一种手写体汉字快速细化分割算法*

A QUICK THINNING AND SEGMENTATION ALGORITHM FOR HAND-PRINTED CHINESE CHARACTERS

刘 平
Liu Ping
(重庆大学计算机系)

摘 要 提出了一种基于游程编码(run length encoding)的手写体汉字细化和分割快速算法。数据的输入与处理并行,汉字图象数据所占存贮空间仅为点阵方式的1/400。能将任何复杂汉字细化并分割为简单直线和弧线笔划。对汉字的结构特征抽取极为有利。此外,本文还提出一种新的边沿描述基段链码,其平均长度比 Freeman 链码短20%。

关键词 汉字识别;分离;平滑 / 边沿跟踪;链码;细化
中国图书资料分类法分类号 TP 391.4

ABSTRACT The paper proposes a quick thinning and segmentation algorithm for hand-printed Chinese characters going on simultaneously. The memory space occupied by the character image data is only one four hundredth of that by the normal character matrix. Any hand-printed Chinese characer can be thinned and segmented into several simple lines, arc and corner strokes. It is very efficient for structual feature extraction for Chinese characters. A new chain code for edge description is also proposed, whose average chain code length is about 20% less than that of Freeman chain code.

KEY WORDS Chinese character recognition; Segmentating; Smoothing/Contour tracing; Chain code; Thinning

0 引 言

预处理通常是文字识别不可缺少的重要阶段,其处理效果直接影响特征抽取,最终影响识别精度。汉字识别预处理的主要困难是:

(1) 汉字字符点阵较大(32×32以上),存贮大量汉字的点阵数据要受到计算机可用内存空间的限制。

(2) 汉字笔划多,笔划间的交叠情况复杂。一般的细化方法较费时且易出现不可细化区以及不必要的分岔[2~3]。

(3) 汉字结构复杂,笔划的分割既困难又费时。针对上述问题,本算法使用的原始数据

* 收文日期 90-5-3

为游程码(run length encoding)数据。它是由G3类传真机输入的Huffman编码数据解码而得，数据压缩倍数约为10[8]。而且采用数据输入与处理并行的方法，任何时刻只保存当前两相邻行上的游程码，不保存整字。若字符点阵行数为80，则所占存贮空间仅为点阵方式的1/(10×40)=1/400。

本算法利用G3类传真机作数字化输入设备，在IBM PC/AT机上实现。传真机与AT机用一块智能接口卡连接，该卡以Z80为CPU，实现数据缓冲和Huffman码到游程码的变换，其系统硬件框图如图1所示。

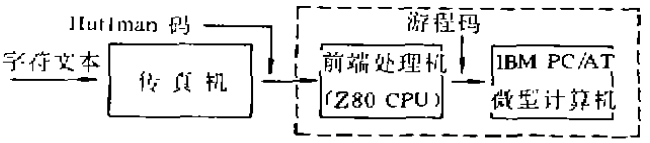


图 1

传统的细化算法大多基于中轴变换以获得字符骨架，用这些算法对汉字进行细化较费时，且细化后还要进行分割。此外，这类算法大多要对字符图象进行反复跟踪和检测，在不保留整字的游程码上很难实现这类算法。

在美术汉字中，有一种带单阴影边框的字体，如：

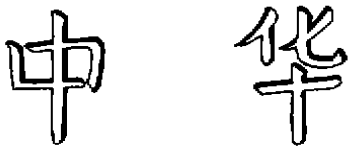


图 2(a)

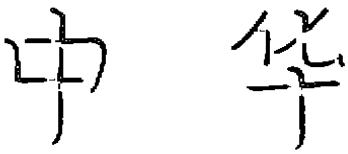


图 2(b)

其中的阴影部分仍然保留了汉字的结构和笔划信息：

更重要的是它能自然地将任何汉字细化并分割为简单直线、弧线和折线笔划。在这一特有效果的启发下，本文提出一种新的细化分割算法。只要对汉字图象进行一遍扫描，便可同时完成汉字的细化和分割，并把分割点处的笔划连接关系记录在相邻表中。本算法同样适用于字母和数字符号等线图(line drawings)。

1　数据形式与基段链码

1.1　数据形式

本算法处理的汉字图象数据为游程码数据，它不是按单个象点而是以扫描线上的连续象点(游程)为单位进行编码。每个游程用它的终点坐标 R_{ij}(或起点坐标 L_{ij})和游长度 W_{ij}(连续象点的个数)表示。也可以用游程的起点和终点坐标表示，即 $L_{ij}=R_{ij}-W_{ij}$。(见图3)

