

一种特征造型的拓扑元素命名机制

高雪瑶¹ 孙立鏖¹ 孙大松²

(哈尔滨理工大学计算机科学与技术学院 哈尔滨 150080)¹ (哈尔滨理工大学计算中心 哈尔滨 150080)²

摘 要 特征造型是 CAD 系统的关键技术,而拓扑元素命名机制是特征造型最根本的问题之一。提出一种以特征为基础,对特征实体的面进行统一命名的方法,并在此基础上对特征实体的边和点进行命名。特征名在建模过程中保持不变,从而实现在模型编辑过程中对拓扑元素的正确记录与维护。针对拓扑边的分裂及曲面相交产生多条交线的问题,给出了参考线的概念及定义原则,有效解决了对分裂边及多交线的区分与引用问题。

关键词 CAD,特征造型,特征,参考线

中图法分类号 TP391.7 **文献标识码** A

Mechanism of Naming Topological Entities in Semantic Feature Modeling

GAO Xue-yao¹ SUN Li-quan¹ SUN Da-song²

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

(Computer Center, Harbin University of Science & Technology, Harbin 150080, China)²

Abstract Feature modeling is the key technique in CAD system, and topological entity naming is one of the essential issues of feature modeling. A feature-based method was proposed to name the surfaces of feature entities, then name edges and vertices of feature entities. Because of invariance of feature name in process of modeling, topological entities can be correctly recorded and maintained in model editing. The conception of reference line and its definition principle were proposed to solve topological edge splitting and multiple intersecting curves generated by two surfaces, which can identify and refer the splitting edges and multi-intersections efficiently.

Keywords CAD, Feature modeling, Feature, Reference line

1 引言

拓扑元素的命名和辨识是 CAD 领域的难点问题之一^[1,2], Bidarra^[3]把它归结为目前参数化特征造型系统需要解决的六大问题之一。

Kripac^[4]首先提出拓扑元素永久命名这一术语,并设计了一个拓扑标识系统,把被引用的拓扑元素 ID 从拓扑标识系统映射到新模型中的相应元素。Capoyreas^[5]和 Chen 等^[6]根据拓扑元素的局部拓扑关系特性、边与点的局部方向性以及特征方向等信息对拓扑元素进行命名,通过比较新旧模型拓扑元素的名字建立起新旧元素间的对应关系。苏晓锋等^[7]提出一种面和边的编码方法,该方法按照子面参数域在父面参数域上的顺序对子面排序,以区分分裂面。陈正鸣等^[8]利用面的方向性定义了三类面面进出关系,通过面面进出关系表示拓扑元素的局部拓扑信息,并通过影响特征集、影响面集和非影响面集进行拓扑元素的辨识。Zheng 等^[9]提出一种基于面的命名方法,用步骤 ID 和内部 ID 来命名面,用子面的邻接面区分不同的子面,并用邻接面命名边和顶点。Yan 等^[10]提出一种基于几何连续性的参数族的确定方法,并在此基础上

建立几何和拓扑元素的命名机制及其在新旧模型中的映射方法。Wang 等^[11]定义了名字传递图(NPG)来实现拓扑实体名的一致性传递,并给出了辨识初始拓扑实体的规则与算法。Marcheix^[12]从拓扑元素的集合,即壳这一角度来考虑拓扑命名问题,定义了壳的分层结构,并提出了一种匹配算法来实现壳名在模型重构过程中的识别和传递。

本文针对特征模型编辑后拓扑元素的不同变化情况,提出了一种基于特征的拓扑元素命名方法,实现对模型中拓扑元素的唯一命名与正确辨识,从而正确修改与重构特征模型。

2 拓扑元素命名方法

在特征造型中,特征是模型的基本单位。在建模过程中,特征名保持不变。基本形状特征是具有最简单特性的特征,所有复杂特征都是由基本形状特征经过特征操作组合而成,因此本文对基本形状特征中拓扑元素的命名方法进行分析,复杂特征中的拓扑元素命名均可在此基础上生成。

