

城市交通环境视景三维仿真

刘晶郁, 李子然, 杨 炜

(长安大学 汽车学院, 西安 710064)

摘要: 为解决城市视景仿真还原度差、系统流畅度不佳等问题, 提出一种基于 Vega Prime 与 VC++ 联合仿真的视景仿真框架; 使用 Multigen Creator 与 3D MAX 联合建模, 在 VC++ 环境下利用 Vega Prime 实现场景动态驱动; 并着重研究了模型优化、节点动态驱动、模型封装等视景仿真关键技术; 利用上述视景建模关键技术, 完成城市交通环境视景仿真系统搭建; 应用结果表明: 关键技术的应用将视景加载时间缩短为初始加载时间的 1/6, 平均帧频提升约 30%, 并提供了丰富逼真的图形画面, 良好的用户感受, 呈现出高还原度的视景仿真效果。

关键词: 城市交通; 视景仿真; 模型优化; Vega Prime

3D Visual Scene Simulation of Urban Traffic Environment

Liu Jingyu, Li Ziran, Yang Wei

(School of Automobile, Chang'an University, Xi'an 710064, China)

Abstract: In order to solve the problem of poor restoration and poor system fluency in urban visual simulation, a visual simulation framework based on co-simulation of Vega Prime and VC++ is proposed. Multigen Creator and 3D MAX are used to model the scene and Vega Prime is used to realize scene dynamic driving in VC++ environment. The key technologies of visual simulation such as model optimization, node dynamic driving and model encapsulation are studied. The construction of the urban traffic environment visual simulation system is completed with the key technologies of vision modeling. The results show that application of key technologies shortens the view loading time to 1/6 of the initial loading time and increases the average frame rate by about 30%. The visual simulation system provides rich and realistic graphic images, good user experience and high-reliability visual simulation effects.

Keywords: urban traffic; visual scene simulation; model optimization; Vega Prime

0 引言

随着计算机科学的飞速发展, 虚拟现实技术日益成熟。作为虚拟现实技术的重要表现形式, 视景仿真技术广泛应用于虚拟驾驶、军事演练、城市交通仿真等领域^[1]。其中, 在城市交通仿真领域, 视景仿真是城市交通仿真研究的必要前提^[2]。城市交通视景仿真以计算机技术为依托, 利用系统仿真模型模拟城市交通系统的运行状态, 其对交通流、交通事故等交通现象的模拟^[3], 为城市道路设计、城市规划等项目提供较为详尽的交通参数和技术支持。但城市交通系统运行复杂、城市环境要素多种多样(包含车辆、建筑、公交站牌、行人等)^[4]。为满足城市交通仿真效果的真实性、流畅性, 提升视景环境的沉浸感, 场景建模及视景仿真技术的运用, 则成为城市交通视景仿真的关键^[5]。对此, 相关文献运用多种方法提升视景仿真效果, 其中, 文献 [6] 利用 multigen creator/Vega Prime 平台完成三维场景建模, 并阐述建模一般步骤, 但未对模型优化展开深入研究; 文献 [7] 采用 multigen creator 与 3DMAX 联合建模建立沈阳市三维模型, 解决模型逼真度问题, 但未对模型

渲染效率进一步探讨; 文献 [8] 利用 3DMAX 三维建模技术以及 Java 编程语言完成了虚拟城市广场漫游导航系统的搭建, 实现了模型与程序之间有机结合, 但功能较为单一, 系统流畅性有待改进。

针对上述问题, 本文基于主流三维建模技术和视景仿真平台, 提出适用于城市交通虚拟环境构建的视景仿真框架, 对视景仿真过程中的关键技术进行分析研究, 并将主要关键技术应用于城市交通视景仿真系统的构建。关键技术的应用显著提升了仿真系统的实时渲染及加载响应速度, 并取得高流畅性、高逼真度的视景仿真效果。

