

中图法分类号: TP391.9 文献标志码: A 文章编号: 1006-8961(2011)07-1262-07

论文索引信息:王永志, 盛业华, 周良辰, 郭飞, 王永君, 张平飞. 基于空间扫描策略的 3 维相交检测算法 [J]. 中国图象图形学报, 2011, 16(7): 1262-1268

基于空间扫描策略的 3 维相交检测算法

王永志, 盛业华, 周良辰, 郭飞, 王永君, 张平飞

(南京师范大学虚拟地理环境教育部重点实验室, 南京 210046)

摘要: 针对复杂虚拟场景中碰撞检测和空间分析等操作实时性差的问题, 提出一种适用于 3 维空间对象间的快速相交检测算法——Space Sweep。该算法首先根据场景内空间对象分布特征, 构建事件点及其列表; 利用空间扫描策略, 自适应地构建一系列假想的空间扫描面; 在扫描面移动的过程中, 将空间对象的状态分为死亡态、激活态和休眠态, 通过只对当前处于激活态的空间对象进行相交测试, 有效地减少了空间对象间不必要的相交计算。该算法提高了虚拟场景中 3 维空间对象间相交检测的效率, 为 3D GIS 中实时空间分析提供了有力的技术支持。最后, 通过对比测试验证了本文算法的实用性。

关键词: 相交检测; 空间扫描策略; 红蓝思想; 虚拟地理环境; 3D GIS

Intersection detection algorithm for 3D objects utilizing spatial sweeping strategy

Wang Yongzhi, Sheng Yehua, Zhou Liangchen, Guo Fei, Wang Yongjun, Zhang Pingfei

(Key Laboratory of Virtual Geographic Environment, MOE, Nanjing Normal University, Nanjing 210046 China)

Abstract: To resolve the problem that the traditional collision detection algorithm and spatial analysis operations in the virtual scene have bad real-time performance, a fast intersection detection algorithm named SpaceSweep for three-dimensional spatial objects is presented. Based on the distribution characters of spatial objects, the event points and their lists are constructed firstly. After that, by using the spatial sweeping strategy, a series of imaginary spatial sweeping planes are constructed adaptively. During the sweeping process, the state of the spatial objects is classified into death state, active state and sleeping state. Only spatial objects in the active state are used to conduct of the intersection testing. Therefore, many unnecessary intersection calculations are abstained, and the efficiency of the algorithm is improved. This algorithm improves the efficiency of intersection detection between the three-dimensional objects in the virtual scene, and provides strong technical support for real-time spatial analysis in 3D GIS. At last, the practicality of this algorithm is validated by comparing tests.

Keywords: intersection detection; space sweep strategy; red and blue methods; virtual geographic environments; three-dimensional geographic information system

0 引言

在 GIS (geographic information system) 领域, 根

据 OGC 标准, 空间拓扑关系分析的实现是基于维度扩展的 9 交矩阵 (DE-9IM), 即几何要素间的相交检测是空间拓扑分析的基础。虚拟地理环境中包含了大量的 3 维地理对象, 场景非常复杂, 地理对象间

收稿日期: 2010-03-03; 修回日期: 2010-09-17

基金项目: 国家高技术研究发展计划项目 (2007AA12Z236, 2007AA12Z207, 2008AA121604); 国家自然科学基金项目 (40801147, 40671147)。

第一作者简介: 王永志 (1983—), 男, 南京师范大学地图学与地理信息系统专业博士研究生, 主要研究方向为虚拟地理环境和 3 维地质建模与模拟。E-mail: gisstaranu@gmail.com。

的两两相交检测,时间耗费非常大,难以满足拓扑分析实时性的需求。而且,随着3维地理对象建模技术的发展,模型越来越逼真,越来越详细,虚拟(地理)场景规模越来越大,空间拓扑分析的效率受到了严峻的挑战。要实现实时的空间拓扑分析,寻求一种高效且可以应用于大数据量复杂的虚拟场景中的相交检测算法,成为亟待解决的问题。此外,在计算仿真、虚拟现实、3D游戏、机器人运动规划等领域,要利用计算机构建具有真实感的虚拟世界,高效的相交检测算法也是必不可少的技术。

