文章编号:1671-4598(2015)07-2508-03 DOI:10.16526/j.cnki.11—4762/tp.2015.07.079 中图分类号:TP315 文献标识码:A

运载火箭试验数据管理及可视化系统设计

毛一春,谌德荣
(北京理工大学 机电工程与控制国家级重点实验室,北京 100081)

摘要:针对现有运载火箭试验数据管理及可视化系统依赖于第三方功能插件,未实现纯浏览器端三维模型渲染及交互,系统对运行环境依赖性强、部署和维护成本高等问题,设计了基于JavaScript框架ExtJS和WebGL框架three.js的运载火箭试验数据管理及可视化系统;首先基于B/S架构设计了系统总体框架,然后针对二维图形数据传输延迟问题设计了基于Web Worker的数据读取方案,最后针对浏览器端三维模型渲染需求提出了基于VTK模型的转换算法,设计了三维模型渲染及交互方案;试验结果显示,该系统不依赖于运行环境和功能插件,能实现浏览器端二维图形和三维模型渲染及流畅交互,为兼容国产化操作系统、降低部署和维护成本创造了条件。

关键词:运载火箭;数据可视化;ExtJS;three.js

Design of Launch Vehicle Test Data Management and Visualization System

Mao Yichun, Chen Derong
(State Key Laboratory for Mechatronics and Control, Beijing Institute of Technology, Beijing 100081, China)

Abstract: The existing launch vehicle test data management and visualization system rely on third-party plug-ins, it can't realize rendering and interaction of three-dimensional model in browser and rely on runtime environment. It can be very cost to deploy and maintain the system. A new launch vehicle test data management and visualization system is designed based on JavaScript framework ExtJS and WebGL framework three.js. First, the overall framework is designed based on B/S architecture, then the data read sheme is designed based on Web Worker to solve the problem of 2d graphic data transmission delay. The test results show that the system doesn't rely on the runtime environment and plug-ins. It can realize rendering and interaction of three-dimensional model in browser smoothly, be compatible with our country homebred operating system and reduce the costs of deployment and maintenance.

Keywords: launch vehicle; data visualization; ExtJS; three.js

0 引言

随着载人航天和深空探测技术的不断发展,航天任务的规模和复杂度不断增加,运载火箭试验数据呈现出数据结构复杂、数据量大、数据种类多、数据处理方法各异等特点[1]。如何利用现代信息技术对海量、多维、动态的试验数据进行有效管理、分析及可视化,帮助试验分析决策人员提高感知理解信息的能力[2],成为航天领域需要重点研究的问题之一。

现有的试验数据管理及可视化系统架构分为C/S架构(基于客户端)和B/S架构(基于浏览器)。前者基于客户端实现试验数据管理、分析及可视化,实时交互性好,但对运行环境依赖性强,部署和维护成本高。后者基于浏览器实现数据管理并调用Matlab或STK等第三方软件或插件实现试验数据分析及可视化[3],对运行环境依赖性低,但需调用第三方软件或插件。论文基于B/S架构,不依赖于插件,同时兼顾C/S架构的功能性和实时交互性,采用JavaScript框架ExtJS和WebGL框架three.js,面向现代浏览器,设计了运载火箭试验数据管理及可视化系统,可实现浏览器端二维图形绘制和三维模型渲染及交互,不依赖于运行环境且无需第三方功能软件或插件支持,可兼容国产化操作系统,部署和维护成本低。

收稿日期:2014-11-13; **修回日期**:2015-01-09。

作者简介:毛一春(1990-),男,江西鄱阳人,硕士研究生,主要从事计算机图形学、数据可视化方向的研究。

谌德荣(1966-),女,湖南常德人,博士,教授,主要从事图像压缩及图像处理技术方向的研究。

1 系统总体设计

1.1 系统架构设计

运载火箭试验数据管理及可视化系统用于对运载火箭飞行试验获取的各专业试验数据进行统一化平台管理,提供数据存储、检索、导出等功能,实现二维图形及三维模型的渲染及交互。系统采用B/S架构[4]进行搭建,系统架构如图1所示,浏览器端采用JavaScript框架ExtJS设计用户界面及交互功能模块、d3.js框架和three.js框架分别设计二维图形绘制和三维模型渲染及交互模块;服务器端采用Apache服务器提供web服务,MySQL数据库进行数据存储,脚本语言PHP设计数据存储管理模块。浏览器端和服务器端之间通过超文本传输协议(HTTP-Hypertext transfer protocol)进行数据通信,同时采用Ajax和Web Worker技术,实现服务器端和浏览器端之间数据异步通信,提高数据传输和图形绘制效率。

