

面向移动终端的三维模型简化与碰撞检测方法研究

沈 瑛 王 辉 王立晖 吴青青

(浙江工业大学计算机科学与技术学院 杭州 310023)

摘 要 移动终端三维场景的绘制与漫游由于其庞大的模型数据量和复杂的外观形态使得实现清晰的场景快速绘制十分困难。为了加快模型的绘制,提出了一种面向移动终端的三维模型的简化与碰撞检测方法,以优化绘制过程。该方法通过二次测量误差半边折叠算法来简化三维模型,并利用八叉树技术对不能显示在屏幕中的场景进行剔除,从而实现三维场景的快速读取、组织和绘制。针对移动设备的屏幕尺寸以及计算能力等限制,实现了适用于移动平台的碰撞检测算法,减少了计算量。实验结果表明,该方法能有效地简化模型,并提高绘制效率,同时减少碰撞检测的计算时间,因而可应用于三维场景的快速逼真绘制。

关键词 三维场景,快速绘制,场景组织,碰撞检测

中图分类号 TP391 **文献标识码** A

Simplifying 3D Models and Collision Detection on Smartphones

SHEN Ying WANG Hui WANG Li-hui WU Qing-qing

(College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract The enormous model data and complex morphology of the 3D rendering and roaming on mobile platform make it very difficult to render the three-dimension scene clearly and quickly. In order to speed up the model rendering, an optimized rendering algorithm was proposed. 3D models were simplified by quadric error metrics half edge collapse. And octree was used to achieve read, organize and fast draw 3D models by culling the scene rapidly which is not necessary in rendering. As the limitations of screen size and computing power on mobile devices, a collision detection algorithm for mobile platforms was put forward. Experimental results show that this method can effectively simplify models and reduce the amount of computation. Thus, it can be applied to realistic fast rendering of 3D scene.

Keywords Three-dimensional scene, Fast rendering, Scenes organization, Collision detection

三维场景是虚拟现实中的重要元素。虚拟环境沉浸感和逼真度是衡量虚拟现实技术的重要因素。由于三维模型结构复杂且种类繁多,三维场景的快速绘制一直都是图形学的重要研究点。为了进一步增强三维场景模型给用户带来的视觉体验,相关应用通常将现有的绘制技术与阴影、光照绘画手段相结合,构造出具有动态光影的视觉效果。但同时也需要消耗大量的系统内存资源和较长的渲染时间。尽管移动终端的硬件以及三维图形加速技术飞速发展,但依旧不能满足绘制复杂 3D 场景的需求^[1-4]。

针对移动设备的计算能力、屏幕大小和网络宽带等限制,为了加快三维模型的绘制与提高网络的传输效率,对模型对象进行了 QEM 半边折叠简化,并在轮廓边上进行了针对性优化。同时,设计和实现了适用于移动平台的基于面向对象八叉树的场景管理和碰撞检测算法。在此基础上,设计和开发了一个基于 Android 移动平台的三维绘制与漫游原型系统,完成了以浙江工业大学校园为实例的应用。

1 面向移动平台的三维场景绘制与漫游过程

截至目前,国内外的许多学者都在移动平台上进行了三

维可视化的尝试,并且取得了较多的成果。赫尔辛基工业大学的 Nurminen^[5]开发了 m-LOMA 系统。该系统重新组织了 VRML 模型,并采用了预裁剪、实时裁剪、LOD 和纹理管理等优化方法,很大程度上提高了渲染效率。但由于该系统采用的是轻量级模型,场景中没有加入地形,导致系统的细节表现力不足,真实性欠佳。冯立男^[6]在 M3G 图形库的基础上完成了 3D 虚拟城市场景,创新性地提出了保留模式和立即模式相结合的建模思想。该方法针对移动设备计算能力的限制,降低了计算复杂度;同时在绘制效果和绘制时间上做出折衷,以满足低内存占用量的需求。官杰杰^[7]在 Android 平台上利用 OpenGL ES 技术实现了一个三维图形渲染引擎。但是这两个系统都没有进行三维绘制的加速,系统运行效率较低。

因此,如何高效且实时地进行有效的 3D 场景管理,简化模型几何数据量,加快三维模型绘制效率^[8],增强场景真实性等都是当前虚拟现实技术的关键点^[9]。在模型简化方面,主要以边折叠^[10]方式为主。二次误差测量(Quadric Error Metrics, QEM)^[11]由于算法简单、运算速度快、简化质量好,成为了解决三维模型对象存储容量和实时绘制的有效性方法。场

