

# 基于Three.js的机械产品自动装配演示

侯严庭

(上海理工大学机械工程学院, 上海 200093)

**摘 要:** 随着前端展示技术的发展,HTML5和WebGL技术在机械领域的应用正逐渐增多,这为机械产品在网页上的装配展示提供了新的解决途径。为了避免网页展示所需的渲染插件的麻烦,提出了基于WebGL的第三方类库Three.js的装配展示平台的构建方法。本文对网页上实现简单三维模型展示流程和机械产品网页装配的功能进行了分析,显示了WebGL的开放性、免插件、跨平台、硬件加速的优势。在网页上实现机械的装配,方便了用户的观看和操作,取得了良好的效果。

**关键词:** HTML5; Three.js; 无渲染; WebGL; 装配

**中图分类号:** TP311 **文献标识码:** A

## Automatic Assembly Demonstration of Mechanical Products Based on Three.js

HOU Yanting

(Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

**Abstract:** With the development of front-end display technology, using HTML5 and WebGL technology in the field of machinery is gradually increased, which provides a new way for the assembly of mechanical products demonstrated in web pages. In order to avoid the trouble of the rendering plug-in needed for the web page display, the construction method of the assembly display platform of the third party class library Three.js based on WebGL is proposed. This paper analyzes the function of displaying simple 3D models on web pages and functions of web page assembly of mechanical products, and shows the advantages of open, plug-in free, cross-platform, hardware-acceleration of WebGL. The realization of mechanical assembly on the web facilitates the viewing and operation of the users, and has achieved good results.

**Keywords:** HTML5; Three.js; no rendering; WebGL; assembling

## 1 引言(Introduction)

随着CAD/CAM技术在制造业领域的应用逐渐扩大,产品的设计效率得到了有效的提高,同时也提高了生产过程的自动化水平。但是,产品的装配环节还存在着一些问题,因为此环节与对于技术人员知识及经验的积累要求较高,这一约束使制造业自动化的发展受到了极大的制约。装配设计的性能和保证产品质量和可靠性,降低产品成本,提高竞争力具有重要意义<sup>[1]</sup>。随着三维画图软件技术的发展,技术人员开始在复杂的CAD应用软件(如Pro/E, UG, Solidworks, Creo等)上进行机械产品的虚拟装配,此种装配方式存在着一些问题。例如:不同软件对于电脑的配置要求不同、不同软件所画的图之间不能互相利用、所需要生成的装配图文件存储

量过大、在向客户展示时对软件依赖性强,不方便操作、对技术人员的专业素质要求高等。寻求一种操作简便、开放性高、可跨平台且对硬件要求低的装配展示方式,成为了研究人员迫切需要解决的问题<sup>[2]</sup>。

随着前端网页技术的迅速发展,在网页上实现产品的展示技术逐渐受到了人们的青睐。基于新标准的HTML5和WebGL技术网页展示方法,使上述难题得到了一定的解决。在网络环境中,用户只需将每个机械零部件保存为单独的文件,通过JSON格式的语言将零件的相关信息反馈给浏览器,借助于浏览器就可以很清晰的了解三维的机械产品的装配顺序,操作简单方便且对三维绘图的软件依赖性低。本文阐述了基于Three.js的机械产品自动装配的动画演示平台,使得用

户能够在客户端可以体会到一些三维模型逼真的装配效果,从而为实物装配提供一定的指导,具有很好的应用价值。

## 2 关键技术(Key technology)

### 2.1 虚拟装配应用现状的研究

机械产品的装配是在一定的精度和技术要求下,将一组零散且无序的零件按照最终所要得到产品的要求进行组合的过程。产品零件装配的序列规划则成为其无法绕开的核心内容,其对产品质量有着决定性的作用。许多学者通过对装配过程的大量研究,提出了自动化途径的方法完成装配,这种方式只能依靠产品的几何拓扑结构进行推理,缺乏装配过程因素和操作因素,易陷入组合爆炸问题<sup>[3]</sup>。据统计,在产品的生产过程中大约1/3以上的人直接或间接从事与装配有关的活动<sup>[4]</sup>,而产品的装配费用所占的成本,大约为整个生产成本的30%—50%(如果产品比较复杂,这个比例则会更高)。除此之外,产品的自动化生产往往被装配环节所制约,产品是否能够实现自动化装配的关键在于其装配性能的好坏。因此,一个合理的装配顺序规划和工艺规划,对于产品设计的改进、成本的降低、产品生产周期的缩短具有重要意义。

