中国科学(E辑)
第31卷 第2期 SCIENCE IN CHINA (Series E) 2001年4月

求解几何约束问题的几何变换法*

高小山 黄磊东 蒋 鲲
(中国科学院系统科学研究所,北京 100080)

摘要 提出两种基于图表示求解几何约束问题的方法. 第1种方法能线性地处理无循环约束的几何约束问题. 第2种方法可以解决含循环约束问题. 这一算法的复杂度与Owen, Hoffmann的三角分解法一样是二次,但解题范围有所扩大. 实际上这一算法可以解决所有关于简单多边形的约束问题. 这一算法的核心是将几何变换引入基于关系图的约束求解算法.

关键词 几何约束 关系图 约束图 自由度分析 几何变换 几何自动作图

几何约束求解或几何自动作图是参数化CAD的核心问题,且在连杆生成、计算机视觉、机器人等领域都有应用[1]. 解一个几何约束问题,通常的方法有符号计算法[2]、数值计算法[3]、基于规则方法[4~6]、基于图论算法等. 这些方法在解决问题的范围和速度方面有各自的优点和局限性. 实际的几何约束问题求解系统一般是这几种方法的混合,以达到最佳效果. 基于关系图的算法始于Owen[7],其基本想法是首先将几何约束问题表示为一个关系图或约束图,再用图论中的算法将约束问题表示为一个可以通过直接构造的作图问题(比如尺规作图). 自Owen的工作以后人们提出了很多基于关系图的算法[5, 8~11]. 特别是Hoffmann在文献[12]中给出了将一般约束问题分解为可作图子集的算法,使基于关系图的算法在一定意义下达到了顶峰.

但是这些结果并未完美地解决几何约束问题. 基于关系图算法的基本想法是将一个大的几何约束问题化为若干个小的几何约束问题,然后再将这些小的问题"装配"起来. 这一装配过程需要求解方程. 在2维情形,如果求解方程的个数总是$\leqslant 2$,我们称这一几何约束问题为无循环约束问题. 反之称为有循环约束问题. 无循环约束问题可以完整地解决,但有循环约束问题因为需要解多于2个以上的联立方程,实际计算起来还会遇到问题. 所以将有循环约束问题变为无循环约束问题是基于关系图方法的关键.

本文将给出两个基于关系图的新算法,其特点如下:

(1) 对于无循环约束问题,可以给出一个线性的判定与解决算法. 这一算法可以处理任意自由度的几何体与任意约束度的约束,因此原则上可以处理三维问题.

(2) 对于有循环约束问题,本文给出的算法是一个二次算法. 这一复杂度与Owen和Hoffmann算法的复杂度相同,但解题范围有所扩大,可以解决这两个算法不能解决的问题. 实际上所有关于简单多边形的约束问题均可以由这一算法解决.

2000-06-25 收稿
* 国家杰出青年科学基金(批准号: 69725002)和国家基础研究发展规划(批准号: G1998030600)资助项目

(3) 本文给出的算法对于完整约束与欠约束问题均适用.

本文的核心思想是将几何变换引入关系图以解决循环约束问题. 这一想法应该还有发掘余地,从数千年几何作图研究的成果中寻找得力工具应是解决几何约束问题的关键之一.

1 无循环约束问题的线性算法

1.1 几何约束问题的图表示

本文提出的方法和算法是基于图表示的,所以在求解一个几何约束问题之前,我们先要把问题化成图表示的形式. 一个几何约束问题在结构上分为两部分:几何体和几何约束. 几何体有点、直线、圆、线段、圆弧等. 几何约束则是它们之间的相互关系,如点在线上、点与点之间的距离、直线与圆相切等等. 我们用图(G)的顶点表示几何体,用图的边表示几何约束. 如果两个几何体之间存在约束关系,那么它们对应的顶点相邻,即两顶点之间存在边.

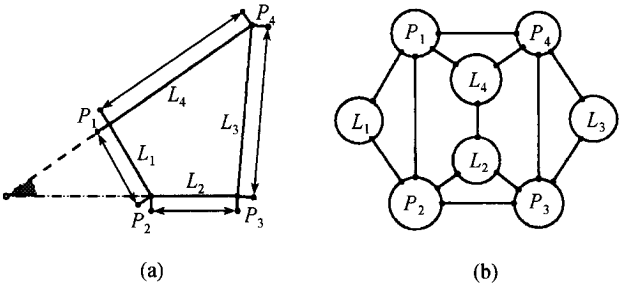
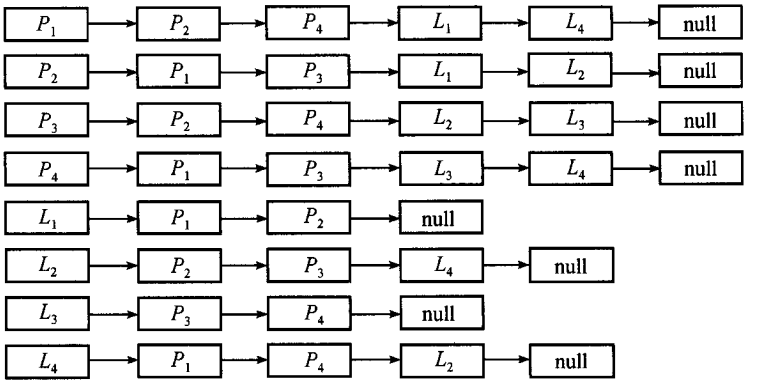