近年来,国内外学者对如何提高3维空间对象间相交检测效率进行了大量的研究,提出了许多有效的检测方法,主要分为空间分解法和层次包围盒法两类^[1]。下面对这两类方法进行简要的归纳和总结。

1)空间分解法。Ganter和Isarankura首先将空间分割技术引入相交检测中^[2-3],提出了一种空间分解方法,将包含几何对象的空间划分成一系列子空间,只在两个几何对象的重叠子空间进行相交检测,进而减少了几何对象两两相交计算的数目。典型的空分解方法有多维二叉树(K-D tree)、八叉树(octree)、空间分区二叉树(BSP tree)、四面体网(tetrahedral mesh)、规则网格(regular grid)等。空间分解法适用于空间分布均匀的稀疏场景中几何对象的相交检测,对于一般的场景,最优的剖分尺度很难选择。如果分解过度,会耗费大量的计算时间;如果分解的子空间数目太少,会发生漏判或错判现象。因此,空间分解法在实际的相交检测中应用较少。

2)层次包围盒法。Hahn首次采用层次包围盒技术加速多面体场景的相交检测^[4],通过用几何特性简单的包围盒层次逼近复杂几何对象模型,从而用包围盒来代替复杂几何对象进行相交检测,减少不必要的相交计算。层次包围盒法可以应用于复杂场景中的相交检测,是实际应用较为广泛的方法。其典型的方法有包围球(sphere)、轴向包围盒AABB(axis-aligned bounding box)、固定方向凸包FDH(fixed directions hulls)、方向包围盒OBB(oriented bounding box)、离散方向多面体K-Dop(discrete orientation polytopes)等^[5-8]。其中,AABB几何特性简单,易于相交判断;OBB方向任意性,对几何对象包围紧密。这是目前应用较多的两种包围盒。

平面扫描(sweep line)技术是计算几何领域一个通用的技术,该技术于1976年由Shamos和

Hoey提出后在计算几何领域中得到广泛应用^[9]。1979年Bentley和Ottmann将平面扫描技术引入平面线段交点算法中,提出了BO算法^[10],该算法把效率和实际线段相交个数联系起来,提高了平面线段交点算法的效率,属于“输出敏感型”算法。Domiter和Zalik对2维散乱点集进行带边界约束的Delaunay三角剖分时,应用平面扫描技术将研究区域划分成已经三角剖分的区域与未进行三角剖分的区域,大大提高了Delaunay三角剖分算法的效率^[11]。Krista等人将平面扫描技术应用于空间数据的空间聚类分析中,提出了一种可以处理大数据量空间数据库的空间聚类方法^[12]。Koziara等人将3维包围盒投影到扫描平面,对其进行降维处理,通过2维矩形间的相交计算,来加速3维包围盒的相交判断^[13]。

针对目前3维空间对象间的相交检测算法存在的缺陷,借鉴2维平面扫描技术,提出一种空间扫描策略,进而提出一种基于空间扫描策略的3维相交检测算法——Space Sweep法。该算法的基本思路是:根据空间扫描的分区特性,在虚拟扫描面的移动过程中,将场景内空间对象的状态划分为死亡态、激活态和休眠态。只对处于激活态的空间对象进行相交计算,这样充分利用了检测过程中位于虚拟扫描轨迹前后分区内空间对象之间的联系,对检测过程进行了优化,减少了场景内各个空间对象间大量不必要的相交检测,在显著提高算法效率的同时,也解决了大数据量复杂场景内相交检测的实时性问题。

1 相关概念及空间扫描策略

定义1 空间扫描面(SP)为3维空间中一个假想的平面,其方程为

$$\mathbf{AX} + \mathbf{E} = \mathbf{0} \quad (1)$$

式中, $\mathbf{E} = (e_1, e_2, e_3)$ 为扫描面的起始点, $\mathbf{X} = (x, y, z)$ 为扫描面上任意一点, $\mathbf{A} = (a_1, a_2, a_3)$ 为参数,且满足

$$\begin{cases} a_1 \times a_2 \times a_3 = 0 \\ a_1 + a_2 + a_3 \neq 0 \end{cases} \quad (2)$$

