基于四叉树分割的地形LOD技术综述

王振武 吕小华 韩晓辉
(中国矿业大学(北京)计算机科学与技术系 北京 100083)

摘 要 层次细节(Levels of Detail,LOD)技术是在大规模地形模型简化方面使用得最多的技术,它极大地提高了地形场景的漫游速度。在众多LOD模型中,应用最为广泛的是基于四叉树(Quadtree)分割的LOD算法。国内外学者对LOD模型做了大量的研究工作,文中对基于四叉树分割的LOD算法进行了系统的梳理与总结,对涉及到的核心算法进行了归类并详细分析了各自的优缺点,深入且全面地介绍了其研究现状。

关键词 快速绘制,LOD技术,四叉树,地形

中图法分类号 TP306.1 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.04.005

Survey of Terrain LOD Technology Based on Quadtree Segmentation

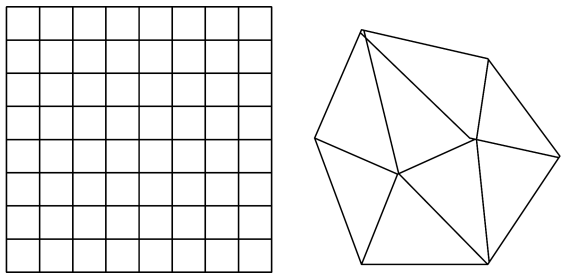
WANG Zhen-wu LV Xiao-hua HAN Xiao-hui
(Department of Computer Science and Technology,China University of Mining & Technology,Beijing,Beijing 100083,China)

Abstract Level of detail(LOD) technology is the most widely adopted technology in the model simplification of large-scale terrain,which greatly accelerates the speed of roaming terrain. Among all the LOD methods,the most popular one is LOD algorithm based on quadtree segmentation. Many domestic and foreign scholars have done lots of research work on the LOD technology. In this paper,the quadtree based LOD algorithms were sorted and summarized systematically,several key algorithms were classified and analyzed,their advantages and disadvantages were analyzed in details,and their current research status were introduced comprehensively.

Keywords Fast roaming,LOD technology,Quadtree,Terrain

1 前言

自20世纪90年代以来,三维地形场景的漫游一直是计算机图形学与虚拟现实研究领域的一个热门课题[1],它在虚拟仿真、场地模拟、电子地图和地质灾害等众多领域都有着至关重要的应用。地形是一种极其复杂的对象,因此对具有真实感的地形场景进行构建与快速绘制一直是学术界的难点问题。LOD技术成为解决该难题最可行的方法之一。在复杂模型的动态显示过程中,需要重点突出的物体图形会在屏幕上占据更多的像素点,而那些不需要突出显示的图形则不需要用大量的多边形进行精确描绘,即对不同的物体要用不同的精度展现。LOD技术就是根据一系列的评价标准,选择不同的精度来实现实时漫游的一种行之有效的方法。目前,对大规模地形数据的模型表示主要有以下两种:规则格网(Regular Square Grid,RSG)模型(见图1(a))与非规则格网(Triangulated Irregular Network,TIN)模型(见图1(b))。



(a) 规则格网 (b) 非规则格网

图1 地形数据的两种格网模型

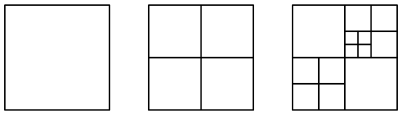
Fig. 1 Two grid models of terrain data

规则格网一般用正方形表示,每一个单元格网都有与之一一对应的高程值,该模型是将地形数据通过高程矩阵的方式来表示,具有结构简单、存储量小、易于裁剪简化以及便于动态处理等优点。生成规则格网的常用方法主要有两种,即四叉分割与二叉分割。其中,基于四叉树的LOD算法是采用构造四叉树模型的方式对地形进行组织(见图2(a)),在地形

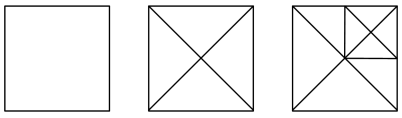
到稿日期:2017-03-01 返修日期:2017-06-28 本文受国家自然科学基金项目(61363075),国家863项目(2012AA12A308),中国矿业大学(北京)越琦青年学者计划项目(800015Z1117)资助。

王振武(1978-),男,博士,副教授,CCF会员,主要研究方向为图形图像处理、数据挖掘,E-mail:wangzhenwu@126.com(通信作者);**吕小华**(1992-),女,硕士生,主要研究方向为图形图像处理;**韩晓辉**(1992-),女,硕士生,主要研究方向为图形图像处理。

渲染时，通过递归遍历四叉树的方式，根据所设定的节点评价系统来判断是否需要加载下一个四叉树子节点的地形数据块。Lindstrom[2] 于 1996 年所提出的 LOD 模型算法是典型的采用四叉分割的方法。另一种常用的二叉树 LOD 方法是通过采用斜边二分法将一个等腰三角形划分为两个相同的三角形（见图 2(b)）。ROAM 算法[3] 是该模型的典型代表，它是对 Lindstrom 算法的进一步改造，具有较优的实时性能，但算法复杂度较高。



(a)基于块的四叉分割



(b)基于三角形的二叉分割

图 2　规则格网的两种分割方式

Fig. 2　Two segmentation methods for regular grid

非规则格网是根据非规则分布的数据点来构造三角网模型，以仿真地表。基于非规则格网的 LOD 算法只需要绘制少量的三角形面片，但它的拓扑结构相当复杂，对处理器和内存的要求很高，绘制模型需要较长时间，效率很低，也较难与视相关的实时漫游处理相结合，因此这方面的研究相对较少，绝大多数地形渲染算法还是基于规则格网的模型来构造的。基于规则格网的模型算法层次清晰、结构规范，容易构建视相关的实时渲染模型；但基于树形结构的构建还是会对系统的运行造成一定的负担，在渲染的过程中也可能会在相邻数据块之间出现不同程度的裂缝，因此在规则格网的 LOD 模型中对裂缝的处理也是不可避免的。

本文对基于四叉树分割的 LOD 算法进行了系统梳理与总结，对涉及到的核心算法进行了分类并分析了各自的优缺点。基于四叉树分割的 LOD 算法的大致流程如图 3 所示。

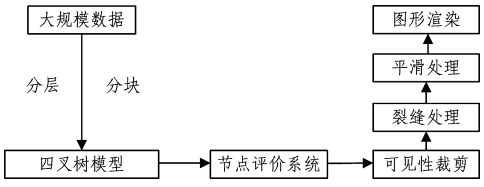


图 3　LOD 算法流程图

Fig. 3　LOD algorithm flowchart

根据国内外现有文献以及算法的操作流程，将基于四叉树分割的 LOD 算法的研究重点大致分为 4 个方面：构造四叉树结构方法的改进[4-14]、节点评价系统的改进[2,9,15-35]、视域剔除方法的改进[23,25,36-42] 以及裂缝修复方法的改进，如图 4 所示。另外，文中还简要叙述了关于平滑处理和基于 GPU 处理的 LOD 模型技术的研究内容及现状。

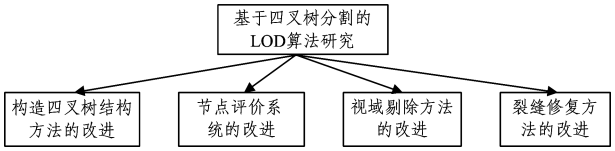


图 4　基于四叉树分割的 LOD 算法的研究分类

Fig. 4　Classification of research on LOD algorithm based on quadtree segmentation

2　四叉树结构的构建

一般来说，对大规模地形数据进行处理时，都会先将整体数据划分成多个小型数据块，并按照层次模型进行组织。通过运用这种分而治之的思想，对模型进行实时漫游实质就是对各层级的小型数据块进行处理与渲染。四叉树分割是最常用和最经典的剖分方法，此方法要求地形的格网数必须是数据行列的取样数目，对于任意格网数的地形，一般是通过补充空值或减少地形格网来适应这一条件。为了避免丢失数据或浪费内存，也出现了其他的处理方法，如文献[4]把研究区域分割成多个边长为定值的正方形子块，当待划分数据区域不足一个正方形时，向包含它们的最小正方形添加空白行或列，如图 5 所示。其中虚线边界内为原数据区，实线边界内为分块后的数据区。

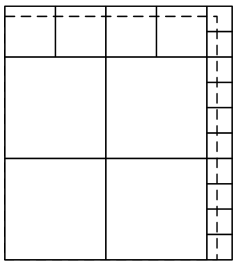
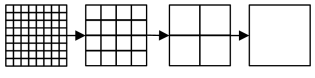


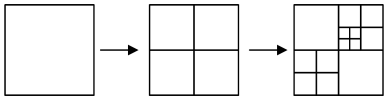
图 5　任意矩形区域的数据分块示意图

Fig. 5　Data blocking of arbitrary rectangular region

四叉树结构模型的构造方法可以分为两种：自底而上和自顶而下。简单来说，自底而上的方式就是使规则格网从四叉树最底层节点逐步向上粗化，直到达到所需要的精度（见图 6(a)）；自顶而下的方式就是使规则格网从四叉树根节点逐步向下细化，直到符合精度要求（见图 6(b)）。文献[2]提出了连续层次细节算法，该算法就是采用自底而上的简化方法生成连续的动态三维地形模型。Geomipmapping[5] 算法是以自底而上的方式构造分层结构的典型分块算法，与自顶向下的细化方法相比，该方法的复杂度由最细致的地形格网的级别决定，效率较低。