本文受国家自然科学基金资助项目(61202202)资助。

沈 瑛(1976—),女,副教授,硕士生导师,主要研究方向为信息安全、图形图像处理与可视化等,E-mail:shenyinying@zjut.edu.cn;王 辉(1990—),男,硕士生,主要研究方向为虚拟现实,E-mail:dream_whui@163.com。

景管理最常用的方式是 BSP 树和八叉树,但两者的使用范围并不相同。BSP 树应用于场景大小有限的室内场景;八叉树主要适用于较为空旷的室外场地。碰撞检测^[11]是提高用户真实体验感的有效方法,可以避免场景的穿透现象。经典的碰撞检测方法有八叉树分割等空间分解法,以及基于层次包围盒技术^[12-13]。

CPU 和 GPU 等设备的发展,以及计算机图形学技术的不断进步,使得三维场景的绘制与漫游有了一定的软硬件基础^[14]。在进行三维场景漫游时,涉及到了“视点”的概念。“视点”类似于“人眼”,是观察物体的起始点。随着视点位置的不同,物体在屏幕上的画面也截然不同。漫游的本质是不断地转移视点,改变或转换视线方向的过程。基于移动终端的三维场景绘制与漫游功能主要分为模型文件读取、场景管理、场景渲染等。模型文件读取是指系统从场景数据库中将 3D 模型文件加载进内存的过程。模型文件主要以 OBJ 和 3DS 文件格式为主。场景管理的主要工作是将场景内的物体划分到设计好的数据结构中,并进行高效的排序与遍历,进而可以快速、高效地实现场景的可见性剪裁。场景渲染的实质是图像绘制,是对三维场景模型的绘制工作。在视点移动的过程中,不断修改位于屏幕内的可见物体,并呈现出物体的整体视觉效果。在三维场景绘制与漫游过程中主要涉及到的技术点如图 1 所示。

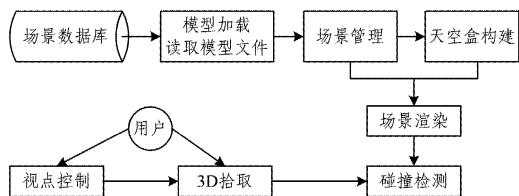


图 1 三维场景的绘制与漫游

2 二次误差测量的边折叠模型简化

渲染是三维场景绘制中最耗时的操作。对于一个复杂的大规模场景,由于规模庞大、架构复杂等因素,模型的存储、传输以及显示都需要耗费大量的资源。加之当前网络宽带和移动终端硬件配置的限制,不能同时将所有三维模型的图元一次性地送入 OpenGL ES 2.0 图形渲染管线进行渲染。因而整体渲染资源受限,严重影响了三维模型在移动平台绘制的实时性。事实上,在很多情况下(如 3D 漫游),对远处的物体只需生成较为粗糙的模型即可满足视觉效果,无需精确绘制。因此,为了加快模型的渲染效率,需要对三维模型进行简化,通过计算方式将原本复杂的模型用一个拥有更少网格面片数或顶点数的粗糙网格来代替,以降低三维模型的几何复杂度。

2.1 边折叠算法

1996 年, Hoppe^[10] 率先提出使用边折叠方法对三维模型进行简化。该方法的主要思想是将一条边折叠成一个顶点,以此来减少边及其所连接的所有三角形。这种方法具有可逆性,即可通过顶点分裂的方式将其还原成初始边及其相应的三角形。

折叠方法主要分类两类:半边折叠和边折叠。前者在折叠的过程中,将折叠后的顶点选在该条边的两个顶点上。后者则是通过计算得出新的顶点位置。相对于半边折叠而言,全折叠的顶点位置具有更强的灵活性,简化效果更逼真。但由于它会产生新的顶点,因此需要更多的系统资源,不利于三

维场景模型的网络渐进传输和网格重建。因此根据三维模型绘制与漫游系统的实际需求,采用半边折叠算法,如图 2 所示。

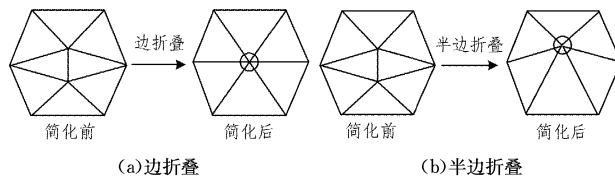


图 2 边折叠与半边折叠操作

2.2 二次误差测量

Hoppe^[10] 提出的边折叠方法中利用能量优化选择边折叠的次序以及新顶点的位置。算法要求建立并求解复杂的全局能量优化方程。该过程计算复杂,难以满足实时性需求。Garland 和 Heckbert^[11] 提出的二次误差测量(Quadric Error Metrics, QEM)算法利用二次误差来控制简化过程,很好地解决了以上问题。该算法将折叠后产生的新顶点到与该条边到相关的所有平面距离的平方和作为误差度量,能够对三维模型进行任意程度上的简化,是一种经典的网格简化算法。