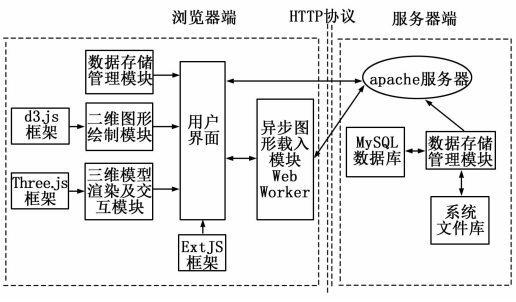

图1 系统架构图

1.2　功能模块设计

运载火箭试验数据管理及可视化系统分为数据存储管理模块、二维图形绘制模块、三维模型渲染及交互模块，各模块主要功能简述如下：

1）数据存储管理模块：根据用户需求完成试验数据的存储、检索及导出，该模块分为服务器端部分和浏览器端部分，服务器端部分负责试验数据的存储及筛选，浏览器端部分提供界面供用户浏览数据列表、检索及提取数据。

2）二维图形绘制模块：运载火箭飞行试验数据包含的二维结构数据种类多、数据量大，每路通道均包含上万组二维结构数据，同时需提供多路同步显示、图形连续播放等功能，用户交互频繁，论文设计了二维图形绘制以及图形数据传输方案。

3）三维模型渲染及交互模块：支持运载火箭三维模型渲染及交互操作，便于试验人员查看火箭不同空间位置的试验数据。系统采用 WebGL 框架 three. js 通过 JavaScript 脚本本身实现浏览器端三维模型渲染及交互，无需任何插件支持，并支持图形硬件加速以实现三维模型的流畅交互。

2　系统方案设计

2.1　数据存储管理

系统所管理试验数据种类多、数据量大、数据结构各异，若采用数据库存储方式，则需将各类数据进行格式转化，且不利于数据传输及图形绘制，若采用文件系统存储方式，则不利于数据信息管理及数据检索。

系统采用结构化存储和非结构化存储相结合的方式，结构化存储是将试验数据以记录的方式存储在关系型数据库中，便于查询及可视化，非结构化方式是将试验数据以文件形式存储在系统文件库中，便于试验人员导出。

为方便试验人员快速查询利用试验数据，系统提供强大的数据检索和提取功能，采用 Ajax 技术实现数据检索与用户界面局部更新，保证了用户交互的连续性，数据检索框架如图 2 所示。

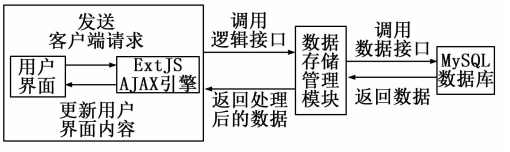

图 2　数据检索框架

2.2　二维图形绘制及图形数据传输

系统采用 d3. js 可视化库实现各专业试验数据的二维图形显示，并提供图形缩放、统计量显示、多路同步显示、图形连续播放、图形保存等功能。用户在浏览器端选择需观测的试验数据，服务器端接收数据请求并与数据库交互，筛选所需试验数据返回给浏览器端进行绘制。但 B/S 架构与传统桌面应用程序不同，服务器端和浏览器端之间的数据通信需耗费时间和系统资源，若按照一般的请求等待方式发送数据，会造成用户长时间操作等待，影响用户的交互体验。

本系统设计了 Web Worker 多线程并行图形数据传输方案，后台数据传输不影响前台界面交互，但同时运行多个 worker 会占用大量系统资源，综合考虑交互流畅性要求和系统资源的合理分配，本系统采用每接受一组图形数据开启两个主 Worker 的方案。两个主 Worker 并发执行，分别负责一次图形数据序列中奇序列和偶序列图形数据的传输。此外，如果用户选择的图形数据还在主 Worker 的等待队列中，系统立即开启一个新的子 Worker 负责此请求的数据传输。子 Worker 和主 Worker 之间通过互斥操作来防止读写冲突，如图 3 所示。

