对象族特征模型几何约束求解研究

刘宪国[1,2] 孙立镌[1]
(哈尔滨理工大学计算机科学与技术学院 哈尔滨 150080)[1]
(辽宁工程技术大学软件学院 葫芦岛 125105)[2]

摘 要 提出一种求解对象族模型的新的几何求解方法。提出两种新类型的组,即可伸缩组和可放射组。在刚性组或非刚性组系统中穷举地使用重写规则的较小的集合,一直到没有可用的重写规则为止,最后的组的集合就表示系统的求解策略。提出并实现一种增量算法,以及在这种新的求解方法中的解选择方法,这些方法都可以高效地找到问题的解,并减少解的个数。

关键词 对象族,几何约束,求解规则,增量算法,解选择策略

中图法分类号 TP391 **文献标识码** A

Research on Geometric Constraint Solving in Family of Objects Feature Model

LIU Xian-guo[1,2] SUN Li-juan[1]
(Department of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China)[1]
(School of Software, Liaoning Technology University, Huludao 125105, China)[2]

Abstract A novel method of solving geometric constraints of family of object models was imposed, and two new type clusters were presented, namely scalable subset and radial subset. Smaller collections of rewriting rule were used exhaustively, and were applied in rigid or non-rigid cluster systems, until there is no available rewrite rules so far, and the final cluster collections represent solution strategy of constraint system. A incremental algorithm and solution selection strategy in the solving method were imposed and implemented. Solutions of constraint problem are found efficiently in this method, and the number of solutions will be reduced.

Keywords Family of object, Geometric constraint, Solving rule, Incremental algorithm, Solving selection strategy

1 前言

在现在的 CAD 系统中,通过使用几何约束详细指定二维草图中的尺寸,并在三维装配建模中定位部件。在一些陈述式建模方法中,如在语义特征造型[1]和陈述式对象族造型[2]中,通过几何约束定义特征类和对象族。

在陈述式对象族模型中,通过几何约束说明对象族的几何属性。为了得出一个陈述式对象族模型的实现,确定对象族的族成员,需要求解这种系统的几何约束。这种几何约束求解器需要满足下面 4 个要求:

(1)应该能够确定施加在载体上的几何约束系统的解,特别是它必须能够求解点、线、面、球体和圆柱体上距离、角度等约束。

(2)应该能够确定一个约束系统是完备约束、欠约束还是过约束系统,并且该系统是稳定的还是不稳定的。

(3)应该能够计算出完备系统的所有解,或者某一个特定的解。

(4)应该能够具有足够高的人机交互建模效率,能高效地计算出参数区间,特别是在系统增加约束后,能高效地对解进行重新计算。

在一个陈述式对象族模型中有距离变量和角度变量,在这些变量上有其它类型约束,如代数约束等。本文提出的几何约束求解器不能求解这种混合类型约束的系统,因此我们假设几何约束的所有变量都是几何变量,在运行求解器之前,所有几何约束的参数值都是确定的。

现有的最成功的几何约束求解是构造性求解器[3],该求解器一般先将问题分解为很多刚性子问题,再对这些子问题分别求解,利用这些子问题的解构造问题的全局解。

对于二维问题来说,所谓问题的刚性是该问题符合 Laman 定理[4]。这个定理以图的形式说明刚性特性,图中的边对应于距离约束,图中的顶点对应于点变量。

Laman 定理 如果图 G 有 $2n-3$ 条边,其中 n 是顶点数。那么图 G 在 R^2 空间是刚性的,对每个子图来说,当且仅当 $e'\leqslant 2n'-3$,其中 n' 是顶点数,e' 是边数。

Laman 规则说明,对于有 n 个点和 e 条边约束的二维约束问题来说,当且仅当 $e=2n-3$ 时该问题是刚性的,或者说是完备约束的;对于具有 n' 个点和 e' 条边的子问题来说,当且仅当 $e'\leqslant 2n'-3$ 时,该子问题是完备约束的。如果约束的数

到稿日期:2011-05-17 返修日期:2011-08-19 本文受国家自然科学基金(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:qdlxg5639@163.com;孙立镌(1944—),男,教授,博士生导师,主要研究方向为计算机图形学与 CAD。

量小于定理中规定的数量，该系统是欠约束的；相反，如果多于定理中规定的数量，该系统是过约束的。这里需要特别指出的是，一个完备约束的系统不一定是稳定的，这取决于约束的实际参数值。过约束系统也不一定是不稳定的，因为对于某些约束参数值来说系统可能有多个解，这种系统我们称为稳定的过约束。

