

基于视景仿真的 3D 购物系统^①

朱 峰

(上海城市管理职业技术学院 人文与信息学院, 上海 200438)

摘 要: 目前大多数的购物网站均呈现千篇一律的方式, 无法像现实超市那样给消费者提供一种购物环境, 实现文化、心理上的满足和享受. 为了弥补以上缺陷, 我们考虑将目前最先进的视景仿真技术和信息管理技术融合在一起, 基于 OSG 技术构建 3D 商业环境及视景漫游、完成基于 3D 图形界面的购物操作流程.

关键词: 视景仿真; OSG; 视景漫游; 3D 场景构建; 3D 对象规划; 3D 对象姿态控制

3D Shopping System Based on Scene Simulation Technology

ZHU Feng

(Dept. of Information Technology, Shanghai Technical College of Urban Management, Shanghai 200438, China)

Abstract: Most shopping sites, represented as a routine of sameness, are dull in shopping environment, shopping in supermarket, we can be responded to social, cultural and environmental factors. In order to make up these deficiencies, Visual Environment Simulation and information management tools have been presented in this work. We build a 3D business environment and vision roaming based on OSG technology to realize the 3D graphic based shopping process.

Key words: scene simulation; open scene graph; scene roaming; 3D scene construction; 3D object layout; 3D object attitude control;

随着网络技术的不断成熟, 物流体系的日益完善, 电子商务也得到了快速的发展, 网络购物正成为一种大众化的生活方式. 目前电子商务的主要模式是 BtoB、BtoC 和 CtoC 等购物网站, 它们的表现方式是网上交易平台(中介)和网上购物. 购物网站呈现的基本元素包括购物区、购物车、收银台、商品挂牌标价、价格优惠率和商品促销等. 当你浏览过大多数的购物网站之后, 就会惊讶地发现, 他们都是一种模式, 一个面孔, 目前的网上购物虽然有许多方便之处, 但他的缺陷也是十分明显的:

①现实超市给消费者提供了一种购物环境, 客户不仅购买产品, 而且实现了一种文化的、心理的享受和满足, 目前的网络购物做不到这一点.

②购物时, 现实超市中的客户可以凭多种感觉器官挑选产品, 而目前网上超市不能产生这种多侧面的感觉. 目前电子商务的主要模式都是基于图片来展示

商品的平面及一些文字描述, 缺少现实购物中所感受的立体视觉.

1 3D购物网站的研究内容

为了弥补这一缺陷, 我们研究将目前最先进的视景仿真技术和信息管理技术融合在一起, 营造一种具有现实购物氛围的 3D 商业环境. 本文主要研究如何构建 3D 商业环境以及基于 3D 图形界面的购物操作流程, 主要研究内容包括以下几个方面:

(1)构建 3D 商业场景, 实现视景漫游. 例如构建大型商场或者超市, 用户可以使用鼠标和键盘在商场中漫游, 选择或者浏览商品, 营造一种与现实生活中逛商场相同的感受, 这是与目前的网上购物系统最本质的区别.

(2)商品的展示和更新. 要构建 3D 商场, 必须解决商品的展示模式, 例如使用模型表示商品还是使用贴

^① 收稿时间:2014-02-11;收到修改稿时间:2014-03-17

图来表示,在通常的视景仿真技术中,场景中的模型是无法被动态更新的,在 3D 商业场景中,需要很方便的更新所展示的商品信息。

(3)商品的选择.用户在商场中漫游时,需要进行购物操作也就是选择商品,如何在 2D 的屏幕环境中准确的选择 3D 场景中的物体。

(4)商品信息的检索.用户选择商品后,需要实时展示商品信息,将选择的商品与后台的商品信息文档关联并进行检索操作。

(5)基于 3D 图形界面的购物操作流程.由于是在 3D 环境中选购商品,因此购物流程与目前的网上购物有明显的区别,需要设计更贴近实际的购物操作模式。

2 系统研究方案

本系统基于 OSG(OpenSceneGraph)技术^[1-3]实现 3D 购物场景以及与购物操作相关的 3D 图形界面。OSG 是一个可移植的、高层图形工具箱,它为战斗机仿真、游戏、虚拟现实或科学可视化等高性能图形应用而设计。它提供了基于 OpenGL 的面向对象的框架,使开发者不需要实现优化低层次图形功能调用,并提供了很多附加的功能模块来加速图形应用开发。在视景开发中,很重要的一个问题是场景中大量物体的管理,也就是场景管理,OSG 提供了基于 Scene Graph 的完整的场景管理机制。OSG 是开源程序,可以自主开发程序界面,并且不受开发工具功能和版权限制,所开发的程序拥有自主版权。同时基于场景图的 OSG 技术具有很高的效率,使得我们在实现 3D 购物系统时,可以更加专注于实际功能,而非 OPENGL 底层细节操作。图 1 概括了 OSG 模型框架:

