

# 基于模糊区域检测的手写汉字笔画提取方法

曹忠升 苏哲文 王元珍 熊鹏

(华中科技大学计算机科学与技术学院, 武汉 430074)

**摘要** 针对手写汉字笔画提取的重点和难点——模糊区域的识别和解析问题,提出了一种新的基于模糊区域检测的笔画提取算法。该算法首先利用细化算法提取的 fork 候选点和 fork 候选点附近的轮廓信息来检测模糊区域;然后利用图模型来对子笔画和模糊区域进行建模,同时通过构造贝叶斯分类器来分析子笔画对的连续性,并通过路径搜索来得到子笔画序列;最后通过进行 B 样条插值来提取细化后的笔画。对比实验结果表明,该算法不仅能够有效地用于模糊区域检测和笔画提取,而且能够避免细化结果在模糊区域内的形状畸变。

**关键词** 笔画提取 模糊区域 细化算法 书写序列

中图法分类号: TP391.43 文献标识码: A 文章编号: 1006-8961(2009)11-2341-08

## A Method for Handwritten Chinese Stroke Extraction Based on Ambiguous-zone Detection

CAO Zhong-sheng, SU Zhe-wen, WANG Yuan-zhen, XIONG Peng

(School of Computer Science and Technology, Huazhong University of Science & Technology, Wuhan 430074)

**Abstract** To effectively identify and interpret ambiguous zones in handwritten Chinese image, a new method for handwritten stroke extraction based on ambiguous-zone detection is proposed. In the method, a candidate set of fork points are extracted using a thinning algorithm, and the ambiguous zones are detected according to the fork points and the contour beside them. Then, the sub-strokes and ambiguous zones can be modeled with a graph, and a Bayesian classifier is built to analyze the continuity of sub-stroke pairs. Finally, sequences of sub-stroke are achieved by searching paths in the graph, and thinned strokes can be retrieved by B-spline interpolation. Experimental results show that the proposed method is effective and accurate for both ambiguous-zone detection and stroke extraction compared to other methods and reduce the shape distortions in ambiguous zones to an acceptable level.

**Keywords** stroke extraction, ambiguous zone, thinning algorithm, writing sequence

## 1 引言

由于汉字是一种结构性非常强的文字,因此结构分析是手写文本图像处理的重要方法,而笔画作为汉字结构的基本组成元素,是字符图像的重要特征。基于不同的应用环境,许多文献对笔画

有各自不同的定义,文献[1]就给出了4种笔画定义,并对这些定义进行了比较。基于后续的书写序列的恢复工作考虑,本文所讨论的笔画是指在正常的书写过程中,从落笔到抬笔之间的书写轨迹的像素点序列,其接近于文献[1]中对自然笔画的定义。

目前,笔画提取的方法主要有细化法<sup>[2-4]</sup>、轮

基金项目:国家高技术研究发展计划(863)项目(2006AA01Z430)

收稿日期:2008-05-19;改回日期:2008-09-01

第一作者简介:曹忠升(1965~),男,副教授。1993年于华中科技大学控制科学与工程系获得生物医学工程专业博士学位。主要研究方向为多媒体数据库、空间数据库、移动数据库、数据仓库和数据开采。E-mail:caozhongsheng@163.com

廓信息法<sup>[5-9]</sup>和段化法<sup>[10-11]</sup>等。其中细化法容易造成伪段和形状扭曲,尽管在一些文献中也提出了一些去除伪段的修正方法,如基于平均笔画宽度<sup>[3]</sup>和最大环准则<sup>[4]</sup>的修正方法,但修正效果并不理想,甚至还会引入其他的畸变;轮廓信息法则是利用轮廓信息,将字符分割成线状区域和相交区域两部分,但由于不能有效地确定相交区域,因此容易造成笔画的变形和丢失;段化法是通过相邻行和列连通段的关系来提取笔画,该方法比较适用于印刷体汉字,但对手写汉字,则精确率不是很高。此外还有小波分析法<sup>[12]</sup>、形态分析法<sup>[13]</sup>等。

不能有效地识别和解析笔画相交和相接区域是造成上述几种笔画提取算法误差的一个重要原因,因为在这些区域,书写过程中原有的计程信息丢失,这些区域称为模糊区域(ambiguous zone or interference area)<sup>[5]</sup>。目前,模糊区域的检测主要有角点检测法<sup>[9,13]</sup>、窗口扫描法<sup>[5-7]</sup>、轮廓截面法<sup>[14]</sup>、方向滤波法<sup>[12]</sup>和段化法<sup>[10-11]</sup>等。其中角点检测法是通过检测轮廓上曲率不连续的点(角点)来定位模糊区域,因此受到曲率估算算法性能的影响。通常采用的曲率检测算法,对较长轮廓线上的点具有较高的检测精度,但对较短线上的点则误差较大。窗口扫描法是先对图像进行顺序的窗口扫描,然后由窗口内背景区域的数量来判断模糊区域的类型,其不仅时间复杂度较高,窗口的大小不易确定,而且模糊区域的最终定位还是需要依赖轮廓上的角点。轮廓截面法容易受到轮廓噪音的影响,而方向滤波法和段化法仅局限于笔画方向特征明显的汉字的检测。

