平线以及 out-of-core 数据调度

4.1 增量地平线(Incremental horizon)

地平线技术(horizon)是众所周知的地表模型可见性剔除方法.地平线是不同地表场景部分绘制形成遮挡的融合,与视点距离超过地平线并且完全位于地平线之下的场景部分是确定无疑被遮挡而不可见的,而剩余可见的场景部分将被绘制并且用来更新地平线.预处理完成之后,层次块地形的头部信息包含了块包围盒及潜在轮廓信息.在运行时刻,首先通过场景地形块 LOD 层次的计算确定相对于当前视点活动的结点.然后依照从前向后的遍历场景地形块的顺序,利用每块区域的潜在轮廓构建增量地平线,并以此进行可见性判定.

通过视域裁剪之后的活动地形块要与当前的地平线进行可见性测试.当前地平线由已经绘制的地形块的潜在轮廓在屏幕空间投影,按照从左到右顺序排列的一系列最高点的集合构成,为追踪得到遮挡地平线,我们把它离散化成与屏幕图像精度相同的一系列高度值.地形块与当前地平线的可见性测试过程如下:

(1) 将块网格包围盒顶部朝向观察者的边投影到定制的地平线图像精度空间,并进行扫描线转换,记录其高度并与原先的地平线高度进行比较;

(2) 如果存在某一个高度值大于地平线上相应投影位置的当前高度值,则该地形块是可见的,否则不可见;

(3) 对可见地形块,投影该块网格的潜在轮廓到图像空间并进行扫描线转换,如果投影高度值大于存储地平线的对应当前高度值,则利用新值替换原有值,更新地平线.

地平线是随着从前向后的场景遍历顺序而不断更新的,因而称为增量地平线.增量地平线的构建以及更新在 CPU 上执行完成.增量地平线的递增更新过程如图 4 所示,视域内场景按照从前向后顺序遍历的不同阶段,增量地平线随访问结点而不断更新,图中曲线表示动态生成的地平线.图 4(a)为初始遍历时的状态,图 4(d)为遍历结束时最终地平线的状态.在遍历过程中,所遍历结点的可见性信息都被计算得到,大量的不可见数据结点被剔除掉,这无论对于数据调度还是场景绘制都会起到非常重要的作用.

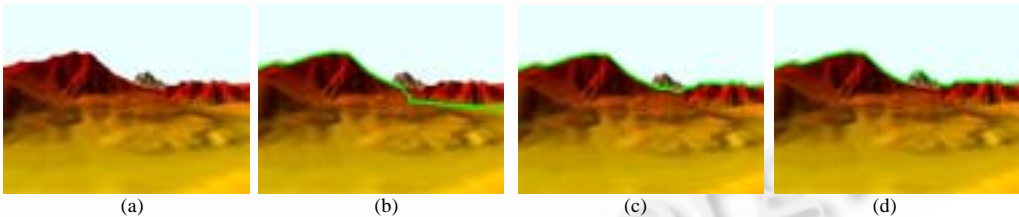


Fig.4 The generation and update process of incremental horizon
图 4 增量地平线的形成与更新过程

4.2 out-of-core数据调度

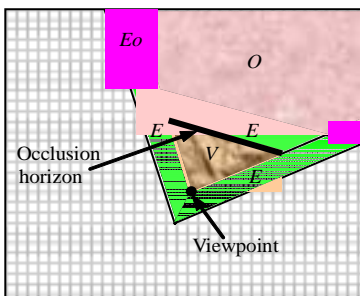


Fig.5 Paging mechanism of terrain
图 5 数据页面调度

超大规模室外地形场景的数据调度由两个因素决定:可见区域和场景的 LOD 层次.实际需要驻留在内存中的场景区域由可见区域和可见区域扩展构成.由于 I/O 属于磁盘操作,往往需要耗费一定的时间,因此调度存在一定的延迟.可见区域扩展是为了适应虚拟地形漫游系统中视点在空间中的六自由度运动,包括上、下、左、右、前、后的移动,以及视线在任意方向上的旋转.它的存在可以避免由于观察快速变化而导致的频繁调度请求.可见,区域扩展的大小由漫游系统中用户对视点的交互操作,包括视线方向矢量旋转的速度以及视点的运动速度来决定.图 5 表示,在我们漫游系统中,需要驻留在内存中的数据页面覆盖了可见区域 V 和可见区域扩展 E .传统算法由于没有有效的可见性信息

指导,所以其需要载入内存的页面覆盖的面积为 $V+E+O+Eo$,其中 O 区域是被地平线遮挡而不可见的区域.可见,我们的系统所需驻留内存的区域要小于传统算法.

