

改进的通视性检查算法*

梁晓辉, 赵沁平

(北京航空航天大学 计算机科学与工程系,北京 100083)

E-mail: lxx@vrlab.buaa.edu.cn

http://www.buaa.edu.cn

摘要: 对虚拟环境中点到点的通视性检查算法进行了研究.在深入分析了一个目前具有代表性的投影覆盖检测(sieve overlap,简称 SO)算法的基础上,对该算法进行了改进,使其更适合大规模分布式虚拟环境及各种仿真的需要.SO 算法利用桶表和单线索二叉树进行地形多边形的管理及各种检索工作.基于感兴趣区的思想,大幅度缩减了桶表的检索范围,使点到点的通视性检查具有更高的真实性和实时性,能够更为有效地为大规模分布式虚拟环境中的动态实体提供各种环境信息.对 SO 算法及其改进算法 SO*在虚拟环境中应用的时间空间效率进行了分析.

关键词: 虚拟环境;通视性检查算法;点到点的通视性检查;投影覆盖检测算法;动态实体;感兴趣区

中图法分类号: TP391 **文献标识码:** A

虚拟现实是一种可以创建和体验虚拟世界的计算机系统^[1].利用虚拟现实技术,可以将真实世界加以抽象和模型化并转换为计算机系统上的虚拟世界,用户通过交互设备可以与虚拟世界中的各个实体进行交互,从而产生与真实世界相似的感觉和体验.构建一个虚拟现实系统的重要步骤是将现实世界中物体的几何属性加以抽取,通过三维建模构造虚拟环境.虚拟环境的实体通常包括静态实体(地形和地表特征物等)以及参与到环境中的动态实体.

通视性检查(intervisibility test)^[2]是针对虚拟环境中物体的几何模型进行的.它被用来检测虚拟环境中的两个物体可否相互“看”到.通视性检查包括较多的内容,诸如检查在某给定点可否看到某一区域,两个区域的相互可视性及两点是否可视等,本文仅讨论点到点可视性的问题.为完成这一工作,必须生成一条联结两个点的线段并且判断该线段与各种静态的地形和地表特征物是否相交,该线段通常也叫做“视线”(line of sight,简称 LOS).

参与到虚拟环境中的动态物体通常包括两种类型,即由人控制的实体(human in the loop)和由计算机控制的实体(computer generated actor,简称 CGA)^[3].点到点的通视性检查对这两类实体都很重要.对 CGA 而言,通视性检查是该类实体获取环境信息的有效手段,通过获取环境信息决定 CGA 在环境中的行为,对提高 CGA 的智能程度有重要意义.对由人控制的实体而言,可以利用通视性检查的方法模拟一些传感器获取输入,从而为操纵者在环境中的决策提供必要的信息.由此可见,研究点到点的通视性检查算法有着重要的意义.

与其他通视性检查算法相比,点到点的通视性检查算法相对较为简单,但在实现时存在计算量大等较多的问题.例如,环境中存在 N 个实体,一个 CGA 系统需要在每帧做 $O(N^2)$ 数量级的通视性检查工作,这使得系统的实时响应特性受到较大的影响.通视性检查算法的关键在于检查的准确性和快速性,但同时解决这两个问题具有一定的难度,尤其是在大规模的基于真实地形数据生成的虚拟环境中.本文首先对已有的虚拟环境中的点到点通视性检查算法进行研究,分析并提出了一个改进算法;随后对改进算法的效率进行分析;最后给出了今后的主

* 收稿日期: 2002-02-25; 修改日期: 2002-06-11

基金项目: 国家高技术研究发展计划资助项目(2001AA115130)

作者简介: 梁晓辉(1970 -),男,山西太原人,博士,主要研究领域为虚拟现实;赵沁平(1948 -),男,山西沁水人,博士,教授,博士生导师,主要研究领域为虚拟现实,人工智能.

要研究内容.

1 点到点的通视性检查算法及改进

1.1 点到点的通视性检查算法

基本的点到点的通视性检查算法包括如下 4 个步骤:首先将地形多边形库中的多边形和连接两个端点的线段投影到水平面上(设为 $X-Y$ 平面),其次利用点的位置关系确定在 $X-Y$ 平面上包含起点及终点投影的两个投影多边形;再次根据投影多边形的邻接关系在 $X-Y$ 平面找到该线段投影穿过的所有多边形;最后判断出线段与多边形在三维空间是否真正相交^[4].

