一种基于块邻接图的手写体文本格线删除及笔画重构算法

饶晓波[1] 邹北骥[2]

[1](湖南大学计算机与通信学院,长沙 410082) [2](中南大学信息科学与工程学院,长沙 410083)

摘 要 格线删除是计算机自动笔迹鉴定系统预处理过程中的关键步骤。在分析已有的手写汉字文本格线删除算法利弊的基础上,采用 Pavlidis 提出的块邻接图表示法来进行格线删除,并提出了格线删除后的笔画重构算法。该算法首先采用链表结构将笔迹图像保存为块邻接图;然后根据该邻接图搜索出满足条件的连通子图,这些连通子图就是需要删除的格线;最后再对因格线删除而断开的笔画进行重构。由于块邻接图只需进行一次水平或垂直扫描就能获得,因此格线的搜索和删除非常迅速;同时该算法还考虑了格线删除后的笔画重构,这样既删除了笔迹图像中的干扰信息,又不改变笔迹图像的特征。实验结果显示,该方法具有很强的抗干扰能力,对格线的删除非常有效。

关键词 块邻接图 格线删除 连通子图 笔画重构

中图法分类号: TP391.4 **文献标识码**: A **文章编号**: 1006-8961(2006)04-0549-06

An Algorithm for Erasing Grid-lines and Reconstructing Strokes in Chinese Handwriting Based on Block Adjacency Graph

RAO Xiao-bo[1], ZOU Bei-ji[2]

[1](*School of Computer and Communication, Hunan University, Changsha* 410082)

[2](*School of Information Science and Engineering, Central South University, Changsha* 410083)

Abstract Erasing grid-line is a key in the Chinese handwriting auto-identify system. After analyzing the advantages and disadvantages of the existed grid-line erasing algorithm we present a grid-line erasing algorithm based on the block adjacency graph(BAG) devised by Pavlidis and the stroke reconstruction algorithm in this paper. Firstly, the scanning graph for Chinese handwriting is changed into the BAG by use of the chain structure, and then the proper Connected Sub-graph according to the BAG is searched. These connected sub-graphs are the grid-lines that should be erased. Finally, the strokes, which left by erasing grid-line, are reconstructed. Because it is easy to get the BAG, searching and erasing grid-lines will be very fast. At the same time, our algorithm can reconstruct the strokes after erasing grid-line, and it not only avoids the disturbing information but also remains the useful characters of Chinese handwriting. Experimental results show our algorithm has strong ability in anti-disturbance and good effect in erasing grid-lines.

Keywords block adjacency graph(BAG), erasing grid-lines, connected sub-graph, stroke reconstruction

1 引 言

在计算机自动笔迹鉴定系统中[1~4],进行多份笔迹比对之前,必须进行预处理以去除笔迹图像中的非笔迹信息。格线删除是预处理的关键技术之一,其效果的好坏直接影响笔迹鉴定系统的性能。在数字图像处理技术中,通常利用 Hough 变换、直

基金项目:国家"973"计划项目(2004CB719404)

收稿日期:2005-02-23;**改回日期**:2005-06-10

第一作者简介:饶晓波(1970~),女。副教授。现为湖南大学计算机与通信学院计算机应用技术专业硕士研究生。研究方向为数字图像处理、模式识别。E-mail:rxbtt@163.com

