

协同语义特征建模技术研究

金瑛浩^{1,2} 孙立镭¹

(哈尔滨理工大学计算机科学与技术学院 哈尔滨 150080)¹ (通化师范学院学生工作处 通化 134002)²

摘 要 为了提高协同语义建模系统的性能,提出了一种新的协同设计方法,亦即使用特征语义表示法和细胞元模型来表示数据和管理模型中的各种数据和信息,然后通过语义匹配的方法来确定子模型间的逻辑和装配关系,通过语义区域划分、“最值空间”等技术来检测和消解协同设计中发生的操作冲突,最后通过创建临时模型来降低全局约束求解的复杂度。这种方法不仅能够完全实现协同语义特征建模的需要,还可以大大提高协同设计的效率。实验表明,该算法具有更强的适应性和实用性。

关键词 CAD,语义特征建模,特征语义,细胞元模型,协同建模

中图法分类号 TP391 **文献标识码** A

Research on Collaborative Semantic Feature Modeling System

JIN Ying-hao^{1,2} SUN Li-juan¹

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

(Students' Affairs Division, Tonghua Normal University, Tonghua 134002, China)²

Abstract To improve the performance of the collaborative semantic feature modeling system, a new method was presented. It represents and manages all the information and data in features with the feature semantic representation and the cellular model, determines the logic and assembled relations between sub-models by matching semantic, detects and solves operation conflicts between client systems by using semantic area dividing and max-value space, and reduces the complexity of constraint solving of whole system by creating the temporarily model. This method can not only achieve all functions but also increase the design efficiency of the collaborative semantic feature modeling system greatly. Experiments on computer show that this new method is more adaptable and practicable.

Keywords CAD, Semantic feature modeling, Feature semantic, Cellular model, Collaborative modeling

1 引言

在协同语义特征建模系统中,一个基本思想就是在产品模型中将特征的形状信息与功能信息有机地结合在一起,共同构成特征的语义,其最终目的就是维护特征模型的有效性^[1-3]。要保证最终的产品模型有效,就必须跟踪用户的每一步操作,并通过约束求解来保证该操作不仅对本地客户端中的子模型有效,对其他客户端中的子模型也同样有效^[4-6]。传统的协同语义建模系统不仅要将本地的所有操作和模型数据传递给所有其他的客户端,还要在本地验证其他客户端的操作和数据,因而效率十分低下。尤其在发生冲突时,冲突消解的算法由于过于复杂而成为提升系统语义特征建模系统性能的瓶颈。

本文结合 HUST-CAID^[7,8] 系统讨论了一种基于细胞元模型^[9,10]的协同语义特征建模方法。HUST-CAID 是由哈尔滨理工大学 CAD 研究所自主研发的、面向机械产品的外观造型设计系统。该造型系统具有如下几方面的特点:在理论上概念定义清晰,逻辑简明扼要,容易被人接受;在具体实现时

不需要引入过多的人为约定和判断,操作简单易懂;将几何处理、拓扑处理以及各种信息处理有机结合在一起,大大提高了系统执行的效率。本文提出的协同建模方法是在 HUST-CAID 系统的基础上改进了原有的细胞元模型,采用特征语义表示法来表示子系统模型数据,并通过语义匹配实现协同建模。这种方法不仅可以大大地减少设计过程中需要传递的数据,还可以提高整个系统验证用户操作有效性的效率和系统的整体性能。

2 相关概念

2.1 特征语义表示法

特征语义表示法是专门针对传统的表示法在语义特征建模系统中出现的问题而提出的表示方法。它不仅在降低存储特征几何信息和拓扑信息复杂度、提高工程信息和功能信息表达能力等方面表现突出,在协同设计、智能设计和多视图等方面也有很广泛的应用。特征语义表示法将产品模型看作是一个由特征依赖图(FDG)组织起来的各种特征的集合。而特征是由一个或多个“表面”组成的封闭实体(这些“表面”又称

