

几何约束系统中参数有效范围的算法研究

张杏莉¹, 李冰清², 王红娟¹

(1. 山东科技大学 信息科学与工程学院, 山东 青岛 266590; 2. 中国电信集团公司 山西分公司, 山西 太原 030006)

摘要: 针对几何约束系统设计过程中经常遇到的参数驱动失败问题, 提出参数有效取值范围的求解算法。以包含距离约束和角度约束的二维尺规构造几何约束系统为例, 分析并总结所有可能的六种三角形刚体类型。识别与所求参数无关的刚体, 通过刚体变换算法减小几何约束系统的求解规模。应用实例表明, 刚体变换算法可以有效简化几何约束系统, 提高算法求解效率。

关键词: 参数有效范围; 刚体变换; 参数驱动; 三角形刚体; 几何约束系统

中图分类号: TP391

文献标志码: A

文章编号: 1672-3767(2015)03-0014-06

Algorithm of Valid Range of Parameters in Geometric Constraint System

Zhang Xingli¹, Li Bingqing², Wang Hongjuan¹

(1. College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao, Shandong 266590, China; 2. Shanxi Branch of China Telecom Group Corporation, Taiyuan, Shanxi 030006, China)

Abstract: To solve the problem of parameter driving failure frequently encountered in the design of geometric constraint system, an algorithm for determining the valid range of parameter values was proposed. Taking the 2D geometric constraint system with only distance constraints and angle constraints as example, this paper analyzed and summarized all the possible six types of triangular rigid bodies. By recognizing rigid bodies irrelevant to the parameter, the rigid transform algorithm decreased the computation size of the geometric constraint system. The example shows that the rigid transform algorithm effectively simplifies the geometric constraint system and improves the solving efficiency.

Key words: valid parameter value; rigid transform; parameter driving; triangular rigid; geometric constraint system

参数驱动是参数化设计的核心技术, 其中, 修改几何约束参数或尺寸标注的值并重新生成所需图形是参数化绘图系统中最常见的操作。设计过程中经常会遇到参数驱动失败的情况, 这是因为设计者事先并不知道正确的参数值而给出了错误的或非法的参数值, 从而导致图形对象重建失败, 这在一定程度上降低了产品开发的效率, 增加了开发难度。如果在用户改变某个参数值之前, 参数绘图系统自动给出该参数的有效取值范围, 将会大大提高设计效率, 降低设计难度, 增加软件的人性化和智能化程度。

Hoffman 等^[1] 针对只包含水平线段和垂直线段的闭合、不自交、完整约束的直线多边形进行研究, 这类多边形中只允许包含水平方向和垂直方向的距离约束, 确定了在保证线段间拓扑结构不变的情况下相关直线上移动的范围, 并指出同一时间只能考虑一个距离参数的取值范围。

蒋鲲等^[2] 将几何实体限定为只包含水平直线和垂直直线的封闭且不自交的多边形, 将需要求解的参数限定为水平距离约束和垂直距离约束, 并给出求解每个参数有效取值范围的代数算法。

收稿日期: 2015-03-06

基金项目: 国家高技术研究发展计划 (“863” 计划) 项目 (2012AA062202); 山东科技大学科学研究 “春蕾计划” 项目 (2010AZZ034)

作者简介: 张杏莉 (1981—), 女, 山西芮城人, 讲师, 博士, 主要从事计算机图形学、参数化 CAD 方面的研究。

E-mail: xlzhang_only@163.com

Joan-Arinyo 等^[3]提出一种基于区间参数的几何约束系统的求解算法,该算法的主要缺点是基于数据取样,不能给出参数的精确取值范围。

Meiden 等^[4]提出的求解方法考虑三维环境中基于点的距离约束和角度约束的完整约束几何约束系统,将几何约束系统中的几何实体分解成三角形和四面体子问题,在计算某个距离约束参数的取值范围时,找出问题的退化子问题并根据退化子问题求出参数的边界值,根据边界值确定参数的有效取值范围。

Hidalgo 等^[5]基于尺规构造几何约束求解器,对文献[4]的方法进行实验验证,正确计算出参数的取值范围,且取值范围可以是不连续空间。

本研究提出一种更高效的计算约束参数有效取值范围的算法,并在实现此目标的基础上对计算过程做最大程度的简化,提高求解效率。

1 问题的提出

文献[5]的求解算法主要分两步:

