

# 基于 WebGL 的舰船模拟训练虚拟三维技术应用

宁 静, 卜乐平, 冯 源

(海军工程大学 电气工程学院, 武汉 430033)

**摘要:** 由于目前国内的舰船模拟训练系统大多均为岸基模拟器, 和半实物半仿真的应用模式, 受制于该传统方式下系统通用性、可扩展性差, 成本高等局限性; 提出一种基于 WebGL 的舰船模拟训练虚拟三维环境, 采用其应用框架 Three.js 图形引擎库在网页中绘制 3D 虚拟交互界面, 可直接通过浏览器登录系统, 从而给出了可跨平台免插件高效的舰船模拟训练界面; 结果表明该方法在人机交互界面的友好性、监控的直观性、训练操作的便捷性、较强的沉浸性等方面具有优势, 亦可推广应用于其他模拟训练领域。

**关键词:** WebGL; 模拟训练; 虚拟三维; Three.js

## Application of Virtual Three-dimensional Technology on Ship Simulation Training Based on WebGL

Ning Jing, Bu Leping, Feng Yuan

(College of Electrical Engineering, Naval University of Engineering, Wuhan 430033, China)

**Abstract:** In view of the limitation of the poor universality and scalability and high cost for traditional mode of ship simulation training system, virtual three-dimension environment on ship simulation training based on WebGL is presented in this paper. It adopts a graphics engine called Three.js to render 3D virtual interactive interface on the web page, logging into the system by a browser. Thus a cross-platform, free plug-in and efficient interface of simulation training on ships is acquired. This method can be applied to other simulation training field as well.

**Keywords:** WebGL; simulation training; virtual three-dimension; Three.js

## 0 引言

目前国内的舰船模拟训练系统大多均为岸基模拟器, 由于船员经常出海, 去训练基地参加训练班、进修班等培训比较困难等因素的制约<sup>[1]</sup>, 舰载模拟训练系统 (OBTS: On-Board Training System) 应运而生, 船员在闲暇时间就可以在舰船上进行训练操作, 节省了训练时间和大量经费<sup>[2]</sup>。由于训练过程中涉及操控设备种类多, 不同型号的舰船设备型号各异, 操纵平台形式多样, 采用半实物仿真培训成本高, 占地面积大, 在舰船上很难实现。随着计算机技术水平的发展, 成本低、真实感强的舰船虚拟模拟训练系统已成为发展趋势。

虚拟模拟训练系统一般将训练模型装在一台主机中, 通过网络与各训练客户端相连, 受训人员可多人同时在客户端上进行训练。近几年, 随着互联网的发展, 越来越倾向于基于 B/S (Browser/Server) 架构的 Web 应用服务, 因此, 原有传统的 C/S 体系架构下的单机版二维人机交互模式, 由于其不可跨平台, 友好性等不足, 如何使其能在浏览器端与设备的三维模型直接进行交互成为了目前的需求。尤其是 WebGL 标准的提出, 将浏览器和 GPU 紧密的关联在一起, 使得在 Web 上呈现三维图形成为了可能, 使模拟训练系统有了新的应用模式<sup>[3]</sup>。应用过程中需要将设备的三维模型与操作界面的二维图形相叠

加, 结合形成一个友好的界面, 然而, 目前国内现有的基于 WebGL 在舰船模拟训练的研究还比较少见, 还是一个新的尝试。

本文基于 WebGL 的 Three.js 三维渲染引擎技术, 在船舶虚拟三维技术方面作出了一些探讨研究, 并在舰船虚拟三维呈现软件中进行了初步应用, 避免了以往 Web3D 应用中, 插件/专有客户端软件安装所带来的安全及用户体验差的问题, 可直接通过浏览器访问, 在人机交互界面的友好性、监控的直观性、训练操作的便捷性、较强的沉浸性等方面具有优势。

## 1 WebGL 及 Three.js

WebGL 是一项基于浏览器 (而不是基于操作系统) 的, 用以在浏览器中绘制、显示三维计算机图形, 可以在多种平台上运行, 并为之交互的技术。2009 年 8 月由 Khronos 提出的一种 3D 绘图标准, 这种绘图技术标准结合了 HTML5 和 JavaScript, 通过为 OpenGL ES 2.0 增加一个 JavaScript 绑定, 将 JavaScript 和 OpenGL ES 2.0 结合在一起, 从而为 HTML5 中 Canvas 提供了硬件 3D 加速渲染<sup>[4]</sup>。WebGL 程序的基本渲染过程如图 1<sup>[5]</sup>。

