

基于背景分析的手写数字切分方法^{*}

马 瑞 夏永泉 杨静宇

(南京理工大学计算机系 南京 210094)

摘 要 考虑到手写数字串的结构特点,提出一种以分析特定背景区域为主,针对具体粘连数字采用不同分割策略的切分方法。文中引入蓄水池的概念来形象描绘出背景区域中字符间的粘连部分,并从中抽取某些特征对字符的具体粘连形态进行了归纳分类。在分割过程中,根据字符的粘连类型选用不同的滴水算法来求得分割路径。实验结果表明了该方法对于手写数字分割的有效性。

关键词 手写数字分割,背景分析,蓄水池,滴水算法

Segmentation for Handwritten Numerals Based on Background Analysis

MA Rui XIA Yong-Quan YANG Jing-Yu

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

Abstract In this paper a scheme is presented for segmentation of handwritten numerals based on analysis of the specified background space. In the scheme, some interesting background regions called water reservoirs are first obtained by considering accumulation of water poured from the top or from the bottom of the numerals to illustrate where numerals touch each other. Then, the touching numerals are classified into six types according to the features extracted from the reservoirs. For numerals segmentation, a set of Drop Fall algorithms is developed and employed to generate the cutting paths with different strategy to separate each touching type. Experimental results show that the proposed scheme is effective for handwritten numerals segmentation.

Keywords Handwritten numerals segmentation, Background analysis, Water reservoir concept, Drop fall algorithm

1 引言

手写数字的分割和识别在许多重要领域中如邮政、财务、税务、金融等都有着广泛的实际应用背景。手写数字因其书写的随意性,字符本身存在断笔以及相邻字符间经常发生相碰、连笔或重叠情况,使手写数字的切分一直成为极具挑战的难题。为此,许多研究者提出了多种理论和算法。目前,常用的方法有:1)投影分析法,如文[1]利用字符串垂直方向的投影来检测分割点的位置;2)外轮廓分析法,如文[2]通过寻找上、下轮廓线的局部凹点来确定分割路径;3)基于前景、背景细化的方法,如文[3]主要根据细化后字符骨架的特征点,通过混和高斯概率密度来求得最佳分割路径。

虽然现有的切分算法具有较好的通用性,能够通过图像的分析自动寻找并确定最优分割路径,适用于各种文字串,但它们对具体的应用对象缺乏针对性,没有考虑手写数字串的特点。如投影分析法对字符发生严重倾斜或交错的情况无法处理,主要原因是垂直投影中连接字符的主体难以确定;当在数字字符间连笔划为直线的情况下,外轮廓分析法由于找不到轮廓线上的凹点而不太适合使用。本文考虑到手写数字串的结构特点,提出一种基于背景区域分析的分割方法。根据对特定背景区域中所获得信息的分析,将字符粘连情况进行了分类,以便针对具体的分割对象选用不同的分割方法。实验结果表明,该方法不仅对简单粘连字符能够较为准确地切分,而且对多处粘连的字符也行之有效。

2 图像预处理

字符串图像经二值化后,很容易产生小内孔或笔划断裂以及字符背景存在噪声干扰等,影响后续切分和识别的准确性。因此,需要对图像进行平滑处理,提高字符图像的质量。首先,可以利用文[4]中的方法得到字符图像的平均笔划宽度,记为 S_w ,通过该值来判断图像中的字符笔划是否过细,以免在对图像进行滤波操作时,某些过细的笔划部分被平滑掉。如果 S_w 小于某一设定阈值 T_w (通常取值为 3),则可根据 S_w 来确定适当的结构元素进行膨胀运算,从而实现笔划加宽的目的。另外,采用文[5]中的方法能够去除细小的毛刺,填补轮廓中的断裂部分,使字符轮廓变得更加光滑。

3 字符粘连类型

虽然数字字符本身笔划简单,但由于人们的书写风格以及工整程度差别较大,经常出现字符粘连的情况。根据实际发生的两字符间的连接方式,可大致分为以下几种基本类型(如图 1 所示):

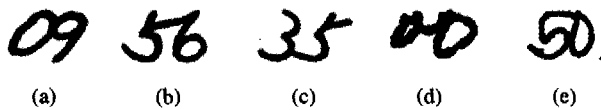
1)简单粘连:相邻两个字符间只有一处连接。

a. 共用粘连:字符间通过共用笔划相切连接。

b. 过渡粘连:字符间通过过渡笔划连接,往往由于书写时的连笔造成。

2)多处粘连:相邻两个字符间有多处连接。其特点是如果字符间在 N 处彼此连接,则会产生 $N-1$ 个封闭环。

^{*} 本课题得到国防基础研究项目(编号:J1500C002)基金资助。马 瑞 博士生,主要研究领域为模式识别;夏永泉 博士生,主要研究领域为计算机视觉;杨静宇 教授,博士生导师,主要研究领域为计算机视觉、信息融合、模式识别、智能机器人。



