

# 一种利用灰度信息和二值信息的手写字符分割

马 瑞, 杨静宇

(南京理工大学计算机系, 南京 210094)

**摘要:** 在字符识别系统中, 字符的有效分割是识别的关键。该文描述了一种两阶段的手写字符分割方法。根据字符倾斜角度将原字符串图像划分出若干个形状不规则的区域, 每个区域含有一条分割线; 针对现有分割方法的某些局限性, 提出采用字符图像灰度信息和二值信息相结合的方式求取分割区域中的分割路径。实验结果表明了该方法对于手写字符分割的有效性。

**关键词:** 手写字符分割; Viterbi 算法; 动态规划; 灰度图像

## Segmentation Approach for Handwritten Characters Based on Gray Scale and Binary Image

MA Rui, YANG Jing-yu

(Department of Computer Science, Nanjing University of Science and Technology, Nanjing 210094)

**【Abstract】** In order to avoid the limitation of the existing segmentation algorithms based on gray-scale image or binary image, this paper presents an approach of handwritten characters segmentation which makes use of the characteristics of gray-scale image and binary image. The characters image is divided into some segmentation regions, each of which contains a segmentation path. Viterbi algorithm is employed to detect the non-linear segmentation path in each region for separating the connected characters. The algorithm performs the search process by combining the information of gray scale and binary images. Experimental results show the proposed approach is effective for handwritten characters segmentation.

**【Key words】** handwritten characters segmentation; viterbi algorithm; dynamic programming; gray-scale image

随着手写字符识别应用领域不断扩展以及分类器识别能力的提高, 字符分割技术已经成为 OCR 研究领域中的关键问题。实践表明, 字符分割不准确是产生误识别的主要原因之一, 单字正确识别率的提高在很大程度上依赖于字符分割的准确性。

目前, 普遍采用的字符切分方法有: (1) 投影分析法<sup>[1]</sup>, 主要利用字符串的垂直投影来检测分割点位置; (2) 外轮廓分析法<sup>[2]</sup>, 通过寻找上下轮廓线相对应的局部凹点来确定分割路径; (3) 基于前景背景细化法<sup>[3]</sup>, 先找出经细化后字符骨架的特征点, 再根据混和高斯概率密度来求得最佳分割路径。这些算法的共同之处是首先对灰度字符图像进行二值化, 然后在二值图像基础上执行分割算法。然而, 图像在二值化过程中经常丢失一部分相关信息, 而影响正确切分和最终识别结果。为此, 一些文献提出了直接在字符灰度图像上进行切分的方法, 但同时灰度图像中存在一些不相关的细节也常常会干扰重要的轮廓信息。

### 1 预处理

预处理主要包括二值图像的平滑以及轮廓跟踪。

字符图像经二值化后, 容易产生小内孔或笔画断裂以及存在噪声干扰等。因此, 需要对图像进行平滑处理, 提高字符图像的质量。为了尽量减少有用信息的丢失, 本文首先利用文献[4]中的方法求出字符平均笔画宽度 $S_w$ , 通过该值对字符图像进行分析, 判定是否存在笔画过细部分, 以免在滤波操作时被平滑掉。当 $S_w$ 小于某一设定阈值时, 则可以根据 $S_w$ 来选定适当大小的结构元素进行膨胀运算, 实现笔画加宽的目的。其次, 参照文献[5]中的方法去除细小的毛刺及填补轮廓中的断裂部分, 使轮廓变得更加光滑。

另外, 在字符图像平滑的同时还进行了边缘跟踪, 得到字符串的上轮廓线和下轮廓线, 提供后面的分割步骤使用。

### 2 字符分割

#### 2.1 全局参数的提取

##### (1) 字符倾角检测

由于人们的书写习惯, 字符笔画往往偏离垂直方向有一个较大的倾角, 因此分割前的重要一步就是确定字符的倾斜角度, 作为字符切分的参考方向。

目前, 有多种方法可用来检测字符倾角。考虑到由同一个人书写出来的字符几乎具有相同的角度, 本文采用基于投影的方法, 通过对整个字符串投影图的方差特性进行判断来确定字符的倾斜角度。将原字符图像朝 Y 轴方向在  $0^\circ$  到  $45^\circ$  范围内逐次递增旋转不同角度, 角度增加步长为  $1^\circ$ 。将每次旋转后的字符图像垂直投影偏差值作为倾斜角度的评价函数, 选择使投影图均方差达到最大值所对应的转角, 即为字符串的倾角。

