

OSG 与 Bullet 的结合研究与实现

贾庆轩, 廖裕宁

(北京邮电大学自动化学院, 北京 100876)

摘要:图形引擎和物理引擎在虚拟现实的仿真模拟和游戏开发等领域扮演着越来越重要的角色。为了使虚拟场景的仿真更加逼真,需要将图形渲染引擎与物理引擎相融合。鉴于此,在充分了解开源图形引擎 OSG 和开源物理引擎 Bullet 的基础上,分析两者各自的特点,并研究如何有效地将两者相互绑定。在研究结果的指导下,构建出一个具有物理特性的虚拟环境。实例的仿真结果验证二者的结合在虚拟仿真中的实用性,这为建立真实的仿真环境提供了一种可行的方案。

关键词:图形引擎;物理引擎;碰撞检测;融合

中图分类号:TP391.9 **文献标识码:**B

Research and Implementation on Binding of Open Scene Graph Engine and Bullet Engine

JIA Qing-xuan, LIAO Yu-ning

(College of Automation, Beijing University of Posts and Telecommunications, Beijing 100876, China)

ABSTRACT: Graphic and physics engines play more and more important roles in areas of virtual reality simulation and the design of computer games. In order to display the virtual reality scene more effectively, we need to amalgamate the graphic rendering engine and physics engine. It described Open Graphic Engine (Open Scene Graph) and Open physics engines (Bullet) separately, analysed their advantages, and then researched the way of binding between them in this paper. At last, we constructed a virtual reality that contains physical property according to the study. The results suggest that the binding is reliable, which provides a scheme for simulation.

KEYWORDS: Graphic engine; Physics engine; Collision detection; Binding

1 引言

随着计算机技术的发展,虚拟现实技术也逐渐地走进人们的生活。虚拟现实技术的兴起,特别是图形引擎的应用和发展,大大加速了各项游戏、应用的开发,使人们能够随时随地的分享各种各样的游戏应用带来的便利与乐趣。图形渲染引擎拥有管理与渲染虚拟场景,处理光照和材质,模拟粒子系统与自然等特效的强大功能^[1-3]。

但是,当场景需要实现较为复杂的碰撞、滚动和弹跳时,图形引擎是不可能完成这个任务的。要完美展现物体的物理特性,必须将图形引擎与物理引擎相融合,借助物理引擎利用物理运动定律,实时计算每帧画面物体的位姿,增强物体与场景间的交互性^[4-7]。

OSG (OpenSceneGraph) 与 Bullet 的开源的特点吸引越来越多的人对其进行研究,并且应用到开发中去。也是由于

其开源性,这两个引擎能够与时俱进,快速地将新技术容纳入自己的体系中。研究工作者不仅可以免费使用这两个引擎,还可以使自己始终站在先进技术的最前端。因此,希望能够为实现二者的结合提供一种可行的方案。还希望使用该方案建立的系统真实、准确、并且能以超过 24 帧/s 的帧率稳定运行。

2 图形引擎 OSG

OSG 是一个专为 3D 计算机图形开发而设计的高层次的可编程接口,广泛应用于虚拟仿真、动画设计以及各种可视化程序中。OSG 基于 OpenGL 进行构建,因此它同时具备了跨平台的特性和较高的渲染性能。此外,OSG 还提供了一系列可供 3D 程序开发者使用的功能接口。包括 2D 和 3D 数据文件的加载、纹理字体支持、细节层次 (LOD) 控制、多线程数据分页处理等等,而这些是 OpenGL 本身不具备的。凭借丰富的功能特性以及开放源代码的协议形式,OSG 已经广泛地被学术界和产业界所认可。目前使用 OSG 进行开发的三

基金项目:国家自然科学基金(61175080)

收稿日期:2013-10-14 修回日期:2013-11-07

维软件数以万计,而 OSG 的开发者团队早已遍及世界各地。

2.1 核心结构

OSG 运行时文件由一系列动态链接库(或共享对象)和可执行文件组成。链接库可分为以下五大类:

- 1)OSG 核心库。它提供了基本的场景图形和渲染功能,以及 3D 图形程序所需的某些特定功能实现。
- 2)NodeKits。它扩展了核心 OSG 场景图形节点类的功能,以提供高级节类型和渲染特效。
- 3)OSG 插件。其中包括了 2D 图像和 3D 模型文件的读写功能库。
- 4)互操作库。它使得 OSG 易于与其它开发环境集成,例如脚本语言 Python。
- 5)不断扩展中的程序和示例集。它提供了实用的功能函数和正确使用 OSG 的例子。

OSG 核心库提供了应用程序和 NodeKits 所需的功能模块。而 OSG 核心库和 NodeKits 一同组成了 OSG 的 API。OSG 核心库中的 osgDB 则通过对 OSG 插件的管理,为用户提供了 2D 和 3D 文件 I/O 的接口。其体系结构如图 1。



图 1 OSG 的结构体系图

2.2 OSG 的数据管理机制

OSG 中使用场景树结构来管理整个三维世界,如果要绘制一个规模宏大的场景,这势必要创建和维护很多的数据对象和节点对象。OSG 常用的数据类型有二维与多位向量、四元数、矩阵以及数组。熟悉三维建模和图像处理的人应该对这不会陌生,下面将简单阐述如何管理节点对象。

OSG 的内存管理机制能很好地管理整个场景的节点,用以维护程序运行,这就是内存引用计数策略,这种策略可以通过自动清理不再引用的内存以避免内存泄露。OSG 提供一种“垃圾收集”的机制,通过“内存引用计数”的形式来管理场景中的节点和绘制体。每一个场景中的对象都维护着一个引用计数值,当它被其它对象引用时,引用计数值加 1,反之减 1。当一个对象的引用计数值为 0 时,它会自动从内存中释放。

因此,对于一个完整的场景树结构,将树的根节点从内存中卸载的时候,它会减少子节点的引用计数,这一动作进而引发连锁的效果,将场景中的所有节点和数据逐一释放。

2.3 OSG 的场景组织结构

OSG 采用包围体层次(BVH)来实现场景图形的管理。包围体指的是将一组物体完全封闭在一个简单的空间形体中,从而提高检测的运算速度。

采用包围体层次结构场景图形通常采用树的状结构来保存信息。一棵场景树包括一个根节点、多级内部的枝节节点,以及多个末端的叶子节点。根节点和枝节节点负责构建树的层次,叶子节点则保存了一个或者多个绘制体的信息。这种场景 BVH 树不仅可以正确地表达场景图形的信息组成,还可以加速场景对象的裁剪,相交测试、碰撞检测等一系列操作。

例如一房子,房子里面有窗户、门和家具等。房子和 3 个子节点(窗、门、家具),每个子节点都包含所需的绘制体信息。这个场景可以描述如图 2 所示。

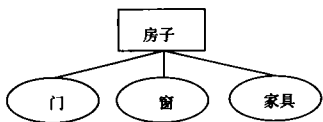


图 2 节点树结构

2.4 渲染方式

场景图形系统允许程序保存几何体并执行绘图遍历,此时所有保存于场景图形中的几何体以 OpenGL 指令的形式发送到硬件设备上。但是该执行机制无法实现前述的诸多高级特性。为了实现动态的几何体更新、拣选、排序和高效渲染,场景图形需要提供的不仅仅是简单的绘图遍历,事实上,有三种需要遍历的操作:

- 1)更新(update):更新遍历(有时也称作程序遍历)允许程序修改场景图形,以实现动态场景。更新操作由程序或者场景图形中节点对应的回调函数完成。例如,在汽车起重机驾驶模拟系统中,程序可以使用更新遍历来改变汽车起重机的位置或姿态,或者通过输入设备来实现与用户的交互。
- 2)拣选:在拣选遍历中,场景图形库检查场景里所有节点的包围体。如果一个叶节点在视口内,场景图形库将在最终的渲染列表中添加该节点的一个引用。此列表按照不透明体与透明体的方式排序,透明体还要按照深度再次排序。
- 3)绘制:在绘制遍历中(有时也称作渲染遍历),场景图形将遍历由拣选遍历过程生成的几何体列表,并调用底层 API,实现几何体的渲染。图形的显示表现为一系列帧的渲染。特别地,渲染过程与显示器的刷新速率是同步的,这是为了避免诸如图像撕裂(image tearing)等问题的出现。

3 Bullet 物理引擎

物理引擎的工作流程就是对渲染场景的每一帧进行物理模拟,再对具有物理特性的模拟实体进行状态与受力分析。然后,进行碰撞检测,找出运动实体间相互的约束信息,并通过牛顿力学原理计算出每个实体新的位移和速度,从而更新实体的方位,得到新的虚拟^[8-10]。

3.1 基本构成

仿真世界(world),包括离散仿真世界(btDiscreteDynam-

icsWorld) 或者软体仿真世界 btSoftRigidDynamicsWorld。这个仿真世界最先创建,用以存放其它所有的对象以及后续的所有计算。当在动态世界中执行帧仿真时,所有上面的阶段都会执行。默认的动态世界实现是 btDiscreteDynamicsWorld。

碰撞图形, Bullet 支持很多不同种类的碰撞图形,也可以加入用户自己碰撞图形。为了让程序拥有更好的性能和质量,根据需要选择适当的碰撞图形很重要。如果遇到复杂图形,可以用多个基本图形复合而成。多个凸图形使用 btCompoundShape 可以组合到一个合成图形或者复合图形,这是一个有凸图形的部件组成的凹图形,有叫子图形,每个 btCompoundShape 子图形相对于 btCompoundShape 都有自己的局部偏移转换方法。

约束,约束只能用在两个刚体之间,而且必须至少有一个是动态的。根据自由度不同, bullet 中的约束分为:点对点约束、铰链约束、滑动约束、锥扭约束、通用 6 自由度约束。

物体的物理特性,包括物理的材质、密度、摩擦因数等等。这里的材质不同于物体本身所具有的材质属性,而是对光是何种反射方式。密度、摩擦因数、反弹系数等特性决定了物体在仿真世界里面的仿真结果。

3.2 物理引擎的工作流程

基本处理步骤如下:

1) 物理模拟。以牛顿力学定理为基础,根据实体对象的物理属性,计算各种力对实体的作用,从而确定其运动状态。

2) 计算各个实体对象所受的合力 (Force) 和合力矩 (Torque)。合力采用向量叠加法计算,合力矩可用平行轴定理来计算。

3) 进行碰撞检测。根据实体的运动状态检测它们之间的交互情况。一般的碰撞检测分为两个检测阶段:粗略检测阶段主要是利用物体的包围盒(从 BB , OBB , Sphere), 配合一种空间划分法 (BSP, OBBTree) 来进行粗略的层次型碰撞检测。目的是快速过滤掉没有发生碰撞的物体。如果物体的包围体发碰撞,那么就进入到精细检测阶段进行更精细的判断。精细检测一般是基于三角形的碰撞检测,目的是找到确切的碰撞点坐标和碰撞法线。

4) 如果有碰撞发生,响应碰撞并进行碰撞处理。一般用冲量定理来计算刚体碰撞后的速度。

4 OSG 与 bullet 的结合

OSG 引擎功能完善,图形处理能力强大,能建立和管理丰富多彩的虚拟场景,但是场景中的物体缺乏交互。有了物理引擎的支持,虚拟场景中的物体能够按照现实世界的物理定律运动起来。物体与物体之间能够发生相互作用,无疑大大增加了真实感。