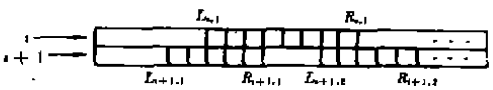


图 3

图3中的游程码为：

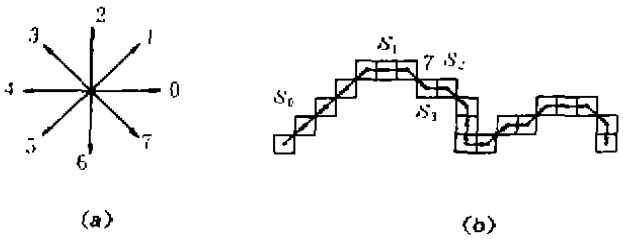
$(0,0)(L_{i,1},R_{i,1})\cdots(0,0)(L_{i+1,1},R_{i+1,1})(L_{i+1,2},R_{i+1,2})\cdots$

其中(0,0)为扫描线的起始标志。

1.2　基段链码

链码是描述图象边沿和骨架的有效手段。传统的Freeman链码对图象的描述是一点一

码，其方向定义如图4(a)所示。图4(b)中的链码为：1111007076601010076
缩写为：$1^4 0^2 7076^2 01010^2 76$



(a) (b)

图 4

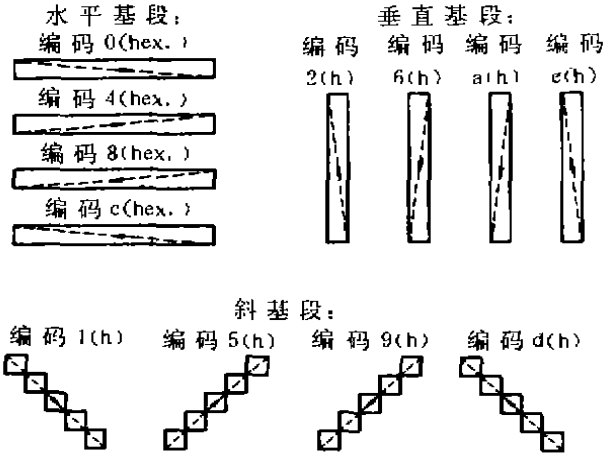


图 5

在实际字符图象中，图象的边沿均由长短不一的象点段(基段)组成。基段共有三类：水平基段(如图4(b)中的 s_1, s_2)；垂直基段(如图4(b)中的 s_3)；斜基段(如图4(b)中的 s_0)。Freeman 链码在任意两个同类型的非斜基段间必须要有一个段间的链码，如图4(b)中的 s_1 和 s_2 间必须要用链码"7"来连接。

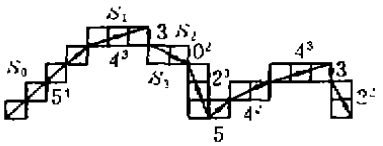


图 6

为了减少链码长度，本文提出一种基于基段的链码。它以基段为单位按对角线编码，有效地减少了段间的链码，其定义和编码如图5所示。

对图4(b)中的图象边沿用基段链码表示为：$5^1 4^3 3 0^2 2^2 5 4^2 4^3 3 2^2$ 见图6。

链长＝10，比图4(b)的链长少3。统计结果表明在描述同一图象边沿时，基段链码平均长度比 Freeman 链码短20%。

2 单边沿跟踪与链表结构

2.1 单边沿跟踪

传统的边沿跟踪算法对线图的跟踪总是来回经过笔道的两边，跟踪的结果是线图的内外边框(如图7所示)。

这种算法比较简单，但不便于汉字的分割和汉字结构特征的抽取。

本文提出的单边沿跟踪算法只跟踪汉字笔道的一边，并在跟踪过程中进行适当的分割，它保留了汉字的结构信息，有利于结构特征的抽取(见图8)。

图 7

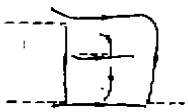


图 8

算法说明：

(1) 跟踪时优先选择上边沿和右边沿。开始跟踪某笔划时，若它的上下边沿还没有明显地显露出来，则两边同时跟踪，直到能区分出上下边沿时才丢弃下边沿的跟踪结果。此后不再改变跟踪边沿。