WebGL 原生 API 是一种十分低等级的接口, 直接来写 3D 程序不仅开发难度大, 并且需要对 OpenGL、三维设计以及 WebGL 的原生 API 非常了解, 从而导致开发效率低下, 因此需要将常用的底层模块进行隐藏封装, 从而使用户在无需了解底层细节的情况下, 快速开发 Web 交互式三维图形应用。目

收稿日期:2016-04-11; 修回日期:2016-05-24。

作者简介:宁 静(1992-), 女, 陕西西安人, 硕士研究生, 主要从事信号检测与处理方向的研究。

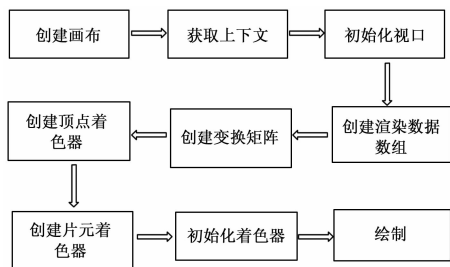


图 1 WebGL 基本渲染过程

前, 已经有不少 WebGL 框架开发出来: Three.js、PhiloGL、Babylon.js、O3D 等, 这些引擎库能够让用户更为方便地进行 3D 图形绘制和动画的制作。本文选用 Three.js 实现, 它源自 GitHub 的一个开源项目, 发展相当迅速, 现在已经发展到 r73 版了, 是一个轻量级的用于在浏览器中创建 3D 计算机图形图像应用程序的 JavaScript 库, 也是目前应用最广泛的 WebGL 框架<sup>[6]</sup>。

所有 WebGL 框架在使用时都需先定义三维世界中所包含的基本元素<sup>[7]</sup>, Three.js 库中包括了场景、相机、模型加载器、着色器、材质、灯光、动画、粒子、物理工具等, 其中, 场景 (scene)、对象 (object3D)、相机 (camera)、渲染器 (render) 是基础。有了这些东西, 才能将物体渲染到网页中去。Three.js 的绘图流程如图 2 所示。

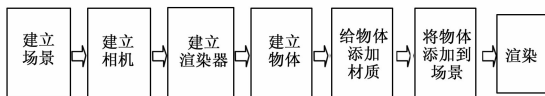


图 2 Three.js 基本绘图流程

场景是所有物体的容器, 如果要显示一个物体, 就需要将对象加入场景中。在 Three.js 中场景只有一种, 用 THREE.Scene 来表示, 代码如下:

```
var scene = new THREE.Scene();
```

相机决定了场景中哪个角度的景色会显示出来。相机就像人的眼睛一样, 人站在不同位置, 抬头或者低头都能够看到不同的景色。与场景不同的是, 相机有很多种。和现实中一样, 不同的相机确定了呈相的各个方面, 设置不同的相机参数, 就能够让相机产生不一样的效果。在 Three.js 中有多种相机, 本文使用的是透视相机 (THREE.PerspectiveCamera)。渲染器决定了渲染的结果应该画在页面的什么元素上面, 并且以怎样的方式来绘制。渲染器 renderer 的 domElement 元素, 表示渲染器中的画布, 所有的渲染都是画在 domElement 上的, appendChild 表示将这个 domElement 挂接在 body 下面, 这样渲染的结果就能够在页面中显示了。渲染应该使用渲染器, 结合相机和场景来得到结果画面<sup>[8]</sup>。

## 2 系统设计

虚拟三维模拟训练系统总体上分为前端三维场景、二维界面, 和后台电力系统仿真, 以及中间的数据交互, 如图 3 所示。其中图像交互界面是交互接口的重要组成部分, 本文基于 WebGL 的舰船虚拟三维技术, 采用 Three.js 引擎, 重点研究系统前端人机交互界面。系统架构采用 B/S 模式, 使用户端不需要安装任何应用程序, 只要打开浏览器即可登录系统进行

相应操作。

目前交互界面主要可分为二维和三维。前台二维, 三维界面主要向受训者提供直观的训练界面和演示平台, 用户的所有操作基本在上面完成, 所以该界面一是要满足受训者的训练需求, 这也是人机交互界面的最基本的要求; 二是要求界面能和人的思维模式相一致性; 三是要求界面易用性。