##### (2) 字符基线提取

手写字符不仅存在角度倾斜, 而且高低不等, 大小不一。另外, 有些字符的上端笔画或下端笔画有时由于书写过长, 其倾斜角度与整体字符串略有差异。这些情况在本文所描述的分割方法中都必须考虑。因此, 根据字符串笔画的分布情况, 可将字符图像在上下方向给以界定。

设  $P_n(y)$  为二值字符图像的水平投影, 则最大投影值对应

**基金项目:** 国防基础研究基金资助项目

**作者简介:** 马 瑞(1975 - ), 女, 博士研究生, 主研方向: 模式识别; 杨静宇, 教授、博士生导师

**收稿日期:** 2006-08-20 **E-mail:** mary@yhoo.com.cn

行即为字符串的基线。

$$P_{\max} = \max_y P_H(y) \cdot \text{baseline} = \max_y \{y | P_H(y) = P_{\max}\}$$

由基线可分别得到字符串的顶线和底线：

$$\text{upBaseline} = \max_y \{y | P_H(y) \leq \frac{1}{3} P_{\max}, 1 \leq y < \text{baseline}\}$$

$$\text{lowBaseline} = \min_y \{y | P_H(y) \leq \frac{1}{4} P_{\max}, \text{baseline} < y \leq H\}$$

其中， $H$  为字符图像高度。由顶线、底线可将字符串图像确定为上、中、下 3 个区域，中间区域是字符串的主体部分。

## 2.2 分割区域的确定

通常采用的垂直切分方法极易产生误分割。如果按照字符书写的倾斜方向进行切分，则会使这种误分割大大减少。因此，本文在求取分割路径前先根据字符倾角确定出隐含连接字符间分界线的纵向分割区域来。对于不同的应用对象，例如字母和数字的拓扑结构有所区别，相应的划分过程也略有不同。以手写数字为例，由于人们日常的书写习惯，有些字符常常出现笔画不闭合的现象，如开口 0、开口 4、开口 6、开口 8。为避免这种现象产生不必要的过分割，需要按照由下向上的次序进行划分，具体过程有以下 3 步（如图 1 所示）：

(1) 以下轮廓线上位于基线下方的所有局部极小值点为出发点，沿字符倾斜方向向下进行画线。特别地，在画线过程中当遇到某些字符下端笔画的轮廓点时，为了使字符不受损坏，需要改变原来的画线方向，从遇到的笔画第 1 个轮廓点开始进行边缘跟踪，直到轮廓倾斜方向发生陡变，并记下陡变处对应的轮廓点作为笔画端点。然后在局部极大值点与笔画端点之间重新连线，目的是绕过笔画，以笔画端点作为新的起始点，沿字符倾斜方向继续向下画线，直到图像底部为止。

(2) 该步骤同样以下轮廓线位于基线下方的每一个局部极小值点作为起始点，沿字符倾斜方向向上划线，直到字符顶线为止。需要注意的是，画线时要尽量避免穿过白像素点。也就是说，当切线经过一个白像素时，应先判断其左右邻域是否存在黑像素，如果有则选择黑像素点进行画线，否则保持不变。