(a)自底而上的方式



(b)自顶向下的方式

图 6　构建四叉树的分层结构

Fig. 6　Layered structure of constructing of quadtree

Von Herzen[6]提出把约束四叉树结构应用到 LOD 技术中，在传统的利用自顶向下的方式构造四叉树结构的基础上添加了一个限制条件，即相邻数据块层级差不超过 1。这种结构在一定程度上避免了裂缝的产生，从而节省了后期大量修复裂缝的时间。文献[7]参考 Geomipmapping 算法，先按照完全四叉树对整个地形高程数据进行划分，然后再进行各种操作，包括降采样、缩放和平移变换等，从而构造合适的地形块金字塔模型。为了方便后续对数据块间裂缝的处理，文献[8]提出了一种新的四叉树划分方法，它要求相邻瓦片共享一行一列，如图 7 所示，9×9 的地形块别分为 4 个 5×5 的地形瓦片。实验证明，此方法大大改善了裂缝修复的效果，但是也占据了大量多余的内存空间。

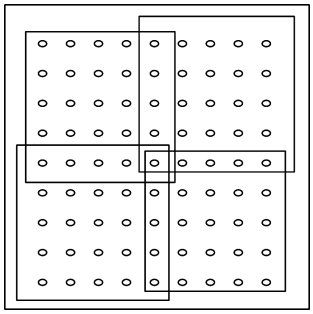


图 7 四叉树分块

Fig. 7 Quadtree blocks

文献[9]在文献[2]的基础上，先按照 Mortan 码的编码规则对原始地形数据进行无损压缩，而后再自下而上地构建不完全四叉树模型，实验证明该方法能有效地简化模型。文献[10]为了提高四叉树地形数据的存取效率，对四叉树组织结构采用逻辑四叉树结构，而非物理四叉树结构。另外，文献[11]提出了动态数据子块的概念，采用规则格网模型将动态子块进行有效组织，最后运用改善后的固定四叉树建立地形模型，此方法易操作、灵活性好、绘制速率高。文献[4]提出了一种新的处理策略，它将数据进行分块处理，再对数据块内部进行分层，使得只有少量数据可以进入内存中。文献[12]在预处理阶段建立了两棵独立的限制性四叉树，分别为数据树和逻辑树。数据树用来存储地形数据；逻辑树用来存储操作数据树的信息，包括 LOD 层次划分的节点信息和视域剔除的相关信息等。另外，该方法在运行过程中加入了硬件的处理，渲染速度表现出较好的优势。鉴于 ROAM 算法较好的连续性和绘制速率，文献[13]在简单实现基于四叉树 LOD 算法的基础上引入 ROAM 算法的数据块分割策略，实验证明改进后的 LOD 算法的渲染帧频较改进前有显著提高。文献[14]构造了一种视相关的约束性四叉树结构，并对后续节点评价系统的建立和裂缝修复环节进行合理处理，最终得到了较好的视觉效果。有的文献[43]采用菱形块的剖分办法，它实际上是规则格网四叉树分割的变形体，实验证明其比传统的四叉分割的简化效果更好。

综上所述，对于基于四叉树分割的 LOD 算法而言，四叉树模型的构造起着至关重要的作用，它的优劣直接影响到后续评价系统、视域裁剪及裂缝修复等过程的处理，因此需要根据数据的规模和特点以及后续数据处理的复杂程度等来合理构造四叉树模型。

3 节点评价系统

地形简化是地形可视化技术的重点研究内容，它使得大规模数据模型能够进行多分辨率显示，可达到“越近越精细绘制，越远越粗糙绘制”的模型实现效果。

一般通过建立节点评价系统来对模型进行合理的分层或分级，从而使得模型能够多分辨率显示，该方法是一种多分辨率技术。影响节点评价效果的因素有：地形块、视距、地形粗糙度、视角、运动矢量以及屏幕投影误差等。Rotteger[15]提出了地形格网的细分原则：地形特征复杂和距离试点较近的区域需要细分，否则可以不细分；且在格网细化上进一步指出了两种常用的地形渲染评价因子，即地形块与视点的距离、地形块的粗糙度。此后，众多学者对节点的评价进行了改进。本文对节点评价系统的相关文献进行了系统梳理，节点评价方法的大致分类如图 8 所示。

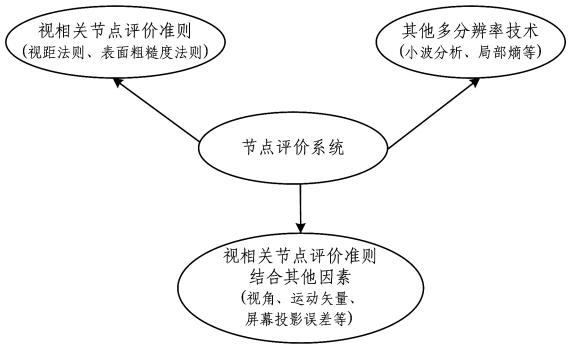


图 8 节点评价系统

Fig. 8 Node evaluation system

3.1 视相关节点评价准则

3.1.1 视距法则

视距指视点到四叉树节点的距离(见图 9)。视距法则主要是根据视距值的大小来确定绘制某地形节点的分辨率(细化程度)，即离视点较远的节点使用低分辨率的地形块渲染，较近的节点则使用高分辨率的地形块渲染。基于视距的评价函数如下：

$$f=\frac{l}{d\times C} \tag{1}$$

其中，视距 l 是视点(x_0,y_0,z_0)与目标节点即中心点(x_1,y_1,z_1)的距离，即：

$$l=\sqrt{(x_1-x_0)^2+(y_1-y_0)^2+(z_1-z_0)^2} \tag{2}$$

为避免计算时开方或乘方操作的复杂性，文献[16]采用如下的计算方法来计算 l：

$$l=|x_1-x_0|+|y_1-y_0|+|z_1-z_0| \tag{3}$$

图 9 中，d 表示该节点处地形块边的长度；C 表示控制整体场景最小分辨率的常量，用来调整地形层次细节模型精度的常数阈值。C 越大，表明需要参与当前场景绘制的地形块节点越多，当前场景的地形模型的分辨率越高。设定一个误差控制阈值 τ，同时通过式(1)计算得到一个 f 值，若 $f<\tau$，则对该地形块节点进行细分操作，即索引到该地形块节点的子节点，直到 $f\geqslant\tau$。

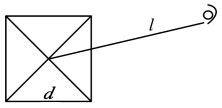


图 9　视距表示

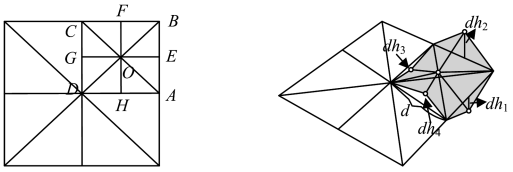
Fig. 9　Sight distance representation

3.1.2　表面粗糙度法则

表面粗糙度法则是指对地形数据起伏程度大的区域块用较高的分辨率渲染，对起伏程度小的区域块用较低的分辨率渲染，这样就可以用很少的三角形面片来仿真真实的地形。区块内的高差值反映了地形的起伏程度，它的数学表示多种多样。

如图 10 所示，图(a)和图(b)分别为同一个地形块在不同观察视角下的表示形式。图(a)是地形块的平面显示，图(b)的灰色部分与图(a)中的 $ABCD$ 区域相对应。其中，d 为该地形块的边长；dh_1，dh_2，dh_3 和 dh_4 分别是各地形块边的中点处的高程值和边界端点高程值的平均值之差，即地形块中心坐标的高程值和边界平均高程值的差值；dh_5 是点 A 和点 C 坐标值的平均值与点 O 的高程值之差；dh_6 同理。利用式(4)来构造 d_0，从而度量该地形块表面的起伏形态。

$$d_0=\frac{1}{d}\max_{i=1,2,\cdots,6}|dh_i| \tag{4}$$



(a)　(b)

图 10　表面粗糙度

Fig. 10　Surface roughness

综合视距法则和表面粗糙度法则的评价函数如式(5)所示：

$$f=\frac{l}{d\cdot C\cdot\max(k\cdot d_0,1)}<1 \tag{5}$$

其中，k 为常量。文献[17]利用节点中心点及矩形中的 4 个顶点间的平均距离来计算视点和节点间的长度，即：

$$\bar{l}=\sum_{i=0}^{i=4}l_i \tag{6}$$

该方法同时取节点的高度的方差来判别地形的粗糙程度：

$$rsd=\sqrt{\sum_{j=0}^{j=n}(h_j-\bar{h})^2/n} \tag{7}$$

综合以上两个因素的评价函数为：

$$f=\frac{\bar{l}}{rsd\cdot C_1\cdot C_2} \tag{8}$$

其中，C_1 和 C_2 均为控制因子。而文献[18]所提出的动态 LOD 地形模型，是在构造节点评价函数时，采用 9 个中心点与临界处的点的相对标准方差来表示控制因子。文献[2]提出了基于块的简化准则，它通过自顶而下的遍历顺序，采用地形的整体粗糙度来决定节点的误差大小。文献[19]在文献[2]的基础上提出一种新的粗糙度计算方法，即最小二乘法，其数学式如下：

$$\mu=\frac{1}{(x_{\max}-x_{\min})(y_{\max}-y_{\min})}\sqrt{\sum_{x=x_{\min}}^{x_{\max}}\sum_{y=y_{\min}}^{y_{\max}}(z_{x,y}-\bar{z})^2} \tag{9}$$