图 1　约束问题(a)及其对应图(b)

如图 1(a)给出的约束问题,约束为四边形四边的长和两条对边的夹角,它所对应的图表示如图 1(b). 这是一个有循环的几何约束问题.

给定一个几何约束问题,我们就可以得到相应的约束图. 为了便于算法复杂性分析,我们给出约束图的存储方式. 我们采用的是邻接表的存储方式. 以图 1(b)为例,它的存储结构如下:



其中第 1 列中的结点称为头结点.

1.2 算法描述和复杂度分析

在介绍具体的算法之前,我们先解释一下两个概念. 一个是约束图(G)中顶点 v 的度,它是指约束图中与顶点 v 相邻的顶点的个数,也即是与顶点 v 相连的边的个数,我们用 DEG(v)表示. 另一个是顶点 v 所代表的几何体的自由度,我们用 DOF(v)表示. 在 Euclid 平面上,点和直线的自由度都是 2,圆的自由度是 3,线段的自由度是 4.

算法的初始想法很自然:给定一个几何约束问题,产生相应的约束图 G;删除 G 中 DEG≤DOF 的顶点,并去除与其相连的边,得到新的约束图 G,把该顶点置入一个队列;重复删除的过程,直到 G 为空,按照队列尾到队列头的顺序,就是我们所要得到的作图序列. 所谓无循环约束情形,就是对于原始约束图 G 和删去顶点后得到的新图,我们总可以找到 DEG≤DOF 的顶点,直到 G 为空.

算法描述:

输入,邻接表;输出, 作图序列.

(1) 置邻接表所有头结点的 Flag 值为 0.

(2) 生成初始化队列.

对邻接表所有头结点进行扫描,把所有 DEG≤DOF 的结点存入一个队列,同时置此结点的 Flag 值为 1.

(3) loop 取队列当前指针所指的结点 v(初始值指向队列的头结点).

```
loop 检测邻接表中 v 的邻接结点 w
 DEG(w) = DEG(w) - 1          /* v 对应约束图中已经消去的结点 */
 if Flag(w) = 0               /* w 未在队列中 */
  if DEG(w)≤DOF(w)
  then 把 w 加入队列,同时置 w 的 Flag 为 1
  endif
 endif
 w 等于 v 的下一个邻接结点
repeat
如果 v 不是队列的最后一个结点,队列指针后移一位.
repeat
```

(4) 由队列尾向队列头的结点顺序就是给定约束问题的作图序列.

注意算法中把 w 加入队列时有一种情况例外. 就是当 w 是一条直线,它的邻接结点中有两个结点的 Flag 值为 0(未在队列中),而那两个结点也恰好是两条直线时,w 不加入队列. 这是因为如果此时将 w 加入队列,构造时将出现一条直线由另两条直线决定的情形,这可能会导致无穷解或者无解.

假设给定几何约束问题中几何体的个数为 n,几何约束的个数为 e, 那么上述算法中第 1 步需要 $O(n)$步运算. 第 2 步生成初始化队列需要对邻接表的所有头结点进行扫描,所以需要 $O(n)$步运算. 第 3 步由两个循环构成,内部循环检测 v 的邻接结点,这里 v 是已经在队列中的结点,它对应约束图中已经被删去的顶点. 因为只有与 v 相邻的顶点才有可能下一次被删去,所以这次循环的次数即 v 的相邻顶点的个数,设此数为 d_i. 外部循环实际上是对队列中的结点进行操作,所有队列中的结点对应邻接表中的头结点,所以这两个循环刚好把邻接表扫描一遍. 因此第 3 步的运算次数总共是 $n+\sum_{i=1}^{n} d_i = n+2e$ 步. 从上述分析得出,整个算法的时间复杂度是 $O(n+e)$.

当给定的约束问题是一个完整约束或欠约束问题,而涉及的几何体只有点和直线,此时 e

$\leqslant 2n-3$，算法的复杂度即为 $O(n)$.