(3) 从字符串顶线开始沿字符倾斜方向向上画线，直到图像顶部为止。类似地，当画线过程中遇到字符上端笔画的轮廓点时，处理过程同步骤 1。



图 1 分割区域的确定

如果以英文字母为应用对象，则按照从上向下的次序进行划分，此时分割区域由上轮廓线的局部极大值点以及字符底线来确定，具体过程与上类似。

## 2.3 分割路径的求取

隐马尔可夫模型中的 Viterbi 搜索算法是一种动态规划算法，本文利用它融合字符图像的灰度信息和二值信息来获取非线性分割路径。

对于给定字符图像中某一个高度为  $H$  的分割区域，可将其构建为一个自上向下的  $H$  层有向图（如图 2 所示） $G = \langle V, E \rangle$ ， $V = \{V_i | i = 1, 2, \dots, H\}$ ，其中  $V_i = \{v_{ij} | 1 \leq j \leq N_i\}$  表示第  $i$  层结点集合， $N_i$  为第  $i$  层结点数目，且对于任意  $1 \leq i_1, i_2 \leq H, i_1 \neq i_2$ ， $N_{i_1} \neq N_{i_2}$ ，结点  $v_{ij}$  对应于图像第  $i$  行像素  $(i, j)$  的坐标； $E = \{ \langle v_{ij}, v_{i+1,k} \rangle, j-1 \leq k \leq j+1, 1 \leq i \leq H \}$  表示从第  $i$  层结点到第  $j$  层结点的有向边集合。于是，将 Viterbi 算法

应用到该有向图中，找出顺向首尾相接的一串有向边集合，就可以得到分割路径。

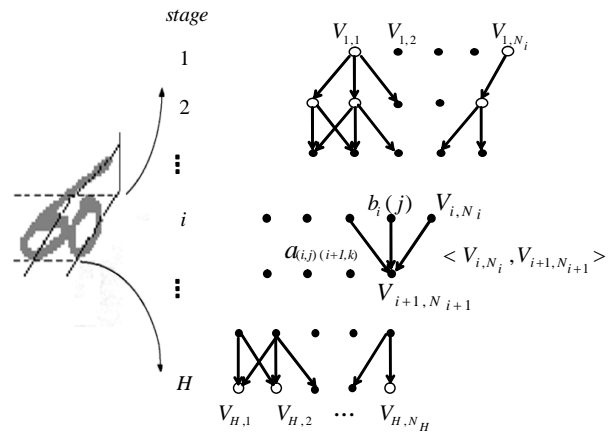


图 2 一个分割区域的有向图表示

理想的最佳分割路径应尽量趋近于字符的轮廓边界，并且路径穿越的黑点数也力求最少。因此，本文主要利用某种惩罚机制来确定分割路径的代价，作为衡量其优劣的标准，则代价最小的路径能够达到最优分割。于是，求取连接字符间分割路径的问题就转化为寻找一条最小代价路径的过程。实现此算法前的一个重要步骤是对下面几个参数值的设定：

(1) 结点转移权值  $a_{(i,j)(i+1,k)}$ ， $1 \leq i \leq H, 1 \leq j \leq N_i, j-1 \leq k \leq j+1$

$a_{(i,j)(i+1,k)}$  表示第  $i$  层结点  $j$  到第  $i+1$  层结点  $k$  的转移权值，这里借助它来决定解码时分割点方向的选择。对于水平文本行的分割，规定第  $i$  层的结点  $(i, j)$  只能向下射出 3 条有向弧，其邻接点分别为  $(i+1, j-1)$ ， $(i+1, j)$ ， $(i+1, j+1)$ 。为了使分割路径尽量保持线性，对结点的 3 种转移方式进行了控制，设定不同权值为

$$a_{(i,j)(i+1,k)} = \begin{cases} 1 & \text{if } k = j-1 \\ \alpha & \text{if } k = j \\ 1 & \text{if } k = j+1 \end{cases}, \quad 0 < \alpha < 1$$

(2) 结点代价函数  $b_i(j)$ ， $1 \leq i \leq H, 1 \leq j \leq N_i$

$b_i(j) = S_w \times f(i, j) + (2 - i/H) \times g(i, j)$  表示分割路径经过第  $i$  层结点  $j$  的代价。其中， $f(i, j)$  和  $g(i, j)$  分别为像素点  $(i, j)$  在二值图像和灰度图像中的像素值。由二值图像中的黑像素来表示字符轮廓，权值大小设为笔画宽度，通过该权值可以使分割路径穿过的笔画数尽量最少。另一方面，在利用灰度信息时，如果两个像素点具有相同的灰度级，则对  $Y$  轴位置较高者赋予较大权值，其目的是保证当分割路径不得不穿越字符笔画时，会尽可能地趋近于字符图像的底部。可以看出，一条最优分割路径应满足路径上所有结点的总代价最小。