最简单的构造性求解方法是组重写方法[5]，也是一种自底向上的方法。这里的组对应于每个子问题，组重写方法也就是子问题重写方法。在该方法中，将具有刚性特性的几何约束看作一个组，识别某些组的特性，将较小的组组合成较大的组。如果该问题是完备约束的，在最后的重写过程中就只剩下唯一一个组。

大部分二维求解器使用多种组重写方法优化表示系统组的数据结构，比如在文献[6]中提出的图构造方法。这些算法表明，如果将一个系统简化成一个单独的组，那么对于原问题来说这个组就表示该问题的一个恰当的解。

现在这种组重写方法还不完善，因为还没有一个完整的重写规则集合用来分解所有的完备约束系统，甚至对于只有尺寸约束组成的系统也不能分解成一个单独的刚性组。在实际应用中，这就意味着一些完备约束系统不能通过求解器求解，也可能会发生将完备约束系统识别成欠约束系统的情况。

根据 Laman 的理论，可以为二维问题设计一种完全是自顶向下的分解算法。然而这种算法的复杂度比较高，而且不能处理稳定的过约束的情况。

在组重写方法的基础上，研究者们也提出了一些三维求解器。然而在三维情况下出现了比二维问题更多的不完整的分解情况。至今还没有人提出满足 Laman 定理的三维求解器。但是对于距离约束的三维系统来说，Sitharam 提出一种具有一般刚性特性的描述，这种方法可能会为三维几何约束系统的求解提供一个发展方向，然而现在还没有一个在此基础上发展起来的求解方法。

很多求解方法都是在自由度分析方法[7]的基础上提出的。自由度分析方法通过一种启发式规则建立一种自顶向下的分解过程，最终确定一个约束问题及其子问题的一般刚性特性，这种求解方法在自由度分析结果的基础上试图找到一个刚性子问题的最小集合。在实际应用中，基于自由度的规则可以为很多约束系统非常准确地确定完备约束情况。然而基于自由度的方法也不能完全解决所有问题，因为现在的基于自由度的方法是基于完备约束的启发式技术。在三维问题中，基于自由度的算法有时候会将过约束系统识别为完备约束系统。

为了能够有效地确定与增量改变有关的约束问题的解决方案，可以使用组重写算法。这种算法速度比较快，而且实现增量算法也比较简单。但是组重写算法只能用在相对较小的问题求解上，特别是应用在可以将问题分解为很小规模的刚性子问题的约束问题中。而基于自由度分析的方法可以求解较大规模的问题，但它的效率比较低，而且没有增量算法。特别是当一个问题不能分解为多个子问题的时候，整个问题必须使用比较低效的符号代数方法求解，而这种方法也是在增加约束后不能使用。

本文提出一种求解对象族模型的新的几何求解方法。我们结合组重写方法和自由度分析方法的优点，提出两种新类型的组，即可伸缩组和可放射组。这种方法不仅能识别刚性子问题，而且能识别只能用自由度方法求解的子问题。

这种方法的基本思想是在刚性组或非刚性组系统中穷举地使用重写规则的较小的集合，一直到没有可用的重写规则为止，这个最后的组的集合就表示系统的求解策略。这样就可以确定系统是完备约束的、欠约束的还是过约束的。

本文还提出了一种增量算法，以及在这种新的求解方法中的策略选择方法，这些方法都可以高效地找到问题的解，并减少解的个数。

2 对象族特征

对象族的概念在不同的应用领域有不同的定义，因此对于对象族模型必须尽可能全面和允许定义某些特定的语义。在机械和工业设计应用上，形状和功能是建模领域最重要的对象，这两个方面都与模型的几何属性和拓扑属性有密切关系。

任意一种陈述式模型的基本元素都是变量和约束。因为陈述式对象族模型是一种陈述式模型，所以该模型的基本元素也是变量和约束。一个对象族内部成员之间的差异通过变量标识，即族成员的几何性质通过几何变量标识，族成员的拓扑性质通过拓扑变量标识。为了明确说明特征和作为一个整体的对象族的不变属性，向一个或多个变量上施加相应的约束，无论是同一类型的约束还是不同类型的约束，比如几何与拓扑性质是由几何与拓扑约束详细说明的。

陈述式对象族模型是语义特征模型的扩展。在陈述式对象族模型中，模型的几何和拓扑不必全部指定。一般情况下，一个陈述式对象族模型表示一个对象族。通过求解拓扑约束确定族成员。