在产品的开发过程中,通常传统的做法都是在借助实物模型的基础上来对产品的装配性能进行分析和评价,此种方法不但花费更多的时间、浪费更多的精力、有时候准确性也很差,而且又不能根据需要及时、快捷地进行修改,使产品的生命周期有所延长,浪费了财力和物力。针对上述问题,研究人员提出了虚拟装配技术,此种技术为解决产品的装配问题提供了新的途径。工程师可以在虚拟环境下完成产品的装配过程。这样,工程师在装配过程中可以融入自身经验的同时,又可以用产生的装配信息完善产品的装配规划,然后进行整理归纳,将经验的形成规则存储到知识库。

对于机械产品的规划路径演示,有技术人员提出可通过真人的现场演示来展现,也有人提出可以以视频的形式展现出来,但这些都存在着一些问题,人员的表达方式是否完整、视频的演示画面是否清晰等。基于three.js的网页展示技术可以使上述问题得到一定的解决。在网络环境中,用户只需借助于浏览器就可以对三维的机械模型的装配流程有清晰的了解。

### 2.2 以HTML5标准为载体呈现三维图像

在没有HTML5标准之前,并不是所有版本的浏览器都支持HTML、CSS和JavaScript,常常因为打开浏览器的版本不同而效果不同<sup>[5]</sup>。HTML5通过对以前的技术进行改进和总结,制订出了新的标准,并且在原来的基础上添加了一些新

的功能,以便打造出一种内容丰富、操作简捷、免插件且跨平台性更好的产品,它可以在各系统平台上实现的无缝链接与交互<sup>[6]</sup>。从目前来看,效果还是相当不错的,部分主流浏览器都已支持HTML5,例如,Firefox、Opera、Chrome、IE9+等<sup>[7]</sup>。总体来说,HTML5有下面的优点<sup>[8]</sup>:

- (1)加入了几个新的Tag,使得浏览器表现更加丰富。
- (2)使得在浏览器中实现了视频与音频的展示。
- (3)取代了Silverlight与Flash,不用任何插件可以在客户端实现它们的功能。
- (4)提高了用户体验。
- (5)将会在网页三维游戏与移动设备应用中发挥重要作用。

### 2.3 基于WebGL的画面渲染

WebGL本身是一套Javascript API,也是一种3D绘图标准,可以在兼容WebGL的浏览器中对交互式的图形场景进行渲染。WebGL可以通过Web页面的Canvas标签,完全集成在支持GPU(Graphics Processing Unit)加速的Web浏览器中。在Web应用程序中WebGL的工作模式如图1所示。

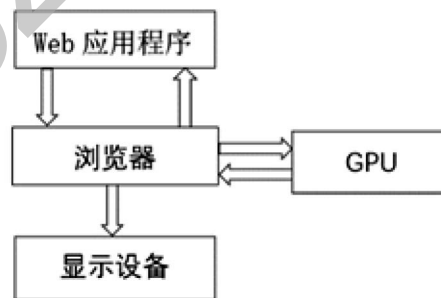


图1 使用WebGL的Web应用程序的工作模式

Fig.1 The work mode of using WebGL's Web application

依据WebGL规范,对底层的WebGL图形接口进行简单的封装,形成了更出色的框架—Three.js。通过掩盖细节来加快产品的开发效率,降低了开发者的负担。在各版本的浏览器3D效果方面,它的表现和兼容性能也很出色。Three.js通过创建所需的基本元素,如点、线、面、矩阵等,通过将相机(Cameras)、物体(objects)、光线(lights)等添加到场景(Scene)中,借助渲染器(renderer)进行场景渲染,即可实现基本的网页三维模型展示<sup>[9]</sup>。

