

语义特征造型中直接操作方法的研究

摘要

在目前的商业特征造型系统中，对特征的直接操作还没有得到广泛的应用。结果，重复的设计过程不仅耗时还导致不能实时有效的得到反馈，缺乏对设计结果的洞察，对特征的互操作也缺少直观性。导致这些结果的一部分原因是由于当前约束求解的速度慢，另一部分原因是由于缺乏互操作方法。

首先，本文分析了语义特征造型系统中直接操作的过程，提出一个允许用户在特征模型上直接选择并修改特征参数的方法，该方法能够让用户对参数值的改变交互地进行，并将模型变化的结果实时地反馈给用户。当用户对模型改变的结果感到满意的时候，便可以选择提交本次参数的改变，然后让系统检查整个模型的有效性。这种方法中关键的地方是能够对特征模型的改变所造成的影响提供实时的反馈，而为了支持对特征参数值交互的修改，每秒钟要产生多次可视的反馈，所有的约束必须以同样的速度进行求解。因此在分析直接操作模型特点的基础之上，提出了基于细胞元的局部化操作思想，并给出了一个局部化的算法以及相关的数据结构。

其次，本文描述一种新的约束求解器，用于求解特征交互过程中两个特征形状在空间上发生重叠的约束。这个约束求解器建立在对特征模型的参数空间进行取样的基础之上，并且为了找到一个有效的模型，利用蒙特卡罗技术来减少样本数据量，该方法较传统的利用约束转换的方法提高了效率。

最后，为增强直接操作过程中的直观性，本文在最后为实型的特征参数提供了强有力的各种 3D 句柄，以及为所有的特征操作提供了预先审查的透明体，并介绍了在直接操作中如何使用这些句柄。

关键词 特征造型；直接操作；局部化；交互约束；句柄

Research on Method for Direct Manipulation in Semantic Feature Modeling

Abstract

In current commercial feature modeling systems, support for direct manipulation of features is not commonly available. As a result, re-designing is time-consuming due to the inefficient feedback, the insight given is rather poor, and user interaction often lacks intuitiveness. This is partly due to the lack of speed of current constraint solvers, but also to deficient interactive facilities.

Firstly, a method is put forward by analyzing the process of direct manipulation in the semantic feature modeling system. In this way, the feature parameter value can be changed interactively, and the feedback of the model changes is real-time, allowing users to directly select feature parameter on the feature model. When the user is satisfied with the changes, the changed parameters can be accepted, and then the validity of the whole model can be checked. The key of this method is to provide real-time feedback when the model changes. In order to support changes of the interactive feature parameter values, many times of visual feedback per second must be generated and all the constraints must be solved at the same speed. According to the characteristics of direct manipulation model, the localization of the manipulation on the Cell-based semantics models is proposed, and the related algorithm and data structures are given.

Secondly, this dissertation proposes a new solver for constraints on interactions, which is used to solve spatial overlap between feature shapes. This solver is based on sampling of the parameter space of the feature model. A Monte Carlo technique is applied to reduce the expected number of samples that are needed to find a valid model. The method in this paper is more available and effective than the traditional method which transforms a new constraint into another one.

Finally, a powerful variety of 3D handles for real-valued feature parameters are provided, with a preview overlay facility for all modeling operations, and also how to use these handles in the process of direct manipulation is introduced.

Keywords feature modeling, direct manipulation, localization, interaction constraint, handle

哈尔滨理工大学硕士学位论文原创性声明

本人郑重声明：此处所提交的硕士学位论文《语义特征造型中直接操作方法的研究》，是本人在导师指导下，在哈尔滨理工大学攻读硕士学位期间独立进行研究工作所取得的成果。据本人所知，论文中除已注明部分外不包含他人已发表或撰写过的研究成果。对本文研究工作做出贡献的个人和集体，均已在文中以明确方式注明。本声明的法律结果将完全由本人承担。

作者签名：李素慧

日期：2009年3月20日

哈尔滨理工大学硕士学位论文使用授权书

《语义特征造型中直接操作方法的研究》系本人在哈尔滨理工大学攻读硕士学位期间在导师指导下完成的硕士学位论文。本论文的研究成果归哈尔滨理工大学所有，本论文的研究内容不得以其它单位的名义发表。本人完全了解哈尔滨理工大学关于保存、使用学位论文的规定，同意学校保留并向有关部门提交论文和电子版本，允许论文被查阅和借阅。本人授权哈尔滨理工大学可以采用影印、缩印或其他复制手段保存论文，可以公布论文的全部或部分内容。

本学位论文属于

保密 ，在 年解密后适用授权书。

不保密 。

(请在以上相应方框内打√)

作者签名：李素慧

日期：2009年3月20日

导师签名：孙立强

日期：2009年3月20日

第1章 绪论

1.1 课题背景

1.1.1 课题研究目的和意义

计算机辅助工业设计(CAID)融计算机辅助设计(CAD)与工业设计(ID)于一体。美国、日本、德、法等国的典型机电产品的造型设计中相继摒弃传统的工业设计方法,而广泛的采用了综合高效和具有创造性的 CAID 技术,我们要努力缩小典型机电产品内在功能质量与国际先进水平差距的同时努力提高产品外观造型质量,将其内在的功能与外观造型设计统一起来,提高产品的总体综合质量和产品附加值,缩短设计周期,加速新产品开发研制,增强创新意识。产品设计是不断修正和优化的设计过程,重复产品的概念设计,结构设计和基于约束的变量化设计三个阶段,在特征造型系统中表现为如何提高产品后期设计修改的灵活性,其中产品模型的可编辑性和易编辑性已成为提高造型能力和造型效率的关键。

传统的基于历程的特征造型系统在产品模型的可编辑性和易编辑性的技术方面存在很多缺陷,这些缺陷涉及到特征造型的三项核心技术:

1. 变量、约束的表示及提高约束变量求解的精确性和效率。
2. 在重构过程中,需保持拓扑元素(点、边、面)的标号一致性,以维护约束集的一致性。
3. 提高模型重构的速度。

在目前的商业特征造型系统中,对特征的直接操作还没有得到广泛的应用。结果,重复的设计过程不仅耗时还导致不能实时有效的得到反馈,缺乏对设计结果的洞察,对特征的互操作也缺少直观性。导致这些结果的一部分原因是由于当前约束求解的速度慢,另一部分原因是由于缺乏互操作方法。本文提供了一种对特征模型进行直接操作方法,大大提高了产品设计的速度,同时给用户提供了更加直观的接口,能及时的将每一步设计结果反馈给用户并给以更深层次的洞察。这样的一种直接操作接口的方法不仅实现了直接操作特征模型的优点而且还保证了一个基本的特征范例实现的问题,如特征参数化和特征有效性维护。特别地,方法为实型的特征参数提供了强有力的各种 3D 柄,以及为所有

的特征操作提供的预先审查的透明体，从而提高产品模型的可编辑性和易编辑性，提高产品后期设计修改的灵活性，使产品设计更符合工程技术人员的需要，更具有商业竞争力，更具有生命力。本课题的研究具有重要的理论意义和实用价值，属于计算机图形学和 CAD 领域的应用基础研究的前沿课题。

1.1.2 国内外相关技术发展现状

传统的 CAD 系统以实体模型表示产品，仅包含产品的几何信息，它不能为下游领域如工艺规划、加工制造等提供信息^[1]。早在七十年代中期，人们就开始研究从实体模型中提取数控加工程序所需的加工信息。源于 CAD 与 CAPP 集成的研究，人们提出了特征(Feature)概念。特征名词的出现，最早见到的文献是 1978 年麻省理工学院 GOSSARD 教授指导的一篇学士论文“CAD 中基于特征的零件表示”和 1979 年的另一篇论文“轴类零件的 CAD 特征描述系统”。1975 至 1980 年间，剑桥大学 CAD 中心有多位学者从事于自动特征识别的研究，该中心的 GRAYER 于 1976 年在他的博士论文中研究了自动生成 NC 程序的方法，但其方法实际上并不是识别特征，是 KYPRIANOU 第一个提出了特征识别思想。随后在八十年代，有关特征识别的研究成为特征技术的研究热点，大量有关特征识别的研究成果发表出来。到 80 年代中后期，为避免特征识别方法中存在的困难，特别是为了使设计人员能够以习惯使用的功能形素进行设计，人们试图直接用特征进行设计^[2,3]。PRATT 和 WILSON 首先提出了基于特征的设计(Design By Features)的概念。自特征概念提出以后，特征技术的研究便蓬勃展开，国内外大量学术团体和学者的研究工作都对特征技术的发展和应作出了贡献。近十年来，特征技术研究不断深入并在实际生产设计中得到了广泛的应用。

特征不仅包含了几何信息，还包括了工程信息，加工信息，材料信息等，这些特征信息是需要保存起来的，最常见的方法对每种信息各用若干个参数来记录，然后通过对参数进行一些程序运算以复原原有的特征，这个过程就是参数化特征造型^[4,5]，特征造型是以特征为基本单位的造型系统。用参数保存信息具有一定的灵活性，可以动态地设计特征，但是只能按照固定预先设计好的模式来设计特征，不能完全根据需要设计复杂的产品。激烈的市场竞争要求产品具有竞争性，竞争性来源于完美的产品，这就势必要求在特征造型系统中模型是完美的，因此，具有限制性的参数化特征造型就不能满足这样的要求了，而需要更高级的载体来保存信息。为了解决这个局限性，人们引入了语义特征造型^[6]的概念。所谓特征语义约束，就是能够保存特征的各种信息以及这些信息之

间的关系的约束^[7]。语义特征就是用语义约束来确定具有几何拓扑关系和工程含义的特征，能全面地反映设计者的意图，具有完全几何意义和工程意义。新型的语义特征造型技术能有效地克服基于历程树造型系统的缺陷，它能使模型的修改不再依赖于模型的历程顺序，重构时间短，语义一致性能得到保证^[8]。

对于产品模型来说，最直接的设计结果便是可视的几何模型，能被设计者的眼睛直接看到^[9]。早期的特征造型系统的本质都是基于几何模型的产品设计过程，即设计者在设计时主要考虑的就是产品的外观形状大小等，对应的操作都是直接对应着几何模型的修改，它的优点是方便直观，但其缺点就是因为产品模型的几何模型的单一性和几何模型本身的局限性使得产品不能反映产品的高级设计意义，设计者的意图不能在几何模型中得到完整的体现^[10]。如何把产品的数据模型和几何模型在产品模型中统一体现出来，既让设计者的意图得到完整的体现，又让设计者能够直观的看到其设计的结果，从而成为许多学者研究的问题。于是人们提出了用基于细胞元模型的语义特征造型的思想尝试解决这些问题^[11]。

第一次明确提出细胞元这种逻辑结构的人是 PRATT，但是他选择使用具有明确体积表示的特征表示法，这样的表示法中包含一些依赖于内部或者外部的子体积。这样就产生了一个并不规则的零件模型。虽然，为了解决这个问题，提出了一些方法，但是到目前为止，还没有一个可行的实现方法。

GOMES 等提出了一个更加详细的混合表示机制，他将对于子体积的分解的进展阶段分为等级，记录了对于模型求解的历史^[12]。然而这种方法强化了细胞之间相互依赖的关系，并且有意的使用了传统的边界的表示法，这就使得这种方法的实现变得复杂且效率低下。

ROSSIGNAC 提出了一种适合特征模型的表示方法：Selective Geometric Complex(SGC)。这是一种细胞元的机制，它允许使用混合的维度来表示物体，具有内部分解的结构。理论上讲，SGC 依赖于 n 维细胞之间的连接，这种连接被一个能够指示出不同维度的细胞之间的边界范围的关联图所捕获，并使用每个细胞中的属性状态来识别哪些细胞正在被构建到零件的点集中^[13]。

上面所描述的 SGC 的特性对人们产生很大的吸引力。但是直到 1995 年，这个理论才在 GNOMES 的 Geometric Engine 上有了第一次实现。这个系统的主要目的是通过为线框造型，曲面造型和实体造型提供一种统一的表示机制来支持协同操作，克服在各种 CAX 系统中不同的产品生命周期活动中(如有限元的分析，制造可行性分析，成本估价等)对表达方法需求的不一致性。虽然 GNOMES 仅仅实现了 SGC 的一部分(Cell 的维度 ≤ 3)，它确实为处理一些上述的难以解决

的问题提供了便利的机制，尤其是为记录特征的交互操作。

BIDARRA R.提出在语义特征造型系统中，每个部件都通过相关联的细胞来表示，这样每个细胞或者完全在形状空间之内或者完全在其之外^[14]。细胞模型中的细胞是模型中特征的边界区域的点的集合。而且，细胞分解是相交驱动的。如：对于任何两个相交的几何体，有些细胞在这两个几何体的相交处，而有些细胞只存在于其中一个几何体内。这样，两个细胞永远不会发生体相交。有些情况，两个细胞可能是邻接的，这时用细胞模型中的拓扑面进行区分。这个面可以认为是有两个“面”。一般来说，位于细胞模型边界上的拓扑面只有一个细胞面。每个形状元素都是通过一系列相连的细胞面和细胞边来表示。因此，每个特征形状在细胞模型中都有一个直接的，明显的表示方法。在物主列表中，可以存放每个细胞所属的形状。细胞的属性由最后加入到细胞物主列表中的细胞的性质所决定。在对细胞模型进行特征操作时，很重要的一点就是要更新细胞模型，这样在特征依赖图中的更新信息才会反映到几何模型中^[15]。在基于历史的系统中，通过并运算和差运算来重构几何模型。而在语义特征模型系统中，只通过一种运算方法即对所有特征的形状空间执行细胞并操作。由于是并运算，所以形状空间的处理顺序与最终的细胞模型无关。由于细胞元的这一显著的特性，使得它在 CAD 造型中的应用日益广泛。

当前网络环境下对于产品数据共享和协同设计的需求和应用日益广泛，而轻量化的 CAD 模型在数据传输方面存在显著优势，研究针对带有拓扑和几何信息的轻量化模型的直接编辑具有重要意义^[16]。国外在这方面已经取得较大进展，新近出现在 CAD 系统中的类似操作可见于 Solid EdgeV17 的直接编辑以及源于 One Space Designer 的直觉式修改^[17, 18]。在当前的造型系统中，定义和修改特征参数(如决定它的位置和方向及尺寸的参数)依然是通过在对话框中输入参数来实现的。然而在当前的造型系统中，对造型的一个参数进行操作(如决定它的位置和方向及尺寸的参数)依然是通过在对话框中输入参数来实现的。这种方法的主要缺点是低效率的反馈，缺乏洞察力以及缺乏直观性。因此，研究对特征的直接操作是非常必要的。

1.2 课题来源及研究内容

本课题全称为“语义特征模型中直接操作方法的研究”，来源于国家自然科学基金项目“基于细胞元表示的语义特征造型系统”(编号 60173055)和“哈尔滨理工大学计算机辅助工业造型系统(HUSTCAIDS)”的实际发展需要。本课题

题是在二者结合的基础上进一步提出的，旨在提高对特征模型直接操作时的效率和可视性，提高整个系统的造型能力和造型效率，并切实有效的应用到 HUSTCAID 系统中，对它做进一步完善，改进对模型编辑的传统方法，使其更符合工程设计人员的需要，更好地体现设计人员的设计意图。本课题的研究具有重要的理论意义和实用价值，属于计算机图形学和 CAD 领域应用基础研究的前沿课题。

本课题从理论和系统设计开发两个方面对基于细胞元特征模型中的直接操作方法的改善进行研究、设计与实现。为了解决当前在直接操作中约束求解的速度低下的问题，将就如何把细胞元表示的特征模型所具有的高效的特性应用与直接操作的问题中进行研究，从而提高系统的求解效率，已达到能够满足用户需求的互操作的效果。因此，就近年来这方面的研究，本课题大致的基本内容包括以下几点：