1.3　在几何定理证明中的应用

因为大部分几何定理均为无循环约束情形的，所以我们可以把上节的算法应用于自动几何定理证明(AGTP)之中. 几何定理证明中的有些方法如 Hilbert 方法[2]、面积法[13]以及吴方法的一个特例[14, 15]都是针对构造型几何定理的. 用上节的算法可以自动求得几何定理的构造型形式.

图 2(a)是平面几何中的 Cantor 定理：E, F, G, H 是圆内接四边形四边的中点，K, L, M, N 是由 G, H, E, F 向对边所作垂线的垂足，四条垂线交于一点. 图 2(b)是该问题的一个部分约束图，即我们只考虑了点 E, M, G, K, 对点 H, L, F, N 可以类似处理. 图中出现的双边是由于涉及的几何体是线段，线段的自由度是 4. 这是一个无 loop 情形的欠约束问题，用我们的算法可以得到如下的构造序列，这个构造序列可以作为 AGTP 的输入：

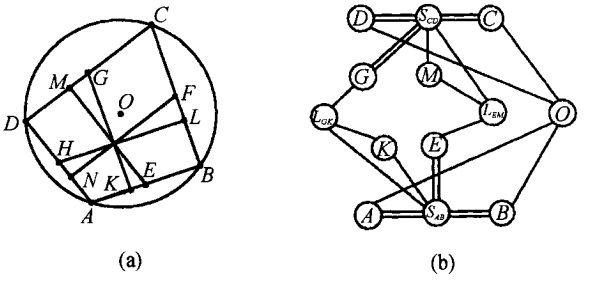


图 2　Cantor 定理(a)及其一个部分约束图(b)

作自由点 O, A.

在以 O 为圆心过 A 的的圆上作自由点 B, C, D.

构造线段 AB, BC, CD, DA 的中点 E, F, G, H.

构造过点 E 垂直于 CD 的直线 l.

构造 CD 与 l 的交点 M.

类似地构造 N, K, L.

注意在算法中几何体的选取不是惟一的，不同的选取顺序会导致不同的构造序列. 例如下面的顺序也可以是 Cantor 定理的一个构造序列：

$$A, B, C, O, D, S_{AB}, S_{BC}, S_{CD}, S_{DA}, E, F, G, H, \cdots.$$

这个构造顺序从一般意义上讲更好，因为在构造前 5 个点时它只需要 2 个线性方程和 1 个二次方程，而先前的构造顺序需要 3 个二次方程. 利用文献[16]中搜索策略，我们可以找到多个和最优的构造顺序，从而可以得到最短的证明.

这个应用类似于文献[17]中的工作，但我们的方法可以处理更多的几何体和几何约束，并且算法是线性的.

因为上节提出的算法可以处理任意自由度的几何体，所以我们可以用它来解决 3 维的问题. 图 3 是 1964 年国际数学奥林匹克竞赛的一道题[13]："设 $ABCD$ 是一个四面体，G 是平面 ABC 上一点. 过点 A, B, C 且平行于直线 DG 的直线分别交 A, B, C 的对立面于点 P, Q, R. 证明 $GPQR$ 的体积是 $ABCD$ 体积的三倍". 下面是用我们算法得到的构造顺序：A, B, C, D, P_{ABC}, G, L_{AP}, L_{BQ}, L_{CR}, P, Q, R, 这里 P_{ABC} 是过点 A, B, C 的平面，L_{AP} 是直线 AP.

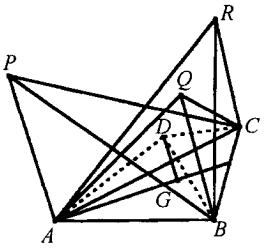


图3 3维作图问题

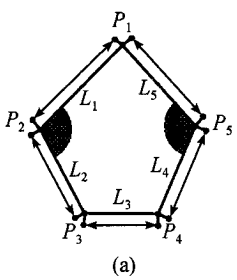
2 有循环约束问题的二次算法

2.1 3种变换

我们先来看一个例子. 给定一个几何约束问题(如图4(a)),它所对应的约束图如图4(b). 按照上一节给出的算法,我们只能删去顶点 L_3,接下来就不能继续下一步了. 其原因在于当删去顶点 L_3 后,图中再也不存在 DEG≤DOF 的顶点了,也就是产生了循环约束. 针对这样的情形,我们提出3种几何变换,可以把有循环约束情形的问题化为无循环约束情形的问题.

2.1.1 刚体变换 如果在一有循环约束的约束图中存在一个完整约束子图可以用第1节的算法解决,则这一子图表示原问题中一个刚体. 所谓刚体变换就是将这一刚体当作一个单独的几何体来处理,这样原来的问题即可得以简化. 这实际上等价于 Hoffmann 的三角分解法[10],其复杂度是二次的.

