ISSN 1000-9825 Journal of Software 软件学报 2000,11(10):1397~1404

基于印刷字符模糊结构分析的字符预分类方法*

卢达[1] 谢铭培[2] 浦炜[1]

[1](常熟高等专科学校 常熟 215500)

[2](复旦大学计算机科学系 上海 200433)

E-mail: ld47@csgz.edu.cn

摘要 提出了一种用于字符预分类的模糊逻辑分析法.对文本字符作印刷结构分析,给出了一个带有容差分析的文本行字符基线精确测定算法,其他有效参考线则是通过聚类分析而获得.模糊逻辑用于确定各字符类的隶属值以保证字符的正确预分类.实验结果表明,这种模糊印刷字符预分类法在SUN 4/490工作站上每秒可有效地处理10^4以上字符,并对不同大小的字符和不同字体的处理结果令人满意.

关键词 字符预分类,印刷归类,基线测定,模糊逻辑,模糊分类.

中图法分类号 TP391

自80年代以来,随着计算机所需处理的文件按指数规律增长和计算机系统功能的不断增强,计算机文本处理得到迅速发展.我们采用对字符预分类方法的目的,是想通过字符印刷结构分析对字符在识别之前进行预处理,以减小字符识别范围,提高字符识别率,进而提高整个文本处理系统的性能.

如图1所示,一行文本行可分解为上、中、下3个区域,它们分别由顶线、上基线、基线、底线所限定.其中,中区是文本行的主要部分,其高度为其他区域的一倍,并可由中线作进一步分离.由于基线出现于每一文本行,所以,基线检测定位是必不可少的.此外,基线用于文本行倾斜角及文本行插入空间的确定方面比用传统的Hough变换和Fouier变换方法计算效率要高得多[1,2].我们的基线测定算法基于一行文本比基于单一字符更可靠、更有效[3],而其他有效参考线则由聚类算法求得[4].容差分析考虑了计算机文本处理过程中不可避免的噪声和失真.为保证算法的灵活性和鲁棒性,模糊逻辑方法用于确定模棱两可的字符对各个印刷字符类的隶属关系[5,6],而线性变换函数的采用和边界条件的推导则保证了其连续性.在下面几节中,首先描述字符印刷结构,接着提出基线检测算法和容差分析,然后给出模糊印刷字符归类方法,最后是实验结果和有关结论.

①上区域,②中区域,③下区域,④顶线,⑤上基线,⑥中线,⑦基线,⑧底线.

Fig.1 Typographical structure of a text line

图1 一行文本的印刷结构

1 字符印刷结构分析

根据字符截取文本行的各区域,可分为基本类字符和辅助类字符[3].

* 本文研究得到国家自然科学基金(No.7870012)和江苏省教委留学回国人员科研基金(No.1997-15-51)资助.作者卢达,1947年生,副教授,主要研究领域为模式识别,图像处理.谢铭培,1938年生,教授,主要研究领域为人工智能,管理信息处理,自动控制,智能仪器.浦炜,1973年生,助教,主要研究领域为图像处理,模式识别.

本文通讯联系人,浦炜,常熟 215500,常熟高等专科学校物理系

本文1998-12-25收到原稿,1999-07-12收到修改稿

© 中国科学院软件研究所 http://www.jos.org.cn

(1) 基本类:该类由包括拉丁字母等在内的绝大多数大字符组成,又可分为:

(a) 上行字符:字符截取文本行的整个上区和中区,如"A","B"和"d".

(b) 下行字符:字符截取文本行的整个下区和中区,如"g","q","y"和"p".

(c) 中行字符:字符仅位于文本行的整个中区,如"e","c","a","m"和"x".

(d) 全区域字符:字符跨越文本行的所有3个区,如"j","(",")","{"和"f".

(2) 辅助类:

(e) 下标字符:字符位于基线附近,如".",",","_"等.

(f) 上标字符:字符位于上基线附近,如"'","~","^"等.

(g) 中行内字符:字符位于中线附近,部分截取中区,如"—".