4.1 OSG 与 bullet 融合的流程

首先对在 OSG 场景中添加模型,创建场景相机、视口,添加灯光、材质等,使物体变得丰富起来。然后利用 bullet 引

擎,根据模型的三维尺寸,创建模型的碰撞几何体(网格化)。在此基础上,可以根据碰撞几何体创建与模型完全一致的刚体或者软体物体。值得注意的是,如果场景中一个模型在不同的坐标中出现,那么并不需要针对每一个坐标处的模型分别创建碰撞几何体,只需共用一个碰撞几何体,只要在创建刚体的时候设置好刚体的变换矩阵就好了。

在进行仿真运算时,首先需要创建一个 bullet 环境并进行仿真环境参数设置,之后根据重力加速度、仿真类型等场景描述创建一个场景。物理引擎中的材质是指物体的表面属性和碰撞属性,可设定反弹系数和摩擦系数。仿真开始之前,必须赋予每个对象的位姿、速度、加速度以及约束关系初值。每运算一个时间步,每个对象的各个参数将会被计算出来。这些参数将用于显示和后续的仿真计算。在对虚拟场景进行仿真计算时,各参数的设置如重力加速度、空气阻尼系数,各物体之间的反弹系数等都会采用实际的数据,以便获得更加真实的仿真效果。

准备工作做完,就可以利用 bullet 引擎对场景中的物体进行仿真计算。在 bullet 经过每一帧的计算后,场景中各个物体对象的变换矩阵也得到更新,物体的新位姿得到更新。当然,物体在图形显示上,也会更新在正确的位置和姿态。

如果物理引擎中的时间步长与图形引擎中的时间步长不同,则在仿真计算上, bullet 会通过插值法来修正运算结果,具体流程如图 3。

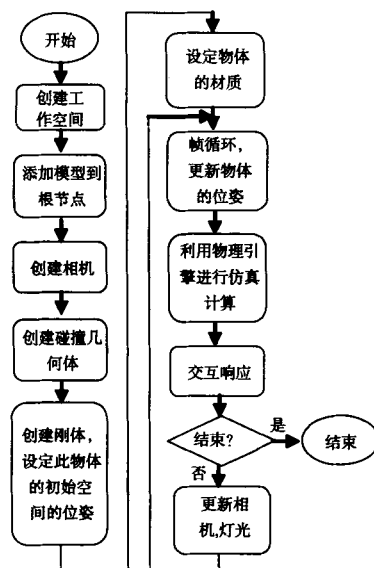


图 3 OSG 与 bullet 相融合的流程

下面一段代码实现了在一个场景中将 OSG 与 bullet 结合起来。

```
_node = (osg::Group *) osgDB::readNode File(
    strCourseName, ddsOptions); // 加载模型
m_pRoot -> addChild(_node); // 将模型节点添加到场景根节点下
```



```
m_pCourseShape = btTriMeshCollision ShapeFromOSG ( _
node,m_pCourseTriangleMesh); // 为对象_node 创建碰撞几
何体
m_collisionShapes.push_back( m_pCourseShape); // 添加
shape,用于清理
btTransform trans; // 设置将要创建的刚体的变换矩阵
trans.setIdentity();
btVector3 pos(origin.x(), origin.y(), origin.z());
trans.setOrigin( pos);
float mass(0.0); // 创建刚体的质量和转动惯量。
btVector3 localInertia(0,0,0);
m_pBody = createRigidBody( mass, trans, m_pCourse-
Shape, localInertia, false, m_pCourseMotionState ); // 创建
刚体
m_pBody -> setDamping( _lin_damping, _ang_damping);
// 添加物理材质
m_pBody -> setFriction( _friction);
m_pBody -> setRestitution( _restitution);
```

4.2 实例实现

通过上文的分析,本文将 bullet 物理引擎与 OSG 图形引
擎结合,实现了高尔夫球的模拟,如图 4 所示。



图 4 高尔夫球在球场中的飞行轨迹

整个高尔夫球场有房屋、树木、铁栏杆、群山等物体,图
中白点是高尔夫球在合力作用下的运动轨迹。整个球场中
只有高尔夫球和树叶能够移动,其它的物体都是固定的。示
例中的仿真结果有如下几点特点:

1) 系统仿真的真实性高,系统里面发生的物理事件与现
实完全一致。

高尔夫球被击打时,以一定的初速度运动。在飞行过程
中,小球受到重力加速度和空气阻力的作用,因此速率会不
断的变化。空气阻力中有一个特殊的分量,叫做马格努斯
力。如果小球的初始速率极大(比如大于 40m/s)、角速度
极大(比如大于 3000r/min),这个力会非常可观,将会对小
球的飞行轨迹产生非常大的影响。在实际的仿真中,笔者充
分考虑到这个因素,结果高尔夫球的轨迹也与实际情况吻合。

球能与球场中的房屋、树木、铁栏杆、群山,还有各种不

同的地形都能发生碰撞。而且由于这些物体的反弹系数不
完全一样,球被反弹的程度也不尽相同。当球在这些物体上
面滚动时,也会因为摩擦因数的差异而受到相应的阻力,而
从滚动的距离和轨迹也会因实际地形的差异而有所不同。

当球穿过树叶的时候,不会被反弹回去,而是获得了一个
很大的阻力,速度急速衰减。另外,球的角速度也会因为
空气阻力的缘故而慢慢衰减。如果球以一个很大的角速度
撞向摩擦系数较大的地面时,很有可能发生往回滚的现象。

2) 系统的准确性高,碰撞外形与其几何图形完全吻合。

物体的碰撞几何体是非常重要的。如果一个物体的几
何图形和其相应的碰撞外形不能匹配的话,则会出现两种情
形:一、小球在运动过程中嵌到物体里面去;二、小球在明明
没有物体的地方被反弹回来。在本示例中,笔者充分考虑
OSG 与 bullet 的坐标之间的异同,经过严密的计算,使物体
的几何图形与碰撞外形完全重合,从而保证了仿真的真实性。
如图 5 所示,场景中白色的线是模型的碰撞外形及面片的法
线。从图中可以看出,物体与其碰撞外形完全吻合。



图 5 场景物体的碰撞外形

3) 系统稳定。

系统的稳定性也非常重要。实例中的场景,面片数超过
30 万个,系统能以 60 帧/s 的速度稳定运行,如图 6 所示。



图 6 系统运行时保持稳定帧率

总之,通过 bullet 引擎的仿真,整个场景表现的栩栩如
生,场景中发生的事件如同现实一般。

5 结束语

本文对开源图形引擎 OSG 和开源物理引擎 bullet 的基本内容和框架结构进行了分析和研究,利用 OSG 和 bullet 的良好扩展性,实现了物理引擎与图形引擎的结合。本文详细地分析了整个融合流程,在此基础上实现了具有物理特性的虚拟场景模拟。

事实上,本文的实例不仅融合了图形引擎和物理引擎,还用到了一个叫 CEGUI 的界面引擎(图 4 中界面上各个图标需要 CEGUI 的支持)。加入其它模块,丰富虚拟仿真平台将是今后研究的重点和方向:

1) 加入三维声音、人工智能、网络通信等多个模块,构成一个功能完整、丰富逼真的虚拟仿真系统。

2) 结合多通道融合技术和传感器技术,将场景以 3D 立体显示的方式呈现给用户,使之有身临其境的感觉。

参考文献:

- [1] 王锐,钱学雷. OpenSceneGraph 三维引擎设计与实践[M]. 北京:清华大学出版社,2009-11:79-86.
- [2] 明芳,李峻林. 基于 OSG 的虚拟场景漫游技术研究[J]. 计算机与数字工程,2011,(3):133-137.
- [3] M Gibber, R Kaye, M Fried. Virtual Simulation in the Surgical

World[J]. Otolaryngologic Clinics of North America, 2009, 42(5):891-900.