其中，μ 是一个常量值，当视点的位置变化时，它不会随之改变；其计算过程只需在前期完成，不需要后期的持续更新。总体来讲，此方法的绘制速度较快，缺点是它的计算量仍较大。文献[20]用一个小格网上的 4 条边的中点高程值与中点位置实际高程值之差的绝对值作为此数据块的地形起伏表示，再取其中的最大值作为该地形块的粗糙度。文献[21]在考虑地形坡度的因素时，用图 11 所示的 8 个中点误差的最大值 $\max(e_1,\cdots,e_8)$ 来代表三维地形粗糙陡度的变化情况，所建立的节点分层评价函数为：

$$f=C\frac{\max(e_1,\cdots,e_8)}{f(d)} \tag{10}$$

其中，C 为评价函数调节参数，d 是视点与目标节点的欧氏距离。

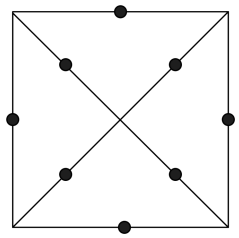


图 11　三维地形的中点误差

Fig. 11　Point error in 3D terrain

由于各节点之间所表示的矩形地形区域的面积各不相同，因此仅将面积作为计算的标尺会造成不必要的误差。文献[29]主要考虑地形块的节点误差和视点位置两个因素，构造了一种新的节点评价计算式：

$$\begin{aligned}L_1&=\max(\mathrm{abs}(vertx-viewx),\mathrm{abs}(verty-viewy),\\&\quad\mathrm{abs}(vertz-viewz))\\enabled&=error\times Threadshold<L_1\end{aligned} \tag{11}$$

其中，L_1 为观察点与节点之间的相对距离，$error$ 表示此处的评判差错，$Threadshold$ 代表门限值，$vert(x,y,z)$ 为节点中心的坐标值，$view(x,y,z)$ 则表示观察点的坐标值。当 $enabled$ 为真时，继续细化该节点，否则停止细化。通过调整 $Threadshold$ 的值可以得到较好的地形显示效果，但这样做可能会人为造成一定的误差。

3.2　视相关节点评价准则与其他因素相结合

国内外众多学者在考虑传统节点评价方法的基础上，也融入了其他的相关因素。有些文献[22-23,44]将运动矢量和观察矢量纳入考虑范围，把传统的评价计算式改进为：

$$f=\frac{l\cdot v}{d\cdot c_1\cdot\max(c_2\cdot d_0,1)\cdot c_3\cdot c_4\cdot v_p}<1 \tag{12}$$

其中，c_1，c_2，c_3，c_4 为各因素的调节因子，v 表示运动矢量，v_p 表示观察矢量。文献[30]在借鉴 Rotteger[15] 观点的基础上，采用了基于视相关的细分评价准则，除了考虑视点外，还考虑了模型与实现在不同轴向的夹角的影响。文献[24]在考虑视点距离及视点移动速度的动态因素和地表静态粗糙度的基础上，构建了一个动静结合的评判元素，达到了实时调度地形数据的目的，以保证图像的交互性帧率。

文献[25]考虑了粗糙度和最小尺寸两个因素，通过透视

变换的原则(见图 12)充分考虑屏幕投影所带来的误差,利用式(13)来决定是否继续细化 LOD 的层次。

$$\tau=\frac{l\times L\times\lambda}{2\times\tan(\alpha/2)\times d} \tag{13}$$

由于原始的屏幕误差式会带来较大的时间代价,因此简化式为:

$$error(i)\approx\frac{k\cdot L}{\alpha\cdot D}(object-error(i)) \tag{14}$$

其中,D 为视点到数据块中心点的距离,L 表示该节点所在数据块边长的最小值,k 为控制因子,α 为屏幕观察视角,$object-error(i)$为第 i 个子数据块的粗糙度误差。文献[14]同样在透视变换原则的基础上,结合地形表面特征和视点相关等影响元素,把误差评估的准则定义为:

$$e(C_i)\approx\frac{k'\cdot L}{X\cdot D}\cdot S(C_i) \tag{15}$$

其中,$S(C_i)$代表节点的静态评估误差。文献[27-28]在传统节点评价方法的基础上考虑了屏幕投影误差,构造了各自的评价系统。文献[9]按照透视变换的原则构造出一种新的节点评价函数。文献[26]通过对近年来的各种评价系统进行系统钻研,也考虑了多种相关因素,对文献[9]中的评价函数进行了改进,从而简化了计算表达式,大大减少了节点判断的次数,从而使得四叉树的细化速度得到了很大的提高。

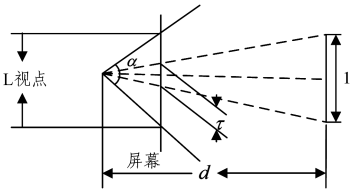


图 12 透视变换的原理

Fig. 12 Principle of perspective transformation

3.3 其他多分辨率技术

鉴于小波分析在时频和空频局部化方面具有较好的特性,文献[31]运用它来多分辨率地处理纹理影像块,实现大规模纹理影像的分层与分块存储,从而构造出离散的四叉树结构。为了加快纹理图像压缩的速度,并保证其质量,根据小波分解时随着分辨率级别的变化其能量分布呈现不均匀的特点,可以很方便地对各分辨率级进行多层次处理。文献[32,45]也构造了基于小波分析的层次地形模型算法,并在此基础上对各层级做了一定精度的简化压缩,然后再构成四叉树的各节点结构。文献[33]通过引入熵的概念,提出了一种局部熵的新节点评价机制,构造出一系列相关计算式来操作数据块节点,从而很容易地确定出它的显示精度。该机制使得算法的复杂度大幅降低,同时也具有一定的滤波平滑效果。为了确定四叉树结构中每个节点是否被分割,文献[34]提出一种适用于视点相关几何图像的节点误差测度函数,此函数可以用于确定不同数据块的显示精度,从而有效地对模型进行简化。文献[35]选择图像聚焦因子的差错判断准则作为节点评判的依据,该方法有效地结合了焦点与投影误差。文献[46]充分考虑四叉树节点与其子节点之间的拓扑关系,并把它运用到 LOD 模型的简化过程中,同时结合基于外存的调度技术,达到了较好的地形仿真效果。文献[47]在经典 Chunked LOD 算法的基础上,运用改进的 Kobbelt 插值方法实时地添加富有真实感的高分辨率格网,以满足逼真视效的需求。

3.4 小结

综上所述,LOD 算法的多分辨率技术也是 LOD 技术的重点研究内容。评价系统的构造直接关系到整个 LOD 模型的简化效果和用户体验。出于时间与空间复杂度的考虑,很多学者简单地将视距法则和表面粗糙度法则结合来构造评价准则,由于评价计算式的构造方法与参数不同,模型绘制的效果会受到一定影响。不可否认,这种做法确实略显片面,因此国内外众多学者在前人的基础上综合了其他的相关因素,包括屏幕误差、运动矢量和观察矢量等。评价系统的考虑因素越多,就能得到越真实和精确的视觉体验,但同时也会对处理器和内存造成一定的挑战,因此必须兼顾仿真效果和绘制速率才能构造出实用的实时漫游系统。也有学者把小波分析等一些新的概念和思路应用到多分辨率模型中,拓宽了 LOD 技术在三维可视化领域的研究范畴。

4 视域剔除

视域剔除的目的是在图形的渲染过程中尽可能地将无效的多边形剔除,它在复杂场景的漫游中至关重要,可以极大地减少多边形面片的绘制数量,进而减轻显卡存储的负担。视域剔除方法[42]的主旨为:一个图形元素如果在视域平头四棱锥(View Frustum,VF)(见图 13)内部,则保留;否则为不可见,需要剔除。视域平头四棱锥包含 6 个面,即上下、远近和左右裁剪面,其中每个平面的数学方程表示为 $Ax+By+Cz+D=0$。通过人为定义指向 VF 内部的方向为平面的正方向,把顶点的坐标值代入 6 个方程中,然后通过检验 6 个结果的符号就能知道此顶点是否在 VF 内部,即是否可见。

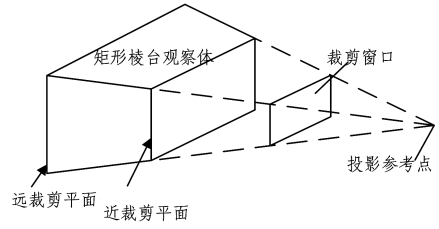


图 13 视域平头四棱锥

Fig. 13 View frustum

一些传统方法在处理大规模数据时,绘制的速率比较低。文献[36]提出了一种投影三角形(见图 14)扫描算法,它在不考虑上下裁剪面和近裁剪面的前提下,将视域平头四棱锥投影到某一平面上,然后根据投影图形对其中的节点逐一进行测试和判断。在各种实时动态四叉树 LOD 模型中,此可见性裁剪方法比对所有多边形进行独自裁剪的策略的效率更高。文献[37]也采用了视见体投影三角形算法,并在此基础上根据海平面场景的特殊性只考虑了视见体的左右两个面。在特定场景下,它实现了较为理想的结果。文献[38]利用物体的空间拓扑结构对投影三角形扫描算法进行了改善,实现了对