1 视景开发框架设计

城市交通视景仿真总体框架如图 1 所示。视景仿真主要包括城市交通三维模型建立、场景模型优化、Vega Prime 配置、VC++ 程序设计四部分。为满足视景仿真实时性高、加载响应迅速等要求, 需将模型建立、模型优化、Vega Prime 配置、程序调用四部分有机结合, 充分发挥各部分优势。

视景仿真步骤如下:

1) 城市交通环境三维建模。利用 Multigen Creator 与 3D MAX 对城市交通场景联合建模, 模型包括路面地形模型、道路环境模型、动态场景模型三部分, 其中动态场景模型用于程序调用。将场景模型储存为 flt 文件。

2) 场景模型优化。对场景进行改进, 通过模型节点优

收稿日期: 2019-06-07; 修回日期: 2019-06-26。

基金项目: 中央高校基本科研项目(300102229112)。

作者简介: 刘晶郁(1963-), 女, 陕西西安人, 教授, 硕士研究生导师, 主要从事车辆安全、设计等方向的研究。

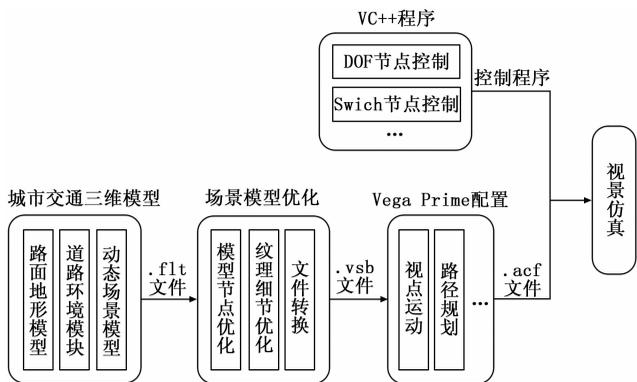


图 1 城市交通视景仿真总体框架

化、纹理细节优化等技术，以提高渲染效率。并将模型 flt 文件封装为 vsb 格式的二进制文件，可明显提升加载效率，缩短模型加载响应时间。

3) Vega Prime 配置。利用 Vega Prime 面向对象、功能模块化、平台兼容性好的优点^[9]，完成模型文件（vsb 文件）加载，城市交通视景环境初始配置（视点位置、视点运动、路径规划等）工作，并生成环境初始配置文件（acf 文件）。

4) VC++ 程序。编制 VC++ 主程序，并调用 Vega Prime 函数库进行联合开发，实现场景驱动以及视景中运动节点（DOF 节点、Switch 节点）的控制。

2 视景系统关键技术分析

2.1 3D MAX 与 Multigen Creator 联合建模技术

作为视景仿真过程中的关键要素，三维模型的建立需要具备高效、便捷、可视化程度高等特点，选择合适的视景仿真建模软件，对视景仿真系统的搭建十分重要。本文采用 3D MAX 与 Multigen Creator 联合建模，其建模流程如图 2 所示。

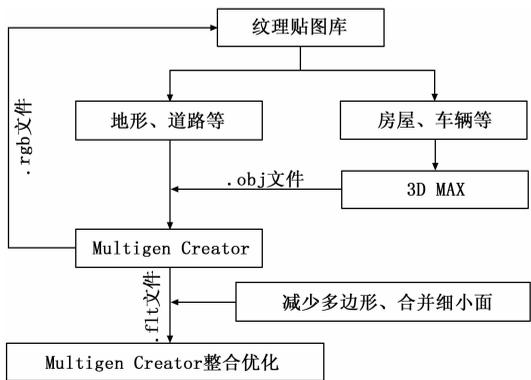


图 2 Multigen creator 与 3D MAX 联合建模流程图

城市交通场景中模型元素复杂多样，为兼顾场景精度与流畅度，本文采用虚拟场景分开再合并的方式进行建模。联合建模流程如下：

1) 建立纹理贴图库。采用实地拍摄、颜色材质替换等方法收集贴图素材，并对素材进行补光、裁剪等图像处理，

丰富纹理贴图库；

2) 建立三维模型。利用 Multigen Creator 多边形建模的优势，进行地形、树木、道路等模型的搭建，利用 3D MAX 强大的点、线、面以及几何体编辑功能建立复杂模型，如车辆、房屋等；

