·研究简报·

(22) 117－124

手写数学符号的基元识别方法*

赵学军　余楚中　杨波　潘保昌
（重庆大学人工视觉实验室，重庆，400044；第一作者 31 岁，男，讲师，博士生）

TP391.4

摘　要　对手写数学字符的联机识别进行了研究。首先分析 94 个常用数学符号的结构，指出这些符号均由 10 个基本结构元组成；其次抽取数学符号的三个重要特征：基元矢量、基元之间的位置关系和基元长度矢量并采用了一种基元排序法。同时考虑匹配值和不匹配值以及增加几何约束改进了传统的动态规划匹配方法，并用改进的 Kuhn－Munkres 算法求最佳匹配。在对 20 人无限制书写的数学符号的识别试验中，正确识别率为 92.52%，误识率为 3.03%，拒识率为 4.45%.

关键词　识别；数学符号；特征提取；字符匹配
中国图书资料分类法分类号　TP391.4

计算机

0　引　言

由于计算机技术的高速发展，人们可以快速处理大量的数据，如对大量信息的存储、查询、统计等。但数据的输入速度远远低于数据处理的速度，这大大妨碍了人们对计算机的使用。为了解决数据的输入问题，人们作了长期的积极的努力，提出了许多方法[1~3]，如各种基于键盘的输入方法，但在拟定一份稿子时，为了集中注意力完成稿子的制作，人们常常喜欢用笔和纸书写而不采用键盘，因为用键盘输入会打断正在进行的思维。所以许多人致力于笔输入方法的研究。十多年来，已经提出了大量的研究报告，也已开发出能识别机械打印字符、书写印刷体字符和联机手写字符等多种系统。虽然在有些领域如自由手写体字符的识别并不尽人意。在脱机识别西文印刷体、手写体字符，联机识别西文字符，脱机联机识别印刷体、手写体汉字等领域都在积极的研究和探索[4~6]，已经或不久将会有优秀的识别系统在市面上流行。

但对自然科学工作者，特别是数学工作者来说，这些发展和成就都似乎与他们的关系不大，因为这些系统并不能解决或完全解决他们的数据输入问题。他们中不少人仍然使用手写文稿进行交流，原因是没有好的自然的数学表达式的快速输入系统。目前能输入数学表达式的系统全部采用键盘输入，不仅需要长时间的学习、记大量的控制命令，而且输入速度慢，这些系统并不受他们的欢迎。他们盼望优秀的数学表达式的笔输入系统问世。

我们对数学表达式中可能出现的字符进行了分类，第一类为阿拉伯数字、英文大小写字母及几个使用频率很高的希腊小写字母和数学符号；第二类为希腊大小写字母；第三类为数

* 收文日期　1997-02-25

学符号。本文就第三类字符的联机识别进行研究。

第三类包含如下字符：

. ′ ″ ‴ * ∵ ∴ ∶ ··· ⋮ ⋰ ⋱ ° ∂ ⊕ ⊗ ∮ ∯ ∰ ∅ ∞ ∝ % △ ∼ ≃ ≈ ≅ ∫ ∬ ∭ ∈
∉ ⊂ ⊄ ⊃ ∪ ∩ ⊆ ⊇ ⌒ ^ ( ) { } + − ± ∓ ÷ — < ⩽ ≮ > ⩾ ≯ = ≠ ≡ ▯
→ ← ↔ ⇒ ⇔ ↦ ∠ ⊥ × / √ \ ∀ ∧ ∨ ı ▽ | ‖ // ↑ ↓ ↕ ⇑ ⇓ ⇕ [ ]

1 字符结构

通过对 52 人随意写出的 620×94 个字符的分析得知，这 94 个字符在一般自由手写体中由下列基本结构元构成：

— | ( ) ⌣ ⌒ ○ / \ ·

依次编号为 0、1、2、3、4、5、6、7、8、9，称为基元代码。

2 特征抽取

2.1 基元矢量

先对输入字符进行预处理，包括触点连续化、字符点阵规范化。然后对坐标点进行顺序跟踪，提取基本(结构)元，按书写顺序得到字符的基元矢量，如“士”的基元依次为“—”、“|”、“—”，基元码依次为 0、1、0，从而基元矢量为(0,1,0). 由于不同的人的书写习惯不同，字符“士”的基元矢量的元素的顺序可能不同，可能为(0,0,1)、(1,0,0)等。而积分号“∫”的基元矢量可能为(8,1,9)、(8,9)、(0,1,0)、(0,1,9)等。