点到点的通视性检查算法都是基于上述思想实现的.实际上,线段与多边形的求交检测很费时间(在三维空间进行),而且很多算法仅面向特定的地形数据库结构,不具有通用性,再有,算法在运行时还对多边形邻接关系有连续性的要求,因此在实际应用时存在一些问题.一些主要的改进工作集中在对地形多边形的组织上,通过构建一个复杂的地形多边形组织结构,提高求交及判断的效率,减少判断时间.在这些改进算法中,SO 算法^[5]提出了一种通用的地形多边形组织方法,并且具有很高的检查效率,比较有效地解决了已有点到点通视性检查算法的一些问题,因此,我们主要针对该算法进行了分析和改进.

SO 算法利用一个包含连接两点线段,且各边与坐标轴平行的最小包围盒,该包围盒被称作过滤筛(sieve).若把 LOS 投影到 3 个坐标轴,分别在轴上有 3 个线段,Sieve 实际上是定义了这 3 个线段.相应的地形多边形也可以投影到 3 个轴上,产生 3 个线段.该算法基于这个前提,即仅当地形多边形投影的 3 个线段与 Sieve 投影的 3 个线段都重叠时才可能与 LOS 相交.

SO 算法首先建立了一个通用的地形多边形的组织结构,这一结构使用了两个主要的数据结构,一个称为单线索三叉树(singly threaded ternary tree),这是一个类似于二叉检索树的结构,每个节点包含 1 个键值和 3 个指针,指针分别指向比其键值小的节点、比其键值大的节点和与其键值相同的节点.这个树在节点插入和检索时具有较高的效率;另一个结构是桶表(bucket list),其中有若干的桶(bucket),每个桶表中存放了按键值排序的数组,数组中的每一项(桶)指向一棵单线索三叉树.

实际应用该算法时,在预处理之前可根据地形三角形在 3 个轴上的投影段的顶点的坐标范围确定每个桶表的最大表项范围并静态划分,随后在预处理中将每个地形三角形插入到具体的单线索三叉树中.经过预处理阶段后,地形数据库被转化为一个通用的基于桶表和单线索三叉树的结构,便于运行阶段的检索.

算法具体使用时包括 6 个桶表,其中 3 个为预处理阶段生成的 X,Y,Z 方向的表,另外 3 个为运行时的附属结构.具体算法如下:

算法 1. SO 算法.

(1) 构建地形三角形的 X,Y,Z 方向的 3 个桶表.根据 Z 向桶表及 LOS 的 Z 向键值寻找发生覆盖重叠的部分,并生成 Working X 桶表,存放覆盖重叠三角形的 X 方向投影.

(2) 根据 Working X 桶表及 LOS 的 X 向键值寻找发生覆盖重叠的部分,并生成 Working Y 桶表,存放覆盖重叠三角形的 Y 方向投影.

(3) 根据 Working Y 桶表及 LOS 的 Y 向键值寻找发生覆盖重叠的部分,并生成 Test 桶表.存放可能相交的三角形.

(4) 在三维空间判断相交性.

(5) End.

算法实际上只用了一个预处理生成的桶表(Z 向桶表),其具体的过程及原因可参阅文献[5].

1.2 算法的改进

SO 算法的关键在于提供了一个快捷的检索结构,通过这一结构简化了相交性判断过程.该算法对某个桶表的检索时间主要由 3 部分组成,搜索桶表中首个检索项的检索时间 T_1 、确定桶表中其他检索项的时间 T_2 和单线索三叉树的检索时间 T_3 .我们的改进是针对 SO 算法的检索时间 T_1 进行的.

如前所述,点到点的通视性检查是 CGA 或人在回路实体获取信息的手段,因此有较高的实时性要求.在实际构建一个虚拟环境时,需要构造许多静态实体,而实体需要大量的多边形.这些多边形通常被三角化,以三角形的方式加以表示和存储,这些三角形的数量通常达到数十万甚至上百万的数量级,所以桶表的表项很多.虽然桶表已预先排序,可用二分检索的方法进行检索,但表长度的增加使得检索效率降低,我们需要研究一个减少桶表查找表项的方法.

事实上,对于动态实体而言,它们仅在虚拟环境的某个区域内活动,每次检索整个地形数据库无疑是没有必要的.另外,针对具体的动态实体而言,它们都有具体的物理属性,因此感知范围受到这些属性的制约.例如,对现实中的人而言,他的感知器官都有一定的作用范围,不可能覆盖整个环境,因此,让动态实体在整个地形数据库中检索也是不真实的,利用这一特点可以缩小检索空间.

