面向字符识别的快速小形变细化算法

龚才春　刘荣兴
(山东大学计算机学院　山东 济南 250061)

摘　要　本文描述了现有字符细化算法的思想及其缺陷,并且在分析细化形变根源的基础上给出了一种面向字符识别的快速细化算法。该算法不仅速度非常快,而且不会产生毛刺和伪分支点,细化后字符骨架形变很小。

关键词　字符识别　细化　骨架　形变

A CHARACTER-RECOGNITION-ORIENTED FAST THINNING ALGORITHM WITH LITTLE DISTORTION

Gong Caichun　Liu Rongxing
(*School of Computer, Shandong University, Jinan Shandong 250061, China*)

Abstract　The paper depicts available thinning algorithms' main ideas and defects, and presents a Fast Character-Recognition-Oriented Thinning Algorithm with little distortion based on analysing origin of thinning distortion. The algorithm is good not only because it's fast, but also because it produces no hairs and no spurious branches which makes the skeleton of the thinned character have very little distortion.

Keywords　Character recognition　Thinning　Skeleton　Distortion

1　引　言

几乎所有的光学字符识别(OCR)都是基于细化算法的,因此,细化算法的好坏很大程度上决定了 OCR 系统的好坏。一个好的细化算法可以减少细化造成的形变,找到能反映字符真实形状的特征点,使系统有较高的识别率;相反,一个不好的细化算法会产生伪特征点,给字符分类带来困难,甚至导致误识或拒识。本文描述了几种 OCR 领域常用的细化算法,分析了其缺陷,并且给出了一种基于字符识别的简单快速、形变很小的细化算法。

2　现有细化算法的缺陷

对于给定的字符图形使笔道宽度变细,从而提取线宽为 1 的字符图像的骨架的操作叫细化。细化主要用来分析字符的结构特征,在光学字符识别中得到广泛应用。为了使细化得到的字符骨架能够尽量反映原始字符的形状,一般细化算法都是提取字符笔道的中心线,只有在数字字符和结构简单的西文字符识别时才可能提取左边界或右边界。

现在使用最多的细化算法是边缘侵蚀细化算法。字符图像的边缘就是与背景点相邻的前景点集合,因此去噪处理后的二值化字符图像(前景为 1,背景为 0)的边缘就是与 0 像素点相邻的 1 像素点。边缘侵蚀细化算法就是循环检测笔道的边缘,并且将边缘像素置为背景,直到所有的笔道都为单像素宽度。很多现有的其它细化算法其实质也是边缘侵蚀。如轮廓跟踪细化算法其实质也是边缘侵蚀,只是它寻找边缘的方法是采用轮廓跟踪的方式,找到一个边缘点后,搜索与该边缘点 8 邻域相邻的边缘点,直到重新回到开始的搜索点。

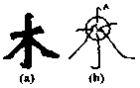

图 1　一般细化算法容易产生的形变

这些方法最致命的缺陷是会产生毛刺和伪分支,如图 1 所示。图 1(b)是图 1(a)中'木'字按这些方法得到的细化结果。图 1(b)A 处为细化产生的毛刺,圈内为细化产生的伪分支。毛刺和伪分支严重影响了字符分类,例如图 1(a)的'木'字就很容易误识为'水'。另外算法需要对字符图像做多次边缘检测和去边缘操作,运算量巨大,速度较慢。

3　快速小形变细化算法

3.1　相关概念

为了算法叙述的方便,对二值化字符图像,我们定义以下概念:

(1) 点段　一行中值为 1(即为前景)的连续像素序列称为点段。用 seg(i,j)表示第 i 行的第 j 个点段,lseg(i,j),rseg(i,j),mseg(i,j),分别表示 seg(i,j)的左端点、右端点、中点。

(2) 相关点段　如果相邻二行的两个点段 seg(i,j)和 seg(i+1,k)满足下列条件之一,则称点段 seg(i,j)和 seg(i+1,k)为相关点段。

$$lseg(i,j) \leqslant lseg(i+1,k) \leqslant rseg(i,j)$$
$$lseg(i,j) \leqslant rseg(i+1,k) \leqslant rseg(i,j)$$
$$lseg(i+1,k) \leqslant lseg(i,j) \leqslant rseg(i+1,k)$$
$$lseg(i+1,k) \leqslant rseg(i,j) \leqslant rseg(i+1,k)$$

(3) 起始段　如果点段 seg(i,j)不存在 i-1 行的相关点段,称点段 seg(i,j)为起始段。

(4) 终止段　如果点段 seg(i,j)不存在 i+1 行的相关点段,

收稿日期:2003-11-08,龚才春,硕士生,主研领域:模式识别与人工智能。

称点段 seg(i,j)为终止段。

(5) 一对多相关　如果一个点段与多个点段相关,称为一对多相关。

(6) 相关段　点段集 seg(i + 1,k1),seg(i + 2,k2),…,seg(i + m,km),如果 seg(i + n,kn)与 seg(i + n + 1,k(n + 1))一对一相关(0≤n≤m)且每个点段长度都小于一定阈值,则称这些点段集为相关段,m 则称为相关深度。

(7) 要点段　起始段,终止段,一对多段和长度大于一定阈值的点段,称为要点段。

(8) 同组要点段　深度小于一定阈值的相关要点段称为同组要点段。其左右端点位置分别取组中各段最小左端位置和最大右端位置。

(9) 端脚段　与某个同组要点段相关的组外点段就称为该同组要点段的端脚段。深度大于一定阈值的端脚段称为有效端脚段,否则为无效端脚段。

如图 2(a)所示字符'来'的段化图,第一笔横由三行标记为 1 的点段组成,这三行点段构成同组要点段,标记为 5 的各点段构成相关段,其中第一行为起始段,最后一行为终止段,标记为 4 的同组要点段有 6 个有效端脚段,标记为 8 的点段为无效端脚段。

