

基于 Web 的 3D 模型浏览与交互系统的研究与实现

郭明金[†], 陈姝

(湘潭大学 信息工程学院 湖南 湘潭 411105)

摘 要 设计了基于 Web 的 3D 模型浏览与交互系统,主要采用基于 WebGL 的第三方库 Three.js 来实现。该系统的建立主要包括场景的构建、场景的交互控制和模型的云存储。模型是场景构建的前提,用户可从外部导入所需模型或利用系统自带的简单模型进行模型的构建。场景交互技术主要包括模型的平移、旋转、缩放、拾取、剖切、3D 动画模型的播放以及信息的显示等。模型的云存储是指用户可以通过登录账户将 3D 模型存储到服务器,日后需要时直接从服务器中加载即可。

关键词 3D 模型;WebGL 技术;Three.js 库;服务器

中图分类号 TP311.5

文献标识码 A

Research and Implementation of 3D Model Browsing and Interaction System Based on Web

GUO Ming-jin[†], CHEN Shu

(College of Information and Engineering, Xiangtan University, Xiangtan, Hunan 411105, China)

Abstract 3D model browsing and interaction system based on Web is designed, which mainly adopts the third-party library Three.js based on WebGL to realize. The establishment of the system mainly includes the construction of the scene, the interactive control of the scene and the cloud storage of the model. The model is the premise of the construction of the scene, users can import the required model from the outside or use the system's simple model to build the model. The interactive technologies of the scene mainly include the translation, rotation, zooming, picking up of the model, playing of the 3D animation model and display of information and so on. The cloud storage of the model means that the user can store the 3D model on the server by logging into the account, and can directly load it from the server when needed.

Key Words 3D model; WebGL technology; Three.js library; server

随着计算机技术的不断发展,计算机图形学在建筑、科学、医药、商务和娱乐等各种领域的应用越来越广泛,目前已经成为各个应用领域中极其重要的技术,成为计算机视觉、图像处理、多媒体技术、模式识别等各个学科的技术基础。Web3D 作为三维计算机图形学与网络技术的产物,其发展和应用非常广泛,它可以在网页上创建一个服务器,使得

用户可以直接通过浏览器实现 3D 场景的实时渲染与互动^[1]。

随着 3D 技术的应用领域逐渐增多^[2],普通用户对其需求越来越强烈,为了满足这些用户的需求,必须要创建出一个适合大众用户且操作简单、方便快捷的 3D 模型浏览与交互工具。然而,目前的 Web3D 技术却存在一些问题。首先,这些技术都

收稿日期 2018-05-08

基金项目 湖南省自然科学基金资助项目(2017JJ2252),湖南省教育厅青年项目资助(16B258),广西南宁市第五批特聘专家项目(南发办(2017)61 号)

作者简介 郭明金(1993—),女,湖南永州人,硕士研究生,研究方向:计算机视觉。

[†] 通讯联系人, E-mail: 1139612683@qq.com

必须要安装特定的插件,插件的安装会影响网页的跨平台性及稳定性,特别是限制了其在移动终端网页中的应用;其次,许多技术仅仅针对特定行业,其通用性不强,比如 Unity3D 只适合游戏网页的开发;再次,技术实现较为困难,开发成本高,并且制作效率低下。为了解决这些问题,本系统采用 WebGL 技术^[3],WebGL 的优势是不用在浏览器上安装插件,就能被应用到网页中去以创建多种多样的复杂 3D 模型,在硬件条件相同的情况下提高 3D 模型的渲染性能^[4]和渲染效果。目前,国内 Web3D 相关产品比较稀缺,资源和资料也较稀少,所以,提出的基于 Web 的 3D 模型浏览与交互系统是虚拟现实技术亟待解决的重要技术问题之一,应用前景非常好^[5]。

1 系统方案设计

1.1 系统架构

该系统采用 B/S 结构,由三部分组成,分别是客户端、服务器端和数据库。服务器端采用 Java 编写,对数据库中各表进行了封装,从而使得数据的安全性得到了保障。不同客户端如手机、电脑等通过 Web 浏览器导入或加载模型,服务器接收其请求,并将模型传送到客户端,客户端的浏览器通过 WebGL 加速渲染模型^[6],形成三维场景。用户可以在客户端的场景中实现场景漫游和模型平移或缩放等操作。同时,客户端可以向服务器发送请求,查看数据库中存储的模型。系统架构图如图 1 所示。