到稿日期:2011-03-08 返修日期:2011-05-17 本文受国家自然科学基金(60173055)资助。

金瑛浩(1978—),男,博士生,讲师,主要研究方向为计算机图形学与 CAD、语义特征造型、人工智能, E-mail: cnxiaobai@163.com; 孙立镭(1944—),男,教授,博士生导师,主要研究方向为计算机图形学与 CAD、语义特征造型。

为语义面)。在特征语义表示法中,特征用一个四元组 $F(P, S, A, R)$ 来表示。其中 P 是为特征的各个表面提供建模依据的参数的集合,也可看作三维坐标系统中若干个点的集合; S 是特征 F 的语义,它决定了 P 中各点的使用方法以及特征实体各个表面的构建方法,是由一系列方法和参数序列组成的。语义 S 表明了构造特征 F 的 n 个面所使用的方法,以及这些方法所使用的参数序列(一个参数可能被多个方法所使用)。 A 是特征的属性集合,用来表示特征在产品模型中所具有的各种性质(如增属性或减属性等)。 R 则是特征的约束集合,用以确定特征在建模过程中所能接受的各种方法和操作等。这一点与传统的表示不同,传统的表示法并没有对特征的受动性进行约束,因而要等用户操作结束后,通过对产品模型的所有特征进行约束求解才能确定用户操作的有效性。而特征语义表示法使用约束集合 R 对所有特征的受动性进行约束,在设计者对某一个或多个特征进行某个操作前就可以确定其有效性,从而节约了大量有效性维护的计算时间,大大提高了系统的性能。

2.2 细胞元模型

“细胞元”模型是语义特征造型系统中一种比较重要的高级数据模型。它主要分3层结构:数据模型层、几何模型层和视图模型层。数据模型层处于“细胞元”模型的最低层,用于保存所有特征的信息以及特征之间的相互联系。几何模型层处于“细胞元”模型的中间层,是一种过渡性的几何模型,主要功能是全面完整地表达特征的所有属性。视图模型层则是“细胞元”模型的最上层,可以为设计者提供不同阶段所需要的各种视图。视图是根据用户的需要(如面向生产或面向加工等),从“细胞元”几何模型映射出来的模型。它面向终端用户,最接近真实效果。这种分层结构不仅可以保留语义特征造型系统的所有优点,还可以通过特征依赖图(FDG)使得特征修改不依赖于历程树,因而在造型过程中能更有效地维护特征的语义(特征模型有效性维护)。

从本质上看,“细胞元”模型是将实体描述为相互邻接或离散的“细胞元”集合。而“细胞元”又是一个关于点集合的描述,这个点集包含于模型各个面的所有形状特征中。因此,每个形状特征都是由一组相关的“细胞元”的子集所表示的。各子集之间的关系由各自的“所属列表”来维护。这种结构十分有利于管理特征元素,因此受到国内外众多学者的重视。

3 协同建模

3.1 数据传输与语义匹配

采用特征语义表示法表示模型,一个突出的优点就是各客户端只要向其它客户端传输本地零件的语义信息即可,而无需传输大量的几何信息、拓扑信息等数据。因此在 HUST-CAID 系统提供了用于显示其它客户端设计情况的窗口——同步窗口。同步窗口显示的模型信息并不是时间驱动的,而是一种特殊的驱动方式——有效操作驱动的,即同步窗口只有在其所对应的客户端中设计者完成一个合法操作后才会触发更新显示内容的事件。也就是说,每个客户端都要先对本地设计者的操作进行有效性验证。如果操作有效,则重新构建本地零件模型的语义,并将新生成的语义发送给所有其它客户端,其他客户端将根据接收到的新语义更新相应的同步窗口;如果操作无效,则撤销本次操作,将本地的零件模型恢

复到上一次的有效状态,本地客户端不向其他客户端发送模型语义,其它客户端也无需更新自己的同步窗口。由此可见,在这种驱动机制下,不需要一个独立的服务器来验证所有的操作,可以大大地提高有效性验证的效率。

