

语义特征造型中对象族模型研究

刘宪国¹ 孙立镛¹ 王其华² 张 辉¹

(哈尔滨理工大学计算机科学与技术学院 哈尔滨 150080)¹(大连水产学院信息工程学院 大连 116023)²

摘 要 提出了一种新的对象族模型,即陈述式对象族模型。定义了陈述式对象族模型,给出了模型的几何结构和拓扑结构,通过约束详细说明了特征的语义,以约束图的形式给出了该模型中特征的表示,给出了模型的语法描述。通过实例验证了模型的有效性,克服了传统的基于历程建模的不足,提高了 CAD 建模设计效率,降低了设计成本。

关键词 对象族,特征造型,几何约束,拓扑约束

Research on Family of Object Model in Semantic Feature Modeling

LIU Xian-guo¹ SUN Li-juan¹ WANG Qi-hua² ZHANG Hui¹

(College of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China)¹

(College of Information Engineering, Dalian Fisheries University, Dalian 116023, China)²

Abstract A new family of object model, i. e. declarative family of object model, was imposed in this paper. The model was defined, geometric and topological structure of this model was presented, semantic of feature was specified by constraints, constraint graph was used to represent features in the model, grammar specification of model was provided. Validation of the model was verified in a instance, and insufficient of traditional procedural-based modeling was overcome, design efficiency of CAD modeling was developed, and design cost was reduced.

Keywords Family of object, Feature modeling, Geometric constraint, Topological constraint

1 概述

在很多领域用到了几何模型,如 CAD、场景建模、医学等领域。在这些领域中需要的几何模型不是静态不变的,而是能够捕捉形状不断变化的模型。单个的 CAD 模型通常看作一个单独的对象,一个特征模型可以表示一个对象族。通过改变模型中特征的参数来维护模型的多样性。每个模型表示不同但相似的对象,这些所有可维护对象的集合就是对象族,集合中的对象就是族成员。

在下面几种情况下用到了对象族 CAD 模型,一种情况是,构造一系列相似的产品,如不同尺寸的工具;另一种情况是重用某模型作为一个大型 CAD 模型的一个部件,为该模型赋适当的值。这样为对象族建模不仅可以提高设计生产效率,而且可以降低设计成本。

在传统的基于历程和边界表示的几何模型中存在 3 个不能解决的问题:一致性命名问题、特征优先问题和维护特征的语义问题。本文在自主研发的 HUST-CAID 系统中引入语义特征模型^[1](Semantic Feature Model, SFM)的概念,以单元模型(Cellular Model, CM)为基本几何元素,提出一种陈述式对象族模型(Declarative Family of Object Model, DFOM)。

2 陈述式模型

相对于传统的基于过程和基于规则的模型,陈述式模型

不是说明模型所有可能的拓扑变化,而是明确地陈述对象的多个属性。陈述式模型由变量和约束组成,其中变量是存在于所有对象中的元素,有不同的属性,而约束是通过施加变量间的某种关系声明不变的属性,利用约束求解器确定满足所有约束的变量的数值。

在几个陈述式场景建模系统中用到了约束^[2,3],在这些系统中,模型由位于三维空间中的几个对象组成,对象间建立位置约束,比如对象 A 必须位于对象 B 的左侧。该建模系统确定了场景中对象的可变的配置,并且将这些配置展示给用户,但是该系统不支持为固定的几何和拓扑属性的对象族建模。

在现有的 CAD 系统中使用几何约束指定草图中的几何关系^[4]。在二维几何元素(如线、圆等)上施加该约束,从而约束其维数度(如长度、半径等)、它们的相对位置和方向(如距离、角度等)以及其它的关系(如相邻、相切等)。

