

基于部件可视锥的复杂目标遮挡剔除方法

张叶廷¹ 朱庆¹

(1 武汉大学测绘遥感信息工程国家重点实验室, 武汉市珞喻路 129 号, 430079)

摘要:针对结构错综复杂的目标实时绘制中的遮挡剔除问题,提出了部件可视锥概念,用一个圆锥区域描述部件在考虑其邻近范围其他部件联合遮挡情况下的可视区域,并提出了基于部件可视锥的遮挡剔除算法,实现了实时绘制过程中可见性的快速判断,并以一个典型的虚拟建筑环境为例测试了本文方法的有效性。实验证明,利用可视锥能简化绘制过程中的可见性识别,大大提高实时绘制效率。

关键词:部件可视锥; 遮挡剔除; 复杂目标; 实时绘制

中图分类号: P208

随着数据采集手段及数据精度的提高,三维数据越来越精细,在提高目标逼真建模细节程度的同时,也大大加剧了场景的复杂性。如何在现有的硬件基础上高效、逼真地表现复杂三维场景是三维可视化研究的一个前沿性热点问题^[1]。影响三维场景绘制速度的一个重要因素是场景对象的可见性识别。遮挡剔除因此是最直接有效的加速复杂场景绘制的方法,其基本目标是检测由于被别的物体所遮挡而从当前视点看不到的一些物体。遮挡剔除对较大的深度复杂性的场景特别有效,因为在这种场景中可以完成大量的遮挡剔除。这方面的研究已经有 30 多年的历史^[2]。遮挡剔除算法可以被概括性地分为以下 4 类:① 基于图像精度的算法^[3-6]。② 基于对象精度的算法^[7-11]。③ 基于区域的算法^[2,12-15]。④ 基于硬件可视性查询的算法^[16-18]。

对于包含复杂目标的场景如建筑环境来说,由于建筑物目标中自身内部的大量相互遮挡关系,严重影响整体的绘制效率,而此情况不能通过常规的可视性判断如视域剔除等有效改善。为此,本文根据复杂目标的几何结构及数据组织等特点,提出了部件可视锥概念,通过预先计算复杂目标中的组成单元——部件的可视锥,可以在绘制时快速判断部件是否可视,从而大大减少不必要的绘制数据量。

1 部件可视锥概念

在复杂目标中,很多部件互相遮挡,而且部件间的遮挡一般为不完全遮挡,因此用不含遮挡融合的方法无法有效剔除。而单独对每个部件使用遮挡剔除算法的成本又太高,几乎和绘制部件自身的成本相等,起不到加速的作用。如果对每个部件中的多边形进行背面剔除,虽然理论上也可以剔除一半的多边形,但由于此方法无法与批量绘制一起使用,从而必须频繁切换绘制状态,反而会导致效率下降。

如果将空间中能够看见部件的方向事先计算并保存,在实时绘制时通过查询当前视点是否位于可视的方向上,就可以快速判断部件对于当前视点的可视性。本文采用空间采样的方式,在以模型几何中心为球心、模型外接包围盒对角线长度为半径的球面上均匀布点。以这些球面上的点作为视点,部件几何中心为目标点,来计算此部件的可视情况,并根据计算结果将球面上的采样点标记为可视点或不可视点。这些由可视点组成的区域就可以作为部件可视区域的近似表达,当视点位于部件中心到某个可视点的矢量附近时,此部件就可以被认为可视。

在实际应用中,这些用离散点来表达的部件

可视区域不便于保存和使用,因此,为了减少可视区域的存储量,并简化使用方式以及提高效率,本文提出了部件可视锥的概念。

部件可视锥是一个能够将部件上所有点到其可视点的矢量全部包括的最小圆锥区域。通过将该部件的可视区域用圆锥区域来表示,部件的可视性判断就可以通过简单的矢量夹角计算快速得到,有效提高了存储的方便性和使用效率。