(2) 遇交叉、分岔或转角时便进行分割(详述于后)。

(3) 由于仅使用当前两扫描行上的游程码而未保留整字，故跟踪是多字多笔划同时进行的。

(4) 在跟踪过程中按游程间的交叠情况生成链码，跟踪的结果用基段链码表示。

(5) 按字符的书写位置进行字间的分离。

在图象的游程码表示中，相邻两行游程间的交叠情况是非常复杂的，它们可以是两段交叠，一段与多段交叠，多段与多段交叠(见图9)。

这给算法的设计带来相当大的困难，要一次考虑到两行间所有相交叠游程的各种组合情况是不太可能的。但若每次只考虑两段间的交叠，则所有交叠情况仅有4种(见图10)。

在上述每一种情况下再分别考虑它们与其它段的交叠情况。按左右不相交、左相交、右相交和左右相交4种情况，共4×4=16种交叠情况。用这16种交叠情况便可穷尽游程间所有交叠的组合。

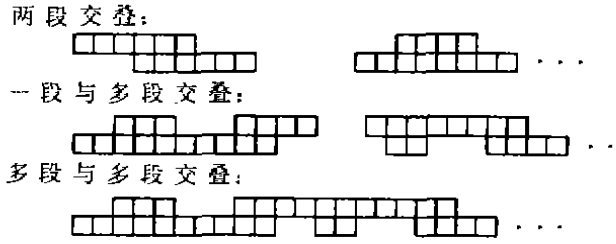


图 9

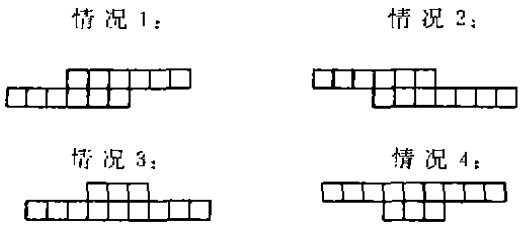


图 10

2.2 链表结构

在边沿跟踪过程中不断对边沿进行编码(挂链)，跟踪完毕便得到若干基段码链，其格式如下：

$$\overbrace{N_1\,N_2\cdots N_m}^{\text{相邻表}}-X_0-Y_0\overbrace{C_1C_2\cdots\ C_i}^{\text{链码域}}\ X_iY_ib$$

其中：· $Nj(1\leqslant j\leqslant m)$ 是与本链 i 相连的链的编号。

· $-X_0$-Y_0 是本链 i 起点坐标的负值，它兼作相邻表与链码域的分隔符。

· $C_k(1\leqslant k\leqslant i)$ 是第 k 个链码，且 $C_k=D_k\times 4096+W_k$. D_k 是 C_k 的基段码元(4 比特)，

W_c 是 C_c 的基段长度(12 比特)。

· X,Y 是本链 l 的终点坐标。

· b 是链尾分割标志,$b=1$ 表示链尾有分割,$b=0$ 表示链尾自然结束。

· $head(l)$,$end(l)$ 分别是链 l 的链首和链尾指针。

3 细化与分割

由于采用保留字符图象单边沿的方法,因而细化是单边沿跟踪的结果。它不是传统的中轴变换细化,而是用单边沿轮廓代替中心骨架,这对于象文字这类线图是完全可行的。以下主要讨论有关笔划分割的问题。

本算法的分割点选定为笔划的交叉、分岔和转角处。虽然汉字笔划间的相交情况很复杂,但就相交点附近的情况而言主要有10种[9]:

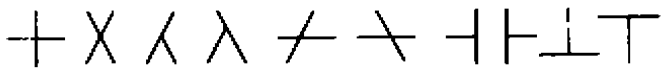


图 11

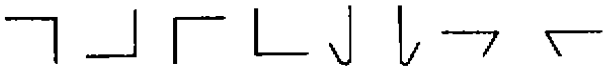


图 12

图 13

笔划转角的情况有8种(见图12)。

通过对上述笔划相交和转角情况的实际字符图象的分析,得到如下三条分割原则:

(1) 按水平方向分割;

(2) 在笔道宽度急剧变化处分割;

(3) 在笔划的自然分岔处分割。

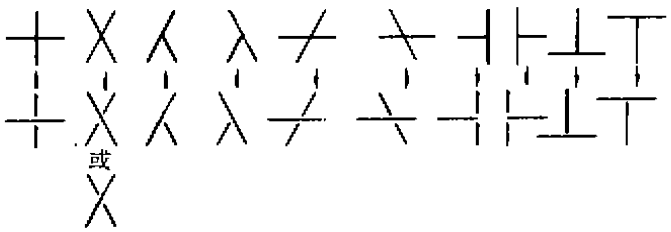


图 14

为消除算法对笔道宽度的敏感,原则(2)使用笔道宽度变化率确定是否需要分割。当两行上的相邻游程的长度之比$<\frac{1}{2}$或>2时,便在它们之间进行分割(如图13所示)。