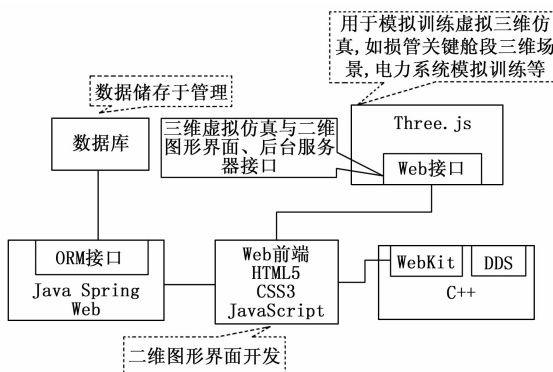


图 3 系统体系结构

舰船三维模型界面逼真的再现了整个船体的空间结构, 鼠标拾取可通过拖拽来旋转船体, 鼠标滑轮则可以根据用户需求放大或缩小船体, 通过漫游和后台数据交互, 将电力系统所涉及到的参数在二维版面真实的显示出来。二维界面主要包括设备参数栏, 控制设置栏, 三维界面显示窗, 性能监视器, 小地图。

## 3 系统实现

### 3.1 基本环境的设置与搭建

为确保浏览器能够正确运行 WebGL 程序, 首先需要对浏览器进行设置, 以开启对 WebGL 的支持, 本文使用 Firefox 浏览器, 如图 4 所示。双击将 webgl.force-enabled 的值设置为 true, 将 webgl.disabled 的值设置为 false。

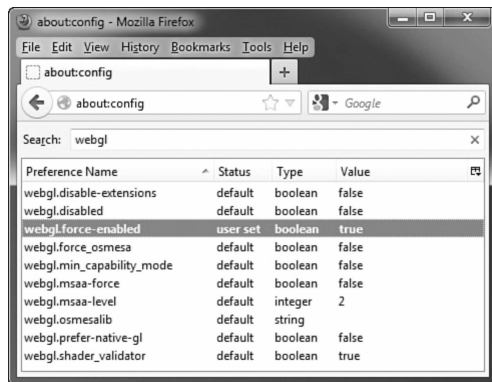


图 4 Firefox 下 WebGL 相关的设置

在渲染器的设置中, 本文选取 Three.js 中渲染效果较好的 WebGLRenderer; Three.js 中为用户提供了正交相机和透视相机这两种相机类型, 本文采用透视相机, 可以逼真地实现用户体验; Three.js 中用 THREE.Scene () 来设置三维场景; 用 THREE.AmbientLight ()、THREE.SpotLight () 来设置 Three.js 中提供的 4 种光源其中的环境光和点光源。

### 3.2 三维模型的加载

大部分三维程序都是从模型文件中读取三维模型的顶点坐标和颜色数据等相关信息,而当所需建立的模型结构很复杂时,可以在专门的建模软件中进行绘制,其中模型文件便是由三维建模软件生成的。本文中模型文件由三维建模软件 Blender 生成,后导入 Three.js 引擎中。Blender 是一款非常流行的建模软件,可以将三维模型导出为 OBJ 格式,该格式是基于文本的,易于阅读和理解,也易于转化为其他格式,导出时将模型的材质信息存储在外部的 MTL 格式的文件中。

3D 世界是由点组成,两个点能够组成一条直线,3 个不在一条直线上的点就能够组成一个三角形面,三维物体是由多个三角形或四边形组合而成的网格模型(在 Three.js 中,用 THREE.Mesh 表示),我们通常把这种网格模型叫做 Mesh 模型。THREE.Mesh=function(geometry, material) 为其构造函数,其中,第一个参数是一个 THREE.Geometry 类型的对象,表示模型定点间的连接关系;第二个参数定义了材质,即表面的样式,有可能是单色或渐变色,也有可能贴有纹理。

在代码中,最主要的是对加载的三维模型文件进行解析,只有解析了的模型才能在后面的代码中使用。模型解析完成后,按 4 步调用 geometry,依次为:计算 geometry 中每一个面的重心 geometry.computeCentroids();计算每一个面归一化后的法向量 geometry.computeFaceNormals();计算一个可以包围 geometry 的椭圆 geometry.computeBoundingSphere();最后得到完整的 geometry 对象。MTL 文件按同样的过程最终得到完整 material 对象后,新建 mesh 变量引入 geometry 和 material;

```
var mesh=new THREE.Mesh(geometry, material)
```

