

## 一种联机手写算术表达式识别系统

姜寒<sup>①</sup> 刘长松<sup>①</sup> 李士阳<sup>①</sup> 邹燕明<sup>②</sup>

<sup>①</sup>(智能技术与系统国家重点实验室清华信息科学与技术国家实验室清华大学电子工程系 北京 100084)

<sup>②</sup>(诺基亚北京研究院多模态多媒体用户界面研究组 北京 100176)

**摘要:** 该文介绍了一个完整的联机手写算术表达式识别系统。在笔划分组与符号识别方面,采用预处理结合搜索的方法优化置信度函数;在结构分析方面,提出了一套全新的基于符号分级的算法,并利用二叉树这种数据结构存储识别结果。实验表明,对于书写较为整齐的算术表达式,此系统能够对较复杂的结构予以正确率较高的分析与识别。

**关键词:** 联机手写识别; 算术表达式; 结构分析; 符号分级

**中图分类号:** TP391.43

**文献标识码:** A

**文章编号:** 1009-5896(2010)05-1126-05

**DOI:** 10.3724/SP.J.1146.2009.00549

## An On-line Handwritten Arithmetic Expression Recognition System

Jiang Han<sup>①</sup> Liu Chang-song<sup>①</sup> Li Shi-yang<sup>①</sup> Zou Yan-ming<sup>②</sup>

<sup>①</sup>(State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China)

<sup>②</sup>(Multimodal Multimedia User Interface, Nokia Research Centre, Beijing 100176, China)

**Abstract:** In this paper a complete on-line handwritten arithmetic expression recognition system is introduced. During the process of stroke grouping and symbol recognition, pre-processing strategy and searching procedure are combined to optimize the confidence function. A novel structural analysis algorithm based on symbol ranking is described, and recognition results are stored in trinary trees as the data structure. Experiments show that relatively neatly written expressions with fairly complex structures can be analysed by this algorithm with a high correction rate.

**Key words:** On-line handwriting recognition; Arithmetic expressions; Structural analysis; Symbol ranking

### 1 引言

随着文字识别技术(联机的和脱机的)的发展,向计算机输入文字变得越发容易,但是实时的数学公式输入仍是比较繁琐的。从上世纪 90 年代开始,关于数学表达式电子化出现了很多新的工作。作为这个领域的一篇有代表性的综述文章,文献[1]除了介绍了上世纪后期出现的很多可以用于数学表达式识别的方法之外,更重要的是明确了数学表达式识别问题的定义和基本框架。Zannibbi 在文献[2]给出了一种基于树变换的公式识别方法;Tapia 在文献[3]中利用并改进了 Zannibbi 的思想,将最小生成树用于结构分析,提出了一种快速的结构分析方法;而 Vuong 在文献[4]中提出了一种需要使用者实时参与的系统,可以边书写,边反馈,边修改;文献[5]提

出了一种搜索框架,利用搜索的方法解决公式识别所涉及的全局优化问题。另外,也有文章专门论述笔划分组与字符识别的问题(例如文献[6])。本文利用了文献[6]中提出的方法并加以改进,并加入新的基于符号分级的结构分析算法,完成手写算术表达式的联机识别的。还有一类方法纯粹基于统计学习,这类方法的代表是文献[7, 8]。

本文安排如下:第 2 节将对手写算术表达式联机识别问题进行具体的描述;第 3 节介绍笔划分组和字符识别方法;第 4 节介绍基于符号分级的公式结构分析算法;第 5 节对整个系统,尤其是结构分析部分进行了实验;第 6 节给出结论,并提出对未来工作的展望。

### 2 问题描述与有关假设

输入符号限制为 10 个数字、小数点、四则运算符(其中除号包括{÷}和{/}两种)、圆括号、分数线和根号这个集合。据此,每个字符的笔划数假定为不超过 3。输出为字符串形式的识别结果。

