

文章编号 :1007-130X(2007)002-0066-03

ROAM 算法及其在地形可视化中的应用^{*}

The ROAM Algorithm and Its Application in Terrain Visualization

魏楠¹,江南²

WEI Nan¹,JIANG Nan²

(1. 南京大学城市与资源学系,江苏 南京 210093;2. 中科院南京地理与湖泊研究所,江苏 南京 210008)

(1. Department of City and Resources, Nanjing University, Nanjing 210093;

2. Nanjing Institute of Geography and Limnology, Nanjing 210008, China)

摘 要:三维地形可视化是 GIS、计算机仿真、虚拟现实等领域中的关键技术之一,而基于多层次细节的实时优化自适应网格动态地形渲染算法(ROAM)凭借其简单性和可扩展性成为解决海量高程数据地形可视化的常用方法。本文详细介绍了 ROAM 算法的原理及其特点,进而针对实际应用提出对算法的改进,并进行了具体实现。实验结果表明,使用 ROAM 技术能够真实地反映地形面貌。改进后的算法避免了 ROAM 对视点距离的敏感性,提高了运算速度,能够满足三维地形场景显示的应用需求。

Abstract: 3D terrain visualization is one of the key techniques in GIS, computer simulation and virtual reality. The ROAM algorithm is based on multiple levels of detail. Its simplicity and expansibility make it a common terrain visualization method. This paper presents the theory and characteristics of the ROAM algorithm. Then the authors give an improved algorithm and its implementation based on ROAM according to practical applications. The experimental results show that the ROAM algorithm can authentically realize terrain visualization, and the improved algorithm can avoid the sensitivity of ROAM on viewpoint distances. Thus the computing speed is increased and the application demands are satisfied.

关键词:ROAM;层次细节;地形可视化

Key words: ROAM; level of detail; terrain visualization

中图分类号: TP391.9

文献标识码: A

1 引言

目前,地形可视化一直是计算机图形学的研究热点,并广泛应用于地理信息系统、计算机仿真、虚拟现实等领域^[1,2]。通常的地形算法是通过高程数据构建三角网格来逼近真实的地形。随着信息采集技术的发展,利用海量数字高程数据生成具有三维真实感的地形已成为地形可视化的必然趋势。但是,目前的图形硬件设备在处理海量数据方面还存在一定的困难。为了减少地形渲染时硬件的压力,需要采用真正的 3D 渲染技术层次细节技术 LOD(Level of Detail, 简称 LOD)。ROAM 是诸多 LOD 算法中广泛选择的方法之一,它能够根据视点的位置和模型的起伏形

状而动态地计算模型的细节层次,减少每帧渲染多边形的数量,并具有较高的视觉质量^[3]。

目前,国内外对于地形可视化已经进行了大量研究,其中多以模型的简化、不同分辨率和不同 LOD 层次划分计算为根本来构造同原始地形相近似的地表模型。由于 ROAM 算法能够实时、动态、高效地对模型进行划分,在地形可视化领域受到了广泛的关注。随着实际项目中的开发应用,该算法暴露出了一些不足。不同的研究者提出的各种改进方法中主要集中在对 ROAM 算法分块效率的改进上,进而提高了该算法在更大范围的地形建模中的算法效率。本研究在分析 ROAM 算法优缺点的基础上,针对 ROAM 算法对视点距离非常敏感这一缺点提出了改进办法,并将改进算法应用到实际中,取得了良好的效果,弥补

^{*} 收稿日期:2006-03-19;修订日期:2006-05-29

基金项目:中国科学院知识创新工程领域前沿项目(CXNIGLAS-A02-012)

作者简介:魏楠(1983-),男,河南新乡人,硕士生,研究方向为虚拟现实和地理信息系统;江南,研究员,博士生导师,研究方向为地理信息系统、遥感应用、计算机应用系统设计与开发及虚拟现实工程等。