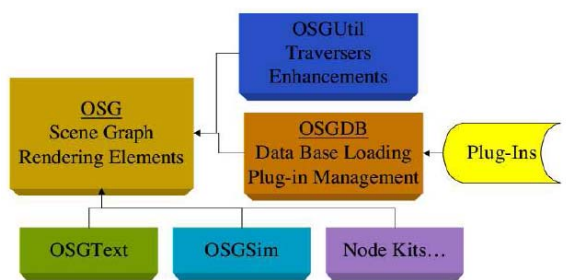


图 1 OSG 模型框架

3 基于 3DMAX 构建 3D 场景

我们利用 3DMAX 建模软件构建一个典型的 3D 商场场景。其构建流程大致如下:首先构建天空包围盒,确定整个场景的中心坐标。然后构建商场模型及其室内模型,为每个模型配置材质,同时配置场景照

明系统,渲染并烘培材质存储至纹理,最后采用广告牌技术构建商场外景。

3.1 构建天空包围盒

用包围盒来模拟远景是 3D 场景构建中的常用技术。通常采用两种包围盒:方形包围盒及球幕形包围盒。方形包围盒简单且不易产生纹理裂缝,但是立体感稍差;球幕形包围盒立体感强,但是比较难贴纹理。本系统采用球幕形包围盒。由于构建的商场模型很大,因此天空的包围球尺寸很大,而天空纹理相对较小,若将纹理布满整个球幕会导致纹理精度降低。因此,在贴纹理时设置为 U,V 方向纹理重复,这样可以提高纹理精度,但不可避免地会出现纹理裂缝。为了增强远景的真实感,我们在球幕下方设置一圆柱形包围盒,采用广角镜拍摄的环形城市远景图片作为纹理贴图,构建相对完美的模拟远景。图 2 显示了本文的天空包围盒结构。

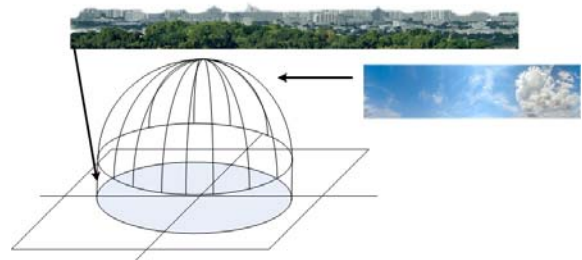


图 2 天空包围盒

3.2 构建商场模型

构建商场模型分为以下几个步骤:(1)构建商场大楼轮廓,仿照大型商场构建典型的五层商场大楼,正门带外挂玻璃楼梯。图 3(a)显示了商场大楼的基本轮廓。(2)构建室内场景,由于要放置柜台,因此,室内为放置其他对象:修饰地板、天花板、楼梯等部分,利用小相框对墙做部分装饰,同时在商场内部正面构建商场名称展示牌。图 3(b)显示了内部轮廓,(c)显示了楼梯部分细节。(3)配置照明系统,即配置室内灯光及是外日光部分。(4)对建筑的所有对象配置相应材质,例如,窗户及大门的玻璃材质、地板的木材质、相框材质、天花板材质、楼梯金属材质等。(5)加载 VRAY 渲染器,利用配置好的灯光及材质渲染场景并将结果存储至纹理,该过程也称为烘培。图 3(d)显示了室内的渲染效果。在 3D 商场购物系统中,相对于不断更新的展示商品,商场建筑本身是固定不变的,如果用渲染效果图静帧的方法去渲染,显然是太慢了,因此考虑使用

VRAY 提供的烘培技术(BAKE 3D)来加快渲染速度^[4]. 烘培贴图是目前为止最快得到最佳渲染的方法, 它的工作原理是把每一次渲染输出的模型表面贴图, 将影响光照等信息收集, 并用一张展开贴图的方式保存下来. 但是烘培贴图也有一定的弊端, 那就是如果把场景具有强烈反射的物体进行烘培, 例如不锈钢物体等, 这时候移动视角, 会发现具有反射性质的物体表面并没有发生变化. 因为 BAKE 3D 只支持贴图, 所以在视图中物体的反射也是由一张贴图来代替的. 所以在设计的商场中尽量不采用反射性质的材质, 这样才会得到更高的真实度, 当然使用 BAKE 3D 烘培贴图有时光照会有阴影计算错误, 可以将所有的模型烘培出来, 再进行细部的修改. 图 4 显示了几个主要部件的烘培贴图.