关于输入的其他限制有:第一,属于某一符号

2009-04-13 收到, 2009-11-25 改回

国家 973 计划项目(2007CB311004)和国家自然科学基金(60772049, 60872086)资助课题

通信作者: 姜寒 jehovah0121@qq.com

的笔划应该在时间上不被属于其他符号的笔划打断；第二，不同符号之间没有连笔，且不同符号间的笔划没有粘连。特别地，不需要遵循严格按照从上到下，从左到右的顺序进行书写的要求。

整个系统的框架基本符合文献[1]中提出的经典框架，如图 1 所示。

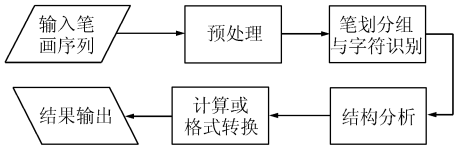


图 1 数学表达式识别系统的经典框架

### 3 笔划分组与字符识别

由于假定了属于同一字符的笔划在时间上不被打断，因此笔划分组的过程仅需要得到一个序列  $\{s_k\}$ ，其中第  $k$  个元素表示第  $k$  个分组的最末一笔的编号，然后每个分组识别为一个字符。分组的原则是选取“可能性最大”的  $\{s_k\}$ 。一个笔划分组方案  $\{s_k\}$  的置信度  $C$  定义为  $C = \frac{1}{K} \sum_{k=1}^K C_k$ ，其中  $K$  是分组(即切分出的字符)个数， $C_k$  是第  $k$  个分组的置信度。 $C_k$  的定义为

$$C_k = \lambda_R C_{Rk} + \lambda_N C_{Nk} + \lambda_A C_{Ak}$$

这里  $C_{Rk}$  是将当前第  $k$  个笔划组送入单个字符识别核心所返回的置信度，称为识别置信度。 $C_{Nk}$  称为笔划数置信度，是第  $k$  个笔划分组中笔划个数  $N_k$  的函数。由于假定的输入字符集，我们可以指定， $N_k \geq 4$  时  $C_{Nk}$  数值较小。此外，新加入了  $C_{Ak}$ ，称为补充置信度，针对实验中出现的一些具体情况进行人工补偿。相应地， $\lambda_R, \lambda_N, \lambda_A$  是各个置信度分量所占的权重。

选出置信度  $C$  最高的笔划分组方案作为切分结果，并由识别核心选用相应的字符识别结果。对于书写随意性较强的小数点，本文单独利用外接框面积进行小数点的识别。

对于一个笔划序列，其中所有可能的笔划组(第  $m$  笔到  $n$  笔， $m \leq n$ )的置信度值  $C(m, n)$  均被计算出来并存入一个上三角矩阵，对于所有的切分路径进行遍历，求出置信度最大的路径。图 2 给出了一个置信度矩阵的例子。其中深色部分标出了一个完整的分组路径。

由于这一方法本质上是穷举搜索，对于有  $L$  个笔划的笔划集合而言，不加限制分组的方式有  $2^{L-1}$  种。加入单个字符笔划数不大于 3 这一限制后，对于较大的  $N$ ，可能的分组方式数量仍很可观。预处

$C(1,1)$	$C(1,2)$	$C(1,3)$	$C(1,4)$	...	$C(1,K)$
0	$C(2,2)$	$C(2,3)$	$C(2,4)$	...	$C(2,K)$
0	0	$C(3,3)$	$C(3,4)$	...	$C(3,K)$
0	0	0	$C(4,4)$	...	$C(4,K)$
...	...	...	...	...	...
0	0	0	0	...	$C(K,K)$

图 2 置信度矩阵和笔划分组路径

理模块的加入较好地解决了这一问题。本文假定了属于不同符号的笔划之间相互没有粘连，这等同于认为相交的笔划都属于同一字符。这种利用笔划相交信息提前合并笔划组的方法叫做预合并。类似地，利用笔划间简单的位置关系提前确定两个笔划不可能属于同一分组的方法叫做预切分。