在我们的漫游系统中,数据页面调度机制通过双线程实现,在主线程和 I/O 线程之间采用互斥机制使其同步运行协同工作.当进行漫游时,主线程首先利用增量地平线技术求出可见区域和可见区域扩展.对应场景的四叉树组织结构和当前活动的层次细节结点,分别得到可见区域和可见区域扩展覆盖下的活动结点集合.如果活动结点集合中一个活动结点的数据实体并不在内存中,则将该结点加入数据载入请求队列,然后依据结点的误差值作为优先级对整个载入请求队列进行排序.一般情况下,可见区域内活动结点的优先级高于可见区域扩展内的结点,层次较低的结点(粗糙结点)的优先级高于层次较高的结点.I/O 线程则释放不在当前可见区域和可见区域扩展中的结点所对应的实体模型数据,并根据数据载入请求队列的优先级执行外存数据页面的读取.同时,主线程将需要绘制的可见区域内的实体数据送给 GPU 进行处理.GPU 根据得到的不同层次的模型数据,执行有效的基于顶点的几何过渡操作,并将最终的几何网格采用所需绘制方式进行绘制.由于帧与帧之间很强的相关性,实时数据调度过程中只有少数页面需要读入.增量地平线的运用使那些不在可见区域或可见区域扩展的活动结点因为不可见而不需要载入内存,又由于每次装载的数据都是代表某一块地表场景区域的批数据,因此我们基于增量地平线的调度算法可以减小数据调度粒度,节省系统资源的耗费.

5 实验结果与分析

为了测试漫游算法的性能,需要从两个方面入手,一方面是漫游时的帧速率,另一方面是漫游场景的过程中进行数据 I/O 的情况.我们的实验测试所使用的硬件环境为 P4 2.4GHz 的 CPU,内存大小 512M,显卡为 GeForce4.我们采用 4 块地形数据作为实验输入源数据(见表 1).4 块地形分别包含了地形维数、预处理生成的文件大小、真正三角形数目、退化三角形数目(采用三角带技术保存模型所致)以及生成四叉树的深度.对于两块较大的地形模型 Puget Sound 和 Wenatchee 采用了固定路径漫游和自由交互漫游两种实验方式,而对于其余两块较小的地表模型,由于场景较小,所以仅采用自由漫游方式.在进行固定路径的漫游时,对 Puget Sound 和 Wenatchee 分别采样 38 575 帧和 39 136 帧.而对所有模型进行自由交互漫游都采样 10 000 帧.漫游时显示窗口分辨率大小为 1024×768,绘制的方式为几何模型加地表纹理.

Table 1 Terrain models for experiments of navigation

表 1 漫游实验所用的地形数据

	Dimension	File size (M)	Real triangles	Deg. triangles	Tree depth
Crater ILake	513	2.0	283 135	153 541	4
South bay	1 025	10.0	1 460 374	795 744	5
Puget sound	8 193	563.8	71 842 336	38 301 810	8
Wenatchee	8 193	257.9	32 667 175	17 498 037	7

表 2 给出了在整个固定路径漫游期间,载入内存的数据块、从内存卸载的数据块、载入内存数据块所占字节、从内存卸载的数据量的统计值.“IH on”和“IH off”分别代表增量地平线技术开启和禁用两种情况.无论是数据页面数目还是数据的字节数,采用增量地平线技术算法的调度数量都较少,因此表明我们的调度算法要优于以往没有增量地平线支持的调度算法.在实时运行阶段,每块数据页面的大小一般为 25K~35K,一次读取的时间为 2.3ms 左右.在漫游实验中,传统算法由于在内存中驻留了更多的数据块,其内存占用量往往比本文的算法多大约 10%.从表 2 也可以看出,由于可见区域扩展的存在,本文所控制的数据调度的比例并不很高.增量地平线提供可见性信息不仅可以控制数据调度,还可以有效地降低每帧中绘制场景的几何复杂度,从而加速整个地表场景的漫游.这可以从如图 6、图 7 所示的帧速率曲线图和每帧绘制三角形数目曲线图中得到体现.

