**分类号**  **学号**

**学校代码 密级**

****

硕士学位论文

**题目：基于Web的增材制造预处理平台的设计与实现**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **学位申请人** | **：** |  | **张鹏** |  |  |
| **学科专业** | **：** |  | **计算机技术** |  |  |
| **指导教师** | **：** |  | **李国宽副教授** |  |  |
| **答辩日期** | **：** |  | **2021年6月XX日** |  |  |

**A Dissertation Submitted in Partial Fulfillment of the Requirements**

**for the Professional Master Degree**

**Design and implementation of additive manufacturing pretreatment platform based on Web**

**Candidate : Peng Zhang**

**Major : Computer technology**

**Supervisor : Prof. Guokuan Li**

**Huazhong University of Science and Technology**

**Wuhan 430074, P. R. China**

**June, 2021**

**独创性声明**

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

**学位论文版权使用授权书**

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密□， 在 年解密后适用本授权书。

本论文属于

不保密□。

（请在以上方框内打“√”）

学位论文作者签名： 指导教师签名：

日期： 年 月 日 日期： 年 月 日

# 摘 要

增材制造（Additive Manufacturing, AM）俗称3D打印，是一种以数字模型文件为基础，通过软件与数控系统将材料进行累加制造的方法。随着“中国制造2025”战略的提出，新的制造业将是汇集物联网、云制造、人工智能等前沿技术的大体系。针对3D打印最关键的数据处理部分，目前主要可分为开源软件和商业专用软件两类，开源软件大多只是对国外开源软件进行汉化处理，无法根据自己需求进行优化；商业专用软件则价格昂贵且操作复杂，不利于3D打印技术的普及和大众化发展。并且许多数据预处理软件载体都是本地客户端，操作完全离线，无法对数据文件进行统一管理，类似的模型处理过程没有整合封装，不利于3D打印产品的个性化定制与平台化应用发展。而随着Web3.0的时代到来，网速的快速增长，浏览器引擎的处理速度飞速提升，实时三维数据渲染的方案越来越完善，为3D打印预处理平台的构建打下了良好的基础。因此，本文通过对3D打印数据预处理过程中涉及到的流程与数据存储形式进行研究和分析，结合Web应用的优势，将打印过程分为几个核心模块，并自主开发出一个基于B(Browser)/S(Serve)模式的在线模型数据预处理应用。

首先，介绍了目前3D打印的软件需求以及行业发展现状，并根据打印流程分析，将数据处理分为几个关键模块，并简要介绍了Web应用的整个架构和全栈开发知识体系。其次，详细地阐述了系统中的数据结构，以及处理过程中涉及到的几个核心算法，采用前后端分离的开发模式实现了对系统用户和模型数据的管理，还有导入导出三维模型文件、分层切片、轨迹规划和GCode生成的功能。最后，部署在云服务器上，且在不同配置的电脑上对该网页应用进行试验验证。试验结果表明，增材制造数据预处理流程可以与Web应用良好地结合，用户交互界面合理，操作可行性强，算法执行效率高，基本上可以满足3D打印的数据处理需求，并且具备较高的扩展性和可移植性。

**关键词：**3D打印；智能制造；Web应用；Three.js；全栈开发；

# Abstract

英文摘要字体为Times New Roman，小四，1.5倍行距。

英文摘要和关键词应与中文相对应。英语摘要用词应准确，使用本学科通用的词汇；摘要中主语（作用）常常省略，因而一般使用被动语态；应使用正确的时态，并要注意主、谓语的一致，必要的冠词不能省略。

**Key words:** Optical microscopy; Brain; Neuron

**目** **录**

[摘 要 I](#_Toc65572950)

[Abstract II](#_Toc65572951)

[1. 绪论 1](#_Toc65572952)

[1.1 课题背景及研究意义 1](#_Toc65572953)

[1.2 3D打印技术简介及原理 2](#_Toc65572954)

[1.3 国内外研究现状 2](#_Toc65572955)

[1.4 本文主要内容 2](#_Toc65572956)

[2. Web应用原理研究 3](#_Toc65572957)

[2.1 Web技术发展简介 3](#_Toc65572958)

[2.2 单页面应用（SPA） 3](#_Toc65572959)

[2.3 WebGL原理 3](#_Toc65572960)

[2.4 基于B/S模式的全栈架构 4](#_Toc65572961)

[2.5 本章小结 4](#_Toc65572962)

[3. STL文件模型简介及切片算法研究 5](#_Toc65572963)

[3.1 STL模型文件简介 5](#_Toc65572964)

[3.2 STL文件拓扑 6](#_Toc65572965)

[3.3 STL文件拓扑重建 6](#_Toc65572966)

[3.4 STL模型分层处理算法 6](#_Toc65572967)

[3.5 本章小结 6](#_Toc65572968)

[4. 轨迹填充路径算法研究 7](#_Toc65572969)

[4.1 常见的路径规划算法 7](#_Toc65572970)

[4.2 直线扫描填充算法 7](#_Toc65572971)