由陈述式对象族模型表示的对象族由命名为载体的几何变量和命名为构造体的拓扑变量组成[8]。载体用来定义分割空间的曲面，如平面载体定义一个平面和该平面的两个侧面；构造体基本上表示点的集合，由载体定义的子空间的相交构造，如体特征、曲面特征、曲线和各个分离的点；通过子空间约束建立载体和构造体之间的关系。对象族的几何属性通过载体上的几何约束标识，拓扑属性通过载体上的拓扑约束标识。这种表示方式可以使系统详细说明几乎任意一个对象族。文献[9]中详细说明了这种陈述式表示的几何和拓扑。

变量和约束频繁地结合，可以合并成特征。这种特征可以更高层次上的抽象提供语义，使其更接近对象族的功能。在陈述式对象族模型中，本文定义了一种特征作为变量和约束的子集。针对体对象建模问题，每个特征必须包括一个变量表示该体对象。

一个陈述式对象族模型定义一个实现的集合，即所有可能的满足约束的对象模型，以此表示所有的族成员。这种实现的几何表示是细胞元模型[10]，该细胞元模型也用在语义特征模型的几何表示中。利用这种表示方式，建模系统可以很方便地建立对象族的几何和拓扑与实现的几何和拓扑之间的关系。

为了确定实现，首先要通过求解几何约束确定实现的细胞元模型的最小单元的形态；然后通过求解拓扑约束确定哪些最小单元包含材料，也就是哪些最小单元是实特征，哪些最小单元是空特征。求解拓扑约束代替语义特征模型中的特征依赖分析过程，因为通过求解拓扑约束可以确定所有可能的实现，而且可以清楚地说明陈述式对象族模型。

一个陈述式对象族模型可以有零个、一个或有限个由细胞元模型表示的实现。这些实现对应于族成员。然而，根据

陈述式对象族模型可以更好地定义族成员，将一个对象族成员定义为具有准确的一个实现的陈述式对象族模型。通过向一个陈述式对象族模型中增加约束表示一个更大的族，比如增加变量值这个约束。同样，也可以定义一个给定的族的子族作为一个陈述式对象族模型。因此在族成员之间就定义了一种完备的相关关系，这种关系可以通过比较陈述式对象族模型进行测试。这种方式与比较实现的几何表示相比更容易，因为比较几何表示时很难保证一致性命名。

3 求解规则

为了求解这种组特征的系统，本文利用一个重写规则集合将系统重写成一个单独的刚性组。

可以将重写规则看作是一种模式，在这种模式中有输入组，也有输出组，除此之外还有根据输入组的配置确定输出组的配置的过程。如果在该模式中找到一个符合输入组样式的组的集合，就应用一个重写规则，向系统中增加相应的输出组，并且这个过程确定了输出组的配置。

一个模式指定了很多给定类型的输入组和很多模式变量，通过求解算法使这些模型变量与系统中组的点变量相匹配，这样就会使变量数和组的类型相匹配。一个模式也可能会指定一个输入组来匹配任意一个给定变量的组。如果在一个模式中一个变量的名字多次出现，那这个变量一定是由几个组施加约束的点变量。

为了确定一个重写规则输出组的配置，需要对每个输入组的配置应用规则。假如一个重写规则的输入有两个组，而每个组都有两个配置，那么对于输出组来说需要计算 4 个不同的配置。

针对二维问题，本节提出 3 个重写规则，下面详细阐述。

如果 $a_1, a_2, a_3, \cdots, a_n \in R^2$，

规则 1

$\mathrm{RADI}(a_1, [a_3, a_2, \cdots]) \cup \mathrm{RADI}(a_2 [a_1, a_3, \cdots]) \rightarrow \mathrm{SCLB}([a_1, a_2, a_3])$

当两个可放射组（A 和 B）共享 3 个点，这 3 个点中包含每个组的中心点时应用此规则。找到一个匹配后，将一个新的可放射组（R）添加到系统中，并且计算组的配置 C_R。

规则 2

$\mathrm{SCLB}(A=[a_1, a_2, \cdots]) \cup \mathrm{SCLB}(B=[a_1, a_2, \cdots]) \rightarrow \mathrm{SCLB}(A \cup B)$

两个可伸缩组如果共享两个点，可以将其合并为一个新的可伸缩组，这个规则可以重复使用。

规则 3

$\mathrm{SCLB}(A=[a_1, a_2, \cdots]) \cap \mathrm{SCLB}([a_1, a_2, \cdots]) \rightarrow \mathrm{RIGD}(A)$