经过上述过程，输入的笔划序列被分组、识别，得到了若干符号以及每个符号的一些位置信息。

### 4 基于符号分级的结构分析算法

尽管数学表达式使用的字符集远远小于汉字识别，但在空间结构和逻辑结构方面复杂性都大大增加，结合书写上的随意性，使得识别难度大大增加。

为了避免孤立的考虑符号两两之间的位置关系，本节提出一种基于符号分级的结构分析算法：

- (1) 基于规则和阈值对每两个符号间的位置关系进行初步判定；
- (2) 将符号按其统领能力进行分类；
- (3) 借助前两步得到的信息对符号间的统领关系进行初步判定；
- (4) 对符号按统领关系进行分级；
- (5) 生成最终的表达式树。下面逐步详细说明，并在最后给出结构分析算法复杂度的估计。

#### 4.1 位置关系的判定

设置一矩阵  $P$ ，用来存放对位置关系的初步判定结果。设  $K$  是笔划组数(即字符数)，则  $P$  为  $K \times K$  矩阵。计算矩阵  $P$  的原则如下：(1)  $P$  的第  $k$  行记录与第  $k$  个字符  $s_k$  有关的字符，每一列记录与  $s_k$  呈现相应关系；(2) 此时位置关系包括六种，即上、下、左、右、左下、被包围，分别对应  $P$  的各列。利用外接框的重叠、外接框间的距离等位置信息，可以计算出矩阵  $P$ ，所需时间复杂度为  $O(K^2)$ 。

#### 4.2 符号分类与统领关系的判定

在对符号分级，以便反映出表达式的层次结构之前，还有两点准备工作要做。

首先是符号分类。在算术表达式中，具有统领能力的符号只有两种：根号和分数线。可以把符号分为 3 类：第 0 类为根号，第 1 类为分数线，第 2 类为其它符号。

其次是统领关系的初步建立。设置矩阵  $D =$

$(D_{ij})_{K \times K}$ , 初始状态置为 0。令其第  $k$  行表示  $s_k$  与其它符号的直接统领关系。此后若  $D_{ij} > 0$ , 则表示  $s_i$  统领  $s_j$ ; 若  $D_{ij} < 0$ , 则表示  $s_i$  被  $s_j$  统领。

#### 算法 1

对每个符号  $s_k$ , 利用数组  $T$  和矩阵  $D$  进行如下判断:

(1) 若  $s_k$  上方符号为分数线  $s_l$ , 下方符号不存在或者存在但不是分数线, 则置  $D_{kl} = -1, D_{lk} = 1$ 。

(2) 若为对称情况, 则置  $D_{kl} = -2, D_{lk} = 2$ 。

(3) 若  $s_k$  被根号  $s_l$  包围, 则置  $D_{kl} = -1, D_{lk} = 1$ 。

(4) 若  $s_k$  上方符号为分数线  $s_m$ , 下方符号为分数线  $s_n$ , 则比较  $s_m, s_n$  的外接框宽度。不妨设  $s_m$  较宽, 此时置  $D_{mn} = 1, D_{nm} = -1, D_{kn} = -2, D_{nk} = 2$ 。

算法 1 中步骤(4)是为了对于一个繁分式, 可判定多条分数线的主次关系。另外不难看出, 执行上述两个步骤的时间复杂度为  $O(K)$ 。

### 4.3 符号分级

按照统领关系矩阵  $D$ , 将符号分成不同级别, 存储在数组  $R$  中。不被任何符号统领的符号为 0 级。若一个符号  $s_k$  被其他符号统领, 且这些符号中级别最高的为  $r$  级, 则  $s_k$  应为  $r + 1$  级。

由上述定义可知, 在确定一个符号的级别之前, 必须先确定所有统领它的符号的级别。据此要求, 我们使用如下算法:

#### 算法 2

(1) 检查矩阵  $D$ , 将不被任何符号直接统领的符号级别置为 0 级。

(2) 检查数组  $R$ , 如果  $R$  中所有的元素都已经非负, 则跳出本算法。否则继续。

(3) 设  $R$  中元素最大值为  $M$ 。对所有的  $k \in \{1, 2, \dots, K\}$ , 考察所有统领  $s_k$  的符号  $s_l$ , 若所有  $R_l$  都非负, 且不等于  $M + 1$ , 则置  $R_k = M + 1$ 。转(2)。

尽管整个系统最终的识别率还取决于众多其它因素, 但是这里给出了一个能体现数学的层次结构的方法。不难看出, 算法 2 的复杂度为  $O(K^2)$ 。

### 4.4 生成表达式树

本文采用三叉树(Ternary Tree)结构作为存放表达式的数据结构。三叉树结构是一个有序的根树, 其根结点为主基线上的首字符。每个节点至多有 3 个子节点, 节点  $N$  的 3 个子节点记为  $\text{Left}(N)$ ,  $\text{Right}(N)$  和  $\text{Next}(N)$ , 且不论节点  $N$  对应何种符号, 都用  $\text{Next}(N)$  来记录其在同一基线上的下一个符号(即右邻)。若  $N$  为根号, 则  $\text{Left}(N)$  表示其中包含的子式的主基线首字符; 若  $N$  为分数线, 则  $\text{Left}(N)$  表示分子主基线首字符,  $\text{Right}(N)$  表示分母主基线

首字符; 若  $N$  为其它符号, 则  $\text{Left}(N)$ (如果非空)表示其幂指数主基线首字符。

首先, 前面算法对算术表达式层次结构的体现仅限于根号和分数线, 因此还需要加入对于乘幂这一关系的层次化表示。假设一个符号子集  $L(s)$ , 其中的元素具有相同的级别  $r$ (即在数组  $R$  中对应相同的值)且都被级别为  $r - 1$  的符号  $s$  统领。根据前述算法可以断定,  $L(s)$  中的符号只有左右相邻和乘幂这两种关系。下面算法将对这种情况生成一个三叉树:

#### 算法 3

(1) 设空树  $T$ , 并将  $L(s)$  中外接框左边缘横坐标最小(即最靠左)的字符作为  $T$  的根。

(2) 建立一个数组, 用于存放  $L(s)$  中的符号及其与右上字符和右邻字符的对应关系。右上字符的确定可利用矩阵  $P$  完成。

(3) 利用这一数组, 按照三叉树的定义中关于乘幂和右邻关系的部分, 将  $L(s)$  中的符号穿线成树, 构造为三叉树  $T$ 。

(4) 返回  $T$ 。

不难看出, 算法 3 的时间复杂度为  $O(|L(s)|)$ 。

建立了算法 3 之后, 对于算术表达式层次结构的分析已经趋于完整, 即已经分析了所有可能出现的层次关系。最后需要利用这些信息构造完整的表达式三叉树  $T$ , 这由算法完成。

#### 算法 4

(1) 设置空集合  $X$ 。设级别数组  $R$  中最大的元素为  $M$ 。

(2) 统计所有级别为 0 的符号, 添加入集合  $X$  中。调用算法 3, 将  $X$  作为其中的  $L(s)$ , 返回三叉树  $T$ 。置  $m = 1$ 。

(3) 对  $X$  中的每个符号  $s$ , 设其统领的  $m$  级符号集为  $L_m(s)$ 。用  $L_m(s)$  作为输入调用算法 3, 返回树  $T_m$ 。判断  $T_m$  根节点与  $s$  的关系, 按照三叉树的定义将  $T_m$  加入为  $s$  的一个子节点。

(4) 置  $X = \cup_{s \in X} L_m(s)$ 。若  $m = M$  则算法结束, 返回  $T$ 。否则, 置  $m = m + 1$ , 转(3)。