然而在验证完本地模型合法后,另一个重要问题就是有效的零件模型未必能组成一个有效的产品模型。因此,单靠客户端本地的有效性验证无法保证最终产品有效。然而在各个功能相对独立、结构完整、语义有效的子模型组合成更高级模型或最终模型时,如果新的语义被改变,那么必然是在某两个或多个能够匹配的语义接口附近发生改变的。因为只有在这类地方的特征才可能与其他子模型的特征发生交互。而特征交互又是特征语义改变的必要条件,因此只需要对这些语义接口中的特征进行约束求解即可。为了提高协同建模的效率,尽早发现并排除子模型之间的非法操作, HUST-CAID 系统中采用了语义匹配的方法来解决这一问题。语义匹配的目的是通过各子模型之间的语义来确定子模型之间的逻辑关系和装配关系,并以此来确定可能发生在子模型间交互的特征的集合。当系统中的某一个同步窗口完成更新后,系统就将该窗口所对应的子模型的语义与本地子模型的语义进行匹配。系统首先在本地模型中选择一个最小的语义,并与同步窗口的语义进行匹配。如果语义匹配,则记录该语义到集合 P 中,然后选择下一个语义,直到本地模型中所有的语义都进行过匹配计算;如果 $P=\emptyset$,则说明目前任意两个子模型之间还不会发生语义及特征间的交互;如果 $P\neq\emptyset$,则说明有一个或多个语义匹配,有两个或多个子模型间可能发生交互。 P 中的语义根据关联性可以分为两种:孤立语义和关联语义。孤立语义说明其本身即为一个语义接口,而关联语义说明这些语义所组成的高层语义才是一个真正的语义接口。语义匹配的算法十分简单,只要表示特征的4个要素中,创建语义的方法相同、参数相同或极其相近、正负属性相反即可。如图1所示,(a)中的凸特征 $t1$ 和(b)中的通孔特征 $th1$ 满足上述匹配方法,因此可以认为这两个特征是语义匹配的((c)为匹配后的装配情况)。而通孔特征 $th2$ 虽然创建语义的方法与(a)中的 $t1$ 相同,且正负属性相反,但是其参数与 $t1$ 不同且相差较大,因此不属于匹配语义。

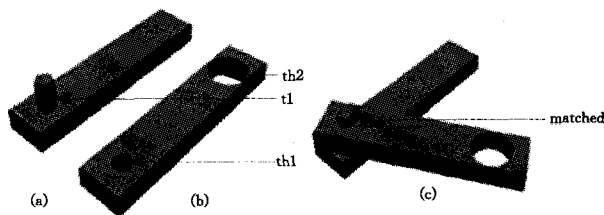


图1 语义匹配

当系统检测到语义接口后,系统同时向本地设计者和同步窗口对应的设计者发出提示并请求确认。用户间通过协商来确定其是否为最终的语义接口。确定后系统将 P 中的语义进行特别标注,以不同的方式进行显示。当设计者对该部分语义进行修改时,就会通知语义接口另外一部分的设计者。当两者都确认修改时,修改才能生效。

3.2 约束求解与冲突消除

在协同设计过程中,实时地进行特征语义匹配检测,就是为了建立各个客户端中零件间的逻辑关系和装配关系,尽早发现并排除客户端之间的操作冲突,减少修改次数,提高协同

设计的效率。子模型的有效性已经由本地的客户端通过约束求解得到了保证,而由存在匹配关系的子模型所组成的更高级的模型是否有效则需要通过另外的约束求解才能确定。特征语义的变化是由特征的子语义改变引起的,而特征语义表示法所表示的特征已经建立了子语义和语义面之间的联系,即通过语义面的变化可以确定子语义的变化情况。而子语义改变的前提条件是不同特征的语义面间发生边界叠加。细胞元模型不仅负责确定各个子语义的精确边界范围,还不断地检测边界叠加事件的发生。