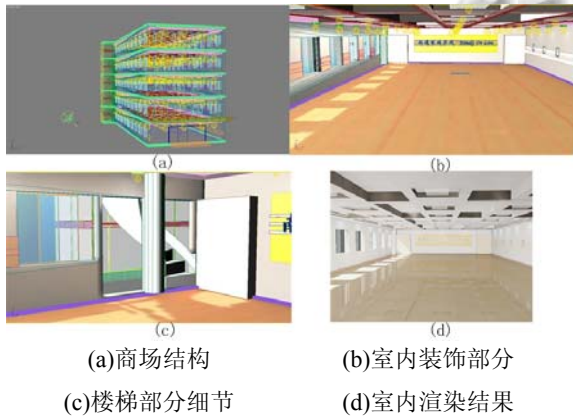


图 3 3D 商场模型构建

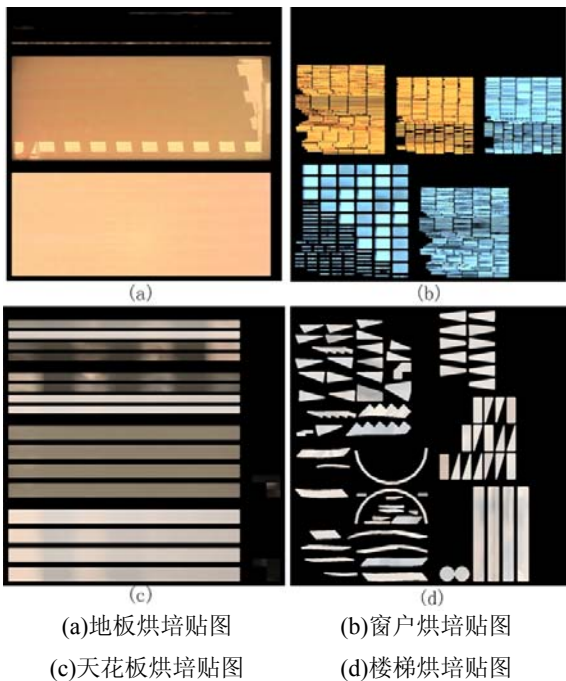


图 4 几个典型的材质烘培结果

3.3 基于广告牌技术构建外景

外景部分主要采用广告牌技术(BILLBOARD)完成^[5]. 广告牌技术是指将节点跟随用户所定义的轴或点旋转, 典型情况是广告牌跟随观察者视角围绕系统 Z 轴旋转用以模拟树木、花草等室外景物. OSG 广告牌类(OSG::BILLBOARD)包含了所有围绕用户所定义轴或点旋转的几何信息. 由于 OSG::BILLBOARD 类继承自 GEODE 类, 因此, 我们可以通过 GEODE 类中所包含的方法来控制几何体是否旋转, 以及是围绕轴还是点旋转, 并且可以定义旋转轴位置. 在 OSG::BILLBOARD 中有三种旋转模式: POINT_ROT_EYE(相对于视角围绕点旋转), POINT_ROT_WORLD(相对于世界坐标原点旋转)以及 AXIAL_ROT(围绕所定义的轴旋转, 旋转轴可以通过 setAxis(osg::Vec3)方法来定义). 在 OSG 中, 也可以通过 setNormal(osg::Vec3) 方法定义广告牌所面向的轴位置^[6,7]. 另外, 在广告牌面向视角方向选择时, 光照计算可能会出错从而导致奇怪的光照效果, 因此, 在广告牌节点中关闭了光照计算. 图 5 显示了采用广告牌技术制作的外景.