针对这个问题,本文首先提出了一种新的模糊区域检测算法,该算法不依赖于轮廓上的角点,其不仅避免由于窗口扫描造成的时间开销,而且具有较强的抗噪性和较高的检测精度。经过模糊区域检测后,字符就被分割为子笔画和模糊区域,再利用轮廓跟踪的方法即可得到子笔画书写轨迹的型值点,同时用图模型对字符进行建模;然后通过节点分析将笔画提取的问题转换为在图中搜索所有满足一定条件的笔画路径的问题;最后对子笔画序列的所有型值点进行 B 样条插值,即得到最终的笔画,算法流程如图 1 所示。

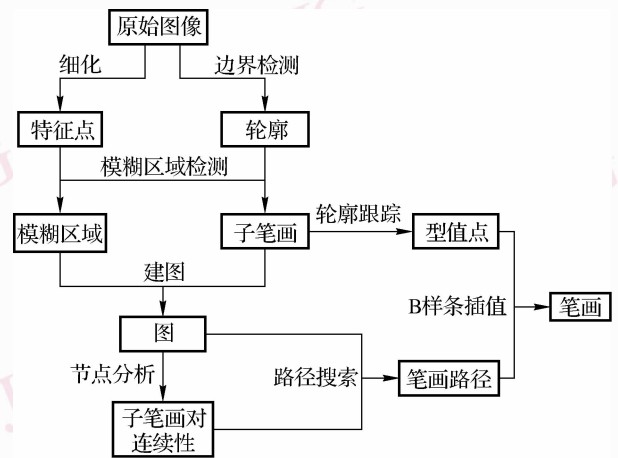


图 1 算法流程图

Fig. 1 The flow chart of our algorithm

## 2 模糊区域

本文所处理的字符图像(图 2(a))是经过去噪和二值化处理后得到的二值图像, $p(i, j)$ 表示原始或细化<sup>[15]</sup>后的图像中第  $i$  行、第  $j$  列上的像素值,分别用 1 和 0 来表示黑色和白色像素, $p_k (0 \leq k \leq 7)$ 表示点  $p$  按顺时针或逆时针方向排列的 8 邻域内的点。

**定义 1** 如果  $p(i, j) = 1$  的 4 邻域中存在值为 0 的像素点,则  $p(i, j)$  为轮廓点,否则为非轮廓点。将原始图像中的轮廓点集合记为  $C$ 。

### 2.1 fork 点

fork 点是指原始图像经过细化以后,由 3 个或 3 个以上的骨架分支相接的点,它唯一地对应了图像上的一个模糊区域。fork 点是一个抽象的概念,由于目前细化算法的缺陷,它通常对应于骨架上一个或多个实际的像素点,这些像素点则称为 fork 候选点。基于细化的笔画提取算法<sup>[2,4]</sup>通常将 fork 候选点和终止点作为骨架的特征点,如图 2(b)所示(其中空心圆表示 fork 候选点,实心圆表示终止点),由文献[3]可知, fork 候选点集合  $S_f$ (下角 f 代表 fork)和终止点集合  $S_e$ (下角 e 代表 end)的定义如下:

$$S_f = \{p \mid (N_c(p) \geq 3) \text{ 或 } (N_b(p) \geq 4)\} \quad (1)$$

$$S_e = \{p \mid N_b(p) = 1\} \quad (2)$$

其中, $N_c(p)$ (下角 c 代表 cross)为像素点  $p$  在其 8 邻域内的黑白像素的交叠数,即

$$N_c(p) = \frac{1}{2} \sum_{i=0}^7 |p_{i+1} - p_i| (p_8 = p_0) \quad (3)$$

$N_b(p)$  (下角 b 代表 black) 为像素点  $p$  在其 8 邻域内的黑色像素点数, 即

$$N_b(p) = \sum_0^7 p_i \quad (4)$$

图像的模糊区域经过细化以后, 通常分裂为一个或多个的 fork 候选点, 同时一些笔画的急促转折也可能形成 fork 候选点 (如图 2(c) 所示), 这些不与模糊区域对应的 fork 候选点称为伪 fork 候选点。本文采用双门限<sup>[2]</sup>的方法来检测伪 fork 候选点, 对于任意的 fork 候选点  $p \in S_f$  和终止点  $e \in S_e$ , 且存在骨架分支  $b_{p,e}$  使点  $p$  和终止点  $e$  连通:

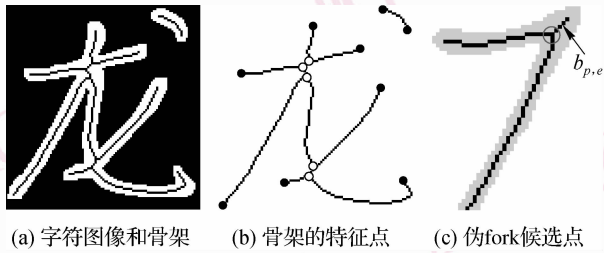


图 2 特征点提取

Fig. 2 Extraction of feature points

若骨架分支  $b_{p,e}$  的长度小于  $T_{b\_min}$  (下角 b 代表 branch), 或者  $b_{p,e}$  的长度介于  $T_{b\_min}$  和  $T_{b\_max}$  之间, 则对于分支  $b_{p,e}$  上的任意一点  $p_i$ ,  $d_c(p_i)$  (下角 C 代表 contour) 即表示点  $p_i$  与距离点  $p_i$  最近的轮廓点之间的距离; 若  $\sum d_c(p_i) > 0.6w$  或  $\max(d_c(p_i)) > w$  则点  $p$  为伪 fork 候选点。其中  $w = 2N_p/L_c$  为平均笔画宽度,  $N_p$  (下角 p 代表 pixel) 为原始字符图像中黑色像素点的个数,  $L_c$  (下角 C 代表 contour) 为原始字符图像轮廓线的长度;  $T_{b\_min}$  是骨架分支的最小门限长度 (本文设为  $0.8w$ ), 而  $T_{b\_max}$  是骨架分支的最大门限长度 (本文设为  $2.5w$ )。