1. 首先为了减轻约束求解器的负担，本文提出一种将操作局部化的方法，从而减少了直接操作过程中约束求解器要求解的约束数量。

2. 在交互阶段，用户不断的拖拽鼠标修改参数值的过程中约束管理机制要不断的利用约束求解器对各种约束进行求解。交互约束是所有约束类型中最为重要的一种，然而，目前的约束求解器只能对其进行自动检测，还不具备将不满足的交互约束自动调整成为满足的交互约束的能力。因此针对这一问题，本文提出了一种新的对交互约束进行求解的方法，使其具备上述的能力。

3. 直接操作特征的一个典型特点是具备直观性，因此本文对此引入了特征句柄的概念，使得操作过程中模型所产生的变化能够在视图中给设计者一个清晰的反映，从而使得设计者能够及时的观察到参数的改变对模型造成的影响。

1.3 论文结构

全文共分六章，具体结构安排如下：

第 2 章，介绍了语义特征造型的基本理论，并在此基础上，对细胞元进行深入的分析，解释了为什么基于细胞元的特征造型是独立于历史的。

第 3 章，分析了语义特征造型系统中直接操作的过程，以及目前存在的问题，并针对这些问题介绍了本章要解决的问题。

第 4 章，本文的重点章节，介绍了本文提出的操作局部化的方法，并给出两个局部化的算法，该方法缩小了约束求解的范围，从而提高了系统执行的效率。

第 5 章，本文的重点章节，直接操作特征参数的过程中参数的改变可能会导致与其他特征产生冲突，本章提出了一种交互约束求解器，它建立在特征参数空间的基础上，利用了蒙特卡罗取样技术，提高了自动调整交互约束使其满足条件的效率。

第 6 章，为增强直接操作的直观性，引入了特征句柄的概念，并介绍了这些句柄在直接操作过程中的应用。

第2章 语义特征造型理论及 HUST-CAID 系统

2.1 特征的定义与分类

国内外许多专家和学者对特征技术进行了研究，分别从不同的角度对特征进行了定义。PRATT&WILSON 的定义：特征是一个零件的表面上有意义的区域，共分为六类特征：通道特征，凹陷特征，凸起特征，过渡特征，域特征，变形特征。这种特征的定义与分类方法，后来被 STEP 标准采用，它主要基于形状特征分类^[19]。

特征的数量巨大，任何系统都不可能支持所有的应用领域的特征。在讨论特征造型时，有必要对特征进行分类。如果将特征分为许多特征族并统一特征的属性，那么，就可将系统设计成支持每个特征族，而不是支持每个特征的特定实现方法。其次，特征分类还可能产生一些共同的特征造型术语。再次，特征分类还有助于研制数据交换标准。从总体上看，特征可分应用特征和通用特征两大类。

1. 通用特征 下面首先给出几个有关特征分类的定义。

定义 1: 类特征(Generic Feature) 从机械产品的几何形状抽象出来的一般性特征。

定义 2: 基本形状特征(Base Form Feature) 是表达一个零件总体形状的特征。

定义 3: 附加形状特征(Additional Form Feature) 附加形状特征是对零件局部形状进行修改的特征。

通用特征就是定义的类特征，它是由基本形状特征和附加形状特征组成，而基本形状特征与附加形状特征又可进一步细分为许多子类，形成一个特征分类的树形结构^[20]。基本形状特征可以单独存在，即基本形状特征可不与其他特征发生联系，而附加形状特征不能单独存在，它必须与基本形状特征或其它附加形状特征发生联系，对它们进行修改。一个零件可以由一个基本形状特征和若干个附加形状特征来描述。

2. 应用特征 应用特征是指工业应用领域中所遇到的各种特征，这些特征有的仍以前叙述的通用特征为基础。在机械产品加工与装配领域中，特征具有几何形状和功能语义的双重含义，但主要强调的是特征的功能语义。从特征功

能语义上看,特征可分为与零件模型相关的形状特征、精度特征、材料特征、装配特征、技术特征、有限元特征和附属特征等。形状特征反映零件的全部几何信息和构造形体的高层信息;精度特征即几何尺寸所容许的加工偏差,包括尺寸公差、形位公差以及粗糙度等信息;材料特征反映的是材料成分、性能和状态这样的非几何形状信息(其中主要包括材料属性/规格、材料处理和表面处理等);技术特征反映零件在性能分析时所使用的信息,如功能参数、操作规程、技术条件等;装配特征是零件在装配时所使用的信息;附属特征反映的是与上述描述特征无关的其它信息,如标题栏、明细表等。从 CAD/CAM 集成角度去考虑,与零件模型相关的形状特征、精度特征和材料特征基本上包含了零件产品设计和制造活动中所需的全部信息。在这三种特征中,形状特征又是精度特征和材料特征的载体,后两者既可作为独立信息模块单独存在,又可直接作为产品的内部属性附加于形状特征之上,从而构成完整的产品模型。在实现时,只需要在描述产品的通用特征的数据结构中增加相应的域即可。因此,特征分类树如图 2-1 所示。

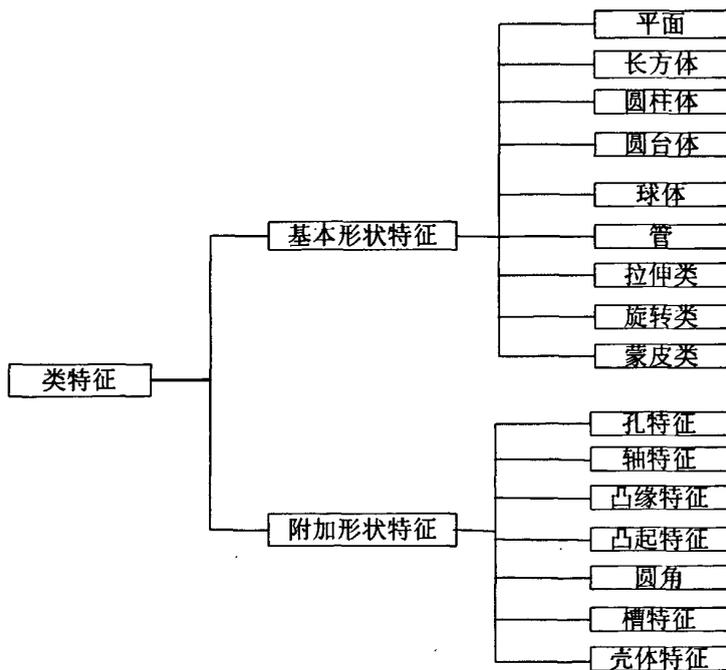


图 2-1 HUST-CAID 特征分类

Fig. 2-1 HUST-CAID feature classification

2.2 语义特征造型

特征造型是目前产品造型中的主要方法之一，而语义特征造型与当前许多造型方法相比，它是一种可以声明的造型方法，即借助于约束^[21]来声明特征属性。这些特征属性构成了特征的语义。

1. 特征参数化形状 由特征形状的元素、参数和属性构成。
2. 特征的有效性条件 由各种约束，即代数约束、尺寸约束、语义约束、相互作用约束组成。
3. 用户输入数据的接口 由几何约束、依附约束组成，它们用来确定该特征的方向和位置。

这种可以声明的造型方法其主要优点是可以自由指定约束类型，因而可以提高模型的可编辑性和易编辑性^[22]。同时在整个造型过程中语义特征造型系统有能力维护特征模型中特征的语义信息，即所有的特征都符合在特征类中定义的语义，不允许对其任意修改，以确保真正体现设计者的设计意图，提高模型的可维护性^[23]。

语义特征造型系统有两个重要的属性：特征语义的定义(即描述)和模型的维护，它们可以很好地分别进行处理：既能详细描述特征语义，又能有效地维护造型过程的整个语义。这两个重要属性决定了语义特征模型的二级结构：模型实体和细胞模型实体。模型实体是经模型操作创造的实体，被保存在第一级结构即特征依赖图中，通过特征依赖关系存储所有的特征实例和约束实例。特征依赖图是语义特征造型中模型管理器核心，它提供了特征模型的高层结构，并以一种结构化的方式包含了模型操作的所有实体和信息。第二级结构是用细胞模型表示重构的特征几何模型，细胞模型是通过一系列相连的细胞集来表示整个零件，这两个重要属性可以清楚的区分模型实体及重构几何模型的细胞模型实体。

2.2.1 特征依赖图

在特征造型系统，存在特征参数约束、几何元素约束和语义约束，他们都可能涉及到拓扑元素(如面、边和点)的引用^[24, 25]。拓扑元素引用在产品的设计过程中必须保持有效性和一致性，这要求产品模型中特征之间必须建立明确的依赖关系。如图 2-2 所示，当基体 Base 的高度增加时，即通孔特征 Through Hole 所依赖的面 f1 重新调整，则 Through Hole 特征也需随面 f1 重新定位。因此特征

依赖关系是一个动态关系，随着所依赖拓扑元素改变而改变。为此，需要特征依赖图(FDG)来记录特征间依赖关系。特征依赖图包含了模型中所有的特征实例，每个实例都有相应的形状信息、参数值和约束关系，这些实例是通过特征间的依赖关系来彼此相连的，所以特征依赖图是模型中所有实体关系的有向图，图中的每一条边都代表一种依存关系，并且指向被依赖的实例。特征依赖图用结构化的方式含盖了所有的实体和模型操作所需的信息和所有的模型操作。

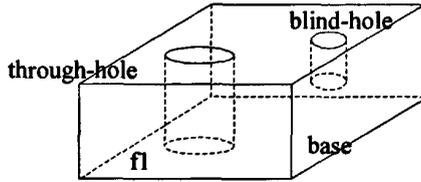


图 2-2 特征之间的依赖关系

Fig. 2-2 Dependence relations among the features

所有的模型操作是在这一级中事先完成的。我们可以随时查询出图中每个实体的当前参数值和依存关系，图中的信息供创建及维护细胞模型使用。特征依赖图为特征模型进行高层次结构上描述，是一个非循环的直接有向图，它记录特征的父特征、子特征及其约束关系。其中结点为特征，边为约束关系。对每一个特征来说，其父特征、子特征可能存在多个、一个或空的情形，图 2-3 为图 2-2 的特征依赖图。特征依赖图反映了整个产品模型特征生成的优先顺序，为基于约束的特征非线性修改和在用户级上特征编辑、特征语义解释与历程无关的实现提供了技术基础。

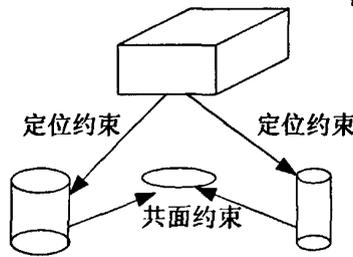


图 2-3 特征依赖图

Fig. 2-3 Feature dependence graph

2.2.2 细胞元模型

在语义特征造型系统中，每个部件都通过相关联的细胞来表示，这样每个

部件或者完全在形状空间之内或者完全在其之外。细胞模型中的细胞是模型中特征的边界区域的点的集合。

细胞分解是相交驱动的。如对于任何两个相交的几何体，有些细胞在这两个几何体的相交处，而有些细胞只存在于其中一个几何体内，这样两个细胞永远不会发生体相交。

有些情况两个细胞可能是邻接的，这时用细胞模型中的拓扑面进行区分，这个面可以认为是两个“面”。一般来说，位于细胞模型边界上的拓扑面只有一个细胞面，每个形状元素都是通过一系列相连的细胞面和细胞边来表示。因此每个特征形状在细胞模型中都有一个直接的、明显的表示方法，在物主列表中，可以存放每个细胞所属的形状。细胞的属性由最后加入到细胞物主列表中的细胞的性质所决定，如图 2-4 所示。

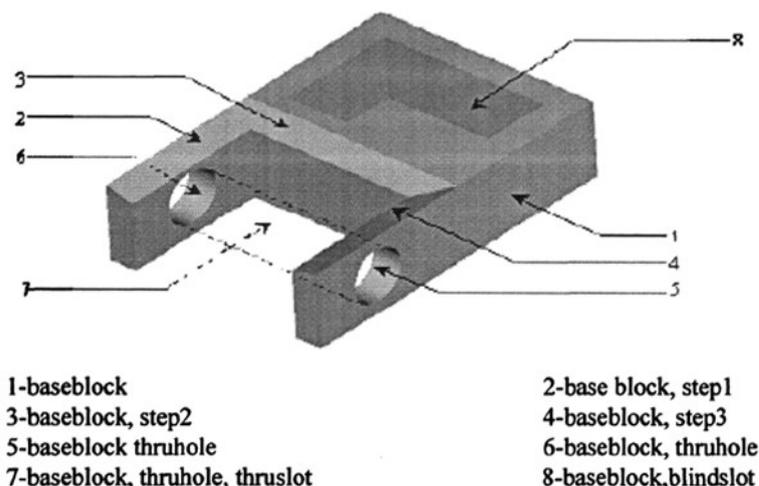


图 2-4 带有物主列表的细胞模型
Fig. 2-4 Cell models with the own list

一个细胞还包含它的体积是否表示了原料的信息，也就是说细胞具有正的属性还是负的属性，一个细胞面还包含该细胞面是否表示了特征模型的边界的信息，也就是是不是原料在一边而非原料在另一边的情况。

细胞模型中的信息可以用来检查两个特征之间是否发生了交叠，因此可以用检测交互约束。

表 2-1 提供一组访问细胞模型中数据的方法，主要有两种信息：一种是拓扑信息即与细胞模型实体相关的结构邻接信息；一种是基于属性的即与不同实体相关的数据。

表 2-1 访问细胞模型中实体的基本方法

Table 2-1 Basic method of visiting cell models

细胞模型(cm)	
cm.cells	返回细胞模型中的所有细胞集合
cm.cellfaces	返回细胞模型中的所有细胞面集合
cm.celledges	返回细胞模型中的所有细胞边集合
细胞(c)	
c.boundary	返回构成体细胞边界的所有细胞面
c.ownerlist	返回一个体细胞所属的形状列表
c.nature	返回一个体细胞的性质
细胞面(cf)	
cf.cell	返回细胞面所属的细胞
cf.ownerlist	返回细胞面所属的形状元素
cf.boundary	返回构成细胞面的细胞边的集合
细胞边(ce)	
ce.owernlist	返回细胞边所属的形状元素

2.2.3 特征依赖图的维护

维护特征模型的过程就是根据每一个模型操作来更新模型^[26]。这个过程在两个层次上都要进行：首先，修改和分析特征依赖图，其次，更新和修改细胞元模型。这部分介绍特征依赖图的维护。

造型操作可以分为两类：特征操作和约束操作。特征操作包括以下几种：

1. 添加特征 该操作产生一个新的特征类，设计者要为这个新的特征类输入一系列的初始化参数。同时，这个类中所有的约束成员也会根据用户提供的参数做相应的初始化。

2. 编辑特征 该操作允许更新特征的任何接口参数。

3. 移除特征 该操作移除模型中的一个特征以及与该特征相关的所有约束。

约束操作与特征操作类似，即向模型中添加约束条件、修改约束条件、删除约束条件等操作^[27]。对现存的特征模型进行约束操作可分为：

1. 先检查是否存在过约束或欠约束现象，若发生过约束，则应减少多余的约束条件解决约束冲突，若发生欠约束，则应增加约束条件以唯一确定模型中

的特征。例如：若指定约束条件为圆柱体的轴线与长方体的中心轴线距离 d 为 5cm，圆柱体的高度和槽的深度相同，如图 2-5 所示，若经过约束添加操作指定圆柱体的轴线与槽右手面相距 5cm，则发生过约束。

