

摘 要

由于 RP 系统不带有 CAD 造型功能,从 CAD 系统导入图形文件又存在诸多缺陷,因而三维图形缺陷诊断与修复成为 RP 系统数据处理的基础和关键。

本文从研究整个金属直接快速成型系统硬件堆积平台,成型工艺出发,分析了国内外关于三维图形缺陷诊断与修复技术的发展趋势,研究了基于图形渲染工具 OpenGL 技术的三维图形显示系统,并结合 CAD 系统几何造型理论,具体分析相关算法,设计并实现了一个三维图形缺陷自动诊断与修复工具。

文章首先介绍了熔焊快速成型及其数据处理相关技术,并对 CAD 系统导出的三维图形缺陷诊断与修复技术的发展,现状进行了较深入的分析,对当前三维图形修复工具做了总体介绍。

其次,基于 OpenGL 真实感显示技术,根据熔焊快速成型工艺实际需要,给出了三维图形缺陷诊断和修复系统的设计方案,包括系统总体架构、设计模型、模块划分、各模块功能实现等。

其中,重点对三维图形点,边和面几何元素的拓扑关系建立算法进行了具体分析,减小了算法的复杂程度,提高了图形加载速度,更重要的是,对后续基于拓扑关系的图形缺陷诊断和修复算法的实现提供了有效的处理依据,在实验环境中进行了验证并取得了较好的效果。

关键词: 熔焊快速成型、STL 文件、拓扑信息、错误诊断与修复

Abstract

Not including CAD, the software system in RP has to import another file which has so many errors. Thus, error diagnosis and correction tool plays an important role in software.

Departing from the introduction of our metal direct rapid prototyping system including the deposit hardware platform and the prototyping software, the investigation of the error diagnosis and correction of STL model exported from CAD, this paper makes a study of the 3D graphics vision system based on the OpenGL, integrated of geometry shaping theory in CAD thus then designs and realizes one tool of error diagnosis and correction of STL model. Also the relative algorithms have been researched.

Firstly, some relevant theories of the fusion welding rapid prototyping and its data process technology are introduced in this paper. Also it gives some analysis to the present condition of the error diagnosis and correction of STL model exported from CAD based on the theories and some current errors diagnosis and correction tool.

Secondly, exhausts some conception of OpenGL vision display technology, which is a popular graphics software interface widely used in graphics system. After that, the advantage and application of them in building systems will be introduced.

Finally, according to the actual requirement of fusion welding craftwork, we hold out the project's design of error diagnosis and correction of STL model, including system total configuration, design model, model distribution and each model's function.

At the same time, emphasize the topological information build algorithm between points and edges, between facets and points also between facets and edges. It decreases complexity of model processing; improve the speed of loading STL model. The most importance is that it provide the efficient method dealing with the diagnosis and correction of STL model on topological information between geometry entity.

Wan Lingna[Control Theory & Control Engineering]

Directed by Prof. Zhang Hua

Key Words: Fusion Welding Rapid Prototyping STL Model Topological Information,
Error Diagnosis and Correction

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 南昌大学 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：丁玲娜 签字日期：2006年6月 日

学位论文授权使用授权书

本学位论文作者完全了解 南昌大学 有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权 南昌大学 可以将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、汇编学位论文。

(保密的学位论文在解密后适用本授权书)

学位论文作者签名：

丁玲娜

导师签名：

张

签字日期：2006年6月 日

签字日期： 年 月 日

学位论文作者毕业后去向：

工作单位：

电话：

通讯地址：

邮编：

第一章 绪论

1.1 课题来源及其意义

1.1.1 课题的提出

智能材料结构在军事、航空航天、汽车、船舶和医学领域有着广泛的应用^[1]。相对与应用广泛的非金属成型，金属结构成型研究并不多。其瓶颈是：金属材料的智能化，即金属结构的传感器嵌入技术以及在传感器埋入过程中使用的快速成型技术。基于材料堆积的金属零件直接快速成型制造技术恰如其分地成为最有效的制备方法^[2-3]。

自上个世纪 80 年代后期问世以来，快速成型技术（RP）在 10 余年时间到了迅猛发展，但主要集中在非金属材料成型领域。在金属成型领域，焊接在加工成本，加工工艺等方面存在优势，因而熔焊快速成型制造技术成为最有希望实现高致密度金属零件经济、快速制造的方法之一。它的基本过程是：首先在 CAD 系统中造型出零件的曲面或实体模型；接着，零件的三维数据模型经数据转化和处理，生成了零件二维截面轮廓数据信息；其次，进行实体二维截面的轨迹填充处理，生成熔焊加工轨迹；最后，由焊接电弧将焊丝或添加材料熔化，逐层堆积，快速生成零件的三维实体。

由于目前普遍使用的快速成型制造系统（RPM）还不具备实体造型功能，必须依靠 CAD 系统为其提供满足快速成型输入条件的三维图形数据文件。在众多的 RP 和 CAD 数据接口类型当中，STL 模型结构简单，读取便捷，获得了众多 CAD 系统厂商的接口支持，成为了 CAD 系统与 RP 系统之间默认的事实标准。但是，在 CAD 系统下 STL 转化过程中，由于 STL 数据文件本身是作为对 CAD 系统造型设计的三维实体模型表面近似，在一定精度要求上，不能够完整的描述三维几何模型，存在一定的缺陷和错误。该缺陷在几何上表现为实体表面的不完整。它影响了后续切片处理，造成切片后的二维截面轮廓的断裂和不完整，从而对熔焊快速成型制造的制造精度、表面质量、制作时间、制作成本甚至制作成败都有重要影响。

1.1.2 课题的背景

江西省机器人与焊接重点实验室自 2003 年开始智能结构熔焊快速成型技术研究工作以来, 获得了国家重大基础研究前期专项, 江西省自然科学基金的支持。本课题隶属于其中国家重大基础研究前期研究专项“光纤智能金属结构熔焊快速制造技术的基础研究”(编号: 2005CCA04300)。江西省自然科学基金项目“焊接快速精密成型过程模型及控制研究”(编号 0650092)。主要从事快速成型领域三维实体图形数据处理和转换相关内容的研究。

1.2 熔焊快速成型制造技术

快速成型(RP)技术经过10多年的发展, 已经逐步实现商业化。但是, 目前快速成型系统的使用仅限于大型公司、研究机构和大学。材料方面, 大多数是非金属材料, 限制了其应用范围的进一步扩展。效率方面, 目前传统的快速成型工艺制造产品原型的时间并不算短, 通常需要几小时、几天、几周不等。

为了克服上述传统快速成型工艺的缺陷, 开发设备简单、使用金属作为成型材料、制造效率高、零件致密度和机械性能好的新型快速成型工艺已成为国际上快速成型技术的研究热点和一个重要的发展方向。

1.2.1 熔焊快速成型制造技术的机理

熔焊快速成型制造技术是快速成型技术的一个分支, 它类似于熔融沉积制造(FDM)工艺。它与传统的焊接——材料与材料之间的连接不同, 制造的零件全部由焊缝组成, 所以也叫做全焊缝金属零件制造技术。因此, 它本质上是采用电弧热使金属熔化与过渡沉积的焊接工艺, 利用 CAD 所提供的实体三维数据控制焊接设备, 采用分层扫描和分层堆焊的方法来制作零件。成型的基本步骤如下:

(1) CAD 造型: 首先在 UG、Pro/E、AutoCAD 等商用 CAD 软件平台上将设计思想转化为零件的 CAD 实体模型或表面模型, 然后对模型进行近似处理, 转化成以表面三角形逼近的 STL 格式文件输出。

(2) 文件处理: STL 格式文件输入快速成型的软件系统进行纠错及修复处理, 重新生成以表面三角形逼近的实体模型。

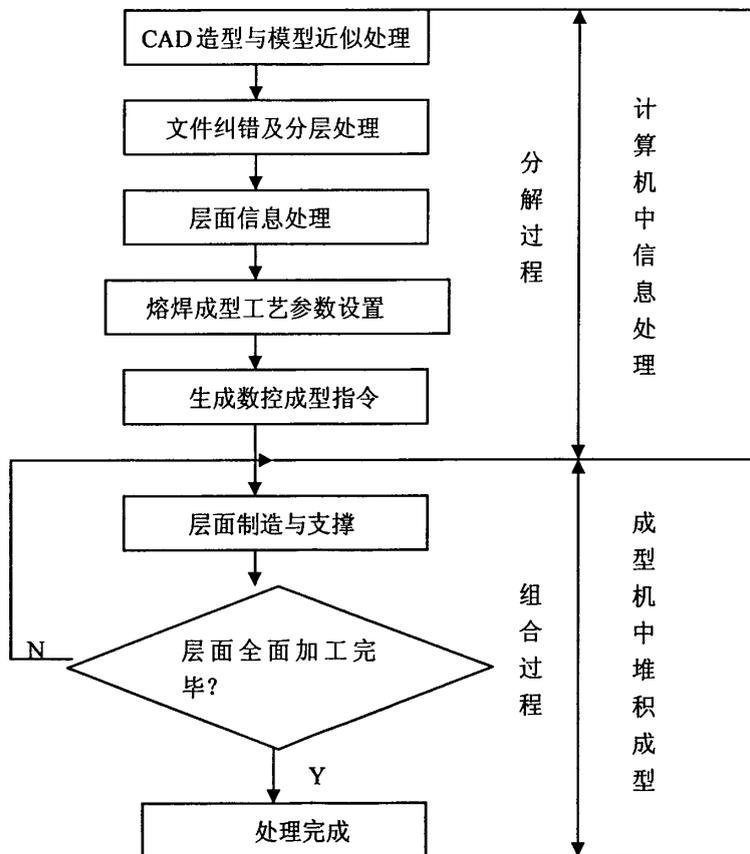
(3) 分层切片: 沿焊接成型的高度方向, 每隔一定间距, 用切片处理软件从 STL 文件中“切出”设定厚度的一系列片层, 以便提取加工截面的轮廓信息。

(4) 路径规划：对零件每一层加工截面的轮廓信息进行处理，形成焊接熔敷时焊炬的运动轨迹。

(5) 输入加工参数：输入焊接熔敷工艺参数，包括焊接电流、电弧电压、焊接速度等。

(6) 生成焊接熔敷加工文件：根据路径规划和输入的加工参数生成焊接熔敷加工文件。

(7) 零件加工：焊接熔敷加工文件输入机器人控制器或成型系统控制器，控制成型系统的工作，逐层进行焊接熔敷，直到所有的截面加工完毕生成三维实体原型或产品零件。其主要成型过程如图 1-1 所示：



1-1 熔焊快速成型的流程示意简图

1.2.2 熔焊快速成型制造技术的分类及特点

焊接快速成型制造技术的分类主要是根据快速成型系统所使用的焊接熔敷

工艺的不同来进行,包括埋弧自动焊(SAW)、熔化极气体保护焊(GMAW)、非熔化极气体保护焊(GTAW)、等离子焊(PAW)、激光焊(LAW)、电子束焊(EBW)等。

熔焊快速成型制造技术的主要特点是:

- (1) 制造成本低;
- (2) 制造形式灵活;
- (3) 能够根据要求直接制造出形状复杂的金属模具,降低模具制造的难度。

1.2.3 熔焊快速成型制造技术的国内外研究现状

熔焊快速成型技术起源于 20 世纪 60 年代末,联邦德国 Krupp Thyssen 钢铁制造公司及瑞士 Sulzer 公司首先开始开发成型焊接工艺,将焊接工艺用于制造全由焊缝金属组成的大型压力容器。研究表明,成型焊接工艺不仅效率高,而且由于焊缝金属具有较高强度和硬度,使得最终制造出的零部件非常耐用。对三维焊接技术进行研究并取得进展的大学和研究小组还有:英国 Nottingham 大学,葡萄牙 Minho 大学,澳大利亚 Wollongon 大学,美国 Souther Methodist 大学、Tufts 大学、Kentucky 大学,以及德国、中国、印度、韩国及新加坡的研究小组。葡萄牙 Minho 大学,澳大利亚 Wollongong 大学的 Fernando Riberio 博士和 JohnNorrish 教授利用机器人熔化焊接工艺实现了零件的三维焊接熔敷成型,其切片软件用 Autolish 语言开发,零件三维造型之后输入切片软件进行切片处理并生成机器人路径,同时输入焊接工艺参数,然后机器人路径代码和焊接工艺参数通过 RS232-C 串口下载到机器人控制器,控制焊接熔敷工艺的进行。

美国肯塔基大学的 Zhang Yu ming 与国内哈尔滨工业大学的王其隆教授合作,采用一种特殊的焊接工艺进行焊接熔敷快速成型制造。并对焊接熔敷成型过程中的 CAD 建模、文件处理、分层切片、加工矢量路径规划、进行了较详细的研究,此外着重研究了焊接过程的热量和质量流动机理及其控制措施,对焊接电弧的引弧、熄弧进行控制,以获得较好焊接熔敷成型效果^[36]。

1998 年 8 月,在美国得克萨斯州奥斯汀召开的第九届实体自由建造论坛上(9th Solid Free Fabrication Symposium),韩国、印度和德国的两个研究小组提出了利用焊接熔敷工艺与铣削法相结合的混合制造新概念^[37]。韩国的研究小组提出焊接和 5 轴数控铣削联合用于金属零件直接快速成型。印度工学院和德国

Fraunhofer 制造技术和自动化学院的研究者组成的研究小组提出焊接和 2~1/2 轴数控铣削联合用于金属零件直接快速成型,其中具有复杂形状的表面由倾斜的铣刀铣削而成^[38]。国内对于熔焊快速成型技术的研究尚处于起步跟进阶段,为数不多的研究者作了初步的探讨。总之,纵观国内外熔焊快速成型技术的研究尚处于基础研究阶段,而且由于焊接过程的复杂性,实现三维焊接快速成型制造技术的商业化尚且需要一段较长的时间。

1.2.4 熔焊快速成型制造技术目前存在的主要问题及发展前景

目前已有焊接快速成型技术所存在的主要问题:

(1) 由于快速成型技术中的软件技术是影响成型表面质量的关键,而软件技术中的 STL 格式切片存在许多缺陷和不足,有待于进一步完善。

(2) 熔焊快速成型制造技术除了几何建模、分层切片、路径规划等快速成型技术的共性问题的研究之外,还必须在焊接工艺方法、熔滴控制、热输入控制、零件的显微组织和性能控制等方面做进一步研究,提高成型零件的质量,使熔焊快速成型制造技术向实用性和商业化方向迈进。对此,有以下几个方面尚待进一步发展完善^[39]:

a) 针对熔焊快速成型的特点设计专用软件,实现分层切片、焊接熔敷路径规划算法的优化及成型过程的计算机仿真。

b) 通过实验手段及数学分析,实现在成型过程中实时调整熔滴大小、焊接电流、送丝速度、焊接速度、焊丝直径等各个焊接工艺参数的优化配置,并建立焊接参数专家数据库。

c) 在焊接熔敷堆积的过程中,增加对各个焊接参数的监控,提高过程参数的采集与整个系统控制的整合力度,为提高零件的精度提供保证。

d) 在多材质焊接熔敷堆积直接成型复杂的功能合金零件方面,需要对软件、工艺方法、设备的设计与改造进行进一步的研究。

1.3 熔焊快速成型中的数据处理相关技术

1.3.1 熔焊快速成型中的数据处理系统概述

能够完整准确描述物体 CAD 模型是快速成型制造(RPM)的前提和关键。

根据工程实际需求,与两种设计理念相对应,CAD模型由两条途径可获得。如图1-2所示。一种是通过逆向设计方法,其优点在于对一些形状不太规则难以用拓扑四边形域分割的物体(如人体骨骼、某些工艺品等)有较好的应用效果。另一种是通过正向设计方法,即从产品的功能要求出发,经过若干设计过程,借助于CAD软件系统(Pro/E, AutoCAD等)直接进行产品CAD造型,得到其三维实体模型。该设计方法比较成熟,对任何复杂程度零件均适用。