我们进行字符印刷结构分析的目的之一是要将文本中各个字符正确地归入上述7个字符类.

2 基线检测算法

文本行基线检测是字符印刷结构分析的关键.实际基线定位、检测有效算法描述如下:

(1) 设文本行 T 由 n 个在 T 中自左至右顺序排列的字符单元(ch_i)组成,即 $T=\{ch_1,ch_2,\ldots,ch_n\}$,设 $P=\{p_1,p_2,\ldots,p_n\}$ 表示一行文本行中字符 ch_i 边框底边中点 $p_i=(x_i,y_i)$ 的集合.由于文本中大多数字符为上行字符和中行字符,它们都以基线为基准,所以 P 为寻找实际基线的基础.

(2) 设线段 $\overline{p_ip_{i+1}}$ 的斜率 $y'_i=\frac{y_{i+1}-y_i}{x_{i+1}-x_i}$ 的集合 $Y'=\{y'_1,y'_2,\ldots,y'_{n-1}\}$.通常斜率多接近于0,也就是说,相邻字符 ch_i 和 ch_{i+1} 多以基线或底线为基准.

(3) 设 Y'_{mp} 为用聚类分析并由集合 Y' 中获得的最基本的斜率集合,即

$$Y'_{mp}=\{y'_i\mid|y'_i-y'_{mp}|\leqslant\varepsilon,\ i=1,2,\ldots,n-1\},\tag{1}$$

其中 y'_{mp} 为基本斜率,ε 为聚类系数.

整条基线的初始斜率近似值 m_{appr} 由式(2)求得:

$$m_{appr}=\frac{\sum\limits_{y'_i\in Y'_{mp}}y'_i\Delta x_i}{\sum\limits_{y'_i\in Y'_{mp}}\Delta x_i},\tag{2}$$

其中 $\Delta x_i=x_{i+1}-x_i$.

(4) 设 B_p 为通过 P 中各点的斜率为 m_{appr} 的直线方程 $y=m_{appr}x+b$ 在 Y 轴上截距的集合.即

$$B_p=\{b_i\mid b_i=y_i-m_{appr}x_i,\ i=1,2,\ldots,n\},\tag{3}$$

其中 b_i 是通过 p_i 点斜率为 m_{appr} 的直线方程的截距.

由于基线斜率近似等于 m_{appr},以基线为基准的字符的边框底边中点将共线且考虑以 b 聚类,一些分布稀疏的其他小群组字符则以基线为校准方向.类似于 Y'_{mp},可得到最基本的截距集合:

$$B_{mp}=\{b_i\mid|b_i-b_{mp}|\leqslant\delta\},\tag{4}$$

其中 b_{mp} 为基本截距,δ 为半容差值.

(5) 设基线为 P_{bl},则

$$P_{bl}=\{p_i\mid b_i\in B_{mp}\}.\tag{5}$$

对 P_{bl} 作线性回归可得精确表示基线的方程 $y=m_{bl}x+b_{bl}$[7],其中

$$m_{bl}=\frac{\begin{vmatrix}\sum\limits_{p_i\in P_{bl}}x_iy_i & \sum\limits_{p_i\in P_{bl}}x_i\\ \sum\limits_{p_i\in P_{bl}}y_i & N_{P_{bl}}\end{vmatrix}}{\begin{vmatrix}\sum\limits_{p_i\in P_{bl}}x_i^2 & \sum\limits_{p_i\in P_{bl}}x_i\\ \sum\limits_{p_i\in P_{bl}}x_i & N_{P_{bl}}\end{vmatrix}},\quad b_{bl}=\frac{\begin{vmatrix}\sum\limits_{p_i\in P_{bl}}x_i^2 & \sum\limits_{p_i\in P_{bl}}x_iy_i\\ \sum\limits_{p_i\in P_{bl}}x_i & \sum\limits_{p_i\in P_{bl}}y_i\end{vmatrix}}{\begin{vmatrix}\sum\limits_{p_i\in P_{bl}}x_i^2 & \sum\limits_{p_i\in P_{bl}}x_i\\ \sum\limits_{p_i\in P_{bl}}x_i & N_{P_{bl}}\end{vmatrix}}.\tag{6}$$

© 中国科学院软件研究所 http://www.jos.org.cn

式(6)中,$N_{P_{bl}}=\sum_{p_i\in P_{bl}}x_i^0$ 表示 P_{bl} 中元素的数目.