定义2 在扫描的过程中SP会在某些特殊的点停顿,进行一些分析操作,这些点被称为事件点。事件点可以是任何分析算法感兴趣的空要素,包括对象之间的交点和特征点等,具体定义为

$$Event(X_i) =$$

$$X_i \mid \{Intersect\ Point\ Set, Feature\ Point\ Set\} \quad (3)$$

式中, $X_i = (x, y, z)$, 代表空间位置。

定义 3 算法在事件点位置对空间对象进行的所有分析和操作, 以及对空间对象进行的称为事件点触发的动作。在相交检测的算法中, 主要是空间对象间的相交计算, 具体定义为

$$Act(P) = Event(P) \mid \{Analyze\ Operation\} \quad (4)$$

式中, P 代表事件点。

定义 4 事件点依照算法确定的空间排序关系进行存储, 从而构成的事件点集合称为事件点列表。事先能够确定的事件点列表在扫描过程中不再变化, 称为静态事件点列表; 需要在扫描过程中动态更新的事件点列表称为动态事件点列表。

将 2 维空间中的平面扫描技术拓展到 3 维空间, 参考 2 维空间中的定义给出空间扫描策略的描述如下^[10, 14-15]:

假设 3 维场景中有一个空间对象的集合 $S = \{E_1, E_2, \dots, E_n\}$, 空间扫描面 SP 从事件点的起始端向未末端移动, 在 SP 对场景扫描的过程中, 动态维护 $S \cap SP$ 的空间对象集合, 同时要满足以下两个规则:

规则 1 位于 SP 左端的空间对象已经参与了分析操作, 将不再参与以后的分析操作。

规则 2 位于 SP 右端或与 SP 相交的空间对象是将要进行分析操作的对象, 需要对这些空间对象的集合进行实时更新。

在 SP 扫描过程中, 根据每个空间对象与扫描面 SP 的关系, 可以将空间对象集合 S 分为以下 3 种状态的空间对象 (如图 1 所示, SP 是当前进行的扫描面, 虚框表示的是已经或者即将进行的扫描面, T1—T23 是一系列空间三角面对象)。

1) 死亡态 (dead state) 空间对象, 位于扫描面 SP 左侧的空间对象, 它们是已经进行过相交测试的空间对象, 将不再参与求交计算;

2) 激活态 (active state) 空间对象, 与扫描面 SP 相交的空间对象, 它们是正在进行相交测试的空间对象;

3) 休眠态 (sleeping state) 空间对象, 位于扫描面 SP 右侧的空间对象, 它们是暂时不参与相交测试的空间对象, 只有当扫描面到达它们所处的事件

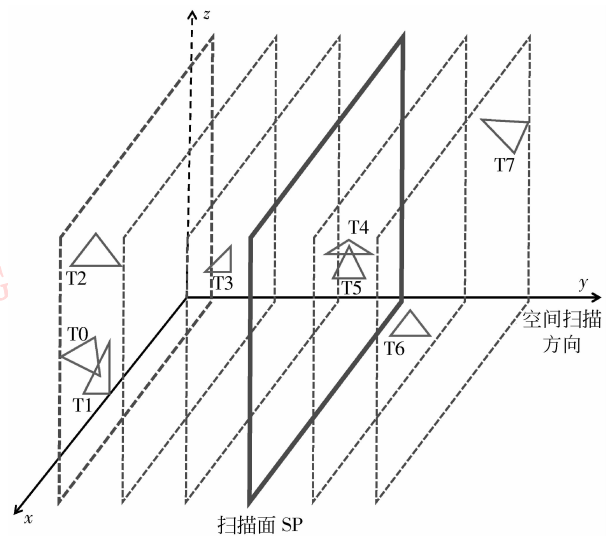


图 1 空间对象 3 种状态

Fig. 1 State of spatial objects

点时, 才触发相应的求交计算。

在相交检测的过程中, 随着扫描面的移动, 每个空间对象的状态发生相应转变, 转变过程如图 2 所示。仅对处于激活态的空间对象进行求交计算, 而不考虑处于死亡态和休眠态的空间对象。因此, 利用空间扫描策略进行相交检测, 可以有效地减少实际求交运算的次数, 进而提高检测效率。