QEM 算法的简化过程为:定义顶点 $v = [v_x, v_y, v_z, 1]^T$, 其平面方程为 $ax + by + cz + d = 0$, 且 $a^2 + b^2 + c^2 = 1$ 。若 P 是包含顶点 v 的所在平面,设 $p = [a, b, c, d]^T$ 。根据定义,顶点 v 的二次误差 $\Delta(v)$ 为该点与其相关面片 $S(v)$ 的距离平方和,因而可以记作:

$$\Delta(v) = \sum_{p \in S(v)} (v^T p)(p^T v) = \sum_{p \in S(v)} v^T (pp^T) v = v^T \left[\sum_{p \in S(v)} K_p \right] v \quad (1)$$

其中, K_p 用于计算空间中任意一点到平面 P 的距离的平方,可定义为:

$$K_p = pp^T = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix} \quad (2)$$

记 Q 为顶点 v 的二次误差矩阵,表示为:

$$Q = \sum_{p \in S(v)} K_p \quad (3)$$

设顶点 v 的初始二次误差 $\Delta(v)$ 为 0。在边折叠的过程中,删除边的两个端点 v_i 和 v_j ,产生新的顶点 \bar{v} 。因而新顶点 \bar{v} 的二次误差矩阵可记为:

$$Q' = Q_i + Q_j = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ q_{14} & q_{24} & q_{34} & q_{44} \end{bmatrix} \quad (4)$$

一般而言,新顶点的位置可以通过简单或优化两种方式选出。

1) 简单方式。从边折叠的端点或中点选取,即从 v_i, v_j 和 $(v_i + v_j)/2$ 中选出。这种方法比较简单,但不能保证生成新顶点的二次误差最小。

2) 优化方式。从二次误差即 $\Delta(\bar{v})$ 局部最小值处选取。局部最小值可以通过计算 $\partial \Delta / \partial x = \partial \Delta / \partial y = \partial \Delta / \partial z = 0$ 求解,因此新顶点 \bar{v} 的最佳位置和折叠二次误差分别可以表示为:

$$\bar{v} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (5)$$

$$\Delta(\bar{v}) = \bar{v}^T(Q_i + Q_j)\bar{v} \quad (6)$$

2.3 改进二次误差测量

从上述二次误差测量算法中可以看出:二次误差测量在简化过程中仅仅考虑了距离度量。这将导致简化后的网格呈现出均匀分布的特性,因此拐点、折痕以及尖锐边等重要的几何细节容易丢失。当二次误差测量直接应用于明显的轮廓特征物体(如机械、建筑等三维模型)时,其效果往往不够理想。因此,在二次误差测量算法的基础上提出了一种基于轮廓边的半边折叠算法:将轮廓边进行针对性的简化,尽量保证三维模型的外部形状特征。具体的简化策略为:

1) 轮廓边简化。由于二次误差测量中是在两端点折叠代价 v_1 较小处进行新顶点的选取,因此可以将此值乘以大于 1 的权重值 W_a ,以扩大折叠代价。新的折叠代价记作 v_2 。其中, $v_2 = v_1 W_a$ 且 $v_2 > v_1$ 。在边折叠时,将 v_2 作为代价值重新进行代价排序。在此基础上选择出折叠代价较小的顶点作为新顶点,以此来尽量避免轮廓曲线被折叠。

2) 其他边简化。利用二次误差测量算法用折叠后的局部最小值替代折叠代价进行处理。

图 3(b)、图 3(c)均为简化后顶点为 180 的模型简化结果。其中,图 3(b)采用 QEM 简化策略,图 3(c)采用改进后的 QEM 简化算法。从图 3 中的简化效果来看:两者的简化效果总体相似,但图 3(c)的简化结果能更好地保持住模型的三维轮廓特征,验证了优化算法的有效性。

