

基于 HPGL 文件的图元优化排序^①

蔡 明

(暨南大学 信息科学与技术学院, 广州 510632)

摘 要: HPGL 文件的图形元素是以绘图人员设计的先后顺序记录于文件中, 导致以此为基础进行的绘图或加工过程的无效行程大量增加且效率低下, 且 HPGL 文件曲线图元均自动插补为大量小线段, 以简单查找排序的方法处理耗时长. 针对此问题, 以 HPGL 文件记录的图形元素为对象, 提出了一种利用哈希思想动态改变图元始末点及图元绘制顺序的优化算法, 使其转化为 DXF 文件图元优化问题, 进而实现了 HPGL 中图形元素及其控制点的优化排序并生成优化后的 HPGL 文件, 使得以此文件为基础的绘图或加工路径得到优化.

关键词: HPGL 文件; 图元; 哈希算法; 轨迹优化

Optimization Sequencing of Graphic Elements Based on HPGL File

CAI Ming

(School of Information Science and Technology, Jinan University, Guangzhou 510632, China)

Abstract: In HPGL file, the graphic elements are recorded according to the sequence made by the designers. A lot of useless journey is made by using the file and it is inefficient in drafting or machining process. In order to solve the problem, an algorithm based on the graphic elements whose start point and endpoint could be dynamically changed Along the drafting path was put forward. The sequence of the graphic elements and its reference points could be optimized by Using the algorithm, the question is converted to DXF, and the drafting or machining path could be reduced based on the optimized HPGL.

Key words: HPGL file; graphic element; Hash algorithms; optimizing path

HPGL 是惠普图形语言的标准化版本. 惠普公司设计该语言的目的是提供可被多种外部设备支持的一致功能, 从而减轻程序开发工作. 目前大量的 CAD/CAM 软件均支持 HPGL 格式文件的数据输出, 所有数据均以点结合向量指令的形式给出, 指令如表 1. 这种格式的数据, 将任意曲线均自动插补, 输出小线段的形式, 在满足插补精度的应用下, 方便了后续的程序开发工作. 但由于 HPGL 文件对图元采用的是先设计先记录的顺序, 设计人员在设计过程中的无序性使得 HPGL 文件将以此实际设计顺序存储图元, 使得以此 HPGL 文件为基础转换而成的绘图或加工代码将会使绘图或加工设备按照此设计顺序进行实际操作, 导致绘图或加工路径的无序性, 在工作过程中空行程增多, 效率降低, 设备磨损加快, 经济成本增加. 针对此

类问题有文献提出了对绘图或加工过程中路径的优化. 穆海华针对大型数控绘图机, 对输入的 DXF 文件进行解析, 其中对曲线进行插补计算, 生成绘图轨迹并优化^[1]. 莫蓉不仅对导入的文件解析后的加工轨迹进行优化, 而且还运用 OpenGL 技术进行了运动轨迹的仿真, 使运动轨迹优化结果可预览^[2]. 胡胜红则对国内切雕加工系统中软件功能进行研究与实现, 设计了适合激光加工软件的 DXF 文件接口程序, 包括 DXF 文件中面域图元的提取与优化算法, 多图元复杂 DXF 文件的图元优化排序算法(最短距离法), 针对激光标刻过程中螺旋路径填充加工, 研究了多边形中轴生成算法^[3]. 吕欣泽则在激光雕刻中提出空间雕刻的轨迹优化. 陈树敏等则在排样切割中进行轨迹优化^[4]. 但因其采用的优化算法均针对少量图元, 并且实际使用前

^① 基金项目: 广东省产学研项目(2011B090400345)

收稿时间: 2013-01-29; 收到修改稿时间: 2013-03-11

还需进行插补; 本文研究 AutoCAD 设计的 HPGL 文件, 所有曲线均自动进行插补为小线段, 为后续工作提供便利, 但由于插补之后图元数量猛增, 采用最短距离, 算法复杂度为 $O(n^2)$ 的算法耗时不可忽视; 因此对 HPGL 文件的路径优化的耗时问题进行优化是十分必要的。