[4.3 GCode生成算法 7](#_Toc65572972)

[4.4 本章小结 7](#_Toc65572973)

[5. Web应用系统实现与测试 8](#_Toc65572974)

[5.1 应用开发环境 8](#_Toc65572975)

[5.2 界面设计 8](#_Toc65572976)

[5.3 功能模块设计 8](#_Toc65572977)

[5.4 服务器部署 9](#_Toc65572978)

[5.5 系统展示 9](#_Toc65572979)

[5.6 本章小结 9](#_Toc65572980)

[6. 总结与展望 10](#_Toc65572981)

[6.1 本文主要内容及结论 10](#_Toc65572982)

[6.2 本文的主要创新点 10](#_Toc65572983)

[6.3 展望 10](#_Toc65572984)

[致 谢 11](#_Toc65572985)

[参考文献 12](#_Toc65572986)

[附录1 中英文缩写对照表 14](#_Toc65572987)

目录编写要求：

1. 目录自动生成时，只需列出二级标题；
2. 中文宋体，小四，行距：1.5倍；
3. 注意避免出现不具辨识度的标题：如材料与方法、实验结果、分析与讨论、结论，若属于这种类型的论文时，可以把结果部分拆开，根据结果拟定题目。

# 绪论

## 课题背景及研究意义

中国近十一年来一直位列全球制造大国之首，因此制造业是事关国民经济命脉的主要支柱产业，代表了国家的科技发展与综合实力水平。2015年两会上初次提出“中国制造2025”的重要战略规划与目标，未来的制造业无疑是要结合物联网、云制造、人工智能等前沿信息技术，朝着智能制造的方向发展。

3D打印又被称为增材制造、积层制造，是一种以数字模型文件为基础，运用粉末状金属或塑料等可粘合材料，通过逐层打印的方式来构造物体的技术，该技术最早提出于20世纪80年代。常用于模具制造、工业设计等领域，后逐渐用于一些产品的直接制造，与传统加工工艺对比起来，增材制造加工效率高，耗材低，易于成型，且可以满足定制化需求，是制造业中比较颠覆性的技术。目前，根据打印的工艺和成型材料，可以将3D打印分为立体光刻、熔融沉积成型、立体平板印刷、选择性激光烧结、电子束自由成形制造和分层实体制造等方法[1]。3D打印的过程可以由离散和堆积两大部分组成：离散是使用一系列的平面对数据模型进行切片处理，获得每一层平面的轮廓；分层切片后根据不同的工艺要求，对每个轮廓进行计算处理，得到模型每一层的加工路径，再将加工路径翻译成打印机器可执行的加工代码，使得机器可以根据生成的代码进行加工，这就是堆积的过程。

在3D打印整个流程中最核心的部分就是对模型数据的处理，不同的参数以及计算算法都将直接影响产品的质量和精度。针对模型的数据处理就要涉及到模型导入、模型拓扑重建、模型切片、扫描路径生成等步骤，在实际生产过程中可能要使用到多个软件，并且大多数开源软件都是一种黑盒模式，无法进行自适应优化处理，同一个模型使用不同软件得到的打印代码都是有偏差的，因此无法满足专业人士的需求，而且处理过程的繁琐也不利于3D打印技术面向大众进行推广。

众所周知，自二十一世纪以年来发展最快的就是互联网行业，以互联网为载体的信息技术已经逐渐改变传统生产及商业模式，引发全球工业产业的技术升级。尤其在近几年基于浏览器的应用蓬勃发展，每年的在线应用都是成倍增长，甚至客户端软件都开始逐步开发Web端应用，比如CAD绘图，Web版Photoshop以及在线视频剪辑等。所以研究如何将3D打印流程结合Web应用，使得操作精简化、计算统一化、数据共享化、功能平台化，这将具有十分重要的现实意义。

## 国内外研究现状

### 3D打印技术研究现状

3D打印与传统制造工业相比，最直观的感受就是它不需要模具，而是可以利用计算机建模的数据，就可以通过逐层累加的方式进行填充打印。相对于传统的材料去除(切削加工)技术，是一种“自上而下”材料累加的制造方法。自20世纪80年代末增材制造技术逐步发展，期间也被称为“材料累加制造”(Material increase manufacturing)、“快速原型”(Rapid proto-typing)、“分层制造”(Layered manufacturing)、“实体自由制造”(Solid free － Form fabrication)等[4]。从名称各异的叫法中我们就能得知该制造技术的特点。美国材料与试验协会国际委员会对增材制造和 3D 打印有明确的概念定义。3D 打印是指采用打印喷头或其他打印技术沉积材料来制造物体的技术，3D 打印也常用来表示“增材制造”技术，在特指设备时，3D 打印是指相对价格或总体功能低端的增材制造设备[5]。