- [4] 有人,陈定方. 虚拟现实环境中的物理模拟和物理引擎应用的研究[J]. 湖北工业大学学报,2008,23(2):7-9.
- [5] 欧阳慧琴,陈福民. 物理引擎与图形渲染引擎绑定的研究与实现[J]. 计算机工程与设计,2008,29(21):5580-5582.
- [6] 牛红攀,高勇,侯忠明. 图形引擎与物理引擎结合的研究与实现[J]. 计算机仿真,2011,28(6):299-303.
- [7] 杨俊杰,臧淑英,姜美芳. 虚拟现实中的物理引擎 ODE 与虚拟引擎 OSG 结合探讨[J]. 测绘与空间地理信息,2010,33(1):66-68.
- [8] M Joselli, et al. A new physics engine with automatic process distribution between CPU-GPU[C]. ACM, 2008:149-156.
- [9] 刘炜,陈宁. 基于碰撞检测的坦克运动与地形匹配性研究[J]. 江苏大学学报(自然科学版),2012,26(4):356-360.
- [10] 王树军. 三维游戏引擎中物理引擎关键技术的研究[D]. 天津大学,2007.

[作者简介]



贾庆轩(1964-),男(汉族),山东省菏泽市人,教授,博士生导师,主要研究领域为空间机器人和虚拟现实技术。

廖裕宁(1988-),男(汉族),广西壮族自治区梧州市人,硕士研究生,主要研究领域为虚拟现实仿真。

(上接第 229 页)

参考文献:

- [1] Sanjay Kumar. Selective Laser Sintering: A Qualitative and Objective Approach[J]. Literature, Modeling and Characterization, 2003-10:43-47.
- [2] Song Bo, Dong Shujuan, Liao Hanlin Christian Coddet. Process parameter selection for selective laser melting of Ti6Al4V based on temperature distribution simulation and experimental sintering[J]. Int J Adv Manuf Technol, 2012,61:967-974.
- [3] X C Wang, T Laoui, J. Bonse, J. P. Kruth1, B. Lauwers L. Froyen. Direct Selective Laser Sintering of Hard Metal Powders: Experimental Study and Simulation[J]. Int J Adv Manuf Technol, 2002-19:351-357.
- [4] Bai Xingwang, Zhang Haiou, Wang Guilan. Improving prediction accuracy of thermal analysis for weld-based additive manufacturing by calibrating input parameters using IR imaging[J]. Int J Adv Manuf Technol, DOI: 10.1007/s11665-010-9685-2.
- [5] Yan Ying Du, Yu Sheng Shi, and Qing Song Wei. Forming Simulation and Experimental Verification of Combined Formation of Selective Laser Sintering and Cold Isostatic Pressing[J]. Journal of Materials Engineering and Performance, 2011,20(2):185-190.

- [6] 任继文,殷金菊. 选择性激光烧结金属粉末瞬态温度场模拟[J]. 机床与液压,2012,40(1):114-117.
- [7] 任继文,刘建书. 扫描路径对激光烧结温度场的影响[J]. 机床与液压,2010,38(19):32-35.
- [8] 周文晓,王翔,崔瑞,梁建亮. 粉末薄层选区激光烧结温度场数值模拟与实验研究[J]. 现代制造工程,2012,(5):12-16.
- [9] 沈显峰,王洋,姚进,汪法根,杨家林. 直接金属选区激光烧结热应力场有限元模拟[J]. 激光技术,2005,29(4):343-346.
- [10] 李旭东. 从仿真看 3D 打印及其若干问题[C]. 第八届中国系统建模与仿真技术高层论坛论文集,2013:468-483.

[作者简介]



张龙(1983-),男(汉族),甘肃金昌人,硕士研究生,主要研究领域为计算机仿真,CAE 软件的二次开发。

李旭东(1959-),男(汉族),江苏南通人,教授,博士,博士研究生导师,主要研究计算材料学,材料的微结构计算。

郭德昌(1986-),男(汉族),甘肃兰州人,硕士研究生,主要研究领域为计算机仿真,流固热耦合分析。