

傅开鸿,刘茗溪,李朝奎.基于 WebGL 技术的虚拟地理环境构建方法[J].湖南科技大学学报(自然科学版),2023,38(2):73-79. doi:10.13582/j.cnki.1672-9102.2023.02.010

FU K H, LIU M X, LI C K. Virtual Geographic Environment Construction Method Based on WebGL Technology[J]. Journal of Hunan University of Science and Technology (Natural Science Edition), 2023, 38(2): 73-79. doi: 10.13582/j.cnki.1672-9102.2023.02.010

基于 WebGL 技术的虚拟地理环境构建方法

傅开鸿^{1,2}, 刘茗溪³, 李朝奎^{1*}

(1.湖南科技大学 测绘遥感信息工程湖南省重点实验室,湖南 湘潭 411201;

2.湖南科技大学 地球科学与空间信息工程学院,湖南 湘潭 411201;

3.五凌电力有限公司,湖南 长沙 410004)

摘要:针对当前构建虚拟地理环境过程中存在的冗余度高、环境配置复杂、系统硬件要求高、缺少自然环境模拟和真实性较差等问题,利用 WebGL 技术及 Three.js 框架在 Web 端构建基于 B/S 结构的虚拟地理环境,并以三维虚拟场景为例验证所提方法的可视化效果及可行性.结果表明:综合运用 WebGL 技术在 Web 端构建的轻量化虚拟地理环境具有良好的交互性、共享性和真实性.研究结果有助于实现地理信息的可视化,促进虚拟地理环境在各个领域的广泛应用.

关键词:虚拟地理环境;WebGL;Three.js;交互漫游;自然环境模拟;碰撞检测

中图分类号:P209 文献标志码:A 文章编号:1672-9102(2023)02-0073-07

Virtual Geographic Environment Construction Method Based on WebGL Technology

FU Kaihong^{1,2}, LIU Mingxi³, LI Chaokui¹

(1. Hunan Provincial Key Laboratory of Surveying and Mapping Remote Sensing Information Engineering,

Hunan University of Science and Technology, Xiangtan 411201, China;

2. School of Earth Sciences and Spatial Information Engineering, Hunan University of Science and Technology, Xiangtan 411201, China;

3. Wuling Power Co., Ltd., Changsha 410004, China)

Abstract: For the current process of establishing virtual geographic environment (VGE), there are some problems, such as high redundancy, complex computer environment configuration, high system hardware requirements, lack of natural environment simulation, and poor authenticity. This paper tries to use WebGL technology and Three.js framework to establish a VGE based on browser/server structure on the website. The 3D virtual scene is taken as an example to verify the feasibility of the proposed method which proves that establishing lightweight VGE on the website depended on WebGL technology has great interactivity, shareability and authenticity. The research is helpful to realize the visualization of geographic information and promote the wide application of virtual geographic environment in various fields.

Keywords: virtual geographic environment; WebGL; Three.js; interactive roaming; natural environment simulation; collision detection

收稿日期:2021-08-01

修改日期:2022-09-23

基金项目:国家重点研发计划资助项目(2018YFB0505401);国家自然科学基金资助项目(42171418);湖南省自然科学基金创新群体项目资助(2020JJ1003)

*通信作者,E-mail: chkl_hn@163.com

随着虚拟现实等技术的快速发展,面向复杂地理问题求解和地理分析的虚拟地理环境已成为地理信息科学领域的重要研究方向^[1].虚拟地理环境通过计算机实现对现实世界和虚构环境的映射与重建,是研究人地关系和大规模破坏性地理问题的虚拟实验室,是实现多角色协同研判地学问题的科学工具^[2].地理信息系统技术、大数据技术和“互联网+”技术的高度融合^[3],推动了虚拟地理环境在各行各业中的广泛应用.