2. 由于约束条件有多种类型，改变某种类型的约束条件可能会影响其他类型的约束条件。例如改变特征的工程约束条件可能会影响到几何约束条件，加工一个金属零件时，若金属层较厚，则切削力较大，所需的加紧力也大，因而零件会产生较大的弹性变形和热变形，即零件的几何形状由于压力，温度等工程条件而改变。如果发生这种情况，则应该找出所有与之相关的约束条件经约束修改操作重新限定约束条件或在零件加工的后期阶段消除变形以保证加工质量。

3. 检查已创建的特征模型是否符合新给定的约束条件，若不符合则作进一步处理。

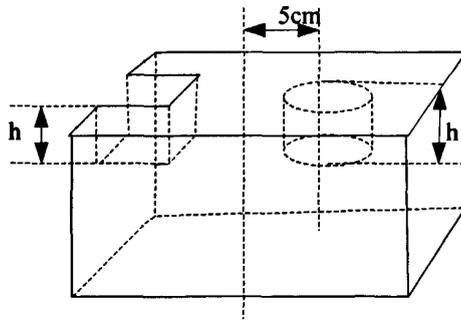


图 2-5 约束添加操作后验证特种模型的有效性

Fig. 2-5 Validating the model after adding constraints

2.2.4 细胞元模型的维护

一个造型操作的另一个步骤是更新细胞元模型，从而将特征依赖图中发生的变化反映到几何模型中。对于每一个造型操作，这个步骤的执行都包括两个阶段：第一个阶段是对细胞元模型的增量式的重新计算；第二阶段是根据存储在细胞实体中的特征信息以及当前特征之间的依赖关系来解释细胞元模型。

在基于历史的操作系统中，使用两个不相关的集合操作(并和差)来计算几何模型。而在语义特征造型系统方法中仅仅用一种集合操作——并，来计算细胞元模型：通过对所有特征中的外形域执行非规则的细胞并操作来完成。因为仅仅使用并操作，所以对外形域处理的顺序与最终的细胞模型没有关系^[26]。通过每

个外形域和目前产生的其他的细胞之间的非规则的细胞元操作来对前面所述的细胞分解进行计算，在这个过程中，关键的是能够为每个细胞和细胞面产生正确的物主列表，从而使得每个实体能够准确的知道自己所属的外形域或者形状面。

图 2-6 给出了一个不规则细胞元并操作的例子，该操作是在一个长方形的槽插入到只包含一个基体的时候发生的。图 2-6 a)所示的是细胞元并操作之前的细胞的物主列表。操作执行之后，基体的细胞被分解成为两个细胞，其中一个与槽特征所共享，如图 2-6 b)所示。当这些细胞面发生分裂的时候，每个细胞面的物主列表也发生了变化。

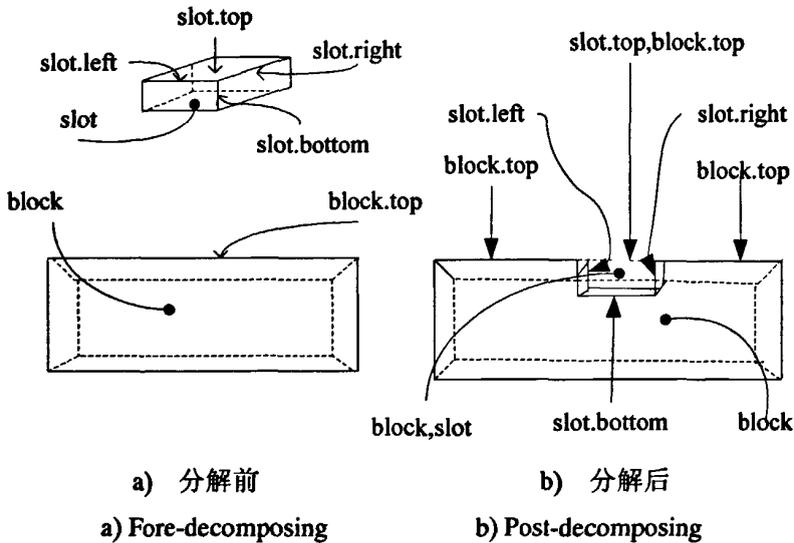


图 2-6 细胞元并操作导致物主列表的变化

Fig. 2-6 Add operation of cell leading to changes of the owner list

根据特定的特征操作，对细胞元模型的计算的概况如下：

1. 添加特征实例 一个新特征的外形域加入到当前的细胞模型中。为此，执行了非规则的细胞元并操作，对细胞元进行分解计算，同时更新改变后所影响的细胞和细胞面的物主列表。

2. 移除特征实例 此过程分为 3 步：首先将所有与该特征相关的特征从物主列表中删除；然后将物主列表为空的细胞从细胞模型中删除；最后将具有相同物主列表的邻接细胞和细胞面合并。

3. 编辑特征实例 在这种情况下只需考虑被模型操作修改的特征以及所有与之相关的特征。首先将该特征从细胞模型中移出，然后再带入新的参数将其

重新添加到细胞模型中。

2.2.5 细胞元独立于历史顺序的解释

对细胞模型的解释就是要决定表示细胞的点集是否属于这个特征即决定细胞的性质^[28, 29]。在此, 需要考虑到特征的顺序关系。

1. 决定细胞的性质 基于优先准则可以为模型中的所有特征定义一个整体顺序, 通过这些序号可以将每个细胞的物主列表排序, 细胞的性质取决于物主列表中最后一个特征的性质。设一个细胞的物主列表中有 n 个元素, 细胞的性质由第 n 个元素唯一决定, 可以得出如下属性:

- (1) 一个细胞的性质是独立于物主列表中不存在的特征的。
- (2) 如果第 n 个元素不变, 对前 $n-1$ 个元素进行重新排列时, 细胞的性质不变。

从上述两个属性可以看出, 一般情况下, 对一个细胞而言, 虽然物主列表中特征序列号不同, 但这个细胞的性质可能相同。

2. 优先准则 在进行模型操作时, 有些特征会因此发生变化, 所以特征的序列号是动态的, 根据模型操作的执行而改变。由上述属性 2 可知, 在任何时刻对特征模型结构的解释, 特征原始创建的顺序不起决定作用, 相反, 特征间真正的依存关系才是决定特征序列号的关键因素。所以, 特征依赖图中特征间的动态依赖关系始终都要反映到现有的特征造型中, 从中可以得出两个准则:

- (1) 特征依赖图中的每个边都体现了特征间的顺序关系, 如果特征 f_2 依赖于特征 f_1 , 则 f_1 的序号先于 f_2 。
- (2) 对于性质不同的两个独立的特征 f_1 和 f_2 , 如果对特征 f_2 进行模型操作后两特征相交, 则 f_1 的序号先于 f_2 。

由以上两个准则, 无论是有依存关系的特征, 还是可能发生相交的两个独立特征, 都可以确定他们的顺序关系。这样, 模型中特征的整体顺序就可以确定下来。以上只给出了部分特征的顺序关系, 由此, 还可以通过集合运算进行进一步推广: 设集合 s 中有三个元素: x, y, z 。

- (1) 传递性 如果 x 优先于 y , y 优先于 z , 则 x 优先于 z 。
- (2) 对称性 如果 x 优先于 y , 则 y 不优先于 x 。
- (3) 自反性 x 不优先于 x 。

准则 1 中的依存关系可以从特征依赖图中得出, 这样可以解决一部分特征

次序关系问题。准则 2 指出性质不同的两个独立的特征经模型操作后相交时可确定它们的顺序关系。

2.3 HUST-CAID 语义特征造型系统的体系结构

2.3.1 HUST-CAID 语义特征的定义及表示方法

在语义特征造型系统中语义特征是一个包含工程含义的几何原型的外形，是一种封装了各种属性和性能的功能要素，其中包括几何信息、有效性约束条件及特征模型接口，如图 2-7 所示。几何信息描述了特征的几何形状要素，可将构成几何体的边界区域用几何要素：如特征面、特征边、特征点来表示，例如：一个圆柱有顶面、底面、侧面三个特征面。有效性约束条件可分为几何约束(包括形位约束和尺寸约束)、拓扑约束、工程约束三类。几何约束用来约束各几何元素的固定联系。如：指定两个圆柱体的轴线平行且距离为 2 厘米对于模型中的大部分特征，并不是所有的特征面都能形成有效的特征边界，如在封闭的环路内部的特征面。我们将这种性质称为拓扑有效性，并采用边界约束(拓扑约束)来限制。边界约束有两种类型，一种是 **OnBoundary** 即特征面在特征边界上；一种是 **NotOnBoundary** 即特征面不在特征边界上。而且这两种参数类型都是参数化的，可以指定特征面是完全在边界面上还是部分在边界面上。例如：盲孔的顶面不在特征边界上，用 **NotOnBoundary** 表示，而底面在边界上，用 **OnBoundary** 表示。由于几何和拓扑约束不能完整的描述特征的功能信息，因此，需要定义更高层次的语义信息即工程约束，其中包括：资源约束(如加工的工具、安装、定位等)和条件约束(如冷却条件、加工环境、工艺流程等)。这两种类型的约束之间并不是孤立的，某类约束条件的改变可以引起其它约束条件的变化。在特征的初始化阶段，指定约束条件及参数值可能需要外界提供数据，而特征类接口恰恰实现了用户和特征实例的关联。

在语义特征造型系统中，一个特征的语义被封装在一个特征类中。也就是说，一个特征类就是该特征的描述集，每一个特征类都是针对一类特定的应用而建立的，即每个特征类都和一类形状相对应，如一个通孔和一个圆柱形状类相对应，每个形状都包括一套形状元素，如形状面和形状边。对于一个圆柱形来说包括圆柱顶，圆柱底，圆柱侧面三个形状元素，这些形状都是参数化的参数值由用户给定，或者是自动从已有的其他特征中取得，而且这些形状还要指定特征的性质是添加性质(特征的加属性)，或者是删除性质(特征的减属性)。当

所用的参数通过特征类接口输入到特征类中设置好之后就形成了形状实例(特征实例)，一个形状实例所占据的特定的空间称为形状空间。

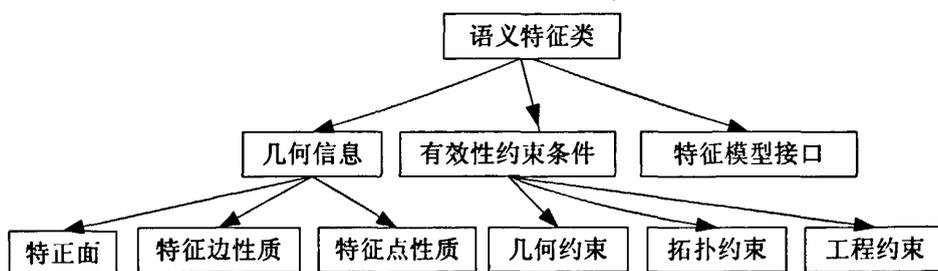


图 2-7 语义特征的组成

Fig. 2-7 Components of semantic feature

2.3.2 HUST-CAID 语义特征造型系统的体系结构

HUST-CAID 系统包括两个主要的功能体系：特征库管理器和特征建模器。

1. 特征库管理器 特征库管理器是用来描述构成特征库的特征类和管理特征库的，当系统运行时这些特征类会自动装载到特征建模器中供其使用。特征库管理器由一个特征类管理器和一个图形用户接口构成；特征类管理器由分析器、类模型、发生器组成，如图 2-8 所示。用户可以使用特征类用户接口来创建一新特征类或者编辑一个已存在的特征类。

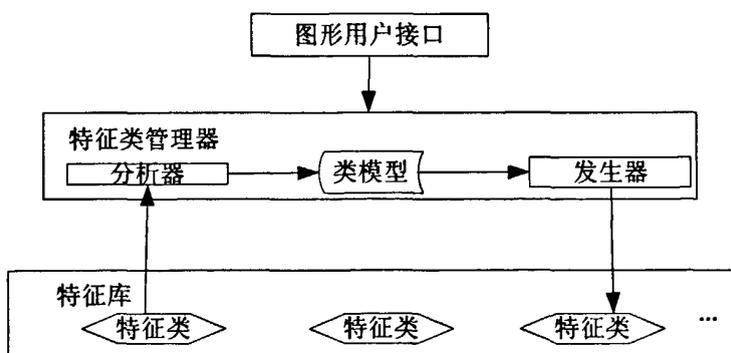


图 2-8 特征库管理器体系结构

Fig. 2-8 Architecture of the feature library manager

2. 特征建模器 特征建模器是用来创建和操纵特征模型的，如图 2-9 所示。特征模型管理器接收用户接口传来的命令，并将它们划分成多个基本任务后送

至其他管理器，它负责控制所有的模型操作及有效性维护，而且在特征模型管理器中存有特征依赖图。每个模型操作的过程都是由特征管理器负责监督执行；模型的有效性维护是借助于一个约束管理器、一个特征几何管理器和一个相互作用管理器来完成的。约束管理器负责所有的约束求解任务，维护特征依赖图中的所有约束。特征几何管理器保存细胞模型中产品的几何模型并负责进一步模型操作后的更新工作。特征相互作用管理器负责分析细胞模型，用来检测由模型操作所引起的非法特征相交情况。

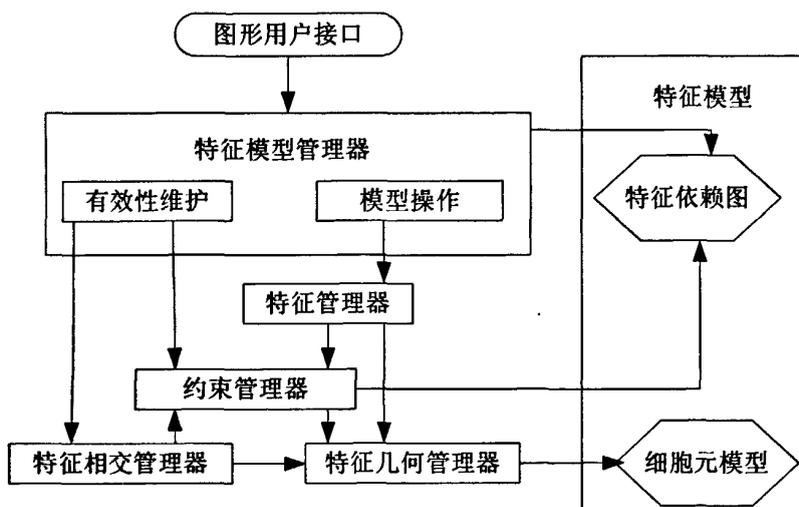


图 2-9 HUST-CAID 系统特征建模器的体系结构

Fig. 2-9 Architecture of the HUST-CAID feature modeler

2.4 本章小结

本章从特征的基本定义出发，描述了特征的分类、特征间的约束关系及求解方法，重点介绍了基于细胞元的语义特征造型系统的基本概念和基本思想，这些是进行语义特征造型研究的理论基础。然后，简单介绍了传统的基于参数化的 HUST-CAID 系统(哈尔滨理工大学计算机应用技术研究所以自行开发的一个基于特征参数化造型系统)，在此基础上介绍了新一代 HUST-CAID 语义特征造型系统体系结构：特征库管理器体系结构和特征建模器体系结构，这些为建立新一代 HUST-CAID 语义特征造型系统提供了基本保障。