空间场景的预裁剪处理。文献[39]运用一种包围盒来进行裁剪，这里的包围盒是使用正方形制成的，以视锥体的中心位置为中心，以数据块长、宽、高最大值的一半作为半径来建立包围盒结构。文献[48]通过构造基于包围盒的裁剪矩阵来判断节点的可见性，在一定程度上实现了快速漫游的目的。

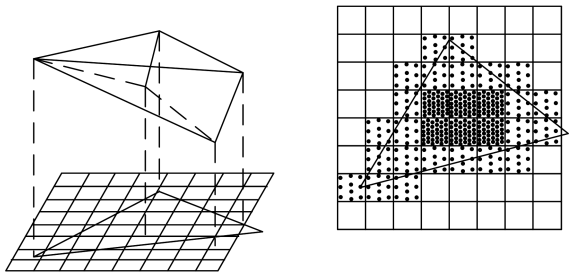


图 14　投影三角形扫描算法

Fig. 14　Projection triangle scanning algorithm

文献[40]运用层次包围盒的方法来实现可见性裁剪。前期为四叉树多级模型中的所有节点搭建包围盒，后期自上向下地对它们进行基于包围盒的裁剪判断。

文献[39]将视点位置 P 作为原点，将视点到远裁剪面的距离 R 作为半径，首先确定与此圆相交的数据块，对圆内的数据块利用视景体进行裁剪，对视景体内的数据块进行绘制，否则暂时将其剔除。裁剪采用正方形包围盒的方式进行。文献[25]在基于视锥体裁剪方法的基础上采用非严格包围盒的办法进行可见性剔除，只要判断出某子块的 4 个边角点是否在投影体内部即可，只有 4 个点都不在投影体内部时才剔除掉此地形块，否则使之可见。文献[41]采用一种层次视锥体裁剪法，它实时更新每一次的视点变化并进行相应节点的剔除操作。文献[42]首次采用一种结合图像缓存的裁剪方法对大规模的地形模型进行裁剪，此算法大大提高了地形漫游的速率，解决了地形起伏度估算偏差导致的图像缓存判断错误的问题。文献[23]采用一种新的雷达视锥体法(见图 15)，用圆台代替传统的四棱锥。经验证，这种做法极大地减少了数学计算量。文献[49]对规则格网地形数据进行分块并编号，在预处理阶段建立包围球结构并记录相关信息，大大减少了视锥裁剪的计算冗余和时间。

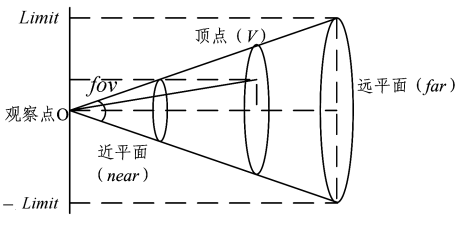


图 15　利用雷达表示法表示视锥体的原理

Fig. 15　Principle of using radar representation method to express cones

总的来说，这些构造视锥体的方法多数是采用不同形式的包围盒来剔除局部数据，能大量减少需要参与绘制的地形数据。可见性裁剪可以使得整个 LOD 算法的几何计算量呈现骤减的效果，归根结底它是根据具体的四叉树构建模式与模型简化形式来设计裁剪的过程。

5　裂缝的修复

经过评价系统的处理，数据块的不同区域可能会具有不同的分辨率，从而导致相邻四叉树节点的层级不同，因此在它们的公共边界上就会产生“裂缝”(如图 16 灰色部分所示)，在地形数据成像后就会出现缝隙，给人以地面有间断的视觉感。

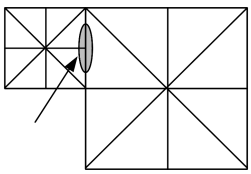


图 16　裂缝的形成

Fig. 16　Formation of cracks

目前，对于格网间裂缝的修复或消除还没有非常理想的方法，但国内外学者进行了很多尝试。这些方法可被归纳为：减边或加边法、垂直边缘(vertical skirt)法、构造限制性四叉树方法、调节顶点高度法、“纺织”法以及其他方法，如图 17 所示。

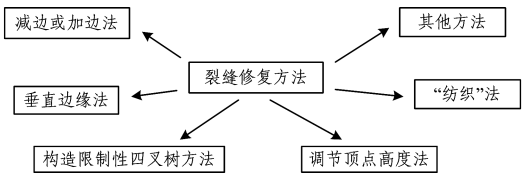
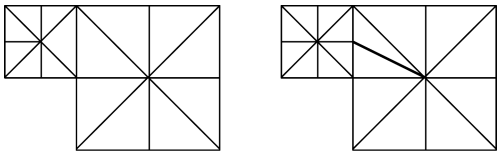


图 17　几种常用的裂缝修复方法

Fig. 17　Several common methods for repairing cracks

5.1　减边与加边法

如图 18 所示，减边与加边法是通过改变地形节点的多边形的连接方式来实现的，在两个相邻地形节点的拼接处删掉较高分辨率数据块上造成裂缝的一条边(见图 18(a))，或者从较低分辨率数据块中增加一条边(见图 18(b))。



(a)减边　　(b)加边

图 18　减边与加边法

Fig. 18　Minus edge and added edge methods

相比较而言，加边法能更全面地解决问题，即使两个相邻地形节点之间的层级相差任意大，它也能很好地修复裂缝。减边的策略比较简单，容易实现，当两个相邻地形节点之间的层级差距不超过 1 时，它可以得到较理想的结果；但它的适用性差，即不能很好地处理层级差大于 1 的情况。文献[39]采用减边法得到了较为理想的结果。通过对比分析发现，边删除算法会降低地形的表达精度，当相邻地形块的精度相差较大时，这种情况更为严重。文献[50]也是采用减边的方法，通过设定树的深度为均匀变化来使相邻节点的细分深度不超过 1。由于限制邻接地形块精度差最大为 1 的边插入算法增加了地形表达的三角形数量，因此文献[11]采用不限制邻接地形块精度差的递归边插入算法来解决裂缝问题。文献[37]结

合了减边和加边策略，实现了实时动态的连续多层次细节场景。

很多文献采用减边和加边法或是结合其他方法来消除裂缝，均达到了较好的效果。文献[17]使用线性函数曲线拟合以及幂函数曲线拟合的方式来对边界裂缝进行提取，并在裂缝采集的过程中结合减边与加边方法来消除裂缝。如果相邻节点的层级差大于1，便采用加边的策略；若相差1，则分两种状况分别做适当处理。Rottger[51]通过建立完备的布尔矩阵来存储所有四叉树节点的状态，再结合减边或加边技术构造出无缝地形的多分辨率模型，此方法是比较灵活的解决策略。GeoMipmapping算法主要是根据相邻地形块的层级差距情况适当使用减边和加边相配合的方法，但是此方法对于相邻子块层级差大于1的情况也不适用。针对GeoMipmapping算法中存在的缺点，文献[52]做了进一步的改进。对于相邻两个地形片的细节层级差大于1的裂缝难题，文献[52]提出了一种强制分割策略，即对局部地形片进行强制细化，从而确保相邻地形块间的层级差小于1。此算法简单且容易实现，能够实现实时的高效渲染，很好地解决了地形裂缝问题。

5.2 垂直边缘法

垂直边缘法[53]的核心思想是在各地形块边界处产生垂直边缘来填充裂缝。如图19所示，三角形 ABC 表示裂缝，多边形 $ABCD$ 表示边缘或裙边，边缘顶部为该地形块的边界所连的折线段，底部是地形块细化程度最大时在该边界处的最小值，从而构造出类似于裙子的把缝隙遮住的效果，给人较好的视觉感受。这种方法在相邻数据子块的边界处可能造成重叠绘制，与实际地形高度斜变的特征略有不同，在局部细节处会造成地形过渡不自然，但由于数据量很小，因此整体效率较高。文献[2,4,7,13,24,27,36,54-55]均采用了此方法，从而有效地避免了缝隙的出现。文献[56]提出一种新的裂缝消除方法，该方法利用较高细节格网总是距离观察者较近的特点，通过在相邻层次的细节层次之间产生补丁图形来避免不同层次间的格网裂缝。此方法具有预处理过程简单、计算量小、不会增加CPU的负担等优点；但其缺点为垂直向下，与实际地形高度斜变的特征略有不同，破坏了格网的规则特征，容易造成接缝处陡峭。

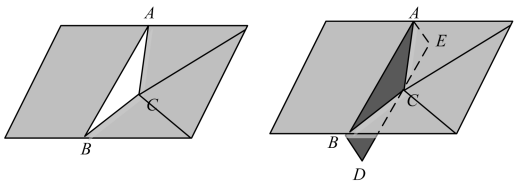


图19 垂直边缘法填充不同层次相邻节点之间的缝隙的原理

Fig. 19 Principle of filling gap between adjacent nodes at different levels by vertical edge method