Table 2 Statistics on geometric data exchange while navigating Wenatchee and Puget Sound

表 2 Wenatchee, Puget Sound 地形漫游内外存几何数据交换统计表

	Puget Sound		Wenatchee	
	IH on	IH off	IH on	IH off
Blocks uploaded	12 944	14 996	2 538	2 807
Blocks unloaded	12 487	14 416	2 211	2 399
Data uploaded (M)	398.9	455.6	71.2	82.9
Data unloaded (M)	381.8	421.8	58.0	66.5

如图 6 和图 7 所示的固定路径漫游,采用我们的方法所获得的帧速率是不采用增量地平线方法^[14]的 2.5 倍~3 倍,与 Lloyd 算法^[7]相比也提升了 50%的加速性能.虽然 Lloyd 算法也采用了地平线技术,但由于我们利用潜在轮廓所构建的增量地平线具有更高的地平线表示精度,并且能够有效地减少所需绘制地形块的数目,因此取得更高的绘制效率和性能是顺理成章的.此外,轮廓保持的简化方法使得在地表漫游时的每一帧都更加符合人类视觉观察的实际需求,能够得到更佳图像效果,这也是以往算法所不具有的特点.

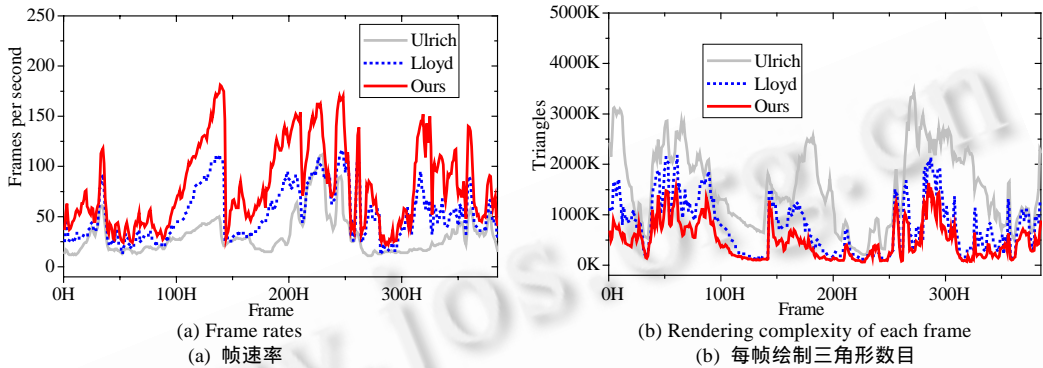


Fig.6 Experiments for Puget Sound with 38 575 frames sampled

图 6 Puget Sound 指定路径漫游(共采样 38 575 帧)

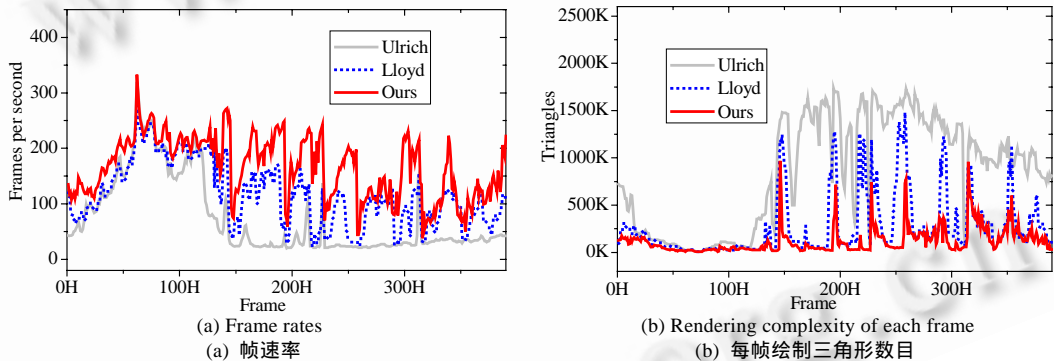


Fig.7 Experiments for Wenatchee with 39 136 frames sampled

图 7 Wenatchee 指定路径漫游(共采样 39 136 帧)

表 3 是所有漫游情况数据统计表.对超大规模 out-of-core 的模型数据,我们的方法加速效果非常明显,但是对于小规模的地形来说,加速效果并不十分突出,甚至出现了帧速率稍有下降的情况.这是因为对小规模的地形场景,I/O 几乎微不足道,每帧绘制的三角形数目又较少,耗费一定的时间生成增量地平线并进行可见性判定并不会获得很大的加速效率,当加速效率不能弥补付出的 CPU 计算代价时,帧速率就会下降.因此,对超大规模并具有较多遮挡的地形场景漫游的实时加速,是我们的方法优势之所在.此外,对外存中的纹理数据,也可以采用与几何模型类似的调度策略和 LOD 计算方法.