拓扑约束声明了对象族所有成员必须满足的恒定的拓扑属性,确立模型所有可能的拓扑改变^[5]。利用拓扑约束限制形状的拓扑变化范围,而利用几何约束参数化模型的外形。

3 陈述式对象族模型

3.1 提出背景

在现有的 CAD 系统中,不能准确地说明特征的语义,不能适当地维护特征语义的有效性^[6]。使用特征表示设计意

到稿日期:2010-04-25 返修日期:2010-07-05 本文受国家自然科学基金“基于细胞元表示的语义特征造型研究”(60173055)资助。

刘宪国(1981—),男,博士生,主要研究方向为计算机图形学与 CAD, E-mail: qdixg5639@163.com;孙立镛 教授,博士生导师;王其华 硕士,讲师;张 辉 硕士,助教。

图,但是实际的特征的语义和用户的意图总是会产生不一致的情况。这就导致当改变模型的参数时,特征间的意外交互产生不理想的模型,而在现有的系统中很难创建这些对象族或者不可能表示这样的对象族。

现有的 CAD 系统不能恰当地维护所有特征实例或对象族的语义,因为这些系统使用了基于历程的模型,利用了边界表示法。基于历程的模型是过程式模型,不适于说明对象族的恒定属性。边界表示法没有包含确定特征语义的足够的拓扑信息。

3.2 模型定义

任何一种陈述式模型的基本元素都是变量和约束。模型成员的几何和拓扑信息由几何变量和拓扑变量表示。在一个或多个相同类型或不同类型的变量上施加约束,用来指定不同的特征和族的不同属性,如几何属性和拓扑属性分别由几何约束和拓扑约束指定。

陈述式对象族模型(Declarative Family of Objects Model, DFOM)是语义特征模型(Semantic Feature Model, SFM)的扩展。在 DFOM 中,模型的几何和拓扑不必全部指定,一般情况下,一个 DFOM 表示一个对象族(Family of Object, FO)。通过求解拓扑约束确定族成员(Member of Family, MF)。

对象族模型由几何变量和拓扑变量组成^[7],其中几何变量称之为载体,拓扑变量称之为结构体。载体定义了分隔空间的面,比如平面载体定义了一个平面和平面的两个侧面。结构体主要表示点集,比如体、曲面、曲线和单个的点,由载体定义的子空间相交的方式构造。载体和结构体通过子空间约束建立联系。载体上的几何约束详细说明了对象族的几何属性,结构体上的拓扑约束详细说明了对象族的拓扑属性。这种表示方式可以使设计者以陈述的方式详细说明所有的对象族。

3.3 模型的几何结构和拓扑结构

为了表示不同外形的对象族及其几何和拓扑属性,我们使用几何变量(也称为载体)、拓扑变量(也称为结构体)以及它们的关系(子空间约束)来表示这种特征模型。

在很多对象族表示方式中用到了载体,因为载体可以定义在任意维度上,在我们的系统中,我们将载体定义为将三维欧氏空间分割为 3 个子空间的函数,分别标记为 IN , OUT 和 ON 。每个对应于 IN 和 OUT 的子空间都是相互连接的,这些子空间是三维点的集合。标记为 ON 的子空间是分割 IN 和 OUT 子空间的曲面。这 3 个子空间是相互分割的空间,即每个点或者属于 IN 空间,或者属于 OUT 空间,或者属于 ON 空间。这样一个载体可以通过一个三元函数表示,即

$$C: \mathcal{R}^3 \rightarrow \{IN, ON, OUT\}$$

由等式 $g(p)=0$ 描述的代数曲面可以通过载体定义,即

$$C(p)=IN \Leftrightarrow g(p) < 0$$

$$C(p)=ON \Leftrightarrow g(p) = 0$$

$$C(p)=OUT \Leftrightarrow g(p) > 0$$

基于简单代数曲面的载体的例子如图 1 所示。

在 DFOM 中,一个载体是一个几何变量,也就是不必确定载体的几何自身。对基于代数曲面的载体来说,也就意味着定义曲面的代数函数的系数也是变量,通过数值约束为这些变量赋值,通过代数约束为这些变量建立相互关系。根据