1.3.2 熔焊快速成型数据处理的数据格式分类

RP与CAD之间,要求有一种中间数据格式,它既能满足快速成型制造的要求,便于RP系统接受和处理,又便于不同CAD系统生成。用于描述实体模型的数据格式具体有以下几种格式。

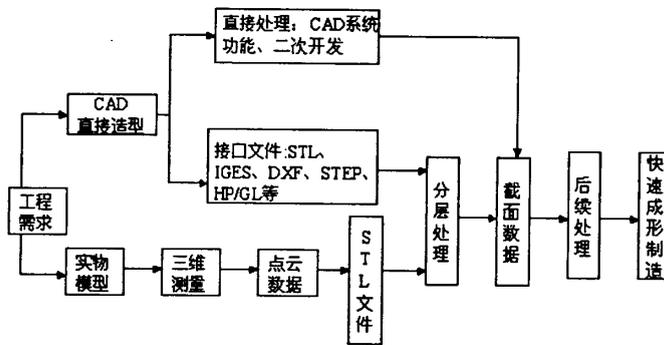


图 1-2 快速成型数据处理系统

(1) IGES(International Graphics Exchange Standard)格式

IGES 是大多数 CAD 系统采用的一种国际图形转换标准,用于支持不同文件格式间的转化。但是,由于不同的 CAD 供应商可能对 IGES 标准有自己不同的解释,从而使各自产生的 IGES 文件有所差异。

(2) HPGL(HP Graphics Language)格式

HPGL 是 HP 公司开发的一种用来控制自动绘图机的语言格式,它已被广泛地接受,成为一项事实标准,一些快速成型系统用 HPGL 来驱动它们的成形头。

(3) STEP(Standard for The Exchange of Product)格式

STEP 是一种正在逐步国际化的产品数据交换标准。目前,典型的 CAD 系统都能输出 STEP 格式文件,有些快速成型技术的研究工作者正试图借助 STEP 格式,直接对三维 CAD 模型进行切片处理,以提高快速成型的精度。

1.3.3 熔焊成型中 STL 文件缺陷检测与修复方法综述

STL 文件仅仅是对 CAD 系统中生成的三维图形的近似, STL 模型自身存在许多不足之处。根据 STL 模型的正确性规则, 对其进行诊断加以修复。本文正是基于这个方向的研究与探索, 从 CAD 系统中 STL 文件转化过程精度控制和误差分析谈起, 试图分析 STL 模型缺陷产生的原因, 提出有效检测与修复 STL 模型缺陷的方法。

在几何造型中, 所描述的物体都是规则物体, 统称为几何模型。一个完整的几何模型应包括物体的各部分几何形状及其在空间的位置, 即几何信息和拓扑信息。在 CAD 系统中生成的三维几何实体模型一般都是规则物体, 但转化为 STL 格式文件后, 几何信息部分丢失, 并且完全不存在拓扑信息, 造成了对造型物体的不完整甚至是错误的描述。因此有必要用规则物体的检测法则, 对 STL 模型进行缺陷诊断, 并作相应的修复。

STL 文件中包含了大量的三角形, 但是是无序的, 散乱分布的, 在对其进行处理时, 将要多次遍历, 直接的影响数据处理软件运行的时间效率。因此必须对 STL 文件进行拓扑化, 以建立起所有三角面片, 边, 点等几何元素间的关系。此外, 在对 STL 模型缺陷诊断和纠错, 直至 STL 模型分层的时候, 也需要毗邻信息的建立。

最初对 STL 模型中的错误完全依赖于熟练技术工人的手工修复。手工修复是一项非常枯燥, 麻烦, 费时的的工作, 通常需要一到数天的时间, 这增加了快速成型的成本, 降低了效率, 使得快速成型花费的时间有时比传统的制造方法还要长, 完全体现不出优势。

2000 年, 杜忠于等人采用邻接图的方法实现对模型错误的诊断, 该软件只提供了移动, 删除, 增加等修改功能帮助用户交互式地纠正各种错误。2001 年张剑峰, 黄因慧等人在分析 STL 模型规则的基础上, 针对 CAD 模型表面三角化后存在的问题, 提出了建立 STL 模型的拓扑结构关系, 对 STL 模型的整体合法性进行检测, 确定缝隙边界, 提出了修复算法, 对其实现了修复, 但其修复效果不尽如人意, 丢失了 CAD 造型设计的原有意图^[6]。2003 年, 交互式修复错误虽然比完全手工修复有了很大的进步, 但是仍然费时费力, 对操作人员要求较高。人们希望能更进一步, 实现完全由软件来完成的错误诊断与修复工作。

1.4 论文的主要工作

本文所要介绍的是一种熔焊成型过程 STL 文件错误检测与修复工具的设计与实现,其中,着重对 STL 文件几何元素拓扑信息的建立算法,STL 文件缺陷诊断修复算法进行了研究。

该系统软件是以 Windows 为开发平台,选择 VC++6.0、OpenGL 作为开发工具,独立开发的一套适用于熔焊快速成型的 STL 模型错误自动诊断和修复的软件系统。论文主要内容包括:

- 1、CAD 系统中 STL 格式文件转换方法的研究;
- 2、基于 OpenGL 技术 STL 数据文件的真实感显示研究;
- 3、STL 文件几何元素拓扑信息的建立算法研究;
- 4、基于熔焊快速成型 STL 数据文件检测和纠错处理算法及实现。

1.5 本章小结

针对快速成型领域对数据处理及其相关技术的需求,提出了课题研究的主要内容,并参照数据处理与转换技术在国内外的发展状况,详细论述了完成课题研究所涉及的理论基础、所采用的解决方法以及课题研究的实际意义。

第二章 金属直接快速成型系统概述

熔焊快速成型技术是 FDM 技术的延伸，通过焊接电弧融化焊丝堆积成型。江西省机器人与焊接自动化重点实验室建立了一套以高速铣床为工作台熔焊快速成型系统，该系统包括快速成型执行机构、焊接质量控制系统以及快速成型处理软件系统。其系统的结构框图如图 2.1 所示。

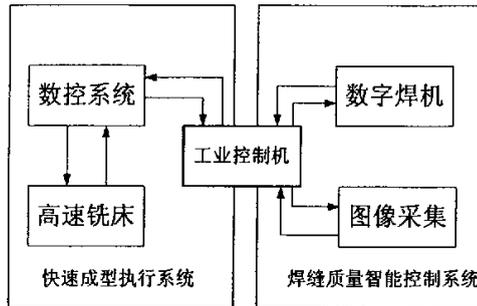


图 2.1 快速成型系统框图

2.1 金属直接快速成型硬件平台

作为一个金属零件快速成型系统，必须要有一个堆积零件的平台，一个进行堆积成型的执行机构。在本课题中，我们拟采用如图 2.2 所示的成型平台。

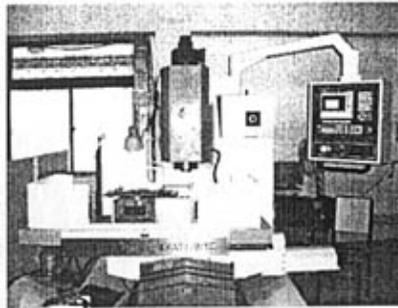


图 2.2 快速成型平台结构

为了保证成型过程的可控性，保证成型件的精度，我们采用数控高速铣床制成三维焊接快速成型平台。将焊炬安装在铣床的 Z 轴之上，通过 X-Y 轴的运动使的工件作水平运动，当焊丝融化堆积在平台上，制造金属零件。这种方法与传统的 FDM 方法很相似，只是在切片处理时要根据焊接的特点设计层高和进行轨迹规划。

在计算机进行切片处理后，将生成数控代码，我们给铣床配置的是华中“世

纪星”HNC-21 数控系统，该系统能够实现四轴三联动。具有开放式体系结构，内置嵌入式工业 PC 机，高性能 32 倍中央处理器。

同时，我们考虑到采用的是焊接电弧熔化焊丝层叠加工的方法，为了保证焊接质量，故要求在焊接过程中，始终保证焊丝在焊枪的前进方向之前，所以，工件需要实时转动，我们在数控铣床上配备的一个旋转工作台，旋转工作台的控制也是由世纪星的数控系统来控制的。

为了保证对成型过程进行实时控制，我们将数控系统作为下位机，而采用 PC 机作为上位机，上位机的作用是实时控制高速铣床每个轴的运动位置，获取每个轴的实时位置，并且将一些控制策略传给下位机。这一功能的实现在硬件上需要华中理工“世纪星”HNC-21/22 数控系统网络数控模块的支持。

2.2 金属直接成型过程数控网络系统的设计

2.2.1 华中世纪星 HNC-21/22 网络数控系统分析

(1) 网络数控系统及其分类

20 世纪 90 年代，国外的数控系统完成了从 16 位机向 32 位机和伺服驱动从直流式向交流全数字式的转变，数控系统体系结构从封闭转向开放，从而使数控系统可充分利用计算机技术的丰富资源，能根据控制对象的要求迅速、灵活地更换软硬件，并能及时吸收新技术，使得数控技术发展步伐加快，开发周期缩短。

网络数控就是通过网络、Internet/Intranet 将制造单元和控制部件相连，或将制造过程所需资源(如加工程序、机床、工具、检测监控仪器等)共享。网络化包括两个方面：内部网络(现场总线网络)和外部网络。内部网络是指数控系统内 CNC 单元与伺服驱动及 I/O 逻辑控制等单元以现场总线网络连接。外部网络指的是数控系统与系统外的其他控制系统或外部上位计算机以网络连接。通过网络实现对设备的远程控制和无人化操作、远程加工程序传输、远程诊断和远程维修服务、技术服务，并提高机床生产率。

(2) 华中世纪星 HNC-21/22 网络数控系统^[7]

华中世纪星 HNC-21/22 网络数控系统通过网卡，串口等网络传输接口，实现了数控系统与系统外部计算机的连接。通过网络实时获取高速铣床每个轴的运动位置，并且，实现了对高速铣床三轴合速度进给，快速进给和主轴速度等速度信

息的输出效率的调节。该系统平台符合 ONC 技术规范，具有开放性。基于 PC 体系结构，以软代硬、简化硬件、易于生产、使用和维修。充分利用已有的华中 I 型数控系统研究成果。硬件通用，软件跨平台。研制的硬件平台-PC 嵌入式 + FPGA 硬件平台，可适用于三个不同的软件平台基于 DOS 的软件平台、基于 LINUX 的软件平台、基于 WINDOWS 的软件平台。

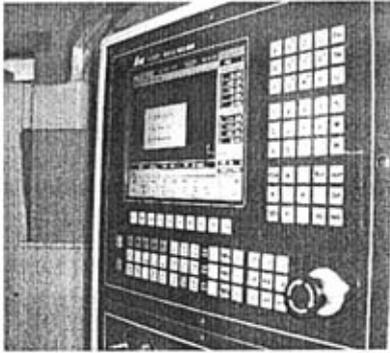


图 2-3 HNC-21/22 数控系统控制操作面板

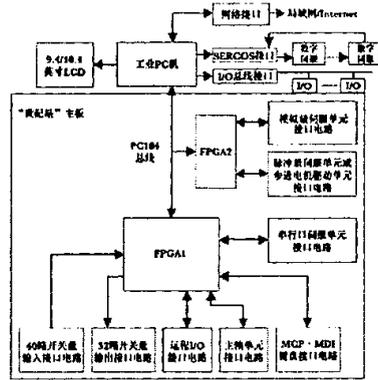


图 2-4 HNC-21/22 CNC 装置的硬件平台

(2) 数控系统的硬件体系结构

HNC-21/22 数控系统硬件平台基于 PC 体系结构，如图 2-4 所示。系统内部采用数字量、脉冲量、模拟量标准接口或现场总线连接各功能部件，系统与外部采用网络化的连接方式，可与 CAD/CAM、FMC/FMS 和企业 Intranet 集成，实现数控系统的网络化。

HNC-21/22 数控系统充分利用工业 PC 机的接口功能强大的特点，采用 10.4/9.4 英寸真彩 TFT 液晶显示屏，8MB Flash ROM(可扩展至 72MB)程序断电存储，16MBRAM (可扩展至64MB)加工缓冲区，配备硬盘和软盘，可装载和存储大量加工程序，具有巨量程序加工能力，不需DNC，可直接加工高达100MB的G代码程序，利用以太网、RS232等接口，轻松实现机床联网。

(3) 数控系统的软件体系结构

HNC-21/22 CNC装置的软件结构如图2-5所示。图中虚线以下的部分称为底层软件，它是HNC-21/22 CNC装置的软件平台。虚线以上的部分称为应用层软件（过程层软件），它包括编辑程序、参数设置、解码程序、PLC管理、MDI、故障显示等与用户操作有关的功能模块。对于不同的数控装置，其功能的区别都在

这一层，或者说功能的增减均在这一层进行，各功能模块均可通过NCBIOS与底层进行信息交换，从而使该层的功能模块与系统的硬件无关。

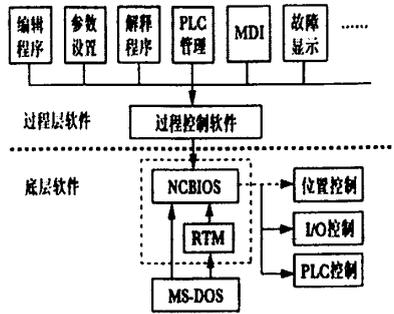


图 2-5 HNC-21/22 CNC 装置的软件平台

2.2.2 数控网络系统客户端软件接口研究

基于华中世纪星 HNC-21/22 数控网络系统采用的是典型客户机/服务器（C/S）网络结构，在 NetBIOS 网络通讯协议规则下，实现了数控系统与外部计算机的实时通讯。该系统以数控机床作为客户端，以工控机作为服务器端。其中，在 PC 机上，对其数控系统开放的机床各轴位置，主轴速度，快速点定位合速度，进给合速度参数实现实时的显示，并对速度输出效率给予实时的修调。

网络传输芯片是数控网络系统客户端软件的核心结构。它提供了网络传输数据所必须的基本功能，具体功能有检测网卡的状态、添加网络名字、删除网络名字、呼叫和监听网络连接、发送和接收数据、挂起网络会话、取消网络会话等。