阴影能够增强室外场景的真实感程度,为观察者提供 stronger 的视觉表现力.Shadow map 是一种实时生成阴影的方法^[21].它是一种基于图像的生成方法,由于矩阵变换和深度比较可以很快速度由当前主流图形硬件实现,所以它可以快速生成阴影.可见性的计算是任何阴影生成算法整体的一部分.给定一个点光源和视点,只需要找到视点可见的表面和相对于光源可见的表面,就可以计算出阴影.然而,对于交互应用程序,精确计算出可见表面非常困难且耗时巨大.尽管针对大规模的虚拟场景,Govindaraju 提出了快速计算 PVS_E (相对于视点可见物体集合)和 PVS_L (相对于光源可见物体集合)的算法^[22],并利用 cross culling 技术进一步减少阴影计算量以产生精确的本影,但是其基于多处理器运行框架的方法使人望而生畏,而室外地形场景有其结构特殊性,因此我们利用增

量地平线所确定的可见集来进行大规模室外场景的实时阴影生成.

Table 3 Statistics on frame rates and rendering complexity while navigating

表 3 地形漫游平均帧速率与平均每帧绘制三角形数目统计

Approach Models	Our approach (Course)		Ulrich (Course)		Lloyd (Course)		Our approach (Interactive)		Ulrich (Interactive)	
	Frame rate (fps)	Triangles per frame (Kpf)	Frame rate (fps)	Triangles per frame (Kpf)	Frame rate (fps)	Triangles per frame (Kpf)	Frame rate (fps)	Triangles per frame (Kpf)	Frame rate (fps)	Triangles per frame (Kpf)
Puget Sound	80.7	479.5	31.3	1559.0	52.8	770.2	70.9	449.3	28.4	1702.7
Wenatchee	169.6	121.5	73.1	845.7	116.0	277.8	102.8	235.5	49.3	1134.2
South Bay	x	x	x	x	x	x	243.5	119.0	221.3	193.5
Crater Lake	x	x	x	x	x	x	390.2	66.7	454.0	72.3

本文中, PVS_E 就是在进行场景的增量地平线动态遮挡剔除之后得到的相对于视点可见场景部分的集合, 这样得到的 PVS_E 是一个保守集合, 但这已经使场景复杂度降低了 1 个甚至更高的数量级. PVS_L 的计算是在 PVS_E 基础之上采用近似计算方法得到. 由于地形地貌有其特殊性, 它的阴影生成都是场景不同部分之间相互遮挡形成的, 因此, 我们先把 PVS_E 的所有元素加入到 PVS_L 之中, 然后对位于 PVS_L 边缘部分的场景, 依照太阳(光源)的方向快速访问临近的场景部分, 如果其临近场景部分的潜在轮廓易于形成遮挡, 则将该部分场景加入到 PVS_L 之中. 通过以上方法可以近似计算所有 PVS_L 结点. 根据 shadow map 的基本原理先进行相对于光源的绘制, 然后采用相对于视点的两遍绘制方法, 生成整个场景具有真实感效果的阴影. 同时, 前面轮廓保持的简化技术, 使得投影者投影形成的边缘具有较高的保真度, 此外, 采用反混淆技术^[23,24]也可以尽量减少阴影边缘的锯齿效果和失真, 图 8 是太阳光(平行光)从不同方向照射超大规模复杂地形场景所生成的阴影图.