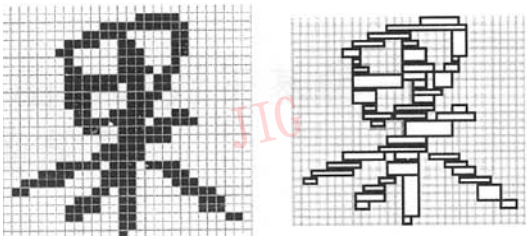
方图分析以及基于专家知识的方法来判断和删除格线。Hough 变换[5]对于 2 维形状的检测和识别非常有效,它能将 2 维平面任意方向的直线判断出来。其基本思想是将图像中每一像素点进行坐标变换,原图中某一直线上的所有点在变换空间的某个位置上形成峰点,这样将原图上的直线检测问题转化为找变换空间中峰点问题。Hough 变换检测直线需对原图中每个像素点计算其变换空间中对应的曲线。由此可见,该方法的运算量是非常大的,运算速度慢。直方图分析是对图像点阵区域进行坐标旋转后,再进行行(X 轴)或列(Y 轴)方向上的投影,得到横向或列向的黑像素的统计直方图,格线点阵区域在直方图上呈现出波峰。以此为基础,加上必要的修改措施来判断格线,该算法思路简单,对于印刷体文本效果较好,而对于手写体文本,在很大程度上依赖于手写文本的书写质量和规范。例如,书写时文字穿越格线、笔画与格线重叠等,文本文字和格线就难以区分,格线的删除也就具有很大的难度。基于专家知识的格线删除因专家知识的有限,各种情况考虑不完全,效果欠佳。Pavlidis 提出了一种基于块邻接图表示的图像表示方法[6],Yu 和 Jain 在此基础上提取表格中的字符和数字符号[7,8]。本文同样基于块邻接图表示来进行文本的格线删除,同时还提出了格线删除后的笔画重构算法。该方法只需进行一次水平或垂直扫描就能获得笔迹图像的块邻接图,因此格线判断迅速、准确;同时该方法还能够很容易地重构由于格线删除而断开的逸出笔画。实验证明,此算法对于自由书写的手写体文本的格线删除效果很好,并且能很好地保留笔迹文本的所有特征。

2 块邻接图表示

2.1 块邻接图的定义

块邻接图又简称为 BAG(block adjacency graph)图,它表示由图像中连续黑像素组成的矩形块之间的邻接关系图(图 1)。要获得块邻接图,只需对图像进行水平或垂直扫描,同时记录每行或每列所有连续黑像素的行程长度。在此基础上,将满足合并条件的相邻行程合并为 BAG 块,同时记录下各 BAG 块的邻接关系。

设相邻的 2 个行程用矩形坐标表示分别为 (X_{10},Y_{10},X_{11}),(X_{20},Y_{20},X_{21}),其中,X_{i0} 表示第 i 个行程的左横坐标,Y_{i0} 表示第 i 个行程的纵坐标,X_{i1} 表示第 i 个行程的右横坐标。



(a) 原图 (b) “果”字的水平块邻接图

图 1 “果”字的水平块邻接关系图

Fig. 1 The level block adjacency graph of “fruit” character

如果 2 个行程满足左右横坐标均相等的条件,即 $X_{10}=X_{20}$ && $X_{11}=X_{21}$,则上下两行程块合并为一个 BAG 块 $N_i(x_0^{(i)},y_0^{(i)},x_1^{(i)},y_1^{(i)})$,其中,$x_0^{(i)}=X_{10}$,$y_0^{(i)}=Y_{10}$,表示第 i 个 BAG 块的左上坐标;$x_1^{(i)}=X_{21}$,$y_1^{(i)}=Y_{20}$,表示第 i 个 BAG 块的右下坐标;$w^{(i)}=x_1^{(i)}-x_0^{(i)}$,表示第 i 个 BAG 块的宽度;$h^{(i)}=y_1^{(i)}-y_0^{(i)}$,表示第 i 个 BAG 块的高度。

根据 BAG 块的不同邻接关系,将 BAG 块分为 8 种类型:BAG 父块、BAG 子块、起始 BAG 块、终止 BAG 块、正规 BAG 块、合并 BAG 块、分支 BAG 块、起始分支 BAG 块,它们定义如下:

BAG 父块:与当前 BAG 块左边或上边相邻的 BAG 块。

BAG 子块:与当前 BAG 块下边或右边相邻的 BAG 块。

起始 BAG 块:没有 BAG 父块的 BAG 块。

终止 BAG 块:没有 BAG 子块的 BAG 块。

正规 BAG 块:只有一个 BAG 父块和一个 BAG 子块的 BAG 块。

合并 BAG 块:有两个以上 BAG 父块的 BAG 块。

分支 BAG 块:有两个以上 BAG 子块的 BAG 块。

起始分支 BAG 块:没有 BAG 父块,但有两个以上 BAG 子块的 BAG 块。

图 2 所示为某汉字的一部分,它由 7 个 BAG 块组成,它们的类型分别为:块 1、块 2 为起始 BAG 块,块 6、块 7 为终止 BAG 块,块 4 为正规 BAG 块,块 3 为合并 BAG 块,块 5 为分支 BAG 块。