文本行的其他参考线由基线通过聚类分析而得到,这里不作详述.

3 容差分析

字符印刷结构分析是有效参考线在文本行中正确定位的一种统计方法.由于文本处理过程中存在不可避免的噪声,在基线检测算法以及在下节将要描述的字符印刷结构分类中必须考虑容差.如图2所示,设参考线容差为 $\tau=2\delta$(图中用阴影区域表示),且各参考线容差相等.设 H 为包括容差在内的文本行高度,H_l 为基线和底线之间的高度,若 $H_l=4\delta$,则 $H=18\delta$.

Fig. 2 Tolerances of the reference lines

图2 参考线的容差

设 $Q=\{q_1,q_2,\ldots,q_n\}$ 表示一行文本行中字符 ch_i 边框上边的中点 $q_i=(x_i,y_i)$ 的集合,通过 Q 中各点斜率为 m_{appr} 的直线在 Y 轴上截距的集合可表示为

$$B_Q=\{b_i\mid b_i=y_i-m_{appr}x_i,\ (x_i,y)\in Q\}. \tag{7}$$

则文本行高度为

$$H=\frac{1}{\sqrt{1+m_{appr}^2}}(\max\{\omega\mid\omega\in B_p\}-\min\{\omega\mid\omega\in B_Q\}). \tag{8}$$

当 m_{appr} 很小时,

$$H\approx\max\{\omega\mid\omega\in B_p\}-\min\{\omega\mid\omega\in B_Q\}. \tag{9}$$

由此,容差 $\delta=H/18$ 可计算得出,并可用于式(4).

设文本行中两相邻字符的中心距为文本行高度的两倍,φ 为实际基线与 X 轴的夹角,若两相邻字符 ch_i 和 ch_{i+1} 以基线校准,则 p_ip_{i+1} 的方向必在 $\varphi+\theta$ 和 $\varphi-\theta$ 范围之内,如图3所示.

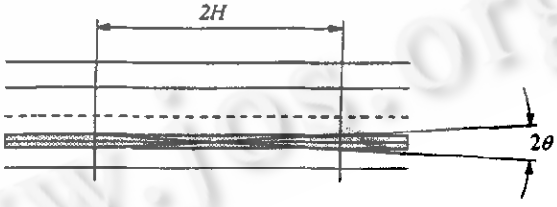


Fig. 3 Tolerance of the slope of the baseline

图3 基线斜率的容差

其中

$$\theta=\tan^{-1}\frac{\tau}{2H}=\tan^{-1}\frac{\delta}{H}, \tag{10}$$

也就是说,其斜率变化范围是 $m_{\varphi+\theta}\sim m_{\varphi-\theta}$.

这里:

$$m_{\varphi+\theta}=\tan(\varphi+\theta)=\frac{\tan\varphi+\tan\theta}{1-\tan\varphi\tan\theta}, \tag{11}$$

$$m_{\varphi-\theta}=\tan(\varphi-\theta)=\frac{\tan\varphi-\tan\theta}{1+\tan\varphi\tan\theta}. \tag{12}$$

当 φ 与 θ 很小时,

© 中国科学院软件研究所 http://www.jos.org.cn

$$m_{\varphi+\theta}\approx\tan\varphi+\tan\theta=m+\varepsilon, \tag{13}$$

$$m_{\varphi-\theta}\approx\tan\varphi-\tan\theta=m-\varepsilon, \tag{14}$$

其中

$$\varepsilon=\tan\theta=\frac{\delta}{H}=\frac{1}{18}. \tag{15}$$

4 模糊印刷分类

如前所述,任一字符由其大小和在文本行中的位置可分为7个印刷类,现分别用符号↑(上标)、↓(下标)、A(上行)、D(下行)、C(中行)、F(全区域)、I(中行内字符)表示,见表1.表中y_q和y_p分别表示字符边框上边、底边中点的坐标.任一确定类字符可用y_q,y_p对表示,例如,(1,7),(1,9),(2,7)和(2,9)分别表示上行、全区域、中行及下行字符.