我们这里只需要刚体变换的一个特例. 如果约束图 G 中存在形如图5(a)的子图,表示线段 P_1P_2, P_1P_3 的长度及它们之间的夹角已知,则 P_1, P_2, P_3, l_1, l_2 代表一个刚体. 我们可以将 P_1, l_1, l_2 用一条连接 P_2, P_3 的新线 l_3 代替,得到如图5(b)的新图. 所有其他边 $l_i, l_j, \cdots$ 与 l_1, l_2 之间的角度关系均可以转化为与 l_3 之间的角度关系. 我们假定除这些角度约束外 l_1, l_2, P_1 不再有别的约束. 我们称这样的替换为一个 SAS 变换. 很明显如果图5(b)中的几何元素已构造出,则图5(a)中的几何元素可以随之构造出来. 在约束图 G 中形如图5(a)的子图我们规定必须满足



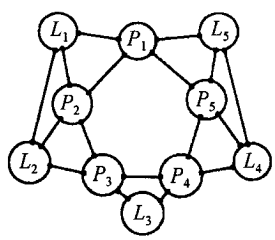


图4 边角边变换实例(a)及其对应的约束图(b)

$$\mathrm{PDEG}(l_1) = 2,\quad \mathrm{PDEG}(l_2) = 2,$$
$$\mathrm{LDEG}(l_1) \geqslant 1,\quad \mathrm{LDEG}(l_2) \geqslant 1,$$
$$\mathrm{PDEG}(P_1) = 2,\quad \mathrm{LDEG}(P_1) = 2,$$

才可以进行 SAS 变换. 进行一次 SAS 变换的复杂度是 $O(\mathrm{DEG}(l_1) + \mathrm{DEG}(l_2))$,它只是对部分原邻接表进行了遍历,所以也即是 $O(e)$.

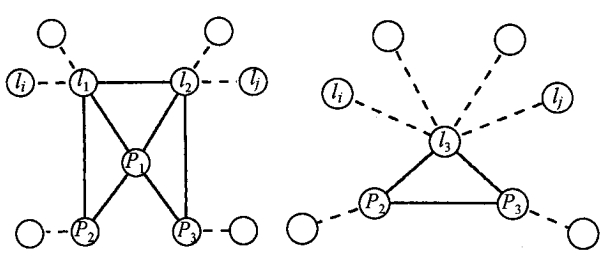


图5 刚体变换前(a)及变换后(b)的约束图

对于图4(a)给出的约束问题,我们只要作两个 SAS 变换,就可以把原来的有循环约束问题化为无循环约束问题.

2.1.2 角度变换 图6(a)给出的约束问题,由于没有给定边 P_1P_2 与 P_1P_4 间的角度约束,所以表面上 P_1P_2, P_1P_4 并不构成一个刚体. 但是由于四边形的其他三个角都已知,所以问题隐含了边 P_1P_2 与 P_1P_4 间的角度已知,因此我们只要把其他三角的任一角度化为 P_1P_2 与

P_1P_4 间的角度，就可以进行一个 SAS 变换将问题解决.

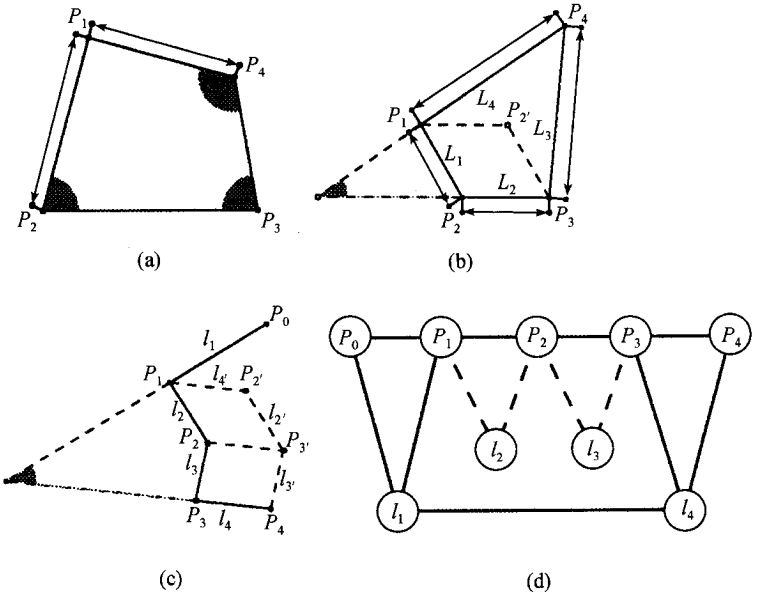


图 6　角度变换(a)和平移变换(b)实例以及平移变换的一般情形(c)和变换前的约束图(d)