感兴趣区(area of interesting,简称 AOI)是分布式虚拟环境中的一个重要概念,在大规模分布式虚拟环境中引入感兴趣区可以有效地减少需要处理的信息量.在最新的 IEEE 关于分布交互仿真的多层次体系结构(high level architecture,简称 HLA)中^[6],定义 Routing Space 概念就是一类 AOI.AOI 的一个最主要的运用就是在网络通信方面.分布式虚拟环境包含了众多的节点,这些节点作为各种动态实体的化身参与到虚拟环境当中,节点之间的交互是通过在网络中交换数据进行的.这些数据包括实体的状态信息、控制信息以及其他交互信息.信息的数量很大,尤其是实体状态信息,几乎每个仿真循环都要发送.在大规模分布式环境中,这些信息占用了很大的通信带宽,因此在通信中采用了基于 AOI 的方法,仅仅在相互可能发生交互的实体间传递信息,有效地减少了网上的信息流量.AOI 实际上是一个区域,它定义了实体所关心的区域范围.这个范围可以根据实体实际的物理特性来确定.一个实体可以有多个 AOI,例如对实体所拥有的每个传感器定义一个 AOI.如图 1 所示为某实体的仿真区域与 AOI 的比较.

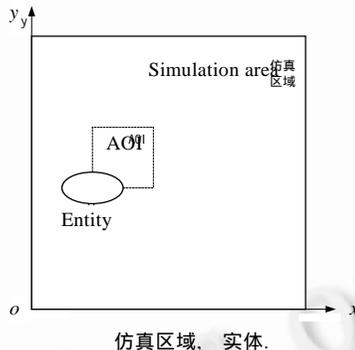


Fig.1 Simulation area and the AOI of entity
图 1 实体仿真区域及其感兴趣区

我们将 AOI 的思想运用到 SO 算法中,对该算法进行了改进,提出了 SO* 算法,使其更适合于大规模的分布式虚拟环境的应用.主要在以下两个方面进行了改进:

(1) 动态实体的 AOI 的确定及表示.

AOI 的范围可以根据动态实体具体的物理特性(如实体的类型、实体所带的传感设备等)来确定.例如,对虚拟的人而言,其视觉和听觉有不同的特征参数,因此可以有不同的 AOI 感知环境的信息.如果实体带有其他传感装置,可以根据传感器的特性来确定其 AOI.AOI 在三维空间采用长方体的方式表示,反映到每个坐标轴上为一条线段.我们在算法的桶表中增加两个指针 TopOfAOI 和 BottomOfAOI,以定义 AOI 的相应范围.这两个指针在动态实体初始化时即可确定.

(2) AOI 指针的更新.

对静止的动态实体而言,其 AOI 是不动的,因此 TopOfAOI 和 BottomOfAOI 指针的内容固定不变.当动态实体运动时,其 AOI 大小不变,位置和区域发生变化,因此 AOI 指针的内容也不断变化.我们可以根据实体运动的连续性及桶表中的表项按键值排序的特点,在每个仿真循环中按照实体运动的各种属性,动态地刷新 AOI 指针

的内容.

设 LOS 两个顶点分别为 StartPoint 和 EndPoint,改进后的 SO*算法如下:

算法 2. SO*算法.

- (1) 根据实体的物理特性初始化实体的 AOI 及相应的指针.
- (2) If 仿真结束
GOTO 7.
- (3) 根据实体的运动规律刷新 AOI 指针.
- (4) If LOS 的终点在 AOI 范围之外
GOTO 2.
- (5) SO 算法.
- (6) GOTO 2
- (7) End.

2 算法的比较与分析

在文献[5]中,将 SO 算法与几种 CGA 系统中已有的算法进行了比较,根据一个小范围的地形数据进行了相交性检测及判断面和时间的统计,统计数据表明,该算法在时间效率上优于已有的算法.本文将 SO 算法与改进算法 SO*进行了比较与分析.以下假设在虚拟环境中有 N 个三角形,由于每个三角形在轴上投影线段有两个端点,所以每个桶表最多有 $2 \times N$ 个表项(假设各个三角形投影线段的端点彼此不同).

2.1 空间复杂度及内存使用效率

本文的方法以 SO 算法为基础,故采用了 SO 算法所用的各种结构.为了在桶表中表示感兴趣区,在每个桶表附加了两个指针,用于指示当前实体的 AOI.从空间占用的角度 go 看,二者差别不大,但利用 AOI,可以设计一个桶表的调度算法,提高内存使用效率.如前所述,桶表具有较多的表项,预处理阶段生成 X, Y, Z 三个方向的 3 个静态桶表,为运行阶段的查询做准备,如果这 3 个表的所有表项都在内存中,内存无疑会耗费很大.因此我们可基于 AOI,将包含 AOI 指针的一定范围的表调入内存,当 AOI 指针超出该范围时,再调入新的桶表表项,这无疑会极大地提高内存的使用效率.