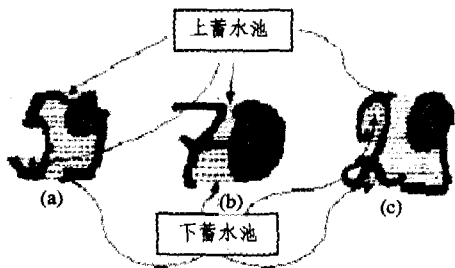
(a) 共用粘连; (b~d) 过渡粘连; (e) 多处粘连。

图1 数字字符的几种基本粘连类型

4 粘连字符分割

4.1 背景区域分析

大多数情况下,两个相连数字之间经常存在着较大的背景区域,为了更形象地描绘出这些区域,本文引入蓄水池(Water Reservoir)的概念^[6]。可以这样认为,这些背景区域实际上是当水流从字符串顶部向下灌入时,由于水流遇到字符连接部分的阻拦,而在字符轮廓的“盆底”部位蓄积形成的水洼,称之为(上)蓄水池(如图2所示)。类似地,如果水流从字符底部向上灌入,相应可得到(下)蓄水池。分析这些蓄水池对于字符切分有着重要的意义,主要因为:1)它集中描绘出字符间的连接部分;2)蓄水池的底部对应候选分割点的位置,缩小了分割点的搜索范围。



点线表示为上蓄水池,虚线表示为下蓄水池。

图2 蓄水池示例

下面所讨论的蓄水池原理就是对蓄水池进行分析,从而获得字符间的粘连情况。

对于给定的连通部件C,首先需要将其外接矩形在垂直方向按宽度比例分别为25%,50%和25%划分成3块区域,从左到右记作 v_1, v_m 和 v_r ;以相同方法得到水平方向的3块区域,从上到下分别记作 h_t, h_m 和 h_b 。然后根据定义可得到所有的蓄水池,从中选择重心落在 v_m 区域内并且高度大于1/8字符高度者,以及由字符笔划围成的封闭环(如图2背景中的灰色部分所示)作为分析对象,做下一步处理。

4.1.1 背景区域特征抽取

考虑到人们不同的书写习惯,抽取下面结构特征作为分析的主要参数:

1) 封闭环特征

N_L —封闭环数目; GL_k —封闭环 $L_k(1 \leq k \leq N_L)$ 重心;
 BL_k —封闭环 $L_k(1 \leq k \leq N_L)$ 外接矩形;

2) 蓄水池特征

N_{UW} —上蓄水池数目; N_{DW} —下蓄水池数目;
 N_w —蓄水池总数; GW_i —蓄水池 $W_i(1 \leq i \leq N_w)$ 重心;
 BW_i —蓄水池 $W_i(1 \leq i \leq N_w)$ 外接矩形;

4.1.2 字符粘连类型判断

1) 定义函数

$f_{hd}(a, b)$ —矩形 a, b 水平方向的重叠部分
 $f_{vd}(a, b)$ —矩形 a, b 垂直方向的重叠部分

$f_{hd}(a, b)$ —矩形 a, b 水平方向的间距

$f_{vd}(a, b)$ —矩形 a, b 垂直方向的间距

$s = \begin{cases} 0 & \text{连通部件为孤立字符} \\ 1 & \text{连通部件为共用粘连字符} \\ 2 & \text{连通部件为过渡粘连字符} \\ 3 & \text{连通部件为多处粘连字符} \end{cases}$

2) 利用特征参数通过以下条件来判断连通部件是否粘连及其粘连类型:

IF $N_L \geq 2$ AND $(\exists L_k, L_l) f_{hd}(BL_k, BL_l) > 0$

IF $(\exists L_k) GL_k \in v_m$ THEN $s = 3$

ELSE IF $(\exists L_k, L_l) f_{hd}(BL_k, BL_l) > 0$ THEN $s = 2$

ELSE $s = 1$

ELSE IF $N_w \geq 3$, THEN $s = 2$

ELSE IF $(\exists W_i) \text{height}(W_i) \geq 3/4 \text{height}(C)$

OR $(W_i, W_j) f_{vo}(BW_i, BW_j) > 0$ THEN $s = 2$

ELSE IF $N_L = 1$ AND $GL_k \in h_m$ AND $\text{width}(BL_k) < \text{width}(C)$

THEN $s = 3$ ELSE $s = 0$

4.1.3 过渡粘连分类

从字符发生过度粘连时背景区域的特点来看,过渡笔划的不同位置决定了过渡粘连的不同表现形式。因此,可以根据蓄水池的数目情况对过渡粘连字符进行以下分类(如图3所示):