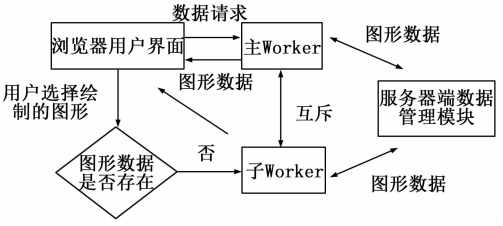

图 3　二维图形数据传输方案

2.3　基于 three. js 的三维模型渲染及交互

WebGL 框架 three. js 提供了图形接口可进行基本图元绘制并实现三维建模，但运载火箭模型结构复杂，用图形库生成和组合三角形或四边形单元，需确定每个多边形顶点坐标，整个模型需组合上万个单元，工作繁杂且开发难度大[5]。现有三维模型数据结构形式多样，其中 VTK（Visualization Toolkit）由于在计算机图形学、数据可视化上的优越性能得到了广泛应用[6]，论文以 VTK 模型为基础，提出并开发了模型转换算法及程序，以实现在浏览器端的三维模型渲染及交互。

2.3.1　模型渲染流程

浏览器端三维模型渲染主要包括以下流程：1）VTK 模型数据以文本形式存储在服务器端，存储有模型的顶点信息和多边形索引信息；2）浏览器端向服务器端发送 JavaScript 异步请求，服务器端接受请求并传送数据；3）JavaScript 解析模型数据文件并生成 geometry 对象，进而生成 Mesh 模型；4）将 Mesh 模型加入场景，并定义交互函数。流程如图 4 所示。

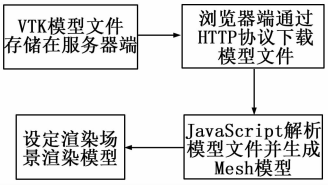

图 4　三维模型渲染流程

2.3.2　VTK 模型数据存储结构

VTK 模型是一种以文本方式表示的三维模型文件，其能够表示点面信息。其数据文件示例如图 5 所示，其中省略了大部分数据，主要包含四部分：VTK 版本及编码信息、点数据、面索引数据、点数目和面数目信息。

各部分存储信息概述如下：

1）vtk DataFile Version3. 0：vtk 文件版本 3. 0。

2）vtk output：vtk 文件名。

3）ASCII：vtk 文件编码：ASCII。

4）DATASET POLYDATA：数据集声明，模型包含三角形或四边形数据。

5）POINTS 35947 float：模型由 35947 个点组成，每个点分量为 float 型。

```
# vtk DataFile Version 3.0
vtk output
ASCII
DATASET POLYDATA
POINTS 35947 float
-0.0378297 0.12794 0.00447467
-0.0447794 0.128887 0.00190497
-0.0680095 0.151244 0.0371953
-0.00228741 0.13015 0.0232201
-0.0226054 0.126675 0.00715587
-0.0251078 0.125921 0.00624226
-0.0371209 0.127449 0.0017956
0.033213 0.112692 0.0276861
0.0380425 0.109755 0.0161689
......
......

POLYGONS 69451 277804
3 21216 21215 20399
3 9186 9280 14838
3 16020 13433 5187
......
......
CELL_DATA 69451
POINT_DATA 35947
```

图 5　VTK 模型数据文件内容示例

6）－0.0378297……：点数据，每 3 个数字表示一个点。

7）POLYGONS 69451 277804：模型由 69451 个多边形组成，数组长度 4 * 69451＝277804。

8）3 21216 21215 20399……：每个面由 3 个顶点组成，后面是 3 个索引数据。

9）CELL _ DATA 69451：模型面个数为 69451。

10）POINT _ DATA 35947：模型顶点个数为 35947。

2.3.3　基于 VTK 模型的转换算法

将原始 VTK 模型文件转换为 three.js 框架的 geometry 对象，进而生成 Mesh 模型，主要包括两大步骤：

1）提取 VTK 文件中的点数据并存储到 geometry 对象的 vertices 数组中；

2）提取 VTK 文件中每个点的索引数据并存储到 geometry 的 faces 数组中。转换步骤如下：

（1）点存储循环。遍历 VTK 模型数据，执行正则表达式匹配三空格分隔的 float 点数据，将点数据存入 geometry 对象的 vertices 数组；正则表达式定义如下：

pattern=/（[\＋|\－][\d]＋[\.][\d|\|e]＋）[]＋（[\＋|\－][\d]＋[\.][\d|\－|e]＋）[]＋（[\＋|\－][\d]＋[\.][\d|\－|e]＋）/g；