第3章 语义特征造型系统中直接操作过程的分析

3.1 引言

在特征造型系统中设计者构造的模型都是基于特征的，每一个特征都定义了丰富的语义。语义的定义是通过各种约束来实现的。在特征造型系统中约束用于表达模型的特性，如定义特征面为共面的，或者将给定的尺寸限制在一定的范围内。正是由于约束举足轻重的作用，特征造型系统必须提供强有力的约束求解技术，其中，几何约束和代数求解技术应用最为广泛。

为了保证特征模型的语义得到维护，必须在每次模型操作之后对其有效性进行检查。通常是通过在对模型的约束进行求解的过程中对其有效性进行检查的：满足所有约束的特征模型才是有效的。如果一个造型系统在其构造模型的过程中能够始终维护模型的语义则称为语义特征造型系统。所以在创建修改模型过程中约束扮演着一个重要的角色。

为了使约束求解器能够应用于交互性操作人们做了很多研究工作，如用户交互结构和几何造型系统。然而，在当前的造型系统中，定义和修改特征参数(如决定它的位置和方向及尺寸的参数)依然是通过在对话框中输入参数来实现的。图 3-1 中给出了带参数的通孔特征实例。

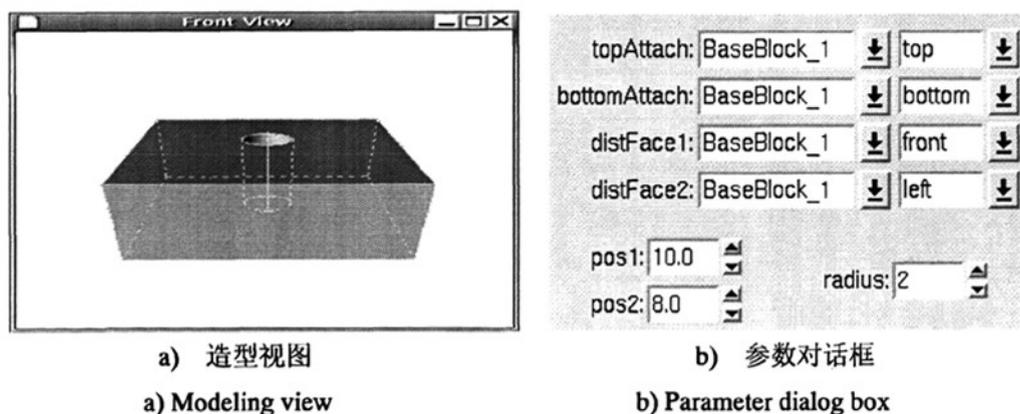


图3-1 通孔特征和参数

Fig. 3-1 A through hole feature and its parameters.

这种方法的主要缺点如下^[30]：

1. 低效率的反馈 使得设计任务的完成速度很慢。设计者每次修改特征参数都不得不等待整个系统的约束被求解，只有在此之后他才能够看到改变的参数对模型的影响以及检查其有效性。

2. 缺乏洞察力 对于模型操作的结果缺乏洞察力。当用户改变了一个参数之后，只能看到原先的模型以或修改过的模型，换句话说，对于哪部分特征发生了改变以及是如何改变的缺乏精确反馈。

3. 不直观 用户只是简单的通过对话框来改变参数，而对话框是不能够反映特征是如何受变化的参数的影响的。

由于这些缺点，用户只能被迫使用试验—错误的方法，来为需要改变的特征参数找到合适的值。

对于直接操作特征来说一个好的交互式措施是能够处理上面提到的三个缺点。在本文里，我们提出了这样一个方法：允许用户在特征模型上选择特征的参数，而且随后对其值的改变可以交互的进行，并将实时的反馈给用户模型变化的结果。当用户对模型改变的结果满意的时候，他可以选择提交接受本次参数的改变，然后让系统检查整个模型的有效性。

这种方法中关键的地方是能够对特征模型的改变所造成的影响提供实时的反馈。为了支持对特征参数值交互的修改，每秒钟要产生多次可视的反馈，所有的约束必须以同样的速度进行求解。

论文中处理了这样的情况：决定一个特征的尺寸，位置和方向的实值特征参数是由设计者交互操作产生的，而特征所在的模型是完全定义的。所谓完全定义是指设计者定义了表示模型的每一个几何约束。因此当对某个特征参数进行操作时，只有依赖于这一特征参数的那些几何方面可能会改变。注意，这些改变可能影响到许多特征，因为特征间一般都彼此依赖。

下面小节介绍了特征直接操作的方法所涉及的各个方面。

3.2 特征的直接操作过程中的技术

3.2.1 特征操作的阶段

对实值特征参数的直接操作包括两个阶段：选择阶段和交互阶段。在选择阶段选择要改变的参数；在交互阶段，设计者会改变参数值，而系统会根据改变的参数值来更新特征模型。

在交互阶段，设计者通过鼠标拖拽一个显示在特征模型上的图标来改变特

征参数的值，其中这个图标就表示这个被改变的特征参数。在拖拽的过程中，模型和它的视图不断地更新以反映对它的修改。我们只需要对模型中受改变的参数的影响的部分进行更新，从而使设计者避免了因不相关部分的改变而分心。

交互阶段需要实时的执行，因为随着鼠标的拖拽不断改变的模型视图需要实时的反馈在视图上。这里的实时是指足够快的保留移动的幻影，也就是说，幻影是指相同物体在不同位置上的连续映像，它表示了一个移动的物体。当系统每秒呈现超过10个映像时就可以将移动的幻影保存起来^[31]。

3.2.2 模型的有效性维护

对一个模型的特征参数进行操作可能会将一个有效的特征模型变为一个无效的特征模型，见图3-1。例如，两个特征间出现不符合需求的交互，或者尺寸不再满足尺寸的约束条件。无效的模型最好在操作模型的过程中就能够检查出来，并且设计者最好立刻可以获知这一情况。假如在模型操作过程中不能够检查出无效模型，例如，因为需要太多的时间检查模型的有效性，那么，应当在模型操作一结束就立刻检查模型的有效性。

然而，假如在对特征参数的操作过程中模型变为无效，那么进一步的操作应当继续进行，因为继续改变无效的参数值可能会使模型重新变为有效。例如，如果图3-1中的模型还包含一个通槽，它放置在孔的左边，并且在修改通槽的位置参数时将它从左边移动到右边，那么，当通槽和孔开始重叠时，模型变为无效，但是，如果将通槽移动到孔的后面，模型就会重新变为有效。

如果对特征参数的操作结束后模型是无效的，那么应当触发一些有效性维护机制来帮助设计者使模型重新变为有效。

3.2.3 约束管理

为了求解模型中的几何约束，需要一个约束管理机制。约束管理机制将具有复杂的设计信息的高层约束模型映射到底层的约束模型，底层的约束模型包含能够被约束求解器利用的原始约束信息。根据求解之后的底层约束模型来更新高层的约束模型。

然而一个典型的特征模型的底层约束模型一般会包含100多个约束变量。大多数的约束求解器由于约束求解算法的复杂度太高导致求解速度很慢以至于不能够满足直接操作中对求解速度的需求。

KRAMER提出了一种基于自由分析度的几何约束求解器^[32]，它将约束求解

过程分为分析阶段和执行阶段。其中，分析阶段独立于约束变量的实际值，而执行阶段依赖于分析阶段的结果和约束变量的实际值。在分析阶段，分析约束模型的结构并产生一个操作约束模型，为了满足模型中的约束操作约束模型记录了操作的顺序。在执行阶段，操作约束模型中的操作根据约束变量的实际值得到执行，结果产生一个满足所有约束的模型。假如约束变量的值需要改变若干次，那么执行阶段就要针对每一个新的满足约束的约束变量值而执行一次。分析阶段仅需要在约束模型的结构改变之后执行，也就是说，在加入、修改或删除一个约束时执行分析阶段。这一方法的一个不足之处在于每次变量值发生改变时，一般都要求解模型中一半数量的约束。

幸运的是，在特征交互阶段，仅仅只是一小部分约束模型需要求解。因为在交互阶段，通常只有一个参数被改变，模型的大部分都没有改变，也就是说他们是刚性的，因此，像这样的刚性的部分在底层约束图中被表示成单一的约束变量，这样就避免了求解所有的约束。

因此，特征直接操作的约束管理应该能够识别出模型中所有刚性的部分，并将每一部分映射成为底层约束模型中的一个独立的约束变量，这时的底层模型是在交互阶段被求解的底层模型。在交互阶段，给定正在更改的当前特征参数的值，所产生的简单约束模型可以用来寻找刚性部分的相关位置和方向。

3.3 本章小结

本章是本文的重点内容，首先在引言部分对特征的直接操作做了简单的介绍，并分析了目前的造型系统中对参数的修改所存在的问题，介绍了直接操作中所涉及的核心技术及本文要解决的问题。

第4章 直接操作中的局部化

根据对特征操作阶段的分析可知在交互阶段需要不断的对模型进行求解，并且对求解的速度要求很高。然而，实际上在直接操作的交互阶段，通常仅仅需要对模型的一部分进行修改而不需要对整个模型都进行修改，这是因为在交互阶段，大多数操作仅仅是一个参数发生改变，模型的大部分都没有发生改变。因此，没有必要对整个系统都进行约束模型求解，我们可以将受参数影响的区域提取出来，从而将对特征操作锁定在这个区域，即将操作局部化。

4.1 局部化可行性

将操作局部化的思想在基于细胞元的语义造型系统中是可行的。本文对细胞元理论做了更深入的研究。下面给出与本文相关的内容。

4.1.1 细胞元相关的定义

定义 1: 一个 m -handled n -torus ($2 \leq n \leq 3$) 被定义为一个 n -ball，它内部嵌有 m 个句柄^[33]。

定义 2: 一个 n -cell 即 n 维细胞元被定义为这样一个集合，集合内部与 m -handled n -torus 或者 n -ball 是异构的，并且它的边界必须能够分裂成有限的低维的细胞元，称作 n -cell 的面。

我们这里所说的“面”与传统的 b -rep 中所说的“面”有所不同，上面所说的“面”有可能是 0 维，1 维，2 维或者 3 维。 b -rep 中所讲到的区域，面，边和向量，分别对应这里的 3 维，2 维，1 维和 0 维的面。

定义 3: 一个细胞的混合体 K 被定义为一个细胞的有限的集合，该集合满足下面条件：

- (1) 如果 σ 是 K 中的一个细胞，则 σ 的所有的面都是 K 中的元素。
- (2) 如果 σ 和 τ 都是 K 中的细胞，则 σ 和 τ 没有交集。

定义 4: 一个细胞元模型被定义为一个特殊的细胞混合体，这个混合体是由有限的 3-cell 组成，这些 3-cell 的边界是 2 维的平面，这样，一个细胞通过一个拓扑逻辑定义的面与相邻的细胞连在一起^[34]。在一个细胞模型中所有的细胞都是相关的，不存在没有任何关系的两个细胞。

定义 5: 一个混合体 K 所在的空间用 $|K|$ 来表示，它是混合体 K 中所有点

的集合。

4.1.2 细胞元相关的特性

基于以上的定义，我们得出了以下的结论：

结论 1: 一个细胞元模型 K 的形状是嵌入在 $|K|$ 所描述的空间里。

结论 2: 在一个细胞模型中没有两个 n -Cell 不是通过拓扑逻辑定义面来连接的(根据定义 4)。

结论 3: 一个细胞模型中的两个 3-Cell 是连接的，如果它们之间有一个拓扑逻辑面存在。如果这两个 3-Cell 是通过 2-Cell 的拓扑逻辑面连接的，则称为是强连接的，否则称为是弱连接的。

结论 4: 细胞元模型可以通过两种方式来改变，增加或减少细胞模型中的细胞元。这些改变可以分为以下两类：

1. **结构改变** 通过从细胞模型中增加或删除细胞元而不改变 $|K|$ 来使细胞模型 K 发生结构改变，也就是不改变 $|K|$ 中的点，不增加额外的点，也不从其中删除任何点。

2. **形状改变** 通过改变 $|K|$ 中的点来使得细胞模型 K 发生形状变化，即向 $|K|$ 中增加点或从其中删除一些点。

结论 5: 因为一个细胞模型 K 是由有限个 3-Cell 的联合组成的，所以改变 K 的形状等于向 K 的细胞的子集中增加或删除点集，也就是说不是所有的 K 中细胞都要被影响。形状变化可以以两种方式来影响细胞模型中存在的细胞：形状变化可以通过向一个 3-Cell C 中增加或删除点集来直接影响 3-Cell。这将导致向 C 中添加或删除细胞；当向与 3-Cell C 邻近的 3-Cell 中增加或删除点集时对其造成间接影响，即会导致 C 的结构发生改变。

从定义和结论可知细胞元是点的集合，改变 K 的形状等于向 K 的细胞的子集中增加或删除点集，也就是说不是所有的 K 中细胞都要被影响。如果能够识别出这些发生变化的细胞就可以对形状中发生改变的地方进行定位。因此，细胞本身所具有的这些特性为把变化的细胞从特征中分离出来提供了基础。

4.2 操作局部化的过程

操作局部化过程的主要任务就是将受参数影响的区域提取出来，这根据特征操作的两个阶段，局部化过程也分为两个阶段：选择阶段和交互阶段。

在选择阶段，当需要修改的参数被选定时，这个参数所影响的特征区域

FR(Feature Region)已经可以确定,因此这一阶段局部化的任务就是将这个 FR 从整个 FDG 中分离出来,此时只是将操作在特征级的范围内进行局部化。

在交互阶段,设计者通过鼠标拖拉代表特征参数的图标的第一次动作中,操作所影响的细胞集合可以确定,因此这一阶段的任务就是将这些发生变化的细胞从 FR 中分离出来,作为一个细胞元变化模型 CCM(Changed Cell Model)。此时是将操作局部化的范围锁定在 CCM 上。

结合上述的分析,得出一个对特征模型进行直接操作的过程模型,如图 4-1 所示。

当有鼠标输入的时候,根据鼠标输入对 FDG 做一次全遍历,并根据约束求解器的求解结果将受到参数影响的所有相关联的特征提取出来作为 FR。这个过程相当于对整个系统做了一次约束求解。

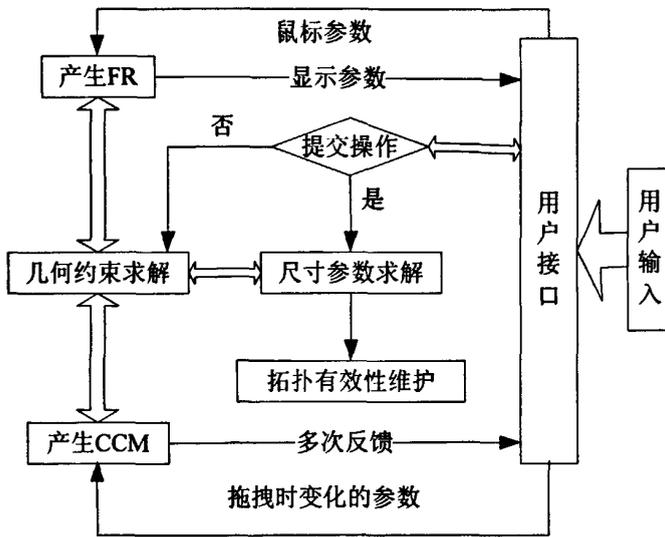


图 4-1 直接操作过程模型

Fig. 4-1 Process model of Direct manipulation

当用户用拖拽的方式第一次改变参数值时对 FR 做一次遍历,并根据约束求解器的求解结果将受到参数影响的所有 Cells 提取出来作为 CCM。这个过程相当于对于 FR 做了一次约束求解。