本文通过构造模糊区域集合  $S_a$  (下角 a 代表 ambiguous zone) 来找出 fork 候选点集  $S_f$  与 fork 点集之间的对应关系。对于任意的  $s_z \in S_a$ ,  $s_z$  (下角 z 代表 zone) 为  $S_f$  的非空真子集, 且唯一地对应了一个模糊区域, 使得处于该模糊区域的所有 fork 候选点都包含于  $s_z$ 。

细化通常是指对字符原始图像不断删除区域边界的像素点, 直到满足单像素宽度为止的迭代过程; 而半细化则是经过  $w/2$  次迭代以后的细化结果。

文献[16]利用半细化的方法检测字符图像中的退化区域。本文将该方法用于构造集合  $S_a$ :

(1) 令  $S_1$  表示伪 fork 点集合,  $S_2 = S_f - S_1$ , 则  $S_a$  可作为  $S_2$  的一个划分。

(2) 对于任意的候选 fork 点  $p_0, p_1 \in S_2$ , 若存在骨架分支使点  $p_0$  和点  $p_1$  连通, 且该分支的长度小于  $T_{b\_min}$ , 则  $p_0$  和  $p_1$  属于同一个子集。

(3) 令  $S_3$  表示原始图像半细化后轮廓点的集合,  $S_4$  表示原始图像细化后骨架点的集合,  $S_5 = S_2 \cap (S_4 - S_3)$ 。对于任意的候选 fork 点  $p_0, p_1 \in S_5$ , 若点  $p_0$  和点  $p_1$  在集合  $S_4 - S_3$  中的 8 邻域内连通, 则点  $p_0$  和点  $p_1$  属于同一个子集。

### 2.2 模糊区域检测算法

轮廓跟踪是指通过顺序找出轮廓点来跟踪出边界, 其思想是从轮廓中的任意点出发, 先按一定的方向 (顺时针或逆时针) 来搜索下一个未被访问过的轮廓点; 然后将该点作为当前点, 继续搜索, 直到回到起始点为止。

定义 2 令  $c_i \in C$  表示轮廓中的任意像素点, 如果轮廓跟踪的顺序为  $(\dots, c_{i-n}, \dots, c_{i-1}, c_i, c_{i+1}, \dots, c_{i+m}, \dots)$ , 则  $c_{i+m}$  为  $c_i$  的  $m$  阶前驱, 记为  $pred(c_i, m)$ , 而  $c_{i-n}$  为  $c_i$  的  $n$  阶后继, 记为  $succ(c_i, n)$ ;  $\widehat{c_i c_{i+m}}$  表示从点  $c_i$  到点  $c_{i+m}$  的轮廓曲线, 且该曲线的长度表示为  $L_s(\widehat{c_i c_{i+m}})$  (下角 S 代表 segment)。

对于任意的  $s_z \in S_a$ , 可将  $s_z$  中的点按顺时针或逆时针方向连接起来形成多边形, 令  $p_c$  表示该多边形的质心 (如图 3 所示)。令轮廓点集合  $S_C = \{c \mid d(p_c, c) < T_r, c \in C\}$ , 其中  $d$  为距离函数;  $T_r$  (下

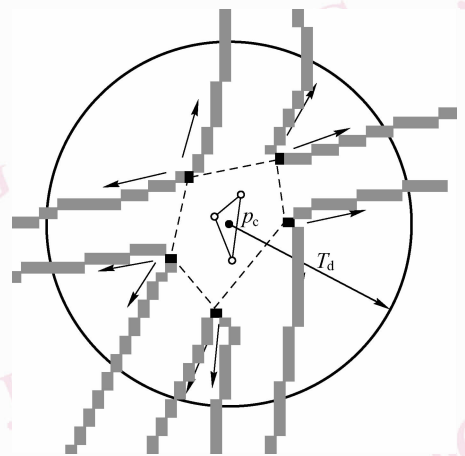


图 3 模糊区域检测

Fig. 3 Detection of ambiguous zone

角  $r$  代表 radius) 是半径的阈值, 以保证整个模糊区域是在以多边形质心  $p_c$  为圆心, 以  $T_r$  为半径的圆的范围内(本文设为  $3w$ )。模糊区域的检测算法步骤如下:

(1) 将模糊区域顶点集合  $S_v$  (下角  $v$  代表 vertex) 置空;

(2) 如果  $d(p_{\min}, p_c) < T_d$ , 那么, 如果直线  $\overline{p_c p_{\min}}$  在笔画内部, 则将点  $p_{\min}$  加入到  $S_v$  中; 否则, 结束; 其中,

$$p_{\min} = \arg \min_{c \in S_c} (d(c, p_c)) \quad (5)$$

$T_d$  (下角  $d$  代表 distance) 为控制模糊区域大小的距离门限值(本文设为  $1.4w$ );

