

文章编号:1006-544X(2005)01-0101-03

基于特征的印刷体数字符号识别系统

刘建华¹, 牛秦洲², 程小辉², 王 勇³

(1. 桂林航天工业高等专科学校 计算机系, 广西 桂林 541004; 2. 桂林工学院 电子与计算机系, 广西 桂林 541004;
3. 上海电力学院 计算机系, 上海 200062)

摘 要: 探讨了一种印刷体数字符号快速识别算法, 采用图像旋转、去噪声、图像分割等手段对含有印刷体数字符号的图像进行预处理, 然后根据印刷体数字的字符结构提取 4 种稳定且有效的结构特征, 即横线特征、竖线特征、水平方向过线数和垂直方向过线数. 根据这些特征构造编码器, 进行编码识别, 使用 VC++ 编程实现. 实验表明, 该算法速度快、精度高, 有一定的抗干扰性.

关键词: 印刷体数字; 识别; 特征提取; 过线数; 编码器

中图分类号: TP391.43

文献标识码: A^①

光学字符识别 (OCR) 方法大致可分为两类^[1]: 基于结构特征的方法和基于统计特征的方法. 使用统计特征的分类易于训练, 且在给定的训练集上能得到较高的识别率; 使用结构特征的方法能描述字符的结构, 在识别过程能有效结合几何和结构的知识, 得到可靠性较高的识别结果.

印刷体数字识别是字符识别的一个分支,有多种方法,如模板匹配法、特征值提取法等.模板匹配法简单,但计算量很大,且费时;特征值提取法是基于特征的识别,关键是选取稳定且有效的特征,其计算量相对较小,识别速度快.本文选用特征值提取法进行数字识别.流程见图1.

1 识别前的预处理

对识别系统而言,图像预处理的质量直接影响到后续的识别效果,因此这一环节非常重要.常用的预处理手段包括^[2-4]:图像旋转、去噪声、细化、二值化、图像分割等.

1.1 图像旋转

一幅图像在扫描时有可能出现一定倾斜,

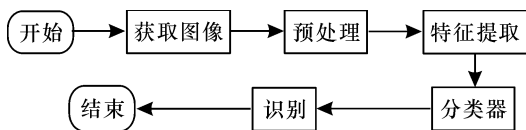


图1 特征值提取法数字识别流程
Fig.1 Main process of printed numerals' recognition in feature extraction

所以在打开一幅位图进行识别前要把图像旋转一定角度，一般是以图像中心为原点旋转一定角度。

1.2 二值化

图像二值化对景物分析、模式识别等具有重要意义. 二值化的方法很多, 有直方图统计法、松弛法、固定阈值法、抖动矩阵二值化法、动态门限法等.

最常用的阈值分割方式是先选取合适的阈值，所有的灰度值大于或等于某个阈值的像素都被判属于物体，其他的像素被判属于背景，或者相反。

256 级灰度國值变换函数表达式如下:

$$f(x) = \begin{cases} 0, & x < T, \\ 255, & x \geq T. \end{cases}$$

其中 T 为指定的阈值. 通常采用直方图技术来确

① 收稿日期: 2004-10-19

基金项目：广西区科技攻关项目（桂科攻 0428002-1）

作者简介: 刘建华 (1972 -), 男, 硕士, 讲师, 研究方向: 计算机图形图像处理、网络信息安全. E-mail: ljh91091cn@yahoo.com.cn

定阈值. 一幅物体与背景对比明显的图像具有包含双峰的灰度直方图, 选择双峰之间的谷作为灰度阈值可得到合理的分割结果.

1.3 去噪声

噪声恶化了图像质量, 使图像模糊, 甚至淹没特征, 给分析带来困难. 消除图像噪声的工作称之为图像平滑或滤波. 由于噪声源众多, 噪声种类复杂, 所以平滑方法也多种多样. 本文采用中值滤波的方法去除噪声.

中值滤波就是用一个含有奇数点的滑动窗口, 将窗口正中的那一点的像素值用窗口内各点的像素中值代替. 一维中值滤波很容易推广到二维, 二维中值滤波器比一维中值滤波器更能抑制噪声.