之后用户每次对参数值的改变都仅仅是对 CCM 进行约束求解,直到用户提交了本次操作才对整个系统做一次有效性维护。

4.3 相关的数据结构

从以上分析中可以看出操作局部化的关键就是如何产生 FR 和 CCM，因此本文分别给出算法来解决这个问题。为了更好的理解这两个算法，下面给出与算法相关的数据结构：

```

typedef struct Feature { //定义特征
    Cell*      pCell; //一个特征由 n 个细胞元组成
    Box        box;  //特征的区域所在的立方体盒子
    Relation*  pRelationHead; //第一个主客关系;
    Constrain* pConstrain; //各种约束
    Feature*   pParent; //父特征
    double     RefMatrix[4][4]; //特征句柄的参考矩阵
};

typedef struct Face { //定义细胞面
    Point      face_point[4]; //一个细胞面由 4 个点来决定
    List       list_COL; //每个细胞面都拥有自己的物主列表,表明自己隶、
    //属于哪个特征
};

typedef struct Cell { //定义一个细胞元
    para1, para2, para3, ...; //公共属性, 如性质, 形状, 颜色等
    Face*      face; //一个细胞由 n 个细胞面组成
    Feature*   pOwnerFeature; //该细胞的物主特征
};

typedef struct Constrain { //定义约束
    Constrain  Edge_Constrain // 边界约束
    Constrain  Postion_Constrain //定位约束
    Constrain  Size_Constrain //尺寸约束
    ...
};

typedef struct Relation { //定义主客关系
    Feature *   guestfeature; //客特征
    Relation*   next; //下一个主客关系
};

```

```

    Constrain* pConstrain;//各种约束
};
typedef struct FDG{//定义特征依赖图
    Feature* hostfeature;//主特征链头指针
    int      NumFeature;//当前特征的个数
};

```

4.4 算法描述

4.4.1 产生 FR

算法 `Create_FR` 基本思想：该算法主要有两个过程：首先找出用户选取的特征，这是通过特征 `box` 粗判和特征中的细胞的细判来完成的；其次利用图的深度遍历方法来遍历选取特征的所有客特征。其目的是找出选取特征变化所能够影响到的所有特征。

下面给出算法 `Create_FR` 描述：

Step1: 由于每一个特征都定义一个可以包容自己的形状盒子，所以首先判断 `pt`(点击鼠标时的当前位置)是否在特征所定义的 `box` 中，如果不在，则查找 `FDG` 中的下一个特征；如果特征的 `box` 包含 `pt`，GOTO Step2。

Step2: 遍历该特征的所有细胞，查看是否有细胞包含 `pt` 点。如果有，则跳到步骤 3。如果没有，就继续遍历该特征的细胞。如果遍历完了该特征所有细胞，也没有发现有细胞包含 `pt` 点，GOTO Step1。

Step3: 把查找出来的特征作为主特征，然后从这个选中特征的主客关系 `pRelationHead` 中取出一个客特征，GOTO Step4，重复该过程，直到遍历完了所有的客特征，GOTO Step5。

Step4: 遍历当前特征主客关系 `pRelationHead`，取出来一个对应客特征，查看该客特征是否在特征列表(`CFList`)中，如果不在，则把该客特征存放到 `CFList` 中。跳到步骤 4，否则继续遍历 `pRelationHead`，直到遍历完 `pRelationHead` 所有客特征以后，GOTO Step3。

Step5: 把产生的 `CFList` 中的特征加入到 `FR` 中。

经过这个算法步骤就可以将 `FR` 从整个 `FDG` 中分离出来，只是这个时候只是对操作在特征级的范围内进行了局部化。在交互阶段，设计者通过鼠标拖拉代表特征参数的图标的第一次动作中，便有足够的信息在细胞级进行局部化。

4.4.2 产生 CCM

动态的将产生变化的 Cells 从 FR 中分离出来有两个过程：暂时的分割和细胞元模型固有的分割。这两种方法都保证了将添加删除和修改参数的编辑操作局部化。

1. 暂时分离 这个分割的过程是通过增加内部的 2-Cell 将模型细分为 3-Cell 使其暂时的发生结构变化以方便编辑操作在这里定位。这种分割只是改变了模型的结构而没有改变其形状。该阶段的分离有下面两种方法：

(1) 通过划分平面分离：每一个模型中与划分平面相交的 3-Cell 被细分为多个 3-Cell，这些位于划分面的同一边的 3-Cell 的集合作为一个 FR。

(2) 通过边界盒划分的分离：这种分离方法有助于自动的进行分离模型。步骤为：首先，找到 FR 的非实体的边界盒；其次，找出特征变化后影响到边界盒的一个面，然后采用二分法找出距离这个面最近的不变化的面(二分法的最小步长为边界盒的 5%)，然后重建边界盒；最后，重复上一步骤的过程，处理所有的有变化的面，最终产生出一个最小的边界盒。

2. 细胞模型所固有的分离 根据定义 2 可知一个细胞元中的所有细胞都是有联系的，换句话说任何两个细胞元都是独立的，这就是细胞所固有的分裂特性。这种方法假设这个细胞模型中的每个特征形状中将要附加或移除的细胞都具有明确的区域表示。这种分割的方法，计算量很少，因为细胞模型根据细胞的体积固有的保持了特征的踪迹。

下面给出 Create_CCM 算法描述：

假设把当前用户选择的特征定义为 *SelectedFeature*。

Step1: 从 FR 中取出一个特征(*CurrentFeature*)，GOTO Step2，当处理完 FR 中所有的特征，GOTO Step7。

Step2: 取出 *CurrentFeature* 的 *Box* 和 *RefMatrix* 以及其父特征对这个特征的约束，然后取出这个父特征对应的父特征的约束，同时和 *CurrentFeature* 的约束进行合并，重复该过程，直到取到的特征是 *SelectedFeature* 为止，并把合并出来约束保存到 *Constrailist*。其目的是为了把 *CurrentFeature* 和 *SelectedFeature* 之间所有的间接约束条件变成直接约束条件，从而方便以后利用。

Step3: 利用约束求解器对 *Constrailist* 中约束进行求解，从而确定取出来的 *Box* 和 *RefMatrix* 是否发生变化。如果 *Box* 和 *RefMatrix* 没有变化，GOTO Step1；如果 *RefMatrix* 有变化，此时把该特征的所有细胞全部加入 CCM 中，GOTO Step1；如果 *RefMatrix* 没有变化，而 *Box* 有变化，GOTO Step4。

Step4: 把 *Box* 作为一个边界盒，并找出在步骤 3 中 *Box* 所变化的面 (*ChangedFace*)，采用步骤 5 来处理 *Box* 和 *ChangedFace*。在处理完所有的 *ChangedFaces* 后，GOTO Step6。

Step5: 在 *Box* 中与 *ChangedFace* 对应平行面之间选取一个新的面，然后利用约束求解器来处理这一个面，如果这个新面发生了变化，则把该面当成变化面，否则把变化面和边界盒之间的区域作为新的边界盒，重复此过程，直到变化面和与此平行的边界盒中对应面的距离为原边界盒对应的距离的 5%，GOTO Step4。其目的是为了采用二分法进一步分割出受 *SelectedFeature* 影响的空间。

Step6: 处理边界盒中所有的 3-Cell，遍历边界盒中的所有细胞，对每一个细胞都利用约束处理 3-Cell，如果该 3-Cell 发生了变化，则加入到 CCM 中，否则不予处理，直到遍历完边界盒中的所有细胞，GOTO Step1。

Step7: 结束。

综上所述将操作局部化方法有效的减少了对整个模型进行约束求解的次数，因为用户对特征的一次编辑动作中，只需要对整个系统做两次求解。虽然在用户拖拽过程中需要重复的对 CCM 进行求解，但通常情况下 CCM 的数据量要比整个系统少的多，因此，如果设 m 为拖拽时需要对系统求解的次数， n 为整个模型的约束数量，则原始的约束求解器算法复杂度是 $m \times n$ ，而引入本文的局部化算法后的约束求解器的算法是 $m \log n$ 。所以当 m 远远大于 n 时，该方法大大提高了编辑特征参数的效率，为直接操作特征的实现和增强操作的直观性提供了基础。

4.5 本章小结

本章对细胞元理论做了更深入的研究，并结合特征直接操作的过程提出了一种基于细胞元表示的分割 CCM 的方法，将操作局部化，缩小了约束求解的范围，从而提高了系统执行的效率。

第5章 直接操作中的交互约束求解器

约束在特征造型中扮演着一个很重要的角色，因为它们为存储特征模型的设计意图提供了可能性^[35]。约束可以被定义为特征模型中的一个特征的属性，规定了一种特定的需求，例如，一个特征的圆柱形状的半径应该是 5。约束也被定义为一个特征模型中几个特征之间的关系，例如，一个特征的值必须与另一个特征的值相等。

特征造型中有很多种约束类型。例如，在特征造型的制造工艺阶段，尺寸约束被用于定义形状的尺寸大小，也就意味着不符合这个尺寸规定的值在制造过程中是无效的是不允许的。再比如，在特征造型的装配阶段，几何约束被用于规定方向，也就是说在装配过程中部件必须按照规定的方向移动才能完成连接操作。

交互约束是一个相对比较新的约束，限制了可能相交的约束之间的范围^[36]。在本文中只讨论了一种比较重要的交互情况，即两个特征形状在空间上重叠的情况。例如，一个盲孔与一个较大的通槽完全重叠，结果是这个盲孔被这个通槽完全吸收了。

这一部分提出一种交互约束中自动调整约束的算法，也就是一种当一个特征模型中存在不满足交互约束的时候能够自动将其调整直到满足约束的算法，以及该算法在 HUST-CAID 特征造型系统中的实现。这个系统支持自动检测和自动调整多种类型的约束。然而对与交互约束而言，直到现在仅仅支持对它的自动检测，也就是能够自动检测到一个特定的模型是否满足一个交互约束，但是还不能自动将一个不满足条件的约束调整为满足条件的状态。将本文描述的算法应用到 HUST-CAID 系统中可以扩展这种能力。

5.1 约束模型

一个特征模型是由特征构建而成，并在高层表示成为特征依赖图，在底层表示成为约束模型。在特征依赖图中，特征之间由所谓的依赖关系联系。例如，当一个已经存在的特征中的特征元素被依附于或者被重新定位于另一个特征中的时候，就会在这两个特征之间产生一个依赖关系，如图 5-1 所示。在这种情况下，这个新的特征依赖于这个已经存在的特征。特征之间的所有约束以及它们之间的依赖关系都在依赖图中描述。

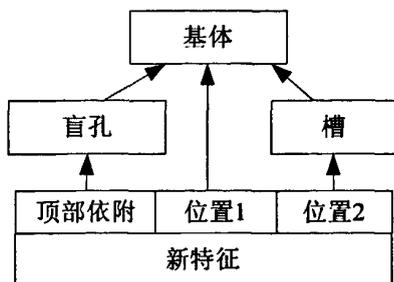


图 5-1 新特征加入之后的特征依赖图

Fig. 5-1 Features dependency graph after adding new features

约束模型是用来检查特征模型是否与存储的设计意图相符合。约束模型是由变量以及这些变量之间的约束关系构成^[37]。在约束模型中变量表示成为实体，并拥有由约束定义的值。例如几何和代数变量。几何变量表达了特征面的位置和方向。它们之间的关系由几何约束来表示，如共面也就是相关的变量的位置是相同的，在面上也就是一个变量的位置在由另外一个变量表示的平面上。代数变量表示了特征参数的实值，例如尺寸。他们之间的关系由代数约束来表示，如相等也就是相关的变量是相等的，和也就是两个相关的变量的实值的和应该等于第三个相关的变量。

如果定义的关系是有效的则称约束是满足的，否则称约束是不满足的^[38, 39]。不满足的约束可以通过改变已经定义好的变量的值重新变为满足的。

求解一个约束模型也就是将所有的约束变为满足的，包括两个步骤：检查约束是否是满足的和将不满足的约束变为满足的约束。

检查约束也就是执行一个布尔测试以决定变量当前的值是否满足已经定义好的关系。例如，检查两个几何变量之间的共面约束，就是检查两个变量是否是同一个位置^[40]。

调整不满足的约束为满足的约束就是根据定义的关系来调整变量当前的值，并且确保其他已经满足的约束不会变为不满足的约束。

5.2 交互约束

特征之间的交互是指能够对彼此的功能意义造成一定的影响。有时候这种影响是在有一定距离的特征之间发生的，也就是说在没有发生重叠的特征之间发生的。例如，从制造的角度去看，两个平行的槽非常的近以至于当第二个槽

的变动可能会导致它们之间的隔墙被破坏^[41]。

本文解决了一种非常重要的特征交互类型，那就是特征形状在空间上重叠的情况。这种交互在特征中会产生很大的影响，所以防止这种情况的发生是非常重要的。

表 5-1 给出了造型系统中能够被交互约束防止的交互类型。例如，对一些特征来说一个分裂交互约束规定了不允许分裂交互的发生。

目前为止，只能自动的检测交互约束。HUST-CAID 造型系统中的交互约束的检测机制使用的是特征模型几何的非拓扑表示，即所谓的细胞元模型，它包含了所有特征的有用信息。

细胞元模型将几何模型表示为一个任意形状的非离散细胞的集合，其中每一细胞要么完全在一个特征形状的内部，要么完全在其外部。特征之间相交的部分会产生新的细胞。每个细胞包含与该细胞在体积上重叠的特征的信息，每个细胞面包含与该细胞面有重叠的特征面的信息。另外，一个细胞还包含它的体积是否表示了原料的信息，也就是说细胞具有正的属性还是负的属性，一个细胞面还包含该细胞面是否表示了特征模型的边界的信息，也就是是不是原料在一边而非原料在另一边的情况。

表 5-1 交互类型

Table 5-1 Interactive type

交互类型	描述
分裂	将一个特征的边界分裂成两个或者更多的子集
分离	引起一个正特征或者其中的一部分与模型的分离
边界清除	引起一个(或部分)负特征的一个闭合面不通
体积清除	引起部分负特征的体积不通
封闭	引起一些负特征的体积变成模型内部的一部分
吸收	引起一个特征完全终止它对模型的作用
几何	引起一个名义上的参数值和实际的特征几何不相匹配
变形	引起一个特征实例展示出另外一个特征的特性
拓扑	与给定的特征的边界被侵犯一致

细胞模型中的信息可以用来检查两个特征之间是否发生了交叠，因此可以用检测交互约束。作为例子，这里给出一个检查体积清除交互约束的算法的例子。

当一个负特征的体积完全被一个正特征所吸收，这两个特征就会发生体积清除交互。如图 5-2 所示，一个通槽特征的部分体积被一个圆柱突台所吸收。

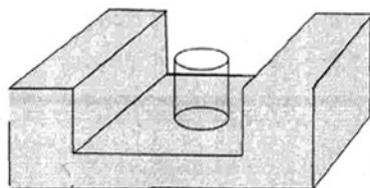


图 5-2 体积清除交互

Fig. 5-2 Interaction of removing volume

这种交互约束类型的检测算法中会检查与负特征的形状交叠的所有细胞是否都有负属性。如果是，则约束是满足的，否则，约束是不满足的(参见算法 1)。

算法 1:

Foreach cell c in cellular model m do

If subtractive feature s overlaps with c then

If c has nature additive then

Return TRUE

Return FALSE

虽然一个交互约束是在受约束的特征的形状上定义的，但是它所涉及的这个形状可能会与模型中其他的一些特征的形状有关系，所以实际所涉及的特征个数依赖于模型的结构，例如，模型中的这些特征是如何彼此定位的，以及这种定位关系还可能随着以后创建的约束而改变。这是一个交互约束满足算法中最难以解决的困难所在。