根据打印材料、形成模型方式不同，3D打印方法主要分为六类，分别是：熔融沉积成形（Fused Deposition Modeling，简称为FDM）、金属激光熔融沉淀（Laser Metal Deposition，简称为LMD）、光固化几何体成形（Stereo lithography Apparatus，简称为SLA）、选择性激光烧结（Selective Laser Sintering,，简称为SLS）、分层实体制造（Laminated Object Manufacturing，简称为LOM）、电子束熔融（Electron Beam Melting，简称为EBM）[6]。其中最常见的3D打印技术就是FDM技术，FDM技术最早是由美国的Scott Crump博士于1988年研制成功，它具有材料成本低、利用率高、成形速度快、设备小巧、操作简单等特点[3]。该技术适用于大多数复杂模型的制造，被广泛应用在建筑模型设计、医疗器械制备、航空机械零件生产等领域。

FDM技术的主要流程是，先通过建模软件得到满足需求的模型文件，再将得到的模型文件通过专业软件进行分层切片处理，根据切片得到的轮廓判定轮廓之间的内外关系，再根据轮廓形成的不同包络区域进行路径规划填充，最后将规划路径转为打印机可执行的G代码。打印机读取GCode之后，加热打印喷头，将融化的材料匀速挤出，沿着每个切片平面方向进行均匀涂抹，直至不断调整高度填充整个模型，就生成了最终的打印形件。FDM打印机结构如图 1.1所示：

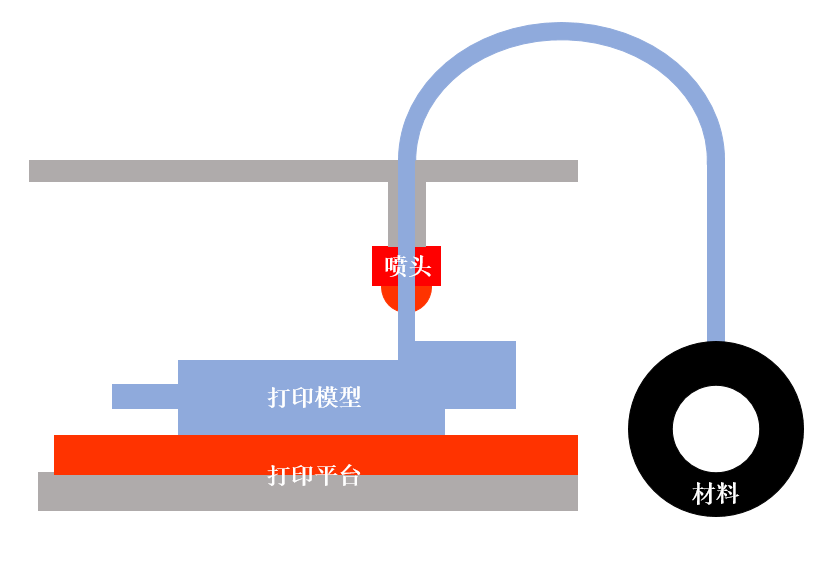


图 1 FDM打印机结构示意图

3D打印的基础流程包括以下五个方面：

1. 三维模型建模：目前市场上三维模型制作方式有很多种，比如人工建模，扫描建模，照片建模等等，根据不同的建模方式选择的软件也有所不同，如：CAD、3DMax、Zbrush、SketchUp等。三维模型生成方式十分灵活，还可以添加渲染、特效、特殊材料等，满足用户的个性化需求。
2. STL文件格式转换：常见的三维文件格式有：ABC（动画模型文件格式）、glTF（动画模型文件格式）、FBX（3D文件格式）、BVH（人体特征动画文件格式）、OBJ（标准3D模型文件格式）、DAE（纯文本模型格式）、STL（三角网格文件格式）、3DS（早期3D文件格式）等，不同的文件格式适用的场景不一样，并且现在市面上有许多软件都可以将不同格式的文件进行互相转换，由于本文研究基于三角面片网格模型，所以我们选择的是STL格式的三维模型。
3. 分层处理：模型在结构上是由一个个三角面片组成，所以需要对STL文件进行分层切片，将同一层相邻的面片顺序存储在每个轮廓的数据结构中，分的层数越多，模型精度就越高，但是耗费的计算时间以及计算空间就越大。
4. 路径规划：分层处理后，将得到各个轮廓的数据集，需要遍历轮廓找出内外轮廓关系，划分不同的填充区域，还要根据打印机特性设置不同的填充方式，保证有效合理地去生成打印轨迹。
5. 打印成品：打印机根据生成的GCode文件，控制打印针头沿着预设的轨迹、速度、喷丝方向进行堆积喷涂，直至最终形成三维实体器件。