(3) 给定步长门限  $T_{\text{length}}$  (本文设为  $1.5w$ ),

$p_{\min\_for} = \text{pred}(p_{\min}, i)$ ,  $p_{\min\_back} = \text{succ}(p_{\min}, j)$ , 其中  $i$

和  $j$  分别是满足  $L_s(\overline{p_{\min} p_{\min\_for}}) < T_{\text{length}}$  和

$L_s(\overline{p_{\min\_back} p_{\min}}) < T_{\text{length}}$  的最大整数, 则  $S_L(p_{\min}) =$

$\{p \mid p \in \overline{p_{\min\_back} p_{\min\_for}}\}$ :

$$S_c = S_c - S_L(p_{\min}) \quad (6)$$

(4) 如果  $S_c$  为空, 则结束; 否则, 转步骤(2)。

若  $|S_v| > 2$ , 则将  $S_v$  中的点按顺时针或逆时针方向连接起来, 从而构成多边形, 作为与  $s_z$  对应的模糊区域。

若  $|S_v| \leq 2$ , 则将  $s_z$  作为未被检测到的伪 fork 候选点集。

### 3 笔画提取

子笔画是指被模糊区域分隔的笔画段, 笔画是由子笔画和模糊区域串联而成, 笔画在模糊区域上的计程信息主要是利用笔画两侧的子笔画来恢复。而利用图模型对汉字进行建模, 即可将笔画提取的问题转换为在图中找出所有满足一定条件的连通路径的问题, 以便通过 B 样条插值来得到笔画书写轨迹的像素点序列。

#### 3.1 建模

本文采用一个无向图  $G = (V, E)$  来对字符图像进行建模, 其中  $V$  和  $E$  分别是图  $G$  中的节点和边的集合, 而字符中的每个模糊区域和子笔画唯一地对应了  $V$  中的一个节点。对于任意的节点  $v_i \in V$ , 且节点  $v_i$  对应于子笔画  $R_i$ , 而节点  $v_j \in V$  对应于一个与子笔画  $R_i$  相连的模糊区域, 则有  $(v_i, v_j) \in E$ 。若子笔画  $R_i$  的两端均与该模糊区域相连, 为了避免在  $G$  中出现平行边的情况, 则必须对节点集合  $V$  增加一个虚拟节点  $v_n$ , 该虚拟节点对应该子笔画  $R_i$ , 并

且有  $(v_n, v_j) \in E$ ,  $(v_n, v_i) \in E$ 。

图  $G$  中度大于 2 的节点对应了模糊区域, 而度为 2, 1 和 0 的节点分别对应了 3 种不同类型的子笔画, 分别为子笔画两端与模糊区域相连, 子笔画一端与模糊区域相连, 子笔画不与任何模糊区域相连。

#### 3.2 型值点提取

本文采用对子笔画轮廓两侧的对对应点进行跟踪的方法来得到子笔画书写轨迹的型值点, 该方法分为以下 3 个步骤:

(1) 起始点定位 对于第 1, 2 种类型的子笔画, 则将对应的模糊区域(若子笔画两端均与模糊区域相连, 则任取一端的模糊区域)的顶点作为起始点(如图 4 所示)。

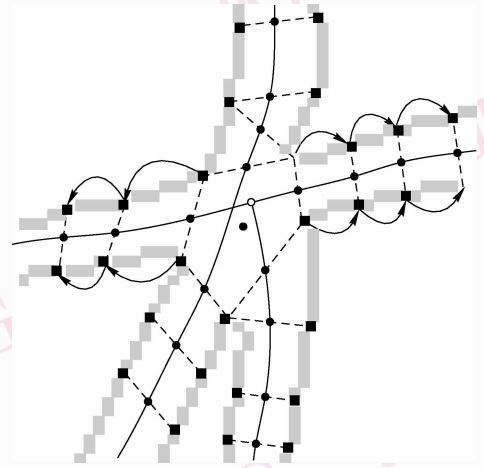


图 4 子笔画型值点提取

Fig. 4 Extraction of data point from sub-stroke

对于第 3 种类型的子笔画, 可先对其轮廓进行多边形拟合, 然后将最长边的中点, 以及该点的法线与轮廓另一侧的交点作为起始点。

(2) 指示点移动 不妨设  $p_0$  和  $p_1$  为当前指示点, 且移动的方向分别为顺轮廓跟踪方向和逆轮廓跟踪方向, 令  $d_0 = d(p_0, p_1)$ ,  $d_1 = d(\hat{p}_0, \hat{p}_1)$ ,  $d_2 = d(\hat{p}_0, p_1)$ , 且  $d_3 = d(p_0, \hat{p}_1)$ , 其中,  $\hat{p}_0 = \text{pred}(p_0, i_0)$ ,  $\hat{p}_1 = \text{succ}(p_1, i_1)$ , 而  $i_0$  和  $i_1$  分别是满足  $L_s(\overline{p_0 \hat{p}_0}) < d_0$  和  $L_s(\overline{p_1 \hat{p}_1}) < d_0$  的最大整数:

① 若  $d_1 \leq d_2$  且  $d_1 \leq d_3$ , 则  $p_0 = \hat{p}_0$ ,  $p_1 = \hat{p}_1$ ; 否则,