在几何上,三维空间中的直线可以用两个相交的面来表示,而点可以用 3 个相交于一点的面来表示。将这一思想拓展到拓切实体的命名上来,实体的拓扑边通过它的两个邻接

面的拓扑名组合来表示,拓扑点通过它的3个邻接面的拓扑名组合来表示。具体命名方法为:以特征为基础,首先对特征实体的面进行命名,然后以面的拓扑名为基础,分别对特征实体的边和点进行命名。以下分别对面、边、点的命名进行分析。

2.1 面命名

基本形状特征按照其生成方式可以分为由扫略类操作和旋转类操作生成,具有可明确定义的面。本文规定其各个面的命名规则为:以特征名和面在该特征内部的名字共同确定。当该特征依附于某一个基准面(依附面)时,规定与该基准面相接的面为该特征的底面,而与之相对的面是特征顶面。具体标记方法为(feature_instance_name.feature_element_ID)。图1中的圆柱体共有3个面:顶面、底面和侧面。图中给出了各个面的命名。

基本形状特征由扫略和旋转操作生成,特征的侧面可能包含多个子侧面,需要对这些子面进行区分。

对于扫略操作所形成的特征,由于扫略可以分为平扫(拉伸)和广义扫两种情况,平扫的路径为简单直线路径,所生成特征侧面的数量与草图边的数量相等;广义扫是由草图轮廓沿着复杂路径扫描而成,其扫描路径可能是二维或三维连续线段,因此每一条草图边所对应的侧面数可以是多个。平扫是广义扫的简单特例,这里主要分析由广义扫生成的特征。对广义扫生成的特征,先给草图轮廓边分配编号,然后对扫描路径进行分段,最后依据路径段号和草图轮廓边号对各个侧面进行命名。本文规定草图轮廓边的走向为空间逆时针走向。假设轮廓有 N 条边,且所在局部坐标系为 uv 坐标系,路径由 M 条线段组成,则对各个侧面命名的步骤描述如下:

步骤1 比较各轮廓边的起点的 u 坐标值,并记录具有最小 u 坐标值的边。

步骤2 判断具有最小 u 坐标值的边数是否为1。若为1,置该边 $ID=1$,并转步骤5,否则转下一步。

步骤3 对所有具有最小 u 坐标值的边,比较这些边的起点的 v 坐标值。

步骤4 记录具有最小 v 坐标值的边,置该边 $ID=1$ 。

步骤5 取 $ID=1$ 的边,将其作为遍历起点。

步骤6 按空间逆时针的走向遍历后续各边。

步骤7 依次置后续边 $ID=ID+1$,并判断 ID 是否等于 N 。若等于 N ,转下一步,否则转步骤6。

步骤8 从扫描路径的起点开始,将第一段路径的标号 L 置1。

步骤9 按路径的走向,置后续路径 $L=L+1$ 。

步骤10 判断 L 是否等于 M 。若等于 M ,转下一步,否则转步骤9。

步骤11 根据路径段号 L 和轮廓边号 ID 对侧面进行命名,标记为(sweep.side_L_ID)。

图2是一个广义扫特征的简单例子,由正方形沿着空间折线段扫描而成,包括底面、顶面和12个侧面。根据上述命名方法,第一段轮廓边与第一段路径共同确定的子侧面记为sweep.side11,与第二段路径共同确定的子侧面记为sweep.side21,依此类推。图中标出了第一段轮廓边对应于3段路

径的各个侧面的标号。

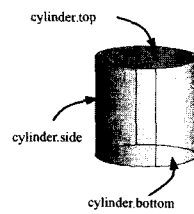


图1 圆柱的特征面

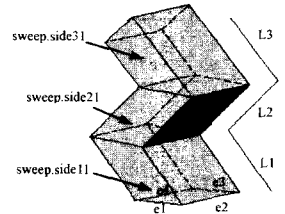


图2 广义扫特征侧面命名

对于平扫生成的特征,扫描路径只有一段,各个侧面的ID直接对应于草图轮廓边的ID,而不需考虑路径的编号。图3是以长方体为平扫特征的一个例子,并给出各个面的标号。

旋转操作所生成的特征是由轮廓沿着旋转轴以某一角度旋转扫描而成。对于这种情况,首先对各轮廓边进行编号,然后根据编号对旋转操作所生成的各个侧面进行命名。设轮廓有 N 条边,具体命名步骤描述如下:

步骤1 对于轮廓的每条边,判断它们的起点到旋转轴的距离,确定其中的最小值。

步骤2 将具有最小值的边作为基准边,按照轮廓的走向找到下一条边,并置其 $ID=1$ 。

步骤3 按轮廓的走向遍历后续各边。

步骤4 依次置后续边 $ID=ID+1$,并判断 ID 是否等于 N 。若等于 N ,转下一步,否则转步骤3。

步骤5 判断基准边是否与旋转轴重合。若不重合,根据轮廓ID依次对侧面进行命名,标记为:(rotate.side_ID);否则只标记前 $N-1$ 个侧面,不考虑基准边对应的面。

上述第5步中,若基准边与旋转轴重合,则在旋转过程中,基准边无法产生相应的面,即rotate.side n 不存在,因此不考虑该面。图4是旋转特征的一个例子,各个侧面分别为rotate.side1, rotate.side2, rotate.side3, rotate.side4。

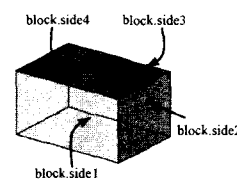


图3 长方体侧面命名

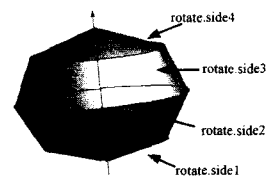


图4 旋转特征侧面命名

当经过某些特征操作后,若一个特征面分裂为若干个子面,则需要对这些子面加以区分。对于这种情况,本文借鉴文献[7]的方法,并对其加以改进:按照子面在父面参数域上的顺序对子面进行排序,并保留父面的ID,用于说明拓朴元素的变化情况。假设特征所在局部坐标系为 uvw 坐标系,首先对父面进行参数化,然后求出各个子面在父面参数面上的参数域,按照先 u 方向、后 v 方向、然后 w 方向的顺序,对各个子面排序。命名规则为:(feature_instance_name.feature_element_ID,f ID)。在图5中,block特征的顶面被两个slot特征分裂为4个面,这4个面的父面ID,即feature_instance_name.feature_element_ID都是block.top,子面ID即f ID,分别为f1,f2,f3,f4。在这个例子中,父面block.top为平面,因此省去了在 w 方向上对子面的排序。

2.2 边命名

在特征造型过程中,由于特征相交、参数修改等操作,会导致边的变化非常频繁,而边的变化会引起不同的操作结果。

在造型过程中,很多时候都要用到边,例如倒角、倒圆等操作。例如在图 6 中的长方体上开一个盲槽,将长方体顶面的一条边分裂为两条。若要对其中的一条边(例如 $e1$ 边)进行倒角、倒圆等操作,就需要正确区分这两条边。

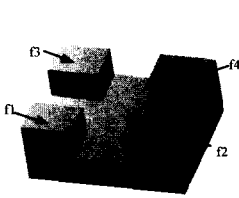


图 5 分裂子面的命名

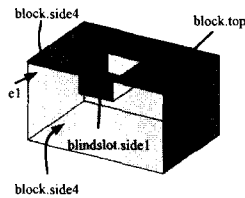


图 6 分裂边的命名

拓扑边可以由两个邻接面唯一确定,而边的端点是由另一个面与这两个邻接面相交形成,因此可以通过由特征定义的特征面来命名模型中的边。命名规则为(adjacent_feature_faces), (end_faces)。其中, adjacent_feature_faces 为边的两个邻接面, end_faces 为边的两个端点所在的起始面和终止面。因此图 6 中的 $e1$ 边表示为(block.top, block.side1), (block.side4, blindslot.side1)。