(3) 结点初始状态  $\pi_j = g(1, j)$ ， $1 \leq j \leq N_1$

寻求分割路径的过程相当于在有向图中以第 1 层为起点，第  $H$  层为终点，沿着有向边的方向找出从起点到终点所有路径中的最佳者。这里，引入前向变量  $\delta_i(j)$  表示沿某条路径到达第  $i$  层时结点  $j$  的最小代价；用  $\psi_i(j)$  来记录第  $i-1$  层中使第  $i$  层结点  $j$  获得最小代价的路径结点，则 Viterbi 搜索算法的简要计算过程如下：

Step 1 初始化第 1 层所有结点

$$\delta_1(j) = \pi_j = g(1, j), \quad 1 \leq j \leq N_1, \quad \psi_1(j) = 0$$

Step 2 计算所有从第 2 层到第  $H$  层各结点的最小代价，并记下每个结点的上一层最优路径结点

$$\delta_{i+1}(j) = \min_{j-1 \leq k \leq j+1} \delta_i(k) \cdot a_{(i,j)(i+1,k)} \cdot b_{i+1}(j), \quad 1 \leq i \leq H-1, 1 \leq j \leq N_i$$

$$\psi_{i+1}(j) = \arg \min_{j-1 \leq k \leq j+1} (\delta_i(k) \cdot a_{(i,j)(i+1,k)})$$

Step 3 计算第  $H$  层最小代价的结点

$$\delta^* = \min_{1 \leq j \leq N_H} \delta_H(j), \quad j_H^* = \arg \min_{1 \leq j \leq N_H} \delta_H(j)$$

显然,从第  $H$  层各结点向第 1 层回溯所得到的最小代价结点  $j_i^* = \psi_{i+1}(j_{i+1}^*)$ ,  $i = H-1, H-2, \dots, 1$  的连线,即为该区域的分割路径。

### 3 实验结果

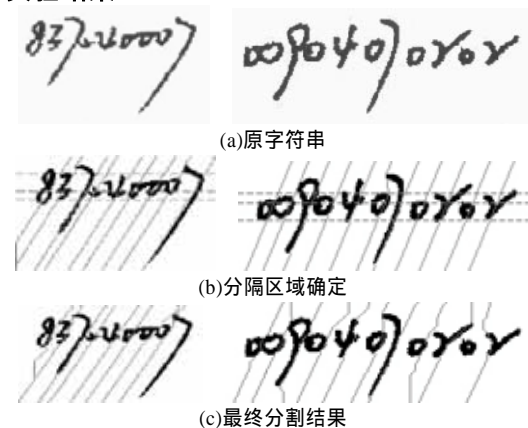


图 3 字符串分割结果示例

以银行支票上手写小写金额字符串为实验数据,从我国银行实地采集的现行支票中提取出 400 例手写数字串,共包含 4 375 个字符,应用本文的分割方法进行了分割处理,

(上接第 164 页)

#### 3.2 公平性

顾客只有先发送支付指令,才能获得服务;商家只有得到顾客的服务确认信息才能得到钱。服务确认信息可以促进诚信交易,缺少它,对顾客和商家都没有好处。

#### 3.3 运行效率

##### (1) 顾客

顾客发送一次支付指令,可以支付任意数值的金额(不能超过微支付每次支付金额的上限,一般为 10 美元),不必产生 Hash 链或进行多次 Hash 运算。在产生顾客指纹信息和服务确认信息时,顾客需要各做一次 Hash 运算。顾客需要保存  $K_{BC}$  和  $C_C$ ,需要防止  $K_{BC}$  泄露。当  $K_{BC}$  泄露后,顾客需要到银行申请更改  $K_{BC}$ 。

