

工程图自动矢量化算法的设计与实现

宋晓宇 王永会

(沈阳建筑工程学院计算机系, 沈阳 110015)

摘要 工程图自动矢量化是CAD领域中远未解决的关键技术之一. 通过对工程图矢量化课题的深入研究, 给出了一种将二值点阵图象快速转换为直线、圆及圆弧的自动矢量化算法. 这种算法保证处理的高速度、灵活性和准确性.

关键词 矢量化 圆弧 圆 图象

中图分类号: TP391.72 文献标识码: A 文章编号: 1006-8961(2000)01-0066-04

The Design and Implementation of the Automatic Vector Algorithm About Engineering Drawing

SONG Xiao-yu, WANG Yong-hui

(Department of Computer Sciences, Shenyang Arch. and Civ. Eng. Inst., Shenyang 110015)

Abstract The automatic vectorization about the engineering drawing is one of the urgent need solution key technology. This paper presents an automatic vector algorithm for converting rapidly a binary matrix image into the straight lines, circles and arcs, based on deeply study on the vectorization of the engineering drawing. Experimental results of complex line drawing confirm performances of these algorithms.

Keywords Vectorization, Circular arc, Circle, Image

0 引言

工程图扫描输入及自动矢量化是CAD领域中远未解决的关键技术之一. 工程图可以通过扫描仪快速、准确地输入计算机, 获得的信息是二值点阵图象信息, 由于信息量大, 如何将工程图有效地在计算机中存档、编辑、绘制新图成为一个新的研究课题. 采用模式识别技术, 将二值点阵图象信息通过预处理(去噪声)、细化处理、矢量化处理等环节转换为矢量信息存储是本课题研究的主要目标.

工程图扫描输入计算机, 经过细化处理后, 变成只有单点线宽的骨架图形, 但仍是一个简化了的二值图象, 占有大量的存储空间. 因此, 必须将图形的二值点阵信息转换为矢量信息, 并确定其图元, 这一

过程称为矢量化处理.

以往的矢量化算法, 多数只对直线进行处理^[1-3], 即使有对圆及圆弧的矢量化处理, 一般采用两种方法: 一是采用直线逼近圆及圆弧^[4]. 这种方法使矢量化后的图形中的图元数量明显增加; 二是采用手工方法对圆及圆弧矢量化^[1,5], 就是先手工选取要量化的圆或圆弧, 再根据圆及圆弧的特征进行矢量化, 这种方法会降低系统的处理速度, 处理起来也比较繁琐. 因此, 对于信息量较大的工程图进行矢量化处理时, 需要寻找一个算法简单, 占内存少, 运算速度快, 能真实地反映图纸原貌, 失真少, 并尽可能提取完整图元的有效方法. 本文在研究、分析以往的算法的基础上, 提出了一种适用于工程图中直线、圆及圆弧的自动矢量化算法, 通过实验, 取得了较好的效果.

1 自动矢量化算法

工程图由一些基本图元组成, 如线段、任意多边形、圆及圆弧等. 在这些图元中, 线段是最基本的, 圆及圆弧和各种曲线均可以由若干条线段逼近而成. 所以, 自动矢量化算法首先考虑得到由若干个图元组成的图形矢量信息. 每个图元由一条或若干条线段信息组成, 即始点、终点坐标值, 或始点、拐点(一个或多个)、终点坐标值. 为此, 自动矢量化算法的步骤如下:

(1) 建立一个中间数据文件, 存放被分离出来的各图元和点的坐标值.

(2) 定义每个像素在 8 个邻接点方向上的优先级, 即跟踪方向. 图 1 将 P_0 、 P_2 、 P_4 、 P_6 定义为直接邻点, 将 P_1 、 P_3 、 P_5 、 P_7 定义为间接邻点, 并规定跟踪惯性规则.

P_5	P_6	P_7
P_4	P	P_0
P_3	P_2	P_1

图 1 邻接点的定义

(3) 按顺序搜索一图元的始点. 如按照从上到下、从左到右的顺序搜索, 首先出现的一个像素点为 1 的位置(二值图象中的图形部分用 1 表示, 背景部分用 0 表示), 并把这个位置的行、列号作为其坐标值记录到数据文件中.