因此为了提高约束求解的效率,当语义匹配关系确立后,就按照所匹配的子模型语义在本地模型中建立相应的最值空间。最值空间是细胞元模型中对特征及其语义面建立的一个正交的空间,它是一个最小的、完全包含了特征或语义面所含几何面的空间集合。最值空间的叠加是特征或子语义面发生改变的必要条件,即语义发生变化时最值空间必然叠加,而最值空间叠加时语义未必改变。因此,当两个或多个子模型存在子语义匹配时,可以先通过细胞元模型提供的检测功能找出发生空间叠加的最值空间,然后将包含在这些最值空间内的特征组成一个新的临时模型,最后通过这个临时模型来判断整个模型的语义是否有效。假设整个协同设计系统中有 n 个客户端,每个模型中的最值空间所组成的集合为 $S = \{S_i | 1 \leq i \leq n\}$,所有的特征集合为 $F = \{F_{ij} | 1 \leq i \leq n, j \in N\}$,发生叠加的最值空间分别为 S_i 和 F_i ,因为不同的客户端设计的模型不可能完全相同,因必然存在 $S_i \subset S$ 且 $F_i \subset F$ 。因此由 F_i 组成的临时模型必然要比上一级模型小。由于这个临时模型包含了所有可能导致语义改变的特征,因此完全可以通过它的有效性来判断上一级模型的有效性,即如果这个模型有效,则说明匹配后的上一级模型也是有效的,否则上一级模型是无效的。这时要根据操作的时间关系,选择最后一个操作,并撤销其产生的所有改变,直到新模型有效为止。这样,通过新模型的约束求解就可以代替全局约束求解,且可以立刻确定并撤销引发冲突的操作,因此可以大大提高约束求解和冲突消解的效率。

3.3 临时模型的创建与数据管理

临时模型作为发生叠加的最值空间中的特征集合,确定了可能发生语义改变或操作冲突的空间范围。虽然这个临时模型的功能和语义甚至外型都不具备实际意义,但是作为验证上一级模型是否有效、操作是否可以继续的中间件,对整个协同语义建模系统的设计效率有着十分重要的影响。为了保证临时模型的创建效率,在协同语义建模系统中,由首先检测到语义匹配的本地系统向其他匹配的客户端发送一个等待验证消息,然后在本地系统中创建一个空的细胞元模型。其所属列表长度与 $|F_i|$ 大小相等,并且将类型定义为“const FEA-TURE *”。因为临时模型只在语义匹配时使用一次,其所包含的特征也已经存在于模型中,因此不需要另外复制这些特征,验证后也不需要删除这些特征。所以只要将 F_i 中元素的指针分别插入到这个新模型的所属列表中,然后将这个临时模型的指针传递给约束求解器,并等待返回结果即可(见图2)。如果临时模型无效,则撤销语义匹配子模型中所有操作中最近的一次操作,直到临时模型有效为止。如果临时模型有效,则对可以匹配的子模型所在的客户端进行一次语义区域划分。

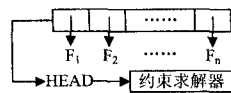


图2 临时模型的所属列表

语义区域划分的目的,就是根据匹配客户端现有的子模型以及匹配关系来确定本地子模型设计的安全范围。在安全范围内的所有特征肯定不会与其他客户端的子特征发生语义冲突,因此这些特征只要在本地图经过有效性验证即可,不需要与其它子模型进行共同验证。因此,语义区域的划分可以进一步提高协同建模的效率。语义空间划分的思想是将本地的所有空间作为全集 I ,将其他匹配的子模型的最值空间作为子集 $L = \{L_i | 0 \leq i \leq n\}$,那么本地模型的有效语义区域 $K = I - L$ 。只要在本地图模型进行的操作,其影响的特征的最值空间包含于 K ,那么这个操作就不需要与任何子模型进行共同验证,只要在本地图验证即可。

在 HUST-CAID 中,本地系统首先将从同步窗口中取得匹配模型的语义,然后按照匹配关系在本地的“细胞元”模型中建立一组空的细胞。这些细胞不具有任何语义和任何信息,也不带有任何元素,只有与匹配模型相同的“最值空间”和客户端 ID。当本地设计者在创建或编辑某一特征时,该特征的“最值空间”如果没有与具有非本地 ID 的“最值空间”叠加,则认为该操作是在有效语义区域的。相反,如果该特征与具有非本地 ID 的“最值空间”发生叠加,则认为是在有效语义区域外的操作,必须与其他子模型进行共同验证。可见,通过语义区域的划分可以减少共同验证的次数,提高冲突消解的效率。