② 若  $d_2 \leq d_1$  且  $d_2 \leq d_3$ , 则  $p_0 = \hat{p}_0$ ; 否则,

③ 若  $d_3 \leq d_1$  且  $d_3 \leq d_2$ , 则  $p_1 = \hat{p}_1$ 。

(3) 终止条件 若存在点  $p \in \overline{p_i \hat{p}_i}$  ( $i = 0, 1$ ), 且点  $p$  为其他模糊区域顶点或已跟踪的点, 则指示点

停止移动。

在轮廓跟踪的过程中,需将对应指示点的中心点序列记录下来,作为子笔画书写轨迹的型值点。

### 3.3 子笔画的连续性

**定义 3** 对于任意的子笔画节点  $v_1, v_2 \in V$ ,若存在模糊区域节点  $v_a \in V$ ,使  $(v_1, v_a) \in E, (v_2, v_a) \in E$ ,且  $v_1$  和  $v_2$  对应的子笔画属于同一个笔画,则子笔画节点  $v_1$  和  $v_2$  在模糊区域节点  $v_a$  上连续,记为  $C(v_1, v_a, v_2)$  或  $C(v_2, v_a, v_1)$ 。

判断与同一模糊区域相连的子笔画对是否属于同一个笔画的子笔画连续性分析,这显然可以看作是一个分类的问题。设  $(v_i, v_j)$  是一个与模糊区域节点  $v_a$  相连的节点对,分别对应子笔画  $R_i$  和  $R_j$ 。本文通过构造贝叶斯分类器来判断节点  $v_i$  和节点  $v_j$  是否连续。设  $R_i$  上(与模糊区域相连的一侧开始)的型值点序列为  $(p_{i,0}, p_{i,1}, \dots, p_{i,m})$ ,其中  $m$  为子笔画  $R_i$  上的型值点数,令

$$\bar{\alpha}_{i,j} = \frac{1}{j} \sum_{k=1}^j \theta(p_{i,k-1}, p_{i,k}) \quad (1 \leq j \leq m) \quad (7)$$

$$E(\varepsilon, \eta) = \min(|\varepsilon - \eta|, 2\pi - |\varepsilon - \eta|) \quad (8)$$

$$\beta_{i,j} = \theta(p_{j,0}, p_{i,0}) \quad (9)$$

$g(1 \leq g \leq m)$  为满足以下 3 个条件的最大值:

$$(1) \sum_{k=1}^g d(p_{i,k-1}, p_{i,k}) < T_s$$

$$(2) E(\theta(p_{i,k-1}, p_{i,k}), \bar{\alpha}_{i,k}) < T_{\alpha_1}$$

$$(3) E(\theta(p_{i,k-1}, p_{i,k}), \bar{\alpha}_{i,1}) < T_{\alpha_2} \quad (1 \leq k \leq g)$$

其中,  $\theta(p_{i,k-1}, p_{i,k})$  是从点  $p_{i,k-1}$  到点  $p_{i,k}$  的方向向量;  $T_s$  为有效支持域的门限长度(本文设为  $2w$ );  $T_{\alpha_1}$  和  $T_{\alpha_2}$  为支持域的门限角度偏移(如图 5 所示)。本

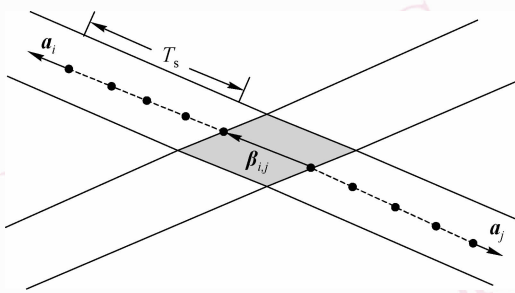


图 5 子笔画的连续性分析

Fig. 5 Analysis of the continuity of sub-stroke

文用  $\alpha_i$  来表示子笔画  $R_i$ (与模糊区域相连的一侧)的方向向量:

$$\alpha_i = \bar{\alpha}_{i,g} \quad (10)$$

分类器采用 4 维特征向量  $X = \{x_0, x_1, x_2, x_3\}$ ,其中  $x_0, x_1, x_2$  和  $x_3$  分别是对以下 4 个特征属性的度量:

(1) 子笔画  $R_i$  和  $R_j$  之间的角度偏差

$$\gamma_{i,j} = |\alpha_i - \beta_{i,j}| + |\pi - |\alpha_j - \beta_{i,j}|| \quad (11)$$

(2) 节点  $v_a$  的度;

(3) 与对应于模糊区域节点  $v_a$  的模糊区域相连的子笔画中,除子笔画  $R_j$  外,和子笔画  $R_i$  的最小角度偏差

$$\gamma_{\min}(i, j) = \min_{(v_k, v_a) \in E, v_k \neq v_i, v_j} \gamma_{i,k} \quad (12)$$

(4) 与对应于模糊区域节点  $v_a$  的模糊区域相连的子笔画中,除子笔画  $R_i$  外,和子笔画  $R_j$  的最小角度偏差  $\gamma_{\min}(j, i)$ ,而类别属性为  $\{C_0, C_1\}$ ,分别表示节点对  $(v_i, v_j)$  的类别为“连续”和“不连续”。

**定义 4** 对于任意的子笔画节点  $v_i \in V$  和模糊区域节点  $v_a \in V$ ,且  $(v_i, v_a) \in E$ ,若不存在任何的  $v \in V$  使  $C(v_i, v_a, v)$ ,则子笔画节点  $v_i$  在模糊区域节点  $v_a$  上终止,记为  $T(v_i, v_a)$ 。