在计算机可视化及多媒体技术蓬勃发展的大环境下,网页与 Web 端应用往往需要呈现大量的三维图形,曾经广泛使用的 Flash 技术由于易泄露隐私、进程服务无法彻底终止等特性,用户体验欠佳,Adobe 公司已于 2020 年停止对其的支持,这也促进了 WebGL 技术的迅速发展,使其成为浏览器的标配.WebGL 技术在满足二维图形应用场景快速创建的基础上,充分利用 GPU,在高性能三维应用的构建中也有优越的表现.使用 WebGL 有 3 个明确的优势:(1)无需插件支持.它通过 HTML 脚本本身实现 Web 交互式三维动画的制作,无需任何浏览器插件的支持^[4].(2)标准统一,跨平台.它利用底层的图形硬件加速功能进行图形渲染,并通过统一的、标准的、跨平台的 OpenGL 接口实现^[5].开发人员可以通过 WebGL 技术将客观世界虚拟化,以可视化的方式直观地与用户交互^[6].(3)丰富的第三方库支持.原始的 WebGL 技术创建三维场景并生成动画的过程复杂、耗时,通过第三方库可以改善这个问题,如 Three.js 是 WebGL 中的一个第三方库,提供一套基于 WebGL 的、非常易用的 JavaScript API,仅需少量易读的代码便可在 Web 端构建出丰富的三维场景,使得开发效率较之前有极大提升.

近年来,围绕着虚拟地理环境构建这一研究内容,国内研究者已经提出了很多方案.张颖超等^[7]利用 OpenSceneGraph 技术及其中的粒子系统,在虚拟环境中对雾效、雪效进行仿真,但其中某些粒子属性缺乏专业数据的支持,创建的场景也缺乏光照、阴影等细节处理,真实性欠佳;蒋元等^[8]采用 Unity3D 开发平台构建虚拟地理环境,但其只对地形地貌模拟、地表细粒度仿真和建筑物模型构建等技术进行探讨,并没有对相应的自然环境进行模拟,存在一定局限性;徐丙立等^[9]尝试运用新一代图形引擎 CryEngine 构建虚拟地理环境系统,该系统功能全面,但是对硬件的要求较高,环境配置也比较复杂,共享性欠佳.

针对目前虚拟地理环境构建中存在的不足,本文采用 B/S 结构及 WebGL 技术在 Web 端进行虚拟地理环境的构建,引入一种基于 WebGL 技术及 Three.js 框架的轻量化虚拟地理环境构建方法,详细讨论该方法的关键技术,包括三维地形环境构建、虚拟场景构建、自然环境模拟和碰撞检测等.该方法的试验效果显著,可为智慧城市建设提供相应的技术支撑.

1 WebGL 技术及优势

WebGL 内嵌于浏览器,基于网页端与用户形成一种近似“云交互”的模式,实现在快速调用硬件提高渲染效率的同时,直接在网页端构建复杂三维模型并生成可与用户交互的 3D 图形,且无需额外插件的支持.

WebGL 有两个层面的优势:(1)网络交互基于脚本语言实现,摆脱了传统网页开发过程中对于专用渲染插件的依赖;(2)面向公众开放,是一项开源的技术,可极大地提高效率.由于其直接在底层 GPU 端工作,所以相对于目前其他典型的 Web 技术,WebGL 更为复杂,这也正是它能够迅速地大量计算并实现复杂的实时三维渲染的原因.

同时,与 OpenSceneGraph 和 Unity3D 等 3D 实现技术相比,WebGL 也有很多优点.OpenSceneGraph 是一个三维图形开发包,需通过特定软件驱动程序,运行环境配置较为复杂,利用该技术开发的系统可移植性和共享性欠佳.Unity3D 作为一款三维创作引擎,对计算机硬件的要求较高,且需要有一定经验的人员才可以利用.WebGL 则不存在上述问题,只需一个浏览器即可实现复杂的三维渲染,利用它在网页中构建虚拟地理环境系统即可在联网的任何地方进行访问,便于广大普通用户的使用.

