文章编号:1001 - 9081(2006)12 - 3047 - 04

一种拓扑元素的标识和识别方法

缪金防,孙学波
(鞍山科技大学 计算机科学与工程学院,辽宁 鞍山 114044)
(mjinfang@163.com)

摘 要:给出了一种拓扑元素的标识方法,并以此方法为基础提出一种称为 TRG 的数据结构,同时给出了拓扑面、边、点的识别算法。系统通过 TRG 记录实体模型的构造历史,跟踪实体重建前后拓扑元素之间的关系。该方法能很好地记录实体的造型历史,再现原设计者的设计意图,给出的算法能正确识别面、边、点。

关键词:实体造型; 实体元素标识和识别; CAD
中图分类号: TP391.41 **文献标识码**:A

Approach to naming and identifying topological entities

MIAO Jin-fang, SUN Xue-bo
(*School of Computer Science and Engineering, Science and Technology University of Anshan, Anshan Anhui* 114044, *China*)

Abstract: An approach to naming and identifying the topological entities was proposed, based on which a data structure named Topological Relation Graph (TRG) was introduced . TRG memorizes the solid process of a part and tracks the topological relation of entities between original and current solid part. Using TRG, solid modeling system can reevaluate part correctly according to the original designer's intent. Algorithms for identifying the solid face, edge and vertex were also given in this paper. The proposed approach can track the modeling history correctly, represent designer's intent of original solid part, also can identify the solid element including solid face, edge and vertex.

Key words: solid modeling; naming and identifying for topological entities; CAD

0 引言

随着基于特征的实体造型技术不断完善和成熟,现在很多商用 CAD 系统都是基于历史的特征造型系统。在这种系统中,实体模型的变动设计并不直接通过对实体形状的参数化驱动来完成,而是通过实体重建机制,如图 1 所示。

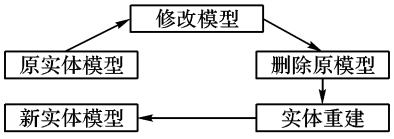


图 1 实体重建过程

比如要修改一个拉伸特征的拉伸高度,其重建过程为:

1) 修改拉伸参数,设置一个新值;

2) 删除旧拉伸实体;

3) 用新的拉伸参数进行实体重建,从而得到具有新高度的拉伸实体。

然而现有的系统在重建一些复杂实体模型的时候常得到无法令人满意的结果。主要有以下三种情况:

1) 重建之后的模型没有完全按照原模型的设计历史正确再现原设计者的设计意图;

2) 在实体重建过程中无法正确跟踪重建前后拓扑元素之间的关系;

3) 无法正确识别重建后模型的拓扑元素。

为避免以上三种情况的发生,在基于历史的特征造型系统中必须提供记录设计历史,跟踪实体重建前后拓扑元素之间关系,正确识别拓扑元素的方法,为实体模型的重建提供可靠保证,这就是拓扑元素的命名与识别问题。

拓扑元素的命名和识别问题在基于历史的特征造型系统中是一个非常重要且具有相当难度的问题,文献[1]把它归结为该类系统中需要解决的六大问题之一,几乎所有的基于历史的特征造型系统都设有单独模块解决这一问题,这方面的研究也是 CAD 领域研究的热点。文献[2,3]分别提出了面的命名和识别方法,但他们的方法不能辨识边和点。文献[4]给出了一种拓扑标号(Topologcal ID)系统,用来映射实体重建前后拓扑元素的依赖关系,他同时给出了一种基于图的方法,用以表示一些实体元素的变化,例如合并、分裂等,但遗憾的是该文并没有给出比较详细的说明,同时 Kripac 所采用的算法也比较繁琐。文献[5] 为每一个拓扑元素分配一个唯一的标识号,但这种方法并不能满足重建机制的要求。因为实体在重建过程中,删除旧实体的同时,其标号也随之删除,无法映射重建前后拓扑元素之间的依赖关系。文献[6]给出了一种简单的面和边的编码体系。文献[7]扩展了 Capoyleas 的标识方法,并通过检查被引用元素和被辨识元素间的影响特征集、影响面集以及非影响面集是否存在子集关系进行拓扑元素的辨识。这一方法中,面面进出关系的确定需要相当的计算量,同时该文没有给出具体算法。文献[8]等在 Kripac 基础上提出了一种称为 NPG 的方法,该方法能正确跟踪实体元素重建前后的面依赖关系,但对两个实体面相交形成多条

收稿日期:2006 - 06 - 21;**修订日期**:2006 - 08 - 18

作者简介:缪金防(1980 -),男,山东东营人,硕士研究生,主要研究方向:计算机图形学、CAD; 孙学波(1964 -),男,辽宁鞍山人,副教授,主要研究方向:计算机图形学、CAD.