载体定义的像平面、球面这样的基元都可以通过几何约束建立相互关系,比如平面载体 P 和一个相切的球面载体 S 之间的约束关系可以这样表述: $tangent(P, S)$ 。

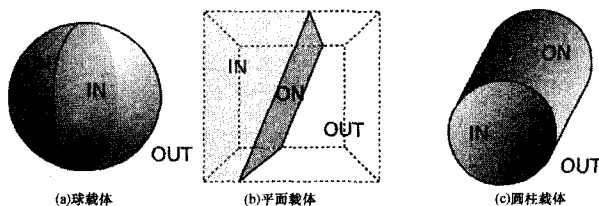


图 1 简单代数载体

对应于 $\mathcal{R}^3$ 的子空间,结构体是表示点集合的变量。子空间约束和拓扑约束确定结构体表示的点集合,子空间约束和拓扑约束确定结构体表示的点集合的具体数值。

子空间约束详细地说明了一个结构体是一个载体的 IN , OUT 和 ON 子空间的一个子集。例如 X 表示一个结构体, C_i 表示一个载体, $S_i = \{p \in \mathcal{R}^3 | C_i(p) = IN\}$ 表示一个子空间,则 $X \subseteq S_i$ 可以用子空间约束 $IN(X, C_i)$ 表示。子空间约束 $ON(X, C_i)$ 和 $OUT(X, C_i)$ 有相似的定义。子空间约束确定了结构体表示的最大范围的点集合。这样,在 n 个子空间 $S_1, S_2, \dots, S_n$ 上, n 个载体 $C_1, C_2, \dots, C_n$ 的 n 个子空间约束限定的结构体 X 表示子空间的交集的一个子集,即 $X \subseteq S_1 \cap S_2 \cap \dots \cap S_n$ 。

在 DFOM 中不用没有任何约束的结构体表示 $\mathcal{R}^3$ 空间。在一个载体上施加 ON 约束的结构体表示一个曲面,在两个载体上施加 ON 约束表示一个或多个曲线,在三个载体上施加 ON 约束表示有限的点集合。在几个载体上施加 IN 或 OUT 约束的结构体表示以这些载体为边界的点集合。例如,在平面载体上施加 IN 约束和在球形载体上施加 IN 约束的结构体表示一个半球体。在平面载体上施加 ON 约束和在球形载体上施加 IN 约束的结构体表示一个圆盘,系统中结构体和载体的例子如图 2 和表 1 所示。

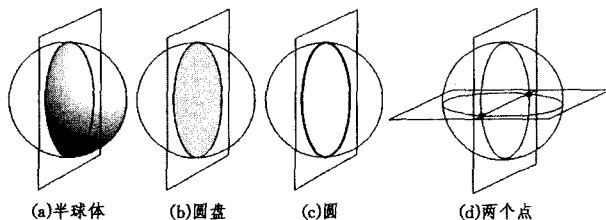


图 2 由球形载体和平面载体构造的结构体

表 1 图 2 中构造结构体约束和载体的关系表

图 2	结构体	球形载体	平面 1	平面 2
(a)	半球体	IN	IN	
(b)	圆盘	IN	ON	
(c)	圆形	ON	ON	
(d)	两个点	ON	ON	ON

3.4 模型的约束图表示

一个 DFOM 可以通过约束图表示,约束图是一个三叉图,节点表示变量或约束,边连接变量节点和约束节点,某一个特定的约束和变量间的边表示在这些变量上的约束。

图 3 是一个约束图的例子,表示一个块对象族。在该图中,矩形表示变量,菱形表示约束。图中没有标出约束间的优先准则。大多数情况下,约束可能是对称的,也可能根据约束

的类型推断约束变量的规则。比如定义几个平面载体模型,底面和顶面通过几何约束相联系,底面和顶面的距离等于实数变量高度的值。一个体结构体通过 6 个面载体上的 IN 约束相关联。