将此规则应用到系统中，会产生一个刚性组，所有的变量都有约束，而且没有其它的重写规则可用。

与策略组相关的配置是问题的特定解。如果不存在与策略组相关的配置，那么该问题是不一致的；如果存在一个或多个这样的配置，那么问题是一致的。

根据问题的一般策略，我们可以确定系统是欠约束的、过约束的还是完备约束的。

(1) 如果问题的一般策略有多个策略组或者有一个非刚性的策略组，那么该问题是欠约束的；

(2) 如果任何一个尺寸约束或角度约束有多个源组，那么该问题是过约束的。这种情况在下面详细阐述。

(3) 如果不属于欠约束情况和过约束情况，则该问题是完备约束的。

需要说明的是这些条件不是互斥的。

在一般策略中，如果在两个或多个组中存在尺寸约束或角度约束，系统可能是过约束的，这取决于这些尺寸和角度存在于哪个组中。尤其是，两个或多个问题组包含同一个尺寸或角度，那么这些尺寸和角度通常是不一致的，因此这种问题是过约束的。

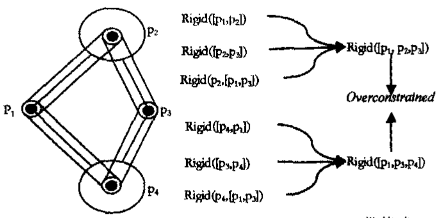
如果在几个中间组或策略组中存在一些尺寸约束或角度约束，那么这种问题不一定是过约束的。通过对原始问题组执行重写规则，推导出不同组中的尺寸或角度的具体值，这些变量可能在所有的组中都有相同的值。

本文通过下面的步骤来确定一个系统是否是过约束的。当向一般策略中增加新组时，对那个组中的每个尺寸或角度都要确定其源组（只有输出、没有输入的组），也即一般策略中的存在尺寸或角度约束的第一个组。

在一些组中可以通过一般策略中依赖的反方向追踪尺寸或角度的源组，检查每一个遇到的组是否包含尺寸或角度约束。如果一个组不依赖任何一个包含尺寸或角度约束的组，那么这个组就是源组。

对于每个尺寸或角度来说，如果正好有一个源组，那么这样的系统不是过约束的，因为每个重写规则可以保证所有的在输入组中的尺寸或角度约束在输出组中也满足；另外，如果有多个源组，那么不能保证在不同的组中有相同的数值，这样的系统就是过约束的。

图 1(a)所示是一个过约束系统，该问题的一般策略如图 1(b)所示。在这个例子中，组 RIGI($[a_1, a_2, a_3]$)和组 RIGI($[a_1, a_3, a_4]$)都有尺寸约束 $\delta(a_1, a_3)$。所以这个尺寸计算了两次，但是使用的是不同的输入组，应用的是不同的重写规则。因此，对于这个尺寸来说，就有两个源组 RIGI($[a_1, a_2, a_3]$)和 RIGI($[a_1, a_3, a_4]$)，所以这个问题是过约束的。



(a)问题　(b)一般策略

图 1　二维过约束问题及其一般策略

4 增量算法

当问题的约束改变时，本文使用增量求解算法更新约束问题的一般策略。这种约束的改变包括增加组或者删除组操作。通过一个二叉图表示问题的一般策略（在算法 1、2、3 中用符号 G 表示），在二叉图中节点表示组或重写规则，有向边连接组和重写规则。即如果一个组是一个重写规则的输入组，那么就有一条边从这个组指向该重写规则；如果一个组是一个重写规则的输出组，那么就有一条边从重写规则指向该输出组。

本节提出的增量算法也记录已经激活的组的集合，即所有的重写规则计算过的组（在算法 1、2、3 中用符号 A 表示）。

问题的一般策略和激活组集合初始化是空的。当用户向问题中增加一个组时（见算法 1 中的 AddCluster 函数），这个组就添加到一般策略中，成为激活组。因为它不是重写规则

的输出，在一般策略中这个组被识别为问题组。算法然后搜索应用于新组的所有可能的重写规则，即当一个组作为输入时就应用重写规则(见算法 3 中的 SearchRewrites 函数)。

当删除一个组时(见算法 2 中的 RemoveCluster 函数)，算法将这个组在一般策略和激活集合中删除，所有的从属组也从一般策略中删除。从属组(见 DependentClusters 函数)是在一般策略中所有由以给定组作为输入的重写规则直接或间接确定的组。然后算法需要确定激活组的一个新集合。为此，当添加特定组时需要确定哪些组必须从激活集合中删除(见 DeactivatedClusters 函数)，通过向激活集合中重写增加的这些组来更新激活集合。在更新了激活集合后，可能会出现这种情况，即需要重写由多个组组合成的组(见算法 3 中的 SearchRewrites 函数)。