1.4 图像分割

图像分割的目的是把图像空间分割成一些有意义的区域. 为了识别单个数字, 要将每个数字的图像从整块图像中分割开来. 单个数字的图像子块像素保持连通, 不同数字的图像子块一般不粘连. 本文选用投影法进行图像分割. 定义各列和的向量 \mathbf{P}_x 为图像在 x 轴上的投影, 即垂直投影; 定义各行和的向量 \mathbf{P}_y 为图像在 y 轴上的投影, 即水平投影. 处理的图像经过阈值化预处理后, 只包含 2 个颜色值: 0 和 255. 0 为黑色, 表示该点属于数字; 255 为白色, 则此点是背景中的一点. 对图像的像素值 $f(x, y)$ 有:

$$I(x, y) = \begin{cases} 1, & \text{当 } f(x, y) = 0, \\ 0, & \text{当 } f(x, y) = 255. \end{cases}$$

由此, 得到求垂直投影和水平投影的公式

$$\mathbf{P}_x = \sum_y I(x, y), \quad \mathbf{P}_y = \sum_x I(x, y).$$

采用投影法进行图像分割的优点是简单快捷. 由垂直投影可以得到列分割点, 由水平投影可以得到行分割点.

2 特征提取

特征值提取法的关键是选取稳定且有效的结构特征, 其计算量相对较小, 识别速度快. 提取不同的特征, 识别率也不同. 本文提取数字的 4 种特征: 横线特征、竖线特征、水平方向过线数和垂直方向过线数.

2.1 横线特征

根据印刷体数字的结构特征, 数字中有可能存在横线. 于是, 在水平方向上, 定义比例: $R_{\text{Hori}} = H_{\text{BlackNum}} / L_{\text{Width}}$. 其中, L_{Width} 为某数字的图像子块的

宽度, 用像素数来度量; H_{BlackNum} 为该数字的图像子块在水平方向上黑像素点连续出现的个数.

若 $0.6 \leq R_{\text{Hori}} \leq 1$, 则认为数字中这些连续出现的黑像素点构成了一条横线. 由于数字中的横线是具有一定宽度的, 因此在水平扫描线顺序扫描时, 相邻的几条横线视为同一条横线.

根据横线在数字中的不同位置, 笔者定义: 从左上角开始的横线为上横线, 从左下角开始的横线为下横线. 以“1, 7”为例, 根据上述描述, 易知: “1”有一条下横线, “7”有一条上横线.

2.2 竖线特征

与横线特征原理相同. 在垂直方向上, 定义比例: $R_{\text{Vert}} = V_{\text{BlackNum}} / L_{\text{Height}}$. 其中, L_{Height} 为某数字的图像子块的高度, 用像素数来度量; V_{BlackNum} 为该数字的图像子块在垂直方向上黑像素点连续出现的个数.

若 $0.4 \leq R_{\text{Vert}} \leq 1$, 则认为数字中这些连续出现的黑像素点构成了一条竖线. 同理, 由于数字中的竖线是具有一定宽度的, 因此在垂直扫描线顺序扫描时, 相邻的几条竖线视为同一条竖线.

根据竖线在数字中的不同位置, 笔者定义: 从左下角开始的竖线为左竖线, 从右下角开始的竖线为右竖线. 以“0”为例, 根据上述描述, 易知: “0”有一条左竖线, 还有一条右竖线.

2.3 水平和垂直过线数

把数字平均分成上、中、下 3 部分, 在每个部分中分别以水平方向的扫描线从左到右穿过数字, 计算每条扫描线穿越黑像素区域互不相邻的交点数, 在上部得到的最大交点数定义为该数字的上过线数, 在下部得到的最大交点数定义为该数字的下过线数. 以“2”为例, 显而易见: 数字“2”的上过线数为 2, 下过线数为 2, 在 $1/2$ 高度处的过线数 (定义为 X 轴过线数) 为 1, 在上 $1/4$ 高度处的过线数为 1, 在下 $1/4$ 高度处的过线数为 1.

与水平方向过线数原理相同, 把数字平均分成左、中、右 3 部分, 在每个部分中分别以垂直方向的扫描线从上到下穿过数字, 计算每条扫描线穿越黑像素区域互不相邻的交点数, 在左部分得到的最大交点数定义为该数字的左过线数, 在右部分得到的最大交点数定义为该数字的右过线数. 以“9”为例, 显而易见: 数字 9 的左过线数为 3, 右过线数 (定义为 Y 轴过线数) 为 3, 在

1/2 宽度处的过线数为 3，在左 1/4 宽度处的过线数为 3，在右 1/4 宽度处的过线数为 3。

3 识别

根据表 1 中的特征而构造编码器进行数字识别,分析其特征. 以上横线、下横线、左竖线、右竖线、上 1/4 高度处的过线数、下 1/4 过线数、X 轴过线数、Y 轴过线数构造编码器(表 2). 每个数字的编码不完全相同,所以可由编码器识别出数字 0~9.