3) 场景模型整合。3D MAX 模型通过 obj 文件导入 Multigen Creator，由于 3D MAX 模型三角面较多，需进行合并细小面、减少多边形面处理^[10]，整合后的场景贴图还可转换为 rgb 格式，丰富纹理库。

2.2 模型数据库优化技术

城市交通环境要素复杂，涉及众多三维实体，如车辆、树木、路灯、立交桥等，需建立数量庞大的三维模型，针对众多模型，需对场景合理优化以满足视景仿真实时性、流畅性的需求。

2.2.1 子纹理贴图优化

为达到逼真显示效果，三维模型建立后常利用纹理映射技术将二维图像映射到模型表面^[11]。但 creator 对纹理贴图分辨率有所限制：贴图长和宽限制为 2 的幂次方个像素单位，即图像分辨率为 $2^n \times 2^n$ 。当贴图分辨率尺寸不满足 2 的幂次方像素单位要求时，系统会自动分配更大内存来存储贴图，以分辨率大小为 180×420 为例，其尺寸分配如图 3 所示。这样会造成内存资源浪费，影响贴图的加载与渲染效率。



图 3 纹理原始尺寸与内存分配尺寸关系图

在众多城市场景要素中，不规则纹理的存在不可避免，就需对其进行子纹理贴图优化，即将各个分散的不规则纹理合并为一个主纹理，合并原理如图 4 所示，其中 A、B、C、D、E 表示子纹理贴图或透明图像留白，以分辨率 1024×1024 图像为例，子纹理合并实例如图 5 所示。主纹理映射方法与其他纹理映射方法相同，需重新建立纹理映射关系。优化后主纹理分辨率最大不应超过 1024×1024 ，以缩短模型加载过程中纹理解压时间。

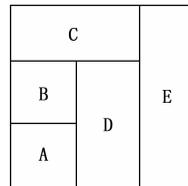


图 4 子纹理合并图

2.2.2 层级结构节点优化

Multigen Creator 采用层级结构进行三维模型的存储^[12-13]，层级结构如图 6 所示，节点的层级结构影响视景

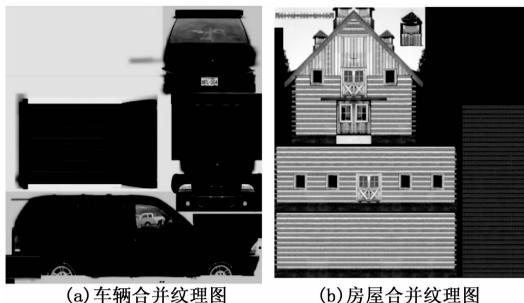


图 5 合并纹理图

渲染过程中节点的生成与剔除。视景渲染时, 会遍历所有数据库节点, 并根据节点位置是否在可视范围内进行节点的截取计算, 渲染可视范围内节点, 并剔除可视范围之外的节点。