实体边的时候,提供的算法存在无法正确识别的情况,同时文章中没有详细给出实体点的标识及识别方法。

本文认为拓扑元素的标识与识别必须满足以下条件:

1)唯一标识拓扑元素并保证实体重建后其有效性;

2)正确记录拓扑元素的分裂、合并等历史操作,并在实体重建时根据这些记录再现实体模型;

3)正确跟踪重建前后拓扑元素之间的依赖关系;

4)识别算法能正确识别出拓扑元素。

本文从满足实体重建功能要求的角度出发,给出一种拓扑元素标识与识别方法,以有效跟踪实体重建前后拓扑元素之间的关系,使实体重建满足实际需要。本文主要针对刚性实体展开讨论,对于其他特征,对本文方法进行适当扩展和改进即可。

1 拓扑元素的标识

1.1 拓扑面的拓扑标识结构

在实体模型中,面是最主要的拓扑元素,CAD 系统中对拓扑面标识方法的优劣直接决定该系统的稳定性。

本文给出了一种在整个系统内唯一标识面的拓扑标识结构,结构如下所示:

$FaceTI = \{FeatID, FaceID, Seq, Num\}$

其中 *FeatID* 是实体模型内特征的标识号,模型为每一个新建的特征分配一个标识号,以此在模型范围内唯一标识该特征,其大小表示创建特征的时间先后顺序,在模型范围内从 1 开始依次递增。*FaceID* 代表特征为其所包含的每一个面分配的一个原始标识号,在特征范围内从 1 开始依此递增。*Seq* 代表面在当前模型中被分裂或合并的次数。*Num* 用来标识面被分割形成子面的标识号。

采用一种叫做拓扑关系图(Topological Relation Graph, TRG)的结构来记录实体模型的构造历史,跟踪拓扑元素在重建前后之间的关系。

定义 1[8] $TRG = \{E, R\}$,其中 E 是一个节点集合,每一个节点代表一个面的拓扑标识结构,用符号 *CC* 表示。R 代表图中节点之间关系集合:

$$R = \{ < CC_i, CC_j > \mid i \neq j, CC_i \rightarrow CC_j \}$$

R 表示节点 CC_i 所标识的实体面与节点 CC_j 标识的实体面之间存在一种父子关系,这种关系由保持,分裂,合并而产生。

TRG 有下列性质:

1)图的根节点表示模型内的一个特征;

2)图中出度为零的节点为叶子节点,所代表的拓扑面存在于当前模型中。

定义 2 如果在原模型中面 $f1$ 和当前模型中面 $f2$ 有相同的 *FeatID* 和 *FaceID*,那么面 $f2$ 称为面 $f1$ 的派生面,$f1$ 称为 $f2$ 的父面,两者之间为父子关系;否则面 $f2$ 为生成面。

如果一个面是生成面,特征会为之分配一个新的 *FaceID*;如果该面是派生面,那么该面继承父面的 *FaceID*。当在模型内生成一个新的特征时,系统为该特征分配一新的 *FeatID*,模型中由于新特征的加入而新生成的面隶属于该特征,特征为其包含的所有生成面分配新的 *FaceID*。

根据对面进行操作的不同,面的 *FaceTI* 分配策略区分如下:

保持:子面继承父面的 *FeatID*,*FaceID*,其中 *Seq* 加 1;

分裂:各子面继承父面的 *FeatID*,*FaceID*,其中 *Seq* 加 1,根据分裂面的生成顺序,依次设置子面的 *Num*,该序号必须从 1 开始;

合并:如果要合并的面不属于同一特征,那么合并产生的面,隶属于 *FeatID* 较小面所在的特征。

下例用一个 TRG 图跟踪了实体的创建过程,实体模型的设计过程为:

1)创建原始模型如图 2(a);