**定义 5** 对于任意的子笔画节点  $v_m \in V$  和模糊区域节点  $v_a \in V$ ,且  $(v_{\min}, v_a) \in E$ ,若存在  $n(n > 1)$  个子笔画节点  $v_0, v_1, \dots, v_{n-1} \in V$ ,使  $C(v_{\min}, v_a, v_i)$  ( $0 \leq i \leq n-1$ ),则称  $v_{\min}$  在  $v_a$  上  $n$ -描绘。

### 3.4 路径搜索

对于  $G$  中任意一条简单路径  $L = (v_0, v_1, \dots, v_n)$ ,若  $L$  的长度为 1,且  $v_0$  为孤立节点,或满足以下两条约束条件,则  $L$  为一条笔画路径:

(1) 端点约束:  $v_0$  和  $v_n$  的度为 1,或者大于 2,且  $T(v_j, v_i)$  ( $< i, j > = < 0, 1 >$  或  $< n, n-1 >$ );

(2) 非端点约束:  $v_i$  ( $0 < i < n$ ) 的度等于 2,或者大于 2,且  $C(v_{i-1}, v_i, v_{i+1})$ 。

设笔画路径的集合为  $S_{\text{path}}$ ,笔画集合为  $S_{\text{stroke}}$ ,在大多数的情况下,笔画路径和笔画是一一对应的,即  $|S_{\text{path}}| = |S_{\text{stroke}}|$ 。对笔画路径上所有子笔画串联后的型值点进行 B 样条插值,就可以得到笔画的书写轨迹。B 样条插值能够理想地描述笔画的位置,而不会造成重要的特征丢失,并且这种表达对轮廓上的噪音并不敏感。但是如果存在  $n$ -描绘于子笔画(如图 6 所示),则可能会出现笔画路径和笔画不对应的情况,即  $|S_{\text{path}}| > |S_{\text{stroke}}|$ ,为此,本文采用以下两条准则来对此进行修正。

**准则 1** 设简单路径  $L = (v_0, v_1, \dots, v_n)$  ( $n \geq 2$ )

满足:(1) $v_0$ 和 $v_n$ 为模糊区域节点;(2) $v_1$ 和 $v_{n-1}$ 分别在 $v_0$ 和 $v_n$ 上 $n$ -描绘;(3)节点 $v_i(1 \leq i \leq n-1)$ 满足笔画路径的非端点约束;(4)节点 $v_i(1 < i < n-1)$ 中不存在 $n$ -描绘子笔画(如图6(a)所示)。若 $L_0, L_1, \dots, L_m$ 是以 $L$ 为子路径的笔画路径,则不妨将 $L_i(0 \leq i \leq m)$ 可以表示为 $(f_i, L, b_i)$ 。令 $F = \{f_i\}, B = \{b_i\}, r = |F|, q = |B|$ ,则 $m+1 = r \times q$ 。根据书写经验可知,其中包含的笔画数量应为 $\max(r, q) < m+1$ 。不妨设 $r > q$ ,由覆盖所有子笔画的约束条件可知,共能产生 $N_{cp}$ 种笔画路径组合,即

$$N_{cp} = \sum_{k=0}^q (-1)^k \binom{q}{k} (q-k)^r \quad (13)$$

对这些笔画路径组合,先利用文献[2]中的SLALOM方法计算每种组合的光滑度总和,然后从中选取最光滑的一个组合。

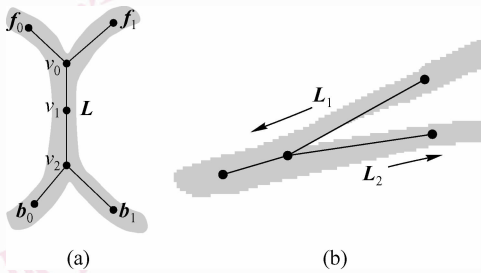


图6  $n$ -描绘子笔画  
Fig.6  $n$ -traced sub-stroke

**准则 2** V型的笔画相接,经常会形成路径中的第1个或最后一个节点为双描绘子笔画(如图6(b)所示),其中箭头表示笔画的书写顺序。由汉字的书写习惯可知, $L_1$ 和 $L_2$ 在书写过程中通常是连续的,且不存在抬笔,所以根据笔画的定义, $L_1$ 和 $L_2$ 应该属于同一条笔画。设 $L_1 = (v_0^{(1)}, v_1^{(1)}, \dots, v_n^{(1)})$ 和 $L_2 = (v_0^{(2)}, v_1^{(2)}, \dots, v_m^{(2)})$ 是两条相异的笔画路径,若 $e_0 = e_1 (\langle e_0, e_1 \in \{ \langle v_0^{(1)}, v_0^{(2)} \rangle, \langle v_0^{(1)}, v_m^{(2)} \rangle, \langle v_n^{(1)}, v_0^{(2)} \rangle, \langle v_n^{(1)}, v_m^{(2)} \rangle \} )$ ,且 $e_0$ 的度为1,则将 $L_1$ 和 $L_2$ 进行B样条插值,然后将其书写轨迹按书写顺序或逆书写顺序进行合并。