2 部件可视锥的计算与应用

2.1 计算部件可视锥的算法

根据复杂目标中各个部件受其他部件联合遮挡较为显著的特点,使用基于视点空间采样的遮挡可视性判断的方法,对从模型外部观察部件时的可视区域进行计算。具体算法如下。

1) 在以整个模型的几何中心为球心、模型包围盒的对角线长度为半径的球面上均匀采点作为视点位置集合,并循环执行步骤2)~步骤4),直到处理完所有视点。

2) 顺序取出视点位置集合中一点设置为摄像机位置,此点到当前部件几何中心的矢量为摄像机方向,绘制整个建筑模型。

3) 打开硬件遮挡查询设置,单独绘制当前计算的部件。

4) 获取遮挡查询结果,将是否可视的标志设置给此视点位置。

通过上述算法可以计算出部件所有可视点集合。此集合可以作为部件可视区域的离散化表示。本文对可视区域中所有的点求出一个最小的外接圆锥,使之能将所有可视点容纳在内。这样,在实际应用中,只需要计算视点是否在此圆锥内部就可以快速判别出此部件的可视性,而以上判断只需要一个夹角计算就可以完成。

为了计算的正确性和方便性,本文为每个部件计算的是一个保守的可视圆锥,即对于所计算的可视圆锥必须包含所有可视区域。实际上,计算此可视圆锥也等价于不可视区域的最大内嵌圆锥,当不可视区域有多片时,只保留各片区域最大内嵌圆锥中的最大值。

由于三维中透视投影的关系,上述可视区域计算算法所得的结果只能代表视点在当前采样点位置的可视性,当采样点半径变化时,可视采样点的分布也会随之变化,因此不能直接用其作为可视圆锥的计算依据。要计算可视区域,需要借助矢量来计算。在图1(a)和图1(b)中,实线圆弧

表示可视采样点的区域,而虚线圆弧表示不可视采样点的区域。一个采样点可视意味着从采样点位置面向部件中心观看时,可以看到部件实体上的至少一个点。连接所有可以被看到的点到其对应的可视采样点的矢量集合构成了此部件的可视区域。本文的保守最小圆锥应该包含所有这些矢量。显然,决定此圆锥参数的应该是经过可视区域边界且处于可视区域最外侧的矢量集合。在图1中, $a_0 a_2$ 和 $a_0 a_3$ 就是这样的矢量。对这些矢量集合求出最小包围圆锥,就可以得到理想的结果。但由于本文的可视采样点是通过硬件遮挡查询思想计算出来的,虽然知道在 a_2 和 a_3 处可以看见部件,但无法获取 $a_0 a_2$ 和 $a_0 a_3$ 这样的边缘矢量上的其他点坐标,因此这些矢量无法准确得到。这里采用适当放大可视区域并计算最值的思路来实现此目标,具体思路如下:

1) 为部件A计算出最小包围球 o_2 ,圆心为 o_2 ,半径为 r_2 。找出所有可视采样点中的边缘点。对每个边缘点 a_2 ,执行步骤2)~步骤3)。

2) 过 a_2 、采样点所在球面对应的球心 o_1 和 o_2 三点作平面,在此平面内,经过 a_2 的直线与球 o_2 相切于两点 k_1 和 k_2 。

3) 在 a_2 周围的可视采样点中寻找一个非边缘的点 a_4 。如果 $\angle k_1 a_2 a_4$ 小于 $\angle k_2 a_2 a_4$,则选择 $k_1 a_2$ 为经过采样点 a_2 处的最边缘的矢量,反之则选择 $k_2 a_2$ 。

4) 对所有非边缘的可视采样点计算并保存 o_2 到该点的矢量。

5) 保留可视采样点中的边缘点标志,使用本文稍后所述的最小包围圆锥算法计算其可视矢量集合的最小包围圆锥的轴线方向 N 和夹角 V 。

6) 最小包围圆锥顶点 S 应在经过 o_2 、方向为 N 的矢量上,且 S 到 o_2 的距离为 $r_2/\sin V$ 。

根据每个部件的可视采样点所保存的单位向量构成的集合,本文根据如下算法计算其可视锥,并将其用三个变量来表达,即轴线方向 N ,起点 S 和最小夹角 V ,步骤如下。

1) 将其所有可视点坐标看作向量单位化,则整个视点集合处于以坐标原点 o 为球心的单位球的球面上。

2) 计算所有可视点(用来表示单位向量的点坐标,下同)的最小包围球,如果求得的球心 O' 与坐标原点 O 不重合,则可视区域的最小包围圆锥小于半球,原球心到新球心的矢量 OO' 就是轴线方向 N (图2(a)),跳转到步骤6);如果求得的球心和坐标原点 O 重合,则可视锥大于半球,此时需要按照