当模型中存在曲面时,由两个面相交产生的边可能会多于一条。例如在图 7 中圆柱体的顶面开一个圆槽,圆柱体的顶面与圆槽的侧面相交生成两条边,此时这两条边的 adjacent_feature_faces 都是 cylinder.top 和 roundslot.side, end_faces 都是 cylinder.side,无法正确区分这两条边。对于这种情况,可以通过参考线 reference 来区分:在特征造型系统中定义一条有方向的参考线,规定沿参考线的方向前进,行进方向的左侧区域为正,以此将需要区分的拓扑实体划分在参考线的两侧,来识别分裂后的拓扑实体。描述规则为(adjacent_feature_faces), (end_faces, reference.orientation), 其中 reference.orientation 用于说明要区分的拓扑实体位于参考线的哪一侧。对于图 7 的两条边,通过定义一条位于 cylinder 的顶面且与两条相交边平行的参考线来区分:

$e1$: (cylinder.top, roundslot.side), (cylinder.side, roundslot.side, reference.positive)

$e2$: (cylinder.top, roundslot.side), (cylinder.side, roundslot.side, reference.negative)

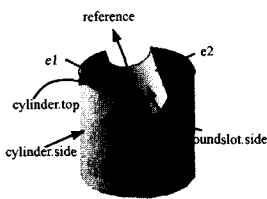


图 7 利用参考线命名边

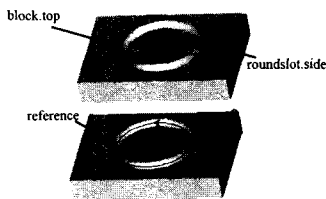


图 8 用封闭参考线区分边

以上例子中,两个面相交生成的交线可被对称划分于一条参考线的两侧,定义的参考线是直线。而图 8 中,在长方体的顶面开一个环形槽,形成内外两条封闭环形边。要区分这两条边,就需要定义一条封闭的曲线参考线:令参考线通过环形槽的中心线。因此,两条封闭环形边可以分别表示为:

$e1$: (block.top, roundslot.side), (reference.positive)

$e2$: (block.top, roundslot.side), (reference.negative)

这里,由于这两条边为环形封闭边,没有起点与终点,因此 end_faces 忽略不计。

由以上例子可以看出,对于边的分裂及存在多条交线的

情况,需要定义参考线并让需要区分的边落在参考线的两侧,以此进行命名。这里给出定义参考线的原则:参考线(包括直线和曲线)必须与特征的某些基准面、基准轴线或对称面相关联,如通过或垂直于轴线、与基准面重合或垂直等;参考线与特征的某条基准线确定一个平面,使要区分的两条边在该平面上的投影落在参考线的两侧;在该平面上,沿参考线的方向前进,行进方向的左侧区域为正;当特征的位置发生变化时,参考线的位置也相应发生变化。

2.3 点命名

在实体的拓扑关系中,顶点可看作面的交点,它的邻接面集刻画了其拓扑局部性质,因此顶点可以由其邻接面集来表示。例如图 9 中的点 $v1$ 为(block.top, slot.side2, block.side1)。

在有些情况下,仅由邻接面集无法唯一确定顶点,例如图 10 中 4 个顶点的邻接面都为 cylinder.top, roundslot.side 和 cylinder.side。这种情况与边的命名方法一样,借助参考线来区分各个顶点。描述规则为:(adjacent_feature_faces), (reference.orientation)。因此图 10 中的 4 个顶点可以分别表示为:

$v1$: (cylinder.top, roundslot.side, cylinder.side), (reference1.positive, reference2.negative)

$v2$: (cylinder.top, roundslot.side, cylinder.side), (reference1.negative, reference2.negative)

$v3$: (cylinder.top, roundslot.side, cylinder.side), (reference1.negative, reference2.positive)

$v4$: (cylinder.top, roundslot.side, cylinder.side), (reference1.positive, reference2.positive)