4 试验结果与分析

图3是一个由4个客户端组成的协同语义特征建模系统设计的模型。这些客户端分别在本地系统中(User1, User2, User3 和 User4)设计了模型的4个部分((a), (b), (c) 和 (d))。从实例中可以看到,通过语义匹配可以精确地建立各个客户端中模型间的逻辑关系和装配关系。由于传统的协同设计在设计过程中很难确定子模型间的关系,因此往往会导致两种情况:一是在子模型设计到某一阶段后才发现之前的某个操作与其它子模型发生冲突,使得之后的很多工作付诸东流;二是需要每个客户端都保留所有其它客户端中的子模型数据,并将其作为本地子模型的一部分共同参与有效性验证,这使得约束求解的效率十分低下。而本文所采用的策略通过传递语义的方法改变了必须通过传输整个模型才能约束求解的验证方法,使得协同语义特征建模冲突检测和消解的效率大大提高。

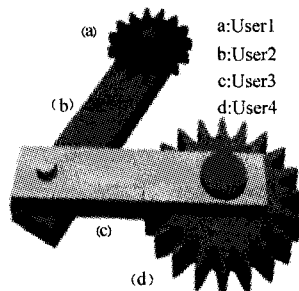


图3 协同设计的模型实例

- [4] Ding W, Wu F, Wu X, et al. Adaptive directional lifting-based wavelet transform for image coding[J]. IEEE Trans. Image Process, 2007, 16(2): 416-427
- [5] Chang C, Girod B. Direction-adaptive discrete wavelet transform for image compression[J]. IEEE Trans. Image Process, 2007, 16(5): 1289-1302
- [6] Pennec E L, Mallat S. Sparse geometric image representations with bandelets[J]. IEEE Trans. Image Process, 2005, 14(4): 423-438
- [7] Velisavljevic V, Beferull-Lozano B, Vetterli M. Space-frequency quantization for image compression with directionlets[J]. IEEE Trans. Image Process, 2007, 16(7): 1761-1773
- [8] Shukla R, Dragotti P L, Do M N, et al. Rate-distortion optimized tree-structured compression algorithms for piecewise polynomial images[J]. IEEE Trans. Image Process, 2005, 14(3): 343-359
- [9] Alani D, Averbuch A, Dekel S. Image coding with geometric wavelets[J]. IEEE Trans. Image Process, 2007, 16(1): 69-77
- [10] Wakin M, Romberg J, Choi H, et al. Wavelet-domain approximation and compression of piecewise smooth images[J]. IEEE Trans. Image Process, 2006, 15(5): 1071-1087
- [11] Mertins A. Image compression via edge-based wavelet transform[J]. Optical Engineering, 1999, 38(6): 991-1000
- [12] Schilling D, Cosman P C. Preserving step edges in low bit rate progressive image compression[J]. IEEE Trans. Image Process, 2003, 12(12): 1473-1484
- [13] Daubechies I, Sweldens W. Factoring wavelet transforms into lifting steps[J]. J. Fourier Anal. Appl., 1998, 4(3): 247-269
- [14] Canny J. A computational approach to edge detection[J]. IEEE Trans. Pattern Anal. Mach. Intell., 1986, 8(6): 679-698
- [15] Li S, Li W. Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding[J]. IEEE Trans. Circuits Syst. Video Technol., 2000, 10(5): 725-743
- [16] Rabbani M, Joshi R. An overview of the JPEG2000 still image compression standard[J]. Signal Process.: Image Process., 2002, 17(1): 3-48
- [17] Taubman D. High performance scalable image compression with EBCOT[J]. IEEE Trans. Image Process., 2000, 9(7): 1158-1170
- [18] Adams M D, Jasper[EB/OL]. <http://www.ece.uvic.ca/~md-adams/jasper/>, JasPer version 1, 900. 1
- [19] Adams M D, Kossentini F. Reversible integer-to-integer wavelet transforms for image compression: performance evaluation and analysis[J]. IEEE Trans. Image Process, 2000, 9(6): 1010-1024
- [20] Zhu J, Wang N. Image quality assessment by visual gradient similarity[J]. IEEE Trans. Image Process(submitted)