1)根据构造计划将几何约束系统分解为多个三角形的退化子问题,若退化子问题直接依赖于参数值 λ ,则直接求解参数 λ 的边界值,若退化子问题间接依赖于参数值 λ ,则移除所求参数 λ 及对应约束,并增加一个或多个退化子问题直接依赖的新约束及参数进行求解,求解所有边界值;

2)在连续的两个边界值中任选一个值判断在该区间几何约束系统是否有解,所有有解区间的并集为参数 λ 的取值范围。

上述算法有两个主要缺点:①求解时需要用构造求解器给出所有几何元素的尺规构造过程,并可能是多次根据不同参数给出不同的尺规构造过程;②在求解参数取值范围时,对所有元素进行多次重复求解,降低了求解效率。

图1是利用文献[5]算法求解参数 λ 的有效取值范围的具体过程。

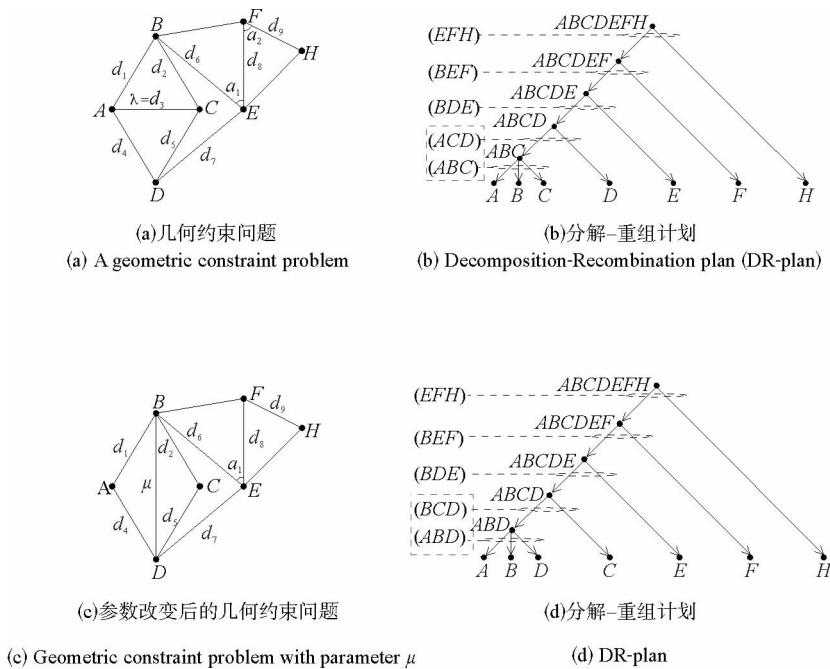


图1 Marta Hidalgo 算法的一个应用实例

Fig.1 An application example of Marta Hidalgo's algorithm

图1(a)为一个二维几何约束问题,图中标有 d_i 的边代表距离约束, a_j 表示角度约束;图1(b)为图1(a)对应的分解-重组计划,即根据尺规构造步骤得到的求解计划,图中的边表示簇的合并,括号里表示每一个合

并步骤中的子问题;由于图 1(b)中 BDE 间接依赖于参数 λ , 因此需移除参数 λ 及对应的距离约束, 并增加参数 μ 及对应的点 B 与点 D 间的距离约束, 对应的几何约束问题如图 1(c) 所示; 图 1(d) 为图 1(c) 对应的分解-重组计划。图 1(b) 和 (d) 中用虚线圈中的部分是两次求解过程中遇到的不同子问题, 未用虚线圈中的部分是重复求解的子问题。

从图 1 可以看出, 使用文献[5]算法, 在求解过程中需要根据不同求解参数多次使用构造求解器给出几何约束问题的尺规构造过程, 并根据尺规构造过程求解新参数的有效取值范围; 同时, 在求解过程中做了许多重复的计算工作, 如对退化子问题 BEF 和 EFH 进行多次求解, 本研究正是基于这一点, 提出文献[5]算法的简化算法。

本研究提出的算法从原几何约束问题着手, 识别其中存在的与所求参数无关的刚体, 借助 D-Tree 分解^[6-8]的思想, 应用刚体变换算法对几何约束系统进行简化, 最后利用简化后的几何约束问题求解参数的有效取值范围。

2 刚体变换算法

刚体是力学中的一个科学抽象概念, 即理想模型。Fudos 等^[9]认为几何约束系统中的刚体是一组相互位置和方向已知的几何元素的组合。高山小等^[10]指出能用无循环约束问题的线性算法求解的完整约束子系统即为原几何约束系统中的一个刚体。而刚体变换将这一刚体当作一个单独的几何体来处理, 原问题即可得以简化。

