

一种有效的手写汉字多步分割方法

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。针对手写汉字字间距及字内距无规则可循,字符间极易发生粘连、交错等现象,提出一种多步分割方法。该方法首先利用 Viterbi 算法将原字符串切分成互不连通的分割块,使非粘连汉字、交错汉字得到正确分割;对于其中宽度较大存在粘连字符的分割块,从候选分割点入手,用非线性分割路径将粘连部分分开;最后再应用 A* 算法找到全局最佳分割位置,使过分割的字符得到完整合并。实验结果表明,该方法对于手写汉字的分割是可行、有效的。

关键词 手写汉字字符分割 Viterbi 算法 最短路径 A* 算法 动态规划

中图分类号: TP391.43 **文献标识码:** A **文章编号:** 1006-8961(2007)11-2062-06

An Effective Multi-stage Segmentation Method for Handwritten Chinese Characters

MA Rui, YANG Jing-yu

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

Abstract Handwritten Chinese characters segmentation is crucial to their recognition in OCR system. In practice, handwritten Chinese characters may be written touching or overlapping each other, and between-character gap or inter-character gap has large variations. For the problem posed by above situations, a multi-stage method of handwritten Chinese characters segmentation is proposed. Firstly, Viterbi algorithm is employed to obtain the no touching or overlapping characters from the separated blocks of the characters string. Then, for the touching characters in the wide blocks, the segmentation is performed by searching non-linear segmentation paths which pass through the candidate segmentation points. Finally, A* algorithm is applied to determine the optimal segmentation paths. As a result, over-segmentation components are merged into whole characters. Experimental results show the proposed method is feasible and effective for off-line handwritten Chinese characters segmentation.

Keywords handwritten Chinese character segmentation, viterbi algorithm, shortest path, A* algorithm, dynamic programming

1 引言

随着字符识别技术的日益成熟及其越来越广泛的应用,字符分割逐渐成为研究中的热点。汉字因其书写的随意性,字体形式多变,字符间极易发生粘连、交错等现象,成为字符分割中的一个难点。近年

来,针对字母和数字的分割技术已取得较好进展,但汉字字符结构比较复杂,与英文字母和数字区别较大,如果采用字母和数字的一些分割方法,显然不太适用。目前对于非限定手写汉字的分割,很少有文章提及到。一般地,手写汉字的相互位置关系有相离、粘连、交错等几种情况,采用普遍的投影法或连通域法^[1],容易将相离或不粘连的部件分开,然后

基金项目:国防基础研究项目(J1500C002)

收稿日期:2006-02-17; 改回日期:2006-07-06

第一作者简介:马瑞(1975~),女,现为南京理工大学计算机与科学系在读博士研究生。主要研究方向为图像处理、模式识别。

E-mail: marui_mary@yahoo.com.cn

将几个分离的部件适当地合并成一个完整的汉字。然而对于字符间发生严重粘连或交叠的情况二者却很难处理,主要原因是找不到准确的切分位置,造成分割结果中粘连字符的错误合并,因此显现出很大的局限性。文献[2]介绍了一种粗切分和细切分相结合的方法,首先通过字符垂直投影分析得到粗切分结果,再对粘连字块的背景图像作细化处理,找出特征点,利用模糊决策准则求出合理的分割路径,然而这种评价分割路径的模糊决策标准不易被确定;文献[3]是从汉字的笔画分析入手,先提取汉字串所有笔划,利用笔划外接框的几何特征按相交和相等不同位置关系对笔划进行合并,来得到最终完整的汉字。该方法对汉字字符间距较小的情况能够较好地解决,但对所有汉字笔划的检测耗时量较大;文献[4]在连通元与汉字单字之间建立了“组件”的概念,以组件为基础,根据两个组件之间的上下、左右和包围关系对组件进行合并,从而实现汉字的分割,但该方法没有考虑组件左中右结构的情况。

本文提出一种多步分割方法,将整个汉字字符串的切分过程分为3个阶段进行:首先利用 Viterbi 算法将非粘连字符及交错字符分开,得到原字符串的初始分割;然后对于粘连字符,通过寻找候选分割点,使用最短路径法做进一步非线性切分;最后应用 A* 算法找到全局最佳分割路径。本文主要工作如下:(1)对于粘连字符的分割,吸取最短路径的基本思想^[5]采用从分割点向上、下两侧寻找最小代价路径的分割方法,较原方法将最短路径的搜索应用于整幅图像来讲,避免了大量冗余路径的求取,并且这种双向搜索表示寻找约束在分割点意义上的最小代价路径,更具有针对性;其次,文献[2]采用反复跟踪背景骨架来寻找粘连切分路径,过程复杂繁琐,相比之下,该方法简捷清晰,搜索范围小。(2)将 AI 中 A* 搜索算法应用到全局最佳分割位置的确定中,在选取适当的启发函数下寻找最优分割路径,与文献[2]中用模糊决策规则对候选分割路径进行分类相比,评价标准更直接,求解过程更快捷、准确。