网络传输芯片是一个 C++对象类 CNCB，提供的操作方法和数据成员如下：

```
class CNCB
{
private:
    NCB m_NCB; //NCB 数据块
public:
    // constructor
    CNCB(); //构造函数
    // helper function
    UCHAR EnumerateLANA(LANA_ENUM *lenum); //检测本地网卡
```

```
void ClearNCB();//将 NCB 数据块清零
UCHAR GetLSN();//取得当前的会话号
UCHAR GetNUM();//取得当前添的名字号
WORD GetLength();//取得 NCB 数据块的长度
void Fill(CNCB ncbSource);//用指定的 NCB 数据块填充当前的 NCB 数据块
UCHAR GetCommand();//得到当前的网络命令
// Name Management Services
UCHAR AddName(PSTR pName);//添加指定的网络名字
UCHAR AddGroupName(PSTR pName);//添加指定的网络组名
UCHAR DeleteName(PSTR pName);//删除指定的网络名字
// Data Transfer Services
UCHAR Call(PSTR pWe, PSTR pTheOther, UCHAR wSendTO=0, UCHAR
wRecvTO=0);//呼叫被叫方
// fill in parameters later
UCHAR Listen(PSTR pWe, PSTR pTheOther, UCHAR wSendTO=0, UCHAR
wRecvTO=0);//处于监听状态
UCHAR Hangup(UCHAR wSessionNumber);//挂起指定的会话联
// Connectionless Data Transfer
UCHAR Cancel();//
UCHAR Send(UCHAR wSessionNumber, LPSTR lpPacket,  UINT wLength,
void (CALLBACK *CallBackFunction)(PNCB pncb));
//发送数据包
UCHAR SendDatagram(UCHAR wSessionNumber, LPSTR lpPacket,  WORD
wLength);//发送数据报
UCHAR Receive(UCHAR wSessionNumber, LPSTR lpPacket,  UINT wLength,
void (CALLBACK *CallBackFunction)(PNCB pncb));
//接收数据包
UCHAR ReceiveDatagram(UCHAR wSessionNumber, LPSTR lpPacket,
WORD wLength);//接收数据报
// General Purpose Services
```

```
UCHAR Reset(UCHAR wSessions,  UCHAR wNCBs);//  
UCHAR GetAdapterStatus(PSTR pName);//取得当前的网卡状态  
UCHAR GetSessionStatus(PSTR pName);//取得指定会话号的状态  
}
```

2.2.3 网络数控系统服务器端软件功能的设计与实现

I 软件运行环境配置要求

在 Win98 系统，特别是其本地连接模块的支持下，网络配置如下：

(1)TCP/IP 中的 NETBEUI 协议；

(2)IPX/SPX 中添加服务；

II 软件系统设计与实现

1. 软件实现之数据结构

(1). 客户端数据结构

它是从客户端向服务器端发送的数据，也就是 cnc 系统向 win 端发送的数据，即 win 端能够向 CNC 索取到的有效信息结构。在网络传输模块测试程序中读取的数据就是一连串按照该结构排列的二进制数据流。读取的方法就是用指针变量按位读取显示在应用程序中。并且，该数据格式为基本的数据格式，具有开放性，可扩展性。

(2). 服务器端数据结构

它是从服务端向客户端发送数据，也就是 win 向 cnc 发送数据，即 PC 机向 CNC 系统能够反馈的输出量。

在数控系统默认的初始状态是进给修调，快移修调，主轴进给修调三个机床速度参量均为零。在程序测试的第一操作就是需要修改三个速度参数为非零状态。具体方法就是将数值写入到服务器端数据结构的相应变量中。

(3). 发送和接受数据结构

该结构用于网络进程控制所用的变量，最重要的变量为 NCB 结构，对 NetBIOS 调用主要是对该结构进行操作。

```
struct SendAndReceiveDataNetBios  
{  
    HANDLE events;    //事件句柄
```

```

char RemoteName[NCBNAMSZ];//被叫方的名字
int NetworkControlFlag; //网络是否连上的标志
int SendReady; //发送和接收的标志字
int ReceiveReady;
Data_Client ReceiveMessageData;
Data_server SendMessageData;
NCB SendNCB; //用于添写网络服务的 NCB 结构
NCB ReceiveNCB; //用于添写网络服务的 NCB 结构
};

```

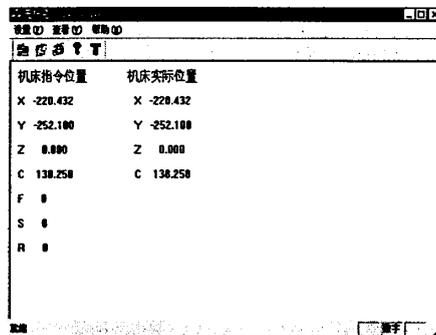
2、数控网络系统基本功能实现

软件基本功能是建立在畅通的网络连接的基础上的。在程序调试过程中，我们通过各种网络状态来测试该网络传输芯片所具有的对网络故障诊断的功能。

在良好的网络环境之下，系统能够根据客户端的数据结构将其提供的数据在服务器端进行监测，并将对机床各轴位置和速度的监测结果用于焊接快速成型的加工中特殊点（起弧和熄弧）的控制。

III、实验测试结果及其分析

(1) 机床未加工状态下，各轴位置信息及速度信息的获取



机床指令位置	机床实际位置
X -220.432	X -220.432
Y -252.100	Y -252.100
Z 0.000	Z 0.000
C 130.250	C 130.250
F 0	
S 0	
R 0	

图 2-7 机床位置信息静态显示

说明：机床指令位置指的是，数控指令在插补算法下应该到达的位置点；机床实际位置指的是，机床实际所在的位置。F 为进给修调，S 为快移修调，R 为主轴速度修调。在机床未加工时，三个速度值都为零。

(2) 机床加工过程中，各轴位置信息及速度信息的获取能够与数控系统控制面版同步实时显示数控系统发送的数据包信息。

如图 2-7 所示：执行程序段 GXX X-300Y-200Z-150C250

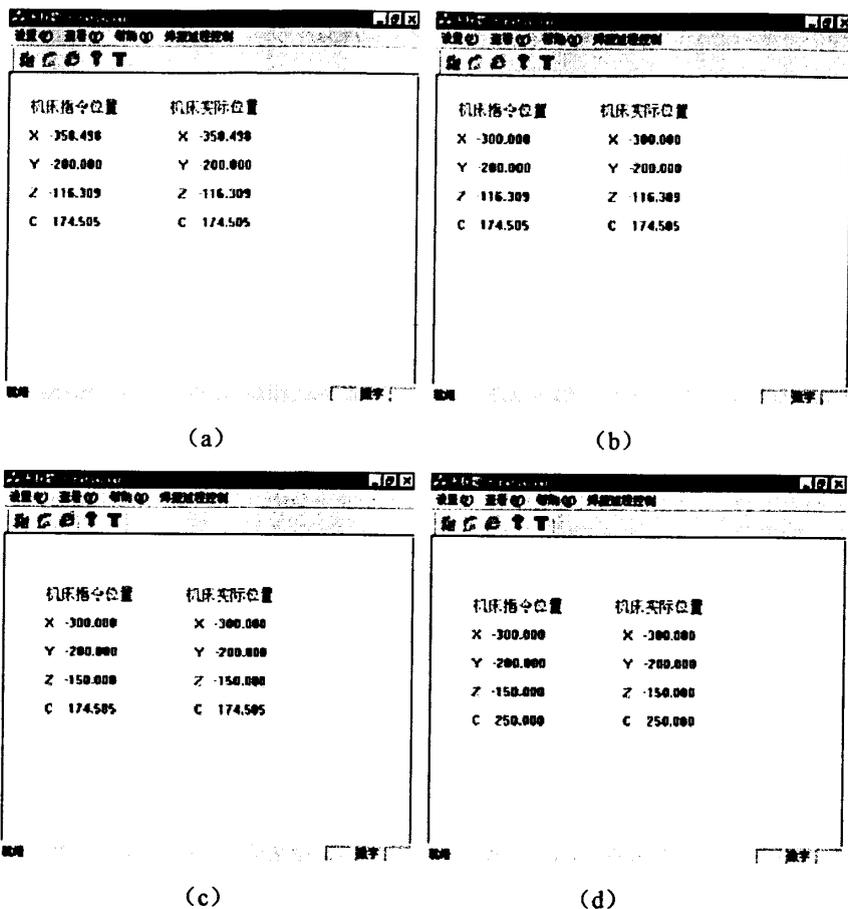


图 2-7 加工过程机床位置信息动态显示

说明：在手动加工时，执行上述程序段。对应图 2-7 看出：机床快速各轴进给的次序分别是 X 轴，Z 轴和 C 轴。

2.3 本章小结

本章介绍了金属直接快速成型系统硬件平台，分析了系统采用的华中世纪星 HNC-21/22 数控系统，根据熔焊成型过程中在特殊位置点(起弧，熄弧点)焊接工艺调整的确具体需求，采用 C++语言，在 VC++环境下开发了数控网络服务器端通讯软件系统。

该系统能够通过工业 PC 机对数控机床各坐标轴的位置，快速进给，主轴速度等信息实时显示，并且在速度参数上能够进行修改，实现对数控机床监控。系统主要的实现功能在于：

1. 采用 C/S 两层架构，将服务器端和客户端用双绞线进行连接。并对服务器端软件进行了 NetBIOS 协议等通讯环境配置，为数控网络通讯实现提供基础。

2. 网络通讯服务器端软件的编制，实现了与下位机端的数据传输。即数控加工中各轴在工件坐标系下的位置信息的接收和对进给修调，快移修调和主轴速度修调等机床速度信息的修改发送。数据包传输的速率可以达到局域网网内数据包传输速率。

第三章 基于 OpenGL 技术 STL 模型的真实感显示研究

对 STL 这样一个海量数据文件进行的诊断, 修复和分层等处理, 其处理结果的显示与观察是必不可少的, 观察包括旋转、区域局部放大等。对于 STL 数据模型的可视化, 三维真实感显示, 计算机图形学中有成熟的算法。但是如果直接按照这些算法从底层开发 STL 模型的显示和变换系统, 效率将十分低下。OpenGL 的出现不仅使人们可以高效地创建光线跟踪的高质量真实感图像, 而且速度比光纤跟踪法快一个数量级。

3.1 OpenGL 技术

OpenGL 是一个功能强大的 3D 图形库。具有 3D 图形原语的绘制命令, 并且可在这些原语的基础上, 构造更复杂的 3D 物体;它是一个三维图像开发工具, 具有高效、简便、功能完善、具有跨平台特性, 用来精确显示 STL 模型是合适的。

OpenGL 最初是 SGI 公司为其图形工作站开发的可以独立于窗口操作系统和硬件环境的图形开发环境, 其目的是将用户从具体的硬件系统和操作系统中解放出来, 可以完全不去理解这些系统的结构和指令系统, 只要按规定的格式书写应用程序就可以在任何支持该语言的硬件平台上运行。它已经成为国际上许多大公司的标准工具^[8]。

3.1.1 图形硬件的软件接口

图形硬件的软件接口指的是: 开发基于图形加速器的图形应用程序需要使用的接口技术。早期安装该软件, 对系统硬件有所要求, 只在图形工作站上使用。随着计算机性能的不断提高和完善, 在普通的 PC 机上也能够使用。和 OpenGL 一样, 使用广泛的图形软件接口还有 Active X 技术。

3.1.2 OpenGL 技术原理概述

OpenGL 中的所有图元最终都是用顶点来描述的, 定点数据除了包含顶点坐标之外, 还包含法向量 RGB 颜色值、纹理坐标和其他的数据。求值器将这些数据转换为顶点并对每一个顶点执行相应的顶点计算, 随后光栅化为片元, 并在执行完每个片元操作以后将光栅化后的数据送入帧缓存。对于像素数据(包括像素、

位图和图像), 其处理方式和顶点数据的处理方式是不同的, 从内存中读入的像素数据经过像素操作(缩放、偏置和映射)后, 将结果数据限制在一定的范围, 然后送入纹理内存中进行组装, 供纹理映射使用, 或者将这些结构数据光栅化为片元后, 执行每个片元的操作, 最后送入帧缓存; 而从帧缓存中读出的数据经过像素操作后被送入到处理器的内存中^[8]

3.1.3 OpenGL 图形操作步骤^[8]

- (1) 利用几何图元构造物体模型, 然后建立物体的数学模型;
- (2) 在三维空间中选择合适的位置放置物体, 并确定观察场景的最佳视点;
- (3) 计算场景中所有物体的颜色。物体的颜色可能来自于程序的设定, 如果在场景中使用了光照和纹理映射, 那么观察到的场景中物体的最终效果取决于光照条件和纹理映射方式;
- (4) 把场景中物体模型的数学描述及颜色信息转换到计算机屏幕的像素点。

3.2 基于 OpenGL 技术 STL 三维图形真实感显示研究

上节讲到, OpenGL 工具所提供的功能性三个函数库的定义和使用方法是随着软件开发平台, 开发工具和开发环境的变化而变化的。本系统软件是在 WINDOWS 平台下, 运用 C++6.0 开发工具, 在 C++ 开发环境下实现的。

3.2.1 OpenGL 应用程序框架初始化

早期 OpenGL 对运行的与图形相关的硬件环境有较高的要求, 只有在图形工作站才能运行平稳。现在随 PC 机性能增强, 在一般的微机上也能运行良好。笔者认为, OpenGL 技术相对 C++ 开发工具来说, 只是 C++ 语言开发的可以实现图形光照, 深度, 着色等渲染效果的一种特殊工具。该工具对应用程序开发环境要求建立三个动态链接库的调用机制。这样, 才能使用 OpenGL 提供的数据类型和功能强大的实现函数。要使用工具 OpenGL, 有以下工作需要完成^[9]:

第一, 要建立应用程序与 OpenGL 的联系纽带, 需在项目文件环境设置的 Link 域中, 加入库文件 OpenGL32.lib, glu32.lib, glaux.lib, 在由 VC Application Wizard 生成的视图类的头文件中, 加入 gl.b, glu.h, glaux.h。

第二, OpenGL 数据最终是以像素格式绘制在窗口中, 所以需要设置显示

设备的像素格式。通过填写结构 PIXELFORMATDESCRIPTOR 来设置显示设备描述表 DC(Device Context)的像素格式属性,并调用函数 ChoosePixelFormat 来择与以上所定义的位图格式最接近且 DC 支持的像素格式,然后用 SetPixelFormat 将其设置成 DC 的像素格式,从而定义了 OpenGL 作图的各项属性。此部分内容可在视图类的 WM CREATE 消息处理函数中完成。

第三,在本系统中,要实现光照,着色,深度等渲染效果,OpenGL 需使用渲染描述表 RC (Rendering Context)完成三维图形的绘制,因此需要建立与设备描述表 DC 相对应的一个 RC,并使其为当前线程所拥有,它通过函数 wglCreateContext 和 wglMakeCurrent 来实现。

第四,调用 OpenGL 函数绘制图形,它包括定义观察区域、设置投影模式、建立光照模型、作图、图形平移、旋转等等。可根据应用需要调用相应 OpenGL 函数,在视图类的 WM - PAINT 或者 WM - DRAW 消息处理函数中实现。

