扫描图象曲线轮廓关键点的提取及其处理

王金鹤
(青岛海信集团技术中心，青岛 266071)

摘 要 针对工程图样中线段的识别和提取问题，提出了一种提取扫描图象曲线轮廓关键点的算法和相应的提取条件准则。该算法可自动逐行检测扫描行中的黑游程，当相继行中的黑游程满足一定的条件准则时，即可提取出相应的黑游程边界点，并形成关键点。依据这些关键点，利用 B 样条曲线理论来拟合曲线，最终即可实现对图象曲线的提取。该算法不仅速度快，抗噪声性能强，且提取识别也取得了满意的效果。

关键词 扫描工程图 轮廓跟踪 B 样条函数
中图法分类号：TP391 **文献标识码**：A **文章编号**：1006-8961(2001)07-0699-04

An Algorithm for the Extraction of Determinant Contour Points of Lines in Scanned Drawings

WANG Jin-he
(*Hisense Engineering Institute, Qingdao 266071*)

Abstract The conversion of drawing image into vector form is of important in many fields, especially in the field of electronic and mechanical applications. In this paper, a new approach for extracting determinant contour points of lines in scanned engineering drawings is proposed. This method can not only obtain determinant contour points placed in line contour of the drawing image but also process these points automatically. The basic idea of proposed algorithm is totally different from that of traditional algorithms, it can remedy some deficiencies which are suffered from traditional algorithms and executes successfully without thinning. This algorithm detects black runs on scanning or consecutive scanning lines automatically, and then extracts determinant contour points from scanned engineering drawings, the main idea of the detecting algorithm is that: first, black runs on current scanning line are detected automatically, and determinant contour points are extracted by detecting the edge points of both current black run and the next black run which is on consecutive scanning line if the relevant condition holds. Second, based on the determinant contour points, the extraction of lines from the original drawing image will be done easily by using B-spline theory. Experimantal results indicate that the algorithm has higher speed and other advantages.

Keywords Scanned engineering drawing, Edge tracking, B-spline

0 引 言

工程图样进行处理和识别的首要任务就是对扫描图样中的线段进行分段和识别[1~5]。线条分段之间的特征点称为关键点。对线段进行分段和识别的常规做法是先对图线进行细化，然后对细化线段进行分段和识别。所谓细化就是把线条的宽度减为单象素宽，但由于细化过程抗噪声性能较差，故细化的结果将影响后继处理阶段结果的正确性。此外，若对整张图象进行细化处理，处理的工作量将很大。本文提出一种确定扫描图象曲线关键点的新算法。该算法速度快，抗噪声性能强，对原图象不需作细化和其他预处理，直接对一个扫描行中各个黑游程图象段

收稿日期：2000-03-30；改回日期：2000-06-12

进行处理,即可直至图象结束.在预先给定的阈值基础上,该算法能自动调节已生成关键点的稀疏程度,且对线条宽度没有限制,效果较佳,最后,对提取出的关键点进行拟合,重构曲线.

1 域的数据结构及几个基本模块

1.1 定义

黑游程:由光栅扫描图象某扫描行上连续黑象素构成的区域称作黑游程.

关键点:图象边界变化处的坐标点.

域:线条轮廓关键点由链表表示,链表的结点称作域.

这里,对图象的检测方式是自上至下、自左至右进行的,因此域的连续生成过程即是轮廓关键点的搜索和生成过程.

1.2 域的数据结构

用 C++ 表述的域的数据结构定义如下:

```
struct node {
  int  x1,y1,x2,y2;
  int  NextEdgePointX1,NextEdgePointX2,NextEdgePointY;
  unsigned char  LeftEdgeMode, RightEdgeMode;
  short int  ChangeCounter;
  struct node *Next;
  struct node *Prev;
}
```

在上述数据结构中,x1,y1,x2,y2 为待输出的左右两个关键点的坐标分量;NextEdgePointX1,NextEdgePointX2,NextEdgePointY 为 node 结构中被检测到的后继边界点;LeftEdgeMode,RightEdgeMode 为图象的两边边界方向变化状态码,码的取值可为 −1,0 或 1;ChangeCounter 为边界方向发生变化后的扫描行累计数.