2 三维模型预处理

三维模型构建流程如图 1 所示.所构建的建筑模型主要利用 SketchUp 软件完成,该软件主要应用于室

内设计、房屋建筑、城市规划、土木和机械工程及电影和游戏制作等行业^[10]。在 SketchUp 中,对虚拟场景进行建模,三维模型的网格处于三维空间中,它是由众多的点以及连接的线组成的一个三维模型框架,每一个点都有着各自基于实地观测的坐标(图 1a)。在 SketchUp 中,绘制完所需的点和线之后,会在模型的线框闭合时生成一个平面(图 1b),所有平面都是 SketchUp 默认的面。由于 WebGL 在处理大量运算时的性能稍显不足,在建立模型的过程中应考虑如何使用较少的平面来获得更好的效果。同时,在建模的过程中,表面的方向性也十分重要。Three.js 在渲染的过程中,反向的表面不予渲染,因而需要通过改变表面的方向使物体的外表面都为正面,以避免三维建筑物模型在网页显示中出现缺少墙体的情况。三维模型构建完成之后,需要添加纹理贴图,使其更加真实(图 1c),并以适当格式导出。本文选择.dae 格式,以便在 Web 页面进行可视化呈现。



图 1 三维模型构建流程

同时,虚拟地形的模拟也是虚拟地理环境构建中的一项重要内容。本文利用 ArcScene 对数字高程模型及地表纹理进行编辑,构建出虚拟地形环境,并导入虚拟场景进行利用。

3 虚拟地理环境构建

3.1 基于 Three.js 框架的虚拟场景构建

仅需少量易读的 JavaScript 脚本即可基于 Three.js 框架构建出丰富多样的三维场景^[11]。首先,需创建相应的核心组件,即场景组件、相机组件和渲染器组件。场景是容纳虚拟环境中各个物体的场所,其技术核心是对光源的整体把控以及追踪需要渲染的物体,三维对象必须添加到场景中才能在 Web 端显示;相机是视点的位置,决定了对虚拟场景中哪一个角度的画面进行展现;渲染器负责渲染指定角度下的场景。

核心组件创建完成后,需在场景中加载三维模型,所使用的模型格式为.dae 格式,在 Three.js 中导入.dae 文件的方法如下所示,导入其他格式的文件方法与之类似。

```
var loadingManager = new THREE.LoadingManager( function ( ) {
    elf.castShadow = true;
    elf.receiveShadow = true;
    scene.add( elf );
} );
var loader = new THREE.ColladaLoader( loadingManager );
loader.load( './collada/model.dae', function ( collada ) {
    var material = new THREE.MeshBasicMaterial( );
    collada.material = material;
    elf = collada.scene;
} );
```

此外,模型加载之后还需在场景中添加光源才能使模型显示出来^[12]。Three.js 中有多种光源可以选择: AmbientLight 为环境光,可作为基础光源添加到三维场景中以满足整体到局部的光照需求; PointLight 为点光源,其表现形式为空间一点,可朝所有方向散射光线;除此之外还有 HemisphereLight, SpotLight 等

个性化光源,多用于三维场景中特殊场景的光效模拟,如反光的水面、暗淡的天空等.本文虚拟场景的构建选择 AmbientLight 作为光源.

通过以上流程在 JavaScript 中创建的 Three.js 场景是具有较高自由度的,可根据实际的开发需求进行灵活调整.Three.js 场景创建流程如图 2 所示.