Table 1
表1

$y_p\backslash y_q$	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9
r_1	↑	—	—	—	—	—	—	—	—
r_2	↑	↑	—	—	—	—	—	—	—
r_3	↑	↑	↑	—	—	—	—	—	—
r_4	↑	↑	↑,(I)	↑,I	—	—	—	—	—
r_5	↑	↑	↑,I	I,(↑)	I	—	—	—	—
r_6	↑,A	↑,A,C	I,C,(↑)	C,I	I,(↓)	I,↓	—	—	—
r_7	A	A,C	C	C,I,(↓)	I,↓	↓,(I)	↓	—	—
r_8	A,F	A,C,F,D	C,D	C,D,↓	↓	↓	↓	↓	—
r_9	F	F,D	D	D,↓	↓	↓	↓	↓	↓

表中圆括号内的类表示为弱类.顶线、上基线、中线、基线及底线的容差带分别用r_1,r_3,r_5,r_7和r_9表示,各容差带之间的区域分别用r_2,r_4,r_6和r_8表示,如图4所示.

Fig. 4 The illustration for the tolerance ranges used in Table 1
图4 用于表1中的容差区间

由于在文本处理过程中噪声和失真的不确定性,字符经检测可能位于文本行中模棱两可的位置,如:y_q在r_1内,而y_p在r_8内,它可能是上行字符或全区域字符,此时,可用隶属函数对该类字符归类.

定义.若一字符的模糊印刷类α为一有序对列,即$\alpha=\{(\Omega,\chi(\Omega))\}$,其中$\Omega\in\{↑,↓,A,D,C,F,I\}$,而$\chi(\Omega)$表示字符对$\Omega$类的隶属程度,$\chi(\Omega)$范围为[0,1].

隶属函数反映字符的大小(y_p-y_q)和字符的位置(y_p+y_q),如图4所示.用$y_0,y_1,\ldots,y_9$在Y轴上表示$r_1,r_2,\ldots,r_9$的界限区间,容差范围$r_i,i=1,2,\ldots,9$归一化并用r'_i表示,归一化的$y'_j,j=p,q$,在r'_i范围内用线性插值法可得$y'_j=\frac{y_j-y_{i-1}}{y_i-y_{i-1}}$.

例:(1,8)类中的隶属函数为

$$\chi_{(1,8)}(A)=1-y'_p=\frac{y_8-y_p}{y_8-y_7},\quad \chi_{(1,8)}(F)=y'_p=\frac{y_p-y_7}{y_8-y_7}, \tag{16}$$

相似地,$\chi_{(2,7)}(\Omega),\chi_{(2,9)}(\Omega)$和$\chi_{(3,8)}(\Omega)$可表示为

$$\chi_{(2,7)}(A)=1-y'_q=\frac{y_2-y_q}{y_2-y_1},\quad \chi_{(2,7)}(C)=y'_q=\frac{y_q-y_1}{y_2-y_1}, \tag{17}$$

© 中国科学院软件研究所 http://www.jos.org.cn

$$\chi_{(2,8)}(D)=y'_q=\frac{y_q-y_1}{y_2-y_1},\quad \chi_{(2,8)}(F)=1-y'_q=\frac{y_2-y_q}{y_2-y_1}, \tag{18}$$

$$\chi_{(3,8)}(D)=y'_p=\frac{y_p-y_7}{y_8-y_7},\quad \chi_{(3,8)}(C)=1-y'_p=\frac{y_8-y_p}{y_8-y_7}. \tag{19}$$

最难确定的(2,8)类的隶属值由边界条件及印刷分类特性确定.在(2,8)类中,可能是上行、下行、全区域或中行字符.