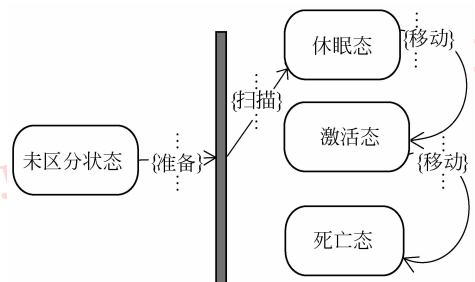


图 2 空间对象状态转变图

Fig. 2 Transform of spatial objects

2 算法描述

为了加快相交检测的过程, 将红蓝思想引入本算法中^[16-19], 通过对不同空间对象的分组, 来加速复杂场景中空间对象间的相交检测。算法流程如图 3 所示。

2.1 空间对象分组和事件点列表的构建

算法首先从外部读取一系列需要进行相交检测

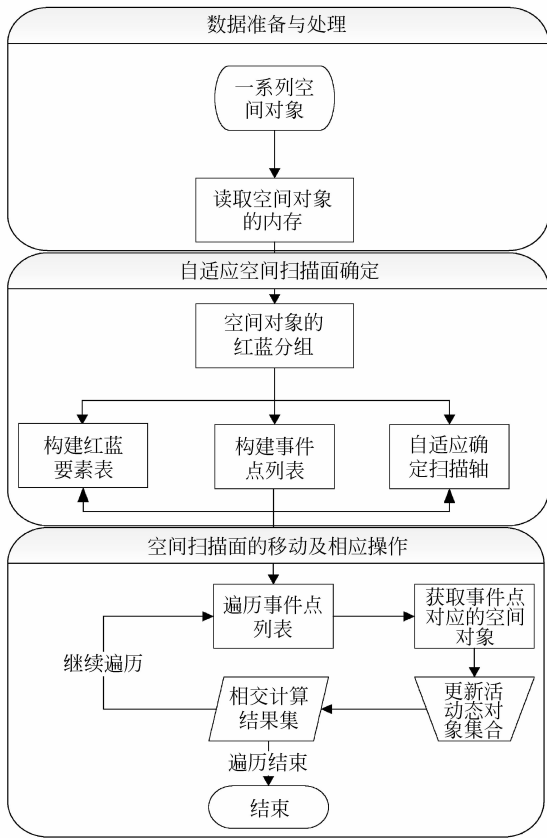


图 3 SpaceSweep 算法的流程图

Fig. 3 Flow chart of SpaceSweep algorithm

的空间对象,按照用户预定义的属性(或空间对象本身具有的分类性质),作为红蓝分组的标准,分别构建红蓝要素分组列表(RedFeatureTable 和 BlueFeatureTable)。在读取每个空间对象的同时,将其 X 最大值 x_{\max} 、 Y 最大值 y_{\max} 、 Z 最大值 z_{\max} ,作为事件点,分别放入顺序表 $xTable$ 、 $yTable$ 、 $zTable$ 中。为了剔除相同值,可以将 3 个顺序表定义为 Set 集合类型。当所有对象都读取完毕后,选择顺序表 $xTable$ 、 $yTable$ 、 $zTable$ 中记录最多的作为事件点列表(EventTable),并对事件点列表中的记录按照坐标从大到小的顺序进行排列。