### 3.3 性能评估

测试一个程序,性能上是否有瓶颈,在 3D 世界里,经常使用帧数的概念。帧数即图形处理器每秒钟能够刷新几次,通常用 fps (Frames Per Second) 来表示。人眼看物体时,成像于视网膜上,并由视神经输入人脑,感觉到物体的像。一帧一帧的图像进入人脑,人脑就会将这些图像给结合起来,形成动画。帧数越高,画面的感觉就会越好。

在 Three.js 中,性能由一个性能监视器来管理,其中 FPS 表示上一秒的帧数,这个值越大越好,点击该图就会变成右侧视图,如图 5 所示。MS 表示渲染一帧需要的毫秒数,这个数字是越小越好,再次点击又可以回到 FPS 视图中。



图 5 性能监视器

### 3.4 交互处理

在系统中,实现三维空间与用户的交互一般是通过鼠标、键盘操作,而鼠标、键盘所触发的事件是通过添加事件侦听实现,代码如下:

```
document.addEventListener("事件名称",函数,false);
function 函数(event){//函数体}
```

本文中用户是通过拖动(即按住鼠标左键移动)鼠标旋转三维船体,通过鼠标滚轮实现船体的缩放。鼠标的响应事件名称有 4 个,分别为:mousemove、mousedown、mouseup、mousewheel。其中,mousemove 与 mousedown、mouseup 相配合,分别完成相应移动动作,而 mousewheel 滚轮事件触发函数实现镜头的缩放。在 Firefox 浏览器中效果如图 6 所示。

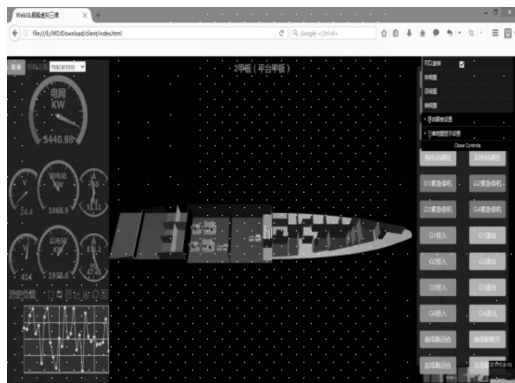


图 6 基于 WebGL 实现舰船虚拟三维场景图

## 4 结语

本文将 WebGL 这一新兴 Web 技术应用于舰船模拟训练,采用了 B/S 结构,受训者在客户端只需要打开浏览器即可登录系统,省去了繁琐的客户端程序安装过程,克服了跨平台和插件所带来的不兼容性。利用基于 WebGL 技术的 Three.js 引擎构建三维交互平台并加载三维模型,实现鼠标对网页三维场景中的模型进行互动,为传统二维系统模式带来了新的思路,从视觉上更逼真的模拟真实的舰船环境。目前,国内关于 WebGL 的研究和探讨还是比较少的,可以说是比较前沿的技术,而 WebGL 标准仍在进一步的发展之中,网络化的交互训练系统还需要进一步的完善。

### 参考文献:

- [1] 张锦荣. 推进海军模拟训练跨越式发展的几点思考 [J]. 军事运筹与系统工程, 2010, 24 (1): 44-47.
- [2] Marine Systems: On-Board Training System [EB/OL]. [http://www.cae.com/L3\\_MAPPS/Products\\_and\\_Services/Marine\\_Systems/Naval\\_Systems/obts.pdf](http://www.cae.com/L3_MAPPS/Products_and_Services/Marine_Systems/Naval_Systems/obts.pdf).
- [3] 刘爱华, 韩 勇, 张小垒, 等. 基于 WebGL 技术的网络三维可视化研究与实现 [J]. 地理空间信息, 2012 (5): 79-81.
- [4] Kouichi Matsuda, Rodger Lea. WebGL Programming Guide: Interactive 3D Graphics Programming with WebGL [M]. Addison-Wesley Professional, 2013.
- [5] WebGLSpecification [Z/OL]. <https://www.khronos.org/registry/webgl/specs/1.0/>.
- [6] 王腾飞, 刘俊男, 周更新. 基于 Threejs3D 引擎的三维网页加密与实现 [J]. 企业技术开发, 2014, 33 (2): 79-80.
- [7] 荣艳冬. 基于 WebGL 的 3D 技术在网页中的运用 [J]. 信息安全与技术, 2015: 90-92.
- [8] Dirksen J. Learning Three.js: The JavaScript 3D Library for WebGL [M]. Packt Publishing Ltd, 2013.