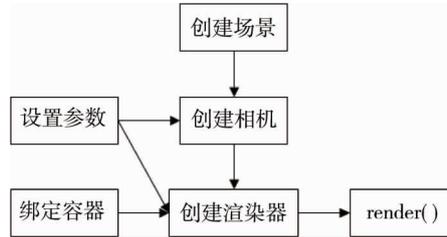


图 2 Three.js 场景创建流程

3.2 实现交互漫游

人机实时交互是虚拟地理环境中应该具有的特征之一.漫游是虚拟地理环境中的一项基础性功能,Three.js 框架提供多种相机控件,主要有轨迹球控件 (TrackballControls)、第一人称控件 (FirstPersonControls)、轨迹控件 (OribtControls)、飞行控件 (FlyControls)、翻滚控件 (RollControls) 及路径控件 (PathControls) 等,这些控件为漫游效果的实现提供了极大的便利,使用这些控件便可以对场景中的相机进行操控.第一人称控件可以第一人称的视角控制相机,由鼠标控制视角变换,角色移动通过键盘来实现^[13].通过第一人称控制器实现虚拟漫游是现阶段比较主流的解决方法,本文也采用这种方法,将第一人称控制器应用到场景中的部分代码如下.

```

var camControls = new THREE.FirstPersonControl( Camera );
camControls.lookSpeed = 0.4;
camControls.movementSpeed = 20;
camControls.noFly = true;
camControls.lookVertical = true;
camControls.consrtainVertical = true;
camControls.verticalMin = 1.0;
camControls.verticalMax = 2.0;
camControls.lon = -150;
camControls.lat = 120;
  
```

至此完成一个基于 Three.js 框架的虚拟场景构建.虚拟场景的 Web 端展示效果如图 3 所示.

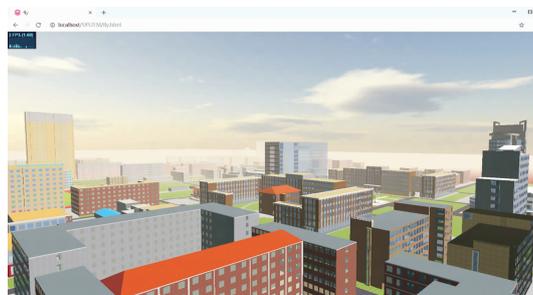


图 3 虚拟场景的 Web 端展示效果

3.3 光照模拟

模拟时间变化也是提高虚拟地理环境真实性的一种有效手段,可用不同时间段的光照效果对其进行展现^[14].对不同时间的光线角度、光照强度以及光线颜色进行模拟,主要使用 DerictionalLight 光源,也可额

外添加 HemisphereLight 光源使得场景更为真实,需对场景进行多次调整才能达到较好的渲染效果.通过设置多个光源实现光源变化,操作时,需先移除一个光源再加载另外一个光源.

光照模拟效果如图 4 所示.由图 4 可知:(1)正午,太阳颜色偏白(0xfffff),同时强度较高(1.0),全局光源强度同样偏高,由于北半球的原因,太阳不是在正上方,即方向向量不是(0,1,0),而是偏向南方,如(1,4,0)(根据模型制作方向更改),效果如图 4a;(2)黄昏,太阳的颜色偏向橙黄(0xffcc33),强度中等(0.6),角度降低,如方向向量为(2,1,0),如图 4b;(3)夜晚,由于没有太阳这个强光源,因此光线颜色发白,甚至泛蓝(0xccccff),光源强度较弱(0.3),全局光源强度降低,如图 4c.若硬件条件好,还可在建筑内部建立光源,模拟灯光.

现实世界中,在用摄像机拍摄光源时,会出现镜头光晕,虽然在实际情况下是尽量避免的,但是在虚拟地理环境中可以通过此手段增加场景真实度.借助相应函数创建出多个模拟光晕的纹理贴图,以增加真实效果.添加镜头光晕通过以下语句引入.

```
flare = new THREE.LensFlare ( texture, size, distance, blending, color, opacity );
```