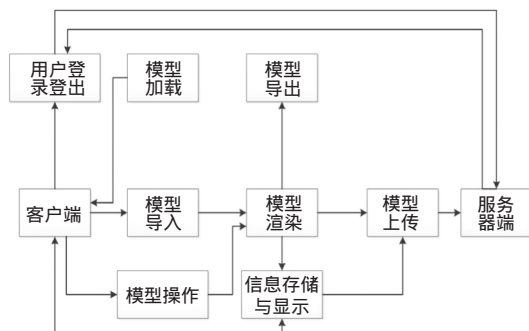


图 1 系统架构图

1.2 功能模块

系统的功能模块主要分为服务器端与客户端,客户端主要用于用户与系统的交互,服务器端则主要负责存储用户上传的模型,并能通过用户的操作实时加载模型到场景中。客户端包括本地导入、本地导出、模型操作、渲染、信息的存储与显示等功能

模块。服务器端主要包括用户登录、用户登出、模型与场景的加载、上传和删除等功能模块。其系统功能模块图如图 2 所示:

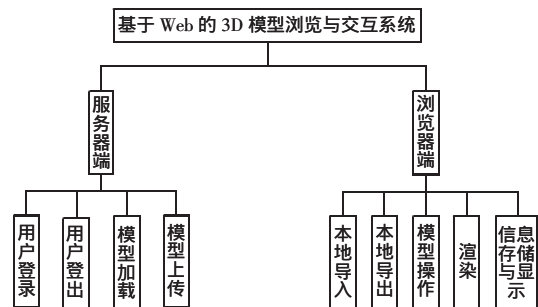


图 2 系统功能模块图

1.3 数据库设计

为了使模型能够云端存储,实现模型的动态加载与上传,用户可登录自己的账号将模型数据存入 MySQL 数据库中,从中加载上传过的模型,通过数据库的查询和删除等操作实现模型的加载与删除等操作。

本系统所涉及的表主要有用户表 tb_user、平台表 tb_platform、模型总表 tb_total_model、模型详表 tb_detail_model 和状态表 tb_status,各个表之间的联系如图 3 所示。

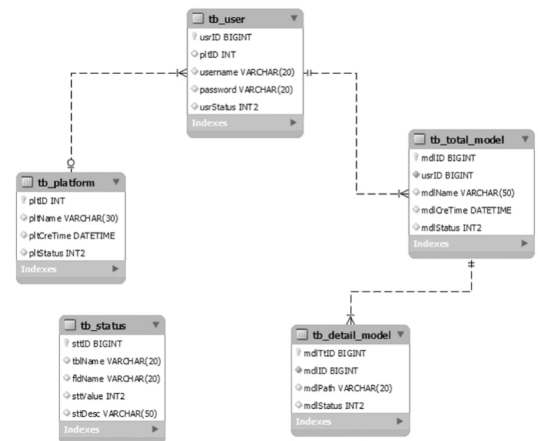


图 3 数据库

2 系统实现

2.1 界面实现

主界面分为三个区域:场景渲染区域、菜单栏区域及信息栏区域,如图 4 所示。场景渲染区域是系统界面最主要的区域,用来对 3D 场景进行实时渲染;菜单栏区域为用户与界面交互的区域,用于用户的注册登录、模型的导入导出、编辑与操作等的控制,采用下拉列表的方式;信息栏区域包括场

