

基于区间算术和四叉树消除偏置曲线自交算法的实现

万 健,徐小媚,叶晓华

(杭州电子科技大学 软件学院,浙江 杭州 310018)

(bingxuxiaomei@163.com)

摘 要:消除偏置曲线自交点可以提高系统对零件细节部分的处理能力,提高了零件的制造精度。总结了消除偏置曲线自交点的各种算法的优缺点,提出了基于区间算术和四叉树的算法。这种算法能够有效快速探测到偏置曲线全局与局部自交的位置。最后介绍了该算法的应用。

关键词:偏移量;自交;区间算术;四叉树

中图分类号: TP391.41 **文献标识码:** A

Eliminating local and global self-intersections of offsets based on interval arithmetic and quad trees

WAN Jian, XU Xiao-mei, YE Xiao-hua

(Department of Software, Hangzhou Dianzi University, Hangzhou Zhejiang 310018, China)

Abstract: Eliminating offset self-intersection could improve the ability to handle the detail of parts and the precision to make the parts. All kinds of method used around world and presented approach based on interval arithmetic and quad tree was presented. This method is good for detecting both local and global interactions. At last, gave examples for this method.

Key words: offsets; self-intersection; interval arithmetic; quad trees

0 引言

工业领域许多 CAD/CAM 系统,往往需要建立有关几何模型的自动的数控刀具轨迹,涉及到偏置曲线的设计与计算。偏置曲线在文献[1~3]有相关的研究,它的应用在文献[4,5]中有详细的说明。偏置曲线的方程式为 $C_d^0(t) = C(t) + \vec{N}(t)d$, 其中 $t \in [0,1]$, d 就是曲线要偏置的距离, $N(t)$ 是 $C(t)$ 的单位法向量。

在偏置曲线形成时在以下的情况下产生自交点:

1) 与原曲线之间的距离远远的小于偏离的距离,偏置曲线往往会发生自交。

2) 原曲线的曲率半径小于偏离的距离,发生循环、尖端的点。

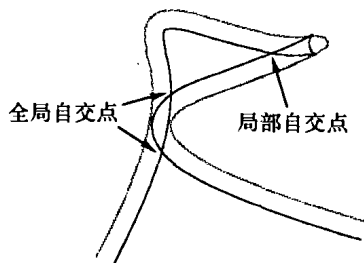


图1 偏置曲线产生全部与局部自交现象

第一种情况产生如图1所示的全局自交,第二种情况产生如图1所示的局部自交。

等距线在 CAD/CAM 系统中有很广泛的应用,但是在原曲线偏置的过程中往往会产生自交形成闭环。这闭环的

产生给各种刀具和零件的制造产生误差,甚至造成在数据扫描时进入死循环。所以消除闭环是 CAD/CAM 系统中的必然要求。

消除偏置曲线的闭环环最主要的目的是为了提高系统对零件细节部分的处理能力,提高了零件的制造精度。根据其处理偏置线还是原曲线,消除自交现象的方法可以分为两大类:对原曲线的处理和对偏置曲线的处理。对原曲线处理的方法有两种:1) 基于像素的算法^[7~9]相当稳定而且具有较高的精度,但其需要大量的内存和计算时间;2) 快速成型激光光斑半径补偿算法^[6],是对原曲线先处理去掉一些奇点(有可能产生自交的点),这种方法较前一种方法效率高,省内存和时间。对偏置曲线的处理,回逆求交判断法^[10,11]是将采用折中降低了精确度但是节省了内存和计算机时间。

本文提出了基于区间算术和四叉树的算法,该算法既能对局部和全局的自交点进行处理,又能节省计算时间、计算机内存。该算法已经在 C++ 测试通过。

1 探测自交算法

本文提出的算法形象的说,也就是递归细分。所谓递归细分就是对区域进行递归分割,它有几个组成的要素:子区域、子线段进行分类,区域终止分割的标准。实现递归细分,要借助四叉树和区间算术。本文的算法借鉴了文献[12]介绍四叉树有关应用,区间算术在文献[13]有很详细的论述。

1.1 树的结构

本算法中四叉树的结构如下所示:

收稿日期:2005-03-04;修订日期:2005-05-06 基金项目:国家自然科学基金资助项目(60303028)

作者简介:万健(1969-),男,福建人,教授,博士,主要研究方向:计算机辅助几何设计与计算机图形学;徐小媚(1981-),女,浙江瑞安人,硕士研究生,主要研究方向:计算机辅助几何设计与计算机图形学、多媒体;叶晓华(1958-),女,浙江杭州人,主要研究方向:计算机辅助几何设计与计算机图形学。

```

typedef struct TNode
{
    NCenter nctr;           //定义一棵树
    pAreaList nodeare;
    float width;           //这个节点的宽度
    int curcount;          //在这个节点里曲线的条数
    struct TNode * bnode;   //指向右兄弟节点
    struct TNode * pnode;   //指向父节点的指针
    struct TNode * cnode[4]; //指向子节点的指针
} tnode, * TreeLink;