2.2 基元间的位置关系

除点“·”和圈“○”外的 8 个基元，我们取其三个特征点：头点(即始点)，中点(即基元的中间点)，尾点(即终点)。

将基元之间的位置关系分为连接和分离两大类：

连接关系有 9 种：头头(11)、头中(12)、头尾(13)、中头(21)、中中(22)、中尾(23)、尾头(31)、尾中(32)和尾尾(33)关系。例如头中连接是前一基元的头点与后一基元的中点相连接。连接的含义是两点的距离小于某个阈值。

分离关系有 8 种：左上(la)、左中(lm)、左下(ld)、中上(ma)、中下(md)、右上(ra)、右中(rm)、右下(rd)关系。例如正上关系是指前一基元在后一基元的正上方。

设 x^1_{max}、x^1_{min}、y^1_{max}、y^1_{min} 分别为前一基元的最大 x 坐标、最小 x 坐标、最大 y 坐标、最小 y 坐标，x^2_{max}、x^2_{min}、y^2_{max}、y^2_{min} 分别为后一基元的最大 x 坐标、最小 x 坐标、最大 y 坐标、最小 y 坐标。

若 $\frac{y^2_{max}+y^2_{min}}{2} > y^1_{max}+\varepsilon$，则

当 $\frac{x^2_{max}+x^2_{min}}{2} < x^1_{min}-\varepsilon$ 时为左上关系；

当 $\frac{x^2_{max}+x^2_{min}}{2} > x^1_{max}+\varepsilon$ 时为右上关系；

其它为正上关系；

若$\frac{y_{\max}^2+y_{\min}^2}{2}<y_{\max}^1-\varepsilon$，则

　　当$\frac{x_{\max}^2+x_{\min}^2}{2}<x_{\min}^1-\varepsilon$时为左上关系；

　　当$\frac{x_{\max}^2+x_{\min}^2}{2}>x_{\max}^1+\varepsilon$时为右上关系；

　　其它为正下关系；

否则，当$\frac{x_{\max}^2+x_{\min}^2}{2}<x_{\min}^1-\varepsilon$时为正左关系；

　　当$\frac{x_{\max}^2+x_{\min}^2}{2}>x_{\max}^1+\varepsilon$时为正右关系。

基元点"·"与其它基元的位置关系均为分离关系。

基元圈"○"与其它基元的位置关系为：内(i)、外(o)、交叉(c)。其中内关系是指其它基元在圈内，如 ⊕、⊗ 等，外关系是指其它基元在圈外，如 % 等，交叉关系是指其它基元与圈是交叉的，如 $\oint$ 等。

由上述方法，可以抽取手写字符的每两个基元的位置关系码，组成位置关系矩阵 R：

$$R=(r_{ij})_{m\times m}$$

其中 r_{ij} 为基元 i 与基元 j 之间的位置关系，$r_{ii}=-1$，$i,j=1,2,\cdots,m$. m 为字符的基元个数。例如字符"±"、"="和"‖"的基元矢量分别为(0,1,0)、(0,0)、(1,1). 其位置关系矩阵 R_1、R_2、R_3 分别为：

$$R_1=\begin{array}{c}\begin{array}{ccc}0 & 1 & 0\end{array}\\ \begin{pmatrix}-1 & 22 & ma\\ 22 & -1 & 23\\ md & 32 & -1\end{pmatrix}\end{array}\begin{array}{c}\\ 0\\ 1\\ 0\end{array},\quad R_2=\begin{array}{c}\begin{array}{cc}0 & 0\end{array}\\ \begin{pmatrix}-1 & ma\\ md & -1\end{pmatrix}\end{array}\begin{array}{c}\\ 0\\ 0\end{array}\quad R_2=\begin{array}{c}\begin{array}{cc}1 & 1\end{array}\\ \begin{pmatrix}-1 & ml\\ mr & -1\end{pmatrix}\end{array}\begin{array}{c}\\ 1\\ 1\end{array}$$

2.3 基元长度矢量

除基元圈不考虑其长度外，其它 9 种基元均可计算其长度。用 l_i 表示字符的第 i 个基元的长度，圈的长度用 -1 表示，从而得字符的基元长度矢量为 $L=(l_1,l_2,\cdots,l_m)$.