步骤 3)~步骤 5)计算轴线方向(图 2(b))。

3) 在所有边缘可视点集中,遍历任意三个点的组合,对每个组合执行步骤 4)。

4) 计算出通过此三点的平面 P ,判断是否所有其他可视点都与坐标原点 O 同在 P 的一侧。如果不存在与坐标原点 O 处于 P 异侧的可视点,则 P 可以确定为是圆锥与所有点包围球的相交平面的备选项。计算出 O 在 P 上的投影 O' ,保存线段 $O'O$ 的长度。

5) 在所有符合条件的相交平面备选集中,找出其 $O'O$ 长度最短的平面,即是顶点为 O 的最小包围圆锥与单位球相交得到的圆环所在平面,矢

量 $O'O$ 即为轴线方向 N 。

6) 求出轴线方向 N 后,计算所有可视点代表的矢量与 N 的夹角,取其中最大值即为圆锥的夹角 V 。

轴线方向 N 、起始点 S 以及最小夹角 V 确定了可视锥范围,此可视锥是实际可视区域的一个子集,但比可视区域更容易计算可视性。在场景漫游过程中,计算每个部件可视锥的轴线方向和视线(从视点到起始点 S)之间的夹角。如果此夹角小于最小夹角 V ,可以快速判别此部件为可视,反之则为不可视。

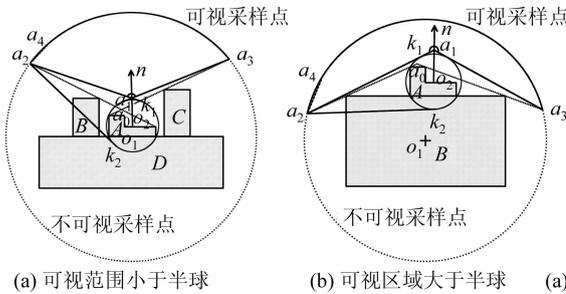


图 1 部件可视锥计算示意图

Fig. 1 Diagram of Computing the Component Visible Cone

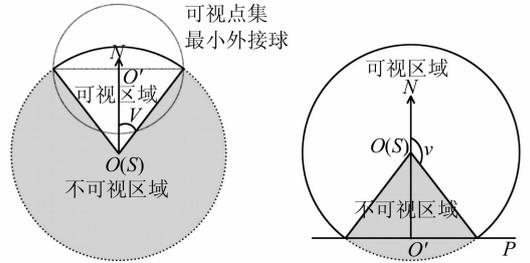


图 2 两种类型最小圆锥区域计算示意图

Fig. 2 Calculation Diagram for Two Types of Minimum Cone Region

2.2 部件可视锥的应用

复杂目标中的部件可视锥并不能直接应用于所有的情况,当视点在模型内部或在模型附近时,由于模型内部遮挡关系不起作用,无法保证可视锥的有效性,因此不能用可视锥进行判断。

在采样球外部,理想圆锥的可视区域将完全被保守圆锥的可视区域所包括,因此所有的视点都可以用保守圆锥进行判断。

由于采样球内部范围被直接判别为可视区域,采样球半径的取值将影响剔除的效果,对其半径取值过大会引起可视区域的无谓扩大。在不考虑球面点采样间隔因素的影响时,本文所介绍的算法与采样球半径的取值无关。只是由于在进行硬件遮挡检测时如果模型太靠近视点,会被剪裁切面裁掉部分面片导致画面破碎而影响检测效果,因此需要在模型最小包围球半径的基础上适当增加一定的长度作为采样球半径。当然,由于本算法初始可视区域数据是建立在采样的基础上的,所以不可避免会有些误差,通过加密采样球可以得到改善。由于出现错判的情况几乎都在可视锥的边缘,而且错判情况下部件的可视像素都很少,对画面质量的影响微弱。

3 实验分析

为验证算法的有效性,本文选取了部件结构的中国古典建筑三维模型进行实验。模型由带有榫卯的部件模型构成,包含了单体部件 1 348 个,三角形数量达到 63 907 个。首先对模型中的所有部件计算了部件可视锥,并在模型周围均匀分布了 100 个视点测试可视锥的剔除效率和正确性。在计算各个部件的可视锥时,使用 36×36 个离散点进行采样,即相邻两个采样点间夹角大约为 10° 。

其中,对一个长条瓦部件的计算结果如图 3 所示,A 标注的长条模型为被计算的瓦部件,黑色点为可以看见的采样点。从采样点的分布来看,不可视区域点主要分布在相对于瓦部件的模型后方,由于瓦的坡度不大,因此模型的顶部区域主要是可视点,这些与视觉判断相符,部件可视锥的计算结果也与视觉判断一致。