表 1 向量组指令

PA	绝对绘图	PD	落笔
PR	相对画弧	PU	提笔

1 基于HPGL文件结构及离散图元信息获取

文件解析: HPGL 格式的数据保存为 PLT 文件, 且数据无换行的连续保存, 每个指令以分号隔开。数据指令的含义模拟一个人的画图的动作, PU 抬笔, PA 移动到指定点, PD 落笔, PA 移动到另一个指定点, 这个过程就画出了一条直线, 连续画线可以继续 PA, 更换直线的起始点则需要先 PU, 再 PA 指定; 可知只有在 PD 之后 PU 之前的所有 PA 均是画图; 每次一个完整 PLT 文件的作图开始都会 PU, PA0,0 归零。根据 HPGL 的定义, 文件解析的流程如图 1。

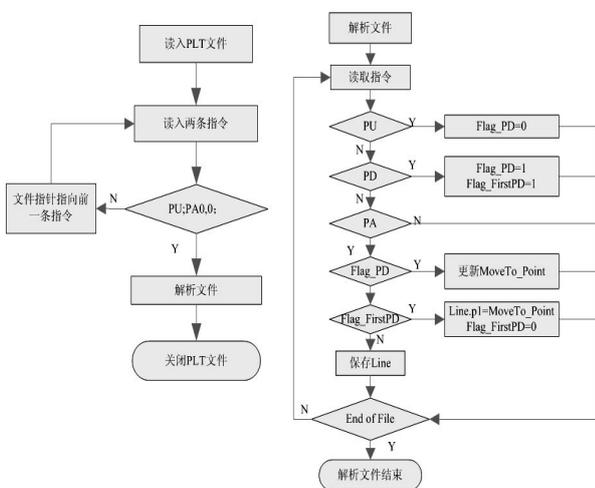


图 1 PLT 文件解析流程图

2 圈查找

圈(Contour)被定义为由连续相连接的按逆时针或顺时针顺序排列组成封闭图形的图元集合, 也是路径优化的基本单位。一个圈存在封闭与不封闭两种, 面积最大的封闭圈定义为外圈, 其他的圈定义为内圈, 而且根据常见需求定义, 一份加工 PLT 文件中, 有且仅有一个外圈, 外圈一般包含若干内圈, 且内圈中不

包含任何图元。

因为绘图或加工路径分为封闭与非封闭两类, 则定义封闭的图形为封闭圈, 绘图或加工的外轮廓即最大的封闭图形为外圈, 其余的为内圈, 则只有外圈内部含有圈, 内圈内部无任何图形。经过提取的数据中含有封闭圈与不封闭圈。

圈查找问题可以类比为最短路径查找的一个特例, 其中重要工作在寻找当前图元的连接图元, 判断两个图元是否相连接, 即判断是否存在一个图元, 其某一端点到此当前待连接图元端点距离足够小($<1.0e-9$)且为唯一最短路径。

根据绘图或加工常见需求, 描述图元的一些性质:

- 1) 如果一个图元存在一个封闭圈中, 那么这个图元仅能存在这一个封闭圈中, 并且左右端点有且仅有一个图元端点与其连接。
- 2) 如果若干相连接的图元没有组成封闭的圈, 那么这些若干图元中的每一个图元永远不可能存在某个封闭圈中, 即不能成封闭圈的图元。

文献[5]中查找连接图元的过程中, 遍历图元利用两路插入, 算法复杂度为 $O(m*n)$, m 为图元数量, n 为图组数量, 但其最坏时间复杂度接近 $O(m^2)$, 即 n 接近 m 的情况。马凯等提出利用算法复杂度为 $O(n^2)$ 的插入排序的思想将所有拟合的小线段进行比较插入^[6], 将相连接的小线段在链表中局部连续, 优化算法解决了雕刻过程中起落刀次数过于频繁, 但适用于少量图元。本文针对大量图元的连接查找提出改进, 采用双哈希连接图元预处理算法复杂度 $O(kn), k \ll n, n \leq m$ 。

图元二维哈希表的建立: 如图 2, 通过一次遍历图元的端点坐标, 找到包含所有图元的最小矩形包络, 得到矩形长 L 、宽 K , 取待哈希处理的区域为长 $L+1$, 宽 $K+1$, 图 2 中的黑色虚线将待处理区域分割, 则每一个端点均能找到唯一对应的区域, 此例中将长、宽均进行 100 等分, 则定址长宽分度值为 $hdeta=(K+1)/100$, $ldeta=(L+1)/100$; 所以左端点 $ptLeft$ 哈希直接定址函数为 $m=(int)((ptLeft.x-dleft)/(L+1/100))$, $n=(int)((ptLeft.y-dbottom)/(K+1/100))$;