3 基元排序

由于不同的人的书写习惯不同，在书写相同字符时会出现多种不同的顺序。虽然数学符号不象汉字那样，有顺序变化的字符不是特别多、变化种类也不是不可预料，但如果对这些顺序的不同情况均加以列举，同样会大大增加摸板数量，大大增加存储空间的开销，同时也增加了查找时间。为了减少由于书写顺序的任意性带来的时间和空间的开销与对不同习惯的不适应性，特为十种基元按一般人的书写习惯，规定一种标准顺序，把实际书写的基元按标准顺序重新排序。比如规定横在竖的前面，则"±"的三种顺序均统一为：(0,0,1)。摸板只考虑排序后的结果，从而存储空间可大大减少。

通过对 94 个字符的分析，规定基元顺序标准矩阵，如表 1 所示。

其中元素 1 表示相应的 X 应在 Y 之前；0 表示 X 在 Y 之前，或不规定 X 与 Y 的顺序。

根据上述基元顺序标准矩阵从书写字符的基元位置关系矩阵 R 导出基元顺序矩阵 S。例如"±"的三种书写顺序为(0,0,1)、(0,1,0)、(1,0,0). 每种顺序的关系矩阵分别为

表 1 标准顺序矩阵

y \ x	—	|	/	\	(	)	⌣	⌢	○	·
—	0	1	1	1	0	0	0	0	1	1
|	0	0	1	1	0	0	0	0	1	1
/	0	0	0	1	0	0	0	0	1	1
\	0	0	0	0	0	0	0	0	1	1
(	1	1	1	1	0	0	0	0	1	1
)	1	1	1	1	0	0	0	0	1	1
⌣	1	1	1	1	0	0	0	0	1	1
⌢	1	1	1	1	0	0	0	0	1	1
○	0	0	0	0	0	0	0	0	0	0
·	0	0	0	0	0	0	0	0	0	0

$$R_1=\begin{array}{c}0\quad\ 0\quad\ 1\\ \begin{pmatrix}-1 & ma & 22\\ md & -1 & 22\\ 22 & 22 & -1\end{pmatrix}\end{array}\begin{array}{c}\\0\\0\\1\end{array},\quad R_2=\begin{array}{c}0\quad\ 1\quad\ 0\\ \begin{pmatrix}-1 & 22 & ma\\ 22 & -1 & 22\\ md & 22 & -1\end{pmatrix}\end{array}\begin{array}{c}\\0\\1\\0\end{array},\quad R_3=\begin{array}{c}1\quad\ 0\quad\ 0\\ \begin{pmatrix}-1 & 22 & 22\\ 22 & -1 & ma\\ 22 & md & -1\end{pmatrix}\end{array}\begin{array}{c}\\1\\0\\0\end{array}$$

相应的顺序矩阵分别为

$$S_1=\begin{array}{c}0\ \ 0\ \ 1\\ \begin{pmatrix}0 & 0 & 1\\ 0 & 0 & 1\\ 0 & 0 & 0\end{pmatrix}\end{array}\begin{array}{c}\\0\\0\\1\end{array},\quad S_2=\begin{array}{c}0\ \ 1\ \ 0\\ \begin{pmatrix}0 & 1 & 0\\ 0 & 0 & 0\\ 0 & 1 & 0\end{pmatrix}\end{array}\begin{array}{c}\\0\\1\\0\end{array},\quad S_3=\begin{array}{c}1\ \ 0\ \ 0\\ \begin{pmatrix}0 & 0 & 0\\ 1 & 0 & 0\\ 1 & 0 & 0\end{pmatrix}\end{array}\begin{array}{c}\\1\\0\\0\end{array}$$

每行中各元素之和的大小为该行代表的基元靠前的可能性大小，因此将上述 S_1、S_2、S_3 中每行元素相加得：

$$S_1=\begin{pmatrix}1\\1\\0\end{pmatrix}\begin{array}{c}0\\0\\1\end{array},\quad S_2=\begin{pmatrix}1\\0\\1\end{pmatrix}\begin{array}{c}0\\1\\0\end{array},\quad S_3=\begin{pmatrix}0\\1\\1\end{pmatrix}\begin{array}{c}1\\0\\0\end{array}$$

在按从大到小的顺序排列得：

$$S_1=\begin{pmatrix}1\\1\\0\end{pmatrix}\begin{array}{c}0\\0\\1\end{array},\quad S_2=\begin{pmatrix}1\\1\\0\end{pmatrix}\begin{array}{c}0\\0\\1\end{array},\quad S_3=\begin{pmatrix}1\\1\\0\end{pmatrix}\begin{array}{c}0\\0\\1\end{array}$$

排序时，基元矢量、关系矩阵均要作相应的变换。排序后，基元矢量均为(0,0,1)，位置关系矩阵均为 R_1.