我们用$[L_i$ 和 $L_j]$表示两直线 L_i、L_j 之间的有向角，或称为全角[13]，则有

$$[L_i, L_j] = -[L_j, L_i],$$

$$[L_i, L_j] = [L_i, L_1] + [L_1, L_j].$$

若 l_1 和 l_2 之间的角度已知，l_2 和 l_3 之间的角度已知，那么我们就可以求得 l_1，l_3 之间的角度. 判断 l_i 与 l_j 之间是否角度已知(给定或可以求得)，在约束图中即表现为是否存在从 l_i 到 l_j 的一条路径，且路径经过的所有顶点都是直线. 如果存在这样一条路径，则 l_i 和 l_j 间的角度已知. 我们把这条路径上的所有直线归入一个等价类，所以我们这里的等价类是一个直线的集合，等价类中每两条直线之间的夹角都可以通过计算求得.

下面给出寻找直线等价类的算法：

输入：邻接表；输出：直线等价类.

(1) 设 p 为邻接表指针，q 为队列指针. 置邻接表所有头结点的 FLAG 值为 0，p 指向邻接表第一个头结点，$i = 0$.

(2) 设 u 为 p 所指结点. 若 u 为空，算法终止；若 u 不是直线或 u 的 FLAG 值为 1，p 指向下一个头结点，重新进行步骤 2；否则 $i = i + 1$，进行下一步.

(3) 置 u 的 FLAG 值为 1，把 u 加入队列 Q_i. 令 q 指向队列 Q_i 的第一个结点(即为 u).

(4) 设 v 为 q 所指结点，检测 v 的所有邻接结点 w，若 w 是直线且 FLAG 为 0，把 w 加入队列 Q_i 的末端，置 w 的 FLAG 值为 1.

(5) q 指向队列的下一个结点. 若为空，p 指向下一个头结点，返回第 2 步. 否则返回第 4 步.

上述算法可以生成所有的等价类. 寻找上述等价类的过程实质上是遍历顶点全是直线的

约束子图. 我们用图论中的宽度优先搜索对约束子图进行遍历,这种遍历只需扫描部分邻接表,所以它的复杂度是 $O(n+e)$.

2.1.3 平移变换 我们来看图1(a)给出的约束问题,它是一个含循环约束问题,用刚体变换和角度变换都不能解决. 我们作如下处理:过 P_1 作 P_2P_3 的平行线,过 P_3 作 P_1P_2 的平行线,两线交于点 P_2' (如图6(b)). 现在我们来看四边形 $P_1P_2'P_3P_4$. 因为 P_1P_4 与 P_2P_3 间的夹角已知,所以 P_1P_4 与 $P_1P'_2$ 的夹角已知. 因此用第1节的方法就可以很容易的求解四边形 $P_1P'_2P_3P_4$, 从而我们也容易得到点 P_2. 这种通过构造平行四边形进行的变换, 文献[6]中称为平行四边形变换, 文献[4]称为相似变换. 图1(a)中问题用 Owen, Hoffmann 的方法是不能解决的,在文献[5, 11]中需要解3个以上的联立方程组,我们可以把它化为尺规作图问题.

平移变换可描述如下,假定有点列 $P_0,P_1,\cdots,P_k$ 满足下列条件:

(1) $|P_{i-1}P_i|$ (点 P_{i-1} 到 P_i 的距离)已知, $i=1,\cdots,k$;

(2) $[l_1,l_k]$ 已知, $l_1=P_0P_1$, $l_k=P_{k-1}P_k$;

(3) $\mathrm{PDEG}(P_i)=2(i=1,\cdots,k-1)$;

(4) $\mathrm{LDEG}(P_i)\leqslant 2(i=1,\cdots,k-1)$, 且与 P_i 相连的直线只能是 $P_{i-1}P_i$ 或 P_iP_{i+1};

(5) $\mathrm{PDEG}(l_1)=\mathrm{PDEG}(l_k)=2, l_2,\cdots,l_{k-1}$ 上只能有点列中的点,

那么我们可以将 l_k 平移到 l'_k (如图6(c)和图7(a)). 平移后我们可以作一个 SAS 变换.

以图6(c)($k=4$)为例我们来说明平移变换的过程以及约束图的变化. 图6(c)在未进行平移变换之前的约束图如图6(d),虚线表示这个约束可有可无. 我们作两个平行四边形后把 l_4 平移到 l'_4,l_2,l_3 平移到 l'_2,l'_3, 从而约束图变为图7(a). 图7(a)中含有满足 SAS 变换的子图,SAS 变换把 l_1,P_1,l'_4 用 l'_1 代替,约束图变为图7(b)的情形. 几何作图过程与上述过程相反,即从图7(b)对应的几何图形得到图7(a),再得到图6(d)对应的几何图形.