在打印的过程中，数据信息流的传递方式如图 1.2所示：

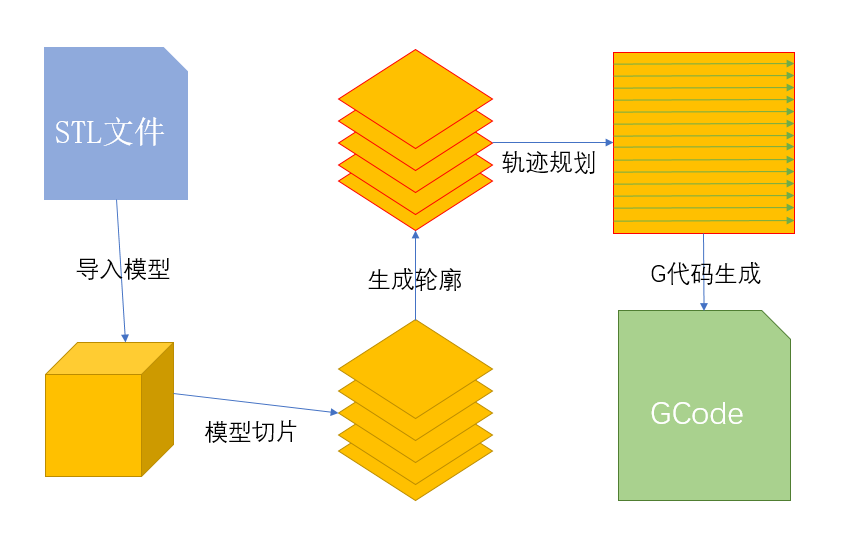


图 2 3D打印过程数据信息流示意图

### Web技术研究现状

1989年欧洲核子研究中心实验室研究员蒂姆·伯纳斯·李（Tim Berners-Lee）发明了万维网[7]，是互联网史上具有划时代意义的事件，自此之后Web应用才得以爆炸式的繁荣起来。下面是Web发展的几个里程碑事件：

1.HTML（Hyper Text Markup Language）：HTML是由万维网发明者Tim Berners-Lee和同事Daniel W. Connolly于1990年创立的一种标记语言[8]。它通过标记式的指令，将影像、声音、图片、文字动画、影视等内容显示出来，是编写静态网页的标准语言，是今后网页必不可少的基本元素。

2. Mosaic浏览器：Mosaic是互联网历史上第一个普遍使用和能够显示图片的网页浏览器，在1993年发表，并于1997年1月7日正式终止开发和支持[9]。由于是第一款具有方便易用的图形界面的浏览器，让当时许多人了解到了Internet，尽管后来停止了前进的脚步，但是它对后来出现的浏览器影响深远。

3.CSS（Cascading Style Sheets）：早期的网页无法提供页面的装饰，只有单调的文字排布，并且要在HTML中内嵌进修饰的代码会使得HTML的结构变得臃肿混乱。因此在1994 年，Håkon Wium Lie 最初提出了 CSS 的想法，联合当时正在设计 Argo 的浏览器的Bert Bos，创造了 CSS 的最初版本，并在1996 年 12 月，W3C 推出了 CSS 规范的第一版本[10]。

4.JavaScript：JavaScript是一种基于客户端浏览器的网页脚本语言，它可以直接和HTML文档整合，其编译过程不需要专门的编译器，通过浏览器进行逐行解释和执行[13]。最早是由网景公司（Netscape）发明，主要目的是想用一个脚本语言来解决一些表单的验证问题，而真正开始影响到Web开发进程的事件是1999年IE4.0发布，内置异步通讯的组件，允许脚本语言发送HTTP请求到服务器，这就是Ajax的原型。

5.Ajax（Asynchronous Javascript And XML）：Ajax是指一种创建交互式、快速动态网页应用的网页开发技术，无需重新加载整个页面的情况下，能够更新部分网页的技术，最早于1998年前后得到应用[12]。在该项技术发明之前，我们网页的交互方式是用户点击一个按钮后，需要等待漫长的数据和服务器的交互，期间用户无法进行任何操作，而Ajax的作用就是让应用向服务器进行请求交互时，用户不必等待结果，而是可以同时对页面做其它操作，使得用户体验得到大大的提高，也为前后端分离的开发模式奠定了基础。

6.Flash：互联网Web应用的兴起，用户对于在网页上获取图像和动画的需求变得越来越强烈，于是Flash应运而生。Flash 最早期的版本称为Future Splash Animator，是1996年11月，乔纳森·盖伊(Jonathan Gay)和他的六人小组编写而成[11]。经过多年的开发迭代，于2005年9月，Flash 8.0发布，增强了对视频的支持，可将其打包成flv文件，同年，Adobe耗资34亿美元并购Macromedia，Flash正式成为Adboe公司旗下的产品。由于可以在浏览器中播放动画，Flash开始异军突起引爆了整个流媒体行业，并在2007年左右，随着Flash 的交互技术进步，网页游戏的浪潮开始席卷全球，在2010年Flash达到巅峰，直至2015年HTML5的兴起和移动互联网的发展，再加上Flash本身的安全性问题，Flash才渐渐地退出历史舞台。

7.jQuery：jQuery是2006年发布的一个免费开源并且跨浏览器的JavaScript框架,其核心设计理念是写更少的代码，做更多的事情（Write Less Do More）[14]。jQuery最大的优点就是减少了开发者对页面DOM树的维护，封装了许多JavaScript处理函数以及Ajax异步调用实现的封装，并且都做了兼容性处理，大大减小了JavaScript操作页面的复杂度，使得JavaScript 代码和 HTML 代码完全分离，便于代码的维护和修改。

