

细胞元特征模型中操作局部化方法的研究

李素慧, 孙立镭

(哈尔滨理工大学 计算机科学与技术学院, 黑龙江 哈尔滨 150080)

摘 要:提出一种直接操作特征过程中将操作局部化的方法,该方法利用语义造型系统中特征依赖图及细胞表示特征的高效性,将发生变化的区域从特征模型中分离出来.利用该方法使得直接操作特征时缩小了对模型进行约束求解的范围,从而满足了直接操作特征时对约束求解的速度要求.同时,通过对数据结构的分析,给出两个相应的分离算法.

关键词:直接操作; 局部化; 细胞元表示; 特征依赖图; 细胞元变化模型

中图分类号: TP391 **文献标志码:** A **文章编号:** 1007-2683(2009)05-0047-05

Localizing Manipulation on the Cell-based Feature Models

LI Su-hui, SUN Li-juan

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

Abstract: In this paper, a method of localizing manipulation when directly manipulating features was presented. This method separated the changed cells from the Feature Model by making use of high efficiency of the cellular representation and Feature Depended Graph (FDG). And it reduced the scope of resolving constraints of the model when directly manipulating the feature. As a result, this method improved efficiency and satisfied the need of resolving to constraint. Meanwhile, two separate algorithms were given after analyzing the interrelated data structure.

Key words: direct manipulation; localizing; cellular representation; FDG; changed cellular model

0 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,每个特征定义了丰富的语义^[1].语义的定义是通过各种约束来实现的.但是目前,几乎所有的特征造型都是参数化的,基于历史的,系统需跟踪每一步模型操作,每个新特征都需要依据现存的边界模型进行定位,而最终的造型取决于边界模型中特征创建的顺序^[2],这种依存关系直接影响造型结

果,因为当进行特征操作时需要重新执行造型的全过程,模型中的特征数量越多,模型重构所耗费的时间就越大.因此当前的造型系统中,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.鉴于这种实现方法缺乏直观性和洞察力,本文提出了一种将操作局部化的方法,即:将对特征的某种操作局限在特定的一块特征区域中,从而通过缩小约束求解范围的方法提高约束求解的速度以达到直接操作特征的要求.

收稿日期: 2008-09-08

基金项目: 国家自然科学基金资助(60173055)

作者简介: 李素慧(1981—),女,硕士研究生, E-mail: 382066700@qq.com;

孙立镭(1944—),男,教授,博士生导师

1 操作局部化概念的提出

通过对话框的方式来修改特征的参数有下面两个缺陷:

1) 低效率的反馈. 设计者每次修改特征参数都不得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及修改过的模型, 换句话说, 对于哪部分特征发生了改变以及如何改变的缺乏精确反馈.

由于这些缺点, 用户只能被迫使用试验——错误的方法^[3]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两个缺点. 解决这两个问题最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. 因为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.

直接操作特征包括两个阶段: 选择阶段和交互阶段. 在选择阶段, 需要修改的参数被选定, 例如, 当圆柱体被选中时, 它的半径和高会在视图中显示出来. 在交互阶段, 设计者通过使用鼠标拖拉代表特征参数的图标来改变特征的参数值. 在拖拉的过程中, 通过不断的更新模型的数据结构和它的显示图形来反映修改的结果.

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^[4]. 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.

2 操作局部化可行性

将操作局部化的思想在基于细胞元的语义造型系统^[5]中是可行的, 这是因为:

1) 基于细胞元的语义特征模型是用语义约束表示的特征, 它不再单纯的依照特征创建的时间来建

模, 而是按照特征之间的相互关系来建模, 并根据这种相互依赖的关系将数据组织成一个特征依赖图 (FDG), 图中的结点表示产品特征和约束特征, 边表示了结点之间的依赖关系, 如图 1 中的 (a) 的 FDG 示意图为 (b). 这种数据结构把逻辑上不相关的数据分离开, 为把部分特征从模型中分离提供了基础.