2 预处理

预处理的目的是为了使得汉字二值图像中保留更多的字符信息,改善图像质量,为分割环节做准备,其中主要处理包括二值化、去噪声和图像增强。

首先可以采用类间最大方差法对原字符图像进

行二值化处理^[6],在此过程中可能会丢失一部分相关信息,或者一些不相关信息被保留,如字符笔划发生断裂或不必要的粘连,以及字符背景存在噪声干扰,而影响切分和识别的准确性。为此,可以使用形态学滤波器^[7]对图像进行滤波,选择圆盘形结构元素对字符二值图像进行闭运算接着再进行开运算,来消除细长毛刺,填补轮廓中断裂部分。这样既能消除噪声;又能保持图像中的细节信息,使字符轮廓变得更加光滑,达到了平滑的目的。

3 非粘连汉字的分割

目前,字符串分割技术中应用较广的方法主要是投影分析和连通域分析^[1]。事实上,利用二者也可以将书写比较均匀的字符分开,但对于字间距较小发生交错的字符,投影分析法分割准确率不高;且分割路径只能垂直进行,给字符带来损伤而影响后期识别;连通域分析法则需要大量计算各连通元的位置、大小等复杂过程。文献[3]提出根据笔划位置关系进行合并的方法,能较好地解决大多数手写体汉字字间距较小而造成的困难,但在两个距离较近的字符宽度约为其他字符宽度一半的情况下,分割结果不准确。因此,本文采用 Viterbi 算法来完成对非粘连汉字的分割。

3.1 Viterbi 算法

隐马尔可夫模型(hidden markov model, HMM)是一种用参数表示的两机理的随机过程。一个是隐含的有限状态马氏过程,另一个是观测序列。观测序列由状态过程通过一组概率分布来决定。

手写字符中的 HMM 模型结构主要由状态以及状态之间的转移方式来描述。考虑到手写汉字的随意性和时间耗费,模型的描述既不宜过细,又要体现汉字的笔划分布特点,可以将原字符串图像 $I_{w \times h}$ 划分为若干个互不重叠的网格区域,则原图像可表示为

$$I = \{v_{ij} | 1 \leq i \leq m, 1 \leq j \leq n\}$$

其中, $m = \lceil H/W_s \rceil$, $n = \lceil W/W_s \rceil$, W_s 为笔划宽度的估计值^[8]。此时字符串图像可视为一个自上向下的 m 层单向图,结点代表图像中网格区域。建立一个 HMM 模型来表述该有向图,每个结点对应于状态, n 是状态总数,有向边表示状态的转移方向,用结点轨迹组成观测序列,其概率分布为分割路径穿过结点的几率大小, m 是观察序列的长度。

HMM 模型中的 Viterbi 搜索算法^[9]是一种动态规划算法,它通过前向迭代过程来解决 HMM 的解码问题。每种可能的状态转移根据前面输出的观测值计算当前状态的概率,然后选择最大概率路径。本文利用它寻求分割路径,相当于在图中沿着有向边方向找出所有路径中的最大概率者,组成顺向首尾相接的一串有向边的集合,即得到非线性的分割路径,这一点优于投影法。下面对模型中的参数进行设置:

(1) 状态转移概率矩阵

$$A = \{a_{ij}\}, 1 \leq i, j \leq n$$

这里只考虑 3 种转移方式,即状态的自转移及向前后相邻状态的转移。为使分割路径尽量保持线性,对状态 3 种转移方式进行了控制,设定转移概率

$$a_{ij} = \begin{cases} \alpha & j = i - 1 \\ 1 & j = i \\ \alpha & j = i + 1 \end{cases}$$

$$0 < \alpha < 1$$

(2) 观测符号概率矩阵

$$B = \{b_k(j)\}, 1 \leq k \leq n, 1 \leq j \leq m$$

$$b_k(j) = 1 - \frac{P_{k,j}}{W_s \times W_s + 1}$$

表示状态 k 在第 j 层的观测符号发生概率, $P_{k,j}$ 为网格 v_{ij} 内的黑像素数目

(3) 初始状态概率

$$\pi_i = \left(\sum_{g=1}^{m/3} b_i(g) \right) / \frac{m}{3}, 1 \leq i \leq n$$

终止状态概率

$$\gamma_i = \left(\sum_{g=2m/3}^m b_i(g) \right) / \frac{m}{3}, 1 \leq i \leq n$$