算法 1　增加组函数

```
Function AddCluster(G,A,c)
    G:generic solution
    A:active set
    c:cluster
begin
    G.add(c)
    A.add(c)
    SearchRewrites(G,A,c)
end
```

算法 2　删除一个组

```
Function RemoveCluster(G,A,c)
    G:generic solution
    A:active set
    c:cluster
begin
    A.remove(c)
    G.remove(c)(* also removes rewrite rules on c *)
    for each x in DependentClusters(G,c)
        RemoveCluster(x)
    for each y in DeactivatedClusters(G,c)
        A.add(y)
        SearchRewrites(G,A,y)
end
```

本算法可以高效地搜索所有可能应用的重写规则，因为算法只为新增加的组搜索重写规则(见算法 3)。由于每个重写规则涉及很少数量的重合组(共享一个或多个点变量的组)，算法构造由新增加的组和与新组有重合的组组成的激活组集合的一个子集(见 OverlappingClusters 函数)，在这个子集中搜索所有可能应用的重写规则。这样，搜索算法只检索很少数量的组和变量。

在增量算法中，使用的模式匹配算法是基于一个子图的匹配算法，该匹配算法可以找到所有同构的子图。通过将一个重写规则转化为一个图作为算法的输入模式(见 PatternGraph 函数)，激活集合中所有应用过重写规则的子集也转化为一个图(见 ReferenceGraph 函数)。对于通过运行图匹配算法返回的每个同构子图，算法可以确定将哪些点变量分配给模式变量，实例化了哪些重写规则，向一般策略中增加了哪些重写规则。

算法 3　搜索重写规则应用

```
Function SearchRewrites(G,A,c)
    G:generic solution
    A:active set
    c:cluster
begin
    subset:= c+OverlappingClusters(A,c)
    reference:= ReferenceGraph(subset)
    for each rule in AllRewriteRules
        pattern:= PatternGraph(rule)
        matches:= SubgraphIsomorphisms(pattern,reference)
    for each match in matches
        rewrite:= instantiate rule from match
        if IsProgressive(rewrite) then
            G.add(rewrite)
            A.add(rewrite.output)
            for each i in rewrite.inputs
                if IsRedundant(i) then
                    A.remove(i)
            SearchRewrites(rewrite.output)
end
```

对于每个可能应用的重写规则，算法首先检查重写规则应用是否是渐进的(见 IsProgressive 函数)。如果是，算法将该重写规则添加到一般策略中。这里规定一个重写规则应用是渐进的，是指重写规则应用或者增加由激活集合约束的尺寸和角度的数量，或者减少激活组的数量。这种机制可以保证算法不向系统中增加多余的组，也保证不删除过约束组。

当向一般策略中增加一个重写规则时，输出组就会成为激活组集合的一部分，并且会在激活集合中删除一个或多个输入组。如果一个组是多余的，就把它在激活集合中删除，即在激活集合中已经由其它组约束所有的尺寸和角度的组就是多余组，一定要删除这种多余的组。

为了确定一个组是否是多余的(见 IsRedundant 函数)，算法需要确定由组约束的尺寸和角度的集合是否是由其它组约束的尺寸和角度约束的集合的子集。准确地确定这些集合是很费时的，所以本节通过确定激活集合中由组约束的尺寸和角度的数量和由每个其它有重合的组约束的尺寸和角度的数量实现。

表 1　成对的组的交集

组的交集	条件
RIDI(A)∩RIDI(B)=RIDI(A∩B)	|A∩B|>1
RADI(A)∩SCLB(B)=SCLB(A∩B)	|A∩B|>2
RADI(A)∩RADI(p_c,B)=RADI(p_c,A∩B)	p_c∈A,|A∩B|>2
SCLB(A)∩SCLB(p_c,B)=SCLB(A∩B)	|A∩B|>2
SCLB(A)∩RADI(p_c,B)=RADI(p_c,A∩B)	p_c∈A,|A∩B|>2
RADI(p_c,A)∩RADI(p_c,B)=RADI(p_c,A∩B)	|A∩B|>2