假设级别为  $m$  的符号有  $K_m$  个, 于是符号总数  $K = \sum_{m=0}^M K_m$ 。(2)中调用算法 3, 需要的时间复杂度为  $O(K_0)$ 。(3),(4)循环执行第  $m$  次时, 对当时  $X$  中的每一个元素都要调用一次算法。设  $s \in X$ , 被  $s$  统领的第  $m$  级符号有  $K_m(s)$  个, 于是有  $K_m = \sum_{s \in X} K_m(s)$ , 则(3)(4)循环执行第  $m$  次时, 时间复杂度为

$$\sum_{s \in X} O(K_m(s)) \leq |X| O(\max_{s \in X} \{K_m(s)\}) \leq K_{m-1} O(K_m)$$

从而算法 4 完整执行的时间复杂度的一个上界为  $O(K_0) + \sum_{m=1}^M O(K_{m-1}K_m) = O(K_0) + O(K_0K_1) + K_1K_2 + \dots + K_{M-1}K_M \leq O(K_0) + O(K^2) = O(K^2)$

#### 4.5 时间复杂度分析

由前面的分析可以看出，结构分析部分每个环节的复杂度都不超过  $O(K^2)$ ，所以总的复杂度上界就是  $O(K^2)$ 。而对第 3 节介绍的切分识别方法进行复杂度估计，其阶次是超过多项式的。因此，以上结构分析算法速度较快。

### 5 实验

系统性能的评价分为 3 个层次：第 1 层是笔划分组正确率，第 2 层是字符识别正确率，第 3 层是逻辑关系(即二叉树的边)正确率。

测试样本是来自 10 位书写者的 31 个公式样本，算术本身具有一定的复杂性，书写方式亦具有一定的随意性。测试样本的一些示例如图 3 所示。这些样本涉及字符 472 个，逻辑关系 441 个。

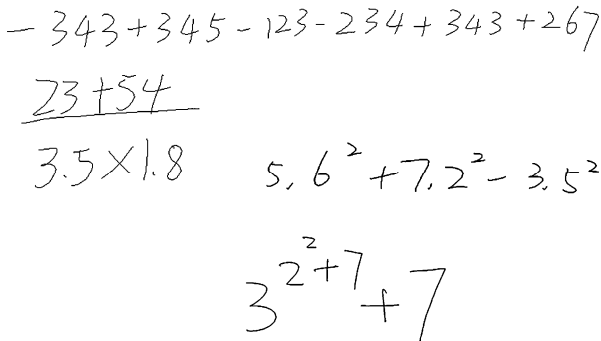


图 3 测试用的部分手写算术表达式样本截图

实验表明，笔划分组正确率为 93.6%；在分组正确的基础上，字符识别正确率为 92.5%，占总字符数的 86.7%；逻辑关系识别正确率为 84.8%。

借助图形接口，可以如图 4 所示看到一些样本笔划分组和字符识别的结果样例，而图 5 给出了两个笔划分组出错的典型样例。其中图 5(a)第 2 项 7/9 与后面的减号分在了一组，并识别为除号；另外最后一项中分数线与分母分在了同一组并被识别为等号。这正是笔划分组和字符识别的两种典型错误：第一，由于除号的特殊性，其分组和识别所依赖的规则较强，以至于对很多形状类似的笔划组将错误的分组在一起并识别为除号。第二，由于书写习惯的不同，根号有时会被切分为两组。第三，由于现