IF $N_{DW} = 0$ AND $N_{UW} > 0$ THEN TYPE_1

IF $N_{UW} = 0$ AND $N_{DW} > 0$ THEN TYPE_2

IF $N_{UW} > 0$ AND $N_{DW} > 0$ AND $f_{hd}(BW_i, BW_j) > 0$ THEN TYPE_3

IF $N_{UW} > 0$ AND $N_{DW} > 0$ AND $f_{hd}(BW_i, BW_j) = 0$ THEN TYPE_4

其中:

TYPE_1: 连接部分只形成上蓄水池,主要是右边字符下半部的笔划触碰到左边字符上;

TYPE_2: 连接部分只形成下蓄水池,主要是左边的某些字符上半部某些笔划较长,如5,6和8;

TYPE_3: 连接部分形成上、下蓄水池,且水平方向有重叠部分,往往是由于书写时的连笔所造成,过度笔划多属于额外笔划;

TYPE_4: 连接部分形成上、下蓄水池,且水平方向无重叠部分。

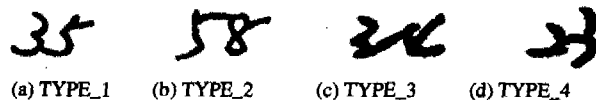


图3 过渡粘连的4种情况

4.2 分割算法

针对连接字符的切分问题,在现有的一些基于轮廓的切分方法^[7]中,滴水算法(Drop Fall Algorithm)由于执行效果相对较好,得到了较广泛的应用。但传统的滴水算法搜索时盲目性较大,对一些明显不合理的路径也要进行试探。本文将该算法应用于特定的背景区域中,避免了上述的局限性。

滴水算法主要是模仿水滴从高处向低处滴落的过程来对粘连字符进行切分。水滴从字符串顶部在重力的作用下,只

能沿字符轮廓向下滴落或水平滚动,当水滴陷在轮廓的凹处时,将渗漏到字符笔划中,经穿透笔划后继续滴落。最终水滴经过的轨迹就构成了字符的分割路径。滴水算法的决定因素主要包括:起始点、移动规则以及方向的确定。

以水滴从字符串左侧向下滴落为例,首先从上到下从左到右扫描图像中的每一行,选择第一个满足像素扫描情况为($\dots 1 * 0 \dots 01$)的白像素(*)为起始点,并规定其运动过程按

顺序遵循以下规则(如图4所示):

- Rule 1: if $n_3 = 0$, then next_pixel = n_3
- Rule 2: if $n_4 = 0$, then next_pixel = n_4
- Rule 3: if $n_2 = 0$, then next_pixel = n_2
- Rule 4: if $n_5 = 0$, then next_pixel = n_5
- Rule 5: if $n_1 = 0$, then next_pixel = n_1
- Rule 6: else next_pixel = n_3

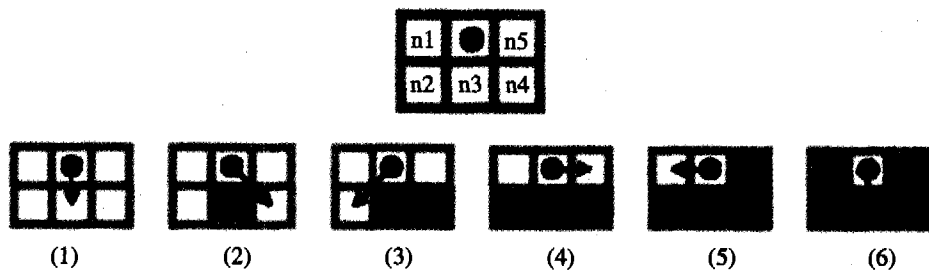
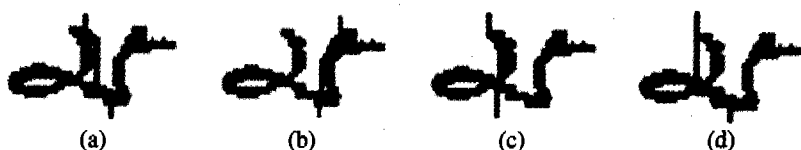


图4 滴水算法移动规则

显然,滴水算法根据起始点,移动规则和方向的不同确定,可从字符串的左、右两侧分别向上或向下进行滴落。因此,滴水算法包括了四种分割方法(如图5所示): Descend-

ing-Left Algorithm、Descending-Right Algorithm、Ascending-Left Algorithm、Ascending-Right Algorithm。



(a) Descending-Left Algorithm; (b) Descending-Right Algorithm;
(c) Ascending-Left Algorithm; (d) Ascending-Right Algorithm.