本节将两个组的交集也定义为一个组，即由两个组共同约束的尺寸和角度约束的集合。确定这个交集可以通过表 1 的规则快速地确定。例如，给出两个刚性组 RIGI([a_1,a_2,a_3])和 RIGI([a_2,a_3,a_4])，通过表 1 中的第一个规则可以确定这两个组的交集是 RIGI([a_2,a_3])。表 1 中也给出了如何根据起始组的类型确定交集组的类型，由交集组约束的点变量的集合是由起始组共享的点变量的集合。这种共享的点的集合必须满足几个附加条件来保证交集组是一个有效组，即需要给出共享点的最小个数和两个可放射性组的两个中心点必须是同一个点。

一个组约束的尺寸和角度的数量通过表 2 所列可以得出。需要指出的是，一个组约束的尺寸和角度的数量通常都多于一个完备约束系统所需要的约束的数量。通过一个配置确定一个组约束的尺寸和角度数值的组合，这些数值总是一

致的。

表 2 由组约束的尺寸和角度的数量

组	尺寸	角度
RIGI([$p_1,p_2,\cdots,p_3$])	$\binom{n}{2}$	$3\binom{n}{3}$
SCLB([$p_1,p_2,\cdots,p_3$])	0	$3\binom{n}{3}$
RADI(p_C,[$p_1,p_2,\cdots p_3$])	0	$\binom{n}{2}$

如果一个组约束的尺寸和角度的数量多于激活集合中每个重叠组构成交集组约束的尺寸和角度的总数量,那么这个组不是多余的;如果一个组中尺寸和约束的数量等于交集组中尺寸和角度的总数量,那么这个组是多余的;不存在少于的情况。

一个问题的一般策略可以用来确定问题的特定解。对于输入组配置的每个组合,通过执行一般策略中部分重写规则的运算确定特定解,也可以通过增量的方法实现。当与问题组相关的配置的集合改变时,从一般策略中确定依赖的重写规则,即确定将这个组作为输入组的规则。只有这些重写规则才需要进行重新计算,这样可以提高系统计算的效率。

需要指出的是,对每个输入组配置的组合需要重复计算特定的解,而每个重写规则又反过来为每个这样的组合产生几个策略,这就会导致特定解的数量比较多,而且计算起来比较复杂。但是对于大部分应用来说,不是所有的解都是需要的,这就面临选择哪个解的问题。所以需要一种解的选择机制来降低解的个数,减少计算的时间。在下一节给出这种解的选择策略。

5 解的选择策略

一般来说,一个几何约束问题有几个解,这时候在几何约束求解器中必须解决的一个问题是解的选择问题,即多解问题。

Bettig 和 Shah 给出了多个解选择方法的综述[11]。作者认为陈述式解选择器是最易用、功能最强大的方法。在这种方法中,作者指定附加约束(即解选择器),将解的个数减少到只剩下唯一的潜在解。

像组的定义一样,在点变量的集合上定义选择约束。但是这些约束不像组,它们可以详细说明点之间应该满足的任意一种关系。唯一需要的是给定这些点的配置,然后计算这些点是否满足这些约束。但是,只能用来检查这些约束是否保证问题是完备约束的,而不能用来确定系统中的任何一个自由度。

在二维模型中,如图 2 所示,通过 3 个点的集合{a_1,a_2,a_3}是否是顺时针或逆时针,就可以选择出一个满足条件的解。

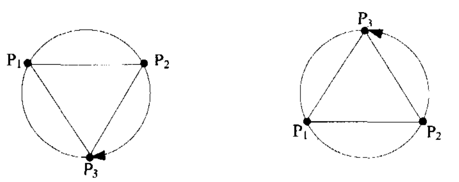


图 2 解选择约束

如果需要在 R^n 空间确定 $n+1$ 个点集合的方向,只需要选取一个参考点,计算其它点的偏移向量集合的行列式即可。本节定义点集合{$a_1,a_2,\cdots,a_{n+1}$}的方向为

$$Orientation(a_1,a_2,\cdots,a_{n+1})=\mathrm{sign}(Det[a_2-a_1,\cdots,a_{n+1}-a_1])$$

如果行列式是正的,那么点集合就是正方向;如果行列式是负的,那么点集合就是负方向。可以通过行矩阵计算行列式的值,也可以通过列矩阵计算行列式的值。

因为一个几何约束问题的解的个数可能会很多,并且计算所有的解又比较费时,所以应该尽可能早地计算解选择,即一达到计算解约束的条件就开始计算。一般来说,一旦确定了一个刚性组,就计算解选择。