(4) 沿始点出发, 在它的周围 8 个邻接点中选定跟踪方向, 即查找像素值为 1 的后继点. 并在同方向上作连续跟踪, 直到不存在后继点为止. 把最末后继点的坐标值顺序记录到数据文件中.

(5) 第(4)步完成后有两种可能, 一是这个坐标点为线段的中间拐点; 二是为线段的终点. 对前种情况, 把这个拐点记录在数据文件中, 并把该点看作新方向的始点, 返回到第(4)步, 继续跟踪新方向的末点. 其结果是在一个图元的数据区中, 顺序记录了一组连续的始点和拐点的坐标值序列. 对后一种情况, 表示本图元跟踪结束. 把最末后继点作为图元终点, 记录在数据文件中, 同时结束这一图元的记录. 为了防止图元中的线段重复跟踪, 删除已跟踪过的线段.

(6) 返回到第(3)步, 按同样的方式搜索下一个图元的始点、拐点, 直到终点, 并把这些坐标值记录到新的图元项上. 上述步骤循环往复, 直到无图元搜

索为止. 于是完成了对整个图形的自动跟踪, 一个图形的矢量数据文件被建立并完成.

(7) 圆及圆弧的矢量化.

(8) 直线的合并.

2 算法中几个关键点的处理

2.1 图元始点的搜索

对图元始点的搜索, 主要有两种方法: 一种是就近搜索方法, 即以前一个图元的终点为开始, 从上至下、从左至右进行搜索. 该方法会使一些像素点搜索不到, 从而产生图元的不完整性. 另一种是每次从上向下、从左向右的搜索方法, 该方法能使所有的像素点被搜索到, 保证了图元的完整性, 但会增加每个始点搜索的次数. 通过对上述两种方法的分析, 发现各有优缺点, 所以, 本文采用将上述两种方法融合的方式, 即搜索下一个图元始点时, 由前一个图元的始点开始, 由上至下、由左至右搜索像素值为 1 的点, 来确定图元的始点. 这样使图象中的每一个像素点都能被搜索到, 既减少了迭代次数, 又保证了图元的完整性.

2.2 跟踪方向及惯性规则

对始点或拐点, 如何选择在 8 个跟踪方向上的优先级将影响线段的走向及图元的完整性. 本算法中规定跟踪方向规则为: 由始点出发时, 先直接近邻, 后间接近邻. 由图 1 上 8 个点表示, 先后次序为: P_0 、 P_2 、 P_4 、 P_6 、 P_1 、 P_3 、 P_5 、 P_7 . 对拐點選取新方向, 规定了跟踪惯性规则为: 原来的步进方向为下一个像素的步进方向优先选用方向. 这样, 使跟踪过程具有一定的惯性, 以保证取得一条完整线段, 使图元尽可能保持其完整性.

2.3 删除点的处理

在上述跟踪过程中, 当一线段在跟踪结束时, 为避免以后重复跟踪而被删去, 结果使在线段的交叉点上, 对尚未被跟踪的直线产生断线现象, 使一条完整的线段被分割成多条彼此分离的线段. 因此, 在本文算法中, 对每个要删除的点 P , 计算 $N = \sum_{i=0}^7 P_i$, 若 $N > 2$ 则该点为交叉点, 保留此点, 否则删去此点.

2.4 圆及圆弧的矢量化

一个圆或圆弧可以用一个内接或外接的 n 边形来近似表示, n 值越大, n 边形就越接近于圆或圆弧. 工程图纸中的圆及圆弧被跟踪矢量化后, 一般都由

若干条首尾相接的线段组成,如图2(a)是圆被矢量化的示意图,图2(b)是圆弧被矢量化的示意图.