2 线条轮廓关键点判别算法

2.1 域的生成原理

用 B 表示第 i 行上某黑游程.$B(X_i^1,Y_i)$ 和 $B(X_i^2,Y_i)$ 分别表示第 i 行黑游程 B 的左、右端点坐标,X_i^1 和 X_i^2 分别表示第 i 行黑游程 B 的左、右端点横坐标,Y_i 表示第 i 行黑游程 B 的纵坐标,N 表示结点链表中某结点(它包含两个关键点).为简便起见,用符号 L_{X1},L_Y 表示被检测的后继左边界点坐标,用符号 L_{X2},L_Y 表示被检测的后继右边界点坐标(如图 1 所示),δ_1 和 δ_2 分别表示两个偏差阈值.

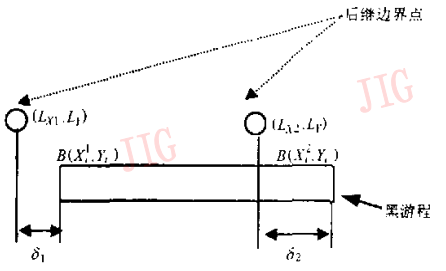


图 1 关键点的生成示意图

若满足邻接条件

$$Y_i = L_Y + 1$$

则域的生成条件为满足以下条件之一:

(1) $|X_i^1 - L_{X1}| > \delta_1$;

(2) $|X_i^2 - L_{X2}| > \delta_2$;

(3) 边界方向有变化(此条件由边界方向判别函数所返回的值确定)

对于链表中某 N 结点,若与正在检测的某黑游程邻接(如图 1),根据 δ_1,δ_2 等偏差阈值,判断是否满足邻接条件及域的生成条件 1 或 2,若符合条件,则生成新域结点 N(即生成新的候选关键点),否则,调用边界方向判别函数 $D(T,X_i^1,X_i^2,Y_i)$,即判断是否符合邻接条件及域的生成条件(3),若符合条件,则生成新域.

2.2 几个模块

线条轮廓关键点算法调用了黑游程生成函数(bool FindBlackRun(int LineNum//行号))、域的生成函数(bool GenerateBlock(struct node * N, int x1,int x2,int y))及边界方向判别函数(bool Directions(struct node * Node, int x1,int x2,int y))等模块.

黑游程生成模块的功能为:读入原始图象某一行,即能输出原始图象中该行的所有黑游程.

域的生成函数(N_1 = GenerateBlock(struct node * N_2, int x1,int x2,int y)),其中,N_1 表示该函数生成的新结点指针,即,函数返回的结果;参数 N_2 表示链表域中某结点指针;x1,x2,y 分别表示某黑游程的左边横坐标、右边横坐标及纵坐标.该函数的功能为:输入已知结点和黑游程坐标,不断后移检测行,结合后继边界点坐标分量和边界方向变化状态码进行判别,若满足判别条件(见图 1),则生成当前两个被检测的后继边界点的坐标分量,并输出为

新的关键点.

边界方向判别函数(bool Directions(struct node * N, int $x1$,int $x2$,int y)),其中,参数 Node 表示链域中某结点指针;$x1$,$x2$,y 分别表示某黑游程的左边横坐标,右边横坐标及纵坐标.该函数的功能为:根据某黑游程相对上一行黑游程的边界走向,结合链域中记录的边界方向变化状态码(LeftEdgeMode, RightEdgeMode),判别边界方向是否发生了变化,若发生了变化,则激发扫描行计数器计数,当计数器累计达到预定的次数时,即可将已纪录的候选关键点转化为新的关键点;若计数器累计计数未达到预定次数,且边界走向又发生变化,则可认为边界只发生了局部"抖动",不应认为边界方向有改变,本程序将忽略抖动,继续向下进行.

2.3 算法实现

输入:矩阵表示的二值图象.

输出:由域组成的链表.

域(内含关键点)的生成算法步骤为:

设第 i 行黑游程段个数为 n_i,图象的高度为 H.