先讨论全区域字符的隶属函数:其边界条件为

$$\chi_{(2,8)}(F)_{y_q=y_1}=y'_p, \tag{20}$$

$$\chi_{(2,8)}(F)_{y_p=y_8}=1-y'_q. \tag{21}$$

全区域字符的隶属函数仅由字符大小决定.设 $s=y'_p-y'_q$,$t=y'_p+y'_q$,隶属函数为线性函数且与字符大小有关,并有如下特性:

$$\frac{\partial\chi_{(2,8)}(F)}{\partial s}=c_1,\quad c_1\text{ 为正常数},\quad \text{当 }\chi_{(2,8)}(F)>0\text{ 时}, \tag{22}$$

$$\frac{\partial\chi_{(2,8)}(F)}{\partial t}=0. \tag{23}$$

由式(20)~(23)可得全区域字符隶属函数:

$$\chi_{(2,8)}(F)=\begin{cases}y'_p-y'_q, & \text{若 } y'_p>y'_q\\ 0, & \text{其他}\end{cases}. \tag{24}$$

同样地,中行字符的隶属函数也仅与字符大小有关:

$$\chi_{(2,8)}(C)=\begin{cases}y'_q-y'_p, & \text{若 } y'_p<y'_q\\ 0, & \text{其他}\end{cases}. \tag{25}$$

上行字符和下行字符隶属值的确定较为复杂,因为隶属值既与字符大小有关,又与字符在文本行中的位置有关.

上行字符隶属函数的边界条件为

$$\chi_{(2,3)}(A)_{y_q=y_1}=1-y'_p, \tag{26}$$

$$\chi_{(2,3)}(A)_{y_p=y_7}=1-y'_q. \tag{27}$$

同时,隶属函数有如下特性:

$$\frac{\partial\chi_{(2,8)}(A)}{\partial t}=c_2, \tag{28}$$

$$\frac{\partial\chi_{(2,8)}(A)}{\partial s}=\begin{cases}c_3 & \text{若 } y'_p<y'_7,\\ -c_3, & \text{其他}\end{cases}. \tag{29}$$

其中 c_2,c_3 为正常数.

由式(26)~(29)可得上行字符隶属函数:

$$\chi_{(2,8)}(A)=\begin{cases}1-y'_q, & \text{若 } y'_p\leqslant y'_q\\ 1-y'_p, & \text{其他}\end{cases}. \tag{30}$$

同样地,下行字符隶属函数为

$$\chi_{(2,8)}(D)=\begin{cases}y'_q, & \text{若 } y'_p\geqslant y'_q\\ y'_p, & \text{其他}\end{cases}. \tag{31}$$

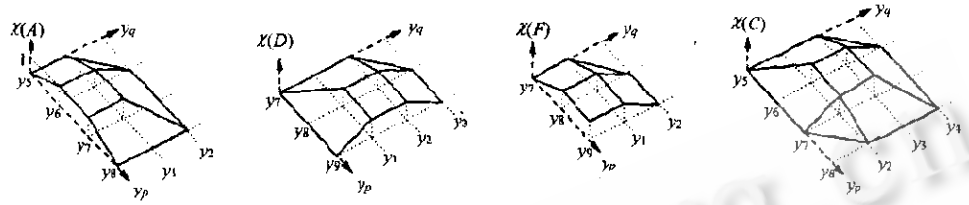
其他不确定字符类的隶属函数可用同样的方法推导出来.各字符印刷结构类的隶属函数如图 5 所示.

5 实验结果

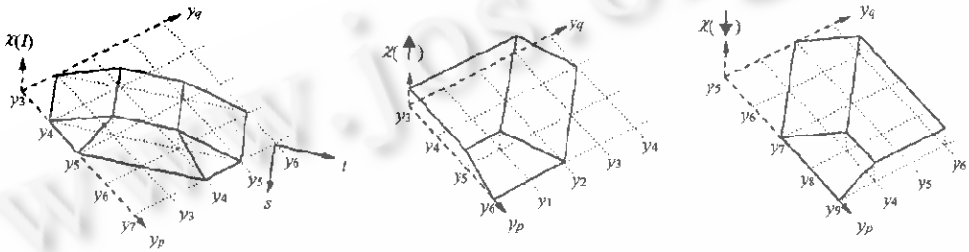
上述算法在 SUN 4/490 工作站 UNIX 系统下用 C 语言完成.文本原始图像由分辨率为 300dpi/inch 扫描仪输入及二值化,并分解为文本块、图形、图片、水平线和垂线.文本块中的各文本行被提取并进行模糊字符印刷结构分析归类.