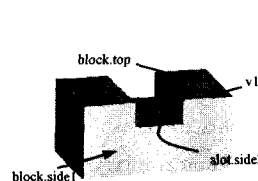


图 9 点的命名

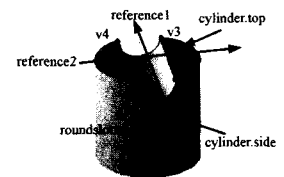


图 10 利用参考线区分点

3 方法实现与实例

将本文提出的拓扑元素命名机制应用到自主开发的语义特征造型系统(HUST-CAID)中,进行特征建模。图 11 是初始模型,在基体 block 上开两条通槽 slot1 和 slot2,其中 slot1 是圆槽。在 slot2 上是一个圆柱(cylinder),且圆柱不与 slot1 相交。对模型进行修改,减小 slot2 的深度,令它与圆柱相交,产生两条平行交线,并对其中一条进行倒圆角操作。利用本文的拓扑元素命名机制可正确区分这两条边,模型修改结果如图 12 所示。

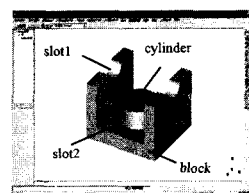


图 11 初始模型

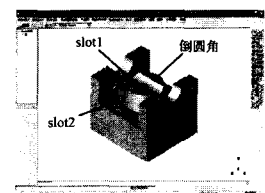


图 12 修改后的模型

(下转封三)

P	9	9	9
R	3.58	5.22	5.22
MT	0.60	0.42	0.42

从评价结果可以看出,四参数与五参数的丢失率是相当的,比三参数情况下减少了0.18,明显优于三参数的检索结果。因此,对纹理图片的检索,不宜选用三参数。

另对四参数与五参数组合以检索结果中相关图像的平均排序评价,得到表2的结果。

表2 相关图像的排序评价

评价量	四参数算法	五参数算法
AVRR	5.11	5.06
IAVRR	4.5	4.5
AVRR/IAVRR	1.14	1.12

可见,五参数查询结果AVRR虽然更接近IAVRR,但区别并不很大。

值得注意的是,实验对各种算法的检索时间进行了记录,在检索400多幅图像时,四参数的检索时间比五参数检索时间减少了1.6秒。这说明四参数的提取速度明显优于五参数。随着检索图片的增多,提取速度将相差更多,这对于实际应用中的图像检索技术是十分重要的。

结束语 综合分析表明,利用灰度共生矩阵进行基于纹理的图像检索,可提取四参数,采用加权的街区距离匹配,可平衡计算量及检索效果的双重需求,获得较满意的效果。同时,也应该看到,面临复杂的用户需求,一个单一的图像检索系统并不能得到最优解。如何通过多种算法的结合来更好地描述图像特征,并利用相关性反馈技术使图像同实际语义相

联系,在未来很长一段时间内都将是图像检索工作者研究的方向。

参考文献

- [1] Yan Ye. Text Image Compression Based on Pattern Matching [D]. University of California, 2002
- [2] Kia Omid E, Doermann David S, Rosenfeld Azriel, et al. Symbolic Compression and Processing of Document Images[J]. Computer Vision and Image Understanding, 1998, 70(3): 335-349
- [3] Karadimitriou K, Fenstermacher M. Image compression in medical image databases using set redundancy[C]// IEEE Proceedings of Data Compression Conference, 1997. DCC '97. March 1997: 445
- [4] Karadimitriou K, Tyler J M. The centroid method for compressing sets of similar images[J]. Pattern Recognition Letters, 1998, 19: 585-593
- [5] Lee Jiann-Der, Wan Shu-Yen, Ma Cherng-Min, et al. Compression Sets of Similar Images Using Hybrid Compression Model [C]// Proceedings 2002 IEEE International Conference on Multimedia and Expo, 2002. ICME '02
- [6] 杨有, 尚晋. 一种政府资源档案图像的二值化方法[J]. 计算机科学, 2007, 34(3): 227-229
- [7] Inglis S J. Lossless Document Image Compression[D]. University of Waikato, New Zealand, 1999
- [8] Hough P V C. Method and Means for Recognizing Complex Patterns[P]. US Patent 3,069,654, Dec. 1962
- [9] Woods R C. Digital Image Processing Second Edition[M]. Beijing: Publishing House of Electronics Industry, 2006: 532-534
- [10] Wong I Y, Casey R G, Wahl E M. Document Analysis System. IBM J. Research Develop, 1982, 26(6): 647-656
- [11] 杨波, 汪同庆, 叶俊勇. 带噪图像的多阈值零树编码方法[J]. 光电工程, 2004, 31(3): 60-63