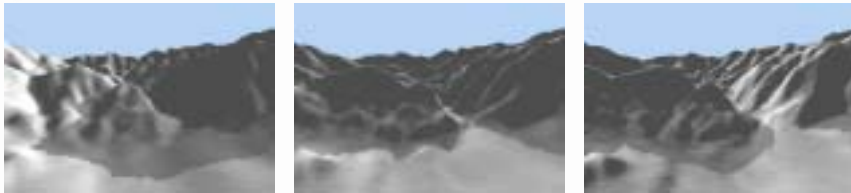


Fig. 8 Snapshots of navigating with shadow from different lighting directions

图 8 太阳光从不同方向照射得到的阴影效果图

6 结 论

本文提出一种高性能的 out-of-core 地形场景实时漫游技术, 该技术可以有效地加速室外场景的绘制. CPU, GPU, I/O 三者有机地结合, 共同完成超大规模地形的实时漫游, 具有低 CPU 负载高三角形吞吐量的特点. 同时, 基于受限法向锥的误差计算方法所实施的场景简化提供了比以往更强的图像表现力, 对地表场景中所有容易构成轮廓的区域部分包括山脊部分、山谷的边界线等提供了很好的显示效果. 因为内部区域冗余的细节被转移, 用来增强轮廓部分的细节, 这更加符合观察者漫游时的视觉需求. 而对于十分平坦的地形场景, 本文的简化方法并不比其他方法具有更多的优越性. 本文还利用增量地平线与硬件支持的 shadow map 技术为超大规模场景实时生成阴影, 进一步增强了场景的真实感程度. 但是, 实际应用中三维地形场景不仅包含表面二维纹理还包含地表的大量三维建筑物和植被, 此时本文的离线预处理以及光照阴影算法难以适用, 因此探索此类更复杂场景的实时绘制是今后我们研究工作的重点. 此外, 我们还将着重考虑进行包括地表光照、天空大气散射等更加复杂因素的室外地形场景的真实感绘制.

致谢 感谢华盛顿大学 Mountain Drainage Basin 研究组提供大规模地形数据 Puget Sound 和 Wenatchee 下载, 以及 VTP 组织的 Crater Lake 和 South Bay 数据. 同时感谢 B. Lloyd 有关地平线地形加速绘制的建议和帮助.

References:

- [1] Cline D, Egbert P. Terrain decimation through quadtree morphing. *IEEE Trans. on Visualization and Computer Graphics*, 2001,7(1):62–69.
- [2] Duchaineau M, Wolinsky M. ROAMing terrain: Real-Time optimally adapting meshes. In: Yagel R, Hagen H, eds. *Proc. of the IEEE Visualization*. Los Alamitos: IEEE Computer Society Press, 1997. 81–88.
- [3] Hoppe H. Smooth view-dependent level-of-detail control and its application to terrain rendering. In: Ebert DS, Rushmeier H, Hagen H, eds. *Proc. of the IEEE Visualization*. Los Alamitos: IEEE Computer Society Press, 1998. 35–42.
- [4] Lindstrom P, Koller D, Ribarsky W, Hodges L, Faust N, Turner G. Real-Time, continuous level of detail rendering of height fields. *ACM Computer Graphics (SIGGRAPH'96)*, 1996,30(3):109–118.
- [5] Pajarola R. Large scale terrain visualization using the restricted quadtree triangulation. In: Ebert DS, Rushmeier H, Hagen H, eds. *Proc. of the IEEE Visualization*. Los Alamitos: IEEE Computer Society Press, 1998. 19–26.
- [6] Röttger S, Heidrich W, Slusallek P. Real-Time generation of continuous levels of detail for height fields. In: Skala V, ed. *Winter School in Computer Graphics, WSCG'98*. Plzen: Science Press, 1998. 315–322.
- [7] Lloyd B, Egbert P. Horizon occlusion culling for real-time rendering of hierarchical terrains. In: Gross M, Joy KI, Moorhead RJ, eds. *Proc. of the IEEE Visualization*. Boston: IEEE Computer Society Press, 2002. 403–410.
- [8] Li S, Liu X, Wu E. Feature-Based visibility-driven CLOD for terrain. In: Rokne J, Klein R, Wang W, eds. *Proc. of the Pacific Graphics*. Canmore: IEEE Computer Society Press, 2003. 313–322.
- [9] Stewart J. Hierarchical visibility in terrains. In: Dorsey J, Slusallek P, eds. *Eurographics Workshop on Rendering*. Vienna: Springer-Verlag, 1997. 217–228.
- [10] Zaugg B, Egbert P. Voxel column culling: Occlusion culling for large terrain models. In: Ebert D, Favre JM, Peikert R, eds. *Proc. of the Joint Eurographics-IEEE TCVG Symp. on Visualization*. Vienna: Springer-Verlag, 2001. 85–93.
- [11] Levenberg J. Fast view-dependent level-of-detail rendering using cached geometry. In: Gross M, Joy KI, Moorhead RJ, eds. *Proc. of the IEEE Visualization*. Los Alamitos: IEEE Computer Society Press, 2002. 259–266.
- [12] Puppo E. Variable resolution terrain surfaces. In: Fiala F, Kranakis E, Sack J, eds. *Proc. of the 8th Canadian Conf. on Computational Geometry*. Ottawa: Carleton University Press, 1996. 202–210.
- [13] Cignoni P, Ganovelli F, Gobbetti E, Marton F, Ponchino F, Scopigno R. BDAM-Batched dynamic adaptive meshes for high performance terrain visualization. *Computer Graphics Forum*, 2003,22(3):505–514.
- [14] Ulrich T. Rendering massive terrains using chunked level of detail control. *Course Notes of ACM SIGGRAPH 2002*. Volume Course 35, 2002.
- [15] Asano T, Ranjan D, Roos T, Ewzl E, Widmayer P. Space filling curves and their use in the design of geometric data structures. *Theoretical Computer Science*, 1997,181(1):3–15.
- [16] Lindstrom P, Pascucci V. Terrain simplification simplified: A general framework for view-dependent out-of-core visualization. *IEEE Trans. on Visualization and Computer Graphics*, 2002,8(3):239–254.
- [17] Cohen-Or D, Chrysanthou Y, Durand F, Greene N, Koltun V, Silva C. Visibility: Problems, techniques, and applications. *Course Notes of ACM SIGGRAPH 2001*. Volume course 30, 2001.
- [18] Stewart J. Fast horizon computation at all points of a terrain with visibility and shading application. *IEEE Trans. on Visualization and Computer Graphics*, 1998,4(1):82–93.
- [19] Johnson D, Cohen E. Spatialized normal cone hierarchies. In: Hughes JF, Séquin CH, eds. *Proc. of the ACM Symp. on Interactive 3D Graphics*. New York: ACM Press, 2001. 129–134.
- [20] Shirman L, Abi-Ezzie S. The cone of normals technique for fast processing of curved patches. *Computer Graphics Forum (Eurographics'93)*, 1993,12(3):261–272.
- [21] Williams L. Casting curved shadows on curved surfaces. *Computer Graphics*, 1978,12(3):270–274.
- [22] Govindaraju N, Lloyd B, Yoon SE, Sud A, Manocha D. Interactive shadow generation in complex environments. *ACM Trans. on Graphics*, 2003,22(3):501–510.
- [23] Reeves WT, Salesin DH, Cook RL. Rendering antialiased shadows with depth maps. *Computer Graphics (Proc. of ACM SIGGRAPH'87)*, 1987,21(3):283–291.