这里引入前向变量 $\delta_i(k)$ 表示沿某条路径到达第 j 层时网格点 k 的最大概率; $\varphi_j(k)$ 用来记录第 $j-1$ 层中使第 j 层的网格点 k 获得最大概率的路径结点,则 Viterbi 搜索算法的简要推导过程如下:

初始: $\delta_1(k) = \pi_i b_k(1), \varphi_1(k) = 0 \quad 1 \leq k \leq n$

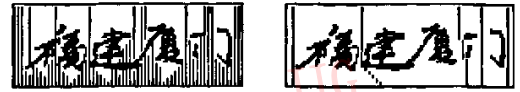
递推: $\delta_{j+1}(k) = \max_{i-1 \leq i \leq i+1} \delta_j(i) a_{ik} b_k(j+1)$

$$\varphi_{j+1}(k) = \arg \max_{i-1 \leq i \leq i+1} (\delta_j(i) a_{ik}) \quad 1 \leq j \leq m-2, 1 \leq k \leq n$$

终止: $\delta_m(k) = \max_{i-1 \leq i \leq i+1} \delta_{m-1}(i) a_{ik} b_k(m) \gamma_k$

$$\varphi_m(k) = \arg \max_{i-1 \leq i \leq i+1} (\delta_{m-1}(i) a_{ik} \gamma_k) \quad 1 \leq k \leq n$$

显然,从第 m 层各网格点向第 1 层回溯所得到的具有最大概率的结点序列就构成了原字符串的基本分割路径,如图 1(a) 所示。



(a) 基本分割路径

(b) 候选分割路径

图 1 利用 Viterbi 算法对字符串进行分割

Fig. 1 Segmentation of non-touching characters

3.2 候选路径的确定

上面求得的分割路径是指在 Viterbi 意义上的,对于整个字符串分割而言还很粗略,其中存在大量冗余路径。因此,需要利用下述知识规则加以剪裁。

规则 1: 对于路径 $Path_i$ 和 $Path_j$

如果 $\{i_1, i_2, \dots, i_m\} \cap \{j_1, j_2, \dots, j_m\} \neq \emptyset$, 且 $\delta_m(i) \leq \delta_m(j)$, 则保留 $Path_j$ 。

规则 2: 对于路径 $Path_{i+1}, Path_{i+2}, \dots, Path_{i+c}$

如果 $\delta_m(k) = 1, i+1 \leq k \leq i+c$, 则保留 $Path_{i+c/2}$ 。

规则 3: 设 $D_{i,j}$ 为路径 $Path_i$ 和 $Path_j$ 间的距离^[9], 如果 $D_{i,j} \leq T_{th}$, 并且 $\delta_m(i) \leq \delta_m(j)$, 则保留 $Path_j$ 。

规则 4: 如果 $\delta_m(i) < \left(1 - \frac{W_s^2}{2(W_s^2 + 1)} \right)^{c_1} \cdot \alpha^{c_2}$,

$1 \leq i \leq n$, 则删除 $Path_i$ 。

其中, c_1 和 c_2 分别为调节参数, c_1 表示路径中允许通过黑像素网格点的最大数目, c_2 表示路径轨迹中选择对角线方向的最多次数。

经上述规则剪裁而保留下来的路径成为候选分割路径,如图 1(b) 所示,它们将原字符串切分成多个分割块,其中字间距较为明显和字间距较小的非粘连汉字能被有效地分开。

当然对于一些字内距较大,字内部件发生分离的汉字,在此过程中可能被错误地分成几部分,这种过分割现象将需进行消除。

4 粘连汉字的分割

粘连的两个汉字极易被当成一个字,造成分割不足。由于汉字的方块字特点,汉字字宽不应超过一定的范围。因此,需要对较宽的分割块进一步拆分。由于实际书写出来的汉字字宽和字间距总有一定的变化,没有一个恒定的值作为判断标准,本文采取逐步求精的方法,由所有分割块估算出汉字的平均宽度 W_s , 对宽度大于 W_s 某个倍数的分割块进行切分。随着分割块大小的不断变化,重新估计汉字