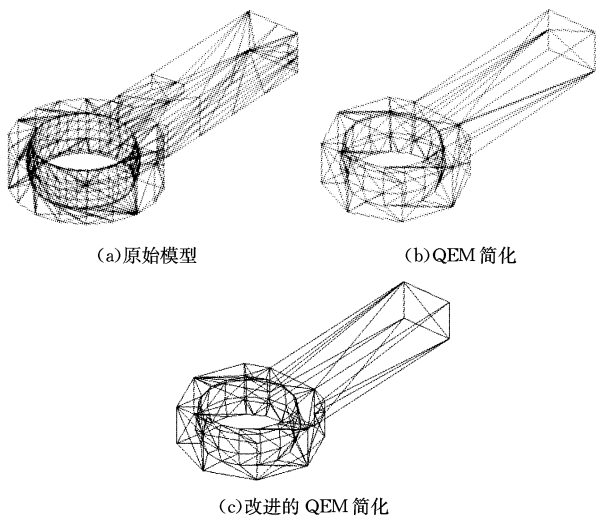


图 3 模型简化结果

3 面向移动平台的碰撞检测方法研究

事实上,在三维漫游系统中,用户看到的始终是整个场景的一部分,绝大部分场景在视锥体之外或是被其他物体遮挡,不需要进行绘制^[15]。因此,在渲染之前需要进行可见性裁剪,先判断哪些图元在可见区域内,然后只将这些位于可见区域内的图元送入图形流水线进行渲染,从而避免不必要的光栅化计算,提高绘制效率。我们考虑采用场景层次划分方法,将场景对象用包围盒进行替代,再按照其空间关系层层归并,完整场景对象的空间管理工作。由于要实现移动平台上的三维场景绘制,场景规模不是很大,而且对场景切换速度要求不是很高,因此可以考虑进行八叉树划分来实现对三维场景的可见性裁剪。

3.1 面向对象八叉树

八叉树是一种空间划分技术。将三维模型按照其所处的

空间位置划分到八叉树层次结构中。通过八叉树划分技术,可以快速地判断哪些物体位于视锥体之外,然后将这些视锥体外的物体剔除,减少送入绘制流水线的图元,进而提升系统的渲染效率。

简单来说,八叉树是将立方体划分为 8 个子立方体,以此递归。然而,传统的八叉树算法中所有的模型三角面均被看成一个整体,并没有面向模型对象的概念,所有的三角面片都相互独立、毫无区别。因此无法包含整个模型对象的几何、物理等特征,场景交互需求也无法得到满足。

针对以上问题,集成了面向对象与八叉树,构建一种面向对象八叉树:将带有 AABB 包围盒的模型对象作为八叉树的结点,而不再是三角面。这种面向对象八叉树既保存了 3D 场景中的几何、物理数据,也可以利用八叉树自身的性质方便地实现场景更新,从而对场景进行有效的组织和管理。八叉树一般有两种存储方式:规则方式和线性方式。

规则方式:包含指向父结点和 8 个子结点的指针,以及指示结点类型的字段。这种方式可以比较快速找到某个指定的结点。

线性方式:先用预定的方法(深度优先、广度优先等)遍历八叉树,将其转换成一个线性结构。根据对场景的不同操作,线性结构可以选择存储所有结点或只存储叶结点。这种存储方式的灵活性比较差,为了找到某一结点,必须从根结点顺次遍历线性表。但这种存储方式适合某些特定操作。

通过结合八叉树的这两种存储方式,根据不同的场景需求选择不同的存储方式来进行操作。具体划分的过程如下:

- 1) 将整个 3D 场景视作根结点,对整个场景建立场景对象包围盒。
- 2) 如果根结点可以进行细分,则将包围盒均匀划分为 8 个子长方体,计算每个子长方体的范围。
- 3) 根据物体空间位置,将其存储到对应的子结点中。
- 4) 根据设定的树高度条件,终止或者继续子结点的创建。
- 5) 删除父结点中存储的物体信息,因为这些信息已经重复包含在每个子结点中。
- 6) 如果子结点的面片数量少于预定的阈值或者子结点达到了预定的深度,则停止划分;否则,对每个子结点继续上述步骤。

3.2 面向对象八叉树的碰撞检测方法

针对移动平台计算能力不足的限制,充分考虑了系统真实性和实时性的要求,设计了适用于移动平台的基于面向对象八叉树的碰撞检测算法。该算法主要针对传统的碰撞检测算法^[9]做了如下修改。

1) 减少碰撞检测次数

在建立面向对象八叉树后,将包围盒视作碰撞检测对象。虽然包围盒紧密型差,但可以快速确定场景中两个物体的包围盒是否有可能发生碰撞,而无需进行物体间的面片相交情况,该过程称作粗略碰撞检测。由于包围盒仅仅用于粗略的碰撞检测,精确度不需要非常高,因此我们选择构建和相交判断都比较迅速的 AABB 作为包围盒,以降低检测算法的复杂性。