4 字符粗分类

为减少查找匹配次数，提高匹配速度，我们先对字符进行粗分类，以再次减少用于查找的模板数量。将 94 个字符分为 5 类：点类、圈类、弧类、横类和竖类。每类平均 20 个字符左右，

每类摸板数量在 100 多个。

4.1 圈类

如果基元中有圈基元，即基元矢量中含有元素 5，则认为该字符属于圈类。圈类字符有：

° ∂ ⊕ ⊗ ∮ ∯ ∰ ∅ ∞ ∝ % △

4.2 点类

如果字符不属于圈类，而每个基元的长度均小于某个阈值，则该字符属于点类。点类字符有：

. ′ ″ ‴ * ∵ ∴ : … ⋮ ⋰ ⋱

4.3 弧类

如果字符既不属于圈类也不属于点类，而有弧基元，且同种非弧基元至多一个，则该字符属于弧类。弧类字符有：

∼ ≃ ≈ ≅ ∫ ∬ ∭ ∈ ∉ ⊂ ⊄ ⊃ ∪ ∩ ⊆ ⊇ ⌣ ⌢ ( ) { }

4.4 横类与竖类

如果字符不属于圈类、点类、弧类，则看其较长的基元，如果较长的基元中有横基元，则属于横类，否则属于竖类。

横类字符有：

+ − ± ∓ ÷ − < ≤ ≮ ≯ ≥ > = ≠ ≡ ≈ ≅ ▯ → ← ↔ ⇒ ⇏ ⇔ ↦ △ ∠ ∟ ⊥

竖类字符有：

× / √ \ < ≤ ≥ ≮ > ≯ ∀ ∧ ∨ ↕ ▽ △ ∠ ∟ ⊥ | ‖ ↑ ↓ ↕ ⇑ ⇓ ⇕ [ ]

5 字符匹配

字符匹配是在给定的模板字符集中寻找与手写字符最相似的字符。字符是用一系列的特征表示的，设 $A_k=(a_1^k,a_2^k,\cdots,a_m^k)$ 为模板字符的特征向量，$B=(b_1,b_2,\cdots,b_n)$ 为手写字符的特征向量，w_{ij} 为基元 a_i 与 b_j 配对的权值，所有这种在一定条件下进行的配对权值的总和作为字符 A 与 B 的匹配值，如果模板字符 A_0 与手写字符 B 的匹配值最小，则 A_0 代表的字符为所识字符。

设 x_{ij} 表示基元 a_i^k 与基元 b_j 配对，

$$x_{ij}=\begin{cases}1 & a_i^k \text{与} b_j \text{匹配}\\ 0 & a_i^k \text{与} b_j \text{不匹配}\end{cases}$$

w_{ij} 表示基元 a_i^k 与基元 b_j 配对的权值，则模板字符 A_k 与手写字符 B 的匹配值为

$$d_k=\min\sum_{j=1}^{n}\sum_{i=1}^{m}w_{ij}x_{ij} \tag{1}$$

约束条件为

$$\sum_{i=1}^{m}x_{ij}=1 \qquad j=1,2,\cdots,n \tag{2}$$

$$\sum_{j=1}^{n}x_{ij}=1 \qquad i=1,2,\cdots,m \tag{3}$$

$$x_{ij} \in \{0,1\} \tag{4}$$

式(1)～(4)组成一个整数规划问题，且是一个 0－1 规划。由于字符匹配的特殊性，除考虑匹配值外，还应考虑不匹配值。如果不匹配值较大，即使匹配值较小，也认为两字符不相同。设 λ_i 为 A_k 中元素 a_i^k 不与 B 中元素匹配的权值，μ_j 为 B 中元素 b_j 不与 A_k 中任何元素匹配的权值。于是考虑了不匹配值的 0－1 规划问题为：

$$\begin{aligned}\bar{d}_k &= \min \sum_{j=1}^{n}\sum_{i=1}^{m} w_{ij}x_{ij} + \sum_{i=1}^{m}\Big(1-\sum_{j=1}^{n}x_{ij}\Big)\lambda_i + \sum_{j=1}^{n}\Big(1-\sum_{i=1}^{m}x_{ij}\Big)\mu_j \\ &= \min \sum_{j=1}^{n}\sum_{i=1}^{m}(w_{ij}-\lambda_i-\mu_j)x_{ij} + \sum_{i=1}^{m}\lambda_i + \sum_{j=1}^{n}\mu_j\end{aligned}$$