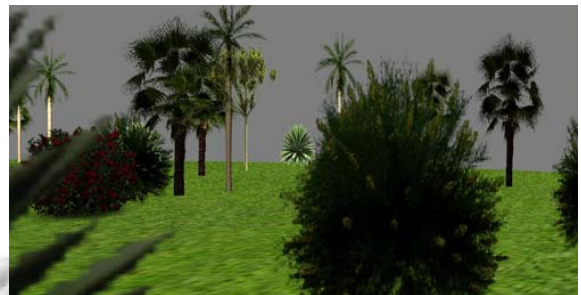


图 5 采用广告牌技术制作的外景

利用广告牌技术可以极大地简化三维物体的建模, 在图形真实感降低少许的情况下可以大幅度提高图形的渲染速度. 作为速度对比试验, 我们曾经在场景中采用有复杂树木花草的 3D 模型, 由于树木、花草的模型几何形状非常复杂, 几何数据庞大, 导致整个场景显示的运算量急速增大, 因此在场景中模型数量仅为 10 的情况下已经无法正常漫游. 采用广告牌技术后, 在外景中放入大量的树木、花草图片, 在数量大于 50 的情况下依然能正常漫游.

3.4 漫游控制系统

漫游是 3D 商场中的核心功能, 文章定义了完善的漫游控制操作, 功能包括: 漫游速度控制(增快、减

慢), 移动控制(键盘控制、前后左右), 上升下降控制(鼠标滚轮), 视角方向控制(鼠标)以及一些功能按钮, 包括回复到初始位置, 启动碰撞检测, 停止碰撞检测. 漫游系统中需要用到的若干重要参数定义如下:

```
osg::Vec3 _direction;//中心至视点的距离
osg::Vec3 _position;//视点位置
osg::Matrixd _matrix;//视图矩阵
osg::Matrixd _inverseMatrix;//转置矩阵
osg::Matrixd _offset;//X 轴和 Y 轴方向的旋转矩阵
double _forwardSpeed;//向前速率
double _sideSpeed;//侧移速率
double _upSpeed;//纵向移动速率
double _speedEpsilon;//最小速率参照量, 用于归零
_doubleSpeed, _sideSpeed, _upSpeed
double _speedAccelerationFactor;//速度增加量, 用于控制
_forwardSpeed, _sideSpeed, _upSpeed
double _speedDecelerationFactor;//速度减少量, 用于控制
_forwardSpeed, _sideSpeed, _upSpeed
double _pitchOffset;//离 X 轴的旋转角度
double _yawOffset;//离 Y 轴的旋转角度
漫游系统实现细节如下:
```

(1)进入系统, 定位初始相机位置. 如图 1.6 所示, 首先将整个场景用球形包围盒确定范围. 设包围球参数为, 球心坐标 $C(x, y, z)$, 半径为 R . 再构建空间直线段 $L[(C_x, C_y, C_z - R), (C_x, C_y, C_z + R)]$, 运用 `osgUtil::IntersectVisitor()`方法进行直线段和空间场景的碰撞检测, 获取离球心最近的点 $P(C_x, C_y, z)$ 作为地面坐标(图 6 中红色点).

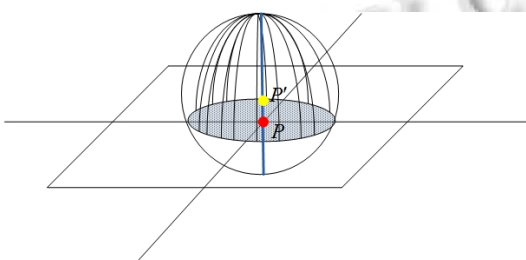


图 6 获取摄像机初始坐标

考虑到人的实际身高, 将 $P' = (C_x, C_y, z + 12.5)$ 作为相机的初始位置(图 6 中黄色点). 这种设置, 可以使购物者在刚进入漫游系统时, 站在商场正中央, 且处于合适的高度.

(2)回复到初始位置功能. 在漫游过程中, 可能由于误操作等问题, 使得漫游点无法回到正常视角位置, 此时可以通过该功能回到初始设置状态. `_position` 设为 P' , `_direction` 设为 $C - P'$, `_inverseMatrix` 设为 $(P', C - P', (0, 0, 1))$, `_offset` 设为单位矩阵, 速率参数 `_forwardSpeed`, `_sideSpeed` 以及 `_upSpeed` 置 0, 此时, 漫游点返回至初始位置, 且面向商场中央, 所有旋转和移动动作终止.