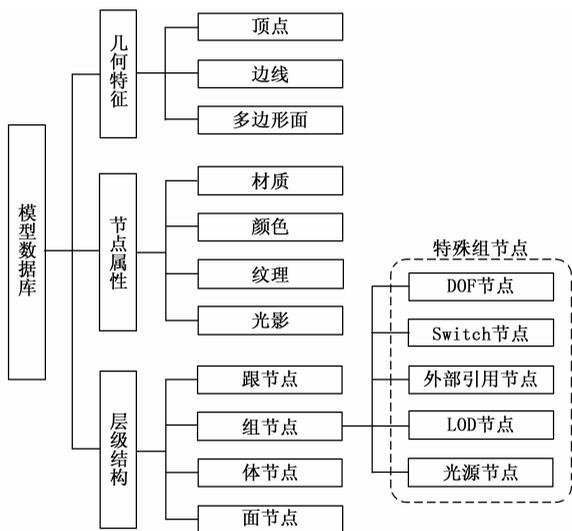


图 6 Multigen Creator 层级结构图

城市交通场景复杂, 元素种类众多, 建模过程中, 为方便模型统一管理, 易于修改、调用, 常采用逻辑结构建立模型数据库, 如创建树、房屋、车道、交通灯、公交站等组节点, 并将树、房屋等体节点放在对应组节点下, 节点信息如图 7 所示。

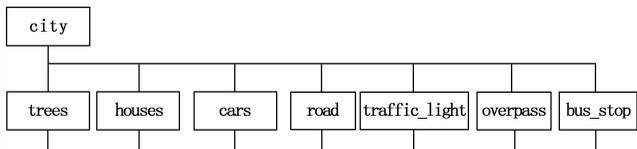


图 7 模型数据库逻辑结构

逻辑结构建立的模型数据库条理清晰、易于管理, 但会影响模型渲染过程中节点的剔除与加载速度。场景制作完成后, 需按照空间位置将组节点重新划分, 本文将整体场景按照空间几何形状分为 7 块, 每块含有多种城市元素, 节点信息如图 8 所示。

2.2.3 模型封装技术

城市交通场景元素复杂, 导致原始 flt 模型文件较大,

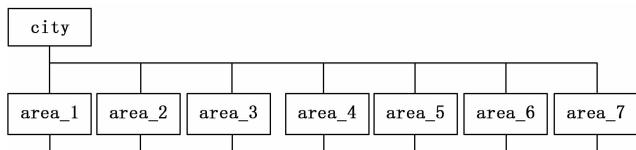


图 8 模型数据库空间结构

严重增加场景加载时间, 利用 to_vsb.exe 工具将 flt 文件封装为 vsb 格式的二进制文件, 可大大提升模型文件加载效率^[14]。VSB (Vega Scene Graphic Binary) 是一种不可编辑的场景文件格式, 专门用于 Vega Prime 模型文件的加载, 其具有模型解压速度快、忽略纹理路径影响、封装元素全面等优点。但在格式转换时也应注意以下两点:

1) 纹理贴图与 vsb 文件需放入同一文件夹。文件转换时, 转换工具忽略贴图加载路径, 只将同一文件夹下的 flt 文件与贴图文件打包入同一个 vsb 文件, 如文件夹下无贴图文件, 则只转换白模型。

2) 建立精细树模型, 不可进行 vsb 格式转换。精细树模型需以替换场景中普通树模型的方式进行加载, 其通过寻找 flt 文件中树木模型路径及名称进行精细树替换, 而 vsb 文件不包含树木模型路径信息, 丢失的信息会影响精细树的生成。

2.3 节点动态驱动技术

2.3.1 Switch 节点驱动

作为城市视景环境重要标志之一, 交通灯模型必不可少, 本文采用 switch 节点技术进行信号灯模拟仿真, switch 节点技术可将不同体模型放置在同一节点下, 并通过函数调用控制节点下模型变换。交通灯模型制作中, 将创建完成的红、绿、黄 3 个体节点放在同一个 switch 节点下, 设置 switch 节点, 并按顺序对 3 个颜色体节点编号 (如表 1), 以便于通过时间间隔控制节点变换, 节点设置如图 9 所示。

表 1 颜色编号

颜色	红	绿	黄
Index	0	1	2
时间间隔/s	25	22	3

VC++调用 vega API 函数获取仿真时间 T , 通过遍历节点找出特定 switch 节点, 并根据仿真时间与不同信号灯之间的时间间隔确定 switch 节点中的 index 值, 然后通过判断车辆位置实现车辆是否闯红灯判定, 节点控制流程如图 10 所示, 控制 switch 节点关键代码如下:

```

vsSwitch * mySwitch; //定义 switch 节点
tempNode = find_named("switch");//寻找 switch 节点
mySwitch = (vsSwitch *) tempNode;//将 switch 节点赋予 mySwitch
Time = vpKernel::instance()->getSimulationTime();//获取仿真时间
mySwitch->setActiveMask(0);//实现红灯渲染