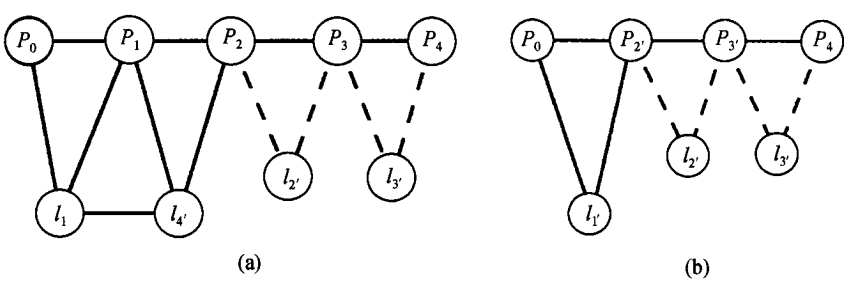


图7 平移变换中(a)及变换后(b)的约束图

平移变换的结果是作一次 SAS 变换,所以我们把平移变换分成两部分:平移和 SAS 变换. 平移所需的运算和中间结点(图6(d)中的 P_1,P_2,P_3,l_2,l_3)的个数成正比,它要把这些中间结点替换成平移后的结点. 因为 $P_0,P_1,\cdots,P_k$ 是约束图中的一条"路径",这些中间结点可以通过一个深度优先检索遍历到,所以这一步的复杂度是 $O(n+e)$的. 因为 SAS 变换的复杂度是 $O(e)$,所以平移变换的总的复杂度是 $O(n+e)$.

2.2 算法描述

本节我们将这三种变换结合起来,给出一个二次算法. 算法描述如下:

输入:邻接表;输出:作图序列.

(1) (角度变换)生成直线的等价类 $L_1,\cdots,L_s$ 使得同一 L_i 中的两直线之间的角度可以计算. 置邻接表当前指针 p 的初始值指向第一个头节点.

(2) 用第 2 节的线性算法处理邻接表,如果已经得到完整的作图序列,程序终止,否则进入下一步.

(3) v 是邻接表当前指针 p 所指结点. 若 v 为空(已到表的末尾),程序终止. 此时如果约束图中所有结点均已删除,那么删除过程的逆序就是构造的顺序;如果此时约束图中还有结点,那么约束问题不可解. 若 v 不为空,如果满足:

(a) v 是一直线,

(b) v 上只有两点 P_0,P_1,

(c) $|P_0P_1|$ 已知,

则进行第 4 步,否则邻接表指针 p 指向下一个头结点,返回第 3 步.

(4) 由 P_1 (P_0 的情形类似)出发寻找 $P_2,\cdots,P_k$ 满足:

(ⅰ) $|P_iP_{i+1}|$ $(i=1,\cdots,k-1)$ 已知.

(ⅱ) 在约束图中与 $P_i(i=1,\cdots,k-1)$ 相连的几何体只能是点 P_{i-1},P_{i+1} 或直线 $P_{i-1}P_i,P_iP_{i+1}$.

(ⅲ) 直线 $P_iP_{i+1}(i=1,\cdots,k-1)$ 上除点 P_i,P_{i+1} 外无其他点.

(ⅳ) $[P_0P_1,P_{k-1}P_k]$ 已知,即直线 P_0P_1 和 $P_{k-1}P_k$ 在同一等价类中.

(5) 若 $k=1$,邻接表指针 p 指向下一个头结点,返回第 3 步.

(6) 若 $k=2$,对 P_0,P_1,P_2 作 SAS 变换. 把新引入的直线 P_0P_2 加到邻接表的结尾,在 P_0P_1 的等价类中加入 P_0P_2,删去直线 P_0P_1, P_1P_2 和点 P_1,同时邻接表指针 p 指向下一个头结点,返回第 2 步.

(7) 若 $k>2$,对 $P_0,P_1,\cdots,P_k$ 作平移变换. 把新引入的直线 l' 加到邻接表的结尾,在 P_0P_1 的等价类中加入 l',删去直线 P_0P_1, $P_{k-1}P_k$ 和 P_1,用 l_i' 替换 l_i, $i=2,\cdots,k-1$,并替换相应的点. 同时邻接表指针 p 指向下一个头结点,返回第 2 步.

根据以前的分析知道,算法第 1 步的复杂度为 $O(n+e)$. 从第 2 步到第 7 步是主循环. 因为每经过第 6 和 7 步一次邻接表增加一个头结点而删去 3 个头结点(两条直线和一个点),所以这个循环的次数不会超过 n 次. 而且邻接表指针 p 前面的满足第 3 步条件的直线不会和新生成的直线有角度关系,因为等价类是继承的,原先不在一个等价类中的直线永远不在一个等价类中.