通讯地址:200031 上海市长乐路 989 号世纪商贸广场 15 楼育碧电脑软件公司;Tel:13482150661;E-mail:weinanwn@163.com

Address:Yubi Software Company,15th Fl,Century Trade Square Building,989 Changle Rd,Shanghai 200031,P.R.China

了以往因单重视算法效率的改进而忽视了改进算法显示效果的缺点。

2 ROAM 算法介绍

ROAM (Real-time Optimally Adapting Meshes, 简称 ROAM) 算法基于面层次模型^[3], 其基本思想是: 在对地形进行三维显示时, 根据视点的位置和视线的方向来计算视点距离地形表面的三角片元的距离, 再根据目标格网的空间粗糙程度来判断是否对地形表面的三角片元进行一系列基于三角型二叉分割的分解和合并, 最终生成逼近真实地形的无缝无重叠的简化连续三角化地形表面^[4~7]。

等腰直角三角形是 ROAM 算法的基本数据单元。对 ROAM 算法主要是对三角形进行分割和合并的操作。从直角顶点作垂线将三角形分为两个等腰直角三角形定义为分割, 此操作的逆过程定义为合并。通过不同的递归分割和合并操作, 算法可以将这个三角形地形表面控制到一定的细节程度上。图 1 显示了对三角形的分割和合并操作。

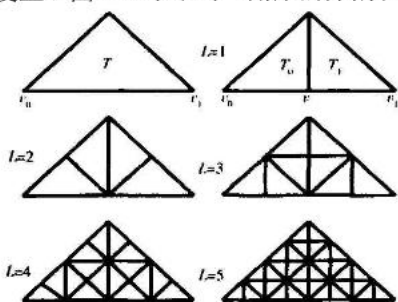


图 1 不同层次 ROAM 三角形的分割与合并

分割和合并是 ROAM 算法的基本操作。为了使模型到达更高的细节程度, 可通过对低细节程度的模型进行递归分割操作; 同时, 这一过程又是可逆的, 即通过合并, 又可以将模型从高细节程度恢复到原本低的细节程度。这样, 自底向上逐步细分互为可逆的算法, 非常容易通过递归调用实现。为了避免在分割和合并的过程中出现裂隙, ROAM 定义了三角形的拓扑关系, 图 2 显示了分割和合并的操作。

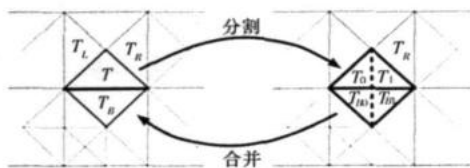


图 2 分割与合并示意图

T_L 、 T_R 分别表示三角形 T 的左直角边邻接三角形和右直角边邻接三角形, T_B 表示 T 的斜边邻接三角形。ROAM 算法定义: 如果要分割三角形 T , 那么它的 T_B 三角形也要同时被分割, 这样就能够保证在 T 和 T_B 邻接边上出现裂隙。在 T_B 被同时分割后, 如果导致 T_B 和它斜边邻接三角形不在同一个细节层次上, 就需要再递归地对这个低层次的邻接三角形进行分割。图 3 显示了如何强制分割。

在 ROAM 算法中, 根据一定的误差尺度, 通过分割和合并操作, 就可以用有限数量的三角形来逼近地形表面, 从

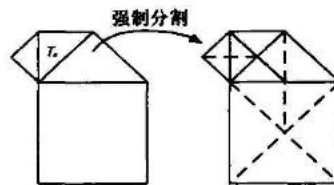


图 3 强制分割示意图

而用简化模型提高视觉效果。因此, 误差尺度的确定是决定 ROAM 算法效果的重要因素。

在地形模型中使用屏幕空间误差来度量原始地形表面和投影到屏幕上的片元之间的差异程度, 这是一种与视点相关的度量误差方法。误差产生的原因是: 当两个相邻三角形 T_0 、 T_1 合并成一个低层次的大三角形时, 两个三角形所共享的直角顶点和新的三角形斜边垂足点之间存在高度差。ROAM 算法使用包围三角形的楔形的厚度表示这种几何误差, 每个三角形所被包围的楔形的厚度都是自底向上建立的。对于三角形 T 及其父三角形 (更高细节的三角形) T_0 、 T_1 , 其楔形厚度 eT 的计算公式如下:

$$eT = \text{MAX}(eT_0, eT_1) + |Z_{v_m} - (Z_{v_0} + Z_{v_1})/2|$$

其中, v_0 、 v_1 是 T 斜边的两个端点, v 是斜边中点。

将以上楔形三角形的厚度段投影到平面上, 从而得到三角形的空间误差。计算三角形 T 的空间误差公式:

$$\delta_{\text{screen}} = \frac{d\delta \sqrt{(e_x - v_x)^2 + (e_y - v_y)^2}}{(e_x - v_x)^2 + (e_y - v_y)^2 + (e_z - v_z)^2}$$

其中, $e = \langle e_x, e_y, e_z \rangle$ 为视点的坐标; d 为视点到投影平面的距离; δ 为世界坐标下的单位长度对应屏幕的像素个数。得出三角形屏幕空间误差的上限, 据此确定出三角形分割和合并的优先层次, 可以进行空间不同层次的划分。

3 ROAM 特点及算法改进

ROAM 作为一种动态的 LOD, 具有以下优点:

(1) 常规静态 LOD 技术将模型分成固定的不同层次, 根据视点同模型在三维空间的距离选择不同细节程度的模型, 不同细节程度的模型之间存在较大变形差异^[8]; 而 ROAM 能够动态地将场景中的不同部分进行 LOD 划分, 能较好地逼近真实地面形态。

(2) 静态 LOD 技术以 2 的平方数目划分地面模型的层次, 若选择不同层次的模型, 计算的三角形数量相差很多; ROAM 根据三角形空间误差动态划分地面模型, 使模型三角形的数量保持基本稳定。例如, 对边长为 1 024 的区域实现 ROAM, 平均生成的三角形数量在 10 000 个左右。

(3) ROAM 技术对空间误差的计算方法可以根据需要灵活选择, 可以选择三角形法向量变形或纹理变形来达到其它应用目的。

ROAM 技术存在一些不足。当以三角形投影误差和视点至三角形的距离作为划分三角形的尺度时, ROAM 对视点距离非常敏感。当距离地形表面非常近时, 随着视点的移动, 地表形状会由于 ROAM 的分割或合并而出现明显变化。对于远距离地形显示系统, 此问题可以忽略; 而对于近距离观察系统, 则需要对算法进行改进。

通过对算法的分析可知, 这些问题的产生是由于当视

点距离模型较近时,依然采用三角形投影误差和视点至三角形的距离的比值来作为界定划分模型的尺度,使得模型的划分层次易受视点距离的移动而改变。例如,当视点接近模型的时候,视点向前移动很小距离,而计算得到的比值却会导致模型被重新划分。这样,用户看到的结果就是在距离模型很近的情况下,很小的移动却会导致地形表面模型被不断地重新分割或合并。

针对这一问题,该文的改进方法是:首先使用垂直的分割平面将地形模型进行 BSP 分割,然后根据不同分割块和视点的距离来实施 ROAM 算法。这样通过 ROAM 来对三角形进行分割和合并,可以将场景中三角形的 ROAM 细节层次稳定在一定范围内;同时,由于使用 ROAM 自底而上的剖分方法,依然可以保持三角形剖分的连续性,使整个地形模型中三角形的数量维持在一个稳定的数目内。在 BSP 分割中,计算划分区域的投影误差,选择分割平面,从而保证地形模型会依据空间形状的几何误差而划分到不同的区域中。