在 100 个视点位置分别应用硬件遮挡剔除和本文算法进行比较。对剔除的测试结果如表 1 所示,总共绘制的模型部件数量为 134 800 个,其中实际可视的部件为 66 539 个,所以最理想的情况

下剔除效率可以达到 50.6%，使用硬件遮挡剔除算法可以剔除的个数为 42 740，剔除率为 31.7%，而使用本文介绍的可视锥时剔除率可以达到 41.8%。

表 1 对每个部件使用可视锥后的剔除效果

Tab.1 Culling Results of Applying Visible Cone to Each Component

	使用硬件 遮挡剔除	使用可视 锥剔除	理想剔除	直接绘制
剔除部件个数	42 740	56 407	68 261	—
剔除比例/%	31.7	41.8	50.6	—
绘制时间/s	5.34	1.58	—	2.15

在上述数据中,理想剔除数据是先绘制完整个模型的所有部件后,使用硬件遮挡查询功能对每个部件查询其可视状态所得到的结果。根据上述数据可以看出,直接应用硬件遮挡剔除算法并不能达到理想剔除率:① 因为其使用包围盒作为遮挡查询的绘制图元,有些部件本身不可视但其包围盒却可视;② 因为各个部件严重受其他部件的联合遮挡,如果一个部件本来不可视,但能够遮挡它的其他部件并没有全部在它之前被绘制,则在查询时,此部件会被认为是可视。与之相反,由于可视锥构建时就使用了理想剔除的结果作为原始数据,因此可以达到近似硬件遮挡剔除的剔除率。

从实际使用中的时间开销上来讲,用可视锥

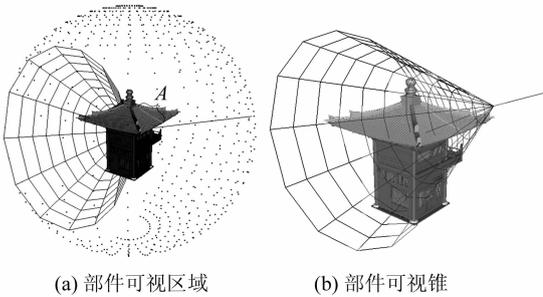


图 3 可视锥计算结果
Fig.3 Computing Result
for Visible Cone

进行剔除的判断开销仅为几个乘法运算,而硬件遮挡查询的效率则依赖硬件。由于其涉及到组织数据、发送到显卡进行查询绘制以及从显卡获取查询结果等几个阶段,因此其时间即使是对于高档显卡也是一个不可忽略的开销。需要指出的是,部件可视锥技术虽然高效,但也会有少量的错误判断,例如图 4 中的两个位置出现了本来可视的部件被错误剔除掉的现象。

造成如图 4 所示错误的原因是采样点个数有限,并不能覆盖所有视点区域,因此有时在可视锥的边缘区域会有不确定的情况。值得说明的是,这与本文中的保守可视锥算法并不矛盾,因为保守可视锥是对于测得的采样点的可视性来说是保守的,而上图的情况是因为可视区域边缘的采样点不能代替其所对应区域内的其他点的可视性。由于这些不正确的情况处于可视锥的边缘区域,因此错误一般发生在视点与部件轴线方向夹角较大时,此种情况对整体的画面质量影响很小,从上图中的两个实例也可以得出此结论。

以上测试是直接使用可视锥的夹角作为判断依据,在此情况下的错误率为 3.3%,其中大部分被错判的部件实际可视像素少于 10 个。当将判断的夹角标准放大 10°时,错误率下降到 0.1%,而剔除率仍然可以达到 30.9%。这里选择 10°的依据是建立可视锥时两个采样点与球心之间连线的夹角大约为 10°。

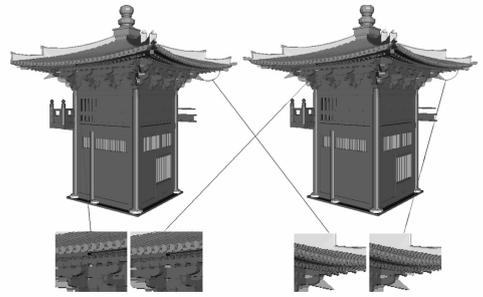


图 4 正确场景(左)与使用部件可
视锥剔除后的场景(右)对比
Fig.4 Contrast of Correct Scene (left) and the Scene
after Component Visible Cone Culling (right)