2) 增强碰撞检测精度





目前大多研究将包围盒的碰撞视为实际三维物体之间的碰撞。但严格来说,由于包围盒的封闭性差,空间范围会大于实际三维物体,因此包围盒的碰撞与实际物体之间的碰撞不能等同。因此,在八叉树空间划分的基础上,增加精确碰撞

检测过程,判断观察者与其对应的空间结点中的每个面片是否相交。

3) 观察者抽象

如果采用长方体来表示观察者,在粗略检测过程中需要比较两个立方体的位置关系,也就是针对一个点是否在另一个立方体内计算 8 次。用视点作为观察者在场景中的化身,只需计算视点和包围盒的位置关系,就能判断观察者和物体是否发生了碰撞。因此观察者抽象能简化观察者与三维模型的相交判断过程,从而达到降低碰撞检测复杂度的目的。其中,5 种主要的包围盒^[11]及其特性如表 1 所列。

表 1 5 种包围盒的方法对比

包围盒	图示	紧密性	检测速度	检测精度
包围球		差	快	差
AABB		较差	快	差
OBB		中等	较快	中等
K-DOP		较好	较快	较好

在充分考虑移动平台的计算能力的基础上,通过基于面向对象八叉树的方法对传统碰撞检测进行了优化。这种方法优于传统碰撞检测,原因在于通过使用包含空间信息的数据结构,降低了离观察者较远而完全不可能出现碰撞的物体对碰撞检测效率的影响,使得更多的计算资源可以被用于提高碰撞检测精度。

在以上策略的基础上,考虑到漫游系统中虚拟人物视点高度的因素,可以对八叉树的绘制进行再一次的精简处理,减少八叉树中结点的数量,从而降低碰撞检测的次数。

对于场景中的建筑物来说,观察者不可能和建筑物中高于观察者的部分发生碰撞,因此在划分包围盒时,可以只对下部进行划分。设虚拟人物的视点高度为 H_p ,建筑、树木和花坛等虚拟物体的高度为 H_o ,则八叉树结点的实际高度为 $H_r = \max\{H_p, H_o\}$ 。在八叉树的构建过程中,由于八叉树的高度被限制在虚拟空间的下部,需要规定三维空间物体与八叉树的包含规则。以 H_r 作为空间长方体的高度,将建筑物划分为若干个子空间;遍历所有的面片,如果一个面片与空间长方体在 3 个坐标轴上的投影上都有交集或重叠,则将这个面片划分到这个子空间中,一个面片可能会被划分到多个子空间中。

如图 4 所示, $\triangle ABC$ 与子空间 1、子空间 2 在 X 轴和 Z 轴投影上有交集,在 Y 轴投影上重合,因此 $\triangle ABC$ 被划分到子空间 1 和子空间 2 中。

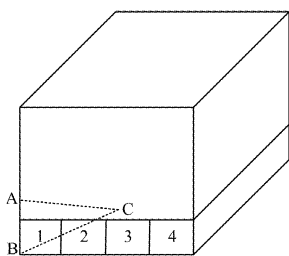


图 4 将面片划分到子空间

在碰撞检测阶段,通过计算观察者相对于包围盒的偏移,可以快速确定与观察者发生碰撞的子空间。之后再观察者依次与该子空间内的所有面片进行精确检测,从而判别物体间的碰撞情况。

精确检测就是计算当前视点与下一帧视点的连线和面片是否有交点,这种方法的本质是判断空间中一条线段与一个三角形是否有交点,如图 5 所示。一般的计算过程为:设平面方程为 $Ax + By + Cz + D = 0$,通过计算该平面的法向量,可以得到系数 A, B, C 的值,进而求解参数 D 。将当前视点 $O(x_1, y_1, z_1)$ 和下一帧视点 $P(x_2, y_2, z_2)$ 的坐标值代入平面方程。如果如下不等式:

$$(Ax_1 + By_1 + Cz_1 + D)(Ax_2 + By_2 + Cz_2 + D) \leq 0$$

成立,则表明 O 和 P 在平面两侧或平面上,线段与平面相交。

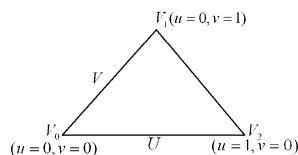


图 5 三角面交点示意图

考虑到移动平台的性能限制,参考 Ller 和 Trumbore^[16]提出的快速相交判断方法,改进了相交判断过程。设三角形上任意一点可以表示为:

$$T(u, v) = (1 - u - v)V_0 + uV_1 + vV_2 \tag{7}$$

其中, V_0, V_1, V_2 分别为三角形的 3 个顶点; u, v 和 $1 - u - v$ 分别为 V_1, V_2 和 V_0 的权重,且满足 $u \geq 0, v \geq 0, u + v \leq 1$ 。线段的射线表示形式为:

$$R(t) = O + tD \tag{8}$$

令 $R(t) = T(u, v)$,求射线与三角形的交点,可得:

$$\begin{bmatrix} t \\ u \\ v \end{bmatrix} = \begin{bmatrix} -D \\ V_1 - V_0 \\ V_2 - V_0 \end{bmatrix}^{-1} (O - V_0) \tag{9}$$