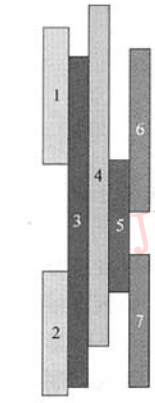


图 2　BAG 块的邻接关系

Fig. 2　Adjacence relationship of BAG block

2.2　块邻接图的数据结构

BAG 块的数据结构可描述为:

```
typedef struct _BNODE
{
  short nType; //BAG 块类型:1、起始 BAG 块 2、BAG
               父块 3、BAG 子块 4、终止 BAG 块
               5、正规 BAG 块 6、合并 BAG 块 7、分
               支 BAG 块 8、起始分支 BAG 块
  short nUpx; // 左上坐标 x
  short nUpy; // 左上坐标 y
  short nLwx; // 结束(右下)坐标 x
  short nLwy; // 结束(右下)坐标 y
  struct _BNODE * pPa rents[150];
                  //BAG 父块指针数组
  struct _BNODE * pCh ildren[150];
                  // BAG 子块指针数组
  short pParentN;
  short pChildrenN;
  BOOL IsInvalid; //该 BAG 块是否已经被删除,或者
                  被合并。
  long nNumber; //编号,方便文件保存时用
} BNODE;
```

例如:设图 2 中块 1、块 2、块 3 的指针分别为 P1,P2,P3,则第 3 块 BAG 块的表示为

```
BNODE3 = New _BNODE {
6,100,100,108,108, pParents[0] = P1; pParents[1] =
P2,pChildren[0] = P4,2,1,1,3
}
```

其表示的内容为该 BAG 为合并块,其起始坐标为(100,100),结束坐标为(108,108),该 BAG 块有两个 BAG 父块,分别为 P1 与 P2,有 1 个 BAG 子块,为 P4。该 BAG 块在当前图像中是有效块,其编号为 3。

2.3　形成 BAG 块的算法描述

(1)$i=0$,提取第 1 行的黑行程;每个黑行程形成 1 个 BAG 块;

(2)$i=i+1$;

(3)i 是否大于图像高度?是,转第 9 步;

(4)提取第 i 行的黑行程;

(5)检查第 1 个黑行程,依次与上行 BAG 块进行比较,若与某 BAG 块 T 符合合并条件,则合并此黑行程于 T 内,否则生成新的 BAG 块;

(6)检查第 2 个黑行程,依次与上行 BAG 块进行比较,若与某 BAG 块 T 符合合并条件,则合并此黑行程于 T 内,否则生成新的 BAG 块;

(7)如此循环,直至检查完该行的所有黑行程;

(8)$i=i+1$,转第 3 步;

(9)结束。

同样方法,竖直扫描可以生成竖直 BAG 块。

3　水平格线删除

3.1　原理

搜索水平格线就是在块邻接图表示的基础上,搜索同时满足以下条件的连通子图 G_k。搜索条件如下:

(1)BAG 块宽度 $w^{(i)}>T_{hw}$,高度 $h^{(i)}<T_{hh}$;

(2)BAG 父块与 BAG 子块横坐标满足:$|x_0^{(i)}-x_0^{(j)}|<T_{hd}$ && $|x_1^{(i)}-x_1^{(j)}|<T_{hd}$

(3)连通子图的宽度 $W^{(k)}>T_{hl}$;

(4)连通子图的高宽比 $H^{(k)}/W^{(k)}<T_{hhw}$

其中,T_{hw},T_{hh},T_{hl},T_{hhw} 和 T_{hd} 均为实验阈值。

满足条件的连通子图坐标为($X_0^{(k)}$,$Y_0^{(k)}$,$X_1^{(k)}$,$Y_1^{(k)}$),其中,$X_0^{(k)}=\min\limits_{N_i\in N_k}\{x_0^{(i)}\}$,$Y_0^{(k)}=\min\limits_{N_i\in N_k}\{y_0^{(i)}\}$,表示第 k 个连通子图的左上坐标;$X_1^{(k)}=\max\limits_{N_i\in N_k}\{x_1^{(i)}\}$,$Y_1^{(k)}=\max\limits_{N_i\in N_k}\{y_1^{(i)}\}$,表示第 k 个连通子图的右下坐标;$W^{(k)}=X_1^{(k)}-X_0^{(k)}$,表示第 k 个连通子图的宽度;$H^{(k)}=Y_1^{(k)}-Y_0^{(k)}$,表示第 k 个连通子图的高度。