按这些分割原则,图11,图12中的笔划将作如下分割(见图14、图15)。

其中因书写风格不同而使交叉(见图14)和竖勾(见图15)有不唯一的分割,但由于记录了分割点处的连接关系,分割的还原体仍是唯一的。

对大量汉字(1200个)字符的处理得到了令人满意的结果。

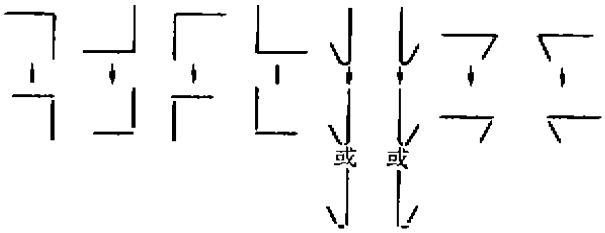


图 15

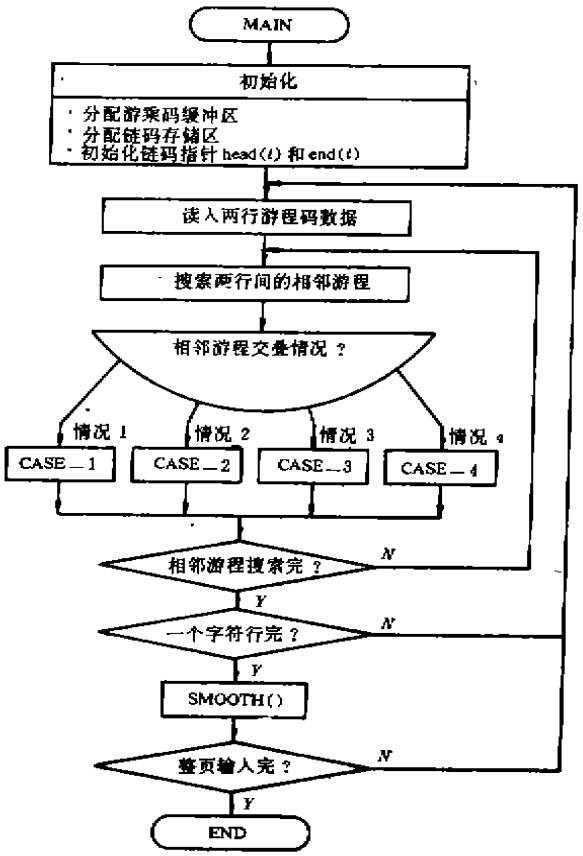


图 16

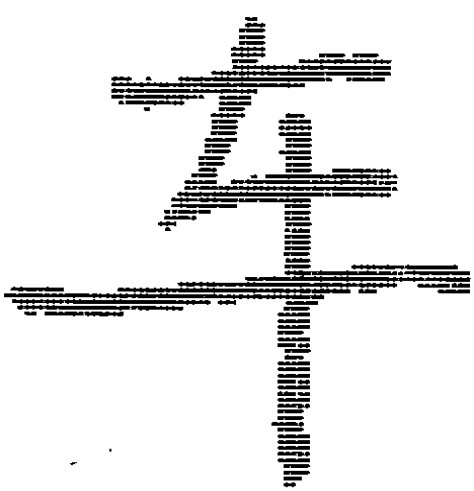
4　平滑处理

由于各种干扰，使输入带有噪声。本算法中的平滑模块 SMOOTH()用于处理常见的孤立噪声和边缘粗糙噪声。孤立点噪声用取笔划直径(最大长度)阈值的方法去掉，边缘粗糙噪声则用将较小凸出或凹陷平滑的方法去出。本算法与传统的平滑算法的区别在于它不是搜索整个字符图象平面，而是搜索字符图象边沿的链码表，因而处理速度较快。

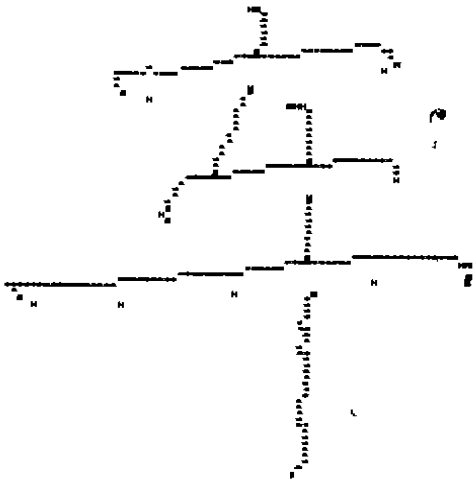
5 算法流程

本算法由主模块 MAIN();跟踪分割模块 CASE_1,CASE_2,CASE_3,CASE_4; 和平滑模块 SMOOTH()组成。它们的调用关系如图16所示,其中的4个跟踪分割模块分别处理前述4种交叠情况(见图10)。