5.3 满足交互约束

向一个约束模型中添加一个新的约束并使其满足有两种方式：扩展约束求解器的约束满足能力，使其能够支持新的约束；创建一个新的转换算法，该算法能够将新的约束转换成一组已经存在的能够被满足的约束类型。例如交互约束，可以创建一个将它们转换成为几何约束的算法，因为系统中已经存在自动满足几何约束的算法；或者可以创建一个新的约束满足算法，能够直接满足交互约束。这里将对这两个方法分别做一介绍。

5.3.1 转换成几何约束

本文中所处理的交互约束只涉及特征形状之间在空间发生交叠的情况。这种情况涉及到模型的几何表示，因此使用几何约束表示这种交叠的约束是合理的。

例如图 5-3 中盲孔顶部入口面上的一个边界清除交互约束，圆柱突台的位置必须以这样的方式来约束：它的底面不能够和盲孔的顶部入口面重叠，也就是两个特征面的圆心的距离之和不能小于他们的半径。这个例子中，一个单一的几何约束足以描述这种约束。

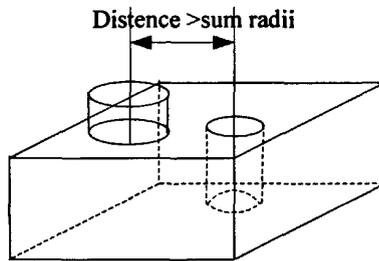


图 5-3 盲孔的一个边界清除交互约束

Fig. 5-3 A boundary clearance interaction of the blind hole

然而，这种方法存在严重的问题：首先，一个单一的交互约束可能需要转换成许多个其他的约束。例如，如果一个通孔依附于一个存在 n 个圆柱突台的表面，那么通孔顶部入口面上的边界清除交互约束就需要转换成 n 个几何距离约束。其次，也许当一个新的特征加入模型时，为了转换交互约束可能要产生一些新的额外的几何约束，例如，当更多的圆柱突台加入到图 5-3 中时，额外的几何约束就需要也随着加入才能够精确的表达交互约束。最后，在算法中为一个任意的模型中的交互约束产生一组正确的几何约束是非常困难的，例如，根据发生交互的特征是一个盲孔(见图 5-4 a))还是一个长方形的袋装槽(见图 5-4 b))，需要决定是否应该产生一个几何约束来避免一个通孔特征上的分裂交互。除了所涉及的特征的类型以外，他们之间的相对位置和方向也会对是否要产生几何约束有所影响。

大量的几何约束和额外产生的约束的问题可以通过从交互约束转换成几何约束的递增方法来解决。这种方法将会重复的检测模型并为不满足的交互约束产生几何约束，直到不再存在不满足的交互约束为止。因此如果没有交互发生就不会产生几何约束，但是，如果有很多交互连续发生也会导致大量的几何约

束产生。然而，为一个任意的模型产生一组正确的几何约束是难以实现的。

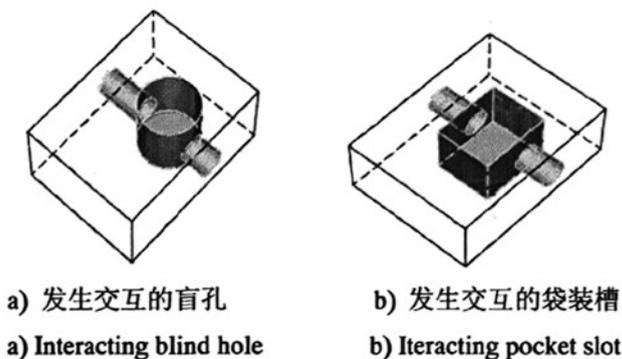


图 5-4 发生交互的盲孔和长方形的袋装槽

Fig. 5-4 Interactive features of blind hole and rectangular pocket slot

5.3.2 直接满足交互约束方法

为了避免从交互约束转换成几何约束所产生的问题，提出了一种新的交互约束的直接满足算法。该算法是基于模型的参数空间的。

在模型中找到没有发生空间上重叠的特定特征的问题类似于空间规划问题。一个典型的例子就是在汽车的车箱里为另一个提箱找到空间的问题。本文中所描述的满足交互约束的方法中应用了构造空间的思想^[42]，只是将特征模型中的参数空间代替构造空间。

参数空间是一个多维的空间，每一维表示了特征的一个参数变量，也就是，一个特征参数可能会在造型系统中被修改，因为用户并不关心参数的实际值。空间中，根据一个或者更多的变量参数的改变，每个点表示了一个来自原始模型的一个模型实例，也就是由用户定义的模型。参数空间的一些区域表示了模型满足了所有的约束，其他的区域表示了模型中的一些约束还是不满足的。

这里给出一个特征模型的参数空间的例子。特征模型由一个带有盲孔的基体，一个脊形和一个圆柱突台。由于交互约束，盲孔的入口面不应该被其它的任何特征阻挡，而且脊形的底面和圆柱突台的底面应该完全依附与它们当前所依附的面。另外，脊形的高度和圆柱突台的位置参数是可变的如图 5-5 a)所示。只有在圆柱突台位于盲孔的左边，也就是一部分位于基体上而另一部分位于脊形上的时候模型中的约束才是满足的如图 5-5 b)所示。这个模型的参数空间是二维的如图 5-5 c)所示。模型中所有的约束都是满足的部分在参数空间中表示为闭合区域部分。

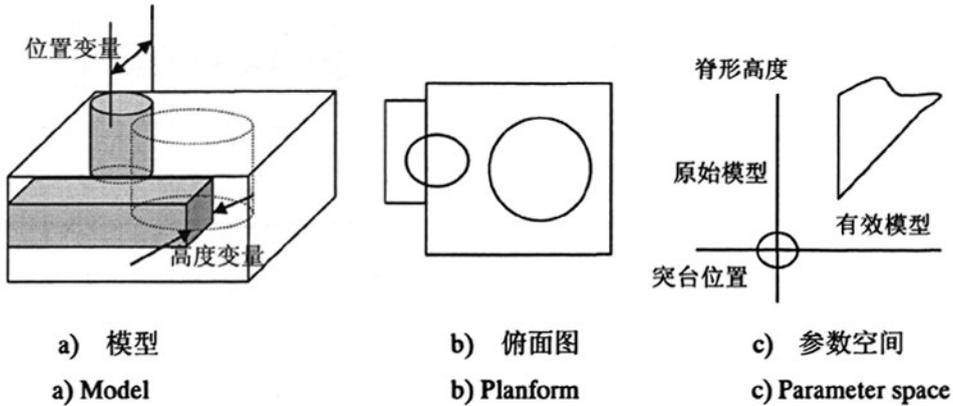


图 5-5 一个特征模型和它的参数空间

Fig. 5-5 A feature model and its parameter space

在直接约束满足技术中，为了找到能够满足所有约束的参数值(包括交互约束)需要在特征模型的参数空间进行取样。对于每一个样本，都可以用几何约束求解器来求解几何约束，用代数约束求解器来求解代数约束，用第三部分所描述的交互约束检测算法来检测交互约束。

为了提高找到表达一个满足所有约束的模型的点的可能性，这里利用了蒙特卡罗技术^[43]。一般来讲，蒙特卡罗方法由于能够真实地模拟实际物理过程，故解决问题与实际非常符合，可以得到很圆满的结果。在这个过程中能够减少样本量而同时提高每个点的精确度。参数空间的取样是用下面的方法来实现的。

为了尽可能的避免对模型做不必要的修改，将参数空间分成了很多个子空间，划分的方法是根据相对模型的修改的程度(相对于原始模型)来进行的。子空间的排序也是根据这个修改的程度以及子空间中至少包含一部分满足所有约束的模型的可能性来进行的。图 5-6 给出了一个模型中参数空间的子空间的例子。样本是在子空间序列中随机的产生的，从序列中的第一个子空间开始，然后根据序列中的顺序进行其他的子空间。

子空间中表示至少一部分可满足约束的模型的可能性依赖于不满足约束的参数值的改变所影响的范围，例如，通过改变表示两个特征的相对位置的约束参数值来移除这两个特征形状的相交性的可能性比通过改变只会影响模型中完全不同的区域的参数值来改变要大的多。图 5-5 中，改变位置参数来移除相交性的可能性要比改变高度参数大得多。

处理样本的过程会为每个子空间产生一些随机的样本，下面对此做了详细的阐述。

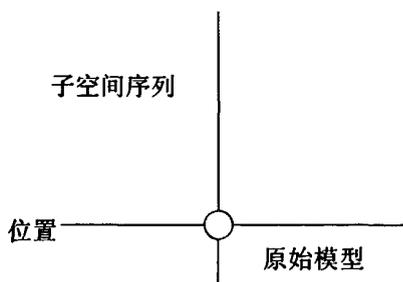


图 5-6 一个模型中参数空间的子空间

Fig. 5-6 subspaces of the parameter space of a feature model

5.4 交互约束求解器

5.4.1 交互约束求解器的结构

交互约束求解器包括三个部分：已经存在的模型有效性处理部分，它在用户修改模型的时候求解所有代数和几何约束；计算细胞元模型部分以及检测交互约束部分。在新的求解器中，如果系统发现模型处于无效状态而用户又没有停止对其求解，这个时候将会从参数空间中产生一个样本，这个样本使得模型变为有效状态，否则，约束求解器将会终止。如图 5-7 所示。

5.4.2 交互求解器的实现

参数空间的实现是通过将每一维分成一系列的间隔来实现的。在这个预处理的过程中，参数空间随着每一维的划分而分成许多子空间。每一维划分的间隔是个固定的长度。

子空间的顺序首先由它们与参数空间中代表原始模型的点的距离来确定，紧接着再由子空间中至少存在一部分满足约束的模型的可能性确定。

子空间与参数空间中代表原始模型的点的距离通过将子空间组成不同的外(Shell)壳来确定。第一个外壳由表示原始模型的点所在的那些子空间组成，第二个外壳包含了那些包围第一个外壳的子空间，如图 5-8 所示。子空间到参数空间中代表原始模型的点的距离等于它所在的外壳的序号。

子空间表示了至少一部分满足约束的模型的可能性依赖于参数改变的意义(相对与代表原始模型的点)。特征模型中表示位置和方向的交互特征参数发生改

变的数量越多则这种可能性越大。

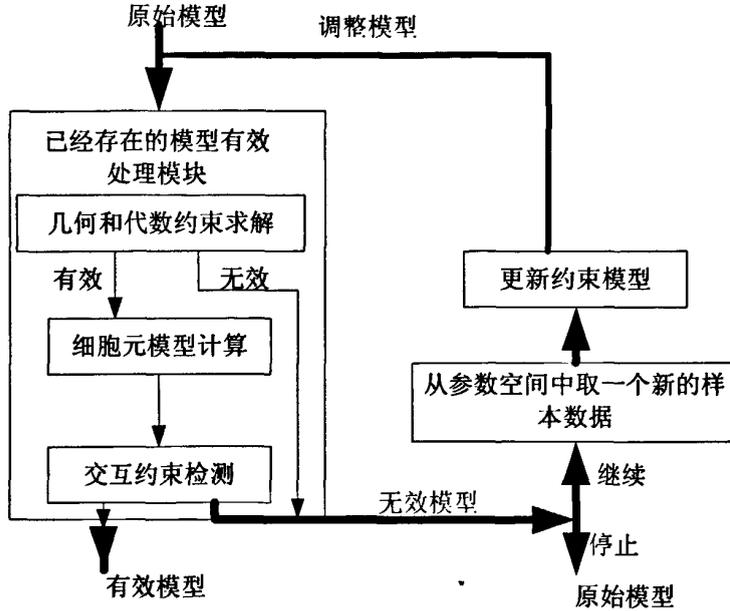


图 5-7 交互约束求解器的结构

Fig. 5-7 Structure of the interaction constraint solver

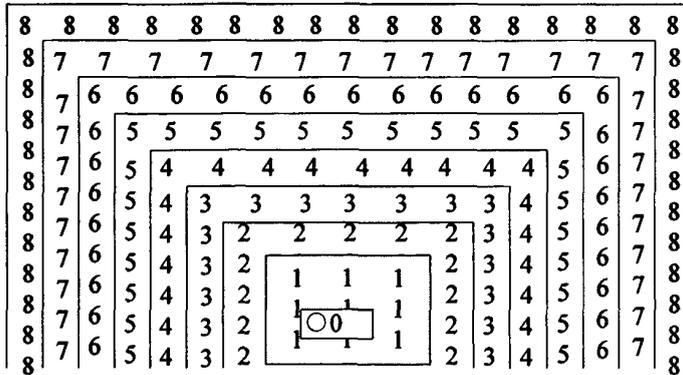


图 5-8 两维参数空间中的子空间外壳

Fig. 5-8 Shells of sub-spaces in a two-dimensional parameter space

与交互特征的位置和方向有关的特征可以在特征依赖图中找到。例如，图 5-9 中决定盲孔特征和突台特征的位置的特征是槽和基体特征。

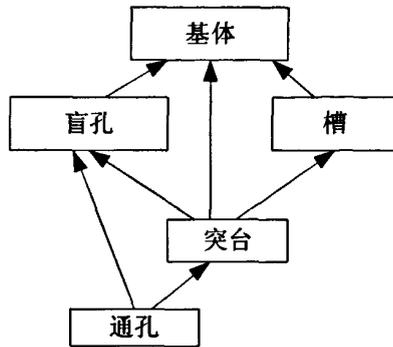


图 5-9 特征依赖图

Fig. 5-9 Feature dependence graph

蒙特卡罗技术在此用来随机的产生子空间的样本，这种技术的概率分布均匀。根据蒙特卡罗的概念，提高了在一个子空间中找到有效模型的可能性。

5.5 实例

本文给出了一种自动求解交互约束的方法。这个方法扩展了当前自动检测交互约束的方法，并提供了自动满足这种约束的能力，也就是自动将模型调整成为一个有效的模型。

这个基于模型的参数空间的方法比将交互约束转换成为几何约束的方法有效的多。主要原因是转换成几何约束的过程中一般都会产生大量的几何约束，并且为一个任意的模型产生一组正确的几何约束的算法是难以实现的。这里将给出一个使用交互约束求解器的例子，这个例子基于图 5-10 所示的模型。在这个模型中，基体的长度，通孔特征的深度以及两孔之间的距离是可变的。

现在，要在模型中的两个通孔之间添加一个长方形的槽。由于在两个通孔特征之间没有足够的空间，因此在通孔附近的交互约束变为无效的如图 5-11 a) 所示。

造型系统视图通过为模型的变量参数创建一个参数空间来自动的满足有效约束，然后从参数空间中不断的取样直到模型满足了所有的约束为止。在没有找到有效模型之前，样本依然会导致通孔特征的交互约束是无效的，如图 5-11 b) 所示。最终找到的有效模型如图 5-11 c) 所示。