8. Google Chrome：Google浏览器，是一款由Google公司开发的网页浏览器，于2008年9月发布beta版本，Chrome使用Apple的开源内核webkit，良好的设计标准和市场反应，促进浏览器快速迭代，直接撼动了IE在浏览器界的地位。由于它便捷的调试工具以及丰富的插件市场，受到了广大Web程序开发者的喜爱。

9.Github：Linux创始人Linus Torvalds在2005年就开发了Git的原型程序，是为版本管理而设计的软件，在发布之初，Git 由于其艰涩难懂，只有部分黑客愿意使用。但随着众多开发者的共同努力，Git迅速成为最流行的分布式版本控制系统，尤其是2008年，GitHub网站上线，它为开源项目免费提供Git存储，无数开源项目开始迁移至GitHub，包括jQuery，PHP，Ruby等等[15]，为互联网行业的飞速发展打下了基石。

10. Bootstrap：2011年Twitter开源的网页GUI框架，作用就好比jQuery对JavaScript进行了浏览器兼容，那么Bootstrap则是兼容了不同浏览器的CSS部分[16]。它带来的好处就是你无需知道CSS，无需前端，只要根据文档说明，就可以快速搭建起一个美观的网页，让开发者重心放在页面的逻辑交互，而不是处理各个浏览器的样式兼容，虽然它会导致页面的千篇一律，但是它为了解决响应式问题所提出的栅格化系统，是今后各式各样的UI框架借鉴的鼻祖。

11. Chrome V8引擎：V8引擎是一个JavaScript引擎实现，最初由一些语言方面专家设计，后被谷歌收购，在2008年9月谷歌对其进行了开源。V8使用C++开发，在运行JavaScript之前，相比其它的JavaScript的引擎转换成字节码或解释执行，V8将其编译成原生机器码（IA-32, x86-64, ARM, or MIPS CPUs），并且使用了如内联缓存（inline caching）等方法来提高性能[17]。有了这些功能，JavaScript程序在V8引擎下的运行速度媲美二进制程序。V8支持众多操作系统，如windows、linux、android等，也支持其他硬件架构，如IA32,X64,ARM等，具有很好的可移植和跨平台特性。

12.Node.js：由于V8引擎的出现，满足了对高性能Web服务器的想象，于是在2009年5月由Ryan Dahl开发，基于Chrome V8引擎的JavaScript运行环境就此诞生[18]。它是一个事件驱动、非阻塞式I/O模型，可以让JavaScript 运行在服务端的开发平台，后面在Node.js的基础上又推出了NPM包管理器，让大量工具包得以共享使用，推动着前端行业的增量式发展。

13.HTML5+CSS3：2014年10月HTML5规范制定完成，HTML5相比于HTML而言，增添了许多标签，可以更多维度的构建出对用户有价值的数据。HTML5的APP Cache以及本地存储功能大大缩短了一些APP的启动时间，同时，HTML5通过将内部和外部的数据直接连接，有效的解决了设备之间的兼容性问题，此外，HTML5具有动画、多媒体模块、三维特性等等，可以替代部分Flash和Silverlight能够实现的功能，并且具有更好的处理效率[19]。CSS3则是在CSS的基础上，采用模块化开发结构包括框模型、动画模型、背景与边框模型、文字效果、列表展示、多列布局等等，还补充了HTML5中的文本样式[20]。对于开发者来说，可以用更简洁的代码，得到更丰富的页面效果，大大地提高了工作效率。

14.前后端分离模式：早期网站的服务模式是发送一个http请求，然后服务器对请求做完处理，最后将静态的资源返回到页面中，因此所有静态资源和业务代码统一部署在同一台服务器上，整个流程如图 1.3所示：