2.2 算法的时间效率

如前所述,对某个桶表的检索时间 T 由 3 部分组成,即搜索桶表中首个检索项的检索时间 T_1 、确定桶表中其他检索项的时间 T_2 和单线索二叉树的检索时间 T_3 .

$$T = T_1 + T_2 + T_3.$$

我们的改进是针对搜索桶表中的首检索项进行的,因此,我们主要对 T_1 进行分析.

搜索桶表中的首检索项是一个标准的二分法检索过程,设平均检索长度为 L ,则 $L = \lg(2 \times N)$,所以 T_1 具有 $O(\lg(n))$ 级的时间复杂度^[7].

设实体的 AOI 在该方向覆盖的三角形为 M ,AOI 包含最多 $2 \times M$ 个桶表表项,因此,新的平均检索长度 L' 为

$$L' = \lg(2 \times M).$$

新的检索时间 T_1' 也具有 $O(\lg(n))$ 级的时间复杂度.

从形式上看,二者的时间复杂度都在 $O(\lg(n))$ 级,但对于大规模的虚拟环境来说, N 与 M 通常具有较大的数量级差异($M \ll N$),因此检索长度差异较大.设 $R = N/M (R > 1)$,则

$$L' = \lg(2 \times M) = \lg(2 \times N / R) = \lg(2 \times N) - \lg(R) = L - \lg(R),$$

故平均检索长度的差值 $\Delta L = L - L' = \lg(R)$.

平均检索长度的差值与 N/M 的关系如图 2 所示.

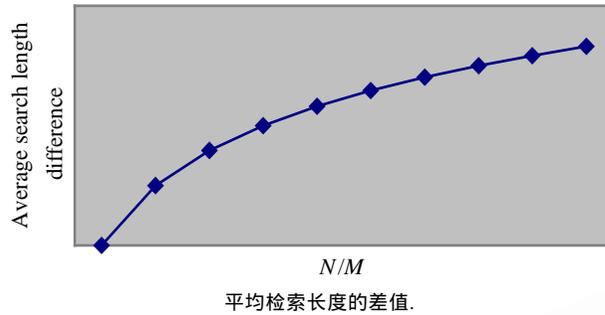


Fig.2 The relationship between average search length difference ΔL and N/M

图 2 平均检索长度的差值 ΔL 与 N/M 的关系

对于大规模虚拟环境中的应用, ΔL 的减少使缩短检索时间变得很重要. 例如, 在 DVENET (distributed virtual environment networking)^[8] 中, 我们基于真实地形数据构建了区域为 220 公里 \times 300 公里的三维地形, 多边形数量达到百万级以上, 而对 CGA 而言, 其 AOI 仅是其中很小的一部分 (对模拟环境中的动态实体的视觉系统而言, 仅有 1 公里 \times 1 公里), 二者的多边形数量级具有很大的差异, 这时, 时间的改善就很可观了. 因此, 我们通过有效地缩小搜索范围, 可以尽快确定两点的通视性, 而且对于虚拟环境中的实体而言, 基于 LOS 的通视性检查只是整个仿真循环处理的一部分. 通过其输入, 可为进一步的分析工作提供基础. 例如在 CGA 系统中, 每帧都要进行通视性检查, 以获取必要的环境信息, 为其行动提供参考依据, 通过缩减通视性检查的判断时间, 可以将更多的时间用于行为的判断与选择. 因此, 基于 AOI 的 SO^{*} 算法对提高通视性检查实时性及行为的真实性都有着重要的意义.

2.3 实例

我们在 DVENET 中的陆地行进 CGA 中进行了实验. 陆地行进 CGA 包括全局路径规划和实时避障两个模块. 在全局路局规划部分, 根据地形数据库和起终点的信息规划一条由起点到终点的航迹^[9]. 在实时避障部分, 根据不同的实体类型进行不同的避障处理, 对于动态实体采用发布信息 and 计算的方式获得, 对于点状的静态实体 (房屋、树木等), 采用 LOS 的方法进行障碍的检测和避障计算工作. 如前所述, CGA 在每个仿真循环中还要进行其他实体和本身的状态维护工作, 因此对实时性的要求很高. 结果表明, 采用本文提出的方法以后, 能够满足 CGA 在分布式虚拟环境下实时性的要求, 陆地行进 CGA 能够较为实时地进行障碍的检测和避障计算等工作.