在 BSP 分割中使用竖直方向的分割面,根据分块区域各个地形块的投影误差对分块区域依次使用东西和南北纵向的分割面进行 BSP 划分。划分中保证在分割面同一侧的投影误差不大于另一侧的投影误差,如图 4 所示。

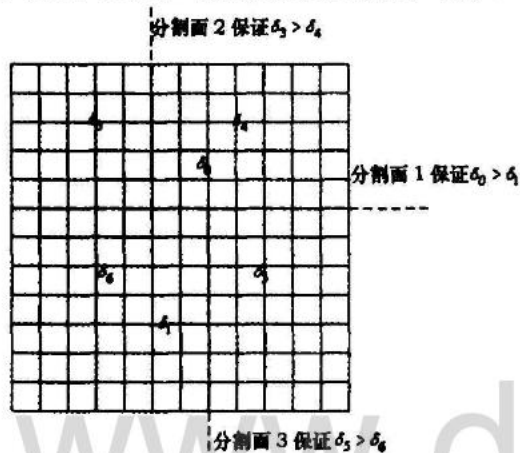


图 4 BSP 分割示意图

为达到较好的执行效率,具体实现时 BSP 分割面的选择要尽量保证分割后两侧的区域具有相等的地形区域。为此,可以对整个分块区域中所有可能的分割面进行计算,然后选择具有最好划分效果的分割面。

渲染地形时,根据 BSP 分割后区域的投影误差和该区域同视点的距离来计算整个区域的 ROAM 模型层次。当视点在一个 BSP 分割区域范围内移动时,始终保持此区域的 ROAM 模型层次,就可以避免模型在距离视点很近时会因为视点距离改变而出现明显变化。

4 ROAM 算法实现与结果分析

实验中采用 1024×1024 的数字地形模型,使用 ROAM 技术,平均每帧的三角形数量保持在 1 万个以上,此数目大大小于不使用 LOD 技术的 200 万个三角形数量,也小于使用静态 LOD 技术的 15 万个三角形数量。可以看出,ROAM 能很好地降低硬件填充压力。

同时,对生成的模型的准确程度进行评价,以投影误差

作为评价的准则,评价公式为:

$$\delta = \sum \delta_i / d_i$$

其中, δ_i 表示单个三角形的变形误差, d_i 表示三角形距离观察视点的距离。

其误差结果为:不使用 LOD 的地形模型的误差为 0,使用静态的 LOD 的误差为 0.2,而使用 ROAM 的误差为 0.26。可以看出,ROAM 技术能够比较真实地反映地形。具体效果如图 5、图 6 所示。

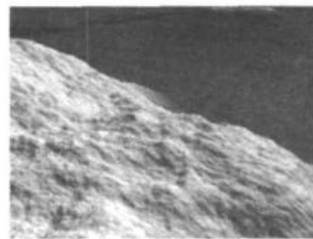
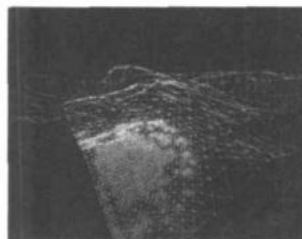


图 5 ROAM 算法网格演示图 图 6 ROAM 算法地形演示图

算法实现过程中采用两个队列管理 ROAM 对三角形的分割和合并,首先以 BSP 划分的区域为基础进行分割,用来避免 ROAM 对视点距离的敏感问题;同时,以当前需要渲染的前一帧为基础进行合并计算,以保持地形模型的连贯性。

根据实验结果,我们比较了 ROAM 算法改进前后的性能和效果。

首先,实际程序的运行中,改进算法由于需要使用 BSP 区域分割,所以要对整个区域进行预先的初始化,而原算法不需要初始化,对于以上实验地形数据进行 5 层的 BSP 分割,改进算法的初始化时间会在 40 至 60 秒之间。