(上接第268页)

- [3] Bhat D, Nayar S K. Stereo and specular reflection [J]. IJCV, 1998, 26(2): 91-106
- [4] Brelstaff G, Blake A. Detecting specular reflections using Lambertian constraints[C]// Proc. Int. Conf. on Computer Vision, 1988: 297-302
- [5] Jin H, Soatto S, Yezzi A. Multi-view Stereo Beyond Lambert[C]// Proc. IEEE Conf. CVPR, 2003
- [6] Lin S, Li Y, Kang S, et al. Diffuse-Specular Separation and Depth

Recovery from Image Sequences[C]// Proc. European Conf. Computer Vision, 2002: 210-224

- [7] Shafer S. Using Color to Separate Reflection Components[J]. Color Research and Applications, 1985, 10: 43-51
- [8] Yang Y, Yuille A, Lu J. Local, global, and multilevel stereo matching[C]// CVPR, 1993: 274-279
- [9] Zhou Wei, Kambhampettu C. Binocular Stereo Dense Matching in the Presence of Specular Reflections[C]// IEEE Conf. CVPR, 2006

(上接第285页)

结束语 在特征造型系统中,拓扑元素的命名和正确辨识是系统实现的一个关键问题。本文分析了现有拓扑元素命名和辨识方法存在的主要问题,提出了一种基于特征的拓扑元素命名方法,有效实现了在模型拓扑结构发生变化时对拓扑元素的唯一命名与正确辨识,为特征模型的正确修改、重构及维护设计者的设计意图打下了良好的基础。

参考文献

- [1] Marcheix D, Pierra G. A survey of the persistent naming problem[C]// Proc. of the Symposium on Solid Modeling and Applications. Saarbrücken, Germany, 2002: 13-22
- [2] 荆树旭, 何发智, 刘华俊. 拓扑元素永久命名综述[J]. 计算机辅助设计与图形学学报, 2007, 19(5): 545-552
- [3] Bidarra R, Bronsvort W F. Semantic feature modeling[J]. Computer-Aided Design, 2000, 32: 201-225
- [4] Kripac J. A mechanism for persistently naming topological entities in history-based parametric solid models[C]// Proc. of the 3rd ACM Symposium on Solid Modeling and Applications. Salt Lake City, Utah: ACM Press, 1995: 21-30

- [5] Capoleas V, Chen X, Hoffmann C M. Generic naming in generative, constraint-based design[J]. Computer-Aided Design, 1996, 28(1): 17-26
- [6] Chen X, Hoffmann C M. On editability of feature-based design [J]. Computer-Aided Design, 1995, 27(12): 905-914
- [7] 苏晓锋, 黄正东, 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
- [8] 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, 24(11): 1270-1277
- [9] Zheng Jiajun, Fan Ming, Tong Ruofeng, et al. A mechanism for persistently naming topological entities[C]// Proc of the 8th International Conference on Computer Supported Cooperative Work in Design. Xiamen, China, 2004: 84-89
- [10] Wang Yan, Bartholomew O N. Geometry-based semantic ID for persistent and interoperable reference in feature-based parametric modeling[J]. Computer-Aided Design, 2005, 37: 1081-1093
- [11] Wang Yanwei, Wu Junjun, Chen Liping, et al. Identity propagation method for tracing alterations of a topological entity in a history-based solid modeling system[J]. Journal of Advanced Manufacturing Technology, 2005, 27(3/4): 305-312
- [12] Marcheix D. A persistent naming of shells[C]// Proc. Ninth International Conference on Computer Aided Design and Computer Graphics. Hong Kong, China, 2005: 259-265