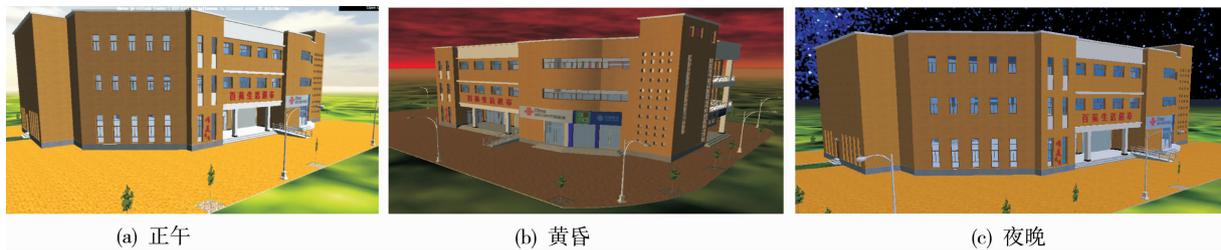


图 4 光照模拟效果

3.4 粒子系统及雨效模拟

在虚拟地理环境的构建过程中,通过对雨、雪等自然现象的模拟,可提高虚拟场景的真实性,增强观察者的沉浸式体验感受,同时可为各类仿真试验提供参考.自然现象的模拟可利用 Three.js 中的粒子系统来实现,粒子系统由大量微观粒子构成,是一种高效的图形生成算法,在快速创建细小物体方面有着优越的表现,如模拟雨滴、雪花等不规则物体,并通过对现实场景中的阻力、重力等进行量化加载,从而营造出逼真的视觉效果.在虚拟地形中的雨效实现如图 5 所示.



图 5 雨效实现

雪效实现与雨效实现类似,雨效实现的核心代码如下:

```
function createRain () {
const box = new THREE.Box3(
new THREE.Vector3(-4000,0,-4000),
new THREE.Vector3(4000,5000,4000)
);
material = new THREE.MeshBasicMaterial( {
transparent: true,
opacity: 0.5,
```

```

map: new THREE.TextureLoader().Load("./image/rain.png"),
depthWrite: false,
});
}

```

天空作为虚拟地理环境中的背景,是增强场景表现力的有效手段.Three.js 框架中对于天空的模拟可以通过添加天空盒(skybox)实现,其核心思想是绘制一个具有纹理的立方体,纹理通常是一些天空、山川等,而后将相机的位置设定于立方体中心,当相机位置改变时,立方体的位置会随之同步变化,这样相机在运动过程中就无法触及场景边缘,这与真实世界中地平线可望而不可即的现象一致.在 Three.js 中添加天空盒的方法如下.

```

scene.background = new THREE.CubeTextureLoader().setPath('textures/cube/skyboxsun25deg/')
.Load(['px.jpg', 'nx.jpg', 'py.jpg', 'ny.jpg', 'pz.jpg', 'nz.jpg']);
var gt = new THREE.TextureLoader().load("textures/light.jpg");
var ground = new THREE.Mesh(new THREE.CubeGeometry(8000,1,8000),
new THREE.MeshPhongMaterial({
color: 0xfffff,
map: gt
}));

```

3.5 碰撞检测

碰撞检测也是虚拟地理环境研究中一项重要的内容,虚拟地理环境需要具有良好的实时性、真实性和交互性,从而为用户提供较好的虚拟场景体验及视觉沉浸感.当观察者的位置在虚拟地理环境中发生变化时,会出现与虚拟场景发生碰撞的情况,为了达到真实性的目标,需在系统中加入交互检测功能,用于检测两个对象之间是否存在碰撞.Three.js 中的碰撞检测主要有直接碰撞检测、代理碰撞检测及基于 RayCaster 和包围盒的碰撞检测^[15].包围盒是在漫游场景中的相机外包装一个不可见的球体或立方体,在渲染过程中随相机一起移动.直接碰撞检测是将需要考虑碰撞情况的场景物体置于待检测数组中,在检测过程中将包围盒直接与场景物体碰撞,该碰撞检测的结果较为准确,但对三维模型的精度要求较高.代理碰撞检测是在需要考虑碰撞情况的场景物体对应位置再建立一个相同的不可见的物体,作为代理检测的对象,该方法在检测准确度方面的优势较为明显,但对于复杂的三维模型,很难建立完全相同的代理对象^[16].为避免前两种方法的缺点,本文采用基于 RayCaster 和包围盒的碰撞检测,其原理为从相机位置向包围盒的每个顶点发出射线,通过遍历射线检测其是否存在与场景中物体顶点相交的情况.若发生相交,即发生碰撞,则计算偏移量并根据偏移量更新相机位置.碰撞检测流程如图 6 所示.