其次,初始化之后,由于改进算法不必在每次渲染时重新计算各个三角形的投影误差,平均帧数会高于原算法 20% 左右。在上面的实验中,改进算法的平均帧数可以保持在 35fps 左右,而原算法在 27fps 左右;同时,改进算法在效果上较好地改善了 ROAM 算法对距离敏感的问题,距离视点较近的一定范围内的地形模型会保持在同一个 ROAM 划分层次上,不会根据视点的移动出现明显改变。

最后,为避免每一次程序运行都需要对整个地形区域进行预先的 BSP 分割初始化,可以对程序的地形区域进行一次 BSP 分割,然后将结果保存下来。以后每次程序开始执行时就不需要重新进行 BSP 初始化,以提高改进算法的初始化时间。

5 结束语

ROAM 技术能够大量减少地形可视化系统中渲染三角形的数量,并可以较好地接近真实地形情况。该技术可以广泛地应用于三维可视化、仿真领域中。本文提出的改进算法可以较好地逼近真实地面形态,并能避免 ROAM 对视点距离的敏感性,满足三维地形场景显示的应用需求。

参考文献:

- [1] 蔡兴泉,李凤霞,战守义.动态地形可视化算法研究[J].计算机工程与应用,2005,41(1):36-37.

(下转第 88 页)

表1 肺炎的原始诊断数据决策表

	size	a	b	c	e	d
x1	100	2	0	1	0	1
x2	200	1	0	2	0	1
x3	150	1	2	0	1	2
x4	60	0	1	3	2	1
x5	250	2	0	1	1	0
x6	90	2	1	0	1	2

步骤1:首先计算决策表1条件属性集C相对于决策属性集D的核 $CORE_C(D)$,容易得到 $CORE_C(D) = \{e\}$ 。因此, $S = \{e\}$, $T = \{a, b, c\}$ 。

步骤2:计算得到属性e的D正域 $POS_{\{e\}}(D) = \{x1, x2, x4\}$ 。因此,在初始状态,有 $X = \{x3, x5, x6\}$,初始状态如决策表2所示。

表2 初始表

	size	e	d
x3	150	1	2
x5	250	1	0
x6	90	1	2

步骤3:设阈值 $k=1$,停止条件为 $k \geq 1$ 。由于 $k=3/6=1/2 < 1$,显然S不是最小约简。

步骤4:分别把T中的属性 $\{a\}$ 、 $\{b\}$ 、 $\{c\}$ 加入到S中,容易得到如下结果: $POS_{\{a,e\}}(D) = \{x3\}$, $POS_{\{b,e\}}(D) = \{x3, x5, x6\}$, $POS_{\{c,e\}}(D) = \{x3, x5, x6\}$ 。如果利用MSH,则有如下结果: $max_size(POS_{\{b,e\}}(D) \setminus \{b, e, d\}) = 250$, $max_size(POS_{\{c,e\}}(D) \setminus \{b, e, d\}) = 250$ 。显然,对于属性b、c,利用MSH无法判断出应该选择哪个属性,MSH已经失去作用。我们利用经过修正的WSH启发函数计算,容易得到 $POS_{\{b,e\}}(d=0) \setminus IND(\{b, e\}) = \{x5\}$, $POS_{\{b,e\}}(d=2) \setminus IND(\{b, e\}) = \{\{x3\}, \{x6\}\}$, $W(S, b) = 212.5$ 。同理,若加入属性c,可以得到 $W(S, c) = 245.1$ 。

步骤5:显然, $F(S, c) > F(S, b)$,因此选择条件属性c加入S中, $S = \{c, e\}$, $T = \{a, b\}$ 。

步骤6:此时,对于论域U由于 $POS_S(D) = U$, $X = \emptyset$ 。搜索过程停止,输出某大型医院肺炎原始决策表的最小约简为 $RED = S = \{c, e\}$ 。