图5 滴水算法的4种分割方法

特别情况下,当水滴处在字符轮廓的局部最小值位置时,会朝左、右方向进行来回地滚动(如图4(4)和(5)),上述规则中的 Rule 4 和 Rule 5 均是对此类情况的一种判断。如果当下面两种情形 $4, 4, \dots, 4 \rightarrow 5, 5, \dots, 5$ 或 $5, 5, \dots, 5 \rightarrow 4, 4, \dots, 4$ 不断反复时,就需要在字符笔划内开始进行“渗漏”过程(Seeping Process):传统的滴水算法一般将轮廓线的极值点作为起始位置,在笔划区域内求取垂直方向的最短路径,直到遇见白像素为止。但当共用粘连的字符发生严重倾斜,并且字符间的笔划重叠部分比较长时,如果仍然采用垂直最短路径进行渗漏,常常会将一个字符的一部分划分到另一个字符中。

鉴于此,本文在沿用传统滴水算法的基础上,在处理该情况下的“渗漏”过程与原方法有所不同。考虑到共用粘连的字符大多数是笔划发生相切,而交叉情况并不常见,因此采用加窗中点连接的方法来尽量保证字符的完整性。即在笔划粘连部分的每一行中寻找一定宽度范围内的中间像素作为分割点,将这些中间像素连线作为粘连笔段内的分割路径。这里,窗口大小可设成2倍的 S_w 。该方法在实际中取得了较好的效果。

4.3 分割算法选择

得到前面的分析结果后,接下来就是如何针对具体的粘连类型选用适当的分割方法。根据字符粘连笔划的特点以及滴水算法的不同规则,本文采取的具体分割策略列于表1。

这里对表中当 $f=3$ 时的情况做下简要说明。对于发生多处粘连的两个字符,需要采取双向寻求分割路径的策略。

表1 本文采用的不同分割算法

字符粘连类型	滴水算法	渗漏过程
$f=1$	Descending-Left Algorithm	中点渗漏过程
$f=2$	TYPE_1	Descending-Left Algorithm
	TYPE_2	Ascending-Left Algorithm
	TYPE_3	Ascending-Left Algorithm
	TYPE_4	Descending-Left Algorithm
$f=3$	Ascending-Left Algorithm +	中点渗漏过程
	Descending-Left Algorithm	

即在位于连通部件 u_n 区域的封闭环中,寻找左上端白像素作为起始点,分别向上、下应用 Ascending-Left 和 Descending-Left 算法,最终完整的分割路径由上下两部分路径的并来确定。

4.4 多余笔划去除

实际书写时字符间存在的过渡笔划,可能是其中一个字符的某个笔划或是笔划的延长部分,也可能是不属于其中任意字符的多余笔划(如图1(d))。对于第二种情况,需要删除额外笔划以保证分割后字符的真实性。可以利用文[8]提出的方法来实现此目的。该方法简单有效,通过判断笔划是否伸出字符细化后的外接框,来决定是否将它删除。

5 实验结果

以银行支票上的手写体小写金额字符串为处理对象,从

(下转第207页)

又,对于 \sim ,满足

① $\forall P, (P, P) \in \sim$

② 若 $(P, Q) \in \sim$, 则 $(Q, P) \in \sim$

③ 若 $(P, Q) \in \sim$, 并且 $(Q, P) \in \sim$, 则 $(P, R) \in \sim$

故 \sim 是一个等价关系。

原命题得证。

结束语 量子通信的方式和通信的双方在空间上的相对距离有关,在本文中,由于通信双方可在同一个封闭系统中实现通信,故选择用辅助量子寄存器交换量子信息的方式为优。

量子进程和经典进程的主要不同点在于量子进程须考虑测量和酉变换的区别,在规则上体现为不同的内部通信动作和不同的输入输出动作,以及它们之间的关系。酉互模拟则

定义出两个进程是否仅通过酉变换达到一致,反映了外部世界和进程间的关系。另外,并发量子进程共同通过某个酉变换后不可被单独酉变换的张量积描述,量子并行和量子并发之间产生联系,这些将在未来的工作中有所体现。