© 中国科学院软件研究所 http://www.jos.org.cn

实验分为两部分:首先对500行印刷字符大小为5～12点不等(1点大小为$\frac{1}{72}$英寸的文本行和600行字符大小为8点的几种常见印刷字体的文本行作有效参考线检测,结果见表2和表3.



(a) Ascender characters (a) 上行字符 (b) Descender characters (b) 下行字符 (c) Full-Range characters (c) 全区域字符 (d) Center characters (d) 中行字符



(e) Internal characters (e) 中行内字符 (f) Superscript characters (f) 上标字符 (g) Subscript characters (g) 下标字符

Fig. 5 The membership functions of each typographical category

图5 各字符印刷结构类的隶属函数

Table 2

表2

Character size①	Error rate for baseline detection②(%)	The rate of ambiguous character③(%)	The rate of merged character④(%)	Ave. run time (μs/character)⑤
5	2.5	0.50	48.65	85
6	0	0.27	28.23	82
7	0	0.10	17.17	84
8	0	0.21	17.59	84
9	0	0.19	13.39	81
10	0	0.62	5.56	81
11	0	0.68	1.90	79
12	0	0.41	1.45	80

①字符大小,②基线检测错误率,③不确定字符率,④连体字符率,⑤运行时间(μs/字符).

Table 3

表3

Character type①	Character line number②	Error rate for baseline detection③(%)	The rate of ambiguous character④(%)	Ave. run time (μs/character)⑤
Times New Roman	120	0	0.21	84
Times New Roman italic⑥	120	0	0.23	85
Verdana	120	0	0.47	84
Arial	120	0	0.56	84
Courier New	120	0	0.51	82

①字符字体,②字符行数,③基线检测错误率,④不确定字符率,⑤运行时间(μs/字符),⑥Times New Roman 斜体.

© 中国科学院软件研究所 http://www.jos.org.cn

从实验结果来看:(1) 字符大于5点的基线100%被正确地检测出来,而大小为5点的字符的基线检测出现2.5%的错误,其原因是随着字符的减小,连体字符迅速增加,见表2.48.65%的字符相连,并且随着字符的减小,连体字符内包含的字符数增多.在实验中,26.28%的连体字符含有3个以上的字符,造成斜率检测不精确.(2) 当基线被正确定位时,模糊字符印刷结构分类将十分有效,因为99%以上的字符能分在确定的字符类中.根据实验结果,有些特殊字符可能落入不确定字符类内.例如,字符"/","(",")","[","]"被分在(1,8)类,";"被分在(3,8)类,但当应用模糊逻辑时,"/"被确定为上行字符,";"被确定为下行字符,其余字符则被确定为全区域字符.(3) 上述基线检测和字符模糊结构分类方法适用于常见各种印刷字体的文本处理.

实验中还检测了4 000行文本行,包含307 221个字符,结果表明,运行速度t与字符的大小和数量无关.当各文本线都检测时,$t=165.5\mu s$/字符,而当整个文本被处理时,基线初始斜率不必每一文本行都判断,因此,运行速度t减少到$82\mu s$/字符.即模糊字符印刷结构分析归类每秒可处理10^4以上字符,与字符识别速度相比,用上述字符预分类还是值得的,这有利于OCR的简化和识别率的提高.

有两个问题需要考虑.(1) 当字符很小时,容差将相应很小,因此,较小的噪声将导致错误分类.如图6所示,字符大小为5点,文本行高度为21.6像素,则$\delta=1.2$.为减少噪声影响,实际处理时取$\delta=2$;但若文本行高度小于16像素,也就是字符小于4点时,按上述方法处理,则文本行顶线和上基线的容差区将重迭.所幸的是,这样大小的字符很少用在标准文本中.(2) 当一文本行中含有大小不同的字符时,小字符无法正确分类,必须分解成另一文本行再作处理.如图7所示,文本行中字符串"IEEE MEMBER"中的字符被确定为中行字符.