对于陈述式解选择方法还存在两个问题,第一个问题是当最坏的情况发生,即需要计算几何约束问题的所有解的情况。解的个数通常是几何元素的指数倍,这时候算法就会是一个 NP-完全问题。第二个问题是为了确定一个解而需要确定很大数量的选择约束的情况。根据问题的参数,这个解经常会是不连续的。对于不连续的解,系统不能对相关约束进行参数化,这也会导致特征模型的异常。

本节提出一个新的基于原型的解选择方法,确定至多一个解,称之为预期解。预期解需要满足下面两个性质:

性质 1 预期解是问题的参数的一个连续函数。

性质 2 预期解唯一匹配给定的原型。

第一个性质保证在问题的参数和预期解之间存在一种直观的关系,第二个性质保证在预期解和原型(用户创建的模型的草图称为模型的原型)之间存在一个直观的关系。这样用户就可以通过原型控制预期解。在预期解和原型之间建立唯一的匹配,意味着对同样的参数值来说发现的其他解都不能与原型匹配,这需要根据匹配定义确定。这样,就可以根据原型模型确定唯一的预期解。

匹配的定义是预期解与原型之间的一种等价关系,这种等价类将配置空间分割成多个等价类。预期解就是与原型在同一个等价类中的解,如图 3 所示。

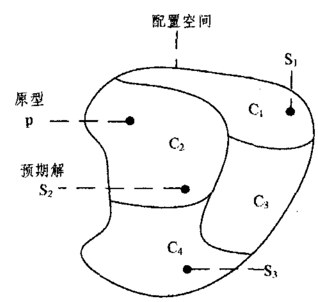


图 3 配置空间分解为多个等价类 C_i 示意图

对于一些参数和原型的组合来说,即使存在一个真实的解,也可能不存在一个预期解。例如图 4 所示的二维系统,对任何一个参数值 $d\geqslant 3$ 时,解是参数 d 的连续函数。但是如果将问题实例化为 $d=2$,那么在同一个等价类中就没有解,即当 $d=2$ 时无解。

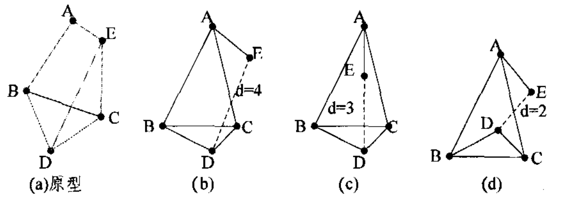


图 4 一个原型 a)和尺寸参数 d 构成的约束问题

当在陈述式对象族定义中使用预期解时,几何约束系统的所有解就是一个较小的对象族。使用预期解施加额外的约束,一个陈述式对象族模型有几个实现。预期解可以保证,如果一个几何系统是完备约束的,那么将会找到至多一个预期解,但根据拓扑约束仍然可以找到多个实现。尽管预期解会根据几何约束的连续性有所变化,但是仍然可以跟踪陈述式对象族模型实现中的拓扑变化。关于拓扑变化的详细内容在文献[12]中有详细阐述。

使用前面提出的组重写算法可以找到预期解。对于每个需要求解的子问题,最基本的工作是产生选择约束,这样对于每个子问题,可以选择一个解。对于特定的子问题,产生的选择约束取决于子问题的类型和原型。因为每个子问题至多需要找到一个解,不需要回朔搜索,所以这种计算预期解的方法也不费时。

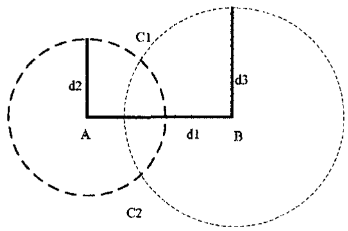


图 5　一个简单的三角形子问题

再考虑图 4 所示的二维系统,该问题可以分解成 3 个简单的三角形问题,即 ΔABC、ΔBCD、ΔADE。求解每个子问题,例如图 5 所示两个圆的交集确定子问题,可以通过求解交点的顺序来区分两个不同的解,即逆时针方向的 ΔABC_1 和顺时针方向的 ΔABC_2。如果在点 ABC 的原型中是顺时针方向的,那么将选择约束 *Clockwise*(A,B,C)添加到系统中;反之,如果是逆时针方向的,将选择约束 *CounterClockwise*(A,B,C)添加到系统中。如果既不是顺时针也不是逆时针的,原型中的点就是共线,这时候需要通知用户。图 5 所示问题的另外一个子问题通过同样的过程求解。