然后根据 Cramer 法则就可以分别求出 t, u, v ,即:

$$\begin{bmatrix} t \\ u \\ v \end{bmatrix} = \frac{1}{(D \times E_2) \cdot E_1} \begin{bmatrix} (T \times E_1) \cdot E_2 \\ (D \times E_2) \cdot T \\ (T \times E_1) \cdot D \end{bmatrix} \tag{10}$$

其中, $E_1 = V_1 - V_0, E_2 = V_2 - V_0, T = O - V_0$ 。更进一步,为了减少移动端计算量,考虑在计算的过程中通过数值排除完全不可能的情况:

- 1) 设 $t = (D \times E_2) \cdot E_1$,若 t 接近 0 则表示射线在三角平面上,实际意义可以理解为运动物体的当前帧和下一帧移动距离极小。在这种情况下可以认为两个物体极限靠近,但还没有相交。
- 2) 若 $(D \times E_2) \cdot T \notin (0, t)$,从实际意义理解,表示射线在一个方向上的分量超过了和向量,故不会相交。
- 3) 若 $(T \times E_1) \cdot D < 0$ 或 $(D \times E_2) \cdot T + (T \times E_1) \cdot D \notin (0, t)$,从实际意义理解,若射线和三角平面有交点,则必须满足 $u \geq 0, v \geq 0, u + v \leq 1$,故不相交。以上判断过程是在计算过程中逐步完成的。

4 实验

移动端三维场景绘制与漫游系统可以对任何符合要求的三维模型进行读取与绘制,可应用的场合有很多,如对其增加游戏的物理引擎即可实现 3D 游戏,也可将校园的模型导入

即可实现校园的漫游。以浙江工业大学为例设计实现校园漫游系统,从而验证三维显示系统的可用性。校园漫游系统的主要功能有 GPS 定位到模型所在位置、三维场景模型显示、与三维场景中事物进行交互(如点击教学楼显示教学楼名称等)。

4.1 三维场景模型绘制与漫游

将三维场景内的物体划分到设计好的八叉树数据结构中,因此可以进行高效的空間排序与遍历。在对场景进行可见性剪裁后,三维场景就显示在系统界面中,主要包括楼房、树、草坪和天空等几部分。在进入场景后,用户可利用左右或上下滑动操作来进行左右转动或前进后退等视角变化,以便查看周围建筑物的情况。在视角变化的同时,三维场景会立即刷新并绘制。其中长按操作利用 3D 拾取技术^[17]可查看具体详细的楼房名称信息,具体情况如图 6 所示。

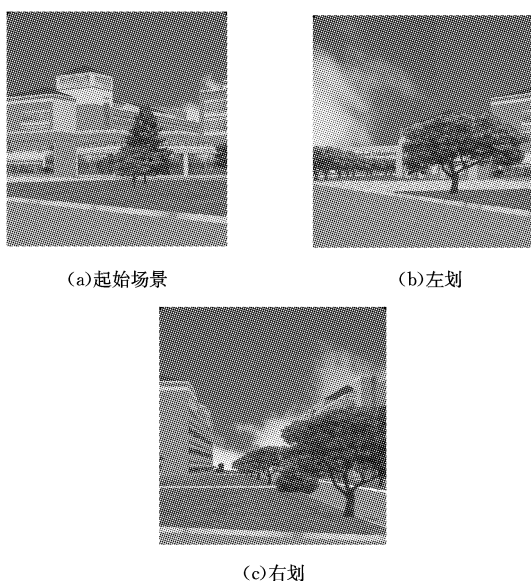


图 6 场景左划、右划视野图

4.2 三维模型简化

以图书馆模型为例,为了更好地展示效果,采用线框图绘制模型,如图 7 所示。图 7(a)为原始模型;图 7(b)为恢复了部分细节信息的网格,保留了原始模型大约 50%的数据;图 7(c)保留了原始网格大约 25%的数据。从表 2 中可以看出:模型简化算法能对网格进行简化,模型简化至 25%时,用时仅为 356.13 ms。