本研究仅对含有距离约束和角度约束的三角形刚体的情况进行考察, 该类三角形刚体等同于 D-Tree 分解中的簇, 如图 2 所示, a 和 d 分别表示角度约束和距离约束。一个满约束三角形刚体中最多包含三个约束, 共有六种刚体类型。

图 2(a) 的刚体类型为 dad, 包含二个距离约束和一个角度约束, 表示线段 AB 、 BC 的长度以及它们之间的夹角已知; 图 2(b) 是图 2(a) 中 dad 刚体类型的一个特例, 其中 AC 是虚边, 表示在几何约束系统中该边不存在, 等价于文献[10]的 SAS 刚体变换, 是刚体变换的一个特例; 图 2(d) 中的三角形 ABC 的刚体类型为 aad, 通过角度变换可转化为刚体类型 ada(图 2(e)) 的情形, 因此, 图 2(d) 和图 2(e) 可以合并为一种刚体类型。

本研究从以上六种刚体类型着手, 通过刚体变换算法简化原几何约束系统, 缩小求解范围。

算法中所用概念解释如下:

①约束图(G)中顶点 p 的度是指约束图中与顶点 p 相邻的顶点个数, 即与顶点 p 相连的边的个数, 用 $D(p)$ 表示;

②约束图中与顶点 p 连接的线段个数, 用 $D_L(p)$ 表示;

③约束图中与顶点 p 连接的点的个数, 用 $D_P(p)$ 表示。

算法的基本思想: 首先对参数 λ 所在直线或所属三角形中的点的 flag 属性赋值 0, 剩余点的 flag 属性赋值 1; 若 p 点满足 $D_L(p)=2$ 且 $p.flag=1$, 则把该点置入队列 Q 中; 取队列头结点 q , 判定 q 所在的刚体类型, 根据刚体类型进行刚体变换, 判断与 q 相邻的两个点是否满足置入队列 Q 的条件, 如果满足, 置入队尾;

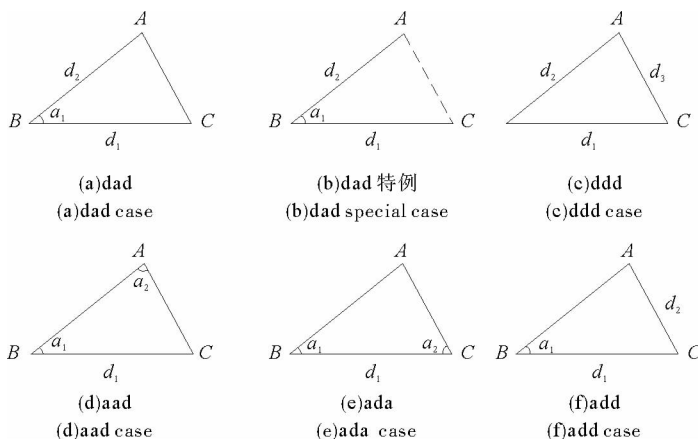


图 2 三角形刚体类型
Fig. 2 Triangle rigid types

重复上一步骤,直到队列为空,算法结束。

为了更直观和易读,下面给出刚体变换算法的伪代码。

算法 1 简化尺规构造几何约束图的刚体变换算法

输入:邻接表

输出:简化后的邻接表

Function SimpleConstraintGraph(G)

 InitQueue(Q)

 For each point p in G Do

 If PointRelatePara (p) Then // p 是参数 λ 所在直线或所属三角形中的点

p .flag = 0

 Else

p .flag = 1

 End If

 If $D_L(p) = 2$ and p .flag = 1 Then

 EnQueue(Q, p)

 End If

 End For

 While not QueueEmpty(Q) Do

 GetHead(Q, q)

 If $T = \text{DetectRigid}(q, \text{node})$ Then

 TransRigid(T)

 If $D_L(T, 1) = 2$ and $T, 1$.flag = 1 Then

 EnQueue($Q, T, 1$)

 End If

 If $D_L(T, 2) = 2$ and $T, 2$.flag = 1 Then

 EnQueue($Q, T, 2$)

 End If

 End If

q .node.flag = 0

 DeQueue(Q, q)

 End While

 Return G

End Function

算法 2 和算法 3 以 ddd 类型的刚体为例来说明。

算法 2 识别 p 点所在三角形是否为刚体及刚体类型

输入:点 p

输出:点 p 所属的刚体类型及刚体信息

Function DetectRigid(p)