其中,空间对象和红蓝对象分组列表对象的结构(实线框表示具有的操作,在此只给出了名称、返回值类型和参数列表省略;虚线框表示具有的属性)如图 4 和图 5 所示。

2.2 3 维空间扫描面的自适应确定

基于空间扫描策略的 3 维相交检测算法核心在于通过空间扫描面 SP 将空间对象分为死亡、活动、休眠 3 个状态,仅对处于活动状态的对象进行实际

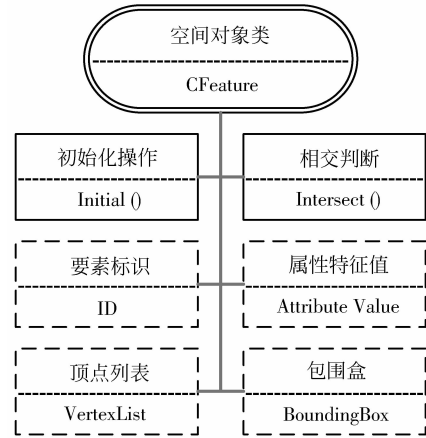


图 4 空间对象类结构图

Fig. 4 Structure of spatial object class

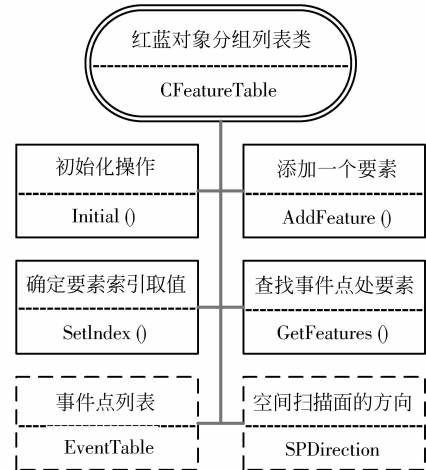


图 5 红蓝对象分组列表类结构图

Fig. 5 Structure of red-blue object class

相交计算,从而减少相交计算的数量,加快检测的速度。因此,有效地对空间对象进行划分,减少处于激活态的空间对象个数,是提高算法效率的关键。本文依据 X 、 Y 、 Z 方向空间对象分布的特征,空间扫描的轴向为

$$\Psi = \max \{ \Psi_x, \Psi_y, \Psi_z \} \quad (5)$$

式中: $\Psi_j = \sum_{i=0}^n Act(S_i)$, $j = x, y$ 或 z ; $Act(S_i)$ 是对处于活动态空间的对象触发的动作计数器。

空间扫描面的确定,最重要的步骤是扫描轴的确定,算法实现过程中通过调用式(3),可以自动计算出扫描轴的方向(X 、 Y 或 Z 轴向),进而可以根据第 1 节中扫描面的定义,自适应地完成空间扫描面的构建。这种自适应方式进行空间扫描面的构建,

使得对空间对象的状态划分更加合理,能够有效地减少同时处于激活态的对象个数,进而减少实际相交计算次数。

2.3 空间扫描面的移动与事件点触发的动作

空间扫描面按照事件点列表中事件点的顺序,从表头逐步向表尾移动,并在事件点位置触发相应的动作。事件点触发的动作包括更新空间对象的状态、实时更新激活态对象集合以及对处于激活态的空间对象间进行相交计算。

以 Z 轴向事件列表和平行于 XY 平面的空间扫描面为例进行阐述,当空间扫描面从前一个事件点 (PreviousEvent) 移动到当前事件点 (CurrentEvent) 时,分别从红蓝对象列表中查找 CurrentEvent 事件点所对应的空间对象,即所有 z_{\max} 等于 CurrentEvent 值的空间对象;从激活态对象集合 (ActiveFeatureSet) 中查找与 CurrentEvent 事件点满足某种逻辑关系的空间对象,即所有 z_{\min} 大于 CurrentEvent 值的空间对象。将 CurrentEvent 所对应的空间对象的状态从休眠态改为激活态,并将其添加到激活态对象集合中;将与 CurrentEvent 满足逻辑关系的空间对象的状态从激活态改为死亡态,并将其从激活态对象集合中删除。

在每次往激活态空间对象集合中添加空间对象时,将其与集合内其他空间对象进行相交计算操作。为了加快计算的速度,先利用包围盒技术,判断参与相交计算的空间对象的包围盒之间的空间关系,如果两个包围盒不相交,那么这两个空间对象也不会相交,这样可以进一步减少不必要的相交计算操作。如果两个空间对象相交,则将其添加到计算结果集中。

空间扫描面移动和事件点触发的动作的主要算法描述如下:

1) 将空间扫描面从前一个事件点处移动到当前事件点处;