第五,图形绘制完毕以后,应断开当前线程与 RC 的联系,并删除 RC,可在视图类的 WM DESTROY 消息处理函数中完成。

(1) 添加 windows 的消息相应函数: wm_create, wm_destory, wm_size

(2) 在 view 类中添加两个保护成员函数,分别是: void init()、BOOL bSetupPixelFormat();

(3) 在 view 中添加两个共有成员: CClientDC* m_pDC; CRect m_oldRect; 并且在构造函数中初始化 m_pDC=NULL;

(4) 编辑成员函数: bSetupPixelFormat()如下:

```

BOOL CStepinGView:: bSetupPixelFormat()
{
    static PIXELFORMATDESCRIPTOR pfd={
        sizeof(PIXELFORMATDESCRIPTOR),
        1,
        PFD_DRAW_TO_WINDOW|
        PFD_SUPPORT_OPENGL,
        PFD_TYPE_RGBA,
        24,

```

```
    0, 0, 0, 0, 0, 0,
    0,
    0,
    0,
    0, 0, 0, 0,
    32,
    0,
    0,
    PFD_MAIN_PLANE,
    0,
    0, 0, 0,
};
int pixelformat; 选择像素格式
if((pixelformat=ChoosePixelFormat(m_pDC->GetSafeHdc(), &pfd))==0
)
{
    MessageBox("ChoosePixelFormat failed");
    return FALSE;
}
if(SetPixelFormat(m_pDC->GetSafeHdc(), pixelformat, &pfd)==FALSE)
{
    MessageBox("SetPixelFormat failed");
    return FALSE;
}
return TRUE;
}
(5) 编辑成员函数: Init();
void CStepinGView:: Init()
{
    PIXELFORMATDESCRIPTOR pfd;
```

```

int n;
HGLRC hrc;
m_pDC = new CClientDC(this);
ASSERT(m_pDC!=NULL);
if(!bSetupPixelFormat())return;
n = :: GetPixelFormat(m_pDC->GetSafeHdc());
:: DescribePixelFormat(m_pDC->GetSafeHdc(), n, sizeof(pfd), &pfd);
hrc = wglCreateContext(m_pDC->GetSafeHdc());
wglMakeCurrent(m_pDC->GetSafeHdc(), hrc);
GetClientRect(&m_oldRect);
glClearDepth(1.0f);
glEnable(GL_DEPTH_TEST);
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
}

```

(6) 编辑 view 类中的 precreatewindow 函数:

```
cs.style |= WS_CLIPSIBLINGS|WS_CLIPCHILDREN;
```

OpenGL 有自己的像素格式, 只用 OpenGL 的客户区才能接受 OpenGL 绘制, 因此一般应该有这两个风格设置。

(7) 编辑 OnDestroy()函数: OnDestroy()函数的作用是删除着色描述表以及绑定在着色描述表上的设备描述表。

```

void CSecondwkView:: OnDestroy()
{
HGLRC hrc;
hrc = :: wglGetCurrentContext();
:: wglMakeCurrent(NULL, NULL);
if(hrc)
:: wglDeleteContext(hrc);
if(m_pDC)
delete m_pDC;
}

```

```
CView:: OnDestroy();  
}
```

(8) 在 OnCreate()函数中调用 Init():

```
int CSecondwView:: OnCreate(LPCREATESTRUCT lpCreateStruct)  
{  
if(CView:: OnCreate(lpCreateStruct) == -1)  
return -1;  
Init();  
return 0;  
}
```

说明：创建的像素格式和着色描述表必须通过通过响应 Windows 消息 WM_CREATE 才能使 OpenGL 内核得到通知。因此，必须在 OnCreate()函数中调用 Init()。

(9) 编辑 OnSize()响应函数：

```
void CSecondwView:: OnSize(UINT nType, int cx, int cy)  
{  
CView:: OnSize(nType, cx, cy);  
if(cy>0)  
{  
if((m_oldRect.right>cx)||(m_oldRect.bottom>cy))  
RedrawWindow();  
m_oldRect.right=cx;  
m_oldRect.bottom=cy;  
glMatrixMode(GL_PROJECTION);  
glLoadIdentity();  
glFrustum(-1.0, 1.0, -1.0, 1.0, 0.0, 7.0);  
glViewport(0, 0, cx, cy);  
}  
}
```

(10) 建立绘图函数 DrawScene(); 属于保护成员函数, 负责具体绘制图形; 并在 OnDraw() 函数中调用 DrawScene() 函数。DrawScene() 函数是用来编写 OpenGL 绘制语句。

3.2.2 STL 模型真实感显示

OpenGL 提供了图形显示的线框和渲染两种显示模式。线框显示适合网格化中 STL 模型的显示和观察。此时, 根据网格的疏密程度, 能辨别出不同数目的三角形近似实体模型。但, 只有渲染才具有深度显示, 故通过 STL 模型实体表面的光滑程度能够判断 STL 文件精度过高或过低。在今后的面片错误显示和错误修复中, 也需要模型的真实感显示。

(1) 空间三角形面片显示

OpenGL 在函数库中, 提供了三角形面片渲染显示的方法。在本系统中, 读入 STL 文件时, 采用动态数组模类 CtypedPtrArray<Cobject, Centity>m_TriChip 来动态存储 STL 模型或者装配中的零件模型, 其三角面片数目由 GetSize 函数来获取。这样, 通过一个一维循环语句就能实现 STL 模型的动态显示。

(2) STL 数据模型旋转和缩放

在快速成型 STL 模型的错误检测与修复系统中, 对文件缺陷的准确判断和真实显示是 STL 模型错误修复的重要步骤。通过改变视点(观察点)的位置来实现 STL 数据模型的旋转。至于 STL 数据模型的缩放, 只是改变观察范围和可视空间的大小。一般来说, 以上两者操作需要结合 WM_LBUTTONDOWN 和 WM_LBUTTONUP 两个消息响应函数来实现。

(3) STL 模型光照, 着色和材质显示

要使 STL 格式实体实现真实感图形显示, 建立光照模型是必须的。为增强真实感图形显示的效果, 也可以加入材质属性的定义, OpenGL 对此均提供了相应的功能强大的函数。在 OpenGL 中, 光照模型由环境反射光(Ambient Light)、漫反射光(Difuse Light、镜面反射光(Specular Light)等组成, 它们可模拟真实的光照效果。函数 glLight 用于设置光源特性, 其参数 GL POSTION 指定光源位置, 参数 GL_ AMBIENT, GL DIFFUSE, GL_ SPECULAR 分别定义 RGBA 颜色模式下的环境反射光、漫反射光、镜面反射光属性, 在启动已设置好的光照以后, 一个简单光照模型便建立起来。在光照中还应加入消隐处理, 以确定可见面, 隐

藏不可见面。

材质颜色是通过材质对 R, G, B 的近似反光率来定义的,如同光源模型,也分成环境反射、漫反射、镜面反射成分,函数 `glMaterial` 实现了材质的设置。材质属性与光源特性的结合就是观察的最终效果。

3.3 本章小结

本章在系统介绍 OpenGL 技术基本原理基础上,结合 VC++开发环境,实现了以下功能:

(1) 设计 OpenGL 与 VC++6.0 的接口。这是一个相对对立的功能模块,可以方便地在 VC++6.0 编译环境中实现移植。

(2) 研究成型零件 STL 文件的三维真实感图形显示系统,使之能够在光照,纹理等渲染环境下,对图形进行旋转、缩放、平移等操作。

第四章 基于拓扑关系之 STL 模型缺陷自动检测与修复方法

为了得到实体的三维表面数据，需要商用的 CAD 系统产生符合成形精度要求的文件格式程序。但不同的 CAD 系统所采用的内部数据格式是不同的，RP 系统无法一一适从，这就要求有一种中间数据格式，它既能满足快速成型制造的要求，便于 RP 系统接受和处理，又便于不同 CAD 系统生成。这种中间数据格式由美国 3D SYSTEMS 公司于 1988 年开发成功，被称为 STL 文件，由于其简单被工业界认为是目前的“准标准”。该公司 1988 年制定了一个接口协议，随后，各商用 CAD 软件包，都提供从 CAD 实体数据模型转换成快速成型技术输入数据模型的接口。

STL 文件是对 CAD 实体模型或曲面模型表面进行三角形面片离散化得到的一种由许多带有法向量的小三角形面片拟合的三维模型。数据格式简单，读取优化处理方便，最重要的是它能对各种 CAD 造型系统的内部文件格式兼容，并且也能和各种快速成型系统实现通讯，成为一个在 CAD 系统和 RP 系统之间默认的标准。

4.1 STL 模型数据转换方法的研究

目前快速成型系统不能容易的直接接收 CAD 系统制作的实体文件，必须在快速成型系统的数据处理软件进行分层之前，将其转化为接口文件 STL 文件，对其进行面化处理。即用数学方法上的一系列的三角形面片对 CAD 模型表面近似处理。在这一转化过程中，数学处理的精度控制决定了导出的 CAD 模型的形状和尺寸精度，进而决定了成形件的形状和尺寸精度。这是本节研究的一个内容。

STL 文件中包含了大量的无序的，散乱分布的三角形，在对其进行缺陷诊断与修复处理时，将要多次遍历直接的影响数据处理软件运行的时间效率，必须对 STL 文件进行拓扑，以建立起所有三角面片，边，点等几何元素间的关系。对于拓扑信息建立算法的研究是本章的另一个内容。

4.1.1 CAD 系统中 STL 转化过程中精度控制和误差分析

快速成型中采用技术成熟的 STL 模型作为研究对象。用 STL 格式的数据模

型作为 CAD 几何模型与成型机所需数据转换的接口。STL 文件的数据格式是采用小三角形来近似逼近三维实体模型的外表面，小三角形数量的多少直接影响着近似逼近的精度。显然，精度选定越高，选取的三角形应该越多。但是，三角形数量的增多会引起计算机存储容量的加大，同时带来切片处理时间的显著增加，另外，过高精度会产生三角形顶点分离，分层后截面的轮廓会产生许多小线段，不利于焊枪和数控机床的扫描运动，导致低的生产效率和表面不光洁。

4.1.2 表面网格化处理对成形件精度影响

用三角形面片近似完整的三维图形，称为网格化处理。一般待成形制品的结构中都含有曲面，在 CAD 造型中，用边界模型能精确的描述制品的形状和尺寸的精度。面化处理以后，原来光滑的曲面模型就变成数学中不光滑的多面体模型，使模型的形状和尺寸精度都降低。近似处理后生成的 STL 模型的后续处理都将在失真的数据基础上进行，必然会降低成形件的形状和尺寸精度。

一个三维曲面体三角化过程一般有两种情况：下面以一个高度为 Z 的短圆柱体单元为例予以说明：

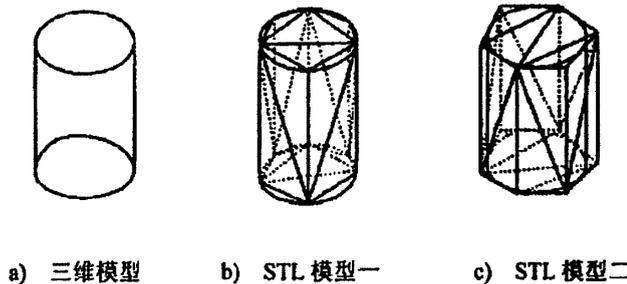


图 4-1 用三维曲面三角化近似同一圆柱

以上两种网格化的结果，是由于弦差选取不同造成的。STL 模型一采用内弦来表示圆弧，很显然必然使 STL 模型所表示的实体小于原三角形，形成负的偏差，如图 4-1 b)；STL 模型二采用外切线来近似圆弧，则使 STL 模型所表示的实体大于原三角形，形成正的偏差，如图 4-1 c)。在 CAD 系统中导出 STL 格式文件时，以上两种情况均存在。

4.1.3 网格化近似处理误差分析

当模型的一个面与快速成型系统制造方向成一定角度时，给定误差不再是切

片的最大误差，而是给定误差的好几倍，如图 4-2，这样在分层处理及后处理时已经在失真数据的基础上进行，必然影响成形工件的形状和尺寸精度。



图 4-2 网格化的给定误差与实际误差

下面以半径为 r 同一个圆在不同精度下对其三角面化后所产生的误差来分析：用 5 个三角面片来划分同样一个圆，此相当于用一个七边形来拟合一个圆。

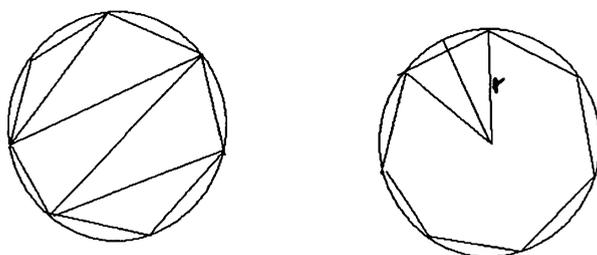


图 4-3 用 5 个三角形来近似一个圆的情形

这样可计算得到拟合后的最大误差为： $\delta_1 = r(1 - \cos \pi / 14)$

其相对误差： $\delta_2 = r(1 - \cos \pi / 14) / r = (1 - \cos \pi / 14)$

再用 7 个三角面片来划分一个圆，此相当于用九边形来拟合一个圆：

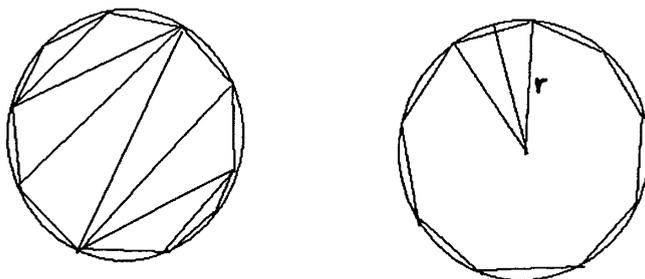


图 4-4 用 8 个三角形来近似一个圆的情形

这样可计算得到拟合后的最大误差为： $\delta_1 = r(1 - \cos \pi / 20)$

其相对误差： $\delta_2 = r(1 - \cos \pi / 20) / r = (1 - \cos \pi / 20)$

由比较结果可知同一个实体用不同个数的三角形面片来划分，得到的误差相对值也不一样，所以确定一个三维实体大致由多少面片来拟合对于分析 STL 文件转化过程中的误差分析有很重要的意义。选取合适的转化参数，就是以下需要研究的内容。

4.1.4 三维造型软件导出 STL 格式文件转换参数及其选定

一般的，精度要求越高，网格化的三角形应该越多。但是，三角形数量的增多会引起计算机存储容量的加大，同时带来切片处理时间的显著增加，另外，过高精度会产生三角形顶点分离，分层后截面的轮廓会产生许多小线段，不利于焊枪和数控机床的扫描运动，导致低的生产效率和表面不光洁。

在从 CAD/CAM 软件导出 STL 文件时，选取的精度指标和控制参数应该根据 CAD 模型的复杂程度及所用的成型工艺要求进行综合考虑。不同的 CAD/CAM 系统输出 STL 格式，但其精度控制参数均为控制角度和网格化长度(弦高)。弦高为实际曲面与近似平面的高度差。控制角是两平面切平面的夹角。但最终反映 STL 文件逼近 CAD 模型的精度指标表面上是空间三角形的数量。