3.2　搜索水平格线的算法描述

(1)$i=0$,$k=0$;判断水平块结点链表中第 i 个 BAG 块 N_i 是否满足 3.1 节条件 1?是,形成第 1 个连通子图 G_k,$k=k+1$;否则向下搜索,直到搜索到满足条件的 BAG 块为止;

(2)$i=i+1$;i 是否到达水平 BAG 块链表的表尾? 是,转第 9 步;

(3)i 是否满足 3.1 节条件(1)? 否,转第 2 步;

(4)i 是否起始 BAG 块? 是,转第 8 步;

(5)i 是否只有一个 BAG 父块? 否,转第 8 步;

(6)i 的唯一 BAG 父块是否只有一个 BAG 子块? 否,转第 8 步;

(7)i 与 BAG 父块的横坐标是否满足 3.1 节条件 2? 是,并入 BAG 父块所在连通子图,转第 2 步;

(8)形成新的连通子图 G_k,$k=k+1$,转第 2 步;

(9)计算每个连通子图的宽度 $W^{(k)}$ 和高度 $H^{(k)}$;

(10)是否满足 3.1 节条件 3 和 4? 是,保留该连通子图;否则,删除该连通子图。

由该算法搜索出的所有连通子图即为水平格线,只要令它们的像素值 $f(x,y)=0$,就可以删除这些水平格线。在格线删除时,保留格线的区域坐标及 BAG 表示的数据,这些数据作为汉字修补的基础数据。

3.3 页面倾斜角度的估计

在笔迹扫描的过程中,由于纸张的倾斜导致图像的倾斜,利用水平格线可以估算出页面倾斜角度。设以 $\boldsymbol{H}=\{G_k\}$ 记录分离出来的直线子图集合,那么页面倾斜角度可估算为

$$\alpha=\frac{\sum_{G_k\in H}W^{(k)}\arctan\left[\dfrac{(y_0^{(l)}+y_1^{(l)})-(y_0^{(r)}+y_1^{(r)})}{2(x_0^{(l)}-x_1^{(r)})}\right]}{\sum_{G_k\in H}W^{(k)}}$$

其中,$x_0^{(l)}=\min\limits_{N_i\in N_k}\{x_0^{(i)}\}$,$y_0^{(l)}=\min\limits_{N_i\in N_k}\{y_0^{(i)}\}$,$y_1^{(l)}=\min\limits_{N_i\in N_k}\{y_1^{(i)}\}$,$x_0^{(r)}=\max\limits_{N_i\in N_k}\{x_0^{(i)}\}$,$y_0^{(r)}=\max\limits_{N_i\in N_k}\{y_0^{(i)}\}$,$y_1^{(r)}=\max\limits_{N_i\in N_k}\{y_1^{(i)}\}$。$N_i$ 表示第 i 个 BAG 块,G_k 表示第 k 个水平格线连通子图,N_k 表示第 k 个被删除水平格线的所有 BAG 块。

4 笔画重构

笔画同格线的关系有相离、相接、相重、交叉 4 种情况。

笔画相离、相接时,直接分离格线对笔画没有任何影响;笔画相重时,虽然直接分离格线后会造成笔画宽度失真,但对特征提取影响不大,不需进行重构;笔画交叉时,由于格线的删除使原来的一个字符被断开成 2 个、甚至更多的几个部分,影响了汉字的完整性,因此必须将断开的笔画进行重构。利用 BAG 父块集 $\boldsymbol{F}_k$ 和 BAG 子块集 $\boldsymbol{Z}_k$ 能够有效地进行重构。