2)加入一个切除特征 *slot*1 如图 2(b);

3)加入另一个切除特征 *pocket*1 如图 2(c);

4)修改 *pocket*1 的参数,使其成为 *slot*2, 如图 2(d);

5)删除 *slot*1 如图 2(e)。

设在图 2(a)中,实体模型为特征 *feat*1 分配的标识号为 1,特征 *feat*1 为面 *f*1_1 和 *f*1_2 生成的面标识号分别为 1,2。属于特征 *feat*1 的面一共有 6 个,为使图看上去比较清晰,我们只以其中的两个面 *f*1_1 和 *f*1_2 为例来说明。

根据节点结构的定义,各拓扑面节点值如下:

$CC(f1\_1) = \{1,1,0,0\}$ $CC(f1\_2) = \{1,2,0,0\}$

$CC(f2\_1) = \{1,1,1,1\}$ $CC(f2\_2) = \{1,1,1,2\}$

$CC(f2\_3) = \{1,2,1,0\}$ $CC(f3\_1) = \{1,1,2,1\}$

$CC(f3\_2) = \{1,1,2,2\}$ $CC(f3\_3) = \{1,2,2,0\}$

$CC(f4\_1) = \{1,1,3,1\}$ $CC(f4\_2) = \{1,1,3,2\}$

$CC(f4\_3) = \{1,1,3,3\}$ $CC(f4\_4) = \{1,2,3,0\}$

1.2 拓扑点的拓扑标识结构

本文根据点在实体模型中其拓扑信息是由其相邻面决定的,本文给出点的拓扑标识结构 $VertexTI = \{FaceTI_1, FaceTI_2, FaceTI_3, \cdots, FaceTI_n, VertexID\}$,其中 *FaceTI* 为点的相邻面,点相邻面的数目一般为多个,*VertexID* 为模型分配给该点的标识号,该标识号在模型内唯一,从 1 开始依此递增。

定义 3 如果当前模型中点 $p2$ 的相邻面和原模型中点 $p1$ 的相邻面数目相同,且 $p2$ 的所有相邻面都能在点 $p1$ 的相邻面中找到父面,则点 $p2$ 是 $p1$ 的派生点,$p1$ 称为 $p2$ 的父点,两者之间为父子关系;否则 $p2$ 为生成点。

如果当前模型中的一个点为派生点,则该点的 *VertexID* 继承其父点的 *VertexID*;若为生成点,模型为该点分配新的 *VertexID*。

1.3 拓扑边的拓扑标识结构

在实体模型中,本文根据边的相邻面和其端点来表示该边的拓扑结构。

$EDgeTI = \{FaceTI_1, FaceTI_2, VertexTI_1, VertexTI_2\}$,其中面 $FaceTI_1$ 和面 $FaceTI_2$ 为该边的两个相邻面,点 $VertexTI_1$ 和点 $VertexTI_2$ 为实体边的两个端点。

定义 4 如果当前模型中边 $e2$ 的相邻面都能在原模型中边 $e1$ 的相邻面中找到父面,且边 $e2$ 的一个端点是边 $e1$ 某个端点的派生点,那么边 $e2$ 是边 $e1$ 的派生边,$e1$ 是 $e2$ 的父边,两者之间为父子关系;如果在原模型中不存在边 $e2$ 的父边,则边 $e2$ 是生成边。

1.4 实体模型的标识结构

我们定义模型的拓扑数据结构如下:

$PartTI = \{PartID, PartArray, EdgeArray, VertexArray\}$

其中 *PartID* 为系统为每一个实体模型分配的标识号,该

标识号能唯一标识一个实体模型。*FeatArray* 内存储该实体模型每一个特征的 *FeatID* 和该特征内的 *FaceArray*。*FaceArray* 内存储某个特征内所有生成面的 *FaceID* 和指向每一个生成面在TRG中的位置指针。*EdgeArray* 和 *VertexArray* 内每一个元素分别存储实体模型内边和点的拓扑标识。

2　拓扑元素的识别

2.1　拓扑面识别算法

根据定义 2 和上文给出 TRG 的性质,本文给出了通过给定在原实体模型中面的 *CC*,来识别重建后当前模型中此面的子面算法描述。

算法 1:

1) 获得当前模型下的特征数组 *FeatArray*;