```

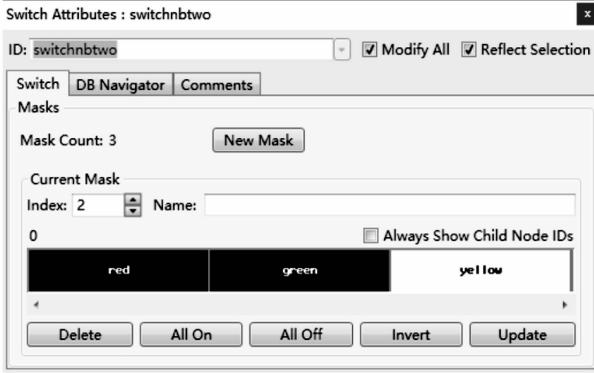


图 9 switch 节点设置

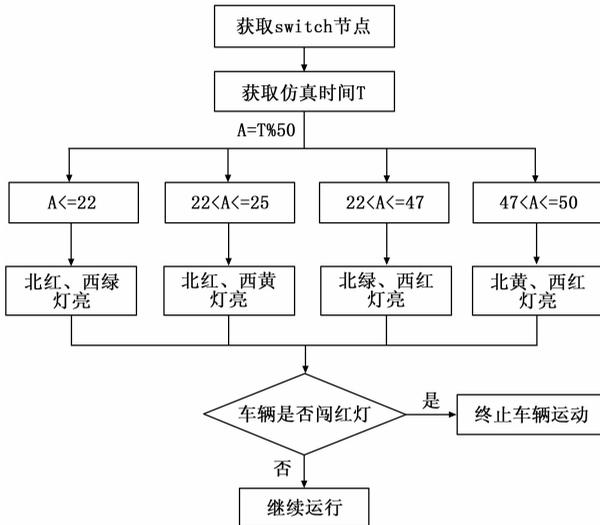


图 10 switch 节点控制流程图

```
mySwitch->setActiveMask(1); //实现绿灯渲染
mySwitch->setActiveMask(2); //实现黄灯渲染
```

2.3.2 DOF 节点驱动

DOF 节点可控制其所有子节点按照设置的自由度移动或旋转，赋予模型运动的能力。城市交通场景中，车轮的旋转与转向、雨刮的摇摆、虚拟仪表盘的建立等都适合 DOF 节点技术创建。

DOF 节点的创建，需对目标模型创建局部坐标系 $O'x'y'z'$ ，其示意如图 11 所示，DOF 节点封装有 2 个仿射变换矩阵 T_1 、 T_2 。 T_1 矩阵用来描述局部坐标系相对于世界坐标系的位置、平移、缩放等变换， T_2 矩阵则模型物体相对于局部坐标系的位置、平移、缩放等变换。其矩阵可表示为：

$$T_1 = T_2 = \begin{bmatrix} A_{3 \times 3} & B_{3 \times 1} \\ C_{1 \times 3} & 1 \end{bmatrix}$$

其中： $A_{3 \times 3} = \begin{bmatrix} T_x & T_y & T_z \\ R_x & R_y & R_z \\ S_x & S_y & S_z \end{bmatrix}$ 可描述局部坐标系及模型

的平移、旋转、缩放等变换； $C_{1 \times 3} = [\Delta x \ \Delta y \ \Delta z]$ 可描述局部坐标系及模型的相对位置； $B_{3 \times 1} = [0 \ 0 \ 0]$ 为透视变