## 3 基于Three.js的机械自动装配展示系统的创建 (Establishment of automatic mechanical assembly and display system based on Three.js)

### 3.1 场景设置

场景实际上相当于一个三维空间,后续添加的物体模型

都是添加到场景中去,它相当于一个大的容器。一般来说,场景的建立十分简单,只需在程序最开始的时候进行实例化,然后就可以将物体模型添加到场景中。其格式如下:

```
var scene;
function initScene() {
    scene=new THREE.Scene();
}
```

### 3.2 摄像机设置

Three.js中,相机依据三维空间中的物体投影到二维空间方式的不同,将投影分为两种方式:透视投影和正投投影。透视投影和日常生活中人眼观看物体的原理是一致的,就是视点越近的物体看起来越大,当物体离得较远时看起来会较小;正投投影则是不管视点和物体有多远的距离,所要绘制的物体都按照统一的大小进行。另外,相机用的是右手坐标系工作原理,使用时需要设置位置坐标和视野中心坐标。其格式如下:

```
var camera;
function initCamera(){
    camera=newTHREE.PerspectiveCamera(40,window.
    innerWidth/window.innerHeight,1,100);
    camera.position.set(-5.00,3.43,11.31);
    camera.lookAt(new THREE.Vector3(-1.22,2.18,4.58));
}
```

### 3.3 光源设置

为了优化场景中的视觉效果,可添加适当的光源来展现相应的主题和效果。为了模型在放大、缩小、旋转时更好的显示,则需要多种光源的结合。在一个场景中可以有多个光源,其基本上都是环境光和其他几种光源的组合。其格式如下:

```
var pointLight;
function pointLight{
    pointLight=new THREE.PointLight(0xffffff,1);
    pointLight.position.copy(camera.position);
    scene.add(pointLight)
};
```

### 3.4 渲染器设置

渲染器就是在已经定义好的场景中,通过添加好相机之后,调用渲染函数使存在于三维空间里的模型以二维平面的形式显现出来的功能。其格式如下:

```
var renderer;
```

```
function initThree(){
    renderer=new THREE.WebGLRenderer({antialias:true });
    renderer.setClearColor(0x000000,1);
    renderer.setPixelRatio(window.devicePixelRatio);
    renderer.setSize(window.innerWidth,window.
    innerHeight);
    container.appendChild(renderer.domElement);
}
```

### 3.5 物体模型导入及自动装配的实现

场景中的模型可以是Three.js中自带的规则模型,也可以是通过3D建模工具导出的模型文件。HTML脚本本身自带一些简单的规则三维模型,如正方体,球等。由于机械装备一般都比较复杂,用简单的几何体组合的方式很难将其表现出来。基于此,可以先通过3dsMax、PRO/E、UG和SolidWorks制作的三维模型,再将其通过软件转化成浏览器可加载的文件格式,添加到场景中。Three.js自带了加载json的方法,所以不需要额外引用插件。现在的JSON格式有两个类型,一种需要JSONLoader加载;另一种需要ObjectLoader加载。本次实验采用3dsmax把模型格式转换成obj,再导入blender处理好模型后导出json文件。其引用的格式如下:

```
new THREE.ObjectLoader().load('models/json/
pump/pump.json',function (model) {
    scene.add(model);
    mixer=new THREE.AnimationMixer(model);
    mixer.clipAction(model.animations[0]).play();
    animate();
});
```

JSON文件中的模型的基本信息描述格式如下:

```
"geometries":[{"
    "uuid":"CFB2696C-B6EA-4FCA-96E3-
EA3C959F148C",
    "type":"Geometry",
    "data":{
        "vertices":[-0.46194,1.25,-0.191341...];//顶点数
        "normals":[-0.92388,0,-0.382683...];//顶点法线向量
        "uvs":[[0.603154,0.532581,0.607054...],[...]];//uv
        "faces":[42,0,1...];//threejs内的类型,存储了顶点
        vertices的索引
    }
}]
```

```

"materials": [{
  "uuid": "686b7161-539c-4515-9ee1-90bad1383a69",
    "type": "MeshLambertMaterial",
    "name": "Blue_S",
    "map": "7e397f39-0de8-45b2-88fd-7006e65ff9e5", //漫射贴图
    "color": 16777215
  }
]
各零件出现的时间和轨迹描述格式:
"animations": [{
  "name": "default", //动画名称
  "fps": 24, //帧频
  "tracks": [{
    "type": "vector3",
    "name": "camTrick_G.position",
    "keys": [{ //动画帧, 每一帧都有当前骨骼的信息
      "value": [-77.460685, -1.56741, -60.086754],
      "time": 1 //当前帧的时间
    }, {
      "value": [-68.295416, -1.56741, -66.434409],
      "time": 156
    }, {
      "value": [-68.562223, -1.482206, -62.153825],
      "time": 157
    }, {
      "value": [0, 0, 0],
      "time": 170
    }
  ]
}]