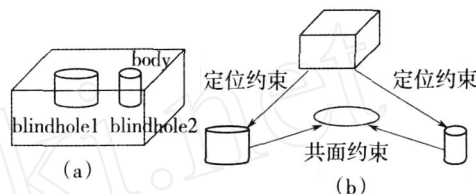


图 1 FDG 示意图

2) 本文通过对细胞元的深入研究得出如下细胞元的相关定义:

定义 1 一个细胞元的混合体 K 被定义为一个细胞元的有限的集合, 该集合满足下面条件: a) 如果 σ 是 K 中的一个细胞, 则 σ 的所有面都是 K 中的元素, b) 如果 σ 和 τ 都是 K 中的细胞, 则 σ 和 τ 没有交集.

定义 2 一个细胞元模型被定义为一个特殊的细胞元混合体, 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. 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 3 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.

从定义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. 形状变化可以以两种方式影响细胞模型中存在的细胞, 如图 2 所示.

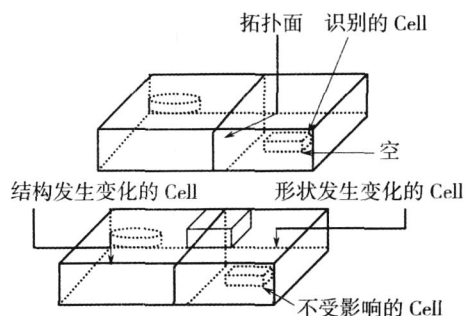


图 2 细胞元模型结构和形状变化图

1)通过向一个 3 - cell中增加或删除点集.

2)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时,也会导致其结构发生改变.

因此,细胞本身所具有的这些特性(尤其是定义 2 中细胞元的独立性)为把变化的细胞从特征中分离出来提供了基础.

3 操作局部化过程

3.1 局部化过程

将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 (Feature Region, FR)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 EDG 中分离出来,此时只是将操作局部化在特征级的范围内.

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 (Changed Cell Model, CCM). 此时操作局部化的范围就锁定在 CCM 上.

结合上述的分析,得出对一个对特征进行直接操作的过程模型,如图 3 所示.

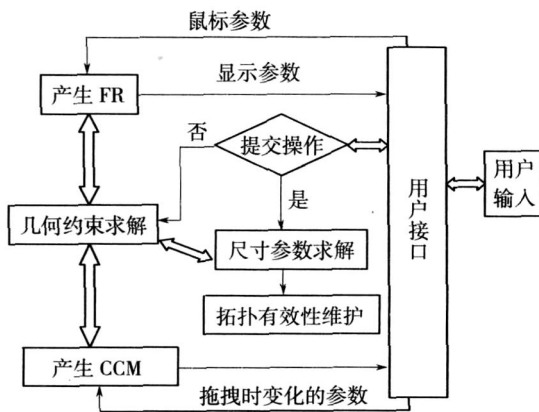


图 3 直接操作过程模型

当有鼠标输入的时候,根据鼠标输入对 EDG 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.

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.

3.2 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 ReMatrix[4][4]; // 特征句柄的参考矩阵
}

typedef struct Face { // 定义细胞面 t
    face_point[4]; // 一个细胞面由 4 个点来决定
    Listlist 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
    // 公共属性,如性质,形状,颜色等
    para1, para2, para3, ...;
    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
    Edge_Constrain // 边界约束
    Postion_Constrain // 定位约束
    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 EDG { // 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 算法描述

4.1 产生 FR

算法 Create FR 基本思想:该算法主要有两个过程:①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;②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:

1)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,则跳到 2).

2)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,则返回 1).

3)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,执行 4),重复该过程,直到遍历完了所有的客特征,跳到 5).

4)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 (CFL ist) 中,如果不在,则把该客特征存放到 CFL ist 中.跳到 4),否则继续遍历 pRelationHead,直到遍历完 pRelationHead 所有客特征以后,返回 3).

5)把产生的 CFL ist 中的特征加入到 FR 中.

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:①暂时的分割;②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.

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

分面的统一边的 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:a)找到 FR 的非实体的边界盒,b)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,c)重复步骤 b)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.

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 Selected-Feature

1)从 FR 中取出一个特征 (CurrentFeature),跳到 2),当处理完 FR 中所有的特征,跳转到 7).

2)取出 CurrentFeature 的 box 和 ReMatrix 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,直到取到的特征为是 SelectedFeature 为止,并把合并出来约束保存到 Constrailist 其目的是为了把 CurrentFeature 和 Selected-Feature 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.