下面我们来分析一下循环内部的情形. 第 2 步的复杂度是 $O(n+e)$;从第 3 步开始寻找一条满足要求的直线. 为了检查直线 v 是否满足条件,我们需要检查 v 上所有邻接结点以找 P_0,P_1. 然后再在 P_0 的邻接结点中找 P_1 这点. 这共需 $O(\mathrm{DEG}(v)+\mathrm{DEG}(P_0))\leqslant O(e)$步. 第 4 步寻找约束图中的一条路径 $P_1,\cdots,P_k$. 具体讲,我们需要检查 P_i 的邻接结点中是否除了点 P_{i-1} 外只有一点 P_{i+1},然后再验证(ⅰ)~(ⅳ)是否满足. 注意检查两线之间的夹角是否已知,只要检查它们是否属于同一等价类. 这一操作可以通过使用适当的数据结构而用两步验证. 因为这条路径可以用一次深度优先检索遍历到,所以第 4 步的复杂度是 $O(n+e)$的;第 6 步的 SAS 变换复杂度是 $O(e)$;第 7 步的平移变换的复杂度是 $O(n+e)$. 所以循环内部总的

复杂度是 $O(n+e)$. 因此整个算法的复杂度是 $O(n(n+e))=O(n^2+ne)$.

2.3 简单多边形的约束问题

为了更好地说明本文提出的算法,我们来看一个较复杂的例子. 图 8(a)给出的约束问题,其相应的约束图如图 8(b),这是一个有循环的完整约束问题. 经过平移后的约束问题化为图 8(c)的情形,由 P'_1 代替 P_1, L'_1 代替 L_1, L'_5 代替 L_5. 这实际上是图 4(a)中给出的问题,可以用 SAS 变换解决.

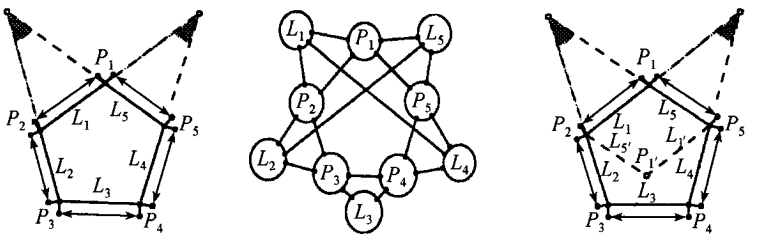


图 8 一个几何约束问题(a)及其对应的约束图(b)和经平移变换后的约束图(c)

上述例子实际上是所谓简单多边形约束的特殊情形. 应用本节提出的算法,可以解决所有简单多边形的约束问题.

设 $P_0,\cdots,P_{n-1}$ 为平面上 n 个点. 以 P_i 为顶点的简单多边形的约束问题只考虑两种约束:

(1) 边长约束: $|P_iP_{i+1}|$ 已知,

(2) 角度约束:直线 P_iP_{i+1} 和 P_jP_{j+1} 之间的角度已知(下标模 n).

文献[6]中指出,使用所谓平行四边形变换,所有简单多边形的完整约束问题均可以解决. 文献[6]中使用的方法是基于推理规则的,且无复杂度分析. 本文所给算法是基于关系图的二次算法. 下面说明我们的算法能解决所有简单多边形的完整约束与欠约束问题.

首先考虑完整约束问题. 对于具有 n 个顶点的简单多边形,应有边长与角度的 $2n-3$ 个约束. 分三种情形:

(ⅰ) 有 n 个边长约束, $n-3$ 个角度约束. 通过平移和 SAS 变换即可解决问题. 图 4(a)和图 8(a)就是这类约束的例子.

(ⅱ) 有 $n-1$ 个边长约束, $n-2$ 个角度约束. 经过平移与 SAS 变换后,原问题可以分别化为下列三种情形:

(a) 与长度未知边相邻的两角未知.

(b) 与长度未知边相邻的一角未知,另一角已知.

(c) 与长度未知边相邻的两角已知.

(a)和(b)可以进一步用 SAS 变换解决. (c)通过一个角度变换可以使得 $[l_1, l_2]$ 已知,再通过平移变换即可与图 1(a)的约束问题类似解决. 也可以用三角剖分法[10, 7]或全局延拓法[4]化为尺规作图问题.