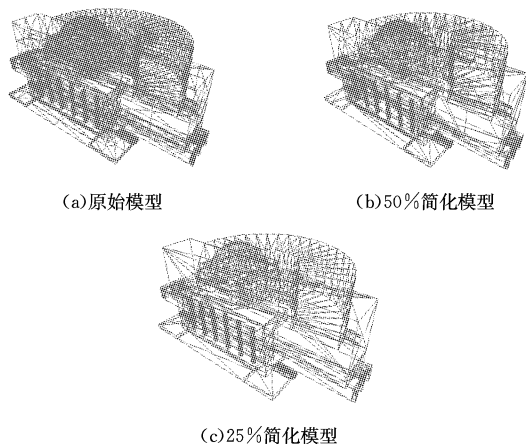


图 7 模型简化效果展示图

表 2 三维模型简化

模型	三角形数/个	简化用时/ms
原始模型	20568	—
50%简化模型	10334	198.37
25%简化模型	5218	356.13

4.3 碰撞检测结果分析

如图 8 所示,在漫游过程中,当遇到障碍物(墙)时用户无法穿透和跨越,而由于此时视点与障碍物相接触,障碍物会被放大,加上像素的限制,图像会呈现模糊颗粒状态。由此可见,该碰撞检测算法能够正确地判断出物体之间是否有接触情况。



图 8 碰撞检测效果图

通过捕获碰撞检测线程,在智能手机端对不同面片数模型进行碰撞检测,分别获得两种碰撞检测相交计算算法的检测时间(单位:ms),具体结果如表 3 所列。

表 3 碰撞检测时间列表/ms

检测方法	模型面片数/个				
	1000	2000	5000	10000	20000
改进前检测时间	7.863	14.234	36.027	89.865	214.679
改进后检测时间	7.125	13.863	34.058	85.962	208.040

结束语 提出了利用面向对象八叉树来实现场景管理,并进行可见性剪裁,以减少图形绘制时的系统资源。使用 AABB 包围盒技术将不可视区域进行快速剔除,加快场景绘制时的渲染速度。同时设计了基于移动平台的碰撞检测算法,加快了观察者与三维模型间的碰撞效率。另外,从上述实验中可以发现,本文提出的 QEM 半边折叠改进方法虽然保持了原始模型的总体效果,但依然无法保持模型的部分尖锐和细节部分。在以后的工作中可以进一步考虑将纹理坐标等综合信息纳入考虑范围,对简化算法进行优化。

参 考 文 献

- [1] TANG M, MANOCHA D, TONG R. Fast continuous collision detection using deforming non-penetration filters[C]//Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. Washington DC, USA, 2010:7-13.
- [2] 唐勇,杨偲偲,吕梦雅,等.自适应椭球包围盒改进进物碰撞检测方法[J].计算机辅助设计与图形学学报,2013,25(10):1589-1596.
- [3] 宋城虎,闵林,朱琳,等.基于包围盒和空间分解的碰撞检测算法[J].计算机技术与发展,2014,24(1):57-60.
- [4] 王立晖.基于移动平台的三维绘制与漫游系统[D].杭州:浙江工业大学,2013.
- [5] NURMINEN A. m-LOMA-a mobile 3D city map[C]//Proceeding of the Eleventh International Conference on 3d Web Technology. New York, USA: ACM, 2006:7-18.

- [6] 冯立男, 戚正伟. 嵌入式三维场景漫游系统的设计与实现[J]. 计算机工程, 2009, 35(18): 257-259.
- [7] 官酷杰. 基于 OpenGL ES 的移动平台图形渲染引擎研究与实现[D]. 北京: 北京交通大学, 2010.
- [8] 徐彩虹, 刘志, 潘翔, 等. 一种基于实例学习的三维模型检索匹配方法[J]. 浙江工业大学学报, 2012, 40(3): 326-330.
- [9] 吕书明, 张明磊, 孙树立. 基于简化和细分技术的三角形网格拓扑优化方法[J]. 计算机辅助设计与图形学学报, 2014, 26(8): 1225-1231.
- [10] HOPPE H. Progressive meshes[C]//Conference on Computer Graphics and Interactive Techniques. New York, USA: ACM, 1996: 99-108.
- [11] GARLAND M. Surface simplification using quadric error metrics[C]//Conference on Computer Graphics & Interactive Techniques. New York, USA: ACM, 1997: 209-216.
- [12] 于海军, 马纯永, 张涛, 等. 基于图像空间的快速碰撞检测算法[J]. 计算机应用, 2013, 33(2): 530-533.
- [13] 何伟, 李勇, 苏虎. 碰撞检测中的包围盒方法[J]. 重庆工学院学报(自然科学版), 2007, 21(23): 148-152.
- [14] 徐峰, 陈敏智. 虚拟校园三维仿真系统的设计及实现[J]. 浙江工业大学学报, 2007, 35(2): 155-158.
- [15] 卢威, 曾定浩, 潘金贵. 支持外观属性保持的三维网格模型简化[J]. 软件学报, 2009, 20(3): 713-723.
- [16] LLER T, TRUMBORE B. Fast, minimum storage ray/triangle intersection[C]//ACM SIGGRAPH. New York, USA: ACM, 2005: 21-28.
- [17] 夏怒. 三维场景中的景物拾取技术[J]. 电脑知识与技术: 学术交流, 2008, 4(1): 196, 199.