文献[22]提出了另一种有效的方法，如图20所示，如同在格网的裂缝处贴上了合适的“补丁”：三角边片CDE。该算法只需要遍历划分完成后的四叉树节点，记录所有产生裂缝的顶点和裂缝的数量，生成一个裂缝列表，以便一次性消除所有裂缝，避免每次绘制都要进行裂缝节点的判断。

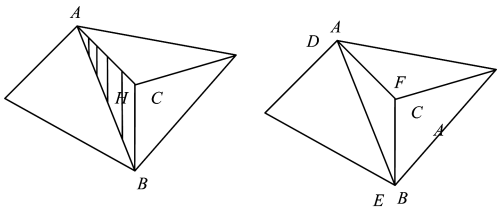


图20 利用“补丁”法修复裂缝的原理

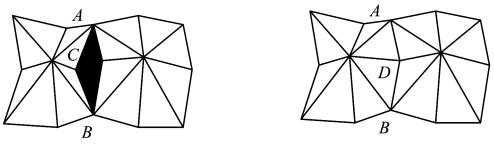
Fig. 20 Principle of using “patch” method to repair crack

5.3 构造限制性四叉树方法

构造限制性四叉树方法[57]就是在地形渲染的过程中，通过设定一系列的约束条件，遵循相邻地形的层次差异不大于1的原则来消除缝隙点。当两个相邻的地形块之间的层次相差大于1时，提高分辨率较低的地形块的细节层次。当然，这种策略也有缺陷，即需要占据较大的内存空间以及时间，灵活性也较差。文献[23]首先运用构造限制性四叉树的方法实现了相邻节点的层级差不超过1，然后在绘制较高分辨率节点时丢掉了产生缝隙的点。文献[51]提出了一种实用的裂缝修复方法，该方法使用的误差评估方法综合考虑了视距和顶点的高度差，从而确保了相邻地形块间的层级差不超过1。文献[58]在保证渲染过程中相邻子节点的层级差不大于1的基础上，利用分裂标记表从整体上建立块内或块间各节点的约束关系，然后针对相邻层级差为1的数据块使用了加边法，该策略同时处理了块内缝隙与块间裂缝。文献[59]在建立受约束四叉树结构的前提下采用双向裂缝修复方法，同时从减边和加边这两个相反的方向对产生裂缝的各个地形块进行处理，简化效果显著，并节约了大量时间。文献[60]也构造了限制性四叉树的模型结构，并加入了预编译算法的处理策略，在剖分四叉树时递归次数大大减少，从而节省了大量的计算时间并有效地避免了裂缝的产生。文献[61]按照格网节点之间的依赖关系来建立限制性四叉树模型结构，实验证明该方法可行，但构造较为复杂。文献[14]采用递归调用的方法，事先对四叉树模型规定了一些约束条件，实现了基于四叉树划分的自适应剖分，对裂缝进行了有效的处理，剖分结果符合四叉树的约束条件，而且还保证了地形表面的连续性。

5.4 调节顶点高度法

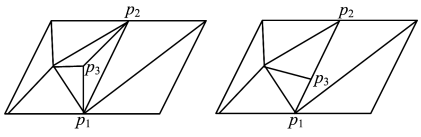
调节顶点高度法指通过改变高分辨率与低分辨率连接处的格网结构来去除缝隙，改变裂缝节点的连接方式，酌情将其去掉，用最相近的顶点代替。如图21所示，把裂缝处的顶点 C 去掉，而用相邻数据块边界处的顶点 D 代替。该算法操作简单、容易实现，而且不会造成额外的多边形面片。许多文献[25,36,62]通过强拉顶点来处理缝隙，并最终取得了较理想的图像展示效果。文献[31]提出了一种适度缝合的方法，其原理也是对节点的高度值进行有效调节，有效地保证了地形漫游的连续性和完整性。文献[63]利用高程平均法对顶点进行线性插值，从而有效消除了裂缝。文献[34]用正切函数控制顶点偏移的方法，实现了多跨度LOD裂缝的消除。文献[30]直接将父层级顶点移动到子层级数据块的边界点上，其操作如图22所示。



(a)　　(b)

图 21　调节顶点高度法

Fig. 21　Adjusting vertex height method



(a)裂缝消除前　　(b)裂缝消除后

图 22　裂缝消除前后的对比

Fig. 22　Contrast before and after crack elimination

5.5　“纺织”法

所谓“纺织”法，就是以某种方式对相邻两数据块之间的区域进行适度缝合，以达到消除裂缝的效果。文献[64]给出一种基于过渡带的空洞修补策略，对过渡带中的某些顶点进行有效填充，修补不同精度地形块之间的空洞，这种结构模型容易构造，硬件加载速度也非常快。文献[65]通过在两个 LOD 数据块之间构造数据交错区(如图 23 中的阴影部分)来实现边界的无缝连接，其边界处的数据以最细层次绘制，通过计算对应的临界粗糙度绘制交错区。

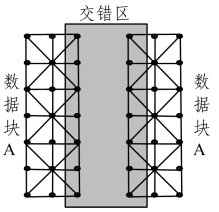


图 23　“纺织”法

Fig. 23　“Textile” method

5.6　其他方法

文献[66]也给出了一种基于四叉剖分的桶序法，其虽然有一定效果，但破坏了缝隙周围的地形模型的拓扑结构。文献[67]针对传统地形漫游方法的缺陷，给出了一种新的裂缝处理策略。此方法基于线性四叉树，其核心思想是在传统地形漫游方法的基础上，综合考虑线性四叉树在邻域搜索方面的独特性质来处理地形表面产生的缝隙。文献[67]的方法在一定程度上避免了裂缝的产生，且对已有裂缝的处理准确、快速，操作简单，通用性强。文献[20]在处理裂缝时结合了格网索引模板技术，经验证，此方法大大减少了处理器的工作量，而且使得 LOD 模型的建立更加灵活。文献[68]给出了一种避免块间裂缝的算法，该算法通过引入顶点次序的有向无环图(Directed Acyclic Graph，DAG)的概念来确定顶点间的相互依赖关系，同时综合考虑了相邻数据块间公共界限活跃点的增多对其他活跃点所带来的变化，实验证明，该方法在绘制图形时具有良好的实时性。文献[69]则是引入 Y 型基元的概念对地形模型进行了无缝处理，经验证，此方法不仅快速地修复了裂缝，而且在一定程度上保证了地形表面的连续。

5.7　小结

裂缝的出现会直接影响所绘制图像给人的视觉感受，但规则格网模型本身所具有的特点使得在渲染过程中相邻数据块之间不可避免地出现不同程度的裂缝。相对来说，减边和加边操作比较简单，但对裂缝的处理较粗糙；构造限制性四叉树的方法也比较简单，且不会制造额外的多边形面片，但是一些约束条件的使用会增加部分计算；垂直边缘法可以有效地避免缝隙的出现，但是在相邻节点块的边界处可能重复绘制，在局部细节处造成不自然的视觉感受，在数据量较小时，总体效率较高；调节顶点高度法的具体做法有强拉顶点的方式、用正切函数控制顶点偏移的方法及其他适度缝合的方法，容易实现，且能有效实现无缝拼接，但会丢失少量数据；“纺织”法能在一定程度上消除裂缝，但也在过渡带对数据进行了重复处理。综上所述，虽然国内外学者们对地形块裂缝的消除做了大量的研究，但是分块 LOD 模型的完美无缝连接还需要新的突破，必须综合四叉树模型的构造方法及其简化算法等具体情况来选择恰当的方法，以减小裂缝带来的视觉影响。

6　平滑处理

在基于四叉树的实时动态 LOD 系统中，随着视点的移动，视距也是时刻更新的，因此各数据块的层级及位置也随之变化，导致多边形面片的顶点数量发生变化。在这期间，图像会发生视点跳跃的现象，给人以地形不真实、有跳跃的视觉感受。如图 24 所示，当对地形块进行细节层次切换时，新增的顶点 p 会产生一个高程差，从而造成视点跳跃的现象。要解决视点跳跃，则需要使顶点 p' 平滑过渡到新增顶点 p 的位置。

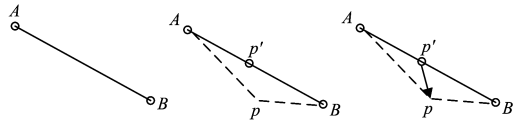


图 24　视点跳跃

Fig. 24　View point jump

目前，LOD 算法模型中平滑过渡处理的相关文献较少，绝大多数文献均运用几何过渡算法来降低不同层级地形数据转化图像跳跃所带来的影响。

几何过渡算法的主要思想是：根据细节层次的选择方式，通过对顶点的位置进行插值运算，来实现不同细节层次的地形块的平滑切换，过渡算法能彻底解决在渲染过程中出现的视点跳跃现象。该方法的缺点是，在进行层次细节切换的过程中，处理器每次都要对增加或者删除的顶点进行插值运算，这会显著影响地形绘制的效率。因此可以通过 GPU 来进行优化运算，完成平滑过渡的处理过程。

在几何过渡算法中，国内外学者主要通过构造各种插值变换来达到预期的目的。文献[5]提出的插值变换的方法有效地消除了突变带来的视觉影响；文献[20]借鉴文献[5]中插值变换的方法，成功地解决了形突变的问题；文献[19]引入了一种“形变”算法来适当地调整突起处节点的高度值，通过构