第 1 步　设置 δ_1,δ_2 等偏差阈值,如图 1 所示;

第 2 步　扫描图象第一行(此时 $i=0$),调用黑游程生成函数 $F(0)$,检查第 1 个黑游程段,并形成第 1 个域 N_1;

第 3 步　循序检查第 k 个黑游程段形成第 k 个域 N_k(k 为整数,$0\leq k\leq n_k$);

第 4 步　置 $i=1$;

第 5 步　扫描图象第 i 行,调用黑游程生成函数 $F(i)$,生成黑游程;

第 6 步　逐一处理由黑游程生成函数生成的各个黑游程段.对于某个黑游程段 $B(X_i^1,X_i^2,Y_i)$,循序检查已形成的各个域 N,调用域的生成函数 $G(N,X_i^1,X_i^2,Y_i)$;该函数的返回值若为 true,则处理由黑游程生成模块生成的下一个黑游程;否则,检查已形成链表的下一个域 N,并再调用域的生成函数,若到达链表表尾,且域的生成函数的返回值均为 false,则生成新域.仿此处理下一个黑游程,直至处理完第 n_i 个黑游程.

第 7 步　判断是否满足 $i<H$,若是,则结束;否,则 $i=i+1$,转第 5 步;

3　曲线的拟合

如图 2 所示,上述关键点提取算法已提取出原图象曲线轮廓边界关键点,由于关键点都是成对出现(对于线宽为 1 的线条,成对出现的关键点重合为一点),因此对每一对关键点,则取其中点,这样由中点组成的点列就代表了曲线的轮廓边界.下面以这些点列为基础,用 n 次 B 样条函数拟合曲线.

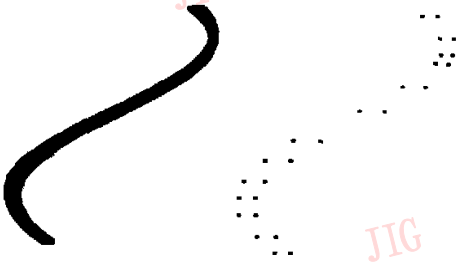


图 2　图象曲线轮廓边界关键点示意图

(左为原图,其线宽大于 1,右为线条轮廓的关键点)

假设这些中点点列是 $P_1,P_2,\cdots,P_m$,n 次 B 样条基函数为

$$F_{k,n}(t)=\frac{1}{n!}\times\sum_{j=0}^{n-k}(-1)^j\times C_{n-1}^{j}\times(t+n-k-j)^n\quad(n\leq t\leq 1)\quad(1)$$

其中,n 为 B 样条次数,$k=0,1,2,3,\cdots,n$

对于给定的点列 $P_1,P_2,\cdots,P_m$,$(n\geq m)$$n$ 次 B 样条函数第 i 段曲线的表达式为

$$B_{i,n}(t)=\sum_{k=0}^{n}P_{i+k}\times F_{k,n}(t)\quad(0\leq t\leq 1)\quad(2)$$

其中,$i=0,1,2,3,\cdots,m-n$

本文将使用三次 B 样条拟合曲线,即 $n=3$.下面首先介绍三次 B 样条曲线的端点位置[6].

对于第 i 段曲线,假设点列为 $P_i,P_{i+1},P_{i+2},P_{i+3}$,当 $t=0$ 时 $B_{i,3}(0)$为曲线的起点,当 $t=1$ 时 $B_{i,3}(1)$为曲线的终点,即

$$B_{i,3}(0)=\frac{1}{3}\times\left(P_i+\frac{P_{i-2}}{2}\right)+\frac{2}{3}\times P_{i+1}\quad(3)$$

$$B_{i,3}(1)=\frac{1}{3}\times\left(P_i+\frac{P_{i+3}}{2}\right)+\frac{2}{3}\times P_{i+2}\quad(4)$$

如图 3 所示,$P_i,P_{i+1},P_{i-2},P_{i+3}$为第 i 段曲线上 4 个点.第 i 段 B 样条起始点 S_1 与 B 样条终止点 S_2 分别位于线段 $P_{i+1}M_1$ 与线段 $P_{i+2}M_2$ 处,M_1、M_2 分别为线段 P_iP_{i+2}和线段 $P_{i+1}P_{i+3}$的中点.

在对曲线进行拟合时,要考虑以下几方面:

(1) B 样条曲线的起始点应为 P_0,终止点应为 P_m.