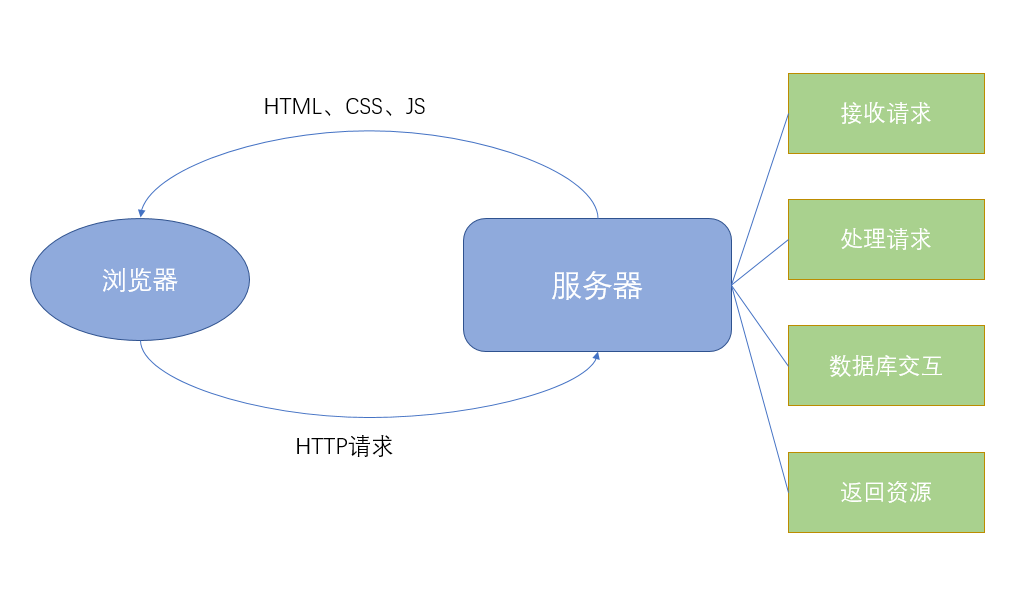


图 1.3 早期网站服务流程

在这样的模式下，前端工作就只有将UI设计师提供的原型图还原实现成静态页面交付于后端工程师，然后后端人员再加入逻辑代码，使得前后端工作极度耦合，相互依赖严重，一旦设计改变，就需要重新走一遍流程，导致开发效率低，代码难以维护。直到2013年左右，前后端分离思想的出现，基于Node.js服务器，让前端去负责View层和Controller层，后端只负责Model层和Service层，由于Node.js中间层的加入，浏览器不再直接请求服务端的API，而是先去请求Node.js，由Node.js对服务端的API发送HTTP请求，Node.js收到服务端返回的JSON格式数据再去渲染HTML页面[21]。前后端分离不仅将开发人员的职责明确区分开来，还大大提升了适配性，如果Controller层掌握在后端手里，那么多端适配的时候，后端就要维护不同环境下的页面，导致模板无法重用，而且大多数时候后端的业务逻辑是一样的，如果前后端无法分离，就多出了很多无意义的工作量。

15.WebGL（Web Graphics Library）：WebGL是一种用于Web的标准3D绘图协议，基于OpenGL ES2.0的JavaScript API进行开发。OpenGL（Open Graphics Library）是一种成熟的跨平台高性能3D渲染API，它由Silicon Graphics Inc.开发，然后在 1992年发布，目前由The Khronos Group管理。它是一种基于渲染管线、独立于硬件，且具有客户端-服务器结构化的API[22]。WebGL初始版本是于2011年3月被提出，WebGL2.0的规范发展与2013年并于2017年1月完成。自WebGL问世后，网页应用又可以开拓出更多可能性，例如3D页游，虚拟现实等。

Web技术经过这二三十年来的发展，已经有了天翻地覆的改变，从最初只能制作一个单纯的静态文本页面，到如今可以承载Web OS、AR、VR和人工智能等前沿科技领域的强大技术。截止到2020年底，全球网站数量已经突破十亿，尤其是近十年来，网页数量呈几何级数增长，各式各样的应用都在朝着Web化的方向发展，因为这不仅仅是将应用迁移到Web端，而是思维开始向云计算、云服务、平台化等潮流转变。当底层硬件性能的壁垒不断被打通，Web几乎是未来应用载体的首要选择。

## 本文研究内容及安排

### 本论文的研究内容

随着Web进入3.0时代以后，Web端实时通信、大数据流传输以及WebGL技术的日趋成熟，我们可以看到Web App正以势不可挡的趋势发展，快速占领各个行业，很多基于客户端的程序都在逐渐向Web端扩展。比如：Web Photoshop，AutoCAD Web App、Tim在线文档等产品。而增材制造预处理过程中所用到的软件都是离线的客户端程序，每个流程的数据信息都是独立使用，没有统一的数据存储，对于每一次的处理记录都无法做到数据追踪，并且增材制造预处理过程中的算法也会不断优化更新，而客户端程序的更新要面临多端适配的问题，并且具有版权的算法在客户端程序里也会面临着被破解的风险，而Web化的好处不仅是无感知更新，即开即用等优点，并且还能将计算云服务化，数据云存储共享化，这是未来互联网应用发展的必然趋势。在五六年前Web应用程序的执行效率确实无法超过客户端，但是随着技术的不断发展，曾经的技术瓶颈被一个个打破，浏览器厂商与行业标准在不断进步趋同，因此基于Web的增材制造预处理平台的实现将具有十分重要的意义。

基于以上原因，本课题对Web技术如何应用到增材制造预处理过程进行了研究，主要研究内容有以下几点：

1. 对WebGL技术和Three.js框架的概念、优缺点及发展现状进行了调研与系统性总结，并对具体应用的开发架构进行了设计和改进，使得模型交互更加合理，提高用户体验。
2. 对主流的前端开发进行了系统性的介绍，并针对当前应用给出了合理的技术选型，对于系统开发所涉及到的功能模块进行了二次封装，提升整个应用的代码可阅读性。
3. 研究了增材制造预处理的数据源，并根据模型数据本身的特性去设计数据库结构，实现了对模型文件的存储，以及系统内部其他数据的存储结构，并封装好公有函数对数据进行增删改查操作。
4. 将增材制造预处理过程中涉及到的几大算法进行研究分析，详细阐述了每个算法的流程，并且根据实际Web应用开发的数据结构，对算法进行了优化调整，并用实际模型对算法进行了测试。
5. 对整个系统进行功能、界面以及交互方式的设计，实现了用户登录注册、模型上传修改、模型导入导出、模型放缩还原、模型切片、轨迹填充、GCode生成和动画演示等功能，并截取主要代码进行辅助说明。
6. 将应用程序分包部署在服务器上，并在不同设备中测试网页的可使用性，并对整个部署过程做了详细的说明。