2) 更新激活态对象集合中对象状态,移除死亡态的对象;

3) 从红对象分组列表中获取当前事件点对应的空间对象集合 (RedFeatureSet);

4) 将 3) 中 RedFeatureSet 中的对象状态从休眠态变为激活态,添加到激活态对象集合中,并执行相应的操作;

5) 对蓝对象分组列表执行 3) 和 4)。

3 实验与分析

本文算法的理论时间复杂度为 $O(n \log 2n)$, 用 C++ 实现。为了验证本文算法,在相同的系统测试环境下,在 PC (主频 1.8 GHz, 内存 1 GB) 上,设计了 2 组实验: 1) 对不同数据模型检测效果实验; 2) 与同类方法性能对比实验。

3.1 对不同数据模型检测效果实验

实验采用的数据模型为地质体、矿体和兔子,实验相关参数及其运行时间统计结果如表 1 所示,检测效果如图 6 所示。

表 1 不同数据模型检测效率对比

Tab. 1 Testing results of three data models

数据模型	三角形对数	相交对数	时间/ μs
矿体	6 130	889	100
地质体	8 384	641	212
兔子	7 400	553	161

综合表 1 及图 6 可以看出,本文的算法能够准确、快速地检测出场景中空间对象的相交情况。

3.2 与同类方法性能对比实验

在该部分的实验中,为了能够更清晰直观地检验本文算法的效率,排除空间索引对其效率的影响,对 3 维空间对象没有采用索引组织。选用基于 OBBTree 的 RAPID 相交检测算法和基于扫描球包围的 PQP 相交检测算法,与本文算法进行对比测试。为了进行对比测试,分别利用具有 500 个三角形对,1 000 个三角形对,5 000 个三角形对,10 000 个三角形对构成的实体表面三角网模型作为实验样本数据。利用 SpaceSweep 法 (本文算法)、RAPID 法和 PQP 法进行了相交检测,算法的测试结果统计表和效率对比图,分别如表 2 和图 7 所示。

通过实验可以看到,本文算法可以有效地减少实际进行相交计算的三角面片的对数,从而大大提高相交检测的效率。此外,RAPID 和 PQP 这两种算法对空间对象模型的基本组成单元做了限定,要求基本几何单元必须为三角面片。本文算法在实现的过程中,不必考虑空间对象模型的基本几何单元,从而打破了这种限定,具有更广泛的实用性。