Fig. 6 A text line with a small character size
图6 小字符文本行

Fig. 7 A special case of text contains different sizes of characters
图7 不同大小字符文本行

模糊字符印刷结构分析归类方法也可用于书写规范的手写字符,此时,字符是大写还是小写显得十分重要.如图8所示,左列第1例,文本行倾斜角也被检测出来;第2例字符被识别为"RosemARiE",用模糊字符印刷结构分类可纠正为"Rosemarie".

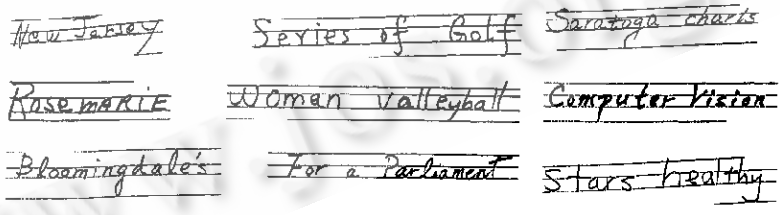


Fig. 8 Sample sets of handwritten characters for fuzzy typographical analysis
图8 用于模糊印刷分析的手写字符样本

6 结 论

本文提出了一种用于改进字符分类和识别的文本块字符印刷结构分析预分类的有效方法.文章给出了精确测定基线的算法,讨论了基线检测的容差和模糊字符印刷结构分类方法.基线检测也可用于文本的倾斜角测定,而基线用于确定字行的空间将有助于由多个文本块组成的段落的布局分析.经过实验,模糊字符印刷结构预分类的处理结果令人满意.

© 中国科学院软件研究所 http://www.jos.org.cn

参考文献

1 Hinds S C, Fisher J L, D'amato D P. A document detection method using runlength encoding and the hough transform. In: Proceedings of the 10th International Conference on Pattern Recognition. Atlantic City: IEEE Press, 1990. 464～468

2 Nakano Y, Shima Y, Fujisacw H. An algorithm for the skew normalization of document image. In: IEEE Pattern Recognition Seciety ed. Proceedings of the 10th International Conference on Pattern Recognition. Atlantic City: IEEE Press, 1990. 8～11

3 DeLuca P G, Gisotti A. Printed character preclassification based on word structure. Pattern Recognition, 1991,24(7):609～615

4 Hartigan J A. Clustering Algorithms. New York: John Wiley & Sons, 1975

5 Kandel A. Fuzzy Techniques in Pattern Recognition. New York: John Wiley & Sons, 1982

6 Pal S K. Fuzzy Mathematical Approach to Pattern Recognition. New York: John Wiley & Sons, 1986

7 Burden R L, Faires J D, Reynolds A C. Numerical Analysis. 3rd Ed., Boston: Prindle, Weber & Schmidt, 1985

A Character Preclassification Method Based on Fuzzy Structure Analysis of Typographical Characters

LU Da[1] XIE Ming-pei[2] PU Wei[1]

[1](*Changshu College Changshu* 215500)

[2](*Department of Computer Science Fudan University Shanghai* 200433)

Abstract In this paper, a new fuzzy-logic approach is presented for character preclassification which gives a precise calculation method for the baseline detection algorithm with tolerance analysis through analyzing the typographical structure of textual blocks. Other virtual reference lines are extracted with clustering technique. In order to ensure correct character preclassification, a fuzzy-logic approach is used to assign a membership to each typographical category for ambiguous classes. The results prove that the proposed fuzzy typographical analysis for character preclassification is able to process to more than 10000 characters per second on a SUN 4/490 workstation and the method has been tested for different font sizes and different types with satisfaction.

Key words Character preclassification, typographical categorization, baseline detection, fuzzy logic, fuzzy classification.

© 中国科学院软件研究所 http://www.jos.org.cn