由于一条直线由两个端点组成, 即左端点, 右端点, 即为了保证直线图元的端点-直线一一对应信息的完整性, 直线分别利用左端点建立一张左哈希二维表, 右端点建立右哈希二维表, 为了将同一区域的端点所

代表的图元聚类, 所以哈希冲突处理采用链地址法, 数据结构定义如下:

```

typedef struct _HASH_STRUCT_
{
    GroupNode *pdata;
    struct _HASH_STRUCT_* next;
}Hash_cell,*Hash_map;
Hash_cell pmapL[100][100];
Hash_cell pmapR[100][100];

```

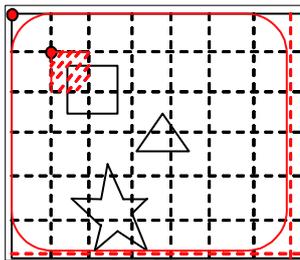


图 2 哈希处理示意图

圈查找过程:

1) 合理性校验: 查重, 删除重复的图元(即重复的小线段)——单哈希表查重(以直线的左端点建立哈希表). 利用同一区域的直线图元链表进行比较, 查看是否存在重复图元.

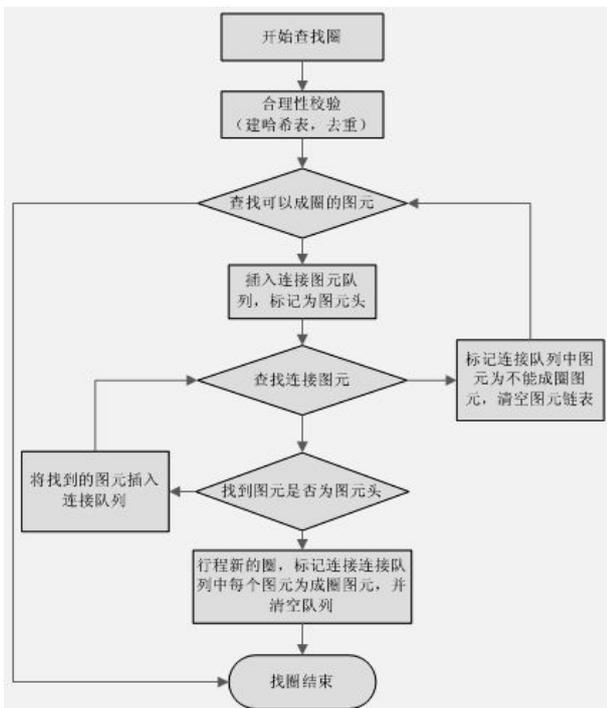


图 3 圈查找流程图

2) 双哈希表找圈(分别建立左端点哈希表, 右端点哈希表): 找圈过程主要为查找图元直线端点的连接点的过程. 则直线端点连接的另一条直线的端点可能存在两种情况, 连接另一条直线的左端点或者右端点. 则要既要遍历左哈希表, 又要遍历右哈希表; 将第一个图元直线直接放入圈队列中, 下一步寻找连接的直线图元, 找到则添加到圈队列中, 如果找到的直线图元与队列头的直线图元是同一个, 则找到的圈为封闭圈, 将圈标记为封闭并添加到圈链表中, 反之找到为未封闭圈, 标记为非封闭圈添加到圈链表.

3 轨迹优化

经过圈查找的处理, 已将 HPGL 大量的插补线段组合成独立的连续的圈, 以圈为加工单位, 待加工的圈数量则骤降, 圈的数量不大于同一文件 AutoCAD 生成 DXF 格式中图元的数量, 此时问题则可看做 DXF 文件优化的问题, 采用文献[6-9]中提到的最短距离法, 将待加工的圈利用最短距离算法优化, 即可找出优化后的加工路径. 效果如图 4, 具体的 DXF 文件图元路径优化算法证明可见文献[6].