参考文献

- Nielsen M A, Chuang I L. Quantum Computation and Quantum Information. New York, NY, Cambridge University Press, 2000
- Knill E H. Conventions for Quantum Pseudocode: [Technical Report]. LAUR-96-2724. Los Alamos National Laboratory, 1996
- Milner R. Communication and Concurrency. Prentice Hall, 1989
- Altenkirch T, Grattage J. A functional quantum programming language. quant-ph/0409065, November 2004
- 张效祥主编. 计算机科学技术百科全书. 第二版. 北京: 清华大学出版社, 2004

(上接第 200 页)

我国银行实地采集的 400 张现行支票中选出 500 例两字符粘连数字,应用本文的多种分割策略针对各种粘连情况进行了分割处理,实验统计结果列于表 2。

表 2 本文分割方法的统计结果

数字串 数目	正确分割率 (%)	错误分割率 (%)	识别率 (%)
简单粘连	93.47	6.53	82.15
多处粘连	89.41	10.59	78.53
总计	92.66	7.34	81.43

从表中可以看出,对 500 个粘连数字串的平均正确分割率为 92.66%。其中分割失败的主要原因是由于:(1)一个数字宽度远远小于另一数字宽度,并且其外接矩形完全包含于另一外接矩形内;(2)某些开口数字如 0、8、4 的不规范书写,影响了分析阶段的判断。

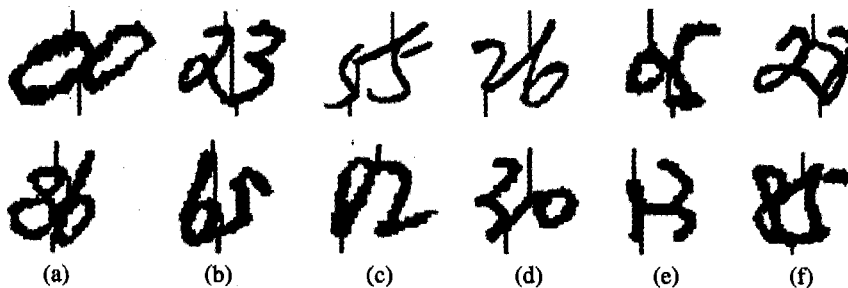


图 7 部分切分结果图

结论 虽然手写数字串的切分方法很多,但是字符粘连情况复杂,实际可以应用的字符切分算法并不多。本文在分析了手写数字串发生粘连时的背景区域特征基础上,归纳出字符之间的连接类型,并针对这些连接类型的具体情况选用不同的分割算法。在实践中对于简单粘连和四处粘连的字符都能得到较好的解决。今后的工作应对多字符粘连的分割作进一步研究。

参考文献

- 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
- Li Yun, Naoi S, Cheriet M, et al. A Segmentation Method for Touching Italic Characters. ICPR, 2004(2):594~597
- Chen Y K, Wang J F. Segmentation of single- or multiple-touch-

由于目前没有一种标准的方法来自动评价和测试字符切分算法,本文依据最近邻法的识别结果来对切分效果进行了衡量。所提出的分割方法识别率达到 81.43%,说明了该方法的有效性。

表 3 使用不同渗漏过程的分割结果比较

数字串 数目	渗漏 过程	正确分割率 (%)	错误分割率 (%)	识别率 (%)
173	传统垂直渗漏过程	92.18	7.82	80.64
	本文中点渗漏过程	83.31	16.69	71.87

实验中,在采用本文渗漏过程的同时,与传统渗漏过程进行了比较。表 3 的统计结果表明了在针对某些粘连类型的情况下,采用加窗中点渗漏过程进行处理,其分割效果要好于垂直渗漏过程,识别率也得到了提高。

图 7 给出了部分切分结果。

- ing handwritten numeral string using background and foreground analysis. IEEE Trans Pattern Anal Machine Intell, 2000, 22: 1304~1317
- Arica N, Yarman-Vural F T. Optical Character Recognition for Cursive Handwriting. IEEE Trans Pattern Anal Machine Intell, 2002, 24:801~813
- Bushofa BMF, Spann M. Segmentation and recognition of Arabic characters by structural classification. Image Vision Comput, 1997, 15:167~179
- Pal U, Belaid A, Choisy C. Touching numeral segmentation using water reservoir concept. Pattern Recognition Letters, 2003, 24:261~272
- Casey R G, Lecolinet E. A Survey of Methods and Strategies in Character Segmentation. IEEE Trans Pattern Analysis Machine Intelligence, 1996(7):690~706
- Elnagar A, Alhadj R. Segmentation of Connected Handwritten Numeral Strings. Pattern Recognition, 2003, 36(3):625~634