造“形变”公式来达到消除突变的效果;这些文献[11,70]均运用了顶点渐变技术来对地形进行平滑处理,对节点细分前后的位置数值做线性插值处理,在一定程度上避免了不同层级数据块转化时所引起的画面不平滑现象。

现有的平滑处理模型优化方法绝大多数是对某些节点进行不同形式的插值,这种做法较为成功地达到了图像自然过渡的效果。采用哪些点以及何种形式等问题需要综合各个数据的调度过程,尽可能地采用一些常驻内存的点与满足计算量限制的形式。

7 基于 GPU 处理的 LOD 模型技术

进入 21 世纪以后,GPU(Graphic Processing Unit)处理数据的能力得到了迅猛的发展,达到了每秒至少处理一千万个多边形格网的效果。大量的研究人员开始着手于将 GPU 运用到复杂三维模型的可视化环节中,借助其快速处理图像的特长来分担 CPU 的部分任务,使得 CPU 与 GPU 实现高效配合,从而达到三维地形的实时快速漫游效果。文献[71]中,CPU 使用相机的标准距离建造了一个粗粒度的四叉树补丁,然后运用 GPU 生成非均匀地形补丁单位,实验证明了该算法的可行性。文献[55]采用了一种 Chunked LOD 算法,通过 GPU 存储地形格网并合理地进行场景组织调度,具有负载低、扩展性高、图像渲染效果好等优点,得到了很好的应用与推广。文献[72]中的 GPU 使用基于屏幕空间投影误差的分层级规则实现了对地形格网的细分。文献[73]运用 GPU Tesellation 技术,通过对粗糙的地形格网进行插值产生新的顶点,极大地提高了图像显示的精度。文献[57]提出了面向 GPU 绘制的三维模型烘焙技术和多尺度可视化方法,达到了较高的绘制效率。文献[74]提出一种基于 GPU-CPU 的解压缩算法,对大型 3D 分区格网 GPU 加速压缩,实践证实此方法比使用单一 CPU 方法的运行速度更快。文献[75]运用一种基于粒子的颗粒状地形仿真方法,其中加入了 GPU 的图像硬件渲染技术,实验结果不太理想,但它是一种较新的思路,有进一步研究的必要。近些年,随着 GPU 的不断发展,主流的图形 API 都相继支持 GPU Tessellation 技术。基于 GPU 处理的 LOD 技术还被运用在一些移动和桌面设备上[77-78],且达到了不错的视觉感受,从而为研究人员在虚拟现实领域的研究指明了新的方向。

虽然硬件图形处理技术的地形模型绘制算法越来越多地渗入到三维模型可视化过程中,但是其发展还存在一些瓶颈,如较理想的数据处理效果对计算机硬件的要求较高,数据在 GPU 和 CPU 之间的频繁交互需要耗费一定的时间等。

结束语 LOD 技术的各个环节之间联系密切,相辅相成,必须考虑各部分之间的相互影响。本文主要从构造四叉树结构方法的改进、节点评价系统的改进、视域剔除方法的改进及裂缝修复方法这 4 个方面对基于四叉树分割的 LOD 算法的相关研究进行了系统梳理,总结和分析了各类算法的优缺点;另外还简要叙述了关于平滑处理和基于 GPU 处理的 LOD 模型技术的研究内容及现状,较为全面地实现了对基于四叉树分割的 LOD 算法的总结。

从现有文献看,基于四叉树剖分的 LOD 技术仍有一些方面不够成熟,需要深入研究,比如裂缝的修复问题,现存方法大多都不能很好地兼顾模型优化效果和处理器的计算速率,这些都还有待进一步研究;另外,基于 GPU 的三维模型可视化技术也有很广阔的发展空间。

参 考 文 献

[1] LI X Y. A hybrid algorithm for terrain simplification[D]. Vancouver: The University of British Columbia, 2003: 25-38.

[2] LINDSTROM P, KOLLER D, RIBARSKY W, et al. Real-Time, Continuous Level of Detail Rendering of Height Fields[C]// Proceedings of the ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques. ACM, 1996: 109-117.

[3] DUCHAINEAUY M, WOLINSKY M. ROAMing Terrain: Real-time Optimally Adapting Meshes[C]// Proceedings IEEE Visualization. Phoenix, AZ, USA, 1997: 81-88.

[4] MA Z T, PAN M, HU J X, et al. A Fast Walkthrough Method for Massive Terrain Based on Data Block Partition[J]. Journal of Peking University (Natural Science Edition), 2004, 40(4): 619-625. (in Chinese)
马照亭,潘懋,胡金星,等. 一种基于数据分块的海量地形快速漫游方法[J]. 北京大学学报(自然科学版),2004,40(4):619-625.

[5] DE BOER W H. Fast Terrain Rendering Using Geometrical Mipmapping[C]// World Wide Web Conference Series. 2000.

[6] VON HERZEN B, BARR A H. Accurate Triangulations of Deformed, Intersecting Surface[C]// Conference on Computer Graphics and Interactive Techniques. ACM, 1987: 103-110.

[7] LI Q, DAI S L, ZHAO Y J, et al. A Block LOD Real-time Rendering Algorithm for Large Scale Terrain[J]. Journal of Computer-Aided Design & Computer Graphics, 2013, 25(5): 708-713. (in Chinese)
李钦,戴树岭,赵永嘉,等. 分块 LOD 大规模地形实时渲染算法[J]. 计算机辅助设计与图形学学报,2013,25(5):708-713.

[8] ZHANG J F, YAO Z H. LOD Algorithm of Terrain Based on Conservative Screen Error and Isolated Division of Quad-tree [J]. Journal of Southwest Jiaotong University, 2013, 48(4): 666-671, 677. (in Chinese)
张俊峰,姚志宏. 基于四叉树孤立分割和屏幕误差的地形 LOD 算法[J]. 西南交通大学学报,2013,48(4):666-671,677.

[9] RUI X P. A LOD Algorithm Based on Unfullquadtree [J]. Journal of Image and Graphics, 2005, 10(9): 3659-3662. (in Chinese)
芮小平. 一种基于不完全四叉树的 LOD 生成算法[J]. 中国图象图形学报,2005,10(9):3659-3662.

[10] CHEN H M, ZHAN S Y, LIU T. Real-time Simulation of Global Terrain Facing CE-1 Missions[J]. Journal of System Simulation, 2009, 21(19): 6135-6139. (in Chinese)
陈宏敏,战守义,刘涛. “嫦娥一号”任务全球地形实时仿真技术研究[J]. 系统仿真学报,2009,21(19):6135-6139.

[11] WANG J Y, HE Y Z, XIE J B, et al. Rapid Protracting Arithmetic about Large Scale DEM Data Based on Embeded Envirment [J]. Acta Aeronauticaet Astronautica Sinica, 2009, 30(1): 132-135. (in Chinese)
王金岩,何亦征,谢剑斌,等. 一种基于嵌入式环境的海量 DEM 数据快速绘制算法[J]. 航空学报,2009,30(1):132-135.

[12] ZHAI R. GPU-based real-time terrain rendering Design and im-

plementation[J]. Neurocomputing，2016，171(1)：1-8.

[13] ZHAO Y X. A modified LOD terrain model based on quadtree algorithm[C]// Proceedings of the 2009 International Joint Conference on Computational Sciences and Optimization (CSO 2009). 2009：259-263.

[14] TAN B，XU Q，ZHOU Y. The Research of Large Scale Terrain Visualization[J]. Journal of Image and Graphics，2003，8(5)：578-584. (in Chinese)
谭兵，徐青，周杨. 大区域地形可视化技术的研究[J]. 中国图象图形学报，2003，8(5)：578-584.

[15] LINDSTROM，PETER. Real-Time Generation of Continuous Levels of Detail for Height Fields[C]// Proceedings of the ACM SIGGRAPH Conference on Computer Graphics. 1996：109-117.

[16] SONG L B，GONG H J，WANG X H. Global Terrain Real-Time Rendering Based on Improved Constraint Quad-Tree and Level of Detail Algorithm[J]. Computer and Digital Engineering，2013，41(10)：1668-1671. (in Chinese)
宋力兵，龚华军，王新华. 基于改进的约束四叉树 LOD 全球地形实时绘制[J]. 计算机与数字工程，2013，41(10)：1668-1671.

[17] LAN L，XU L. Elimination of Boundary Cracks in 3D Environment Art Design[J]. Bulletin of Science and Technology，2015，31(11)：203-206. (in Chinese)
兰玲，徐莉. 3D 环境艺术设计地形绘制中的边界裂缝消除[J]. 科技通报，2015，31(11)：203-206.

[18] ZOU C M，LI Y，LU Y，et al. Dynamic LOD virtual terrain optimization based on quadtree[J]. Journal of Wuhan University of Technology，2009，31(18)：182-185. (in Chinese)
邹承明，李引，陆苑，等. 基于四叉树的动态 LOD 虚拟地形优化[J]. 武汉理工大学学报，2009，31(18)：182-185.

[19] LEI J H，ZENG F X，WU M X. The study of the LOD algorithm based on quad-tree structure[J]. Manufacturing Automation，2010，32(8)：211-214，228. (in Chinese)
雷军环，曾凡喜，吴名星. 基于四叉树的视点相关 LOD 地形仿真算法研究[J]. 制造业自动化，2010，32(8)：211-214，228.