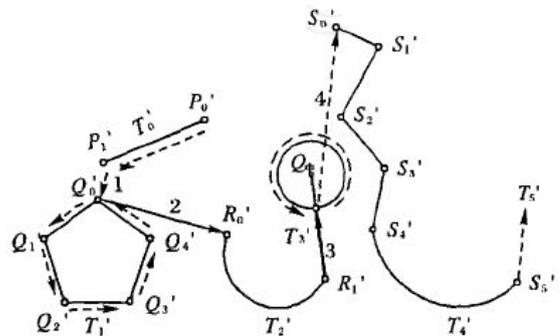


图 4 优化后的图元及其中控制点绘制顺序

4 算例分析

HPGL 自动将曲线插补为小线段, 在满足一般精度下, 直径 12mm 的圆通常插补为 300 条左右的小线段, 所以算例中曲线的数量直接决定了图元总数的数量, 图 5 为图元数量为 3000 左右的样例, 采用顺序处理与哈希处理的耗时结果见表 2. 并且分别对图元数量为 500、10000 进行了处理对比. 从表 2 中的结果可知, 采用顺序处理的方法对大量的图元的耗时是远达不到交互式响应要求的, 而采用哈希进行处理的耗

是可以接受的.

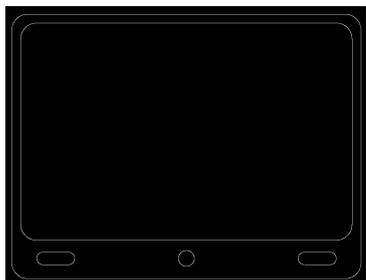


图5 图元数量为3000的算例

表2 优化效率表

图元数量	优化前耗时	优化后耗时	优化效率
500	0.047	0.0001	99%
3000	0.422	0.016	96%
10000	6.17	0.109	97%

5 结语

通过上述分析,本文所提出的基于HPGL优化算法.在耗时与路径优化都是可行且有效的.该算法针对HPGL的大量图元进行处理,在最坏情况下仍可以用空间换时间思想,建立大张哈希表,是整体平均复杂度为 $O(n)$.这在实际的生产过程中是能被操作人员所接受的.通过对HPGL的处理,将问题转化为基于DXF文件的路径优化问题,采用文献[6]中的方法即可

减少大量无效空行程,提高了绘图效率,降低了经济成本.

参考文献

- 1 穆海华,彭芳瑜,陈吉红.基于DXF的大型数控平面绘图机CAM系统.机床与液压,2003,107-109.
- 2 莫蓉.基于DXF的绘图轨迹优化及仿真系统研究实现[学位论文].西安:西北工业大学,2006.
- 3 胡胜红.基于图形计算几何技术的激光加工优化算法研究[学位论文].武汉:华中科技大学,2007.
- 4 吕欣泽.基于DXF文件的激光雕刻系统设计与实现[学位论文].天津:天津师范大学,2010.
- 5 陈树敏,刘强,方少亮等.DXF排样切割中的应用.计算机应用与软件,2012,143-146.
- 6 龚清洪,常智勇,莫蓉,等.基于DXF文件的图元优化排序.计算机应用,2006,26(1):169-171.
- 7 胡胜红.对面域作图的DXF文件优化激光加工路径.工程图学学报,2010,(6):106-110.
- 8 马凯,杨泽林,吕静.基于DXF文件的CAD/CAM刀具路径优化与生成.机床与液压,2011,39(10):39-42.
- 9 甘明,陈小亮,张科威.基于DXF的数控切割加工优化算法的研究与实现.煤矿机械,2010,31(11):130-132.

(上接第167页)

- 4 修保新,刘忠,张维明,阳东升.基于信息粒化理论的主体间任务分配方法.国防科技大学学报,2007,29(3):71-75.
- 5 侯方国,胡圣武.模糊信息粒化理论在空间信息系统地位的探讨.测绘与空间地理信息,2004,27(3):14-16.
- 6 Chapelle O, Vapnik V, Bousquet O, Mukherjee S. Choosing multiple parameters for support vector machines. Machine Learning, 2002, 46(1-3).
- 7 Vladimir C, Ma YQ. Practical selection of SVM parameters and noise estimation for SVM regression. Neural Networks, 2004, 17(1):113-126.
- 8 Keerthi SS, Lin CJ. Asymptotic behaviors of support vector machines with gaussian kernel. Neural Computation, 2003, 15(7):1667-1689.
- 9 Lin HT, Lin CHJ. A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. Taipei: Department of Computer Science and Information Engineering National Taiwan University, 2005.
- 10 严晓明.基于优化GA属性约简的上证指数预测.福建师范大学学报(自然科学版),2011,27(5):29-33.