3.2　细化过程

我们知道,对于字符来讲,细化的毛刺主要来源于字符笔划末端的修饰,因此,要消除毛刺就必须在取笔道中心线之前去掉这些修饰,也就是将这些修饰置为背景。而细化的伪分支都是由于笔道交叉处像素较其它位置宽而产生的,因此要消除伪分支就不能对笔道交叉处做简单的边缘侵蚀,而要用其它方法。

对于二值化字符图像,我们逐行扫描得到所有点段集,并在点段集基础上按照点段间的相互关系,记录图像的同组要点段及其有效端脚段,将所有无效端脚段包含的像素置为背景,这样就清除了字符所有横向笔道的修饰,然后逐列做相同的操作,就清除了字符所有纵向笔道的修饰。去掉字符的横向和纵向修饰,保证了细化骨架不会产生毛刺。图 2(b)显示了去掉横纵修饰后字符'来'的点段情况。

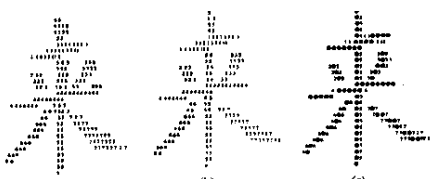

图 2　字符'来'的细化过程

去掉字符的修饰后,字符的细化就变得非常简单了。对所有按行扫描得到的相关段包含的所有点段取中点作为最后骨架需要保留的像素,如果取中点改变了前景的连通性,即使前景中连通的笔道在骨架中不连通了,则做局部调整,将中点附近的点也作为骨架需要保留的像素,保证原始字符图像中连通的笔道,在细化后的骨架中也是连通的。对每一个同组要点段,包括按行扫描得到的同组要点段和按列扫描得到的同组要点段,都直接在其两端点之间连一条直线,直线经过的像素作为骨架需要保留的像素即可。图 2(c)显示了字符细化后的最后结果,用"·"标记的像素集合就构成了字符的细化骨架。

3.3　算法分析

(1) 算法复杂度分析

算法首先获得字符图像的各点段,以及在此基础上的要点段,相关段,同组要点段及其端脚段等,这是对字符图像的一次扫描,所以复杂度为 O(H·W),H 和 W 分别为字符图像的高和宽。在点段基础上得到骨架的过程就是对点段的一次扫描,所以其复杂度为 O(L),L 为得到的点段数目。因此,算法总的复杂度是 O(H·W)。由于不需要循环对图像进行扫描,所以算法速度非常快。

(2) 算法的细化效果分析

在细化过程中,我们首先清除了同组要点段的无效端脚段,这样就清除字符笔划的修饰,因此细化就不会再出现毛刺了。另外,对同组要点段没有采用边缘侵蚀的方法,而是直接将同组要点段的端点用直线连接,直线和相关段细化后的中心线都是单像素宽度,单像素宽度的直线与直线、直线与曲线相交都不可能出现多余分支点,从而本细化算法也就避免了伪分支的出现。由此可知,用此方法得到的细化骨架能够最大程度上反映字符的形状特征,是一种形变非常小的细化算法。

4　结　论

本文提出的面向字符识别的细化算法,是在认真考虑现有细化算法产生毛刺和伪分支点等形变根源的基础上,通过对字符图像行列点段的分析和处理来清除毛刺和伪分支点产生的根源而实现字符细化的。本方法从根源上杜绝了细化产生的毛刺和伪分支点的可能,而且速度很快,特别适合于字符识别领域。

参　考　文　献

[1] 周长乐,手写汉语的机器识别[M],北京:科学出版社 1997.

[2] Amin, Adnan, Singh, Sameer Machine Recognition of Hand-Printed Chinese Characters Intelligent Data Analysis Volume:1997,1(1 - 4):101 ~ 118.

(上接第 47 页)

整个油料管理的态势在地图一目了然。地理数据和它们的属性数据分别由数据文件和属性数据库进行管理。在属性数据库中,每一条记录都包括地理要素 ID 号的字段,通过这个字段,使属性数据库与地图关联。

图 1 是系统数字地图的主界面。它包括有东北三省的交通网络、主要河流和主要城市。

3　结　论

通过上述方法所获得的数字地图,能够适应对图形平台要求不高的小型 GIS 系统的要求,由于采用的是 VC 编程,运行速度更快。对于小型系统而言,自行开发可以节约较大的初期投资费用。利用 MapInfo 转出表提供的数据,无需在 VC 中实现制图功能,充分利用了现有的资源,使开发周期大大的缩短,并减少了开发难度。

由于采用的是文件管理地理数据的方式,且系统又不具备制图和编辑地图功能,会给地图的更改带来困难,这还有待进一步的研究。

参　考　文　献

[1] 陈建春,Visual C ++ 开发 GIS 系统—开发实例剖析,北京:电子工业出版社,2000:2 ~ 7.

[2] 邬伦、刘瑜等,地理信息系统—原理、方法和应用,北京:科学出版社,2001:28 ~ 31.

[3] 罗云启、罗毅,数字化地理信息系统 MapInfo 应用大全,北京:希望电子出版社,2001:36 ~ 46.