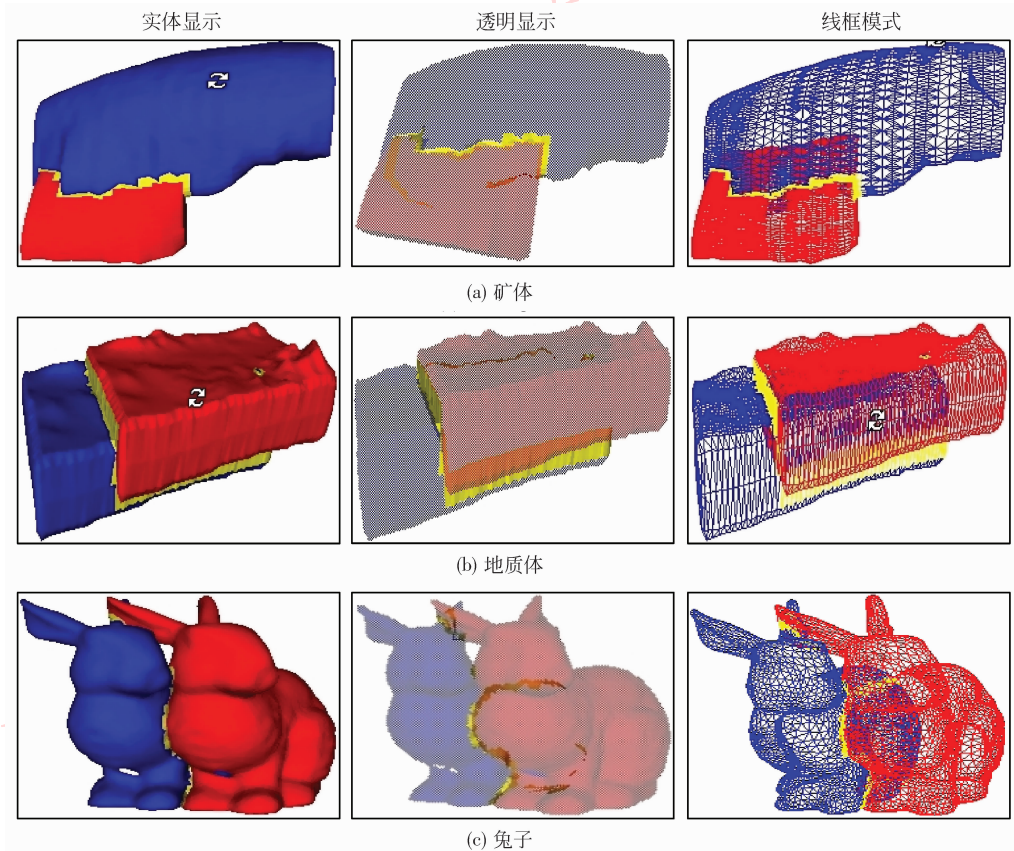


图 6 3 种数据模型检测效果图

Fig. 6 Testing results of three data models

表 2 测试结果统计表

Tab. 2 Testing results of three algorithms

方法	几何单元数目							
	500 对		1 000 对		5 000 对		10 000 对	
	相交对数	时间/ μs	相交对数	时间/ μs	相交对数	时间/ μs	相交对数	时间/ μs
SpaceSweep	59	6	239	13	5 800	133	22 363	457
Rapid	59	7	239	17	5 800	176	22 363	534
PQP	59	44	239	99	5 800	868	22 363	2 530

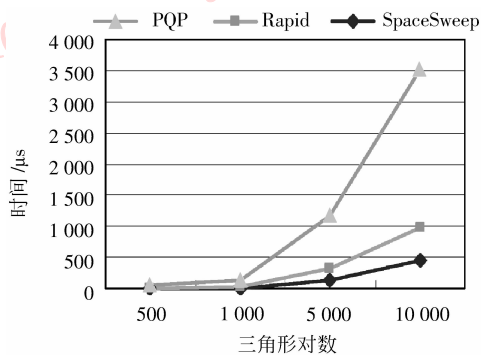


图 7 效率对比图

Fig. 7 Efficiency contrast chart

3.3 算法分析

一般来说,该类算法的消耗可以表示为

$$T = N_v \times C_v + N_p \times C_p + N_u \times C_u + C_d \quad (6)$$

式中, T 是相交检测的总耗费; N_v 是参与相交测试的包围盒的对数; C_v 是一对包围盒相交测试的耗费; N_p 是参与求交计算的基本几何单元的对数; C_p 是一对基本几何单元求交计算的耗费; N_u 是空间扫描面移动时关联对象的个数; C_u 是每个相关联的空间对象进行状态转变需要的耗费; C_d 是指内存开辟、数据遍历等其他耗费。

由于采用的是几何特性简单的 AABB 包围盒,

只需要 M 次 ($M \leq 6$) 比较运算。通过采用空间扫描策略,仅对少量处于激活态的空间对象进行相交检测,这样使 N_v 和 N_p 大大减少。同时,采用了红黑树来管理数据模型和事件点,算法在 C_u 和 C_d 方面的耗费非常少,与前面两项相比基本可以忽略。

4 结 论

相交检测不仅是计算几何领域的基本问题,也是 3D GIS 空间拓扑分析的基础。因此,提高相交检测算法的执行效率,也是 3D GIS 的核心问题。基于此,设计了一种基于空间扫描策略的 3 维相交检测算法。本文算法不仅可以实现简单场景下两两空间对象间的实时相交检测,而且可以实现复杂场景的快速相交检测。由于本文算法不必考虑空间对象模型的构成单元,因此,打破了目前大多数相交检测算法对基本几何单元的限定,具有更广泛的实用性。

参考文献 (References)