为了得到合适的弦高，我们采用数学分析方法，对一些相对简单但较为典型的曲面模型为实例，进行了弦高和控制角的最优设定研究。选取的模型分别为球体-圆柱-圆锥。通过对三维造型软件 Pro/E 对直径为 100mm 的球，直径 100mm 的圆柱，直径 100mm 高 100mm 的圆锥进行分析。

(1) 当控制角固定，设定不同弦高时，对应导出由不同数目的三角形面片近似的实体模型。如表 4-1，弦高的设定从 0.01 开始，以步长 0.01 递增，直到最终的弦高取到 0.2，与之相应的有不同 STL 文件网格化三角形面片数与之对应。其中，在造型软件里头，对弦高的选择有一个取值范围：必须 0.0173 和 173.2051 之间。因此弦高的分析取值从 0.017 开始。

数据分析：

由 EXCEL 数据处理得到如图 4-5 所示的坐标表示情形：行坐标为设定弦高的相对值，纵坐标则是相对弦高设定值下所得到的 STL 格式文件的零件模型网格化面片数目。

表 4-1 网格化实体模型数据表

弦高	直径, 高为 100mm 柱 体网格化面片数目	直径为 100mm 球体 网格化面片数目	直径, 高为 100mm 锥 体网格化面片数目
0.017	476	13860	238
0.02	444	12540	222
0.03	364	8460	182
0.04	316	6396	158
0.05	284	5180	142
0.06	252	4352	130
0.07	236	3596	118
0.08	220	3132	110
0.09	212	2912	106
0.10	196	2496	98
0.11	188	2300	94
0.12	180	2112	90
0.13	172	1932	86
0.14	164	1928	82
0.15	164	1760	82
0.16	156	1596	78
0.17	156	1596	78
0.18	148	1440	74
0.19	140	1440	70
0.20	140	1440	70

纵坐标是不同弦高下得到的球体 STL 文件面片数目, 可以看作是转换成 STL 过程中转化效率的一个近似表征量。所得到的 STL 文件越大, 所需要的用来逼近实体模型的三角面片数目越多。这样, 在模型处理的内存开销上越大, 时间越长处理速度越慢, 效率就会明显降低; 横坐标是转化过程中的控制参数弦高的不同取值, 可以看作是控制精度的一个似表征量, 当弦高越小, 那么转化的精度九会越高; 弦高取得越大时, 反之。

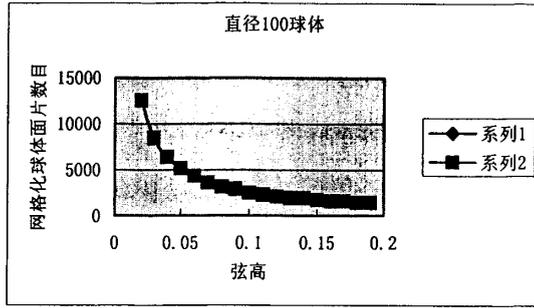


图 4-5 直径 100mm 球体

上述图表中，以中间一段弦高误差在 $[0.06, 0.1]$ 内作具体分析：将离散点分段按横坐标即弦高长度，将曲线划分为四段 $[0.06, 0.07]$ $[0.07, 0.08]$ $[0.08, 0.09]$ $[0.09, 0.10]$ 在每小段进行直线拟合，则相应的拟合直线斜率为 16 个/0.01，16 个/0.01，8 个/0.01，16 个/0.01，由此说明：在以精度降低一个步长的前提下，效率一直在提高，但提高的幅度在前三段一直是非增的，而到了第四段又反而提高，所以从精度合效率的角度考虑，第三段合第四段的交界值就是比较好的。因此范围参数取值在以 0.09 为中心的一个小区间内取值。

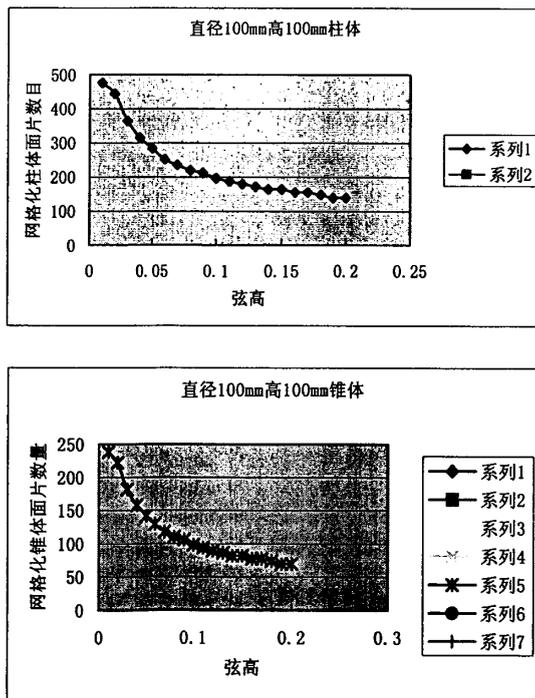


图 4-6 直径 100mm 高 100mm 柱体锥体

同样的方法分析直径 100mm 高 100mm 的圆柱和圆锥，在弦高 0.06~0.1 这

一区间,按横坐标即弦高长度,将曲线划分为四段[0.06 0.07] [0.07 0.08] [0.08 0.09] [0.09 0.10]在每小段进行直线拟合,则相应的拟合直线斜率为下表:

表 4-2 拟合直线斜率

拟合直线斜率	[0.06 0.07]段	[0.07 0.08] 段	[0.08 0.09] 段	[0.09 0.10] 段
球体	756 个/0.01	464 个/0.01	220 个/0.01	516 个/0.01
圆柱	16 个/0.01	16 个/0.01	8 个/0.01	16 个/0.01
圆锥	12 个/0.01	8 个/0.01	4 个/0.01	8 个/0.01

数据分析: 在以精度降低一个步长的前提下,效率一直在提高,但提高的幅度在前三段一直是非增的,而到了第四段又反而提高,所以从精度效率的角度考虑,第三段合第四段的交界值就是比较好的。

由以上三个典型模型分析得出,最佳弦高取 0.09,作为本论文一个分析结果。

以下以某一零件为例,从 Pro/E 系统中输出 STL 文件(网格化)精度,面片数量,文件大小比较:

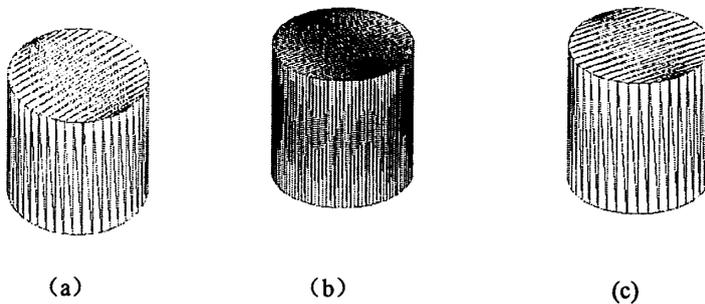


图 4-7 CAD 系统中 STL 格式文件转换

上图中 a, b, c 控制角在 0.5 的条件下,对应弦高 0.06, 0.09 和 0.1 的网格化模型。网格化处理分析可看出:

(1) 曲率大的面(如倒角),精度不同的输出其图形明显不同,而这些地方正是几何模型信息最易丢失所在,出现空洞,悬面的概率大,选取合适的精度对 STL 模型的修复重要的影响;

(2) 网格化选取的精度指标是:控制角度和网格化长度(弦高),但无法实

现自适应网格。

(3) 精度参数的选取。网格化参数定为：弦高 0.09；控制角 0.50。

4.2 STL 文件格式及分类

STL 文件基本单位是小三角形面片。每一面片用三个顶点描述形状，用一个法向量描述面片的方向。正确的面片应该满足右手坐标法则。这成为检测 STL 模型正确与否的一个重要准则。

STL 文件的数据模型分两种格式：BINARY 格式和 ASCII 格式。一般，采用 BINARY 格式表达的 STL 文件大小为用 ASCII 格式文件的六分之一。

4.2.1 BINARY 文件格式

BINARY 文件格式在文件大小上占有优势，除去文件初始说明，属性说明和三角面片总数等 86 个字节后，全部是描述模型的几何信息。BINARY 以浮点数的形式，以面片为基本单位连续存储记录下三角面片的顶点信息和法向信息。

(1) 前 80 个字节存储零件名和其他信息

(2) 接着 4 个字节存储三角面片的个数

(3) 接下来为每 50 个字节为一个三角面片的信息，这 50 个字节的意义分别为：面片法矢量：12 个字节(3 个坐标 X4 字节)；三个顶点坐标：36 个字节(3 个顶点 X3 个坐标 X4 字节)；属性项：2 个字节。

4.2.2 ASCII 文件格式

ASCII 文件格式虽然格式简单，读取方便，每条三角形说明信息中人为地增加了额外说明信息，如 Facet, Solid, Outer Loop 等对实体，面片，法向量的说明信息。因而文件加长，增加了文件的存储开销。

4.3 快速成型中 STL 数据模型的正确性检测

正确性检测主要是指是用 CAD 系统中几何造型理论进行诊断，满足欧式空间条件的约束。在几何造型中，实体模型要求是有效的和无二义性的，即正则物体。因此对 STL 数据模型的正确性检测分为正则性检测，即满足正则物体的欧拉公式，也要满足基于微分几何理论的几何规则检测。

4.3.1 正则性检测

正则性检测,是针对零件整体来说的,又称为整体性检测。作为一个正则性物体的零件,表面都是三维空间的有界闭集。STL 格式文件是通过 CAD 实体或者表面模型进行三角化离散而得到的,相当于用由空间三角形构成的多面体来逼近 CAD 模型。如果将 STL 文件表示的零件模型看作一个多面体,则根据欧拉公式可求出 STL 文件中点数 V 、边数 E 和三角形个数 F 的关系^[11]。

$$\text{即公式为 } X=F-E+V=2(B-G)$$

其中 X 为欧拉示性数, B 为形体个数,一般 STL 文件个数为单个零件,故 $B=1$, G 表示零件表面的亏格(genus),即穿透形体的空洞数。

特别的,现阶段研究的 STL 数据模型无空洞,相应的欧拉公式为: $F-E+V=2$

另外,在 STL 文件中,每个三角形记录三条边,这样每条边均被该三角形和与其共享该边的邻接三角形记录,即每条边被重复记录两次。因此,可得实际边数 E 与三角形 F 的关系为: $E=1.5F$ 。

综合以上两点,进一步得出顶点与三角形个数之间的关系为

$$V=E-F+2=0.5F+2$$

由此得到,在 STL 文件中顶点数约为三角形个数的一半,边数为三角形个数的 1.5 倍。符合这个条件的才可能为正确的 STL 文件,否则肯定是错误的。

同时也可看出,在 STL 文件中存在大量的冗余信息,如果能为其引入拓扑信息,每个顶点只记录一次,将有效的改善 STL 文件的读取和处理效率。故对 STL 数据模型诊断和修复之前,必须对其进行顶点冗余量的快速去除和毗邻关系的快速建立算法。

4.3.2 几何规则检测

几何规则检测是针对离散化 STL 模型的空间三角形而言的。微分几何理论,明确约束了三角形面片的顶点,边和面的几何关系^[11]。因此它也成了 STL 模型正确性检测的重要依据。

STL 文件是一个包含许多三角形小平面数据的文件,同时从文件格式分析得知,其中每个三角形平面都用一个法向量和三个顶点坐标来描述,它所描述的正确实体数据模型应该满足如下三条法则^[12]:

法则 1: 右手法则 三角形平面的法向量方向和它的三个顶点的排列顺序,

符合右手法则。也就是说，每个三角形平面的法矢量方向和其三个顶点的次序相对应，用以区分所表示实体的内部和外部，法矢量是一个单位向量，指向实体的外部。图 4-8 说明了法矢量方向与三角形顶点的关系的错误情况。

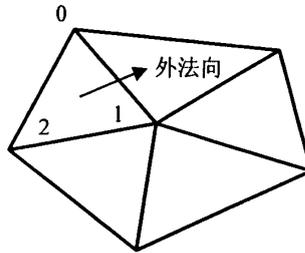


图 4-8 法矢量方向与三角形顶点的关系

法则 2：顶点法则 每相邻的两个三角形平面只能共享两个顶点。

一个三角形平面与其相邻的三角形平面有且只有一条公共边，一个三角形平面的顶点不能落在另一个三角形平面的边上，图 4-9 为不符合顶点法则的情况。

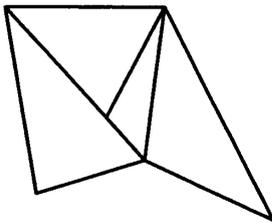


图 4-9 STL 模型违反顶点法则

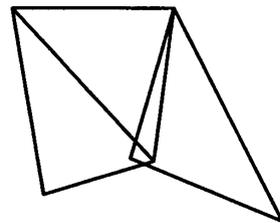


图 4-10 孔洞或者裂缝

法则 3：边法则 三角形的每一条边必须且只能由两个三角形所共有。由于任何实体的表面均是封闭的，因此，当用许多三角形平面来逼近和描述真实实体时，也应该满足实体表面封闭的这一性质，其任意三角形平面的边要由两个不同的三角形平面所共有，否则，实体表面将出现裂缝、空洞、重叠表面或悬面。

4.4 STL 文件错误产生的原因分析

4.4.1 快速原型对输入文件的要求

精度是评价成型制造的重要要素之一。为提高制作精度，人们对成型材料及工艺做了不少的研究，在几何模型向数据模型转化方面也取得了不少进展。但迄今为止，还没有一个对成型技术适用的公认的标准零件格式。现在普遍采用的是

被认为“准”(defacto)标准的由美国 3D SYSTEM 公司开发的 STL 文件格式, 用 STL 格式的数据模型作为 CAD 几何模型与成型机所需数据转换的接口。STL 数据模型用空间小三角平面片近似表面。显然小三角片越多转换精度越高, 但 STL 文件尺寸也越大, 转换时间也越长。

对于成型制造来说, 输入数据的质量和精度与这种技术的实现是同等重要的。因而, 为提高原型制作精度, 使成型制造获得高质量和高精度的数据成为当前 RPM 领域中正在研究的课题之一。RPM 所需的输入数据应该满足以下要求:

- (1) 轮廓数据应该正确, 不发生几何信息丢失、畸变、蜕化等非正常现象;
- (2) 轮廓数据应尽可能精确。需要从计算精度和算法入手加以改进;
- (3) RPM 需要简单的、非相交的、封闭的、连续的轮廓截面形状;
- (4) 尽量减少数据冗余, 节约资源;
- (5) 有利于减少命令文件的尺寸, 节省原型制作时间。

若上述要求不能满足, 很可能出现制作时间长, 精度降低, 甚至制造出一件或一批废品的现象。为达到提高精度和降低制作时间的目的, 人们采用不同的方法进行研究和探讨。

4.4.2 STL 文件本身存在的缺点

把实体模型用小三角平面片逼近内外表面来形成 STL 数据模型。由于三角化算法上的原因, STL 数据模型存在以下不足:

- (1)用小三角平面片近似实体表面明显降低了精度;
- (2)STL 模型中各三角面片间没有拓扑信息, 三角面片间的公用节点需多次存储;
- (3)STL 数据模型三角平面片之间的邻接关系, 则没有表示;
- (4)为提高精度当用多个三角平面片近似时, 很可能出现一个顶点分离。

4.4.3 由于 CAD 造型造成的 STL 错误

当将实体模型转换为 STL 格式时, 一般过程是首先将 CAD 模型的表面数据提取出来, 之后对其进行参数化, 然后对参数曲面进行三角化逼近, 从得到所要求的 STL 文件格式。由此产生的错误主要有以下三种:

- (1) 错误经常出现在引入特征与作图特征的结合处。

在引入特征与作图特征的拟合中,由于它们贴合平面之间的间距不一,很容易造成间距小的地方拟合,而大的地方没有,因而在局部地方产生小缝,结果导致 STL 文件产生三角面片之间产生小缝。

(2) 问题一般出现在不同造型系统之间的转换中。

在 CAD 造型中,文件常常需要在多个 CAD 软件之间进行转换,在转换的过程中,由于各种 CAD 软件中的实体曲面域并不相同,因而在转换中,常常会发生信息丢失的情况,因而也会造成多种错误。

(3) 如果 CAD 造型时如果布尔运算有错误,或者仅仅简单的作了两个实体,但没有进行布尔运算,这个模型转化为 STL 格式后,虽然根据 STL 文件的正确性法则来看,并没有错误,但是这个模型的确存在错误,如果这种错误不加以修正的话,作出的快速原型将是分离的两块,或者严重强度不够。

4.5 STL 文件三角网格拓扑关系的快速建立算法

STL 通过对三角网格模型各个三角形数据点及法矢量的记录保存成型件网格模型几何信息,数据中不包含网格的连接关系等拓扑信息。但是由于 STL 数据本身的结构特点,STL 数据表示的网格模型中存在大量的重复冗余数据。这些重复冗余数据的存在不但减小了系统的数据处理效率,对快速成型系统的内存容量提出挑战,而且直接影响到基于这种模型数据的三角网格拓扑关系重建。

4.5.1 STL 文件冗余顶点的快速剔除算法

几何模型的正则性条件为:正确的 STL 模型,顶点数目为面片数的一半,边的数目为面片数 1.5 倍。但从待读入的 STL 文件格式看出,每个面片描述中,包含 3 个顶点。同样,由正确的 STL 模型几何规则之顶点规则,空间三角形每相邻的两个三角形平面只能共享两个顶点,也就是边至多被两个面片共有。实际由 CAD 系统生成的存在缺陷的面片化的模型边的重复存储的次数远不只两遍。如果同一个数据点属于 N 个三角形,那么这个数据点在 STL 数据中被重复记录 N 次。

由于 STL 数据中仅包含模型的几何信息,没有提供可供借鉴的拓扑关系数据,因此如何仅仅依靠数据点的几何位置信息将重复数据快速去除是一个影响 STL 数据应用的重要因素。在以往的研究当中,根据其 STL 模型显示视图的包

围盒，将模型平均分块细分，逐块进行数据点的读取与处理。但在块与块的交界部分的数据点需要重复读取和处理。最直观的去重方法是每读入一个数据点都与已经读入的数据点进行比较，然后判断该点是否已经存在。但采用不同的数据组织方式加以存储，算法空间和时间效率上会有很大的差别。基于查找海量数据的需求，作者采用动态查找表之一的排序二叉树，对 N 个含有重复顶点进行归并，其查找效率可以达到 $O(\lg n)$ 。

4.5.1.1 顶点数据结构

```
typedef struct PointNode
{
    Point Pt;                /*定义结点*/
    struct PointNode *Branch; /*分支二叉树的入口指针*/
    struct PointNode *Left;   /*左子树指针*/
    struct PointNode *Right;  /*右子树指针*/
    FaceNode *triangleList;   /*包含此点的三角形面片指针*/
}Pointnode, *ptrPointnode;
```

4.5.1.2 顶点排序二叉树基本过程如下所述

a. 首先建立一个点结构 `PointNode`，这个结构中包含点数据变量、左子树指针，右子树指针，三角形面片指针和一个分支结点指针。点数据变量用于存储点数据；左右指针分别指向左右子树；分支结点指针用来指向点数据 X 坐标相等时按 Y 排序的二叉树根结点，在 Y 二叉树中指向 Y 相等时按 Z 排序的二叉树根结点；三角面片指针以用来指向拥有该结点的三角片链表的头指针。

b. 在建立排序二叉树时，为了达到二叉树根结点左右子树大体平衡的目的，首先找出读入的 STL 中所有点中 x 坐标值居中的那个点作为二叉树的根结点。

c. 在找出根结点后，依次取出 STL 文件中的所有点数据，把它们的 x 值分别与从根结点开始的上层父结点进行比较，如果 X 坐标值大于根节点则把它存放在右子树中，反之把它存放在左子树中。如果 X 值相等，就把它插入 `branch` 指针指定的按 Y 大小排序的分支二叉树中。如果 Y 也相等，则继续把它插入由 Y 分支二叉树中结点的 `branch` 指针指定的按 Z 排序的二级分支二叉树中。

如果 Z 也相等, 则不忽略这一点。

d. 如果 STL 文件中的所有数据都已经提取完毕, 顶点排序二叉树建立完毕。

其中寻找平衡二叉树的根节点是这个过程中的一个难点, 本论文中采用了如下的算法:

a. 首先从 head 链表中读入一个数据, head 为 STL 文件数据的入口指针。

b. 定义两个计数器 count1 与 count2;

c. 读入 head 中的下一个数据, 两个数据的 x 值相比, 如果比第一个数据大, 那么 count1 加 1, 如果比第一个数据小, 则 count2 加 1, 若相等舍弃, 再读入下一个数据, 直至所有的点数据都访问完成为止。

d. 再求得 count1 与 count2 的差 sub, 如果 sub 小于等于 1 就说明这个点数据就是要找的二叉树的根节点。再转到 e, 搜索结束; 否则转到 a, 继续搜索。

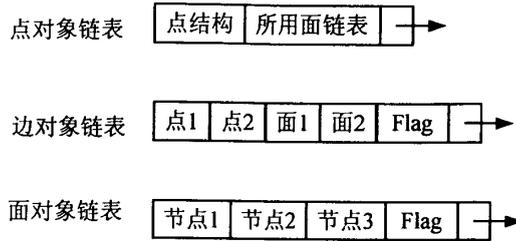
e. 结束。

4.5.2 STL 模型拓扑关系的快速建立

STL 模型的拓扑关系是指三角形三点, 三边和三角形面片的数据相关性。STL 文件是三角形面片无序集合, 只包含了几何信息, 没有反映三角形之间地拓扑关系, 单纯从这种无序地三角形面片中寻找空洞, 裂缝, 重叠, 交叉等非常地困难。由 STL 数据模型正确性检测规则出发, 需要实现以下拓扑信息地查询:

- (1) 对任一三角形面片, 可以查出其相邻地面片;
- (2) 对任一条边, 可以找出包含它的相邻面片;
- (3) 对任一顶点, 可以列出所有拥有这个顶点的三角形面片。

上节算法建立了无重复的点表。由获取拓扑信息的需要, 设计了点, 边和面三个对象链表结构。如图 4-11, 每个面表结点中同时包含指向其三条边的指针, 通过这三个指针能够索引到边表中三条边的数据; 边表结点中除了包括指向构成这条边的两个点数据外还要包括指向共用这条边的两个三角形面片的指针以及共享该边的面片总数标记 flag。显然, 采用这样的结构后, STL 模型的存储冗余大大减小。



如图 4-11 点，边和面对象链表

(1) 基本拓扑信息的分类

STL 文件中显式包含点，面二种基本的拓扑元素，隐式包含边（三角形面片的任意两个顶点构成一条边）。如图 4-12，对于 STL 文件的点，边，面，其拓扑信息归纳起来有九种：1) 由面获得包含点；2) 由点获得引用面；3) 由面获得邻接面；4) 由边获得其端点；5) 由点获得引用边；6) 由面获得其包围边；7) 由边获得其引用面；8) 由点获得共面或边的其他点；9) 由边获得共面或共点的替他边。前两种拓扑信息是基础，通过前面的拓扑信息可以间接获取其他信息。

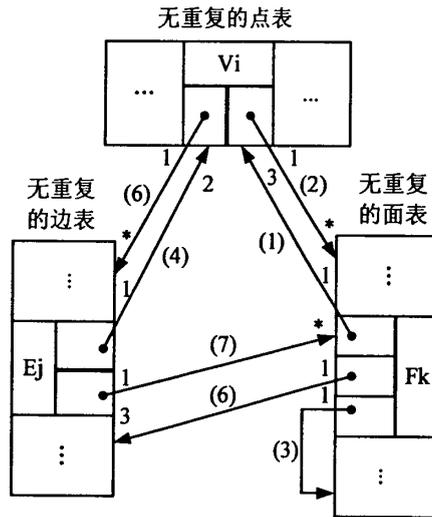


图 4-12 点，边和面拓扑关系

(2) STL 模型拓扑信息的快速建立

第一：建立面片与顶点的邻接关系：原则是每一个面片均是其顶点的邻接面，算法逐一取出零件的每一面片，获得面片三个顶点，并将面片分别加入到三个顶点的邻接面片链表中。

第二：建立顶点间的邻接关系：原则是每一个顶点的所有邻接面的顶点均是其邻接顶点，获得该顶点的所有邻接面（由（1）生成），判断这些邻接面的三个

顶点是否已经在该顶点的邻接顶点表中，将其插入。

第三：建立面片间的邻接关系

(1) 从面片中逐一取出无处理标志的每一个面片 F1 的每一条边的两个顶点 (Pa 与 Pb);

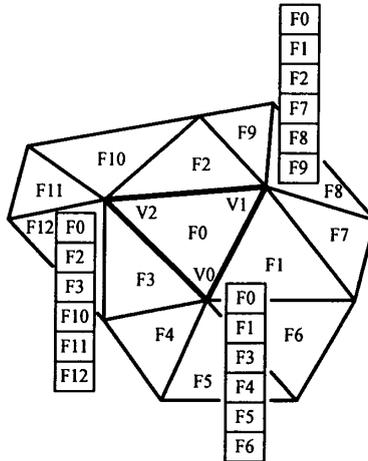


图 4-13 模型拓扑信息快速建立

(2) 分别取出 Pa 点的所有非 F1 且无处理标志的邻接面，把这些邻接面的三个顶点分别与面片 F1 的另外两个顶点(Pb 或 Pc)比较，相同则该面 Pa 片与 F1 相邻，由此可建立由面获得其邻接面的拓扑信息，当然它们的公共边为 PaPb 或 Pc，进而建立由边获得其端点，由点获得邻接边，由面获得其包含边，由边获得其邻接面等拓扑信息。

(3) 接着取出该边的另外一个顶点 Pb，同样的，分别取出 Pb 点的所有非 F1 且未处理过的邻接面片，把这些邻接面片的三个顶点分别与 Pc 点比较，得到公共边为 PbPc 的邻接面。

(4) 以上完成三个点的邻接面的提取，进而得到了其任意两点为公共边的邻接面.完成以上比较后对 F0 作处理标志.在到面表中取下一个无处理标志的面片进行以上的处理，直到遍历整个完整的面表。其中，对面表的处理标志设置的目的的是避免重复拓扑信息的处理。

算法描述如下：

```
for (每个无处理标志的面片 F)
{
    for(short n_Vertex=0;n_Vertex<2;n_Vertex++)
```

```

{
    取出面片 F 中的 n_Vertex 顶点的邻接面;
    for(short m_CompVert=n_Vertex+1;m_CompVert<3; m_CompVert++)
    {
        for(n_Vertex 顶点的非 F 且无操作标志的邻接面 Fi)
        { if(邻接面 Fi 的顶点与 F 面的 n_CompVert 顶点相同)
            break;//

            将 Fi 作为 F1 的邻接面, 将其插入到相应的邻接面表
        }
        if(需要建立相关的拓扑信息)
        {
            if(面 F 没有三条边) 则=
                STL 文件存在缺陷, 补充产生面片没有的边, 建立需要的拓扑信息..
            }
            F 面作处理标志;
        }
    }

```

第四, 根据 STL 模型缺陷的诊断和修复需要, 建立边的邻接面表。具体地, 检查 F 是否已经有顶点为 $m_CompVert$ 和 n_Vertex 决定的边 E, 如果存在, 产生这两点决定的边 E, 加入边表, 并建立面 F 和面 F_i 包含边 E 的拓扑信息。

4.6 基于边信息的 STL 模型缺陷检测修复算法

4.6.1 STL 模型的错误诊断

通过上述点表, 边表和面表 3 个数据文件能使 STL 文件的正确性校验变得十分容易。除了将欧拉公式应用于正则性检验之外, 从建立的边表结构的一个数据项 $flag$ 的取值来完成 STL 模型的几何检验。只要检查一下边表中的标记字, 看其是否为 2 就可以了。

如果标记字 $flag=1$, 表示该边仅为一个三角形所有, 肯定存在间隙、孤立面等错误。标记字 $flag=2$, 该边为两个三角形所有; 标记字 $flag$ 为-2, 拥有该边的

两个三角形在该边的定向相同, 出现零件内外表示矛盾; 标记字 $\text{flag} \geq 3$, 拥有该边的三角片数不小于 3 个, 此时出现重叠面、相交面等错误。

此外, 面表中每个节点存放三个指向顶点的指针和 3 个指向边表的指针, 其次序与面片的法向符合右手定则。具体缺陷检测方法如下:

(1) 法线错误: 对边表和面表进行检验, 凡是两边的差乘之积的方向与法线方向不一致的, 都认为是法线反向错误。

(2) 小缝错误: 对点对象表进行检查, 设定一个很小的参照量, 但凡点距在参照量之内的两点就构成了小缝。此时包含由于点分离引发的小缝和由于三角形一边太短引发的小缝错误,

(3) 孔洞, 重叠面等错误都与棱边被共享的三角形面片数量有关。因此, 通过检测边表结构的标志位来检测错误, 并通过调整边结构和相应的点结构和面结构来处理错误。

4.6.2 STL 模型的错误修复算法

一. 小缝错误

(1) 判断是否 $\text{DISTANCE}(A, B) \leq e$, A 为点表中的任意一点, B 为点表中除了 A 之外的任意一点。 e 为极小容差。如果是, 则转(2); 否则转(1)。

(2) 找出退化点 B 后, 找出边表和面表中所有点 B 的索引序号, 并用点 A 的索引序号来代替, 转(1)。

二. 法线反向错误

这个问题基本不会影响 STL 文件的切层性能, 因此不加修复也可。需重新计算构成三角形的两边的差乘, 用乘积来代替原来的法线。

三. 空洞错误修复

空洞是 STL 文件最常见的错误。在小缝修复中, 设置适当的 e 非常的困难, 故修复后可能会产生空洞。故在修复空洞前, 需先小缝修复, 合并顶点。

首先对边表进行遍历, 找出其中边的标志为 1 的所有边, 将这些边放入一个新建的数组 $V[n]$, 由于组成一个多边形空洞的相邻两边之间总有一点是相同的, 根据相邻边有共同点这个原理可将这些边归并成几个多边形, 并将每一个多边形中的顶点按照多边形的顶点顺序放入一个新建的数组 $V_i[M]$, i 为多边形的个数。