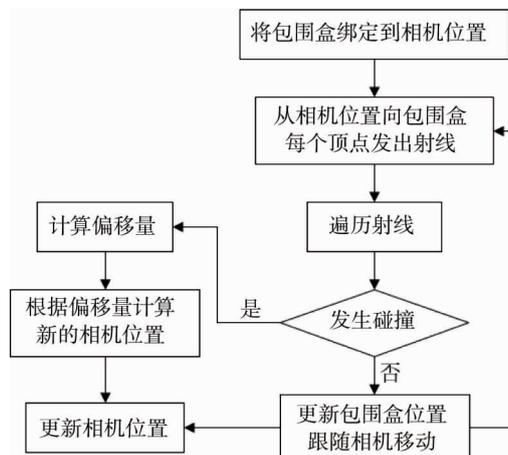


图 6 碰撞检测流程

4 结论

1)通过对原型系统的测试表明,集成 WebGL 技术的系统稳定性较好,运行流畅,与现有的虚拟地理环境系统相比,在真实性、交互性和共享性等方面有显著提升。

2)基于现有成果,可进一步开展对多尺度复杂场景下的路径规划与可行性评估等问题的研究,如应急演练中人群的疏散路径模拟、消防救援路径规划等,以及泥石流、滑坡等自然灾害的模拟。

参考文献:

- [1] 史玮玥,张春晓.国内虚拟地理环境研究综述[J].山东师范大学学报(自然科学版),2019,34(4):426-432.
- [2] 李锋,万刚,蒋秉川,等.虚拟地理环境时空建模及其作战计划推演应用[J].测绘学报,2018,47(8):1072-1079.
- [3] 刘浩,薛梅.虚拟地理环境下的地理空间认知初步探索[J].遥感学报,2021,25(10):2027-2039.
- [4] 刘一臻.基于 WebGL 的三维云景渲染方法分析[J].测绘通报,2020(S1):233-235.
- [5] 赵凯.校园数字植物园平台的设计与实现——以浙江农林大学植物园为例[D].杭州:浙江农林大学,2018.
- [6] 王超.基于移动终端的课堂辅助教学系统的设计与实现[D].湘潭:湖南科技大学,2017.
- [7] 张颖超,邹丽萍,刘佳.基于 OSG 和粒子系统的雪效仿真[J].南京信息工程大学学报(自然科学版),2010,2(2):107-112.
- [8] 蒋元,蔡胜军,何绍勇,等.基于 Unity3D 的虚拟地理环境构建[J].现代计算机(专业版),2014(33):52-55.
- [9] 徐丙立,荆涛,林琿,等.利用 CryEngine 构建虚拟地理环境[J].武汉大学学报(信息科学版),2017,42(1):28-34.
- [10] 方孟元,罗年学,许毅,等.基于 Cesium 的 BIM 与实景三维模型集成可视化研究[J].测绘地理信息,2022,47(2):111-114.
- [11] BALAKRISHNAN J M, CHAVAN S M. Programmers' perspectives on the use of previsualization in the design process of Three.js web animation[J]. American Journal of Art and Design,2022,7(1):1-10.
- [12] 辛全波,秦柳,范俊甫,等.基于 WebGL 的航道数据组织与三维可视化方法[J].测绘地理信息,2022,47(1):81-85.
- [13] 马龙.智能化矿山管控系统中仿真状态与三维展示功能的实现[J].煤炭与化工,2021,44(9):77-79.
- [14] LIU Z, GU X Y, DONG Q, et al. 3D visualization of airport pavement quality based on BIM and WebGL integration[J]. Journal of Transportation Engineering, Part B: Pavements,2021,147(3):04021024.
- [15] 李桢,许华虎.基于 WebGL 的无插件虚拟漫游关键技术的研究[J].计算机应用研究,2020,37(S1):227-229.
- [16] 郭稳稳,仲梁维.网页环境下无插件虚拟场景漫游的研究[J].软件工程,2021,24(6):46-49.