[24] Fernando R. GPU Gems: Programming Techniques, Tips and Tricks for Real-Time Graphics. Boston: Addison-Wesley Professional, 2004.



李胜(1974 -),男,博士,主要研究领域为计算机图形学,虚拟现实技术.



刘学慧(1968 -),女,博士,副研究员,CCF高级会员,主要研究领域为计算机图形学,虚拟现实技术.



冀俊峰(1967 -),男,博士生,主要研究领域为计算机图形学,数字几何处理与绘制.



吴恩华(1947 -),男,博士,研究员,博士生导师,CCF高级会员,主要研究领域为计算机图形学,可视化,虚拟现实.

第 3 届智能 CAD 与数字娱乐学术会议(CIDE 2006)

征 文 通 知

2006 年 10 月 22-23 日, 济南, 中国

Second Conference on Intelligent CAD and Digital Entertainment

October 22-23, 2006, Ji'nan, China

由中国图象图形学学会计算机动画与数字娱乐专业委员会和中国人工智能学会智能 CAD 与数字艺术专业委员会联合主办, 山东大学承办的第三届智能 CAD 与数字娱乐学术会议(CIDE2006), 将于 2006 年 10 月在济南举行。

征稿范围 (不仅限于如下主题)

智能 CAD	数字艺术	计算机动画	数字内容管理
虚拟现实	网络游戏	可视化技术	E-home
模式识别	人机交互	计算机图形学	
图像处理	信息融合	人脸表情跟踪与识别	
多媒体技术	计算机视觉	人工智能	
交互式玩具	运动捕获动画	数字博物馆	

投稿要求

论文具体格式请访问 <http://cide2006.sdu.edu.cn>

电子投稿, 请将 WORD 格式的文件发到: cide2006@sdu.edu.cn

网络投稿: <http://cide2006.sdu.edu.cn>

重要日期

全文投稿: 2006 年 6 月 10 日

录用通知: 2006 年 6 月 30 日

修改定稿: 2006 年 7 月 20 日

联系方式

联 系 人: 王璐 黄月珠

联系电话: 0531-88364810

E-mail: cide2006@sdu.edu.cn