假设空间多边形的顶点数为 N , 顶点信息存储在数组 $V[N]$ 中, 并使顶点在

数组中的排列次序为：当一个人在物体表面沿该多边形的边行走时，裂缝、空洞永远位于人的左侧。

(1) 如果 $N=3$ 则此多边形是三角形，将其加入到面表，三角剖分结束，否则转至(2)；

(2) 令 I 从 0 到 N ，设顶点 $V[I]$ ， $V[I+1]$ 和 $V[I+2]$ 构成的三角形的法矢量为 n_i ；

顶点 $V[I]$ 和 $V[I+1]$ ， $V[I+1]$ 和 $V[I+2]$ 所在三角面片的法矢量分别为 n_1 ， n_2 ，它们的和矢量为 n_a ；如果 $n_i \cdot n_a > 0$ ，将顶点 $V[I]$ ， $V[I+1]$ 和 $V[I+2]$ 构成的三角形作为一个候选三角剖分，并计算 $V[I]$ ， $V[I+2]$ 间的距离 d_i ；

(3) 找出(2)中候选剖分三角形中距离 d_i 最小的三角形作为空间多边形的一个剖分三角形 T ，将其加入到面表之中。

(4) 从点 $V[N]$ 中删除三角形 T 中的顶点 $v[I+1]$ 。令 $N=N-1$ ，重复(1)到(4)的过程。

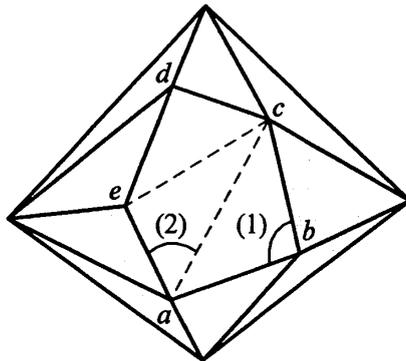


图 4-14 空洞修复草图

参考图 4-14，空洞 abcde，其修复包含以下步骤：

(1) 初始化错误顶点数组 $V[N]$ 。按 a ， b ， c ， d ， e 的顺序送入数组 $V[N]$ ，其中 $N=5$ 。

(2) 从点 a 开始到点 e ，每次取数组三个连续点。总共有 abc ， bcd ， cde ， dea 四个三角形。对三角形 abc ，分别求出三角形面法向量 n_t ，向量 ba 与向量 bc 的合向量 n_a ，进而计算 $n_t \cdot n_a$ 的点乘。若大于零，则将 abc 确定为候选三角形。依次处理三角形 bcd ， cde ， dea 。

(3) 在候选三角形中选择多边形内角对应边最小的三角形 abc 作为剖分三

角形 T, 加入面片;

(4) 从顶点数组中删除点 b , $N=4$;

重复以上四步操作, 直到错误顶点数组为空。

四. 其他错误

上述修复功能实现了主要的文件缺陷的修复, 对于其他错误, 可以采用一种错误转移的方法, 简单有效:

(1) 先将错误相关元素删除, 集中构成空洞, 降低错误复杂度, 增加错误数量。

(2) 对空洞错误进行修复。

4.7 本章小结

导入到 RP 系统的三维图形由一系列的空间三角形面片近似描述。由此, 给成型零件表面带来了缺陷。本章从研究缺陷产生源头入手, 进行了以下工作:

(1) 为了减少近似描述的数据丢失, 首先针对 CAD 系统中 STL 文件格式转换参数及其选定, 提出了最多面片近似三维图形的转换参数取值。

(2) 具体分析了三维图形缺陷产生的原因, 将缺陷分为图形自身造成缺陷和 CAD 造型产生的缺陷。

(3) 运用 CAD 几何造型理论, 分析了数据文件正确性检测标准, 并应用到后续的数据文件缺陷检测方法当中。

(4) 为了缺陷修复的快速进行, 首先对描述三维图形几何元素的拓扑关系建立算法进行研究, 建立了图形点对象, 边对象和面对象表。提高了图形加载速度, 对后续图形缺陷诊断和修复算法设计提供了有效的处理依据。

其中, 采用排序二叉树对 STL 文件进行存储, 将重复顶点去除, 建立了无冗余顶点的点对象表。

(5) 最后, 分析提出了 STL 文件缺陷诊断和修复方法。对较常见的空洞缺陷进行了分析, 提出了相应的算法。

第五章 系统设计与实现

5.1 快速成型 STL 数据诊断与修复系统的设计思路

5.1.1 系统的总体结构模块

该系统主要包括 STL 数据文件加载（模型点，边，面结构建立）、三维图形显示控制、模型缺陷诊断、模型缺陷修复五大模块，各模块包括多个子模块。系统结构图如图 5-1 所示。

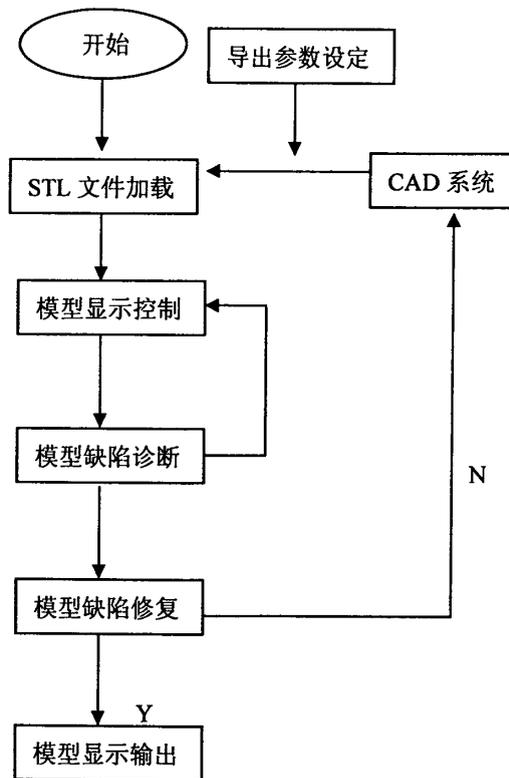


图 5-1 系统工作流程图

数学描述信息建立，缺陷诊断和缺陷修复等模块。即：程序运行时，必须先加载 CAD 系统导出的 STL 格式数据文件的同时，建立成型零件模型的数学描述信息（点，边和面结构以及相互的拓扑信息），然后才能运行数据诊断处理模块，此时产生存在缺陷的点，边和面等几何元素信息。进而运行数据修复处理模块，生成数据后置处理所需的正确的 STL 模型。整个数据处理过程都可以借助三维图形显示和控制模块进行验证。

在系统总体结构确定的情况下，接下来要考虑的是各模块具体的功能。本系统是 STL 模型缺陷诊断和修复系统，它的最终目的是满足零件熔焊成型的要求，输出用于模型分层后处理 STL 数据文件。围绕这个目的，并根据各模块功能的要求，确定了本软件系统的程序框架结构，如图 5-2 所示。

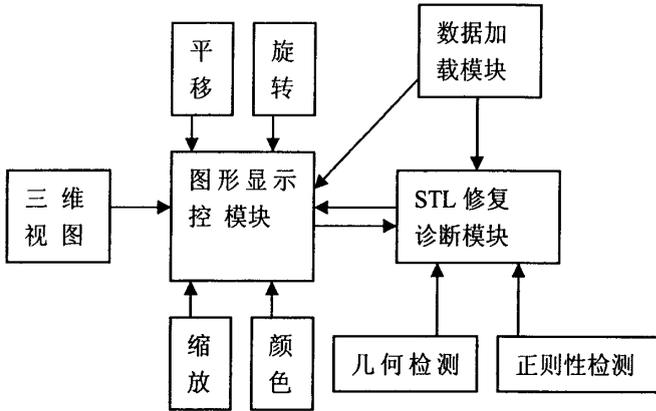


图 5-2 程序框架结构图

5.1.2 系统各功能模块设计

(1) 图形数据加载模块

在图形数据加载的过程中，建立了点表，边表和面表。用户在输入要成型零件的 STL 文件数据后，我们需要设计一些数据结构来完成对这些数据的处理、管理、让这些数据存储入数据结构中，作为零件的视图显示和图形缺陷诊断修复控制的数据来源和基础。其中，有几点比较关键的地方，一是如何提取这些数据；二是如何将这数据存储起来，以后这样方便地取出来使用。下面我们作出了分析和解答。

关于数据格式的提取，需要对 STL 格式的图形文件进行研究，特别是存储空间小，读取速度快得二进制格式文件。

存储数据，选用合适的数据结构是关键。针对 STL 数据量大，处理复杂的特点，采用链式结构对数据进行处理。我们可以通过分别设定几个数据结构，通过不同的数据结构变量来保存数据，来达到各种用途的数据供给。实现的关键数据结构如下：

```
typedef struct Normal
```

```
{
    double dx;
    double dy;
    double dz;
}normal;
typedef struct point
{
    double x;
    double y;
    double z;
}vex[3];
typedef struct STLModel
{
    struct Normal normal;
    struct point vex[3];
};
//顶点链表
struct pointchain{
    point *pt;
    struct pointchain *link;
};
//面表
struct scanschain{
    pointchain *pshead[3];

    struct scanschain *link;
};
//边表
struct scanschain{
    pointchain *pshead[2];
    scanschain *psshead[2];
    int flag;
    struct scanschain *link;
};
```

(2) 图形显示控制模块

三维图形显示模块会利用加载后的数据，自动生成三维图形元素的几何信息，从而得以描述成型零件几何形状，再利用 OpenGL 实现高品质、交互式的三维几何实体模型图形显示。

为了便于用户观察，该模块还提供了对所要成型的零件的放大/缩小、移动、旋转、三维视图显示，使用户便于从各个角度对成型对象和成型轨迹进行观察。

用户可以通过选择“显示”菜单项中的子菜单，对所要成型的对象进行放大/缩小、移动、旋转、三维视图显示等显示操作，这些操作也可以通过软件界面中的工具栏上的图形操作按钮来实现；此外还可以通过单击鼠标右键，从弹出的子菜单中来选择这些操作。可见，该模块所提供的图形操作功能灵活且形式多样。

该模块的难点在于实现对任意形状和几何数据的三维实体模型进行显示、旋转、缩放、移动等图形显示功能的算法；该算法必须解决以下两个问题：

- (1) 如何让零件的成型图形在屏幕的中央进行显示。
- (2) 如何实现对成型件进行平移、旋转、缩放等功能变换。

该模块的结构框架如图 5-3 所示。

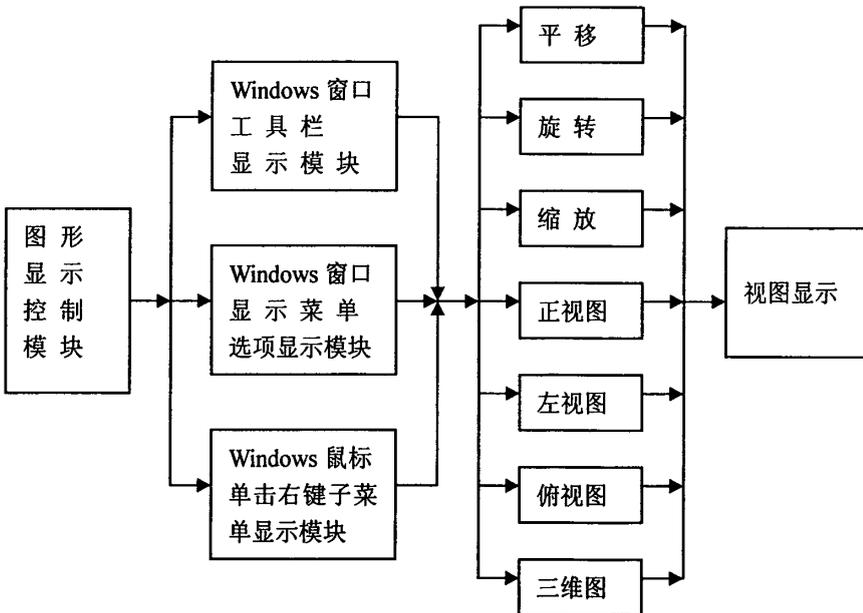


图 5-3 图形显示模块控制结构图

(3) 图形缺陷诊断修复控制模块

三维图形诊断修复控制模块分为正则性诊断修复模块和几何诊断修复模块。在进行正则性修复的基础上，主要对顶点分离和模型非完整性错误进行修复处理。修复流程如图 5-4:

1. 由于 CAD 系统导出精度的原因，同一顶点存在分离现象，该合并顶点的关键是选取合理的近距离判断值。
2. 空洞错误的重新网格化是修复方法的关键，影响了修复的时效性。

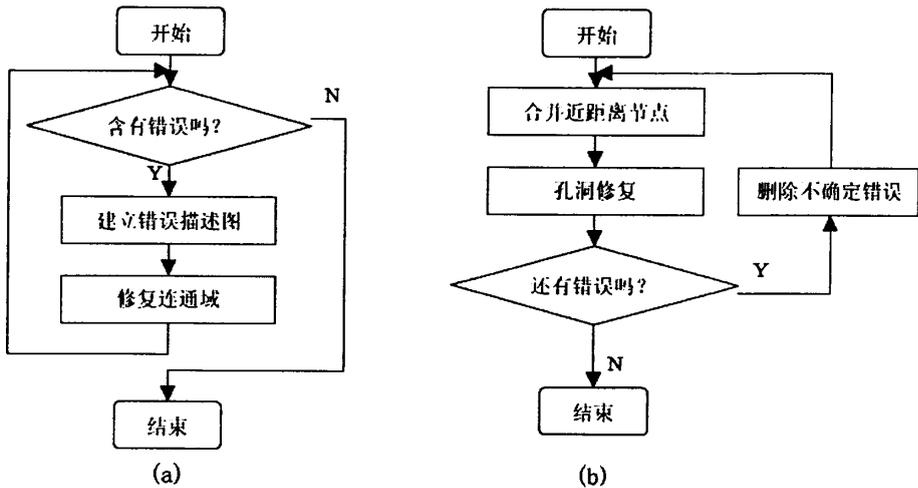


图 5-4 错误诊断修复流程

5.2 快速成型 STL 数据诊断与修复软件实现

5.2.1 系统界面设计

本系统界面是利用 MFC 和 Microsoft 向导编写 C++的图形用户界面开发而成。利用此方法我们可以非常迅速、有效地按照自己需求制作出美观大方的用户界面。其制作过程大致如下:

启动 Visual C++6.0，进入 Visual C++6.0 集成开发环境。

选择 Projects 标签，在项目类型清单中，选择“MFC AppWizard(exe)”项目，在 Project Name 中输入项目名 Remend，在 Location 中指定项目文件所在的文件夹路径。然后单击 OK 按钮。在接下来弹出的对话框中可以选择即将生成框架的各种风格样式。一系列对话框结束之后，Microsoft 向导生成一个空的应用程序框架，程序设计者可在这个框架下添加菜单、工具条、对话框、分割视图、