(ⅲ) 有 $n-2$ 个边长约束, $n-1$ 个角度约束. 由角度变换,再通过 SAS 与平移变换,可以变为下列两种情形:

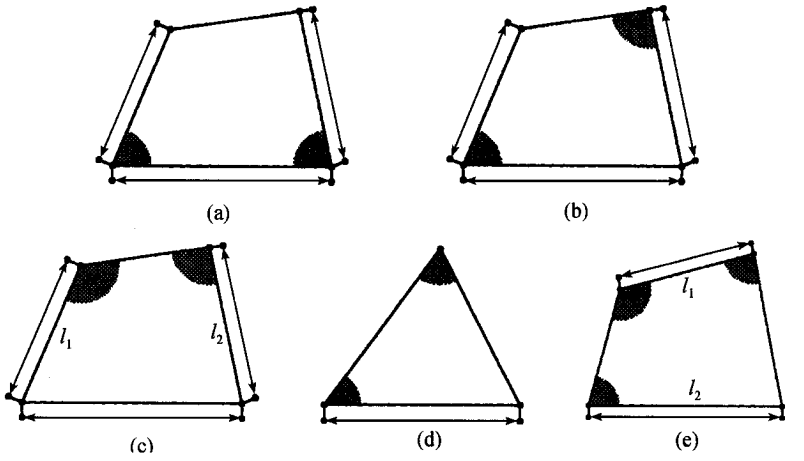


图 9　简单多边形作图问题的 5 种情形

(d) 两长度未知边相邻.

(e) 两长度未知边不相邻.

(d)是一无循环约束问题. (e)可以依次通过角度变换($[l_1, l_2]$已知)、平移变换(类似图 1 (a)的约束问题)、SAS 变换化为(d)的情形. 注意,这是一个特殊的平移变换,其中一个线段长度未知. 我们可以在算法中将这一特殊情形记住. 也可以用三角剖分法[7,10]或全局延拓法[4]化为尺规作图问题.

对于简单多边形的欠约束问题,可以通过增加若干约束使之变为完整约束问题,再用上述方法解决.

参　考　文　献

1 Gao X S. Automated Geometry Diagram Construction and Intelligent CAD. Automated Deduction in Geometry. Berlin: Springer-Verlag, 1999. 232 ~ 257

2 吴文俊. 几何定理证明的基本原理. 北京: 科学出版社, 1984

3 Light R, Gossard D. Modification of geometric models through variational geometry. Geometric Aided Design, 1982, 14: 208 ~ 214

4 Gao X S, Chou S C. Solving geometric constraint systems Ⅰ. A global propagation approach. Computer-Aided Design, 1998, 30(1): 47 ~ 54

5 Lee J Y. A 2D geometric constraint solver for parametric design using graph reduction and analysis. In: Automated Deduction in Geometry. Berlin: Springer-Verlag, 1999. 258 ~ 274

6 Verroust A, Schonek F, Roller D. Rule-oriented method for parameterized computer-aided design. Computer-Aided Design, 1992, 24 (2): 531 ~ 540

7 Owen J C. Algebraic solution for geometry from dimensional constraints. In: Proc 1st Symp Solid Modeling Foundations & CAD/CAM Applications. Washington: ACM Press, 1991. 379 ~ 407

8 陈立平, 涂重斌, 罗　浩, 等. 一种新的面向参数化绘图的约束管理技术. 软件学报, 1996, 7(7): 394 ~ 400

9 董金祥, 葛建新, 高　屹, 等. 变参绘图系统中约束求解的新思路. 计算机辅助设计与图形学学报, 1997,9(6): 513 ~ 519

10 Fudos I, Hoffmann C M. A graph-constructive approach to solving systems of geometric constraints. ACM Transactions on Graphics, 16

(2): 1997, 179 ~ 216

11 Yuan B, Sun J G. A graph based approach to design decomposition. In: The Sixth International Conference on CAD & CG, Shanghai-China. 1999. 984 ~ 988

12 Hoffmann C M, Lomonosov A, Sitharam M. Finding solvable subsets of constraint graphs. LNCS, 1997, 1330: 163 ~ 197

13 Chou S C, Gao X S, Zhang J Z. Machine Proof in Geometry. Singapore: World Scientific, 1994

14 Chou S C. Mechanical Geometry Theorem Proving. D Reidel Publishing Company, 1988

15 Wang D. On Wu's Method for proving constructive geometric theorems. In: Proc IJCAI 89, Detroit, August 20 ~ 25, 1989. 419 ~ 424

16 Chou S C, Gao X S, Zhang J Z. Automated Generation of readable proofs with geometric invariants. Ⅰ. Multiple and shortest proof generation. J of Automated Reasoning, 1996, 17: 325 ~ 347

17 Li H B, Cheng M T. Automated Ordering for Automated Theorem Proving in Elementary Geometry. MM-Preprints, 1999, 18: 84 ~ 97