```

该模块的工作流程及自动装配展示技术的实现, 如图2所示。

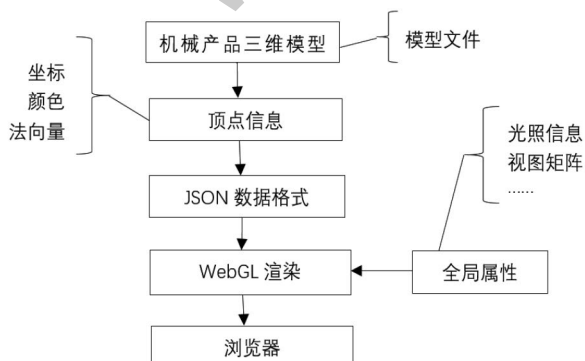


图2 实现流程图

Fig.2 Implementation flow chart

其最终效果如图3所示。

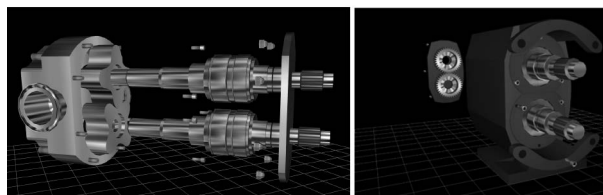


图3 效果图

Fig.3 Effect diagram

#### 4 结论(Conclusion)

通过应用 WebGL 技术, 调用 Three.js、Detector.js、TrackballControls.js、stats.min.js 等加载器, 借助 STLLoader.js 来导入外部建好的模型; 通过 Three.js 自带的渲染器(Renderer)、场景(Scene)、照相机(Camera)、光源(Light)和着色器功能设置, 使得 3D 模型在网页中能够给人以真实绚丽的 3D 效果; 用 TrackballControls.js 实现鼠标的操作功能, 在通过 JSON 语言给出物体的颜色、坐标、开始出现时间、装配的时间等实现网页自动装配功能。

#### 参考文献(References)

- [1] Xinping Yu, Peng Gu, Xia Liu, et al. Modeling technology of virtual assembly system based on UML[J]. International conference of Electronics, Communication and Aerospace Technology (ICECA), 2017(2): 722-726.
- [2] 杜培富. 基于 VRML 的机械产品虚拟装配技术研究[D]. 中国石油大学, 2008.
- [3] 蒋蘋, 何清华, 王奕. 基于 VRML 和 Java 的交互式虚拟装配技术研究[J]. 机械研究与应用, 2008, 21(06): 96-99.
- [4] 张隽, 翟正军. 虚拟装配技术的研究与应用现状[J]. 航空制造技术, 2009(01): 70-73.
- [5] 毕晓明. 简析 HTML5 在移动互联网开发中的应用[J]. 软件工程, 2016, 19(02): 41-42.
- [6] Du Pengyu, You Song, Li Deng. A real-time collaborative framework for 3D design based on HTML5, International Conference Computer Supported Cooperative Work in Design (CSCWD), 2016: 215-220.
- [7] 刘新星. 基于 HTML5 的空间环境数据三维成像研究与应用[D]. 电子科技大学, 2013.
- [8] 霍冬, 郑伟华, 盛步云. 基于 WebGL 的机械产品三维展示技术研究[J]. 制造业自动化, 2013, 35(18): 73-77.
- [9] 汪升华, 唐国纯. 基于 HTML5 的三维思维导图软件开发技术研究[J]. 软件工程, 2017, 20(10): 4-7.

#### 作者简介:

侯严庭(1992-), 男, 硕士生. 研究领域: 计算机辅助设计.