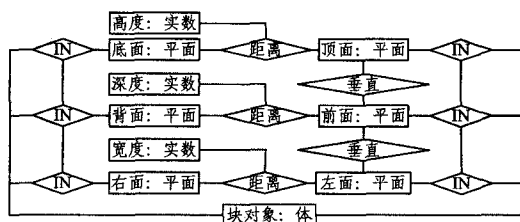


图3 定义块对象族的 DFOM 的约束图

在陈述式模型中,特征类和特征实例没有根本的区别,因为一个陈述式模型可能有多个实现,而模型中的一个特征实例也可能有多个实现。模型中的一个特征可能有同特征类一样的实现集合。

尽管特征本身不能表示一个实体的对象族,但是特征可以看作是一个对象族。比如实体特征块对象经常被作为基底特征使用,所以该特征也是它自身的实体对象族。但是圆柱形暗孔特征就不是一个对象族,因为它通过约束附加在其它特征如块特征上,与其它特征一起组成了一个实体对象族。

图4是一个圆柱凸特征的定义。当在其它 DFOM 中实例化该特征时,系统将该定义拷贝到其它模型中。这样向模型中添加特征时,就将特征定义中的变量和约束添加到目标模型中,而删除特征时将特征定义中的变量和约束删除即可。

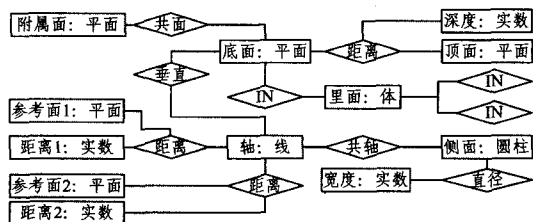


图4 定义圆柱凸特征的 DFOM 约束图

特征应该表示一个形体的形状,也应该包含至少一个表示形体形状的变量。对一些特征体来说,定义几个体的结构体是很有必要的。除此之外,也需要没有体积的结构体,比如表示特征边界的结构体。载体和结构体用来表示规则特征的形状,即没有和其它特征相交的特征的形状。在实现中实际的特征的几何和拓扑取决于施加在特征的载体和结构体上的几何约束和拓扑约束。

一般来说,特征定义中还有其没有包含的一个或多个变量的约束,这些变量和约束的集合称为特征的接口。当实例化特征时,不拷贝这些接口变量,而是用户在目标模型中指定这些接口的使用,即在这些变量上施加特征的接口约束。

利用特征接口详细说明特征间的相互依附关系,在图4所示的圆柱形凸特征中,特征的顶面需要和模型中已经存在的面重合,这样特征就包含共面约束。当在模型中实例化这样的特征时,用户必须指定利用哪个面作为附属面,哪个面作为依附面。在该例中,特征接口也指定了两个位置面,这两个面约束该凸特征的轴的位置。

DFOM 是变量和约束系统,也是表示特征的变量和约束的离散子集合的集成。多个特征组成的 DFOM 表示如图5所示。可以将约束和变量分类为模型约束和变量、特征约束

和变量以及接口约束和变量。模型约束和变量不在任何一个特征中,特征约束和变量是特征的一部分。特征之外的有变量的特征约束是接口约束,这样的变量是接口变量。

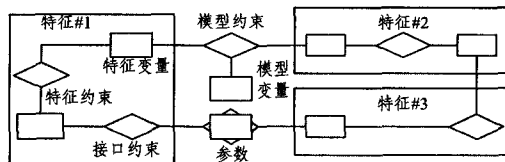


图5 包含多个特征的 DFOM

图5中用菱形和矩形组合表示的参数是变量和数值约束的传递,用户可以引入新的参数,比如实例化特征时可以代替接口变量或者通过向数值约束上添加变量时创建参数。参数值只能由用户改变,建模系统无权修改。