3 结论及今后的研究工作

本文对虚拟环境中的点到点通视性检查算法进行了研究与分析. 我们在对目前在通用性和检索效率方面取得了较好实验结果的 Sieve Overlap 算法进行分析的基础上对其进行了改进, 使之更适合大规模分布式虚拟环境的需要. 改进的算法主要基于感兴趣区 (AOI) 的思想, 大幅度缩减了桶表的检索范围, 在仿真程序的仿真循环中根据物体的运动特性动态刷新 AOI 指针, 既利用了 Sieve Overlap 算法的优点, 又极大地缩减了桶表的检索空间, 减少了其检索的时间开销, 从而为 CGA 或人在回路的仿真器较快地提供各种信息, 为其进一步的各种推理判断提供依据. 今后将在具体的仿真应用中对该算法进行检验和完善, 特别是在根据 AOI 指针进行桶表的动态调度以提高内存利用效率方面. 另外, Sieve Overlap 算法和改进算法都是基于静态情况下的静态实体进行的, 对于环境的动态变化没有考虑. 由于虚拟环境在运行过程中会发生动态的变化 (如毁伤等), 不可能预先将这些变化静态构建到桶表中, 因此需要研究将这些动态的变化信息实时加入到桶表中并进行一致性的维护的方法, 使得算法更适合真实的动态实时变化的虚拟环境.

References:

- [1] Wang, Cheng-wei, Gao, Wen, Wang, Xing-ren. The Theory, Implementation and Application of Virtual Reality Technology. Beijing: Tsinghua University Press, 1996 (in Chinese).
- [2] Intervisibility. <http://www.cs.ucf.edu/m5/compgeom.html>.

- [3] Stytz, M.R. Distributed virtual environment. IEEE Computer Graphics and Applications, 1996,16(3):19~31.
- [4] Rourke, J.O. Computational Geometry in C. New York: Cambridge University Press, 1994.
- [5] Petty, M.D, Mukherjee, A. The sieve overlap algorithm for intervisibility determination. In: Proceedings of the 1997 Spring Simulation Interoperability Workshop. Orlando FL, 1997. <http://siso/ist.ucf.edu/siw/97Spring/>.
- [6] IEEE p1516 Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)——Interface Specification. Draft 1 1, Version 1.3. 1998. <http://hla.dmsa.mil>.
- [7] Kruse, R.L, Tondo, C.L, Leung, B.P. Data Structure & Programming Design in C. New Jersey: Prentice-Hall International, Inc., 1991.
- [8] Zhao, Qin-ping. Distributed Virtual War Environment DVENET. Beijing: Science Press, 2001 (in Chinese).
- [9] Liang, Xiao-hui, Wu, Wei, Zhao, Qin-ping. Research of global path planning method based on generic algorithm and large scale real terrain data. Journal of Computer Research and Development, 2002,39(3):301~306 (in Chinese).

附中文参考文献:

- [1] 汪成为,高文,王行仁.灵境(虚拟现实)技术的理论、实现及运用.北京:清华大学出版社,1996.
- [2] 赵沁平.分布式虚拟战场环境 DVENET.北京:科学出版社,2001.
- [3] 梁晓辉,吴威,赵沁平.大规模真实地形数据中的全局路径规划算法研究——基于遗传算法的实现.计算机研究与发展,2002,39(3):301~306.

An Improved Intervisibility Testing Algorithm*

LIANG Xiao-hui, ZHAO Qin-ping

(Department of Computer Science and Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100083, China)

E-mail: lxh@vrlab.buaa.edu.cn

<http://www.buaa.edu.cn>

Abstract: The intervisibility testing algorithm in the virtual environment that based on the LOS (line of sight) is researched on. After analyzing, the SO (sieve overlap) algorithm, which is more universal and efficient than other point to point intervisibility testing algorithm, is improved to fit more for the requirements of large-scale virtual environment and simulation. SO algorithm uses Bucket List and Singly Threaded Ternary Tree to manage and search the terrain polygon. The searching scale of Bucket List is reduced based on the AOI (area of interesting). This makes the point to point intervisibility testing algorithm more real and efficient and gives the environmental information to the dynamic entity in the large scale virtual environment more efficiently. The space and time efficiency of the algorithm SO* and the SO is also compared.

Key words: virtual environment; intervisibility testing algorithm; point to point intervisibility testing; sieve overlap algorithm; dynamic entity; area of interesting

* Received February 25, 2002; accepted June 11, 2002

Supported by the National High-Tech Research and Development Plan of China under Grant No.2001AA115130