TwoLineWithPoint(p, l_1, l_2) //求点 p 所属的两条线段 l_1, l_2

AnotherPointofLine(l_1, p, p_1) //求点 p 所属线段 l_1 的另一端点 p_1

AnotherPointofLine(l_2, p, p_2) //求点 p 所属线段 l_2 的另一端点 p_2

LineWithTwoPoint(p_1, p_2, l_3) // p_1, p_2 所属线段 l_3

If IsDisConBetTwoLine(p_1, p) and IsDisConBetTwoLine(p_2, p) and IsDisConBetTwoLine(p_1, p_2) Then

$T.1 = p_1; T.2 = p_2; T.3 = p; T.4 = l_1; T.5 = l_2; T.6 = l_3; T.info = \text{“ddd”}$
Return T

Else

Return False

End If

End Funtion

算法 3 刚体变换算法

输入:刚体信息 T

输出:修改后的邻接表

Function TransRigid(T)

If $T.info = \text{“ddd”}$ Then

AngConTrans(l_1, l_3) //所有与 l_1 的角度关系转化为与 l_3 之间的角度关系

AngConTrans(l_2, l_3) //所有与 l_2 角度关系转化为与 l_3 之间的角度关系

DeAdjList(p) //删除以点 p 为头结点的链表

End If

End Funtion

假设几何约束问题中几何体的个数为 n ,几何约束的个数为 e ,算法 2 仅对部分邻接表进行遍历,所以复杂度是 $O(8+e)$;算法 3 进行一次 ddd 类型的刚体变换,其复杂度是 $O(3+e)$,所以算法 3 的复杂度是 $O(3+e)$ 。从上述分析得出,整个算法的时间复杂度是 $O(n^2+e^2)$ 。

3 实例

以图 1 所示实例来说明刚体变换算法的使用情况。首先检测到 $D_L(H)=2$,点 H 所在刚体 EFH ,其刚体类型为 ddd,则进行刚体变换,由线段 EF 代替 EFH ;继续检测到 $D_L(F)=2$,点 F 所在刚体 BEF ,其刚体类型为 dad,则进行刚体变换,由线段 BE 代替 BEF ;此时, $D_L(E)=2$,但点 E 不属于任何一个三角形刚体,程序结束。图 3 给出了图 1(a)所示几何约束系统的刚体变换过程。简化后的几何约束系统如图 4(a)所示。

图 4 为对简化后的几何约束系统使用文献[5]算法求解参数 λ 的有效取值范围的具体过程。

图 4(a)为两次刚体变换后的几何约束问题;图 4(b)为图 4(a)对应的分解-重组计划,与图 1(b)相比较,减少了两次合并求解;由于退化子问题 BDE 间接依赖于参数 λ ,图 4(c)为移除参数 λ 及对应约束,增加参数 μ 及对应距离约束后的几何约束问题;图 4(d)为图 4(c)对应的分解-重组计划,与图 1(d)相比较,也减少了两次合并求解。