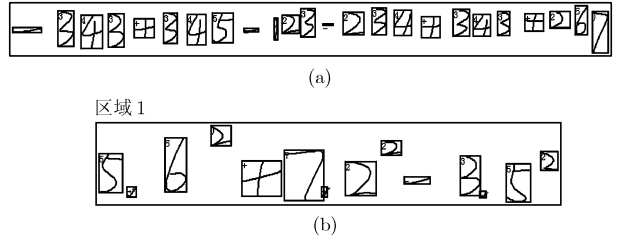


图 4 笔划分组及字符识别结果示例

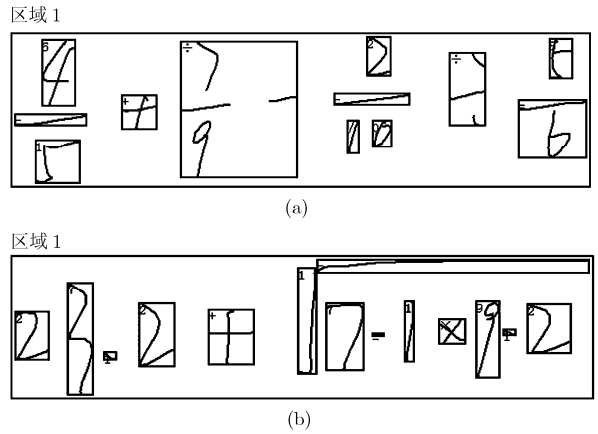


图 5 笔划分组及字符识别错误示例

有的字符识别核心并非针对西文字符设计，因此会将一些不相干的笔划识别为某个字符。除此之外，相似字符会出现混淆，如 {7} 和 {}，{7} 和 {2} 等等。

图 6 给出了一些样本的最终处理结果示例。在笔划分组和字符识别没有严重错误的情况下，逻辑关系判断中典型的错误形式包括：个别字符的遗漏、角标关系与水平邻接关系的混淆等。

```
File Name: .\..\samples\...\samples\081011may05.ehw
Final result: 56+7.2^(2)-35^(2)
File Name: .\..\samples\...\samples\082315may06.ehw
Final result: 27.2+17-1*9.2
File Name: .\..\samples\...\samples\082914may06.ehw
Final result: (0.603*7.5+4-0.001)/(2)/(3))+0.1^(2)
File Name: .\..\samples\...\samples\083814may06.ehw
Final result: -3.3+343-12>-239+303+2.
File Name: .\..\samples\...\samples\091711may05.ehw
Final result: .469-885
File Name: .\..\samples\...\samples\092511may07.ehw
Final result: 23.57+3474
File Name: .\..\samples\...\samples\092610may07.ehw
Final result: 2).93003
```

图 6 部分样本的批处理结果示例

图 7 给出了一个样本及其完整的识别结果。可以看到，图 7(b)第 1 部分 Split result 为笔划分组的结果，形式  $k:(m,n)$  表示第  $k$  个笔划组是由第  $m$  到第  $n$  个笔划构成的；第 2 部分 Recognized characters 为响应的识别结果(包括了一些候选字)；第 3 部分 point 给出了位置关系矩阵  $P$  的数值；最后将识别