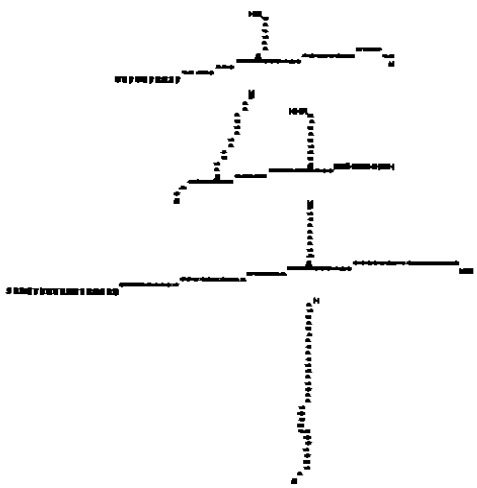
6 实验结果及分析



(a)原始图象



(b)细化分割后的图象



(c)平滑后的图象

chain 1's code:
6,[−193, −1],0002,2006,d001,a001,[194, 9],7fff;
chain 6's code:
1, 7,[−215, −10],d001,6001,c002,f001, 8004,8008,800a,8003,8005,3001,c00a,3001,[173, 13], 0;
chain 7's code:
6, 13,[−194, −14],a002,a002,a004,a002, a002,a003,[188, 29],7fff;
chain 9's code:
13,[−199, −17],0003,2009,[203, 27],7fff;
chain 13's code:
7, 9, 14,[−215, −27],f001,8009,800a,8005, 8007,8001,a002,[182, 33],0;
chain 14's code:
13, 20,[−203, −32], a00b,[202, 43],7fff;
chain 20's code:
14, 22,[−227, −44],c002,f001,8010,800a, 8006,800a,8009,8011,1001,[158,48], 0;
chain 22's code:
20,[−204,−48],a001,a011,a004,0002,a006, a001,a001,[200, 79], 0;

图 17

本算法用 Turbo C 编程，在 IBM PC/AT 机上的运行速度为30字/秒。图17中给出了一个典型的处理结果。其中，(a)为原始输入图象，(b)为细化分割后从基段码还原的图象，(c)为平滑处理后从基段码还原的图象。图中"H"表示链头基段；"E"表示链尾基段；"S"表示分割基段；"＊"表示一般黑象点。若某笔划被处理为一个基段，则该基段的象点全部标以"H"。图(c)由位于其下方的链码表还原而得。由于在跟踪与分割过程中不断有链的合并和删除，因而图中的链号不一定连续，每条链的结构如3.2节所述。

从图17中的处理结果可看出，细化分割后的汉字保留了原字的结构信息，复杂笔划被分割成简单笔划。笔划间的相对位置可用其链的坐标来确定，相互间的连接关系已记录在相邻表中。所有这些信息对汉字结构特征的抽取是极为有用的。

受传真机量化精度的影响，本算法要求书写的汉字不能太小(应大于7×7mm²)，否则分割后的笔划过短而容易在去噪声时丢失。若对算法的关键部分改用汇编语言编写将会使处理速度进一步提高。

参 考 文 献

1 Xie S L, Suk Mirsco. On machine recognition of hand-printed Chinese characters by frature rexaxation. Pattern Recogniton, 1988, 21(1): 1～7

2 Brown R M, Fay T H, Walker C L. Hankprinted Symbol Recognition System. Pattern Recognition, 1988, 21(2): 91～118

3 Pavlids Theodosius. Algorithms for Graphics and Image Processing. Rockville: Computer Science Press Inc, 1982

4 Pavlids T. Structural Pattern Recognition. U. S. A: Springer-Verlay Berlin Heidelberg, 1977

5 Kahan Simon, Pavlids Theo, Henry S Baird. On the Recognition of Printed Characters of Any Font and Size. IEEE Transactions on Pattern Analysis and Machine Intelligence, March 1987, PAMI-9(2): 274～287

6 Cheng Fang-Hsuan, Hsu Wen-hsing, Chen Mei-ying. Recognition of Handwritten Chinese Characters by Modified Hough Tranform Techniques. IEEE Transactions on Pattern Analysis and Machine Intelligence, April 1989, 11(4): 429～439

7 叶乃泰，张析中，夏莹. 汉字微型计算机与汉字识别. 北京：机械工业出版社，1989

8 孙仲康，沈振康. 数字图象处理及应用. 北京：国防工业出版社，1985，204～205

9 陈爱文，陈朱鹤. 汉字编码的理论与实践. 上海：学林出版社，1986，18～27