##### (2) 商家

商家需要验证支付指令中顾客的证书,需要针对每个顾客保存一个  $T_{last}$  值。商家还需要保存  $K_{BV}$ ,在产生商家指纹信息时,需要做一次 Hash 运算。

##### (3) 银行

银行需要针对每个顾客,在一定时期内,签署一个证书(一次签名操作);需要定期维护  $I_C$ 、 $K_{BC}$ 、 $K_{BV}$  以及至少一周内所有顾客的网上交易的记录。银行需要验证顾客的证书(一次签名验证操作),验证顾客和商家的指纹信息(两次 Hash 运算),需要在一周内该顾客所有交易记录的时间戳中查找  $T_C$  来识别商家的重放。这些都可以离线完成。

#### 3.4 方案比较

与 PayWord 相比, MicroPay 方案中不需要顾客的承诺,

图 3 为字符串的分割结果示例。从实验结果上看,字符完全分割正确率为 89.06%。实验中,依据最近邻法对分割结果进行了识别衡量,正确识别率达到 81.45%,说明该方法的有效性。本文方法同样适用于英文字母的分割,并且质量较差的灰度字符图像比黑白图像更适合于此方法。

### 4 结论

本文描述了一种两阶段的手写字符分割方法,并针对现有基于灰度图像或二值图像切分法的局限性,提出采用融合灰度信息和二值信息的方式来求取分割路径,取得了较好的分割效果。为了进一步完善该方法,如何更准确地分割发生严重倾斜的字符串是今后工作的研究方向。

#### 参考文献

- 1 赵斌,苏辉,夏绍玮.一种无约束手写体数字串分割方法[J].中文信息学报,1998,12(3):21-28.
- 2 Li Yun, Naoi S, Cheriet M, et al. A Segmentation Method for Touching Italic Characters[C]//Proceedings of the 17th International Conference on Pattern Recognition. 2004, 2: 594-597.
- 3 Chen Y K, Wang J F. Segmentation of Single or Multiple Touching Handwritten Numeral String Using Background and Foreground Analysis[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(11): 1304-1317.
- 4 陈艳,孙羽菲,张玉志.灰度图像中字符切分方法的研究[J].中文信息学报,2004,18(4):44-49.
- 5 Lee W, Lee D J, Park H S. A New Methodology for Gray Scale Character Segmentation and Recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1996, 18(10): 1045-1050.

也没有以 Hash 链来生成货币,自然避免了多次签名验证操作和商家及银行对货币 Hash 链的反复多次运算,减小了系统的总运算量,提高了运行速度,具有较高的效率。与文献[4]相比, MicroPay 也能完成不同面额的支付,而且因为 MicroPay 提供的是金额数而不是 Hash 货币,彻底突破了 Hash 货币对面额的限制。和文献[5]相比, MicroPay 也能提供面向多个商家的解决方案。当面对  $k$  个商家交易时,文献[5]的方案需要顾客做出一次签名(顾客的承诺),  $2k$  次商家对顾客签名及银行证书的验证,银行需要对每笔交易验证证书; MicroPay 需要  $k$  次商家对银行证书的验证,银行需要对每笔交易验证证书。从传输及存储空间来看,文献[5]中顾客的承诺会随着商家的增多而线性增长,导致支付指令的变长,增加存储和传输的数据量,而 MicroPay 的支付指令是定长,和商家的个数无关。相比之下, MicroPay 的运行效率更高。

#### 参考文献

- 1 钟诚.电子商务安全[M].重庆:重庆大学出版社,2004.
- 2 李明柱.基于 PayWord 的 WWW 微支付模型[J].北京邮电大学学报,2002,25(2).
- 3 张宽海.网上支付结算与电子商务[M].重庆:重庆大学出版社,2004.
- 4 谭运猛,付雄,郎为民.基于多 PayWord 链的新型高效微支付方案[J].华中科技大学学报,2004,32(5).
- 5 王晓明,符方伟.一个安全的多方交易微支付方案[J].计算机工程,2005,31(2).