参 考 文 献

[1] 张叶廷. 三维城市模型自适应可视化方法[D]. 武汉:武汉大学,2008
[2] Cohen-Or D, Chrysanthou Y L. A Survey of Visibility for Walkthrough Applications[J]. Visualization and Computer Graphics, 2003, 9(3): 412-431

[3] Greene N, Kass M, Miller G. Hierarchical Z-buffer visibility[C]. New York: ACM, 1993
[4] Zhang H, Manocha D, Hudson T, et al. Visibility Culling Using Hierarchical Occlusion Maps [C]. SIGGRAPH'97, Los Angeles, 1997
[5] 任重, 华炜, 鲍虎军, 等. 全局遮挡图[J]. 计算机学报, 2005, 28(06): 972-978
[6] Mao Z, Shi J. TOIGP: A New Hierarchical Depth

- Occlusion[C]. Association for Computing Machinery, Hong Kong, 2006
- [7] Teller S J. Visibility Computations in Densely Occluded Polyhedral Environments[D]. CA:UC Berkeley, 1992
- [8] Coorg S, Teller S. Real-time Occlusion Culling for Models with Large Occluders[C]. ACM, NY, 1997
- [9] Hudson T, Manocha D, Cohen J, et al. Accelerated Occlusion Culling Using Shadow Frusta[C]. ACM, New York, 1997
- [10] Xiong Hua, Liu Zhen, Qin Aihong, et al. Predictive Occlusion Culling for Interactive Rendering of Large Complex Virtual Scene[C]. VSMM, Xi'an, 2006
- [11] Tsuji T, Hongbin Z, Hasegawa T, et al. Hierarchical Face Cluster Partitioning of Polygonal Surfaces and High-speed Rendering[J]. Systems and Computers in Japan. 2007, 38(8): 32-43
- [12] Durand F E D, Drettakis G, Thollot J E L, et al. Conservative Visibility Preprocessing Using Extended Projections[C]. ACM, NY, 2000
- [13] Schaufler G, Dorsey J, Decoret X, et al. Conservative Volumetric Visibility with Occluder Fusion[C]. ACM, NY, 2000
- [14] Koldas G, Isler V, Lau R W H. Six Degrees of Freedom Incremental Occlusion Horizon Culling Method for Urban Environments[C]. ISVC Lake Tahoe, NV, 2007
- [15] Yilmaz T, Gudukbay U. Conservative Occlusion Culling for Urban Visualization Using a Slice-wise Data Structure[J]. Graphical Models, 2007, 69(3/4): 191-210
- [16] Mattausch O, Bittner J, Wimmer M. CHC⁺⁺: Coherent Hierarchical Culling Revisited[J]. Computer Graphics Forum, 2008, 27(2): 221-230
- [17] Bo L, Cheng W, Li Bo, Wang Cheng, Li Lijun. Efficient Occlusion Culling with Occupancy Proportion[C]. CSSE, Wuhan, 2008
- [18] Bittner J, Wimmer M, Piringer H, et al. Coherent Hierarchical Culling: Hardware Occlusion Queries Made Useful[J]. Computer Graphics Forum, 2004, 23(3): 615-624
- [19] 朱庆,李逢春,张叶廷. 一种复杂轮廓线的真三维 TIN 算法[J]. 武汉大学学报·信息科学版,2007, 32(3): 259-262
- [20] 许伟平,朱庆,张叶廷. 采用胶囊体进行三维城市模型的实时碰撞检测[J]. 武汉大学学报·信息科学版,2009,34(9):1 030-1 033

第一作者简介:张叶廷,博士,讲师,主要从事虚拟地理环境理论、技术方法和应用研究。
E-mail:zhangyeting@263.net

Occlusion Culling Method for Complex Objects Based on Component Visible Cone

ZHANG Yeting¹ ZHU Qing¹

(1 State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing,
Wuhan University, 129 Luoyu Road, Wuhan 430079, China)

Abstract: Aiming at the critical occlusion culling issues of complicated objects for real-time rendering, the concept of component visible cone is defined to describe the visible region when the occlusion fusion effected by other components are considered, which can be used for fast visibility test. In this paper the occlusion culling algorithm based on the component visible cone is proposed, and a typical virtual architecture environment is employed as a case for experimental analysis, the results prove that the visibility test during the rendering is simplified dramatically by means of the precomputed visible cone, the efficiency of real-time rendering is therefore improved remarkably.

Key words: component visible cone; occlusion culling; complex objects; real-time rendering