(3)移动和旋转控制. 移动由 `_forwardSpeed`(前后), `_sideSpeed`(左右)及 `_upSpeed`(上下)控制, 其速率由 `_speedAccelerationFactor` 及 `_speedDecelerationFactor` 控制. 旋转由 `_pitchOffset`(X 轴)及 `_yawOffset`(Y 轴)控制. 旋转矩阵设置为 `_offset = osg::Matrix::rotate(_yawOffset, osg::Vec3(0,1,0), _pitchOffset, osg::Vec3(1,0,0), 0.0, osg::Vec3(0,0,1))`; 新的视点方向设置为 `_direction = _direction * osg::Matrix::rotate(_directionRotationRate, osg::Vec3(0,0,1))`; 新的视点位置设置为 `_position = _position + ((_direction * _forwardSpeed) + (_direction * osg::Matrix::rotate(-3.14*0.5, osg::Vec3(0,0,1))*_sideSpeed) + (osg::Vec3(0,0,1) * _upSpeed))`

(4)在漫游过程中, 还要以视点为中心, 构建一小型包围盒, 实时进行碰撞检测, 若在旋转或者移动过程中, 检测到碰撞, 则立刻将 `_forwardSpeed`, `_sideSpeed` 及 `_upSpeed`, 包括其速率 `_speedAccelerationFactor` 及 `_speedDecelerationFactor` 设置为 0, 停止运动.

4 购物系统

整个购物流程大致如下: (1)漫游浏览商品; (2)选择商品, 根据选择显示相应商品信息; (3)将感兴趣商品加入购物车. 本文主要研究了在三维场景中的商品的展示与更新及商品信息显示技术.

4.1 商品的展示与更新

文章采用两种方法来展示商品: (1)对于大型商品或者需要精致观察其外部和内部结构的商品, 采用其精确的 3D 模型来表示. 购物者借助于漫游系统可以仔细浏览该类商品的各个细节部分, 而这一优点是目前平面购物网站所无法实现的. 采用模型表示法的缺点是精确模型的构建需要耗费大量的人力和时间, 并且精确模型数据量庞大, 不利于数据更新以及 3D 购物系统的在线应用^[8]. 因此, 在实际应用中, 采用模型

表示的商品应该是展示周期长, 并且更新缓慢的商品, 如家电, 家具等. (2)对于大量小型商品, 如日常生活用品, 其种类繁多, 款式更新快, 并且无需仔细观察其内部结构, 可以采用贴图的方式表示. 图 7 显示了这两种商品表示方法在本系统中的应用.



图 7 商品的展示

从我们的实验效果来看, 相对于模型展示方法, 由于贴图展示的图片均来自于实际商场(欧尚超市), 因此贴图反映了其在货架上的真实形态, 在 3D 漫游时, 给人以非常逼真的感觉. 同时, 由于贴图展示的商品可以大批量更换, 图片可以迅速获取和更新, 且实现简单, 因此, 该方法非常适合模拟超市等主要基于货架销售的商场.

4.2 购物操作流程

本系统中, 商品信息主要包含柜台号、纹理号、商品号、商品名、商品信息、价格以及折扣. 购物车信息包括商品号、价格、折扣、购买数量及价格合计. 购物操作主要包含两个步骤:

(1)商品信息显示与选择. 场景漫游, 看到合适商品时, 单击鼠标左键, 停止漫游, 在屏幕右边缘单击鼠标右键, 弹出商品图片框, 同时在屏幕右边缘显示图片中所有商品信息, 商品信息可以通过鼠标滚轮浏览. 由于一张图片中包含多个商品, 弹出的商品图片可以通过鼠标滚轮放大, 并在框内移动, 便于购物者仔细浏览单个商品, 如图 8(a)所示. 图片的放大缩小功能通过 osg::Image 类实现. 同时为了使显示的商品不影响漫游效果, 弹出的商品信息框均以半透明状态显示, 同时通过 osg::Text 类实现文字显示(注: 由于实现过程中, 未能解决 OSG 的中文显示问题, 因此采用英文显示商品信息). 图 8(b)、(c)和(d)分别表示商品信息框的三种状态: 未选择商品、已购买商品以及选择的商品, 采用不同的颜色表示, 以方便购买者区分商品状态.

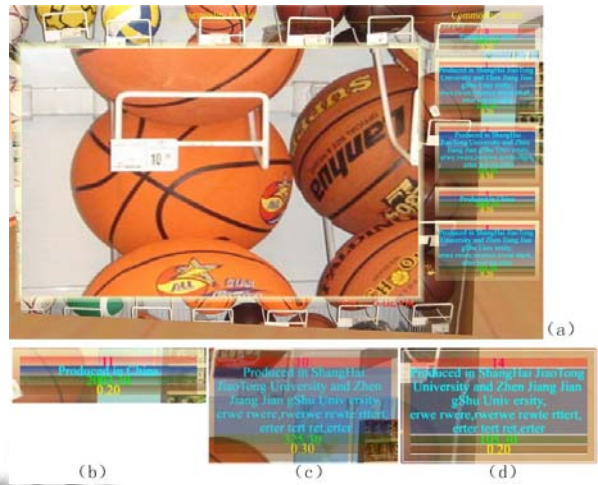


图 8 商品信息显示 (a)实时商品信息显示图 (b)商品信息:未选择商品 (c)商品信息:已购买商品 (d)商品信息:选择商品

图 8 商品信息显示

(2)购物车. 购物车同样采用彩色半透明文字显示. 图 9 显示了购物车状态.