(上接第 224 页)

结束语 本文着重研究了基于多特征通道的行人检测算法, 深入分析了算法原理, 并且加入了基于 DCT 变换后的特征, 通过级联方向梯度直方图特征、颜色空间特征和基于 DCT 的特征通道对数据集进行训练和检测, 行人检测的精度明显提高。实验结果表明: 加入了基于 DCT 变换后的特征通道, 在误检率为 10^{-1} 时, 漏检率下降了 2% 以上, 且平均漏检率也优于已知的传统算法。目前所提算法还有进一步优化的空间, 可以通过减轻运算量降低复杂度, 缩短检测时间。

参 考 文 献

- [1] IKEUCHI K. Computer Vision: A Reference Guide[M]. Springer Publishing Company, Incorporated, 2014.
- [2] 苏松志, 李绍滋, 陈淑媛, 等. 行人检测技术综述[J]. 电子学报, 2012, 40(4): 814-820.
- [3] CAO J, PANG Y, LI X. Pedestrian Detection Inspired by Appearance Constancy and Shape Symmetry[J]. IEEE Transactions on Image Processing a Publication of the IEEE Signal Processing Society, 2016, 25(12): 5538.
- [4] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet Classification with Deep Convolutional Neural Networks[J]. Advances in Neural Information Processing Systems, 2012, 25(2): 2012.
- [5] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation[C]//CVPR'14. 2014: 580-587.
- [6] REDMON J, DIVVALA S, GIRSHICK R, et al. You Only Look Once: Unified, Real-Time Object Detection[C]//CVPR'16. 2016: 779-788.
- [7] DALAL N, TRIGGS B. Histograms of oriented gradients for human detection[C]//IEEE Conference on Computer Vision & Pattern Recognition. 2005: 886-893.
- [8] ZHANG S, BAUCKHAGE C, CREMERS A B. Informed Haar-Like Features Improve Pedestrian Detection[C]//IEEE Conference on Computer Vision and Pattern Recognition. 2014: 947-954.
- [9] WANG X G. Deep learning in image recognition[J]. Communications of the CCF, 2015, 11(8): 15-23.
- [10] DOLLÁR P, TU Z, PERONA P, et al. Integral Channel Features[C]//British Machine Vision Conference, BMVC 2009. London, UK, 2009.
- [11] ZHANG S, BENESON R, OMRAN M, et al. How Far are We from Solving Pedestrian Detection? [C]//CVPR'16. 2016: 1259-1267.
- [12] DOLLAR P, APPEL R, BELONGIE S, et al. Fast Feature Pyramids for Object Detection[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 36(8): 1532-1545.
- [13] FARINELLA G M, RAVI D, et al. Representing scenes for real-time context classification on mobile devices[J]. Pattern Recognition, 2015, 48(4): 1086-1100.
- [14] WALLACE G K. The JPEG still picture compression standard[J]. Communications of the ACM, 1991, 34(4): 30-44.
- [15] STURGES P, ALAHARI K, LADICKY L, et al. Combining Appearance and Structure from Motion Features for Road Scene Understanding[C]//British Machine Vision Conference, BMVC 2009. London, UK, 2009.
- [16] BATTIATO S, MANCUSO M, BOSCO A, et al. Psychovisual and Statistical Optimization of Quantization Tables for DCT Compression Engines[C]//International Conference on Image Analysis and Processing. IEEE Computer Society, 2001: 602-606.
- [17] PETERSON H A, PENG H, MORGAN J H, et al. Quantization of color image components in the DCT domain[C]//Proceedings of SPIE-The International Society for Optical Engineering. 1991: 210-222.
- [18] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2232.