### 本论文的结构安排

第一章，绪论。主要介绍本文课题的研究背景与研究意义，3D打印技术的研究现状，Web技术的研究现状。

第二章，增材制造预处理Web平台的关键技术及原理，首先对系统做需求分析，并设计了具体的系统实施方案，然后研究了Web应用的全栈化开发原理与相应的技术栈和数据库选型，最后对增材制造预处理过程中所涉及到的一些关键算法进行了分析和介绍。

第三章，增材制造预处理过程中相关算法在Web端实现，本章通过对STL模型结构的分析，实现了在网页中对STL模型的上传下载以及导入导出的功能。并对预处理过程中所涉及到的拓扑重建算法、切片算法、轨迹填充算法以及GCode生成算法进行了Web端的迁移与测试，并通过关键代码做辅助说明，介绍了在Web上如何对算法进行优化与改进。

第四章，Web应用系统实现与测试，对本系统的各个流程进行了详细的设计，并通过开发环境的搭建展示出基于前后端分离的Web应用系统架构，然后在应用模块对每个流程进行详细的功能设计，最后叙述了如何将Web应用部署于服务器，并进行使用展示。

第五章，总结与展望，概述本文研究的主要内容，提出系统研究过程中遇到的问题和不足，并对基于Web的增材制造预处理平台进行了展望。

# 关键技术及原理

## 数据存储

由于绪论中已对全文相关的研究背景和进展做了综述，因此，每章的引言中，请用1页左右的版面写一个导引，简要说明本章研究的背景或动机，以起到承上启下的作用，不宜太长。

### MongoDB

### Mongoose

## Web3D原理

### WebGL技术

#### WebGL开发流程

#### WebGL渲染原理

### Three.js技术

#### Three.js介绍

#### Three.js基础渲染组件

#### Three.js渲染流程

## 前后端分离架构

### HTML5技术

### VUE框架

#### VUE简介

#### Vue-router

#### Vuex

#### Axios

### Node.js

### Express.js框架

## 本章小结

本章主要介绍实验研究类的论文正文章节的框架结构。在每章的最后，都需要对该章的内容进行小结，不宜太长，建议1/2-2/3页版面较好。主要小结一下本章用什么理论或方法、做了什么事、得到的重要结果或结论。

# 需求分析及概要设计

## 需求分析

## 系统流程设计

## 开发环境搭建

### Node.js服务器搭建

### Webpack打包工具安装

### 安装数据库

### 建立Git远程仓库

### 选择编辑器

### 项目初始化

#### 前端初始化

#### 服务端初始化

#### 数据库初始化

### 软硬件平台

## 本章小结

# 相关算法及Web端实现

## STL模型文件

### STL模型文件简介

### STL文件和切片

### STL文件的拓扑结构

### STL文件导入导出

## STL文件拓扑重建

### 三角网格读取

### 冗余数据判断

### 拓扑重建算法

### 算法实现与测试

## STL模型等厚分层处理算法

### 分层切片处理流程

### 切片轮廓生成算法

### 算法实现与测试

## 轨迹规划算法

### 常见的路径规划算法

### 直线扫描填充算法

#### 算法简介

#### 数据结构设计

#### 内外轮廓判别

#### 算法实现与测试

## GCode生成

### GCode简介

### 算法实现与测试

## 本章小结

# Web应用系统实现与测试

## 应用开发环境

对全文进行全面地总结，并根据各章节归纳出若干有机联系的论点。按正文的内容分段描述，包括本研究“做了什么（提出\*\*新理论/算法、设计或研发\*\*工艺/仪器）、获取什么结果、得出什么结论”。

## 界面设计

### 登录注册页面

### 模型列表页面

### 工作台页面

## 数据库设计

## 服务端设计

## 项目部署

## 系统展示

## 本章小结

# 总结与展望

## 本文主要内容及结论

## 本文的主要创新点

通常情况下，学位论文的创新点应放在最后一章。

创新点要凝炼，表述要清晰明了，如提出了什么创新的思路，主要特点是什么，相比现有理论或技术的提高是什么、或者有什么新的发现，是否具有重要的科学意义或应用前景。既不能过于简单，也不要太细。

硕士学位论文创新点不宜太多，一般为2个左右即可，要注意归纳创新点，千万不要以为越多越好。论文的创新不以创新点的多少来评定的，而以其创新性的价值来评定。几章的工作合在一起凝炼成一个创新点也不是不可以的。