- [1] Liu Jianxin, Cui Hanguo, Zhang Jing, et al. Optimization of box-based algorithms for collision detection [J]. Computer Engineering and Applications, 2008, 44(18) : 51-53. [刘健鑫, 崔汉国, 张晶, 等. 包围盒碰撞检测算法的优化 [J]. 计算机工程与应用, 2008, 44(18) : 51-53.]
- [2] Ganter M A, Isarankura B P. Dynamic collision detection using space partitioning [J]. Journal of Mechanical Design, 1993, 115(1) : 150-155.
- [3] Zou Yisheng, Ding Guofu, He Yong, et al. Fast intersection algorithm between spatial triangles [J]. Application Research of Computers, 2008, 25(10) : 2906-2910. [邹益胜, 丁国富, 何邕, 等. 空间三角形快速相交检测算法 [J]. 计算机应用研究, 2008, 25(10) : 2906-2910.]
- [4] James K, Hahn. Realistic animation of rigid bodies [C] // Proceedings of SIGGRAPH' 88; Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques. New York, USA: ACM, 1988: 299-308.
- [5] Gino van den, Bergen. Efficient collision detection of complex deformable models using AABB trees [J]. Journal of Graphics Tools, 1999, 4(2) : 1-13.
- [6] Gottschalk S, Lin M C, Manocha D. OBBTree: a hierarchical structure for rapid interference detection [C] // Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. New York, USA: ACM, 1996: 171-181.
- [7] Klosowski J T, Held M, Mitchell J S, et al. Efficient collision detection using bounding volume hierarchies of k-dops [J]. IEEE Transactions on Visualization and Computer Graphics, 1998, 4(1) : 21-36.
- [8] Pobil A P, del, Serna M A. A new representation for robotics and artificial intelligent application [J]. International Journal of Robotics & Automation, 1994, 9(1) : 11-21.
- [9] Shamos M I, Hoey D. Geometric intersection problems [C] // Proceedings of the 17th Annual Symposium on Foundations of Computer Science. Washington, DC, USA: IEEE Computer Society, 1976: 208-215.
- [10] Bentley J L, Ottmann T A. Algorithms for reporting and counting geometric intersections [J]. IEEE Transactions on Computers, 1979, 28(9) : 643-647.
- [11] Domiter V, Zalik B. Sweep-line algorithm for constrained delaunay triangulation [J]. International Journal of Geographical Information Science, 2008, 22(4) : 449-462.
- [12] Alik K R, Alik B. A sweep-line algorithm for spatial clustering [J]. Advances in Engineering Software, 2009, 40(6) : 445-451.
- [13] Tomasz Koziara, Nenad Bicanic. Sweep-plane approach to bounding box intersection [C] // Proceedings of VIII International Conference on Computational Plasticity. Barcelona, Spain: CIMNE, 2005: 1-4.
- [14] Zhao Hongchao. Research and Implementation on Spatial Relation [D]. Beijing: Institute of Computing Technology Chinese Academy of Sciences, 2006. [赵红超. 空间关系的研究和实现 [D]. 北京:中国科学院计算技术研究所, 2006.]
- [15] Wang Mengxiao. Research and Implementation of Spatial Topological Operations Based on Red and Blue Analysis [D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2005. [王梦晓. 基于红蓝思想的空间拓扑分析算法的研究与实现 [D]. 南京:南京航空航天大学, 2005.]
- [16] Mairson H, Stolfi J. Reporting and counting intersections between two sets of line segments [C] // Proceedings of Theoretical Foundations of Computer Graphics and CAD. Berlin, Germany: Springer Verlag, 1988: 307-325.
- [17] Julien Basch, Guibas L J, Ramkumar G D. Reporting red-blue intersections between two sets of connected line segments [J]. Algorithmica, 2002, 35(1) : 1-20.
- [18] Palazzi L, Snoeyink J. Counting and reporting red/blue segment intersections [J]. Lecture Notes in Computer Science, 1993, 709: 530-540.
- [19] Chazelle B, Edelsbrunner H. An optimal algorithm for intersecting line segments in the plane [J]. Journal of the ACM (JACM), 1992, 39(1) : 1-54.