而 $\sum_{i=1}^{m}\lambda_i$、$\sum_{j=1}^{n}\mu_j$ 为与 x_{ij} 无关的常量，因此取

$$\bar{d}_k = \min \sum_{j=1}^{n}\sum_{i=1}^{m}(w_{ij}-\lambda_i-\mu_j)x_{ij} \tag{5}$$

约束条件仍为(1)～(4).

该问题的难点是对权值 w_{ij}、λ_i、μ_j 给出恰当的定义。w_{ij} 的值应是 a_i^k 与 b_j 相似程度的量度，a_i^k 与 b_j 越相似，则 w_{ij} 越小。定义加权矩阵 W 如表 2.

不匹配权值 λ_i、μ_j 的大小则应由基元的长度来确定：长度小的，丢掉影响小；长度大的，丢掉影响大。因此定义：

$$\lambda_i = l_{a_i^k} \qquad \mu_j = l_{b_j} \tag{6}$$

在数学符号中，点与圈具有特殊的意义，因此长度极小且孤立的基元不能丢掉、圈更不能丢掉，所以如果 a_i^k 为点，则 $\lambda_i = 999$；若 b_j 为圈，则 $\mu_j = 999$.

表 2　标准权阵 W

y	x									
	—	|	/	\	(	)	⌣	⌢	○	·
—	0	9	2	1	9	9	1	1	9	3
|	9	0	1	2	1	1	9	9	9	3
/	2	1	0	3	1	1	3	3	9	9
\	1	2	3	0	2	2	2	2	9	1
(	9	1	1	2	0	2	2	2	9	9
)	9	1	1	2	2	0	2	2	9	9
⌣	1	9	3	2	2	2	0	2	9	9
⌢	1	9	3	2	2	2	2	0	9	9
○	9	9	9	9	9	9	9	9	0	9
·	3	3	1	9	9	9	9	9	9	0

上述匹配问题，并没有充分考虑到基元之间的几何关系。由于权值的选择极具经验性，因此不能保证获得理想的匹配。为此我们加入基元位置关系约束条件。只考虑相邻基元之间的位置关系。设函数 $f(a_i^t,a_{i+1}^t,b_j,b_{j+1})$ 测试基元 a_i^t 与 a_{i+1}^t 的位置关系是否和基元 b_j 与 b_{j+1} 的位置关系一致。函数的值小于阈值 ε_0 则一致，否则不一致。于是给 0 − 1 规划问题(5)(2)(3)(4) 增加约束条件：

$$f(a_i^t,a_{i+1}^t,b_j,b_{j+1})x_{ij}x_{i+1j+1}=0 \qquad i=1,2,\cdots,m-1,j=1,2,\cdots,n-1 \tag{7}$$

我们采用求最佳匹配的经典的Kuhn-Munkres算法[7]，该算法采用逐步扩大匹配对的办法，求无几何约束的最佳匹配。

设　$G=A^t+B$ 为顶点集，$E(G)$ 为 G 的边，l 为可行顶标。

$$E_l=\{ij\,|\,ij\in E(G),l(i)+l(j)=w_{ij}-\lambda_i-\mu_j\}$$

为 G 的子图的边集，以 E_l 为边的 G 的生成子图的相等子图记为 G_l。

1) 选取初始可行顶标 l 为

$$l(i)=\max_{j\in B}(w_{ij}-\lambda_i-\mu_j),\qquad i\in A^t$$

$$l(j)=0,\qquad j\in B$$

确定 G_l，在 G_l 中选出一个匹配 M.

2) 如果 A^t 中字符被 M 所匹配，则停止，M 即为最佳匹配；否则取 G_l 中未被匹配的顶 u，令 $S=\{u\},T=\Phi$.