滚动视窗等，按照自己的设计意图搭建起漂亮的图形用户界面。

具体来讲，充分利用了 C++ 环境下经典的文档视图结构，设计了一个文档类 RemendDoc 来保存各类的公用数据，以简化数据的传递；设计了三维视图类 RemendView 和 ScopeView 类用于对图形修复前后的三维显示。这里，为便于修复处理的效果对比显示，采用分割窗体技术，实现了多视图显示。

从图 5-5 可以看出，用户视图窗口由两个窗口区域组成。左窗口是修复后成型件显示窗口，它由 CScopeView 来实现。右窗口是修复前成型件的图形显示窗口，它由类 CRemendView 来实现。

该用户界面的设计步骤如下：

(1) 在类 CMainFrame 中，增加成员变量 m_wndSplitter，并把它的类型设为 CsplitterWnd 类。

(2) 接下来在下面的函数：

CMainFrame:: OnCreateClient(LPCREATESTRUCT lpcs, CCreateContext* pContext)中做如下修改：

```
if(m_wndSplitter.CreateStatic(this, 1, 2, WS_CHILD | WS_VISIBLE))
{
    TRACE("Failed to CreateStaticSplitter\n");
    return FALSE;
}
// First splitter pane
if (m_wndSplitter.CreateView(0, 0,
    RUNTIME_CLASS(CScopeView), CSize(200, 200), pContext))
{
    TRACE("Failed to create lay view pane\n");
    return FALSE;
}
// Second splitter pane
if (m_wndSplitter.CreateView(0, 1,
    RUNTIME_CLASS(CRemendView), CSize(200, 200), pContext))
{
```

```
TRACE("Failed to create preview pane\n");
return FALSE;
}
```

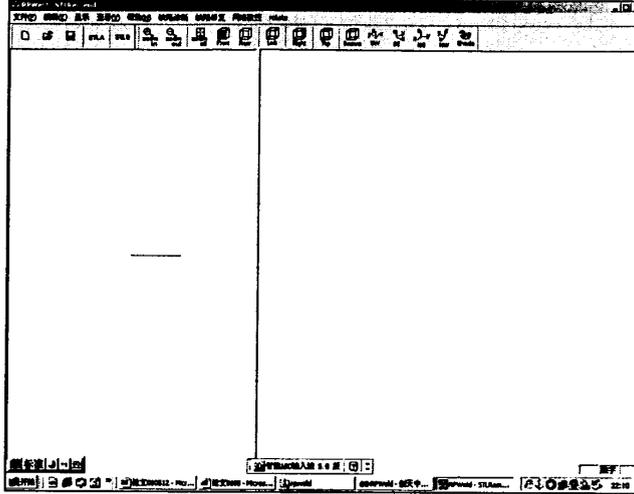


图 5-5 系统界面

5.2.2 系统验证

本软件系统在焊接快速成型软件系统中得到较好的应用。对 STL 模型中的常见错误都能检测与修复,从而为后续熔焊快速成型过程的顺利进行正确的数据模型。以下是一个带有倒角的抽壳薄壁件,用它作为 STL 格式文件缺陷诊断和修复的实例。图 5-6 是薄壁件网格化显示,三角形面片数目为 7556 个,CAD 系统中 STL 转化参数:弦高为 0.09,控制角为 0.50。

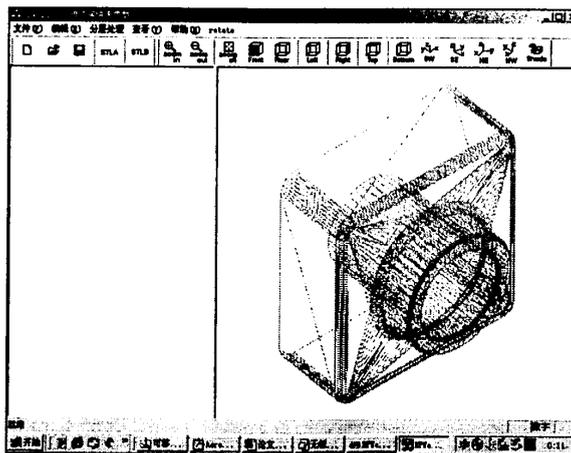
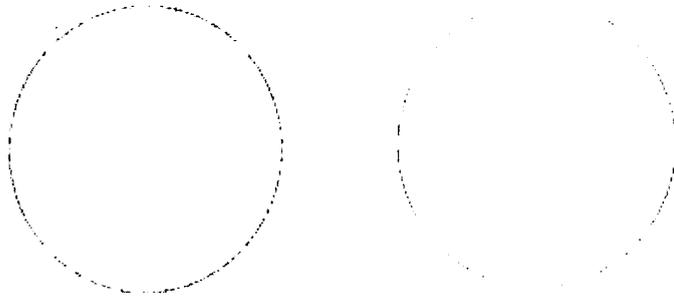


图 5-6 网格化的薄壁件

在 STL 文件数据加载过程中,建立了模型点对象,边对象和面对象。完成了几何元素拓扑化,进而对文件缺陷进行诊断与修复。修复前后的效果验证如下图 5-7 所示:



(a) 修复前零件的分层轮廓放大

(b) 修复后零件的分层轮廓放大

图 5-7 修复前后的零件等厚分层轮廓效果比较

5.3 本章小结

基于 OpenGL 真实感显示技术,根据熔焊快速成型工艺实际需要,详细介绍了 STL 数据诊断与修复系统的设计架构和实现方案。

(1) 提出了三维图形缺陷诊断和修复系统的设计方案,包括系统总体架构、设计模型、模块划分、各模块功能实现等,其中根据系统功能,大体分为后台数据处理部分和前台数据显示部分;具体划分为图形数据加载,图形显示和图形缺陷诊断和修复等三个模块。

(2) 设计并实现了易于操作的,人性化的系统界面,设计了图形显示控制器,图形检测与修复控制器。

(3) 提出了系统实例验证。

第六章 结论

通过自己的努力，完成了本课题的研究，基本实现了既定目标。本课题所完成的主要内容有：

(1) 从实用的角度出发，设计开发了友好的用户界面，使其操作方便灵活。

(2) 设计开发了基于 OpenGL 技术的图形接口程序模块，可适用 VC++ 环境下图形渲染程序。可移植性强。

(3) 在 CAD 系统里，为最精确的导出 STL 文件格式，避免由于 STL 格式转换产生的文件缺陷，详细研究导出转换参数及其选定。

(4) 在分析 STL 格式基础上，运用计算机几何造型理论，在加载图形文件的过程中，详细研究提出了图形点，边和面几何元素的拓扑信息算法，该算法对文件缺陷诊断与修复处理提供了可靠的依据。

(5) 在分析 STL 文件正确性检测规则基础上，对 STL 文件缺陷产生的原因进行分类，分析基于边拓扑信息的缺陷诊断和修复处理方法，重点针对较常见的空洞错误研究了修复方法。最终实现了熔焊快速成型 STL 格式文件缺陷诊断与修复工具软件系统。该软件生成的数据文件能够成功应用于快速成型软件的后续处理。

参考文献

- [1] 王守德,刘福田,程新.智能材料及其应用进展[J].济南大学学报(自然科学版) Vol.16. No.1
- [2] 陶宝祺,石立华,熊克.自诊断自适应智能复合材料结构系统的研究[J].应用科学学报.1999.17(1)
- [3] 陶宝祺.智能材料结构[M].北京.国防工业出版社.1997
- [4] 颜永年,洪国栋,吴良伟.快速成型技术发展状况与趋势研究.1998.11
- [5] 王勇.OpenGL 的选择机制在三维交互应用程序开发中的应用.计算机系统应用.2001.10
- [6] 张剑峰,黄因慧,赵剑峰等.快速成型系统中 STL 模型缺陷的自动修复技术.扬州大学学报(自然科学版)2002(5)
- [7] 武汉华中数控股份有限公司主页: www.huazhongCNC.com
- [8] 向世明 .OpenGL 编程与实例. 电子工业出版社 ,1999
- [9] 白建军等. OpenGL 三维图形设计与制作. 人民邮电出版社,1999
- [10] 吴斌等. OpenGL 编程实例技巧.人民邮电出版社,1999
- [11] 孙立鏊.计算机图形学.哈尔滨工业大学出版社.1999
- [12] 田宗军,李小林,黄因慧.快速成型系统中 STL 文件的缺陷与修复
- [13] 赵吉宾,刘伟军,王越超等. STL 文件的错误检测与修复算法研究.计算机应用. 2003.2
- [14] 谢存禧,李仲阳.STL 文件毗邻关系的建立遇切片算法研究.华南理工大学学报.2000.28(3)
- [15] 蔡小康.智能化的快速成型切片算法.中国机械工程.1997.8(5)
- [16] 礼宾,吴宁祥,谢里阳.基于边的快速分层软件研究.机械设计与制造. 2003(6).
- [17] 胡德洲,李占利等.基于 STL 模型几何特征分类的快速分层处理算法研究.西安交通大学学报.2000.1
- [18] 唐杰,周来水,周儒荣等.STL 文件修补算法研究.机械科学与技术.1999.5
- [19] 成学文,李德群,周华民,崔树标.基于哈希表的 STL 面片冗余顶点快速滤除算法.华中科技大学学报(自然科学版).2004.6
- [20] 孙玉文,刘伟军,刘建.CAD/RP 系统间数据模型的转换方法与发展.航空精密

制造技术.2001.4

- [21] 黄放,快速成型技术及其发展.铸造设备研究.2004.2
- [22] 徐国贤.直接金属沉积成形的软件及工艺研究.硕士学位论文.北京:清华大学机械系,2001
- [23] 李涤尘,卢秉恒,朱均,快速成型制造是实现并行工程和敏捷制造的有效途径,北京:先进制造技术学术会会议论文,1996
- [24] Paul F.Jacobs. Rapid prototyping & Manufacturing . Fundamentals of Sere Lithography, Society of Manufacturing Engineers SME-CASA, Dearbornmich
- [25] Deckard,CR and Beaman JJ. Solid freeform fabrication and sintering. Proc. 15th -Nanmrcsme. 1987 (Vol.8)
- [26] J. P. Kruth. Material Increase Manufacturing by Rapid PrototypingTechniques, A -nnals of the CIPP, 1991,(Vol.4)
- [27] Xue Yan, P Gu, A review of rapid prototyping technologies and systems, Computer -Aided Design, 1995, (Vol209)
- [28] Lamont Wood. Rapid Automated Prototyping . An Introduction, IndustrialPress Inc. New York, 1993
- [29] 陈鸿,商细云,陆宏.激光变长线快速成型系统软件. “21 世纪新产品快速开发技术”会议论文集,陕西科学技术出版社,2000
- [30] Wohlers T. Rapid prototyping & tooling state of the industry. Journal of Tsinghua University Science and Technology . 2002
- [31] 王至尧, 单忠德. 模具制造中的特种加工技术. 中国机械工程,2002,13 (22)
- [32] Choi D.S., Lee S. H., Shin B.S, et al. Development of a direct metal freeform fabrication technique using CO2 laser welding and milling technology. Journal of materialsprocessing technology, 2001(113)
- [33] Bryden G, Pashby I R, Wimpenny D I, et al. Laminated steel tooling in theaerospace industry. Materials & design. 2000(21)
- [34] Obikawa T, Yoshino M, Shinozuka J. Sheet steel lamination for rapid manufacturing Journal of Materials Processing Technology, 1999
- [35] Dickens P, Pridham M,Cobb R. 3D welding. In: Proceedings of the 1stEuropen

- Conference on rapid prototyping. Nottingham: University of Nottingham, 1992.
81~93
- [36] 易树平,哈津,林利红,等. 分层实体制造中金属分层板结合的新方法. 重庆大学学报,2002,25 (2)
- [37] Kevin P. DTM's selective laser sintering technology: new products, new benefits, new manufacturing horizons. In: Nakagawa T, Marutani Y, Imamura M, eds. Proceedings on the 8th International Conference On Rapid Prototyping. Tokyo: The University of Dayton, 2000.
- [38] Khaing M W, Fah J, Lu L. Direct metal laser sintering for rapid tooling: process and characterization of EOS parts. Journal of Materials Processing Technology. 2001(113)
- [39] Kochan D, Kai C C, Du Z. Rapid Prototyping Issues in the 21st Century, Computer in Industry, 1999(39)
- [40] 刘斌,黄树槐.快速原型制造技术中实时切片算法的研究与实现.计算机辅助设计与图形学学报.1997.9
- [41] 张嘉易,刘伟军,王天然,三维模型的适应性切片方法研究.中国机械工程.2003(5).
- [42] 沈以赴等.快速成型中材料成型性的研究进展.材料科学与工程,2001, (12)
- [43] 毕晓亮,叶庆泰,候丽雅等.快速成型软件中 OpenGL 技术.计算机应用研究. 2001.(7)
- [44] 周龙早,刘顺洪,于冬平,基于三维焊接熔敷的快速成型技术.电加工与模具. 2004
- [45] 刘长明,杨工明. Visual C++实践与提高. 中国铁道出版社, 2001
- [46] 何建辉,董方鹏等. 实用 Visual C++6.0 教程. 清华大学出版社,2000
- [47] 胡德洲,李占利等.WINDOWS95 下基于三角形面片特征的快速分层软件研究[A],98 全国快速成型与模具快速制造会议论文集[C].西安: 陕西科学技术出版社.1998.
- [48] 许超,时红霞,赵万华. 基于微束等离子弧三维焊接的直接金属成型系统设计. 机床与液压 ,2003 (4)

- [49] 赵万华,胡德洲.一种多金属直接快速成型新工艺.机电世界.2000
- [50] 马永壮,刘伟军,董遇泰,基于有向加权图递归切片算法的研究.中国机械工程.2003.(7).

致 谢

感谢母校南昌大学对我七年时间的培养和教育,使我在幽雅的环境中学到了最丰富的知识,培养了良好的道德观念。

面对这本学位论文,首先要感谢的是我的导师张华教授,这篇论文是在导师精心的指导下完成的。从师三载,导师严谨的治学态度,渊博的学识,宽以待人的品质,将永远是我学习的楷模。导师不仅在学习上为我们答疑解惑,在生活上也给予学生无微不至的关怀,谨在论文完成之际,向导师表达深深的谢意和崇高的敬意。

特别感谢胡榕华博士在我三年的硕士生活中给我的关心和学习上的指导,感谢实验室老师贾剑平博士,徐建宁博士在学习和生活上的帮助。感谢张俊强,何银水,欧阳快德,朱海滨等各位同窗好友,和他们在一起度过了美好而快乐的时光。感谢2002级在本实验室完成毕业设计的宋美焱同学,在本人完成该论文时她的工作给以了一定的帮助。感谢所有曾经给予我帮助的老师和朋友。

攻读硕士学位期间已发表的论文

- 1、万玲娜，张华，胡榕华. “熔敷堆焊快速成型中数据处理的关键技术”.机械. 2005.12.
- 2、WAN Ling-na, ZHANG Hua, HU Rong-hua. “Preprocessing of STL Model for Deposit Surfacing Welding Rapid Prototyping”.AJWT`2005. 2005.10