### 3.5 修正

对于在模糊区域上终止的子笔画,如果只对指示点的中心点进行B样条插值的话,则会造成相交笔画骨架的空隙,从而破坏笔画的连通性。为了保证笔画的连通关系,本文通常会增加一个型值点。

设 $R_i$ 为终止的子笔画, $Z_a$ 为模糊区域,分别对

应图G中的节点 $v_i$ 和 $v_a$ ,且有 $T(v_i, v_a)$ ,则可将 $R_i$ 上与模糊区域 $Z_a$ 相连一侧的 $g$ 个型值点作为样本进行曲线拟合。为了简单起见,本文采用的是以下三次多项式进行拟合:

$$Y = \alpha + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 \quad (14)$$

通过最小二乘法就可以得到参数 $\alpha, \beta_1, \beta_2$ 和 $\beta_3$ 。设该曲线与笔画 $R_c$ 的交点为 $p_i$ ,若点 $p_i$ 在模糊区域内部,则将 $p_i$ 作为 $R_i$ 新加入的型值点(如图7(a)所示);否则将模糊区域的质心 $p_c$ 作为 $R_i$ 新加入的型值点(如图7(b)所示)。

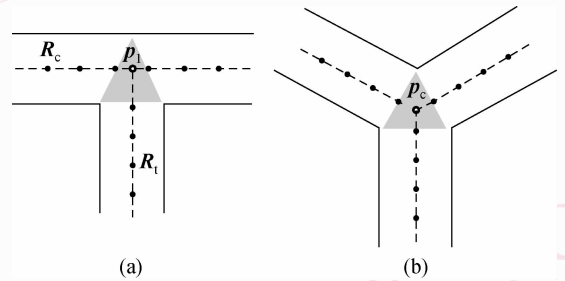


图7 插值修正  
Fig.7 Correction in B-interpolation

## 4 实验结果及分析

本文从HIT-MW中文手写字体<sup>[16]</sup>中随机选取了50幅来自不同书写者的手写汉字文档图像,其中包含了7631个手写汉字。在实验的预处理阶段,将其切分为字符图像,并以大小为 $100 \times 100$ 的Bmp图像进行存储,作为本文的实验样本。实验环境是P4 1.8 GHz,内存512 M的微机。实验分为模糊区域检测和笔画提取两个部分。

### 4.1 模糊区域检测

模糊区域检测实验旨在测试对字符图像中模糊区域的检测效果,测试对象采用分别来自20个不同书写者的200幅字符图像,其中包含了模糊区域1038个。为了验证其检测性能,本文还将本文算法与文献[5]中基于窗口扫描的方法进行了对比,实验结果如表1所示。

表1 模糊区域检测性能比较

Tab.1 Comparison of ambiguous-zone detection

	正确率 (%)	漏检率 (%)	误检率 (%)	处理速度 (字/s)
本文算法	98.1	1.9	3.3	10.4
窗口扫描	85.3	14.7	16.8	1.14

造成本文提出的模糊区域检测算法漏检的原因主要有:(1)平整的“L”形笔画相接,细化后并不会产生 fork 候选点(如图 8(a)所示);(2)检测伪 fork 候选点造成的误差。而造成误检的原因则主要是由于未能检测到的伪 fork 候选点造成的(如图 8(b)所示),由笔画急促转折形成的伪段未能被双门限方法所检测。这种情况可以看作是一种特殊的 V 型笔画相接,由于其可以通过 3.4 节中的准则 2 来消除,因此不会对最终的笔画提取造成太大的影响。

基于细化的算法,但提取笔画的准确率却高于基于细化的算法。并且从图 9 可见,本文算法提取笔画的精度、连续性和抗噪声性都高于基于细化的算法。

表 2 笔画提取性能比较

Tab. 2 Comparison of stroke extraction

	处理速度(字/s)	准确率(%)
本文算法	4.42	94.32
基于细化算法	5.74	89.64

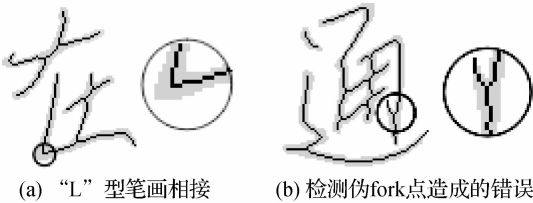


图 8 模糊区域检测错误的样本示例

Fig. 8 Experiment results failed to detect ambiguous zones

## 5 结 论

笔画计程信息是由两侧的轮廓来反映的,但由于受到其他笔画的干扰,从而造成笔画中的某段一侧或两侧轮廓信息丢失,形成模糊区域,不能有效地识别和解析模糊区域是造成目前主要的笔画提取算法提取笔画产生误差的一个重要原因。本文从模糊区域检测的角度上,提出了一种手写汉字笔画提取方法。该方法是先利用细化结果提取特征点,同时根据 fork 候选点及其附近的轮廓信息来检测模糊区域;然后根据笔画书写的连贯性,通过构造贝叶斯分类器的方法来分析与模糊区域相连的子笔画对的连通性;最后,利用 B 样条插值来得到构成笔画书写轨迹的像素点序列。实验表明,利用本文算法提取的笔画具有较高的精度和有效性。