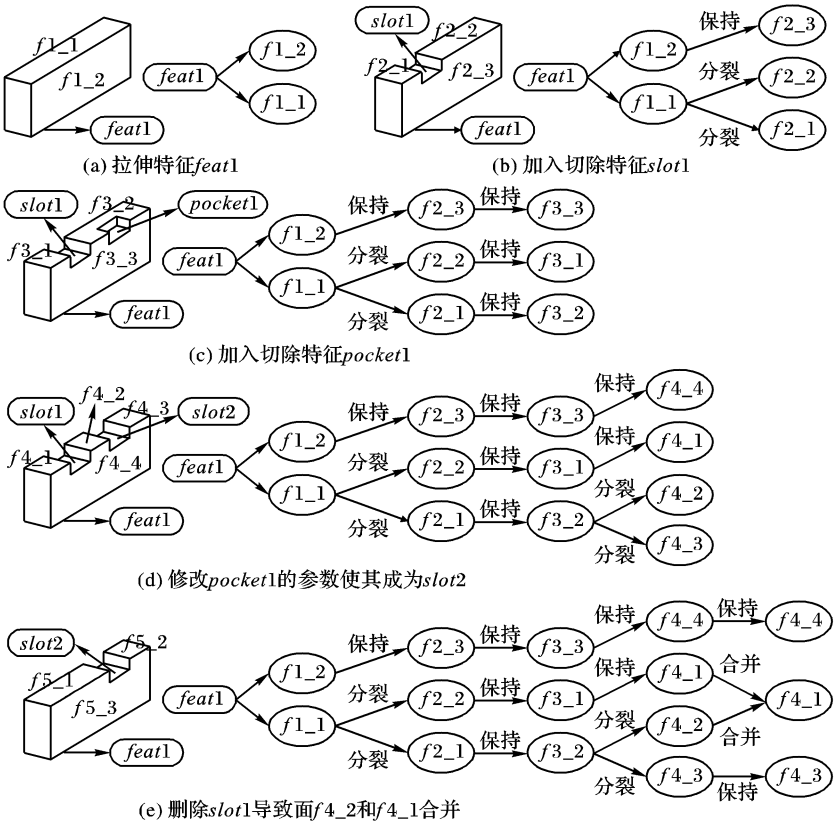
2) 判断 *FeatArray* 中有没有 *FeatID* 为 *CC. CC. FeatID* 的特征,若有设为 *FeatArray*[*i*] 执行第 3 步;如果没有则所要识别的面不存在,结束程序。

3) 判断特征下生成面数组 *FeatArray*[*i*]. *FaceArray* 中有没有 *CC. CC. faceID* 的原始面,若有则记下标识该原始面的节点在 TRG 图中的指针并执行步 4;如果没有则所要识别的面为非法,结束程序。

4) 深度优先搜索,查找 *CC* 在 TRG 图中的位置;

5) 以 4) 的查询结果为始点,深度优先搜索 TRG,直到遍历完所有节点。得到所有叶子节点集合 *VS*,返回 *VS*,程序结束。

VS 即是原模型中节点 *CC* 所标识的面在当前实体模型中的子面。



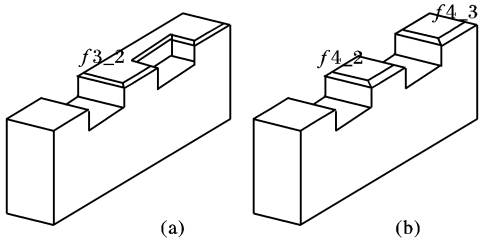
(a) 拉伸特征*feat*1　　(b) 加入切除特征*slot*1

(c) 加入切除特征*pocket*1

(d) 修改*pocket*1的参数使其成为*slot*2

(e) 删除*slot*1导致面*f*4_2和*f*4_1合并

图 2　实体操作对应的拓朴关系



(a)　(b)

图 3　修改切除特征尺寸

下面以一个例子进一步说明算法 1,设图 2(c) 所示的实体模型为原模型,修改该模型 *pocket*1 的参数,使之成为 *slot*2,重建后的当前模型如图 2(d)。给出原模型中面 *f*3_2 的节点标识,根据算法 1 识别图 2(d) 中面 *f*3_2 的子面过程为:

1) 根据 *CC*(*f*3_2). *FeatID* = 1 找到标识特征 *feat*1 TRG 图的根节点;

2) 在 *feat*1 的面数组中 *FaceArray* 查找 *FaceID* 为 *CC*(*f*3_2). *FaceID* 的面,找到节点 *CC*(*f*1_1) 的指针;

3) 从节点 *CC*(*f*1_1) 开始深度优先查找节点 *CC*(*f*3_2) 在 *TRG* 图中的位置;

4) 从节点 *CC*(*f*3_2) 位置开始深度优先查找所有叶子节点,返回节点 *CC*(*f*4_2),*CC*(*f*4_3)。

面 *f*4_2,*f*4_3 即为面 *f*3_2 的子面。

假若对图 2(c) 中面 *f*3_2 添加倒角特征如图 3(a),然后修改该模型 *pocket*1 的参数,使之成为 *slot*2,根据算法 1,由于面 *f*4_2, *f*4_3 是面 *f*3_2 的子面,子面继承父面的特征,于是重建后得到图 3(b) 所示模型。

2.2　拓扑点识别算法

由于在实体模型中,点不存在分裂、合并操作,所以识别当前模型中的一个点 *p*1 是否为原模型中某个点的派生点,只要比较两者对应的 *VertexID* 是否相同即可。若相同,两者为父子关系;否则 *p*1 为生成点。

给定原模型中一个点的拓扑标识 *VertexTI*,通过下文给出的算法 2 可找到该点在当前模型中的派生点。

算法 2:

1）通过给定原始模型中点 $p1$ 的 *VertexTI* 得到该点的 *VertexID*，设为 *VertexID*1；

2）在 *VertexArray* 查找 *VertexID* 为 *VertexID*1 的点，若找到设其 *VertexTI* 为 *VertexID*2，否则转 4）；

3）返回 *VertexID* 为 *VertexID*2 的点；

4）算法结束。

对该算法进行修改，可以得到识别分别属于原模型和当前模型中 2 个给定点是否为派生关系等识别算法。

2.3 拓扑边识别算法

给出原模型中边 $e1$ 的拓扑标识：

$$EdgeTI(e1) = \{FaceTI(f1), FaceTI(f2), VertexTI(v1), VertexTI(v2)\}$$

根据定义 4 在当前模型中识别 $e1$ 的子边算法如下：

算法 3：

1）根据 $EdgeTI(e1)$，得到其相邻面 $f1$，$f2$ 的拓扑标识 $FaceTI(f1)$，$FaceTI(f2)$，端点 $v1$，$v2$ 的拓扑标识 $VertexTI(v1)$，$VertexTI(v2)$；

2）分别对 $FaceTI(f1)$，$FaceTI(f2)$ 调用算法 1，返回 $f1$，$f2$ 的在当前模型中的子面集合分别：

$$VF1 = \{FaceTI(f_1), \cdots, FaceTI(f_n)\}$$

$$VF2 = \{FaceTI(f_1), \cdots, FaceTI(f_m)\}$$

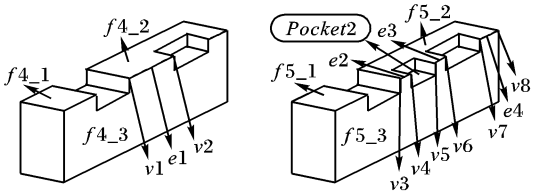
3）在 *EdgeArray* 查找以 *VF*1 和 *VF*2 中的面为邻面的边集：$VE1 = \{EdgeTI(e_1), \cdots, EdgeTI(e_1), \cdots, EdgeTI(e_k)\}$；

4）分别对 $VertexTI(v1)$，$VerTexTI(v2)$ 调用算法 2，得到 $v1$，$v2$ 在当前模型中的派生点分别为：$VertexTI(v3)$，$VertexTI(v4)$；

5）在 *VE*1 中查找以 $VertexTI(v3)$ 或 $VertexTI(v4)$ 为端点的边，得到边集 $VE = \{EdgeTI(e_1), \cdots, EdgeTI(e_h)\}$，返回 *VE*；