```

1.2 探测自交算法

本文里找自交点的方法利用区间算法和四叉树。如果一个节点里,存在两段以上的曲线段或一段非单调的曲线,则有可能存在交点,那么就对这个节点进行再细分;如果该节点里没有曲线段或含一段单调的曲线段,则跳过对这个节点的处理。

算法1 探测自交

(1)input *TreeNode* 即四叉树初始化的值,参数化的偏置曲线 $C(t), t \in [0, 1]$, 定义放自交点的数组 *arrIntersections*, 结束分割的节点的 *endWidth* 的大小。

(2)If 当前节点 *tnode.curcount* > = 2 then 转到(3) Else If *tnode.curcount* == 1 then 转到(4) Else 转到(5)

(3)If *tnode.width* < = *endwidth*. Then 转到(5) Else 调用四叉树对 *tnode* 进行细分,并调用区间算术对该节点的线段进行处理;再转到(2),继续处理 *tnode* 的第一个子节点。

(4)调用区间算术判断该节点曲线的单调性。If 曲线单调 Then 转到(5) Else 转到(3)

(5) *tnode* 含有自交点,取 *tnode.nctr* 放入数组 *arrIntersections*; if *tnode -> bnode* != Null. Then 转到(2),继续处理右兄弟节点; Else 转到(6)。

(6)If *tnode -> pnode* == *TreeNode*. Then 算法结束 Else 转到(5)

在这个算法中,根节点大小有产生的偏置曲线的 X 轴方向的最大宽度, Y 轴方向的最大宽度决定:

$TreeNode.width = \max\{|X_{left} - X_{right}|, |Y_{up} - Y_{down}|\};$

EndWidth 是衡量 *tnode* 有自交点的一个标准,所以它的取法也很重要。在算法中,我们取它为 0.001 个单位。

2 消除自交

在算法1中,我们找出了不管是全局还是局部的自交点。接下来的算法来剔除自交环和全局自交。在我们这种算法中,不需要辨别是局部自交还是全局自交。只需要把自交点对应到曲线上的两点参数 t 的值,再对这一系列的 t 值按大小进行排列。依次取出两点,并判断这两点之间的中间点到原曲线的距离是否小于偏置曲线。

算法2 消除自交

(1)输入原曲线 $C(t), t \in [0, 1]$, 要偏置的距离 d 。

(2)调用算法1,探测偏置曲线,并取回自交点数组 *arrIntersections*。

(3)对自交点数组 *arrIntersections* 进行处理。

If $|C_d^0(t) - C(t)| < |\vec{N}(t)| \cdot d, t \in (t_0, t_1)$ then 剔除

偏置曲线这部分。

3 实例

我们首先对图1中的偏置进行处理,其处理的结果如图2所示;一个比较复杂的例子如图3所示。



图2 简单偏置曲线的处理过程



图3 复杂偏置曲线的处理过程

本算法效率高,省内存,但只适合于二维的自由曲线。以后的工作,是要从二维的扩展到三维至多维空间。

参考文献:

- [1] ELBER G, LEE IK, KIM MS. Comparing offset curve approximation methods[J]. IEEE Computer Graphic Application, 1997, 17(3): 62 - 71.
- [2] PHAM B. Offset curves and surfaces: a brief survey[J]. Computer-Aided Des, 1992, 24(4): 223.
- [3] MAEKAWA T. An overview of offset curves and surfaces[J]. Computer-Aided Des, 1992, 31(3): 165 - 73.
- [4] PATRIKALAKIS NM, BARDIS L. Offsets of curves on rational B-spline surfaces[J]. Engng comput, 1989, (5): 39 - 46.
- [5] SARMA R, DUTTA D. The geometry and generation of NC tool paths[J]. ASME J Mech Des, 1997, 119(2): 253 - 8.
- [6] 张李超, 韩明, 黄树槐. 快速成形激光光斑半径补偿算法的研究[J]. 华中科技大学学报, 2002, 30(6): 16 - 18.
- [7] GURBUZ AZ, ZEID I. Offsetting operations via closed loop ball approximation[J]. Computer-Aided Des, 1995, 27(11): 805 - 10.
- [8] KIMMEL R, SETHIAN JA. Computing geodesic paths on manifolds[J]. Proceedings Nat Acad Sci USA, 1998, 95(15): 8431 - 5.
- [9] MEMOLI F, SAPIRO G. Fast computation of weighted distance functions and geodesics on implicit hyper-surfaces[J]. Journal of Computational Physics, 2001, 173(2): 730 - 764.
- [10] CHOI BK, PARK SC. A pair-wise offset algorithm for 2D point-sequence curve[J]. Computer-Aided Design, 1999, 31(12): 735 - 745.
- [11] LEE IK, KIM MS, ELBER G. Planar Curve Offset Based on Circle Approximation[J]. Computer Aided Design, 1996, 28(8): 617 - 630.
- [12] HJALTASON GR, SAMET H. Speeding up construction of PMR quadtree-based spatial indexes[J]. VLDB Journal, 2002, 11(2): 190 - 137.
- [13] 王德人, 张连生, 邓乃扬. 非线性方程的区间算法[M]. 上海: 科学技术出版社, 1987.