对一幅含有 242 个数字的图像进行识别，识别结果显示正确率达 96.5%，该算法简洁、快捷、实用，有一定的抗干扰性. 由于结构特征的恰当处理，使得识别效果基本上与字体字号无关.

表 1 数字特征
Table 1 Numerals' features

数字特征		0	1	2	3	4	5	6	7	8	9
横 线	上横线	0	0	0	0	0	1	0	1	0	0
	下横线	0	0,1	1	0	0,1	0	0	0	0	0
	左竖线	1	0	0	0	0	0	1	0	1	0
竖 线	右竖线	1	0	0	0,1	0	0	0	0	1	1
	上 1/4 过 线数	2	1	1	1	2	1	1	1	2	2
	下 1/4 过 线数	2	1	1	1	1	1	2	1	2	1
水平 方向 过线	X 轴上 过线数	2	1	1	1	2	1	1,2	1	1	2
	上 1/4 过 线数	2	0,1,2	3	2	2,3	3	3	1	3,4	3
	下 1/4 过 线数	2	0,1	3	3,4	1	3	3	2	3,4	3
垂直 方向 过线 特征	Y 轴上 过线数	2	1	3	3	2	3	3	1,2	3	3

注:横线、竖线特征栏中的数字代表是否存在该类横线、竖线;若为 1,则存在;若为 0,则不存在. 过线数特征栏中的数字代表过线个数(某些数字由于字体大小或噪声会出现几种结果)

表 2 编码器
Table 2 Encoder

数字特征	0	1	2	3	4	5	6	7	8	9
上横线	0	0	0	0	0	1	0	1	0	0
下横线	0	0,1	1	0	0,1	0	0	0	0	0
左竖线	1	0	0	0	0	0	1	0	1	0
右竖线	1	0	0	0,1	0	0	0	0	1	1
上 1/4 过线数	2	1	1	1	2	1	1	1	2	2
下 1/4 过线数	2	1	1	1	1	1	2	1	2	1
X 轴过线数	2	1	1	1	2	1	1,2	1	1	2
Y 轴过线数	2	1	3	3	2	3	3	1,2	3	3

4 结束语

印刷体数字有一些明显的特征，从 10 个数字中寻找稳定而有效的特征来构造编码器，从而识别印刷体数字. 算法使用 VC++ 编程实现. 实验表明该算法速度快，有一定的抗干扰性.

进一步提高识别效果的一些措施有：扩大水平方向和垂直方向的扫描线条数，例如水平方向扫描线增加为上 1/8 过线、上 1/4 过线，上 3/8 过线、下 3/8 过线、下 1/4 过线、下 1/8 过线等；还可以给不同的过线赋予不同的优先权；构造编码器识别之前先进行“细化”，这时候寻求一个高效的细化算法就成为问题的关键.

参考文献

[1] 张宏林. Visual C++ 数字图像模式识别技术及工程实践 [M]. 北京: 人民邮电出版社. 2003 .
[2] 贾永红. 计算机图像处理与分析 [M]. 武汉: 武汉大学出版社, 2001.
[3] 牛秦洲, 李正吾. 图像边缘提取的分维算法 [J]. 桂林工学院学报, 1996, (4): 443 - 446.
[4] 韦春桃, 程晓宇. LOG 算子进行边缘检测的研究 [J]. 桂林工学院学报, 1999, (2): 181 - 183.

Feature – Based Printed Numerals’ Recognition System

LIU Jian-hua¹, NIU Qin-zhou², CHENG Xiao-hui², WANG Yong³

(1. Department of Computer Science and Technology, Guilin College of Aerospace Technology, Guilin 541004, China;
2. Department of Electronics and Computer Science, Guilin University of Technology, Guilin 541004, China;
3. Department of Computer Science, Shanghai University of Electric Power, Shanghai 200090, China)

Abstract: A fast recognition algorithm for printed numeric symbol is introduced. The pre-process to the image with printed numeric symbol is carried out. To be exact, image rotation, anti-noise and image segmentation are used, which is very important to the follow-up recognition effect directly. There are four kinds of steady and effective structure features according to the structure of printed numeric symbol: horizontal line feature, vertical line feature, horizontal through-line number and vertical through-line number. Then the encoder is constructed, recognized and programed with VC++. Thus, a fast and precious algorithm with certain anti-interference can be achieved in experiment.

Key words: printed numerals’ recognition; features extraction; through-line number; encoder