平均字宽,直到最终结果不存在粘连字符为止。

对于粘连字符的具体切分过程,主要包括确定候选分割点和寻找候选分割路径。

4.1 分割点的确定

首先要在粘连字符中找出可能存在分割点的笔段,然后判断出笔段上哪个特征点为分割点。实际上,水平汉字间粘连的原因主要是书写时的连笔和两个距离较近的字符有较长水平笔划、斜笔划彼此连接。根据这一特点,可以提取较长的非垂直笔段,分析它们在图像中所处的位置,找出可能存在的分割点,从而将粘连汉字分开。

(1) 首先在经过细化处理^[10]后的字符图像中找出端点和分支点,记下起点和终点间的笔段,对其进行如下分析,认为满足条件者可能存在分割点。

① $f_{ho}(C_i, C_j) = 0$, f_{ho} 表示将笔段从其所在的连通元删除后,原连通元分为两个左右独立的连通元 C_i 和 C_j 间的水平重叠部分。

② $l_i > T_l$

③ $-\frac{\pi}{2} + \beta_2 < \alpha_i < \frac{\pi}{2} - \beta_1$

④ $|c_i - W_b/2| < T_c$

其中, l_i 为笔段长度; α_i 为笔段倾斜角; c_i 为笔段中心 x 位置; W_b 为分割块宽度。 T_l , β_1 和 β_2 , T_c 分别为设定的阈值。

(2) 大多数情况下,汉字笔划在粘连处常常形成分支点和角点。对于分支点可通过判断与之相连的点数很容易找到,而角点是曲线上的转折点,其附近的骨架线梯度变化应最大。因此,在上面求得的笔段上对各点进行判断,若为角点应满足:

① $p(x, y)$ 到笔段两端 p_0, p_1 连线的投影 $Pro(x, y)$ 最长;如果 $\max(Pro(x, y)) < T_{pro}$, 则认为该笔段不存在角点。

② $p(x, y)$ 与 p_0, p_1 所形成向量间的夹角 $Ang(x, y)$ 最小,如果 $\min(Ang(x, y)) > T_{Ang}$, 则认为该笔段不存在角点。其中, T_{pro}, T_{Ang} 为设定的阈值。

由此得到的角点以及分支点成为候选分割点,如图 2(a)所示。



(a) 符号“■”表示候选分割点 (b) 候选分割路径

图 2 对粘连字符的分割

Fig. 2 Segmentation of two-touching characters

4.2 分割算法的实现

候选分割点确定后,如何从字符粘连处出发寻找适当的切割路径十分重要。许多粘连字符之间存在重叠部分,常常会将一个字符的一部分划分到另一个字符中。本文沿用最短路径的分割方法^[5],但与原方法在求取路径方式上有所不同,即采取双向寻求最短路径策略:以图像中的字符粘连处为起点,分别向上下逐步分析最小代价来寻找分割路径,最终的完整分割路径由上下两部分路径的并来确定。

以向下搜索为例,从候选分割点 (x, y) 出发,每次选择下一处落脚点的位置只有 3 种可能:垂直方向和左、右对角线方向,并规定惩罚原则如下:

$$trans = \begin{cases} 0 & p(x+1, y) = 0 \\ 10 & p(x+1, y) = 1 \\ 1 & p(x+1, y-1) = 0 \text{ 或 } p(x+1, y+1) = 0 \\ 14 & p(x+1, y-1) = 1 \text{ 或 } p(x+1, y+1) = 1 \end{cases}$$

向上搜索惩罚原则与之类似。那么对于一条完整路径的搜索主要有以下几步:

(1) $Path_{up} = \emptyset, Path_{down} = \emptyset$

(2) From $x-1$ to 2

$Path_{up} = Path_{up} \cup \{ \text{上一行代价最小的像素点} \}$

(3) From $x+1$ to $H-1$

$Path_{down} = Path_{down} \cup \{ \text{下一行代价最小的像素点} \}$

H 为字符图像的高度。

(4) $Path = Path_{up} \cup \{ (x, y) \} \cup Path_{down}$

显然,一个最小累积代价路径会尽可能地趋近于字符的轮廓边界,并且路径上的黑像素数目也力求最少,因此能够达到最优分割。为了使分割结果更精确,可以在分割点附近大小为 W_s 的区间内进行最小代价路径的选取来获得粘连字符的候选分割路径,如图 2(b)所示。