在这个例子中，为满足有效约束两个通孔的位置以及它们之间的距离发生了变化。通槽的深度变量和和基体的长度变量都没有受到影响，因此没有被约

束求解器改变。

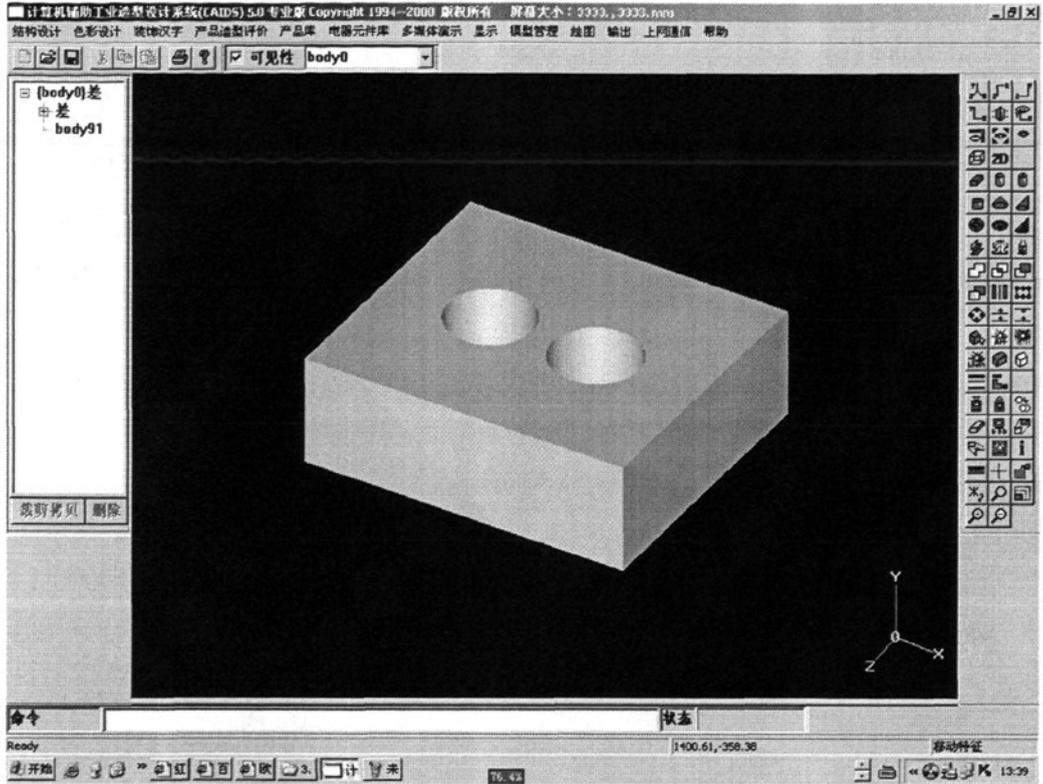
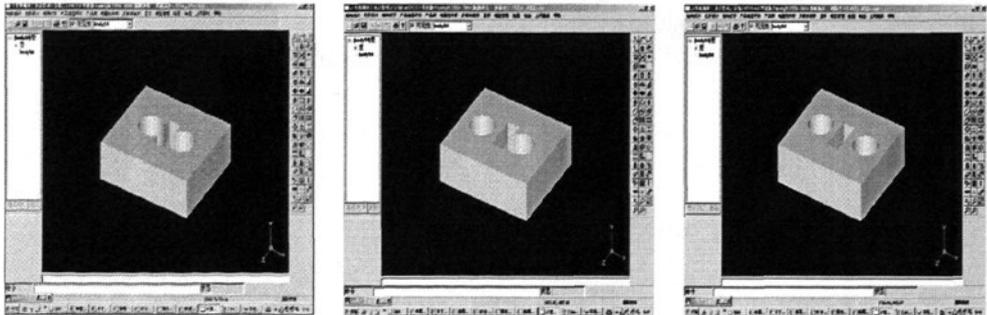


图 5-10 一个产品例子
Fig. 5-10 An example product



a) 发生交互
a) Interacting
b) 调整参数
b) Adjusting
c) 最终结果
c) The result

图 5-11 交互调整过程
Fig. 5-11 Process of Interacting

5.6 本章小结

到现在为止，交互约束求解器还没有考虑任何与约束相关的特定信息，例如为了满足一个给定类型的交互约束，应该以什么样的方式去调整。虽然这使得约束求解器变得更加通用但同时也会变得效率底下。因此，应该扩展交互约束求解器使其具有特定的约束信息，使其求解更有效。

另外，来自非交互约束的信息可以用于减少参数空间的区域。如果一个特征参数变量在变化范围是 $(0,20)$ ，则可以将参数空间中该参数 $(0,20)$ 范围以外的空间变为无效，这将会减少样本数量。

最后，模型中存在这样一些参数，根据这些参数找到有效模型的可能性很大，所以能够发现这样的参数的算法应该被加强。然而当前的算法中与交互特征有关的参数都被相同对待，然而，这些特征参数中有一些根本与交互特征的位置和方向没有关系，对它们的改变永远不会找到有效的模型。因此应该研究一种能够找到这种高可能性的参数的算法。

总之，基于参数空间的满足交互约束的方法在造型系统中是非常有用的。

第6章 直接操作中的直观性

为增强直接操作过程中的直观性, 本文为实型的特征参数提供了强有力的各种 3D 句柄, 以及为所有的特征操作提供了预先审查的透明体。

6.1 特征句柄的类型

不是所有的特征参数都有一个明确的几何意义。对于这样的参数, 不可能为其赋予一个带有自然行为的特征句柄。一个很好的例子就是特征的体积参数。然而, 大多数的参数都有一个简单的几何意义^[46], 例如, 一个突特征的宽度或者旋转角度。本文只讨论具有实值的参数。对于这些参数定义了四种类型的句柄, 它们满足了大多数直接操作的需求:

在这个方法中, 所有的特征句柄都是通过与其相关的特征参考元素(如参考点和参考线)来定义^[46-47]。在操作这些句柄的时候可以根据这些系统的参考元素知道句柄的位置及其行为。任何一种类型的句柄的定义都需要用点来做参考, 这些点定义了句柄在模型中的位置。这些参考将向特征定义中其他任何的参数一样使用约束来定义。用参考元素来定义的一个重要的因素就是能够通过使用约束求解器来确定这些句柄的位置和方向。根据这些参考元素可以很容易的利用三维图形中的几何变换计算出参数变量。其中标准的变换矩阵 T 的定义如下:

$$T = \begin{pmatrix} a & b & c & :p \\ d & e & f & :q \\ g & h & i & :r \\ \dots & \dots & \dots & \dots \\ l & m & n & :s \end{pmatrix}$$

设变换前的点 $P_1(x_1, y_1, z_1)$, 变换后的点 $P_2(x_2, y_2, z_2)$, 则

$$(x_1 \quad y_1 \quad z_1 \quad 1)T = (x_2 \quad y_2 \quad z_2 \quad 1) \quad (6-1)$$

这四个子矩阵的作用为:

3×3 矩阵使立体产生局部比例和旋转变换;

1×3 矩阵使立体产生平移变换;

3×1 矩阵使立体产生透视变换;

1×1 矩阵使立体产生整体比例变换。

下面对各种句柄所表示的参数的计算都是基于这个标准的变换矩阵。

6.1.1 线性句柄

这种类型的句柄沿着一条直线移动并随着鼠标的移动做出线性的反应。除了表示位置的参考点之外还包括一条句柄沿其移动的参考线。当这个句柄被拖拉时，沿着参考线生成产生出鼠标的新的位置，然后通过计算出的句柄的位置和先前的位置产生参数变量。如图 6-1 所示。

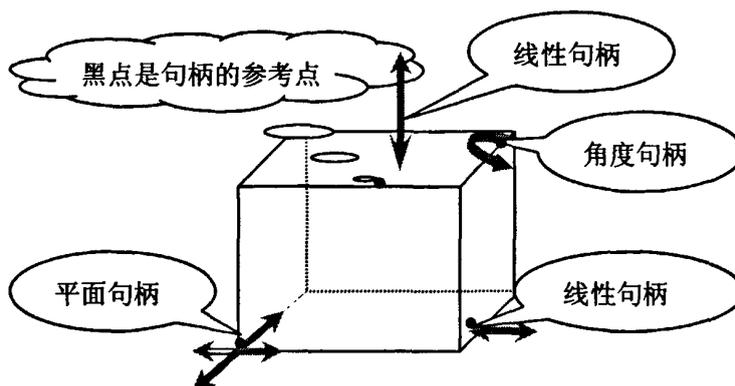


图 6-1 句柄类型

Fig.6-1 Types of handles

对线性句柄的拖拽涉及的是模型的比例变换，其中比例变换包括局部比例变换和整体比例变换。

设沿 x 、 y 、 z 三轴的比例系数分别为 S_x 、 S_y 、 S_z 的局部比例变换，其公式为：

$$(x_2 \quad y_2 \quad z_2 \quad 1) = (x_1 \quad y_1 \quad z_1 \quad 1) \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (6-2)$$

其中局部比例变换矩阵

$$T_s = \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

其中的 S_x 、 S_y 、 S_z 是通过句柄的参考线相对于三个坐标轴的相对位置计算得出。当 $S_x = S_y = S_z$ 时，局部比例变换即可变为整体矩阵变换。

6.1.2 角度句柄

这种类型的句柄沿着一条用户能够定义角度参数的弧线移动。这种句柄除了拥有表示位置的参考点之外还有一个参考线，用来表示角度句柄沿其旋转的旋转轴。当柄被拖拉时，在垂直于旋转轴的平面上产生新的鼠标的位置，这个旋转轴通过句柄的当前位置记为 Point1。为了确定这个句柄对参数的影响还要计算出所旋转的角度。这个角度是指穿过旋转轴的平面上的直线与穿过句柄的起始点和 Point1 的直线之间所夹的角度。见图 6-1。由于参考线是任意的，所以在参考线上可任意取两点 $A(x_a, y_a, z_a)$ 和 $B(x_b, y_b, z_b)$ ，见图 6-2。所求的变换矩阵可以分解成若干个变换的组合(以逆时针旋转为例)：

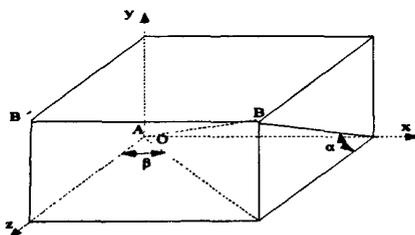


图 6-2 绕空间任意线段 AB 逆时针旋转角 θ 的变换

Fig.6-2 Transform of counterclockwise rotating θ angles surrounding a random segment in space

1. 平移坐标变换 T_1 ，使 A 点移到坐标原点，则

$$T_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_A & -y_A & -z_A & 1 \end{pmatrix}$$

2. 将线段 AB 先绕坐标轴 x 逆时针转，使 AB 落在 xOz 平面上，则变换 T_2 。

$$T_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

式中 α 由下式确定：

$$\cos \alpha = (Z_B - Z_A) / r_2, \quad \sin \alpha = (Y_B - Y_A) / r_2, \quad r_2 = \sqrt{(Y_B - Y_A)^2 + (Z_B - Z_A)^2}.$$

由于绕 x 轴旋转时， x 坐标不变，线段长度 AB 不变，则 z 坐标为 r_2 。

3. 再绕 y 轴顺时针转角 β , 使 AB 与 Oz 轴重合, 则变换 T_3 .

$$T_3 = \begin{pmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

其中, $r = \sqrt{(X_B - X_A)^2 + (Y_B - Y_A)^2 + (Z_B - Z_A)^2}$

4. 此时绕 AB 轴转角 θ , 变成绕 z 轴转角 β 的变换 T_4

$$T_4 = \begin{pmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

然后再将 AB 变换到原来位置的变换, 则有

$$T_5 = T_3^{-1} = \begin{pmatrix} r_2/r & 0 & -(X_B - X_A)/r & 0 \\ 0 & 1 & 0 & 0 \\ -(X_B - X_A)/r & 0 & r_2/r & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_6 = T_2^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & (z_B - z_A)/r_2 & -(y_B - y_A)/r_2 & 0 \\ 0 & (y_B - y_A)/r_2 & (z_B - z_A)/r_2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_7 = T_1^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x_A & y_A & z_A & 1 \end{pmatrix}$$

绕空间任意轴 AB 转角 θ 的变换矩阵为 $T = T_1 \cdot T_2 \cdot T_3 \cdot T_4 \cdot T_3^{-1} \cdot T_2^{-1} \cdot T_1^{-1}$ 。

6.1.3 平面句柄

对平面句柄的拖拽主要涉及到的是模型的移动变换, 它的移动不是沿着直线而是沿着一个平面(即有两条线的自由度)。对于这种类型的柄来说, 需要两条直线来确定一个它所赖以移动的平面。与其他类型的句柄不同的一点是, 通过

操纵它可以同时影响两个参数而不是一个。为了确定参数的变化值，要进行与线性句柄中相似的计算，不同的是相应于每个影响的参数的参考线上都要生成一个鼠标位置。之所以会有这样的句柄是因为它可以使得用户通过定义与外部的两个距离来确定特征的位置。

把空间点 $P(x,y,z)$ 沿 x, y, z 三个方向分别平移 T_x, T_y, T_z 的变换 $(x_1 \ y_1 \ z_1 \ 1)T = (x_2 \ y_2 \ z_2 \ 1)$ ，其中

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{pmatrix}$$

6.1.4 滑行句柄

它也和线性句柄类似，但是，这种情况下，这条线只是一条简单的与视点垂直的线，它与我们通过操作参数将要改变的特征几何没有关系。因此这种句柄与其他任何类型的句柄不同，它只包含一个表示位置的参考点而没有参考线。滑行句柄非常适合操纵那些并没有简单的几何意义或者很难定位的参数，例如一个容器拐角处的半径，如图 6-3 所示。

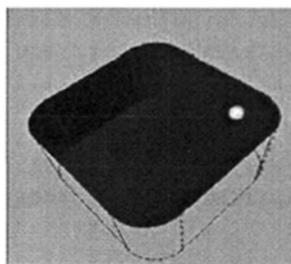


图 6-3 滑行句柄

Fig. 6-3 Slider handle

在特征类的定义中使用这四种类型的句柄，就会使得为了能够改进直接操作特征方法而参数化大量的特征类变得很容易。

6.2 特征句柄的使用

现在我们举例来简要的说明用户是如何用这些句柄来与特征模型进行交互

的。

1. 当用户选择了一个要编辑的特征时, 它将会高亮度显示, 并在外部附加一个透明的且以其他颜色显示的框(我们称作是预先审视的透明体), 与这个特征有关的句柄也会同时显现在屏幕上。然后用户可以通过拖拉这些句柄来对其进行编辑。

2. 当用户进行此操作时所显示的特征会根据参数的变化而不断的更新。这个时候, 所有被这个参数值所影响的特征都将显示为透明的状态。这些变化不是最终的因此可以被撤销。

3. 当用户选择应用模型操作之后, 对模型所做出的改变就会变成最终的操作被提交。

4. 当用户选择撤销模型操作, 则上次应用的修改操作所做出的所有变化都会撤销。

正如第三章所说的那样, 句柄操作的事件中有两个重要的阶段: 选择阶段, 即用户选择一个特征的参数的时候; 交互阶段, 每次各个句柄被拖拉的时候。选择阶段高层的事件流程图如图 6-4 所示。这里用户的选择导致特征所有句柄(即所有显示在特征上的并在 Operator 控制器中注册过的句柄)处于激活状态, 突显的特征是利用第四章提出的分割 CCM 的算法将操作定位之后分割出来的。

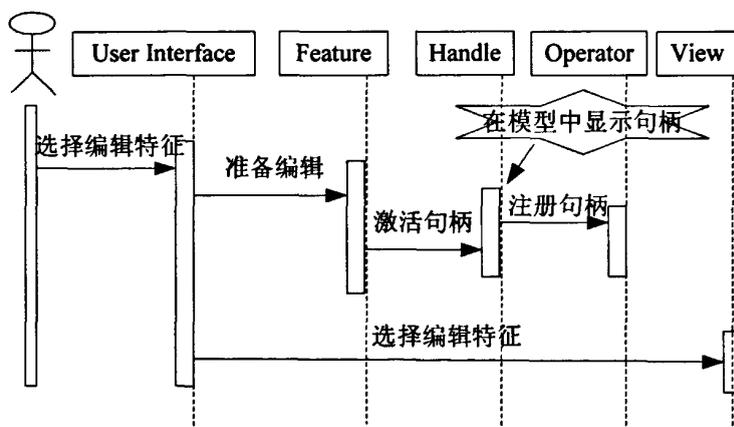


图 6-4 用户选择特征编辑时的交互流程图

Fig. 6-4 Interaction flow chart when choosing and editing the feature

当一个拖拉事件发生时的流程图如图 6-5 所示。拖拉事件以及它所产生的新的鼠标位置被报告给 Operator 控制器。Operator 控制器向 Handle 处理器发送一个消息, 在处理器中计算参数的新的值, 用以更新模型。当然计算时要考虑