## 展望

。

# 致 谢

。

# 

# 参考文献

武汉光电国家研究中心规定：硕士学位申请人的文献阅读量不少于600篇（高于学校规定的40篇），其中外文文献不得少于2/3；近五年的论文不得少于1/3；绪论部分应对所读文献加以分析和综合。

**参考文献格式**：

中文书刊：作者按中文写法，姓在前、名在后；英文书刊：作者按英文习惯写法，如在前、姓在后，名用首字母缩写、姓用全称。一般6人以内须列出全部作者，6人以上写6人再加“等”（英文加“et al”））。每个参考文献的最后不加标点符号。

1. 图书：最多列出6个作者，作者与作者之间用逗号分隔. 书名. 版本（第×版）. 译者.出版地：出版者，出版年. 起页-止页（可选）
2. 期刊：最多列出6个作者，作者与作者之间用逗号分隔. 文章名. 期刊名（全称），年号，卷号（期号）：起页-止页或论文编号
3. 会议论文集：最多列出6个作者，作者之间用逗号分隔. 文章名.见（英文用“in”）：会议名称（或论文集）. 会议城市，国家，会议时间，出版者，出版年：起页-止页
4. 专利：专利申请者. 专利题名. 专利国别，专利文献种类，专利号，出版年
5. 学位论文：作者. 题名：[博士（或硕士）学位论文]. 保存地点：保存单位（如华中科技大学），年份

**参考文献**

1. G. K. Awari et al. Additive Manufacturing and 3D Printing Technology: Principles and Applications[M]. CRC Press, 2021
2. M. Chalfie, S. R. Kain. Green fluorescent protein: properties, applications, and protocols. Hoboken, New Jersey: Wiley-interscience, 1998
3. Tosto Claudio et al. Fused Deposition Modelling (FDM): New Standards for Mechanical Characterization[J]. Macromolecular Symposia, 2021, 395(1)
4. 卢秉恒,李涤尘.增材制造(3D打印)技术发展[J].机械制造与自动化,2013,42(04):1-4.
5. Wohlers Associates, Wohlers Report 2013--Additive manu-facturing and 3D printing state of the Industry Annual WorldwideProgress Report, 2013, USA
6. 陈志茹,夏承东,李龙,等.3D 打印材料[J].金属世界,2018(05):15-18.
7. Neil Savage. Weaving the web[J]. Communications of the ACM, 2017, 60(6): 20-22.
8. Tabarés Raúl. HTML5 and the evolution of HTML; tracing the origins of digital platforms[J]. Technology in Society, 2021, 65.
9. Aiello M. The Browser Lament[J]. In: The Web Was Done by Amateurs. Springer, Cham, 2018.
10. Claussnitzer, Melina, Cho, Judy H, Collins, Rory, et al. A brief history of human disease genetics[J]. Nature, 2020, 577(7789), 179–189.
11. Stian Reimers, Neil Stewart. Adobe Flash as a medium for online experimentation: A test of reaction time measurement capabilities[J]. 2007, 39(3), 365–370.
12. Jesse James Garrett. Ajax: A new approach to web applications[EB/OL]. 2005, <https://www.adaptivepath.com/publications/essays/archives/000385print.php>.
13. G. Richards, S. Lebresne, B. Burg, and J. Vitek. An analysis of the dynamic behavior of JavaScript programs[J]. In: ACM Sigplan Notices, 2010, vol. 45, no. 6, pp. 1–12.
14. J. Resig and others. jquery: The write less, do more, javascript library. Disponvelem, 2009.
15. 姚雅斐. 软件生态系统的健康性评估[D]. 中国矿业大学, 2020.
16. Tim Hesterberg[J]. Bootstrap, 2011, 3(6), 497–526.
17. 秦梦远, 傅忠传. 基于Chrome调试协议的Chromium浏览器V8引擎漏洞检测方法[J].智能计算机与应用,2020,10(02):320-324.
18. Tilkov S, Vinoski S. Node.js: Using JavaScript to Build High-Performance Network Programs[J]. IEEE Internet Computing, 2010, 14(6):80-83.
19. 陆凌牛. HTML5与CSS3权威指南[M]. 机械工业出版社, 2011:60-100.
20. S Mohorovi ci c. Implementing Responsive Web Design for Enhanced Web Presence[C]. Information & Communication Technology Electronics & Microelectronics (MIPRO), 2013 36th International Convention on IEEE 2013:1206-1210.
21. 杜艳美, 黄晓芳. 面向企业级web应用的前后端分离开发模式及实践[J]. 西南科技大学学报, 2018, v. 33; No.130(02):86-90.
22. Chandler, Jennifer Sara. Techniques for Visualizing Particle-Based Flow Fields[D]. ProQuest Dissertations and Theses Full-text Search Platform, 2016.

# 附录1 中英文缩写对照表

3D Three-dimensional（三维）

CT Computer tomography (计算机断层层析成像)

MRI Magnetic resonance imaging（磁共振成像）

PET Positron emission computed tomography（正电子发射断层成像）

…