5 最佳分割路径的确定

前面,绝大多数汉字都得到了有效地分割,但是由于某些不正确分割路径的存在,导致了过分割现象。为了从候选分割路径中挑选出正确的来,可认为所有的分割路径构成一个全集,每个子集都是一种分割可能,取置信度最高的作为最终分割结果。由于在寻找一组最优分割路径的过程中会产生很多分支,本文利用启发式搜索,即在搜索过程中,通过动态规划算法确定候选分割路径的最小分割代价,将其作为置信度的衡量标准,对每一个搜索位置进

行评价,从最好的位置向下搜索,以提高搜索效率。

A*算法^[1]是一种典型的启发式搜索算法,通过选择合适的估价函数,在启发函数的制约下,指导搜索沿最有希望的分支进行,以求得最优解。算法中对位置的估价函数可表示为

$$f(s) = g(s) + h(s)$$

其中, $f(s)$ 是结点 s 的估价函数, $g(s)$ 是开始结点到结点 n 的实际代价,它是已知的, $h(s)$ 是结点 s 到目标结点最佳路径的估计代价,它体现了搜索的启发信息。将此算法应用到最终分割路径的选取过程中,构建一个有向图(如图 3 所示)来表示当前的字符图像,图中的结点对应候选分割路径。于是求取最终分割结果就转化为寻找一条从开始结点到目标结点最小代价的路径。

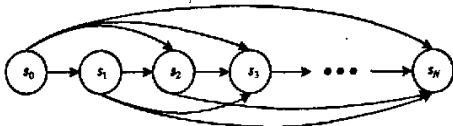


图 3 分割路径按自左向右的顺序构成有向图,每个结点代表一条候选分割路径

Fig. 3 The directed graph composed of candidate segmentation paths, a node represents a path

实现 A* 算法的关键是选取适当的启发函数,为此给出:

(1) 路径 $Path_i, Path_j$ 间的代价函数:

$$Cost_{i,j} = \frac{W_b}{p[\text{width} = W_b]}$$

其中, W_b 为路径 $Path_i$ 和 $Path_j$ 间的分割块 $C_{i,j}$ 的宽度; $p[\text{width} = W_b]$ 表示存在宽度为 W_b 字符的概率,可近似认为 $p[\text{width} = W_b] = P_1 * P_2$, 这里 $P_1 = \frac{\min(\bar{W}_c, W_b)}{\max(\bar{W}_c, W_b)}$ 表示 $C_{i,j}$ 的宽度与平均字宽的差异程度,二者的差别越小, $C_{i,j}$ 为字符的可能性就越大; $P_2 = \frac{\min(W_b, H_b)}{\max(W_b, H_b)}$ 表示 $C_{i,j}$ 高度与宽度的差异程度,分割块高、宽差别越不明显,则越可能是一个字符, \bar{W}_c, H_b 分别为汉字平均字宽和分割块汉字高度。

(2) 结点的启发函数:

$$\hat{g}(s_0) = 0, \hat{g}(s_n) = \hat{g}(s_{n-1}) + Cost_{s_{n-1}, s_n}$$

$$\hat{h}(s_0) = W, \hat{h}(s_n) = \hat{h}(s_{n-1} - 1) - W_b$$

$$\hat{f}(s_n) = \hat{g}(s_n) + \hat{h}(s_n)$$

其中, W 为整个字符串宽度。

(3) 为减少计算时间消耗量,规定判决条件

$$\langle \hat{f} \rangle = \frac{1}{N} \sum_{i=1}^N \hat{f}(i)$$

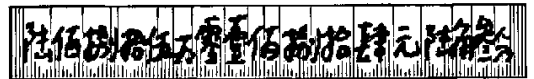
其中, i 为 A* 算法 OPEN 表中的结点; N 为 OPEN 表中的结点数。若结点 i 使得 $\hat{f}(i) > \langle \hat{f} \rangle$, 则舍弃该点。

6 实验结果

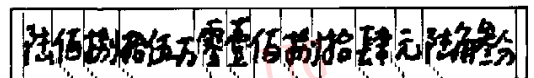
以银行支票上手写体大写金额字符串为例,实验数据为我国银行实地采集的 400 张现行支票,从中提取出手写体大写金额字符串图像。图 4 为其中一个字符串的分割过程。图 4(b)、(c) 是对非粘连汉字的分割,得到非粘连字符如“肆”、“元”等以及交错字符“壹”和“佰”,而第 10 个字符“捌”出现过分割现象;图 4(d)、(e) 是对粘连汉字的分割;图 4(f) 为最终分割结果,通过摒弃不正确的路径,使过分割字符“捌”得到了完整合并。