[20] LI C H，WANG C B. Large-Scale LOD Terrain Rendering Based on Hierarchical Classification[J]. Journal of Donghua University (Natural Science Edition)，2010，36(4)：351-355. (in Chinese)
李晨辉，王长波. 基于层次划分的大规模 LOD 地形绘制[J]. 东华大学学报(自然科学版)，2010，36(4)：351-355.

[21] LIN Q P，ZHOU D Y，ZHANG K. Rapid Generation of 3D Terrain Scene in Battlefield Environment Simulation[J]. Electronics Optics & Control，2009，16(12)：38-42. (in Chinese)
林清漂，周德云，张堃. 战场环境三维地形快速生成算法研究[J]. 电光与控制，2009，16(12)：38-42.

[22] WU J，YANG Y，GONG S，et al. A New Quadtree-based Terrain LOD Algorithm [J]. Journal of Software，2010，5(7)：769-776.

[23] DENG Z H，ZHANG T T，LIU R H，et al. A Fast and Effective Method for Large-Scale Terrain Rendering [J]. Journal of Northwestern Polytechnical University，2010，28(6)：936-940. (in Chinese)
邓正宏，张婷婷，刘瑞恒，等. 大规模地形渲染技术的研究与实现[J]. 西北工业大学学报，2010，28(6)：936-940.

[24] HU A H，HE Z Y，MA X P. Rendering Technique for Large Scale Terrain Based on LOD[J]. Acta Geodeaticaet Cartographica Sinica，2009，38(12)：28-31. (in Chinese)
胡爱华，何宗宜，马晓萍. 基于 LOD 的大规模地形实时绘制方法[J]. 测绘通报，2009，38(12)：28-31.

[25] ZHU J，GONG J H，QI H，et al. A Simple and Efficient Algorithm for Real-Time Rendering of Large Scale Terrain[J]. Geography and Geo-information Science，2005，21(2)：24-27. (in Chinese)
朱军，龚建华，齐华，等. 大规模地形实时绘制算法[J]. 地理与地理信息科学，2005，21(2)：24-27.

[26] LUO J X，TANG J. LOD Algorithm Based on Improved Quadtree-split and Node Storage[J]. Computer Engineering，2009，35(20)：202-204. (in Chinese)
罗景馨，唐琎. 基于改进四叉树分割和结点存储的 LOD 算法[J]. 计算机工程，2009，35(20)：202-204.

[27] ZHAO Y X，MA Y. A modified LOD terrain model based on quadtree algorithm[C]// Proceedings of the 2009 International Joint Conference on Computational Sciences and Optimization (CSO 2009). 2009：259-263.

[28] LINDSTROM P，PASCUCCI V. Visualization of Large Terrains Made Easy[C]// Conference on Visualization. IEEE Computer Society，2001：24-26.

[29] JIN H L，GAO J X. An Algorithm for Real-Time Visualization of Large-Scale Terrain[J]. Journal of Zhengzhou Institute of Surveying and Mapping，2006，23(1)：65-68. (in Chinese)
靳海亮，高井祥. 大规模地形实时可视化算法[J]. 测绘科学技术学报，2006，23(1)：65-68.

[30] YI C L，ZHAN Q M，XU W Q，et al. Simplification Technology for Real-time Rendering of Large-scale 3D Terrain[J]. Geomatics and Information Science of Wuhan University，2012，37(5)：555-559. (in Chinese)
尹长林，詹庆明，许文强，等. 大规模三维地形实时绘制的简化技术研究[J]. 武汉大学学报(信息科学版)，2012，37(5)：555-559.

[31] WEI Y M，GU T Y，YANG B，et al. Rapid Continuous Rendering Technology of Terrain Based on Wavelet[J]. Computer Engineering，2007，33(19)：218-220. (in Chinese)
魏迎梅，谷天阳，杨冰，等. 基于小波的地形连续快速绘制技术[J]. 计算机工程，2007，33(19)：218-220.

[32] WANG P，ZHAI Z J，JIANG H M. Research on Real-Time Rendering for Massive Multi-Resolution Terrain[J]. Measurement & Control Technology，2008，27(2)：32-35. (in Chinese)
王培，翟正军，姜红梅. 大规模多分辨率地形实时绘制算法研究[J]. 测控技术，2008，27(2)：32-35.

[33] YIN G S，WANG H L，ZHANG J，et al. A Fast Real-Time Rendering Method of 3D Terrain[C]// The IEEE International Conference on Computer Science and Software Engineering. 2009：1-4.

[34] NIE J L，LIU S，GUO D L，et al. GPU-Accelerated Fast Rendering Method for Multi-Resolution Geometry Image[J]. Journal of Computer-Aided Design & Computer Graphics，2013，25(1)：101-109. (in Chinese)
聂俊岚，刘硕，郭栋梁，等. GPU 加速的多分辨率几何图像快速绘制方法[J]. 计算机辅助设计与图形学学报，2013，25(1)：101-109.

[35] LI S，LIU X H，WANG W C，et al. Combination of Hierarchical

Visibility with Level of Detail Terrain Model[J]. Chinese Journal of Computers,2002,25(9):945-952. (in Chinese)
李胜,刘学慧,王文成,等.层次可见性与层次细节地表模型相结合的快速绘制[J].计算机学报,2002,25(9):945-952.

[36] ZHAO Y B,SHI J Y,et al. A Fast Algorithm for Large Scale Terrain Walkthrough[J]. Journal of Computer-aided Design & Computer Graphics,2002,14(7):624-628. (in Chinese)
赵友兵,石教英,等.一种大规模地形的快速漫游算法[J].计算机辅助设计与图形学学报,2002,14(7):624-628.

[37] WU J,XU X G,WANG J G,et al. Real-time Walkthrough of Large Scale Ocean[J]. Journal of System Simulation, 2006, 18(9):2505-2507. (in Chinese)
吴晶,徐晓刚,王建国,等.大规模海面的快速漫游[J].系统仿真学报,2006,18(9):2505-2507.

[38] WANG J,TANG X A,JING N. A Real-Time Walkthrough Algorithm of Large-Scale and Complex Scene[J]. Computer Engineering and Applications,2003,39(16):88-90. (in Chinese)
王钧,汤晓安,景宁.一种大范围复杂场景的快速绘制算法[J].计算机工程与应用,2003,39(16):88-90.

[39] CHEN G,XIONG X H. A Study of Dynamic LOD Algorithm for Huge Quantities of Topographic Roaming[J]. Bulletin of Surveying and Mapping,2007,36(4):46-48. (in Chinese)
陈刚,熊兴华.海量地形漫游中动态 LOD 算法研究[J].测绘通报,2007,36(4):46-48.

[40] GAO Y,DENG B S,CHEN P,et al. Fast Rendering Algorithm for Large-scale and Complex Scene Based on Continuous HLOD [J]. Journal of Chinese Computer Systems,2007,28(7):1267-1271. (in Chinese)
高宇,邓宝松,陈鹏,等.基于连续 HLOD 的大规模场景快速绘制算法[J].小型微型计算机系统,2007,28(7):1267-1271.

[41] ZHI M,ZHENG Y T. HLODs for fast rendering of large-scale environments[J]. Journal of Computer Applications, 2007, 27(9):2126-2128. (in Chinese)
智明,郑姨婷.HLODs 模型在场景快速绘制中的应用[J].计算机应用,2007,27(9):2126-2128.

[42] FENG C S,WANG L L,YANG Y D. Fast Rendering Method Based on View Frustum Culling and Image Caching[J]. Journal of System Simulation,2006,18(z1):94-98. (in Chinese)
封春升,王莉莉,杨跃东.基于视域剔除和图像缓存技术的复杂场景快速绘制方法[J].系统仿真学报,2006,18(z1):94-98.

[43] HOU S Y,ZHAO X S,GUAN Y Q. Seamless expression of block LOD model based on multi-resolution half-edge [J]. Geography and Geo-Information Science,2015,31(4):18-21. (in Chinese)
侯绍洋,赵学胜,官亚勤.基于多分辨率半边的分块 LOD 模型无缝表达[J].地理与地理信息科学,2015,31(4):18-21.

[44] YI J,ZHAO L. Research on Terrain Visualization Based on LOD Dynamic[C]// Seventh International Conference on Measuring Technology and Mechatronics Automation. 2015:901-904.

[45] KALEM S,KOURGLI A. Large-scale Terrain Level of Detail Estimation based on WaveletTransform[C]// Proceedings of the 10th International Conference on Computer Graphics Theory and Applications(GRAPP-2015). 2015:258-264.

[46] ZHANG Z F,ZHANG N. A LOD algorithm based on out-of-core for large scale terrain rendering[C]// 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer(MEC). 2013:2168-2171.

[47] SONG X S,QUAN J C,ZHAO X Y. Real-time detail enhancement terrain algorithm based on Chunked LOD [J]. Computer Engineering and Design,2014,35(2):578-582. (in Chinese)
宋省身,全吉成,赵秀影.基于 Chunked LOD 的实时细节增强地形算法[J].计算机工程与设计,2014,35(2):578-582.