结束语　本文引入两个非刚性组特征的类型,即可伸缩组特征和可放射组特征,与传统的刚性组特征一起,使用这 3 个类型的组特征求解几何约束问题。使用一个简单的组重写方法,求解只有一个刚性组特征类型的大型约束问题。还提出了一种增量算法,以及在这种新的求解方法中的策略选择方法,它们都可以高效地找到问题的解,并减少解的个数。

参 考 文 献

[1] Bidarra R, Bronsvoort W F. Semantic feature modeling[J]. Computer-Aided Design, 2000, 32(3): 201-225

[2] Bronsvoort W F, Bidarra R, Meiden H A V D, et al. The increasing role of semantics in object modeling[J]. Computer-Aided Design and Applications, 2010, 7(3): 431-440

[3] Raghothama S. Constructive Topological Representations[C]// Proceedings ACM Symposium on Solid and Physical Modeling. Cardiff: ACM Press, 2006: 39-51

[4] Laman G. On graphs and rigidity of plane skeletal structures [J]. Journal of Engineering Mathematics, 1970, 4(4): 331-340

[5] Durand C, Hoffmann M. A systematic framework for solving geometric constraints analytically[J]. Journal of Symbolic Computation, 2000, 30(5): 493-519

[6] Durand C, Hoffmann C M. A systematic framework for solving geometric constraints analytically[J]. Journal of Symbolic Computation, 2000, 30(5): 495-519

[7] 戴春来. 基于自由度分析的耦合几何约束求解[J]. 计算机辅助设计与图形学学报, 2010, 22(12): 2168-2176

[8] 赵伟. 自由形状特征的重用与抑制[D]. 杭州: 浙江大学, 2008: 13-17

[9] 刘宪国, 孙立镌, 王其华, 等. 语义特征造型中对象族模型研究[J]. 计算机科学, 2011, 38(3): 286-289

[10] Bidarra R, Kraker K J D, Bronsvoort W F. Representation and management of feature information in a cellular model[J]. Computer-Aided Design, 1998, 30(4): 301-313

[11] Bettig B, Shah J. Solution selectors: a user-oriented answer to the multiple solution problem in constraint solving[J]. Jounal of Mechanical Design, 2003, 125(3): 443-451

[12] 孙立镌, 刘宪国, 于春风, 等. 跟踪对象族模型拓扑结构变化研究[J]. 计算机科学, 2010, 37(12): 255-258

(上接第 253 页)

[5] Kim J, André E. Emotion recognition based on physiological changes in music listening[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, 30(12): 2067-2083

[6] 张俊利, 蔺嫦燕. 脉搏波波形特征信息检测及与部分血流动力学变化相关分析[J]. 生物医学工程与临床, 2008, 12(2)

[7] 葛臣. 脉搏信号在情感状态识别中的研究[D]. 重庆: 西南大学, 2010

[8] LI Yun, WU Zhong-fu, LIU Jia-min, et al. Efficient feature selection for high-dimensional data using two-level filter[C]// Proc of Int Conf on Machine Learning and Cybernetics. Shanghai: IEEE, Aug. 2004: 1711-1716

[9] Isabelle G, André E. An Introduction to variable and feature selection[J]. Journal of Machine Learning Research, 2003, 3: 1157-1182

[10] 毛勇, 周晓波. 特征选择算法研究综述[J]. 模式识别与人工智能, 2007, 20(2): 211-218

[11] Mitra P, Mutthy C A, Pal S K. Unsupervised feature selection using feature similarity[J]. IEEE: Transactions on Pattern Recognition and Machine Intelligence, 2002, 24(3): 301-312

[12] 陈希孺. 概率论与数理统计[M]. 合肥: 中国科学技术大学出版社, 2009: 123-129

[13] 盛骤, 谢式千, 潘承毅. 概率论与数理统计(第三版)[M]. 北京: 高等教育出版社, 2001: 129-133

[14] Dorigo M, Stützle T. Ant Colony Optimization[M]. MIT Press, 2004: 64-116

[15] Ma Chang-wei, Liu Guang-yuan. Feature Extraction, Feature Selection and Classification from Electrocardiography to Emotions [C]// Proc. Int. Conf. Computational Intelligence and Natural Computing. Wuhan, June 2009: 190-193

[16] Pan Li, Zheng Hong. Genetic Feature Selection for Texture Classification [J]. Geo-spatial Information Science, 2004, 7(3): 162-166

[17] Peng Han-chuan, Long Fu-hui, Ding C. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(8): 1226-1238

[18] Nguyen M H, de la Torre F. Optimal feature selection for support vector machines[J]. Pattern Recognition, 2010, 43: 584-591