景信息显示栏、模型基础信息栏和模型材质栏,用于指导用户使用该系统及显示系统的其他相关信息。

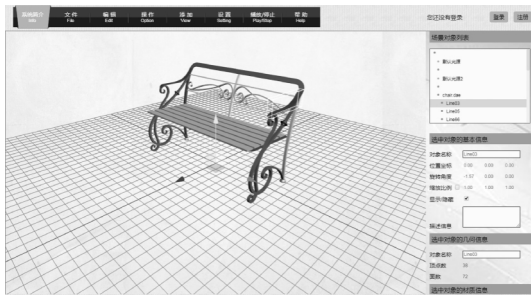


图4 主界面实现图

2.2 主要功能实现

(1)本地导入

系统支持导入外部模型到场景中渲染,外部模型格式可选,包括 wrl、obj、dae、stl、ply 等国际通用格式,系统还支持解析 3D 动画模型文件,可以支持 dae 格式和 vrmf 格式的 3D 动画模型的播放。

(2)本地导出

系统支持导出场景以及场景中的模型,支持导出 js、stl 和 obj 格式,可导出场景中的所有信息,包括灯光及其基本信息,模型及其基本信息与材质信息,以及场景列表中的信息。

(3)模型加载

用户在登录状态下,选择加载后,客户端向服务器发送模型 ID,服务器根据模型 ID 查找相应的模型信息,包括 3D 模型和其贴图,将其返回给客户端,加载到浏览器中。

(4)信息存储与显示功能

信息存储与显示功能主要用于模型基本信息的展示,用户修改模型时模型信息会实时显示在信息栏,同样的,若用户在信息栏修改模型信息,修改的信息也会实时渲染到场景中。信息栏的显示功能主要包括场景对象名字列表、选中对象的基本信息、几何信息和材质信息。

3 系统测试与分析

3.1 实验环境

系统的开发环境分为前端和后台。前端开发工具采用 WebStorm 8,页面设计语言采用 HTML5,页面排版布局采用 CSS 样式处理,尽可能的减少页面刷新操作,前端语言采用 JavaScript 语言,并运用 jQuery.js 及 Three.js 框架进行设计,后台语言采用 Java 语言,数据库采用 sql 语言,前端与后台数据交互使用 ajax 技术。

系统的运行环境为支持 WebGL 的浏览器,本系统 PC 端测试采用 Windows 7 操作系统上的 Google Chrome 65 浏览器,移动端测试环境使用 Android 的 UC 浏览器。

3.2 实验分析

(1)渲染比较

本系统是基于 Web 的 3D 模型浏览与交互系统,需要在浏览器中渲染出客户端软件的渲染效果。因此,对本系统导入模型后的渲染效果和 3dsMax 2014、Cortona3D 和 BS Contact 软件的渲染效果进行了比较,如图 5 所示。