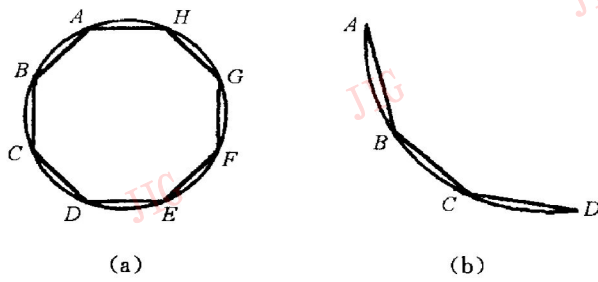


图2 圆及圆弧的矢量化示意图

本文算法中矢量化出来的图元分为两类:一类是不含拐点的图元,另一类是含有拐点的图元.圆或圆弧矢量化后,应该由一组首尾相接的线段构成,因此在提取圆及圆弧时,只对含有拐点的图元进行判断.判断某组线段是否构成一个圆或圆弧的基本步骤如下:

(1) 取前3条线段 L_1, L_2, L_3 .

(2) 分别计算 L_1 与 L_2, L_2 与 L_3, L_3 与 L_1 的夹角 α, β, γ 和三条线段的长度 d_1, d_2, d_3 .

(3) 若满足下述两个条件: ① $|\alpha - \beta| \leq T_\alpha$ (T_α 为角度阈值) 且 $(\gamma < \alpha) \&\& (\gamma < \beta)$; ② $|d_1 - d_2| \leq T_d$ 且 $|d_2 - d_3| \leq T_d$ 且 $|d_3 - d_1| \leq T_d$ (T_d 为距离阈值), 则这3条直线属于一个圆或圆弧内逼近的三条直线. 否则转步骤(6).

(4) 根据 L_1, L_2, L_3 的垂直平分线计算圆的圆心 c_i 和半径 r_i .

(5) 设 $L_1 = L_2, L_2 = L_3$, 取下一条线段设为 L_3 , 返回步骤(2).

(6) 分别计算半径的均值 $R = \sum_{i=1}^N r_i / n$, 圆心的均值 $C = \sum_{i=1}^n c_i / n$, 圆心的偏差 $\Delta C_i = |c_i - C|$ ($i = 1 \dots n$).

(7) 设阈值 $T_n = R/3$, 如果 ΔC_i 小于 T_n 的数目大于 $n/2$, 则将该组线段识别为弧. 若第一条线段与最后一条线段首尾相接, 则该组线段识别为圆.

计算两条直线间的夹角的方法如下:

设已知两条直线对应的向量分别为 $(a + bi), (c + di)$, 两向量之间的夹角为 α , 根据公式有:

$$\alpha = \arccos \left| \frac{a \times c + b \times d}{\sqrt{(a^2 + b^2) \times (c^2 + d^2)}} \right|$$

直线的向量可以这样确定, 设直线 AB , 其中 $A(x_1, y_1), B(x_2, y_2)$, 设其向量为 $a + bi$, 则有 $a = x_2 - x_1, b = y_2 - y_1$, 即该直线的向量为 $(x_2 - x_1) + (y_2 - y_1)i$.

2.5 直线的合并

由于图象质量、细化处理及直线与其它线或圆弧相交等原因, 使一些本来连续的直线识别后得到多条分隔开的直线段. 为了将这些直线段合并为一条直线, 我们采用的方法是, 扫描每一条直线, 对该直线分别向上、向下延伸, 直到不能延伸为止. 如图3中的直线 AB 和 CD , 应合并为一条直线.

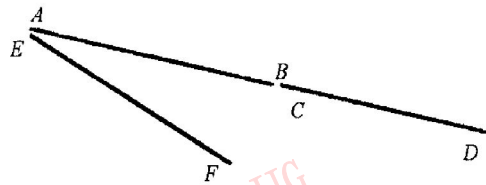


图3 直线的合并示意图

首先向上延伸, 设当前处理的直线为 $CurLine$.

(1) 如果 $CurLine$ 的上方没有与之邻接的图元, 则结束向上延伸; 否则取这个邻接图元为 $AdjLine$.