到句柄的行为。当设置好新的参数值之后，Operator 控制器就会命令约束求解器来对模型进行求解。这个求解的过程利用了第四章分所介绍的操作局部化的算法以及第五章所介绍的新的交互约束满足的方法。根据约束求解所得到的新的模型值来对预先审查的透明体进行渲染和显示。

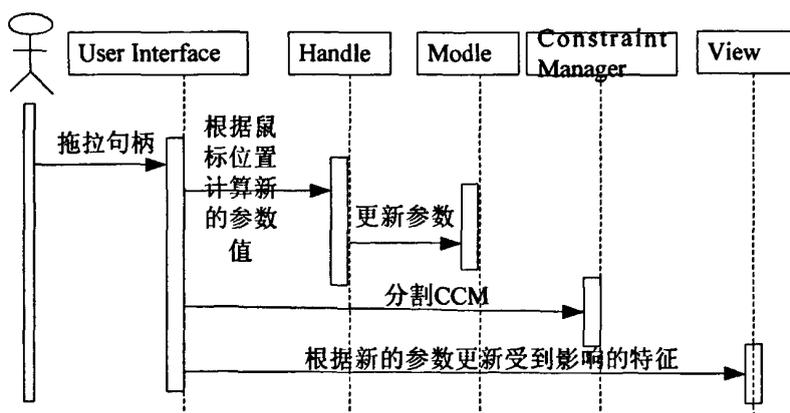


图 6-5 用户拖拉句柄时的流程图

Fig. 6-5 Flow chart of dragging handle

6.3 本章小结

为了使操作结果具有直观性引入了几种简单而具有直观的几何意义的特征句柄。当用户通过特征句柄对模型进行直接操作之后，通过一个透明的预先审查的覆盖体给用户提供了实时有效而且具有深层洞察力的反馈。用这用方法，用户能够通过先前的模型比较，清楚地看到操作之后对模型所产生的影响。

结论

本课题是哈尔滨理工大学 CAD 研究所的国家自然科学基金项目(编号 60173055)“基于细胞元表示的语义特征造型研究及应用”的一部分。该基金项目以哈尔滨理工大学计算机辅助工业造型系统(简称 HUST-CAIDS)为基础,探求更能满足用户需求的造型设计方法。本课题旨在提高对特征模型直接操作时的效率和可视性,提高整个系统的造型能力和造型效率,从而使其更符合工程设计人员的需要,更好地体现设计人员的设计意图。

本文在对 HUST-CAIDS 模型研究的基础上,参考 BIDARRA 等人的研究成果对语义特征造型中的直接操作领域做了深入的研究,主要完成了以下工作:

1. 提出了将对特征的操作局部化的思想,并在此基础上给出将操作所影响的特征区域从整个模型中分离出来的两个具体的算法,该方法由于减少了参与模型重构的数据量,所以大大提高了重构的速度,从而使得在直接操作过程中将参数对模型的影响实时反馈给用户成为可能。

2. 在直接操作某个特征参数的过程中,随着参数的变化可能会使该特征与其它特征产生冲突,本文提出了一种自动求解交互约束的方法,该方法基于模型的参数空间,扩展了当前自动检测交互约束的方法,并提供了自动满足这种约束的能力,也就是自动将模型调整成为一个有效的模型,因此比传统的将交互约束转换为几何约束来调整交互约束的方法有效的多。将所研究的内容在 HUST-CAID 系统上的实验结果证明,该方法提高了对特征操作的直观性,同时也提高了造型能力和造型效率,使造型结果更符合设计人员的需要。

3. 为增强直接操作过程中的直观性,提出了特征句柄的概念,并利用三维图形学中的变换矩阵为实值的特征参数定义了 4 种简单的 3D 句柄,以及为所有的特征操作提供了预先审查的透明体,从而使用户能够通过先前的模型比较,清楚地看到操作之后对模型所产生的影响。

基于细胞元的语义特征造型的直接操作技术是 CAD 发展历程中的一个新的方向,虽然现在还处于初期的研究阶段而且还有很多待解决的问题如:有关底层几何数据如细胞元的信息冗余量大且修改操作复杂,从而导致算法效率不高,以及第五章所说的交互约束求解还有很多地方有待优化,但它却给特征造型技术开辟了一个崭新的领域,并具有重要的理论意义和实用价值因此,需要不断加以完善,不断吸取更新更好的方法来完善造型设计功能。

参考文献

- [1] SUDARSAN RACHURI, FENVES STEVEN J, SRIRAM R D. A Product Information Modeling Framework for Product Lifecycle Management[J]. CAD Computer Aided Design, 2005, 37(13): 1399-1411.
- [2] BRONSVOORT W F, BIDARRA R, NYIRENDA P J. Developments in Feature Modeling[J]. Computer-Aided Design and Applications, 2006, 3(5): 655-664.
- [3] VAN DEN, BERG E, BRONSVOORT W F. Freeform Feature Modeling: Concepts and Prospects[J]. Computers in Industry, 2002, 49(2): 217-233.
- [4] 孙立铸. 计算机图形学[M]. 哈尔滨: 哈尔滨工业大学出版社, 2000: 50-54.
- [5] 俞杰, 耿卫东, 潘云鹤. 面向产品设计的 CAD 原型系统 CMCAD[J]. 计算机应用研究, 1999, 4(3): 96-97.
- [6] BIDARRA R, BRONSVOORT W F. Semantic Feature Modeling[J]. Computer-Aided Design, 2000, 32(3): 201-225.
- [7] THOMPSON A, CYNTHIA LEVY, ROGER MANNING. A Generative Model for Semantic Role Labeling[J]. Lecture Notes in Artificial Intelligence (Subsidies of Lecture Notes in Computer Science), 2003, 2837(6): 397-408.
- [8] 孙东光, 李隆春, 郭连水, CAD 系统语义特征模型的研究与实现[J], 计算机集成制造系统——CIMS, 2002, 8(3): 193-196.
- [9] 叶修梓. 从工业界的角度看 CAD 技术的研究主题与发展方向[J]. 计算机辅助设计与图形学学报, 2003, 10(15): 1194-1199.
- [10] NOORT A, HOEK GFM, BRONSVOORT W F. Integrating Part and Assembly Modeling[J]. Computer-Aided Design, 2002, 34(12): 899-912.
- [11] HOFFMANN C M, JOAN-ARINYO R. CAD and the Product Master Model[J]. Computer-Aided Design, 1998, 30(11): 905-918.
- [12] TER HAAR R, BRONSVOORT W F, DE KRAKER K J. A Semantic Feature Model for Conceptual Design of Warships[C]. Proceedings International Conference on Computer Applications in Ship Building, Portsmouth, UK, 2007, 3(18-20): 79-88.
- [13] BIDARRA R, DE KRAKER K J, BRONSVOORT W F. Representation and Management of Feature Information in a Cellular Model[J]. Computer-Aided Design, 1998, 30(4): 301-313.

- [14] NYIRENDA PJ, BIDARRA R, BRONSVOORT W F. A Semantic Blend Feature Definition[J]. *Computer-Aided Design and Applications*, 2007, 4(1-4): 795-806.
- [15] VAN DER, MEIDEN HA, BRONSVOORT W F. Tracking Topological Changes in Feature Models[C]. *Proceedings-SPM 2007: ACM Symposium on Solid and Physical Modeling*, Beijing, China, 2007: 341-346.
- [16] KIM JUNHWAN, PRATT MICHAEL J, IYER RAJ G. Standardized Data Exchange of CAD Models with Design Intent[J]. *CAD Computer Aided Design*, 2008, 40(7): 760-777.
- [17] 龚雄, 赵建军, 王启富. 边界表达模型的局部几何更新方法[J], *机械科学与技术*, 2007, 26(1): 92-95.
- [18] VAN DER MEIDEN HA, BRONSVOORT W F. A Constructive Approach to Calculate Parameter Ranges for Systems of Geometric Constraints[J]. *Computer-Aided Design*, 2006, 38(4): 275-283.
- [19] 唐荣锡, 汪嘉业, 彭群生, *计算机图形学教程*[M]. 北京: 科学出版社, 1990: 32-38.
- [20] 潘双夏, 张帅, 冯培恩. 基于工程约束的产品参数化建模策略研究[J]. *计算机辅助设计与图形学学报*, 2001, 9(8): 840-845.
- [21] ANDERL R, MENDGEN R. Modeling with Constraints: Theoretical Foundation and Application[J]. *Computer-Aided Design*, 1996, 28(3): 155-168.
- [22] 唐良红, 孙立镛. 基于特征的参数化设计和实体模型编辑的研究[J]. *工程图学学报*, 1999, 2(6): 71-77.
- [23] 王启付, 田显钊, 陈钢. 面向三维参数化特征造型系统的自适应装配技术[J]. *华中科技大学学报*, 2007, 3(35): 80-93.
- [24] KRIPAC J. A Mechanism for Persistently Naming Topological Entities in History-based Parametric Solid Models[J]. *Computer-Aided Design*, 1997, 29(2): 113-112.
- [25] WU J. Topological Naming Mechanism in Feature-based Modeling[J]. *CADDM*, 1997, 7(2): 71-78.
- [26] BIDARRA R, BRONSVOORT W F. Persistent Naming through Persistent Entities[C]. *Proceedings of Geometric Modeling and Processing 2002-Theory and Applications*, Wako, Saitama, Japan, 2002: 233-240.
- [27] 段海波, 马枚. 利用用户定义特征实现基于约束设计[J]. *计算机辅助设计*

- 与图形学报, 2001, 13(3): 193-196.
- [28] BIDARRA R, NYIRENDA P J, BRONSVOORT W F. A Feature-based Solution to the Persistent Naming Problem[J]. *Computer-Aided Design and Applications* 2005, 2(1-4): 517-526.
- [29] 高雪瑶, 孙立铸. 语义特征造型中特征修改优先准则的研究[J]. *计算机工程与应用*, 2007, 43(29): 36-39.
- [30] BIDARRA R, VAN BUNNIK A, BRONSVOORT W F. Direct Manipulation of Feature Models in Web-Based Collaborative Design[C]. *Proceedings of the 2004 ASME Computers and Information in Engineering Conference*, Salt Lake City, UT, USA, 2004: 645-652.
- [31] BALIJEPALLI ARVIND, LEBRUN THOMAS, GAGNON CEDRIC. A Modular System Architecture for Agile Assembly of Nan Components Using Optical Tweezers[C]. *Proceedings of SPIE-The International Society for Optical Engineering*, Prague, Czech Republic, 2005, 5908: 1-10.
- [32] LOURENÇO D, OLIVEIRA P, NOORT A. Constraint Solving for Direct Manipulation of Features[J]. *Journal of Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 2006, 20(4): 369-382.
- [33] WU D, SARMA R. Dynamic Segmentation and Incremental Editing of Boundary Representations in a Collaborative Design Environment[C]. *Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications*, Ann Arbor, Michigan, USA, 2001: 289-300.
- [34] TAI-HSUAN WU, DAVOODI A. PARS: Fast and Near-optimal Grid-based Cell Sizing for Library-based Design[C]. *2008 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Portland, Oregon, USA, 2008: 107-111.
- [35] VAN DER, MEIDEN H A, BRONSVOORT W F. Solving Topological Constraints for Declarative Families of Objects[J]. *Computer-Aided Design*, 2007, 39(8): 652-662.
- [36] PERNG D B, CHANG C F. Resolving Feature Interactions in 3D Part Editing[J]. *Computer-Aided Design*, 1997, 29(10): 687-699.
- [37] HU ZHI-HUA, DING YONGSHENG, ZHANG WENBIN. An Interactive Co-evolutionary CAD System for Garment Pattern Design[J]. *CAD Computer Aided Design*, 2008, 40(12): 1094-1104.

- [38] BIDARRA R, BRONSVOORT W F. History-Independent Boundary Evaluation for Feature Modeling[C]. Proceedings of the 1999 ASME Design Engineering Technical Conferences, Las Vegas, Nevada, USA, 1999, 12(16): 342-345.
- [39] BIDARRA R, MADEIRA J, NEELS W J. Efficiency of Boundary Evaluation for a Cellular Model[J]. CAD Computer Aided Design, 2005, 37(12): 1266-1264.
- [40] NOORT A, BRONSVOORT W F. Enforcing Model Validity by Automatic Adjustment[J]. Journal of Computing and Information Science in Engineering, 2001, 1(4): 311-319.
- [41] GHOSH RANADHIR, GHOSH MOUMITA, YEARWOOD. A Fully Automated CAD System Using Multi-Category Feature Selection with Restricted Recombination[C], Proceedings-6th IEEE/ACIS International Conference on Computer and Information Science, Melbourne, Australia, 2007: 106-111.
- [42] BIDARRA R, TEIXEIRA J C. Intelligent form Feature Interaction Management in a Cellular Modeling Scheme[C]. Proceedings Second Symposium on Solid Modeling and CAD/CAM Applications, Montreal, Canada, 1993: 483-485.
- [43] FISHMAN G S, MONTE CARLO. Concepts Algorithms and Applications[J]. Techno-Metrics, 1997, 39(3): 336-338.
- [44] WOO YOONHWAN, LEE SANG HUN. Volumetric Modification of Solid CAD Models Independent of Design Features[J]. Advances in Engineering Software, 2006, 37(12): 826-835.
- [45] RYU C, SHIN J G, KWON O H. Development of Integrated and Interactive Spatial Planning System of Assembly Blocks in Ship-Building[J]. International Journal of Computer Integrated Manufacturing, 2008, 21(8): 911-922.
- [46] CHEUTET V, CATALANO C E, PERNOT J P. 3D Sketching for Aesthetic Design Using Fully Free-Form Deformation Features[J]. Computers and Graphics, 2005, 29(6): 916-930.
- [47] PALAGIN O, ROMANOV V, SACHENKO A. Virtual Laboratory for Computer-Aided Design: Typical Virtual Laboratory Structure and Principles of its Operation[C]. 2007 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Dortmund, Germany, 2007: 77-81.

攻读硕士学位期间所发表的学术论文

- [1] 李素慧, 孙立镛. 基于细胞元特征模型中操作局部化方法的研究. 哈尔滨理工大学学报, 已录用.
- [2] 孙立镛, 李素慧. 基于特征参数空间的交互约束自动求解方法. 计算机应用研究. 已录用.

致谢

本文的选题、课题研究及撰写工作是在导师孙立铸教授的悉心指导下完成的。在一年半的研究生学习期间以及一年的毕业设计过程中，孙老师对我的工作生活及学习给予了极大的关心与帮助，尤其是在完成毕业设计课题上给了我许多机会去学习和实践，并在许多实际应用问题上传授给我许多宝贵的思维方法和解决问题的有效手段，这为我今后的发展奠定了坚实的理论和实际基础。孙老师严谨的学风和渊博的知识使我受益终身。在此谨向孙老师致以崇高的敬意和衷心的感谢。

在课题的进行中，我也得到了计算机应用技术研究所的老师和同学们的大力帮助，在此向他们表示最诚挚的谢意。