图 9 购物车信息显示

购物时, 购物者只要点击相应商品信息, 就可将商品加入购物车中, 初始数量为 1. 在购物车界面商品数量部分(图 9 中红色框内部分)选择相应商品栏, 单击鼠标左键, 即可增加商品数量, 且单类商品总价(图 9 中蓝色框内部分)及所有商品总价(图 9 中黄色框内部分)将会根据商品数量的变化实时计算并显示. 反之, 单击鼠标右键则会减少商品数量, 当数量减为 0 时, 再次显示购物车时该类商品自动从购物车中消失.

商品选择与信息显示的实现细节(见图 10)包含以下几个步骤:

(1)在 1.1 节基于 3DMAX 构建完商场模型之后, 在商场内精确计算展示柜位置, 并将其加入商场模型中, 同时在商场的场景图中找到所加入的展示柜, 并在其节点内添加柜号附加信息, 其所连接的纹理贴图添加纹理号附加信息.

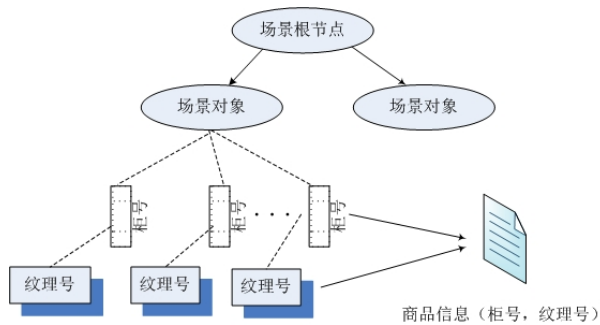


图 10 获取商品信息流程

(2)商品选择的实现. 在商场内漫游时, 若看中某柜台商品, 单击柜台, 系统通过 `osgUtil::LineSegmentIntersector` 在视线方向构建碰撞检测线段, 进行碰撞检测, 查找最近的碰撞检测点, 并通过该点获取其节点路径 `nodePath` 信息, 通过 `nodePath` 找到柜台节点, 并获取柜号信息以及纹理号信息.

(3)通过柜号+纹理号, 检索商品信息文档, 查找该文档, 获取对应的商品信息.

5 结论

文章将目前最先进的视景仿真技术和信息管理技术融合在一起, 致力于营造一种具有现实购物氛围的 3D 网上购物环境, 主要研究了构建 3D 商业场景, 基于 Open Scene Graph 技术实现视景漫游. 并研究了基于模型的商品展示方式和基于贴图的商品展示方式, 以及商品的信息显示方式. 最后模拟了较完善的购物

流程. 通过构建的模拟系统, 我们可以深刻感受到 3D 购物的乐趣和前所未有的真实感. 但是, 离实用化还存在较大的差距, 例如大量的实体对象建模、模型数据压缩与实时传输等问题需要进一步研究.

参考文献

- 1 Martz P. OpenSceneGraph Quick Start Guide. Louisville, USA, 2007.
- 2 Wang R, Qian XL. OpenSceneGraph 3.0: Beginner's Guide. Packt Publishing, 2010.
- 3 Wang R, Qian XL. OpenSceneGraph 3 Cookbook. Packt Publishing, 2012.
- 4 Xu K, Stewart J, Fiume E. Constraint-based automatic placement for scene composition. Graphics Interface, 2002: 25-34.
- 5 Armstrong WW, Green WM. The dynamics of articulated rigid bodies for purposes of animation. Proc. of Graphics Interface'85. 407-415.
- 6 Moore M, Wilhelms J. Collision detection and response for computer animation. Comput Graph, 2002, 22: 289-298.
- 7 Ghosh PK. A solution of polygon containment spatial planning and other related problems using minkowski operations. Computer Vision Graphics and Image Processing, 1990, (49): 1-35.
- 8 Shinya M, Forgue MC. Laying out objects with geometric and physical constraints. The Visual Computer, 1995,11(4): 188-201.