(2) 如果 $AdjLine$ 的斜率与 $CurLine$ 的斜率相近, 且 $AdjLine$ 的下端点与 $CurLine$ 的上端点距离小于阈值 T , 则直线向上延伸, 并重新计算延伸后的斜率, 设 $CurLine = AdjLine$.

(3) 如果 $CurLine$ 的上方仍有与之邻接的图元, 则取下一个图元作为 $AdjLine$, 转(2); 否则结束向上延伸.

向下延伸的方法与向上延伸的方法基本相同, 这里不再给出具体算法.

3 应用实例

本文提出的算法可以对工程图纸中的直线、圆及圆弧进行自动矢量化处理, 从而提高了矢量化处理的速度. 按照本文描述的算法, 对某工程图纸的一小部分进行矢量化, 得出了比较满意的效果(见附图1、2).

4 结束语

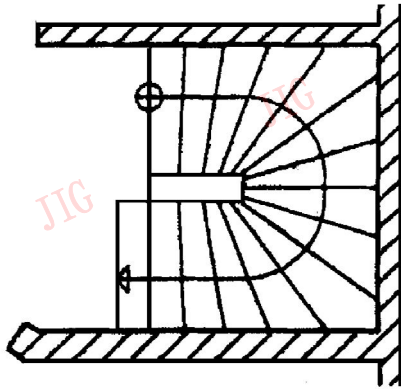
本文提出了一种对工程图进行自动矢量化处理的算法, 在微机上用 Watcom C++ 10.0 编程实现. 该算法避免了复杂的数学计算问题, 使整个识别过程简洁、快速、准确, 适用于对各种工程图进行矢量化处理. 本文工作仍在继续研究和完善中, 通过进一步的研究, 会使这一新的研究成果更加完善和实用化.

参 考 文 献

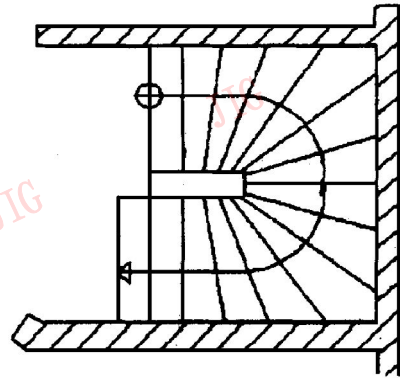
- 1 王德英, 李新友, 唐泽圣, 黄汉文. 图纸图象上线条的交互拾取. 计算机学报, 1997, 20(10): 925~ 932.
- 2 谭建荣, 彭群生. 基于图形约束的工程图扫描图象直线整体识别方法. 计算机学报, 1994, 17(8): 561~ 569.
- 3 王金鹤, 欧宗英, 夏晓东. 工程扫描图象的直线整体识别算法. 中国图象图形学报, 1998, 3(11): 912~ 917.
- 4 王建华, 童秉枢. 工程图形的自动跟踪识别算法. 微型计算机应

用, 1993, 14(1): 31~ 33.

- 5 武仲科, 焦海星, 戴国忠. 一种线段和圆弧的逼近方法及其在工程图纸矢量化中的应用. 计算机辅助设计与图形学学报, 1998, 10(4): 328~ 332.
- 6 王 刚, 林贻佳, 宋乐乐. 直线及圆弧矢量化的统计算法. 中国图象图形学报, 1998, 3(4): 273~ 276.
- 7 郑南宁. 计算机视觉与模式识别. 北京: 国防工业出版社, 1998.
- 8 Donald Hearn, M Pauline Baker. Computer Graphics. 北京: 电子工业出版社, 1998.



附图 1 光栅图象



附图 2 矢量化结果



宋晓宇 1988年毕业于哈尔滨工业大学, 获硕士学位, 现为沈阳建筑工程学院计算机系副教授. 主要研究领域为模式识别和图象处理、GIS 管理信息系统与网络数据库应用.



王永会 1993年毕业于沈阳建筑工程学院计算机系, 获学士学位, 现为沈阳建筑工程学院现代教育技术中心讲师. 主要从事工程图纸的自动输入与模式识别、图象处理、网络数据库应用等领域的研究工作.