模型中定义特征的子集必须是离散的,即没有变量或者约束属于多个特征,这就保证了在模型中删除特征时的安全性,对其它特征的语义不会产生影响。但是一个模型中也包含不属于任何特征的变量或者约束,比如模型变量和约束,这种情况下允许用户添加构造几何体,如基准面,允许在这些特征中添加约束。

可以通过载体、结构体、几何和拓扑约束,利用简单的模型定义特征的新类型,然后指定哪些变量和约束包含在新特征中。也可以在模型中编辑特征,或者重用模型的部分看作特征使用。

一个 DFOM 的表示可以概括为如下语法:

模型=变量集+约束集+特征集

特征=变量集+约束集

变量=载体|结构体|实数(等)

约束=几何约束|子空间约束|拓扑约束(等)

载体=平面|球体|圆柱体

结构体=体|曲面|曲线|点

几何=距离|角度|重合(等)

子空间=IN|OUT|ON

拓扑=材料|边界|交互(等)

4 模型实现和建模实例

一个 DFOM 定义了一个实现的一个集合,这里的实现对应于模型中满足约束的所有对象,通过单元模型(Cellular Model, CM)作为实现的几何表示。一个 DFOM 含有零个或多个族成员,可以根据 DFOM 定义族成员。一个 DFOM 通过 F 表示,该模型的族成员通过 M 表示,族成员的定义如下:当且仅当

- (1) M 有同 F 一样的变量集合;
- (2) M 有一个 F 的约束的扩展集合;
- (3) M 只有一个实现。

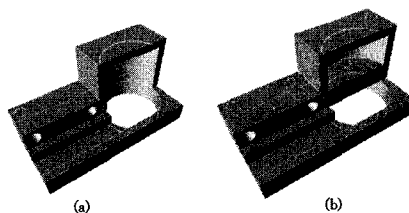


图6 DFOM 实例

在自主研发的 HUST-CAID 系统中,创建了 DFOM 表示的两个实现,如图 6 所示。用户可以自由选择任意一个实现,也可以增加约束来降低实现的个数。如果向暗孔中增加约束,使得该暗孔没有阻隔,那么用户只能看到图 6(a)中的实现,因为实现图 6(b)中的暗孔中间是阻隔的。

结束语 本文提出了一种新的对象族模型——陈述式对象族模型,定义了该模型的几何结构和拓扑结构,给出了简单模型的约束图,在系统中实现了几个简单的模型。该几何模型可以维护特征的语义,克服了传统几何模型的基于设计过程的一些缺点。但该模型很不完善,后期工作主要是对该模型的几何约束求解和拓扑求解,以及对支持对象族建模的工具进行研究。

参 考 文 献

- [1] Bidarra R, Bronsvoort W F. Semantic feature modeling[J]. Computer-Aided Design, 2000, 32(3): 201-225
- [2] Bonnefoi P F, Plemenos D, Ruchard W. Declarative modeling in

computer graphics; current results and future issues[C]// Proceedings ICCS 2004, International Conference on Computational Science. Poland: Poland, 2004, 3039: 80-89

- [3] Gaildrat V. Declarative modelling of virtual environments, overview, issues and applications[C]// Proceedings 3IA'2007, International Conference on Computer Graphics and Artificial Intelligence. Greece, 2007, 10
- [4] Hoffmann C M. Constraint-based computer-aided design [J]. Journal of Computing and Information Science, 2005, 5(3): 128-187
- [5] 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
- [6] van der Meiden H A, Bronsvoort W F. Modeling Families of Objects; Review and Research Directions[J]. Computer-Aided Design, 2009, 6(3): 291-306
- [7] 龚雄. 陈述式几何约束系统的原理与方法研究[D]. 武汉: 华中科技大学, 2007

(上接第 265 页)