3)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 ReMatrix 是否发生变化.如果 box 和 ReMatrix 没有变化,跳到 1);如果 ReMatrix 有变化,此时把该特征的所有细胞全部加入 CCM 中,然后跳转到 1);如果 ReMatrix 没有变化,而 box 有变化,跳到 4).

4)把 box 作为一个边界盒,并找出在 3)中 box 所变化的面 (ChangedFace),采用 5)来处理 box 和 ChangedFace 在处理完所有的 ChangedFaces 后,然后跳到 6).

5)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%.此

时返回 4),其目的是为了采用二分法进一步分割出受 SelectedFeature 影响的空间。

6)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,以后回跳到 1)。

7)结束

4.3 实现过程的分析

图 4 是选择阶段的一个高层的事件流程图。用户的选择导致特征所有参数图标(即所有显示在特征上的并在 Operator 控制器^[6]中注册过的参数图标)处于激活状态,突显的特征是利用第三部分提出的 Create FR 算法将操作定位之后分割出来的,以便设计者能够观察到参数修改的过程中模型是如何发生变化的。

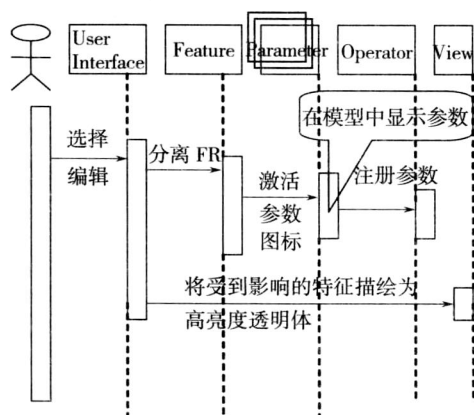


图 4 用户选择特征编辑时的流程图

图 5 是当一个拖拉事件发生时的流程图。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 Parameters 处理器发

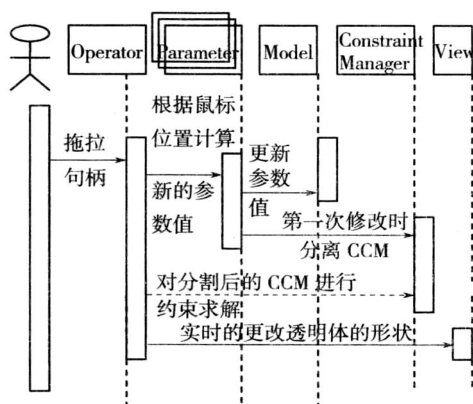


图 5 用户拖拉参数图标时的流程图

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第一次参数修改时产生的 CCM 进行求解。根据约束求解所得到的新的模型值来对模型进行渲染和显示。

5 结 语

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,所以整体上该方法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。

参 考 文 献:

- [1] BRONSVOORT W F, B DARRA R, NYRENDA P J. Developments in Feature Modeling [J]. Computer-Aided Design and Applications, 2006, 3(5): 655 - 664.
- [2] B DARRA R, BRONSVOORT W F. Semantic Feature Modeling [J]. Computer Aided Design, 2000, 32(3): 201 - 225.
- [3] RAFAEL Bidarra, ANDRÉ van Bunnik, W LLEM F. Bronsvort, Direct Manipulation of Feature Models in Web-based Collaborative Design [C]//ASME Design Engineering Technical Conferences September 28 - October 2, 2004: 30 - 34.
- [4] DAN IEL Lourenço, PEDRO Oliveira, Alex Noort, Rafael Bidarra. Constraint Solving for Direct Manipulation of Features [J]. Journal of Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 2006, 20(4): 369 - 382.
- [5] RAFAEL Bidarra, KLAS Jan de KRAKER, W LLEM F. Bronsvort, Representation and Management of Feature Information in a Cellular Model [J]. Computer-Aided Design, 1998, 30(4): 301 - 313.
- [6] RAFAEL Bidarra, ALEX Noort. Advanced Direct Manipulation of Feature Models [C]//Proceedings of the 2nd International Conference on Computer Graphics Theory and Applications-GRAPP 07, Barcelona, Spain, March 8 - 11, 2007: 27 - 33.

(编辑:王 萍)