4.1 重构原理

笔画重构就是通过插入 BAG 块将断开的笔画连接起来,形成完整的字符。如图 3 所示,设 BAG 父块集中某 BAG 块的坐标为 $\boldsymbol{N}_f(x_0^f,y_0^f,x_1^f,y_1^f)$,BAG 子块集中某 BAG 块的坐标为 $\boldsymbol{N}_z(x_0^z,y_0^z,x_1^z,y_1^z)$,定义水平距离为

$$d_1=|x_1^f-x_0^z|$$

$$d_2=|x_1^z-x_0^f|$$

垂直距离为

$$h=|y_0^z-y_1^f|$$

插入 BAG 块的数目为

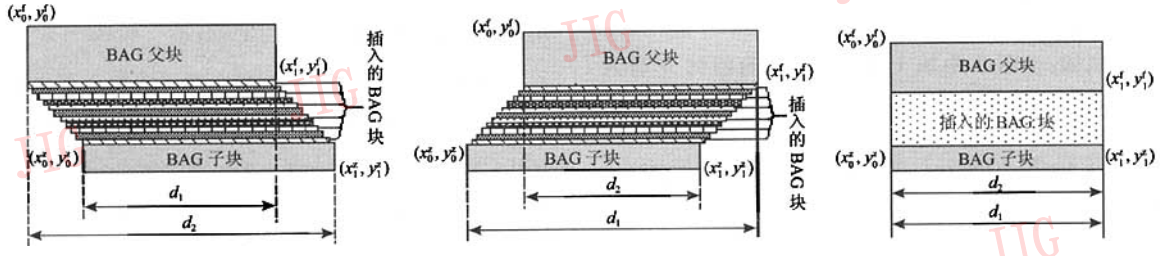
$$n=h$$

高度步长为

$$\Delta Y=1$$

起点步长为

$$\Delta X_0=\frac{x_0^z-x_0^f}{n+1}$$



(a) $d_1<d_2$ (b) $d_1>d_2$ (c) $d_1=d_2$

图 3 笔画重构情况

Fig. 3 Strokes reconstruction case

终点步长为

$$\Delta X_1 = \frac{x_1^z - x_1^f}{n+1}$$

4.2　重构算法描述

(1)计算 F_k(BAG 父块集)中所有 BAG 块与 Z_k(BAG 子块集)中所有 BAG 块之间的距离 d_1 和 d_2 以及高度差 h;

(2)搜索所有 $d_1 = d_2$ 的 BAG 块对(如图3(c)),如果 $h < T_h$,则在两 BAG 块之间新增一个 BAG 块 $(x_0^f, y_1^f, x_1^z, y_0^z)$;

(3)如果 $d_1 \neq d_2$(如图 3(a)、图 3(b)),则 $d = \min(d_1, d_2)$;

(4)搜索所有 $d < T_d$ && $h < T_h$ 的 BAG 块对,在两 BAG 块之间插入 n 个 BAG 块。第 $i(0 < i < n-1)$ 个 BAG 块坐标为

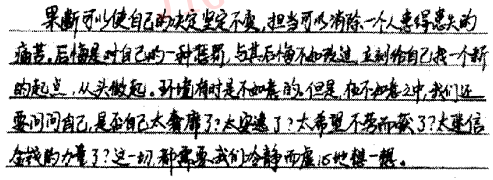
$$N_i(x_0^f + (i+1) \times \Delta X_0, y_1^f + i \times \Delta Y, x_1^f + (i+1) \times \Delta X_1, y_1^f + (i+1) \times \Delta Y)$$

其中,T_d,T_h 为实验阈值。

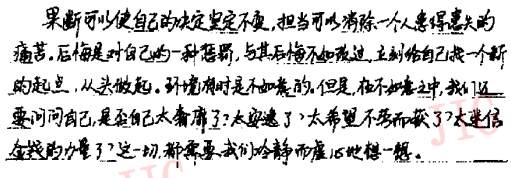
竖直长格线的删除和笔画的重构与水平格线类似。

5　实验结果

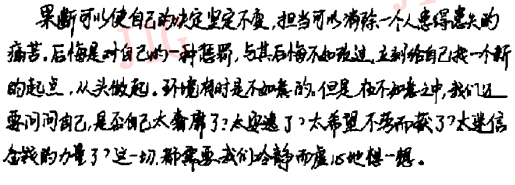
分别采用直方图法、Hough 变换法和基于块邻接图的格线删除及笔画重构方法,对不同格线分布、不同格线粗细程度、不同扫描倾斜(±5°范围)程度的笔迹图像,进行了格线的判断和删除实验,实验结果如图 4 所示。