3) 若 $N_{G_{l1}}(S)\supset T$，则转(4)；若 $N_{G_{l1}}(S)=T$，取

$$a_l=\min_{i\in S,j\in T}\{l(i)+l(j)-w_{ij}\}$$

$$\bar{l}(v)=\begin{cases}l(v)-a_l, & v\in S\\ l(v)+a_l, & v\in T\\ l(v), & \text{其它}\end{cases}$$

$$l\leftarrow\bar{l},G_l\leftarrow G_{\bar{l}}$$

4) 选 $N_{G(l)}(S)-T$ 中一顶 y，若 y 已被匹配，且 $yz\in M$，则 $S\leftarrow S\cup\{z\},T\leftarrow T\cup\{y\}$，转(3)；

否则，取 G_l 中一个 M 的可增广轨 $P(u,y)$，令 $M\leftarrow M-E(P)$，转(1).

但若考虑几何约束，则上述无约束问题的最佳解不一定是约束问题的可行解。我们提出如下算法来求保持几何约束的最佳匹配。

首先用上述 Kuhn-Munkres 算法求出无约束问题的最佳匹配，若其相邻几何关系不满足，则删去所选择的一个匹配对(把其权值置为 999，一删去该匹配对)，重复上述过程，直到发现能满足几何约束条件的匹配为止。

(1) 设 $d=0,W^d=W$；

(2) 设 $d=d+1,W^d=W^{d-1}$ 用 Kuhn-Munkres 算法求解以 W^d 为权值的 0 − 1 规问题，其最优解为 m。

(3) 对每个匹配对 $(a_i^t,b_j)\in M^d,1<i\leqslant m,1<j\leqslant n$，比较 a_i^t 与 a_{i-1}^t 的位置关系是 r_{ii-1}

否和 b_j 与 b_{j-1} 的位置关系 $\bar{r}_{j,j-1}$ 相同，选差值最小者，计为$(a^{i}_{j_0}, b_{j_0})$，若最大差值小于给定阈值 ε_0，则转(4)，否则令 $uf_i = 999$，转(2).

(4) 对所有 $i \leqslant m, j \leqslant n$，设 $x_{ij} = m^{i}_{j}$.

6　试验结果

在试验中，我们找了 20 个学生，每个学生书写 94 个数学符号 5 次，书写 20 × 94 × 5 共 9400 个数学符号.试验结果为：正确识别率为 92.52%，误识率为 3.03%，拒识率为 4.45%.

参 考 文 献

1 Govindan V K, Shivaprasad A P. Character recognition——areview. Pattern Recognition, 1990, 23: 671～683

2 Sin D G, Ham Y K, Park R H. On-line Recognition of cursive Korean characters using DP Matching and Fuzzy concept, PatternRecognition, 1994, 27: 1605～1620

3 chou S L, Tsai W H. Recognizing handwritten Chinese characters by strokesegment matching using an iteration scheme. Int J Pattern Recog Artif Intell, 1991, 5: 175～197

4 刘迎健，戴汝为. 在线手写汉字识别的字形结构排序法. 自动化学报，1988，(3)：207～214

5 潘保昌. 浮动模板法——一种抽取字符特征的方法. 计算机学报，1983，(6)：469～477

6 李健. 笔式计算机及其输入方法研究：[学位论文]. 重庆：重庆大学光电精密仪器系，1995

7 王树禾. 图论及其算法. 上海：上海科技文献出版社，1985. 118～123

The Basic Element Method of Recogniting Handwritten Mathematical Symbols

Zhao Xuejun　Yu Chuzhong　Yang Bo　Pan Baochang

(Laboratory of Artificial Vision, Chongqing University)

ABSTRACT　An on-line recognition approach for the handwritten mathematical symbols is presented. Firstly, it analyses the structures of 94 commonly-used mathematical symbols and concludes that all of them consist of 10 basic elements. Secondly, it proposes a new method of basic element ordering and reduces the number of standard symbols by extracting three primary features of mathematical symbols, namely, basic element vector, relative positions between basic elements, and basic element length vector. Finally, the traditional dynamic programming method is improved by considering matching and unmatching value and adding geometric restraints. Through improved Kohn-Munkres algorithm. During the recognition test of mathematical symbols handwritten by 20 subjects, correctness rate reaches 92.52%, incorrectness rate 3.03% and refusal rate 4.45%.

KEYWORDS　recognition; mathematical symbol; feature extration; character match