6）算法结束。

VE 中的边即为原模型中边 $e1$ 在当前模型中的子边集合。



(a) 修改图2(c)的尺寸　　(b) 加入切除特征 *pocket*2

图 4　修改、添加切除特征

图 4（a）为修改图 2（c）中模型尺寸，加入切除特征 *pocket*2，重建后形成图 4（b）。给出图 4（a）中边 e1 的拓扑标识，利用算法 3 在当前模型图 4（b）中识别其子边的过程如下：

1）首先根据 $EdgeTI(e1)$ 得到其相邻面 $f4\_2$，$f4\_3$，端点 $v1$，$v2$；

2）然后调用算法 2 得到在图 4（b）中 $v1$，$v2$ 的派生点分别为 $v3$，$v6$；

3）调用算法 1，找到面 $f4\_2$，$f4\_3$ 在图 4（b）中的子面分别为 $f5\_2$，$f5\_3$，在当前模型图 4（b）的 *EdgeArray* 中得到以面 $f5\_2$，$f5\_3$ 为临面的边为 $e2$，$e3$，$e4$；

4）在边集 $e2$，$e3$，$e4$ 中以 $v3$ 或 $v6$ 为端点的边为 $e2$，$e3$。

当前模型图 4（b）中，边 $e2$，$e3$ 即是图 4（a）中边 $e1$ 的子边。

3 算法实现

采用本文提出的拓扑元素标识及命名方法，我们在 Parasolid 几何造型平台上，用 VC++6.0 开发了基于特征的实体造型原型 TDSMS。

公告牌是 Parasolid 提供的一种跟踪拓扑变化中父子关系的功能。TDSMS 通过 Parasolid 提供的造型函数构造实体模型，公告牌中保存有当原实体模型和当前实体模型中拓扑元素变化关系，TDSMS 把这些变化关系记录在 TRG 结构中，系统根据 TRG 中存储的内容完成实体重建。TRG 记录的历史操作信息保证 TDSMS 正确重建实体模型，重建前后拓扑元素之间的依赖关系得到了很好的跟踪。

4 结语

拓扑元素的标识及其识别在基于历史的特征造型系统中是一个非常关键的环节。拓扑元素的组织结构是否优秀直接决定造型系统的稳定性。本文在前人研究的基础上给出了实体面、边、点的拓扑标识结构和一种跟踪历史操作记录的结构 TRG 图，同时给出了实体面、边、点的识别算法。根据不同的特征对实体面进行分组，这种组织结构能有效提高识别算法的效率。

本文的方法能有效跟踪拓扑元素重建前后的依赖关系，可以确保实体重建能正确再现原设计者的设计意图。能较好标识和识别刚性实体元素，但对于实体曲面曲线的标识和识别存在不能正确识别的情况。

针对本文方法在标识和识别实体曲线曲面方面的不足，下一步工作主要针对有效标识和识别曲线曲面元素这一问题展开。

参考文献：

[1] BIARRA R, BRONSVOORT WF. Semantic feature modeling[J]. Computer-Aided Design, 2000, 32(3): 201-225.

[2] REQUICHA AA, CHAN SC. Representation of geometric features tolerace and attributes in solid modellers based on constructive geometry [J]. IEEE Computer Graphics and Applications, 1986, 2(3): 156-166.

[3] TURNER GP, ANDERSION DC. An object-oriented approach to interactive feature based design for quick turnaround manufacturing [A]. Proceedings of ASME Computers and Engineering Conference [C]. California, 1988, 1: 551-555.

[4] KRIPAC J. A mechanism for persistently naming topological entities in history-based parametric solid models[J]. Computer-Aided Design, 1997, 29(2): 113-122.

[5] CAPOYEAS V, CHEN X, HOFFMANN CM. Generic naming in generative constraint-based design[J]. Computer-Aided Des 1996, 28(1): 17-26.

[6] 苏晓锋，黄正东，朱林，等. 形状特征中的拓扑元素编码体系[J]. 计算机辅助设计与图形学学报，2000，12(2)：137-141.

[7] 陈正鸣，高曙明，张凤军，等. 一种拓扑元素的命名和辨识方法[J]. 计算机学报，2001，11(24)：1271-1277.

[8] WANG YW, WU JJ, CHEN LP, *et al*. Identity propagation method for tracing alterations alterations of a topological entity in a history-based solid modeling system[J]. The International Journal of Advanced Manufacturing Technology, 2005, 27(3): 305-312.