[48] LI Y,SUN S,XU H. A LOD algorithm for terrain-load in the scene of high-speed railway virtual reality based on quad-tree [C]// Proceedings 5th International Conference on Computational Intelligence and Communication Networks(CICN 2013). 2013: 500-502.

[49] WANG Q,GAO B L,GAO R J,et al. LOD algorithm based on quadratic tree surrounded by ball and screen error [J]. Microelectronics and Computer,2016,33(5):127-132. (in Chinese)
王倩,高保禄,高锐军,等.基于四叉树包围球和屏幕误差的 LOD 算法[J].微电子学与计算机,2016,33(5):127-132.

[50] ZENG W,HAN Z X,ZHU X F. Research of Application of LOD Algorithm to 3D Terrain Simulation[J]. Journal of System Simulation,2009,21(1):292-294. (in Chinese)
曾维,韩占校,朱学芳. LOD 算法在 3D 地表模拟中的应用研究[J].系统仿真学报,2009,21(1):292-294.

[51] HOPPE H. Smooth View-dependent level-of-detail control and its application to terrain rendering[C]// Proceedings of the IEEE Visualization Conference. 1998:35-42.

[52] REN H P,JIN B. Real-time Rendering of LOD Terrain Based on Quadtree[J]. Journal of Huazhong University of Science and Technology(Natural Science Edition), 2011, 39(2): 6-10. (in Chinese)
任宏萍,靳彪.基于 4 叉树的 LOD 地形实时渲染技术[J].华中科技大学学报(自然科学版),2011,39(2):6-10.

[53] THATCHER U. Chunked LOD:Rendering massive terrains using chunked level of detail control[EB/OL]. http:// www.vterrain.org/LOD/Papers.

[54] YU W L,ZHANG L M. Large-scale LOD Adaptive Terrain Rendering [C]// 2nd International Conference on Advanced Design and Manufacturing Engineering. 2012:2450-2453.

[55] ULRICH T. Chunked LOD: Rendering massive terrains using chunked level of detail control[EB/OL]. (2003-12-08). http:// www.vterrain.org.

[56] WAN M,LIANG X,ZHANG F M. Improved Crack Removing Algorithm for Quad-Tree Terrain Rendering[J]. Journal of System Simulation,2015,27(7):1520-1525. (in Chinese)
万明,梁霞,张凤鸣.一种四叉树地形渲染裂缝的改进消除算法[J].系统仿真学报,2015,27(7):1520-1525.

[57] CHEN J,WU S,XIE B X. Visual modeling of complex 3D model for GPU rendering [J]. Journal of Wuhan University(Information Science Edition),2014,39(1):106-111. (in Chinese)
陈静,吴思,谢秉雄.面向 GPU 绘制的复杂三维模型可视化方法[J].武汉大学学报(信息科学版),2014,39(1):106-111.

[58] MA C Y，CHEN G，CHEN Y，et al. An Algorithm of Avoiding Crack for Rendering Terrain Based on Pyramid Model[J]. Control & Automation，2009，25(15)：157-159. (in Chinese)
马纯永，陈戈，陈勇，等. 基于金字塔模型的地形格网裂缝消除算法[J]. 微计算机信息，2009，25(15)：157-159.

[59] WANG Z，HU M，LI X. Quick simplification for LOD terrain with dynamic multi-resolution based on quad tree[J]. Journal of Computer Applications，2007，27(7)：1641-1643. (in Chinese)
王臻，胡敏，李响. 基于四叉树的动态多分辨率 LOD 地形快速简化[J]. 计算机应用，2007，27(7)：1641-1643.

[60] WANG D C，WAN W G，TANG J Z，et al. Preprocessing LOD algorithm for large scale terrain based on restricted quadtrees [J]. Computer Engineering and Applications，2007，43(24)：107-109. (in Chinese)
王道臣，万旺根，唐经洲，等. 基于限制性四叉树 LOD 大规模地形预处理算法[J]. 计算机工程与应用，2007，43(24)：107-109.

[61] CHEN G. Research on Hierarchical Description and Real-time Rendering of Virtual Terrain Environment[D]. Zhengzhou：PLA Information Engineering University，2000. (in Chinese)
陈刚. 虚拟地形环境的层次描述与实时渲染技术的研究[D]. 郑州：解放军信息工程大学，2000.

[62] MC ROBERTS D A K，HARDY A. Level of Detail for Terrain Geometry Images[C]// Conference on Computer Graphics，Virtual Reality，Visualisation and Interaction in Africa. 2007：25-30.

[63] ZHANG J F，SUN D P，XU D H. High-scale Terrain Adaptive LOD Algorithm Supported by 3D_DP and Quad_TIN [J]. Geography and Geo-Information Science，2014，30(3)：29-32. (in Chinese)
张俊峰，孙大鹏，许德合. 3D_DP 和 Quad_TIN 支持下的大规模地形自适应 LOD 算法[J]. 地理与地理信息科学，2014，30(3)：29-32.

[64] YU Z，LIANG X H，MA S，et al. A Real-Time Rendering Method for Large Terrain Including Details in Different Resolutions[J]. Journal of Computer Research and Development，2010，47(6)：988-995. (in Chinese)
于卓，梁晓辉，马上，等. 一支持大规模多种精度地形的实时绘制算法[J]. 计算机研究与发展，2010，47(6)：988-995.

[65] LENG Z G，TANG X A，HAO J X，et al. Study on Dynamic Rapid Rendering Technique of Large Scale Terrain[J]. Journal of System Simulation，2006，18(10)：2832-2835. (in Chinese)
冷志光，汤晓安，郝建新，等. 大规模地形动态快速绘制技术研究[J]. 系统仿真学报，2006，18(10)：2832-2835.

[66] TAN B. Research on terrain three-dimensional reconstruction based on remote sensing image[D]. Zhengzhou：PLA Information Engineering University，2001. (in Chinese)
谭兵. 基于遥感影象的地形三维重建技术的研究[D]. 郑州：解放军信息工程大学，2001.

[67] ZHANG J B，LIU X G. The Application of Linear QuadTrees in Rendering Terrain Surface[J]. Computer Engineering and Applications，2002，38(8)：46-47. (in Chinese)
张剑波，刘修国. 线性四叉树在基于 LOD 的地表模型绘制中的应用[J]. 计算机工程与应用，2002，38(8)：46-47.

[68] YIN Y，CHEN G J，WU W. An Algorithm of Avoiding Crack for Rendering Parting Terrain[J]. Journal of Computer-aided Design & Computer Graphics，2006，18(10)：1557-1562. (in Chinese)
殷媛，陈国军，吴威. 地形分块绘制中的边界裂缝处理算法[J]. 计算机辅助设计与图形学学报，2006，18(10)：1557-1562.

[69] HAN M，TANG S T，LI Y. Real-time Visualization Algorithm of Large-scale Terrain [J]. Computer Engineering，2008，34(13)：270-272. (in Chinese)
韩敏，汤松涛，李洋. 大规模地形实时可视化算法[J]. 计算机工程，2008，34(13)：270-272.

[70] TANG X A，CHEN M，SUN M Y. An Approach of Simplification and High-speedRendering for Terrain Model[J]. Journal of Wuhan University(Information Science Edition)，2001，26(4)：310-314. (in Chinese)
汤晓安，陈敏，孙茂印. 地景模型的简化与快速绘制方法研究[J]. 武汉大学学报(信息科学版)，2001，26(4)：310-314.

[71] WANG D，ZHU Q S. Adaptive LOD terrain rendering with GPU tessellation[J]. Journal of Computational Information Systems，2015，11(20)：7489-7496.

[72] DONG L M，ZHANG B，ZHAO X S. A Seamless Segmentation Algorithm Based on GPU Tessellation [J]. Journal of Wuhan University(Information Science Edition)，2016，41(11)：1-6. (in Chinese)
董路明，张斌，赵学胜. 一种基于 GPU Tessellation 的地形无缝绘制算法[J]. 武汉大学学报(信息科学版)，2016，41(11)：1-6.

[73] NIE J L，ZHANG J W，GUO D L. Application of Smooth Transition to GPU Fabricated Terrain Rendering [J]. Small Microcomputer System，2014，35(1)：121-125. (in Chinese)
聂俊岚，张精卫，郭栋梁. 细节平滑过渡的 GPU 构网地形渲染[J]. 小型微型计算机系统，2014，35(1)：121-125.

[74] ZHAO J Y，TANG M. Connectivity-based segmentation for GPU-accelerated mesh decompression[J]. Journal of Computer Science and Technology，2012，27(6)：1110-1118.

[75] CRAIG L，PATRICK M. A GPU-based level of detail system for the real-time simulation and rendering of large-scale granular terrain[C]// 22nd International Conference in Central Europe on Computer Graphics，Visualization and Computer Vision. 2015：169-178.

[76] SUÁREZ J P，TRUJILLO A. An efficient terrain Level of Detail implementation for mobile devicesand performance study[J]. Computers，Environment and Urban Systems，2015，52：21-33.

[77] LIU X，LIU Z，MEI X D. Based on real-time LOD simplified rendering rendering method[J]. Journal of Hunan University of Science and Technology(Natural Science Edition)，2015，30(4)：92-96. (in Chinese)
刘晓，刘镇，梅向东. 基于实时 LOD 简化绘制的渲染优化方法[J]. 湖南科技大学学报(自然科学版)，2015，30(4)：92-96.