（2）面索引循环。遍历 VTK 模型数据，执行正则表达式匹配 POLYGONS 关键字定义的面索引数据，并将面索引数据存入 geometry 对象的 faces 数组，对于 3 顶点和 4 顶点定义的面，正则表达式定义如下：

pattern3 ＝ /3 [] ＋（[\d]＋）[]＋（[\d]＋）[]＋（[\d]＋）/g；

pattern4 ＝ /4 [] ＋（[\d]＋）[]＋（[\d]＋）[]＋（[\d]＋）[]＋（[\d]＋）/g；

（3）分别计算面的中心、面的法向量、顶点法向量及包围 geometry 的椭圆等参数，three.js 引擎将用于三维模型渲染。

2.3.4　模型渲染及交互实现

模型转换完成后需在浏览器端构建三维场景实现模型渲染及交互，主要包括：创建场景、创建摄像机、添加模型、设定渲染器、添加控制器、开启渲染循环等步骤，渲染结构图如图 6 所示，效果图如图 7 所示。

交互基于鼠标或键盘事件以及渲染循环实现。系统实现基本的三维动态交互包括旋转、缩放，模型旋转过程可转化为摄像机绕目标的旋转过程，缩放过程可转化为摄像机在其与目标连线所在的直线上移动的过程。通过添加控制器 TrackballControls 捕获鼠标或键盘事件，进而调整摄像机位置，同时需重新对场景进行渲染，系统调用 requestAnimationFrame 函数不断执行渲染操作并采用监听机制，保证了资源的有效利用及三维模型的流畅交互[7]。

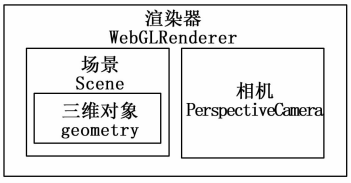

图 6　three.js 渲染结构图

图 7　三维模型渲染效果图

3　结论

论文设计了基于 JavaScript 框架 ExtJS 和 WebGL 框架 three.js 的运载火箭试验数据管理及可视化系统。系统支持试验数据的浏览、检索、提取，实现了浏览器端二维图形绘制及三维模型渲染，模型渲染基于本地显卡，在浏览器端对模型的旋转、缩放等交互操作均十分流畅，能够满足可视化的需求。系统不依赖于运行环境，无需任何插件或第三方工具软件的支持，可兼容国产化操作系统，降低了部署和维护成本。

参考文献：

[1] 张　军. 无线电遥测系统及在兵器试验中的应用 [M]. 北京：国防工业出版社，2011.

[2] Anon. Automated test and data management [J]. Microwave Journal，2006，49 (5)：284－290.

[3] 陈爱琳，刘经宇，刘丽霞. 基于 STK 的导弹飞行数据可视化系统设计 [J]. 计算机测量与控制，2013，21 (11)：3014－3016.

[4] 孙立坚，刘纪平，王　亮. B/S 结构下基于 Web Services 技术的空间分析方法的研究 [J]. 测绘科学，2005，30 (1)：60－62，110.

[5] 左　正，胡　昱，段云岭. 基于第 5 代 HTML 标准的拱坝工程三维可视化网络平台 [J]. 计算机辅助设计与图形学学报，2014，26 (4)：590－596.

[6] 高　鹏，刘　鹏，乔　梁. 基于 Web 的医学影像三维可视化实现方式 [J]. 中国数字医学，2013，8 (7)：78－82.

[7] 王珩玮，胡振中，林佳瑞. 面向 Web 的 BIM 三维浏览与信息管理 [J]. 土木建筑工程信息技术，2013，5 (3)：1－7.