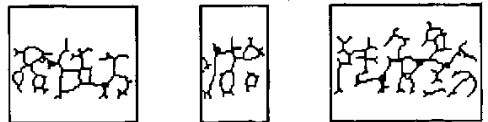
(a) 原字符串图像



(b) 利用 Viterbi 算法得到原字符串的基本分割路径



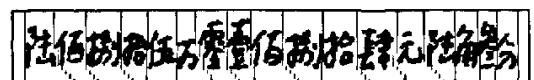
(c) 经规则剪裁后得到的候选分割路径



(d) 粘连汉字候选分割点的确定



(e) 对粘连汉字分割后得到的所有候选分割路径



(f) 利用 A* 算法求得的最佳分割结果

图 4 字符串的整个分割过程

Fig. 4 Example of the proposed total segmentation stages for a handwritten legal amount string

表 1 给出了分割统计结果,从中可以看出,该方法获得了较好的分割效果。

表 1 分割统计结果比较

Tab. 1 Comparisons of several segmentation algorithms statistical results

方法	分割成功率 (%)	过分割数	分割不足数	错误分割数
文献[2]	81.4	126	425	142
文献[3]	75.3	355	113	452
文献[4]	74.1	373	118	474
本文	82.6	118	398	133

注:400 个字符串共 3 726 个字符

7 结 论

针对手写汉字形式变化多端,字符间极易发生粘连、交错等现象而造成的分割困难,提出了一种切实有效的手写汉字多步分割方法,对于交错、单处笔划粘连等字符能够得到较好的解决,但对于不常见的字符间发生多处粘连的情况,由于很难找到一条经过所有分割点的路径,因此得不到正确结果,进一步的研究对复杂粘连字符的分割。

参考文献 (References)

- 1 Lu Y, Shridhar M. Character segmentation in handwritten words: An overview[J]. *Pattern Recognition*, 1996, 29(1): 77 ~ 96.
- 2 Zhao S Y, Chi Z R, Shi P F, et al. Two-stage segmentation of

- unconstrained handwritten Chinese characters [J]. *Pattern Recognition*, 2003, 36(1): 145 ~ 156.
- 3 Wang Rong, Ding Xiao-qing, Liu Chang-song. Handwritten Chinese address segmentation and recognition based on merging strokes[J]. *Journal of Tsinghua University (Science and Technology)*, 2004, 44(4): 498 ~ 502. [王嵘,丁晓青,刘长松. 基于笔划合并的手写体信函地址汉字切分识别[J]. 清华大学学报(自然科学版), 2004, 44(4): 498 ~ 502.]
- 4 Lu Yue, Shi Peng-fei, Zhang Ke-hua. Segmentation of free-format handwritten Chinese characters based on structure features of characters[J]. *Acta Electronica Sinica*, 2000, 28(5): 102 ~ 104. [吕岳,施鹏飞,张克华. 基于汉字结构特征的自由格式手写体汉字切分[J]. 电子学报, 2000, 28(5): 102 ~ 104.]
- 5 Wang J Jean. Segmentation of merged characters by neural networks and shortest path[J]. *Pattern Recognition*, 1994, 27(5): 649 ~ 658.
- 6 Cheriet M, Said J N, Suen C Y. A recursive thresholding technique for image segmentation[J]. *IEEE Transactions on Image Processing*, 1998, 7(6): 918 ~ 921.
- 7 Gonzalez R C. *Digital Image Processing (2nd ed)* [M]. Beijing: Publishing House of Electronics Industry, 2002: 429 ~ 431. [冈萨雷斯. 数字图像处理(第二版,英文版)[M]. 北京:电子工业出版社, 2002: 429 ~ 431.]
- 8 Arica N, Yarman-Vural F T. Optical character recognition for cursive handwriting[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002, 24(6): 801 ~ 813.
- 9 Tseng Y H, Lee H J. Recognition-based handwritten Chinese character segmentation using a probabilistic Viterbi algorithm[J]. *Pattern Recognition Letter*, 1999, 20(8): 791 ~ 806.
- 10 Lin J Y, Chen Z. A Chinese-character thinning algorithm based on global features and contour transformation[J]. *Pattern Recognition*, 1995, 28(4): 1445 ~ 1458.