$$\frac{\sqrt{\frac{2}{3}}}{\sqrt{\frac{4}{5}} + \sqrt{7.6}}$$

(a)

```

C:\WINDOWS\system32\cmd.exe
[Debug] Split result:
0: < 0, 0>. 1: < 1, 1>. 2: < 2, 2>. 3: < 3, 3>. 4: < 4, 4>. 5: < 5, 5>. 6: < 6, 7>. 7: < 8, 8>. 8: < 9, 10>. 9: < 11, 12>. 10: < 13, 13>. 11: < 14, 14>. 12: < 15, 15>. 13: < 16, 16>.
[Debug] Recognized characters:
0: < + 1 ->. 1: < 2 1 0>. 2: < - - .>. 3: < 3 3 9>. 4: < - 1 .>. 5: < + 1 0>. 6: < 4 . 1>. 7: < - 1 .>. 8: < 5 3 0>. 9: < + 4 1>. 10: < + - 0>. 11: < 7 1 .>. 12: < - 1 4>. 13: < 6 0 0>.
[Debug] point: the array for position relations.
Description of columns:
0. enclosing
1. upper
2. enclosed
3. left
4. right
5. lower left
6. lower
   0 1 2 3 4 5 6 7
0:  3 -1 -1 -1 -1 -1 4 2
1: -1 -1 0 -1 -1 -1 2 -1
2: -1 1 0 -1 -1 -1 3 -1
3: -1 2 0 -1 -1 -1 4 -1
4: -1 3 -1 -1 -1 -1 13 -1
5:  8 4 -1 -1 9 -1 -1 7
6: -1 4 5 -1 10 -1 7 9
7: -1 6 5 -1 9 -1 8 13
8: -1 7 5 -1 -1 -1 -1 -1
9: -1 4 -1 7 10 -1 -1 -1
10: 13 4 -1 9 -1 -1 -1 12
11: -1 4 10 9 12 -1 -1 13
12: -1 4 10 11 13 -1 -1 -1
13: -1 4 10 9 -1 -1 -1 -1
File Name: .\..\samples\S6214october28.ehw
Final result: <<<sqrt<<2>/<3>>>>><<sqrt<<4>/<5>>>>>+<sqrt<7.6>>>
请按任意键继续. . .
  
```

(b)

图 7 完整识别结果示例

出的表达式以文本形式输出。

此外，程序运行的计算机具有 3 GHz 主频的 CPU，在包含对每个样本的识别结果进行控制台输出的情况下，处理 31 个样本的总时间约为 1 s，说明本系统所使用的方法具有较好的实时性。

### 6 结论

本文提出的针对算术表达式的结构分析算法具有一定的鲁棒性，较高的识别率以及较快的运行速度。在切分和单个字符识别不出现重大错误的前提下，该结构分析算法可以识别结构具有一定复杂性的算术表达式。另外，这一算法完全有可能推广到更一般的数学公式识别系统中。

由于系统的切分与字符识别过程均基于人工指定的规则和阈值，故而效果还不够理想。可能的改进包括：第一，将规则体系完善，不仅使用笔划的外接框信息，更加入印刷中心、凸包等更为细致的信息；第二，用训练的方法指定规则阈值，使得所用规则更为可靠。

### 参考文献

[1] Chan K F and Yeung D Y. Mathematical expression recognition: A survey [J]. *International Journal on Document Analysis and Recognition*, 2000, 3(1): 3-15.

[2] Zabimni R and Blostein D. Recognizing mathematical expressions using tree transformation[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002, 24(11):

1455-1467.

[3] Tapia E and Rojas R. Recognition of On-line Handwritten Mathematical Expressions Using a Minimum Spanning Tree Construction and Symbol Dominance[M]. *Graphic Recognition*. Berlin, Germany: Springer, 2004: 329-340.

[4] Vuong B Q, Hui S C, and He Y. Progressive structural analysis for dynamic recognition of on-line handwritten mathematical expressions[J]. *IEEE Pattern Recognition Letters*, 2008, 29(5): 647-655.

[5] Rhee T H and Kim J H. Efficient search strategy in structural analysis for handwritten mathematical expression recognition. [C]. 11th International Conference on Frontiers in Handwriting Recognition, 2008: 19-24.

[6] 虞瑾, 丁晓青. 联机手写公式中字符的切分与识别[J]. *电视技术*, 2007, 1: 148-150.

[7] Shi Y, Li H, and Soong F K. A unified framework for symbol segmentation and recognition of handwritten mathematical expressions[C]. *International Conference on Document Analysis and Recognition*. 2007, 2: 854-858.

[8] Luo Z X, Shi Y, and Soong F K. Symbol graph based discriminative training and rescoring for improved math symbol recognition[C]. *International Conference on Acoustics, Speech, and Signal Processing*, 2008: 953-956.

姜 寒: 男, 1984 年生, 硕士生, 研究方向为联机手写文字识别。

刘长松: 男, 1969 年生, 副教授, 长期从事文字识别、文本图像处理、模式识别、中文信息处理等方面的教学和科研, 以及产品开发工作。