换。通过矩阵变化可实现 DOF 节点驱动。以雨刮摇摆为例，设置雨刮旋转中心为局部坐标系中心点，确定 x 、 y 坐标轴，设置雨刮旋转自由度，并确定旋转范围，如图 12 所示。

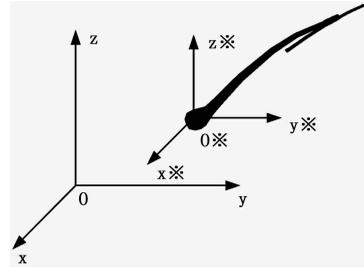


图 11 DOF 节点示意图

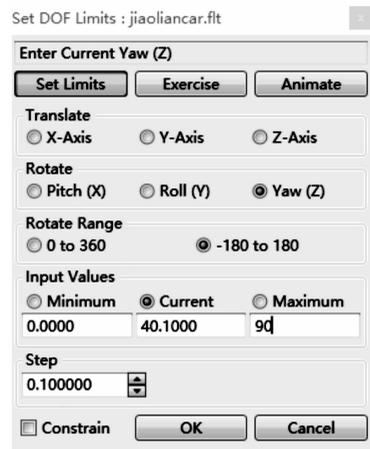


图 12 DOF 节点设置

与 switch 节点驱动类似，DOF 节点驱动也需对模型节点遍历，并找出特定 DOF 节点，对节点模型设置运动状态，实现驾驶人与车辆模型间的交互功能。控制 DOF 节点关键代码如下：

```
vsDOF * wiperDOF; //定义 DOF 节点
wiperDOF = (vsDOF *) m_car_obj -> find_named("wiper"); //寻找节点并将其赋予 wiperDOF
Time = vpKernel::instance()->getSimulationTime(); //获取仿真时间
wiperDOF ->setRotateH( 2, true); //驱动 DOF 节点,并设置每一帧旋转角度为 2°
wiperDOF ->setRotateH( 0, true); //设置每一帧旋转角度为 0°,DOF 节点旋转停止
```

3 仿真实例及效果

应用上述视景建模与仿真技术，构建城市交通视景仿真系统，硬件配置：CPU 为 Intel i7-8700 3.2 GHz，内存 16 G，硬盘 2TB，显卡为 GTX 1070；软件配置：操作系统 Windows10，Multigen Creator，3D MAX，Vega Prime，Microsoft Visual C++。

视景驱动效果如图 13~14 所示。采用 Multigen Creator 与 3D MAX 联合建模，将多种元素模型有机结合，资源利

用合理, 可在满足场景逼真度的同时, 兼顾视景仿真的实时性。节点动态驱动技术的运用, 使场景要素更加丰富, 场景更贴近真实, 还原度更高, 仿真效果良好。



(a)居民楼视景 (b)办公楼视景

图 13 房屋视景仿真效果



(a)交通路口视景 (b)立交桥视景

图 14 交通路口和立交桥视景仿真效果

场景加载与渲染效率方面也有较大提升, 模型优化技术应用前后实时性能对比如表 2 所示。从表中得知, 简单场景下, 模型封装技术可明显提升场景加载效率, 其加载时间为 6 s, 约为未使用封装技术场景加载时间的 1/3, 模型优化技术可一定程度上提升场景流畅度, 其剔除和渲染时间相较未使用优化技术的场景有约 2 ms 的缩短; 复杂场景下, 优化与封装技术的优势进一步扩大, 模型封装技术的使用可将场景加载时间缩短为原来的 1/6, 约为 21 s, 而在采用模型优化技术后, 场景平均帧率维持在 37 帧/s 左右, 提升约为 30%, 平均剔除和渲染时间为 8.4 ms 和 8.2 ms, 缩短为原来的 1/2。