## 4.2 笔画提取

提取笔画时,先从实验样本中选取了 400 幅图像,并标记其中的子笔画对的连续性(共计 9 023 对子笔画)后,再对子笔画连续性分析中的贝叶斯分类器进行训练,然后从其余的实验样本中随机选取 500 幅字符图像进行测试。为了验证本文算法进行笔画提取的效果,将本文的算法与文献[2]中基于细化的笔画提取算法进行了比较,实验的结果示例如图 9 所示。

## 参考文献 (References)

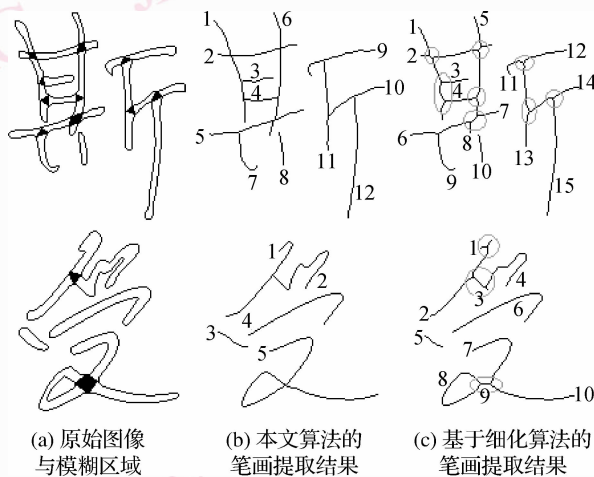


图 9 笔画提取结果比较

Fig. 9 Comparison of extracted stroke

- Liu Cheng-lin, Kim In-jung, Kim J H. Model-based stroke extraction and matching for handwritten Chinese character recognition [J]. Pattern Recognition, 2001, 34(12):2339-2352.
- Qiao Yu, Yasuhara Makoto. Recovering dynamic information from static handwritten images [A]. In: Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition[C], Tokyo, Japan, 2004: 118-123.
- Qiao Yu, Nishiara Mikihiko, Yasuhara Makoto. A framework toward restoration of writing order from single-stroked handwriting image[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, 28(11): 1724-1737.
- Liu Ke, Huang Y S, Suen C Y. Identification of fork points on the skeletons of handwritten Chinese characters[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1999, 21(10): 1095-1100.
- Plamondon Rejean, Privitera C M. The segmentation of cursive handwriting: An approach based on off-line recovery of the motor-temporal information[J]. IEEE Transactions on Image Processing, 1999, 8(1):80-91.

由表 2 可知,虽然本文算法的处理速度略低于

- 6 Chouinard C, Plamondon R. Thinning and segmenting handwritten characters by line following[J]. *Machine Vision and Applications*, 1992, **5**(3): 185-197.
- 7 Baati A E, Charfi M, Alimi A M, *et al.* Recovery of temporal information from off-line arabic handwritten[A]. In: *Proceedings of the 3rd ACS/IEEE International Conference on Computer Systems and Applications*[C], Cairo, Egypt, 2005: 127-132.
- 8 L'Homer E. Extraction of strokes in handwritten character[J]. *Pattern Recognition*, 2000, **33**(7): 1147-1160.
- 9 Lee Chungnan, Wu Bohom. A Chinese-character-stroke-extraction algorithm based on contour information[J]. *Pattern Recognition*, 1998, **31**(6): 651-663.
- 10 Liu Xia-bi, Jia Yun-de. An algorithm of line-segment extraction and thinning for character images[J]. *Journal of Image and Graphics*, 2005, **10**(1): 48-53. [刘峡壁, 贾云得. 一种字符图像线段提取及细化算法[J]. *中国图象图形学报*, 2005, **10**(1): 48-53.]
- 11 Sun Xing-ming, Yang Mao-jiang, Liu Guo-huan, *et al.* A totally knowledge-based stroke extraction method for Chinese characters[J]. *Journal of Computer Research and Development*, 2000, **37**(5): 543-550. [孙星明, 杨茂江, 刘国华等. 完全基于结构知识的汉字笔画抽取方法[J]. *计算机研究与发展*, 2000, **37**(5): 543-550.]
- 12 Su Yih-ming, Wang Jhing-fa. A novel stroke extraction method for Chinese characters using Gabor filters[J]. *Pattern Recognition*, 2003, **36**(3): 635-647.
- 13 Hu Chang-yuan, Zhang Fu-yan. A method for decomposing complex shape into strokes[J]. *Journal of Computer-Aided Design and Computer Graphics*, 1996, **8**(6): 408-414. [胡长原, 张福炎. 复杂形状区域的一种笔画分解算法及其应用[J]. *计算机辅助设计与图形学学报*, 1996, **8**(6): 408-414.]
- 14 Wang Jian, Zhang Shu-sheng, Wang Guang. An joint-point-based skeletonization algorithm of line images[J]. *Journal of Computer Research and Development*. 1999, **36**(6): 725-731. [王健, 张树生, 王广. 基于节点的线状图骨架提取算法研究[J]. *计算机研究与发展*. 1999, **36**(6): 725-731.]
- 15 Shaikh N A, Shaikh Z A. A generalized thinning algorithm for cursive and non-cursive language scripts[A]. In: *Proceedings of the 9th International Multitopic Conference* [C], Karachi, Pakistan, 2005: 1-5.
- 16 Su Tong-hua, Zhang Tian-wen, Guan De-jun. Corpus-based HIT\_MW database for offline recognition of general-purpose Chinese handwritten text[J]. *International Journal of Document Analysis and Recognition*, 2007, **10**(1): 27-38.