(2) 强制 B 样条曲线通过曲率较大的关键点.

(3) 强制 B 样条曲线,使其被限制在由关键点

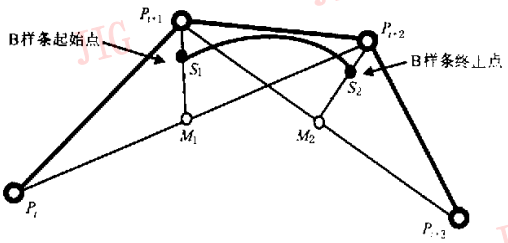


图 3 B 样条曲线拟合示意图

点列组成的各个条型区域内(如图 4 所示,第 i 段 B 样条曲线被限制在由关键点列所包含的区域内,并通过曲率较大的 P_{i-1} 点).

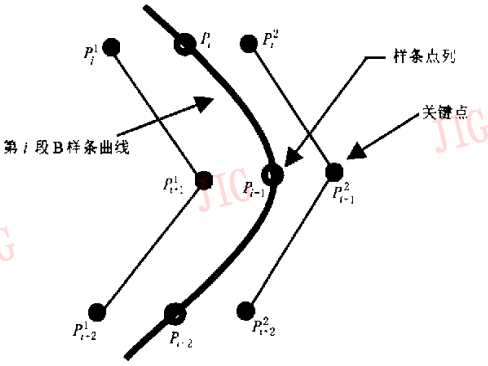
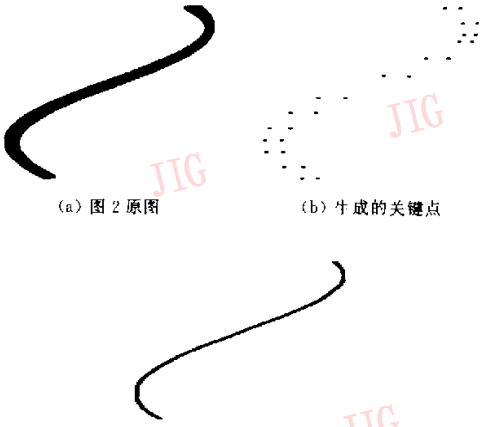


图 4 曲线拟合示意图

在生成 B 样条曲线过程中,通过动态修改、添加样条曲线的点列,即可达到上述 3 条要求(限于篇幅,具体算法从略).

4 结束语

本文提出了一种自适应提取扫描图象曲线轮廓关键点新算法,该算法通过自动逐行检测扫描行中的黑游程,当相继行中的黑游程满足本文提出的条件准则时,即能提取出相应的黑游程边界点,并形成关键点. 该算法对原图象不需作任何细化和其他预处理,直接对一个扫描行中各个黑游程图象线段进行处理,直至扫描处理完整个图象结束. 在预先给定阈值的基础上,该算法能自动调节已生成关键点的稀疏程度,且对线条宽度没有限制,最后对生成的关键点进行了拟合,即实现了对曲线的识别(见图 5(c)). 由该算法所生成的关键点还可用于图象的细化记录、图象的边缘追踪、图象处理与分析以及图象识别等领域.



(a) 图 2 原图 (b) 生成的关键点 (c) 为拟合的曲线

图 5 用本文算法提取的扫描图象曲线示意图

参 考 文 献

1 Pavlidis T. Algorithms for graphics and image processing. Rockville: Computer Science Press, 1982.
2 郭丙炎等. 工程图纸扫描输入后的智能识别方法. 中国机械工程, 1992,3(6),5~7.
3 李新友等. 面向图纸自动输入与交互设计的节则图象处理系统-IMCAD. 计算机学报,1994,17(11),866~871.
4 张树生等. 机械工程图的智能输入与识别研究. 西北工业大学学报,1995,13(2):217~221.
5 朱建新等. 基于区域件分解的工程图纸识别方法. 华中理工大学学报,1995,6(2):6~15.
6 刘子建,黄红武,黄素华. 计算机图形处理原理与 CAD 应用技术. 长沙:湖南科学技术出版社,1992.

王金鹏 1963 年生. 博士. 主要研究方向为计算机图形图象处理、模式识别、计算机视觉、机器人控制等.