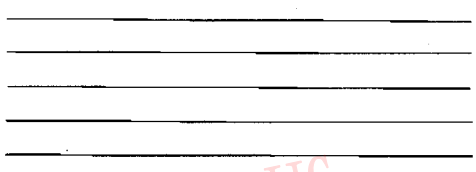
(a) 原始笔迹图像



(b) 采用直方图删除格线后的笔迹图像



(c) 采用 Hough 变换或 BAG 表示删除格线后的笔迹图像



(d) 采用 BAG 删除出来的格线集

图 4　实验结果

Fig. 4　Experimental results

由扫描仪获得的原始笔迹图像如图 4(a)所示;采用直方图的方法删除格线后的笔迹图像如图 4(b)所示;采用 Hough 变换法和采用基于块邻接图的方法删除格线并进行笔画重构后的笔迹图像效果相差不大,如图 4(c)所示;原始笔迹图像采用基于块邻接图的方法删除出来的格线集如图 4(d)所示。显然,直方图法去除格线无法很好的去除自由书写的手写体文本的格线,同时断开的笔画难以重构。更有甚者,如果文字的书写距离太近,可能会导致直方图无法正确检测格线。

在 PⅢ-450 上测试 Hough 变换和基于块邻接图表示的格线删除的速度显示,对于同样一篇自由书写的笔迹文档,Hough 变换去除格线所需的时间为 10 ~ 20ms,而采用基于块邻接图表示的方法去除格线所需的时间只有 2 ~ 5ms。显然,后者的运算速度较前者快。

由实验可知,基于块邻接图的格线删除及笔画重构算法不仅可以高速、有效地删除笔迹图像的格线,而且还可以重构由于格线删除而断开的笔画,使得穿越格线的笔迹能完整保留。因此,采用该算法删除格线非常有效。

6　结　论

在计算机自动笔迹鉴定系统中,格线的删除是预处理的关键步骤之一。利用数字图像处理技术中的

一些算法删除格线效果不甚理想,要么格线难以完全删除,要么运算量太大、速度较慢。采用基于块邻接图的格线删除及笔画重构算法,不仅运算简单,程序处理速度快,格线消除干净,而且笔迹的整体特征不会因此而改变。通过实验验证,该方法不仅能快速、准确、不失真地判断和删除格线,而且能够很好地重构由于格线删除而断开的逸出笔画。因此该方法可以广泛应用于自动笔迹鉴定系统及相关领域。

参考文献(References)

1 Said H E S, Tan T N, Barker K D. Personal identification base on handwriting[J]. Pattern Recognition, 2000, **33**(1): 149 ~ 160.

2 Greening C M, Sagar V K, Leedham C G. Handwriting identification using global and local features for forensic purposes [A]. In: Proceedings of the European Convention on Security and Detection [C], Brighton, UK, 1995: 272 ~ 278.

3 Arazi Benjamin. Automatic handwriting identification based on the external properties of the samples [J]. IEEE Transactions on Systems, Man and Cybernetics, 1983, **13**(4):635 ~ 642.

4 Steinke Karlo. Recognition of writers by handwriting images [J]. Pattern Recognition, 1981, **14**(1-6):357 ~ 364.

5 Lu Yue, Shi Peng-fei, Zhang Ke-hua. Detection and removal of base line from document images [J]. Journal of Computer Research & Development, 2001, **38**(5):552 ~ 556. [吕岳, 施鹏飞, 张克华. 文档图像中书写线的检测与去除[J]. 计算机研究与发展, 2001, **38**(5):552 ~ 556.]

6 Pavlidis T. Algorithm for Graphics and Image Processing [M]. Rockville, MD, USA: Computer Science Press, 1982: 199 ~ 201.

7 Yu Bin, Jain Anil K. A generic system for form dropout[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1996, **18**(11):1127 ~ 1134.

8 Jain Anil K, Yu Bin. Document representation and its application to page decomposition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, **20**(3):294 ~ 308.