平均分为两部分和一部分。

3. 程序运行的结果得到的 4 个非链型的子代数和 2 个链型的子代数与参考文献[15]中子代数的个数是相一致的。

4. 我们现在已经得到了乘积的格蕴涵代数数的具体形式,由滤子、LI-理想的结构性质,我们可以通过定理 6 得到乘积格蕴涵代数数的滤子和 LI-理想的具体形式。

结束语 本文进一步研究了格蕴涵代数中各种滤子间的关系,得到的几种滤子是相互等价的。讨论了乘积格蕴涵代数子结构的性质,得到了乘积格蕴涵代数的滤子只存在平凡滤子和素滤子的结论。通过使用 MATLAB 软件编程,得到了两个有限链型格蕴涵代数乘积的子代数、滤子、LI-理想的具体形式,并且通过一个例子,展示了程序的运行结果。格蕴涵代数是一类重要的逻辑代数,与其它逻辑代数有密切的关系,如何通过它们的子结构挖掘出它们之间的关系,是我们下一阶段的主要研究工作。

参 考 文 献

- [1] Xu Yang, Qin Keyun. Lattice properties of lattice implication algebra[C]// Symposium on Applied Mathematics. Chengdu Science and Technology University Press, 1992: 54-58
- [2] 徐扬. 格蕴涵代数[J]. 西南交通大学学报, 1993, 28(1): 20-27
- [3] Song Zhenming, Wang Jianping. Lattice implication homomorphism in implication filter spaces[J]. Chinese Quarterly Journal of Mathematics, 1998, 13(4): 17-23
- [4] Jun Y B, Roh E H, Xu Y. LI-ideals in lattice implication algebras [J]. Bull Korean Math. Soc, 1998, 35(1): 13-24
- [5] Jun Y B, Xu Yang, Qin Keyun. Positive implicative and associative filters of lattice implication algebras[J]. Bull Korean Math. Soc, 1998, 35(1): 53-61

- [6] 刘军. 基于格蕴涵代数的格值逻辑系统及格值归结原理的研究 [D]. 成都: 西南交通大学, 1998
- [7] 秦克云, 徐扬. 格蕴涵代数的超滤[J]. 西南交通大学学报, 1999, 34(1): 51-54
- [8] 刘卫国, 陈昭平, 张颖. MATLAB 程序设计与应用[M]. 北京: 北京高等教育出版社, 2002
- [9] Xu Yang, Ruan D, Qin Keyun, et al. Lattice-Valued Logic[M]. Berlin: Springer-Verlag, 2003
- [10] Liu Yong-lin, Liu San-yang, Xu Y, et al. LI-ideals and prime LI-ideals in lattice implication algebras[J]. Information Sciences, 2003, 155: 157-175
- [11] 朱华. 格蕴涵代数中滤子和理想的研究[D]. 成都: 西南交通大学, 2005
- [12] Ma Jun, Xu Yang. On structure and complementary elements of lattice implication algebra[J]. Fuzzy Mathematics and Systems, 2005, 19(1): 49-56
- [13] Xu Yang, Chen Shuwei, Ma Jun. Linguistic truth-valued lattice implication algebra and its properties[C]// IMACS Multiconference on CESA. Beijing, October 2006
- [14] Jun Y B, Yang Xu, Ma Jun. Redefined fuzzy implicative filters [J]. Information Sciences, 2007, 177(6): 1422-1429
- [15] Xu Yang, Ma Jun, Lai Jiajun. Sub-algebras of finite lattice implication algebra[C]// Theoretical Advances and Applications of Fuzzy Logic and Soft Computing, ASC 42. 2007: 813-821
- [16] Lai Jiajun, Xu Yang. On WLI-ideal space and the properties of WLI-ideals in lattice implication algebra[C]// Korean Society for Computational and Applied Mathematics 2008, J Appl Math Comput. 2009, 31: 113-127
- [17] Xu Yang, Qin Keyun. On filters of lattice implication algebras [J]. The Journal of Fuzzy Mathematics, 1993, 1(2): 251-260