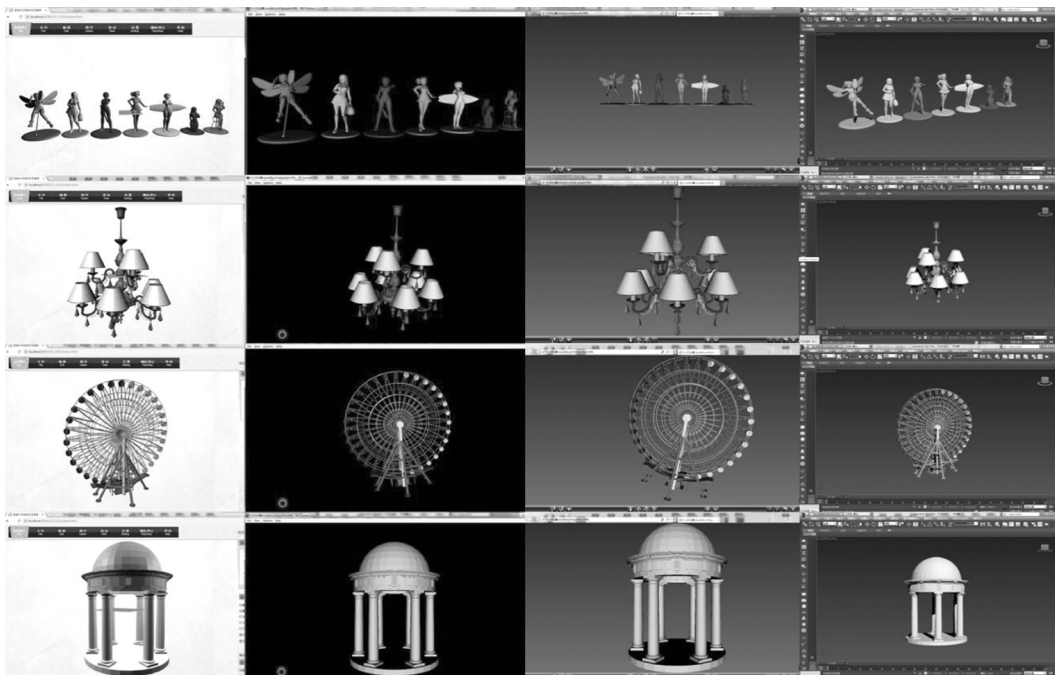


图5 模型在不同软件中的渲染效果比较图

图中四行四列展示了四个模型在四种系统中的渲染效果,其中,本系统为第一列,Cortona3D、BS Contact 和 3DMax 则分别为第二列、第三列以及第四列。从图中可以看出,本系统与三个软件的渲染效果存在一定的差异,不过都达到了比较好的渲染效果,四者的渲染效果相差不大。可见本系统达到了预期的渲染效果。

(2)性能测试

性能测试主要测试系统的 3D 场景渲染性能,

帧率是检测系统渲染性能的关键指标之一。帧率就是在渲染场景时图形处理器每秒钟的刷新次数。帧率越高说明系统展示的动画效果越流畅。通常每秒 30 帧及以上就能达到较好的实时渲染的效果,就能让人觉得较逼真^[7]。在本次测试中,测试了系统在承载三种不同大小的模型时在场景中渲染帧率,模型大小分别是 5 M、10 M、25 M,每隔 10 秒记录一次,结果如表 1 所示。

表 1 渲染帧率记录表

模型大小	序 号								平均值
	1	2	3	4	5	6	7	8	
5M	43FPS	47FPS	47FPS	48FPS	50FPS	47FPS	50FPS	46FPS	47FPS
10M	37FPS	40FPS	39FPS	42FPS	43FPS	41FPS	38FPS	43FPS	40FPS
25M	34FPS	34FPS	36FPS	40FPS	36FPS	33FPS	34FPS	31FPS	35FPS

由表 1 中的数据可以看出,在不同的时刻,渲染帧率在不停的变化,但是都波动幅度不大。总体来说,系统承载 25 M 以内的数据量时,渲染帧率都保持在每秒 30 帧以上,场景渲染效果较好,操作流畅,用户体验感较好。

4 结 论

通过对基于 Web 的 3D 模型浏览与交互系统的研究,介绍了模型信息的导入与加载、导出与上传以及 3D 场景交互和场景的实时渲染等方法。系统还支持解析 3D 动画模型文件,可以支持 dae 格式和 vrmf 格式的 3D 动画模型的播放。在 PC 端或移动端的浏览器中将 3D 数据进行实时渲染,并确保用户在模型操作过程中的各个功能的正常使用,提高系统的交互性以及实用性。

参考文献

- [1] MIDORI T, TAKAHIKO H, SHOJI T. Color naming experiments using 2D and 3D rendered samples[J]. Color Res. Appl. 2015, 40(3): 270—280.
- [2] KRALIK M, ZAKOVA K. Interactive WebGL model of hydraulic plant[J]. IFAC-Papers OnLine. 2015, 48(29): 146—151.
- [3] DIRKSEN J. Learning Three.js: the javascript 3D library for WebGL[M]. Brimingham: Packt Publishing, 2013.
- [4] ANDERSSON S, GORANSSON J, ANDERSSON J G S. Virtual Texturing with WebGL [D]. Chalmers University of Technology, 2012.
- [5] SAWICKI B, CHABER B. Efficient visualization of 3D models by web browser[J]. Computing. 2013, 95(1): 661—673.
- [6] 夏金瑶. 基于 WebGL 的 EAST 可视化系统[D]. 合肥: 中国科学技术大学, 2017.
- [7] 杨润斌, 梁文章, 陈姝. 基于 WebGL 的 3D 动画实时播放系统[J]. 计算机系统应用. 2015, 24(11): 58—63.