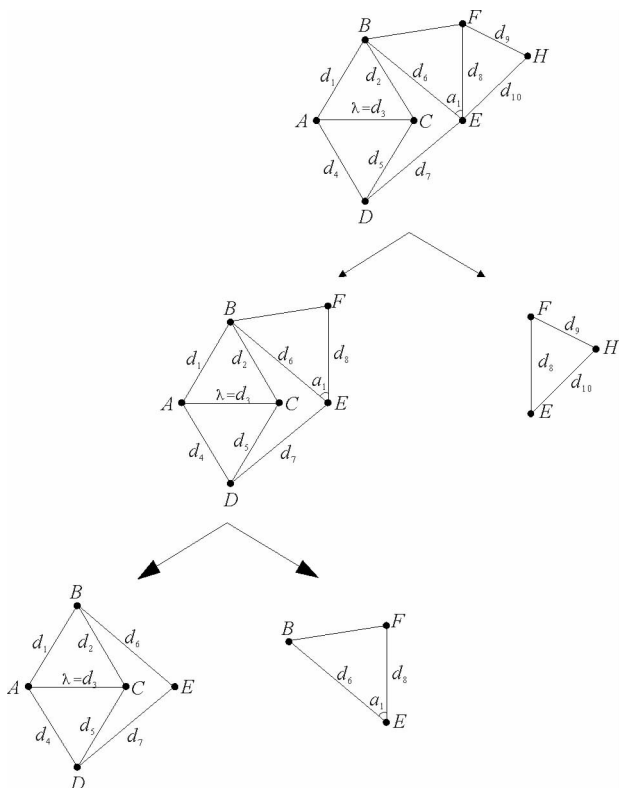


图 3 刚体变换过程

Fig. 3 The rigid transformation process

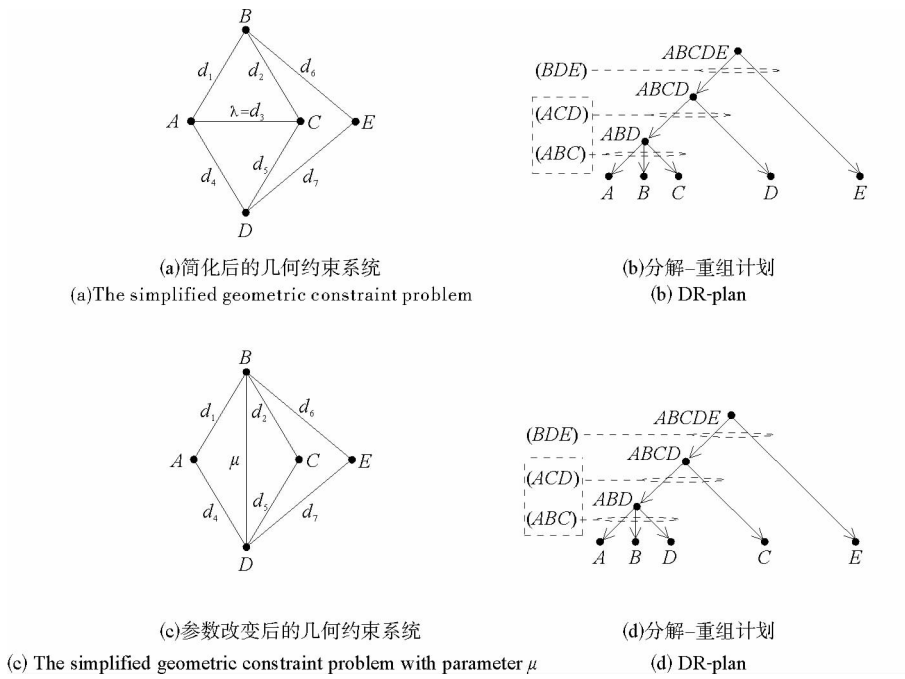


图 4 刚体变换后的求解过程

Fig. 4 The solving procedure after rigid transformation

本实例是一个简单的几何约束问题,对于具体应用过程中出现的复杂几何约束问题,应用本文提出刚体变换算法进行简化处理,求解次数明显减少。

4 结论

针对文献[5]求解算法中对部分几何约束子图重复求解的问题,提出通过刚体变换简化几何约束系统,提高求解效率。对含有距离约束和角度约束的三角形刚体的所有情况进行考察,通过分析总结,给出二维尺规构造几何约束系统中所有可能出现的六种三角形刚体类型,对每一种刚体类型给出刚体变换算法。实例验证表明,基于刚体变换的参数有效取值范围求解算法是高效可行的,此外,该算法思想也适用于几何约束系统求解过程。

参考文献:

- [1]Hoffmann C M, Kim K J. Towards valid parametric CAD models[J]. Computer-aided Design, 2001, 33: 81-90.
- [2]蒋鲲,朱长才,高小山. 参数化 CAD 中参数的有效范围[J]. 计算机辅助设计与图形学学报, 2003, 15(8): 1016-1020.
Jiang Kun, Zhu Changcai, Gao Xiaoshan. Valid range of parameters value in parametric CAD[J]. Journal of Computer-aided Design & Computer Graphics, 2003, 15(8): 1016-1020.
- [3]Joan-Arinyo R, Mata N. Applying constructive geometric constraint solvers to geometric problems with interval parameters [J]. Nonlinear Analysis, 2001, 47: 213-224.
- [4]Van der Meiden H A, Bronsvort W F. A constructive approach to calculate parameter ranges for systems of geometric constraints[J]. Computer-aided Design, 2006, 38: 275-283.
- [5]Hidalgo M, Joan-Arinyo R. Computing parameter ranges in constructive geometric constraint solving: Implementation and correctness proof [J]. Computer-aided Design, 2012, 44: 709-720.
- [6]Owen J C. Algebraic solution for geometry from dimensional constraints [C]//ACM Symposium on Foundations of Solid Modeling. New York: ACM, 1991: 397-407.
- [7]Joan-Arinyo R, Riera A S, Marta S V, et al. Revisiting decomposition analysis of geometric constraint graphs [M]//ACM Solid Modeling. New York: ACM, 2002: 105-115.