通过实验,MSA算法运行一定时间后无法得到最小约简。通过上述分析可以发现,进行到步骤4时出现了两个相同的MSH值。根据MSA算法无法判断应该选择属性b或c,并且MSA算法根本没有考虑到这种情形。本文提出的IMSA算法很快得出了最小约简。

5 结束语

本文提出了一种属性相对约简启发式算法IMSA,并通过其在某大型医院肺炎病例中的应用实例表明,该算法能够得到决策表的最小约简,验证了算法的有效性和可行性,并且提出了一种基于Rough集的启发式特征提取函数WSH。与MSH相比,WSH考虑了所有隐藏规则集的质量而不仅仅考虑最重要规则集的支持度。

今后,我们将在该算法的时间复杂度和空间复杂度方面做进一步的优化,以期能够在算法复杂度方面有一定的优化。另一方面,我们也将在该算法求解最小约简的完备

性问题的理论上进行研究,最终形成一个在理论上完备、实际应用中比较成熟的算法。

参考文献:

- [1] Pawlak Z. Rough Sets[J]. International Journal of Information and Computer Science, 1982, 11(5): 341-356.
- [2] 刘清. Rough集及Rough推理[M]. 北京:科学出版社,2001.
- [3] Zhong N, Dong J Z, Ohsuga S. Using Rough Sets with Heuristics for Feature Selection[J]. Journal of Intelligent Information Systems, 2001, 16(2): 199-214.
- [4] 王国胤. Rough集理论与知识获取[M]. 西安:西安交通大学出版社,2001.
- [5] 苗夺谦, 胡桂荣. 知识约简的一种启发式算法[J]. 计算机研究与发展, 1999, 36(6): 681-684.

(上接第68页)

- [2] 黄超超, 凌永顺, 吕相银. ROAM动态地形渲染算法研究[J]. 计算机仿真, 2005, 22(1): 216-219.
- [3] Duchaineau M. ROAMing Terrain: Real-Time Optimally Adapting Meshes[A]. Proc of the Conf on Visualization[C]. 1997. 81-88.
- [4] Vincent L, Soile P. Watersheds in DigitM Spaces: An Efficient Algorithm Based on Immersion Simulations[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 1991, 13(6): 583-598.
- [5] 涂超. ROAM算法原理及其应用研究[J]. 辽宁工程技术大学学报, 2003, 22(2): 176-179.
- [6] 柯希林. 基于RSG虚拟地形环境的ROAM实时绘制[J]. 测绘科学与工程, 2003, 24(1): 38-41.
- [7] [美]Shreiner D. OpenGL参考手册[M]. 孙守迁, 王剑, 林宗楷, 等译. 北京:机械工业出版社, 2001.
- [8] 廖巍, 汤晓安, 陈宏盛, 等. 一种视点相关的多分辨率地形实时绘制方法[J]. 计算机工程与科学, 2003, 25(6): 9-12.

(上接第75页)

- [2] Aydin H, Melhem R, Mossé D, et al. Optimal Reward-Based Scheduling for Periodic Real-Time Tasks[J]. IEEE Trans on Computers, 2001, 50(2): 111-130.
- [3] Pillai P, Shin K G. Real-Time Dynamic Voltage Scaling for Low Power Embedded Operating Systems[A]. Proc of the Symp on Operating Systems Principles[C]. 2001. 89-102.
- [4] Hong I, Potkonjak M, Srivastava M B. On-Line Scheduling of Hard Real-Time Tasks on Variable Voltage Processor[A]. Proc of ICCAD'98[C]. 1998. 653-656.
- [5] Aydin H, Melhem R, Mossé D, et al. Power-Aware Scheduling for Periodic Real-Time Tasks[J]. IEEE Trans on Computers, 2004, 53(10): 584-600.
- [6] Alenawy T A, Aydin H. Energy-Constrained Performance Optimizations for Real-Time Operating Systems[A]. Workshop on Compilers and Operating Systems for Low-Power[C]. 2003.