表 2 关键技术应用前后实时性能对比

方案内容		对比项			
		加载时间/s	平均帧频/(帧/s)	平均剔除(cull)时间/ms	平均渲染(draw)时间/ms
总场景 (复杂场景)	应用模型优化技术	79	37	8.4	8.2
	应用模型封装技术	27	25	15.9	14.4
	同时应用封装和优化技术	21	37	8.6	8.3
	未使用优化和封装技术	124	24	16.6	14.2
立交桥 场景(简单场景)	应用模型优化技术	17	59	3.2	2.8
	应用模型封装技术	6	59	5.5	5.5
	同时应用封装和优化技术	6	60	2.9	2.9
	未使用优化和封装技术	19	58	5.9	5.5

4 结论

本文提出一种基于 VC++ 与 Vega Prime 联合仿真的城市交通视景仿真框架, 解决了城市视景仿真中实时性差、还原度较低的问题, 并就其中关键建模技术、模型优化技术以及节点动态驱动技术作重点研究。Multigen Creator 与 3D MAX 联合建模技术满足场景逼真度的同时, 兼顾了仿真实时性需求, 模型优化技术的使用大大提升了视景加载及渲染效率, 节点动态驱动技术丰富了场景元素, 场景还原度更高。示例效果显示, 优化技术的应用使视景加载与渲染效率大幅提升, 视景仿真系统提供了逼真的图形画质, 并以较高的城市交通还原度, 实现了高沉浸感的视景仿真效果, 为进一步的城市交通流模拟训练、驾驶人应激等研究提供基础。

参考文献:

- [1] 赵沁平. 虚拟现实综述 [J]. 中国科学 (F 辑: 信息科学), 2009, 39 (1): 2.
- [2] 陈鹏, 童小华, 宋二非. 城市道路交通虚拟场景的快速建模方案 [J]. 系统仿真技术, 2017, 13 (2): 105-111.
- [3] 商蕾. 城市道路交通流仿真系统研究 [J]. 武汉理工大学学报 (交通科学与工程版), 2010, 34 (3): 587-590.
- [4] 夏萍. 基于虚拟现实技术的复杂城市道路交通仿真平台研究 [D]. 湖北工业大学, 2011.
- [5] 张昱洁. 基于 MultiGen Creator 和 Vega 的虚拟城市仿真技术研究 [D]. 长沙: 湖南大学, 2008.
- [6] 徐恩, 李学军, 邹红霞, 等. 基于 Creator/VP 的三维虚拟环境建模 [J]. 系统仿真学报, 2009, 21 (s1): 121-123.
- [7] 陶维刚, 许振峰, 毛婧一. 3DS Max 与 MultiGen Creator 联合建模在建立三维城市模型中的应用 [J]. 测绘与空间地理信息, 2014, 37 (7): 123-124, 127.
- [8] 周小江. 基于 VRML 的虚拟城市广场漫游导航系统的设计与实现 [D]. 成都: 电子科技大学, 2014.
- [9] 王远明, 卢宽, 贾倩, 等. 基于 Unigine 的舰载航空视景仿真技术研究 [J]. 系统仿真学报, 2017 (9): 2087-2092.
- [10] 郑海春, 董秀成, 王孝平, 等. MultiGen Creator 虚拟现实三维建模优化技术的研究与实践 [J]. 西华大学学报 (自然科学版), 2011, 30 (6).
- [11] Liu P, Peng Z H. Three - Dimensional Modeling of Virtual City: Taking MultiGenCreator for Example [J]. Advanced Materials Research, 2013, 711: 617-622.
- [12] Weijun P, Feng X, Wenbin Q, et al. Visual simulation airport training system based on C919 performance [A]. IEEE Workshop on Electronics, Computer & Applications [C]. IEEE, 2014.
- [13] Li Z, Wang X. The application and implementation of real-time three-dimensional video scene in virtual target aiming training [J]. Applied Mechanics and Materials, 2014, 556-562: 5288-5293.
- [14] 王明印, 韦群, 徐恩, 等. 基于 Creator/Vega Prime 的大场景虚拟现实关键技术研究 [J]. 系统仿真学报, 2009, 21 (s1): 117-120.