(上接第 282 页)

表 1 列出了采用本文方法前后, 协同语义特征建模系统调用全局约束求解器次数的比较情况。其中“总特征数”是指最终完成的模型中的特征数量(即各个客户端中特征数量之和)。“使用前”和“使用后”是指在完成模型后, 各个客户端调用约束求解器次数之和与模型总特征数的比值。比值越小, 说明在设计相同规模的模型时约束求解器被调用的次数越少, 系统性能也会越高。由于本文提出的算法采用语义区域划分的方法, 将很多操作的有效性验证限定在本地系统中, 并只有在本地系统验证有效且确定它对其它客户端中的子模型有影响时才需要共同验证, 因此这种方法对全局约束求解的调用次数明显减少。

表 1 约束求解调用(平均)次数比较

总特征数	使用前	使用后	效果
1~20	0.55	0.23	好
20~50	0.71	0.29	优
50~100	0.83	0.33	优
100 以上	0.92	0.36	优

另外从表 1 中可以看到, 由于传统的“细胞元”模型在进行特征交互检测时需要把所有的特征之间的关系进行约束求解计算, 即需要进行 C_n^2 (n 为模型中的特征数量) 次求解, 当模型中特征数量增加时需要进行的约束求解次数就会大幅度增长, 因此对于复杂模型来说, 传统“细胞元”模型的性能就会受到限制。而本文算法中提出的改进的“细胞元”模型, 在传统“细胞元”模型的基础上构建了特征的“最值空间”, 使得特征交互检测被限定在“最值空间”相交的特征之间进行。这样就大大减少了特征交互检测的实际计算量, 因而可以大大提高系统的整体性能。

结束语 冲突检测和消解是语义特征建模系统性能的瓶颈之一。采用任务角色和权限等方法的传统解决方案不仅机制复杂, 需要大量的计算时间, 还受网络质量、模型复杂度以

及约束求解效率等因素的影响, 因此很难满足和适应当前国际市场对建模系统性能的要求。而本文介绍的方法不仅机制简单, 数据传输量少, 还可以进行快速的冲突检测和消解, 因此具有广泛的应用前景。

参 考 文 献

- [1] Bidarra R, Bronsvort W F. Semantic feature modeling[J]. Computer-Aided Design, 2000, 32(3): 201-225
- [2] Bronsvort W F, Bidarra R, Nyirenda P J. Developments in feature modeling[J]. Computer-Aided Design and Applications, 2006, 3(5): 655-664
- [3] De Lucia P A, Fasano F, Scanniello P G, et al. Enhancing collaborative synchronous UML modeling with fine-grained versioning of software artifacts[J]. Journal of Visual Languages and Computing, 2007, 18(5): 492-503
- [4] Anderl R, Mendgen R. Modelling with constraints: theoretical foundation and application[J]. Computer-Aided Design, 1996, 28(3): 155-168
- [5] Lee J Y, Kim K. A 2-D geometric constraint solver using DOF-based graph reduction[J]. Computer-Aided Design, 1998, 30(11): 883-896
- [6] Ge Jian-xin, Chou Shang-ching. Geometric constraint satisfaction using optimization methods[J]. Computer-Aided Design, 1999, 31(14): 867-879
- [7] 金瑛浩, 孙立铸. 基于特征语义的模型表示法研究[J]. 计算机科学, 2011, 38(1): 286-289
- [8] 金瑛浩, 孙立铸. 语义特征建模系统中参数范围计算的研究[J]. 计算机工程与应用, 2011, 47(5): 13-16
- [9] Bidarrat R, de Kraker K J, Bronsvort W F. Representation and management of feature information in a cellular model[J]. Computer-Aided Design, 1998, 30(4): 301-313
- [10] Bidarra R, Madeira J, Neels W J, et al. Efficiency of boundary evaluation for a cellular model[J]. Computer-Aided Design, 2005, 37(12): 1266-1284