

图片中印刷体数学公式的自动识别

数学与计算科学学院 陈颂光

指导老师：黎培兴副教授

摘要： 未能识别数学公式妨碍了科技文献的电子化，故有必要研究数学公式识别技术。数学公式识别可分为符号识别和结构分析两个主要步骤。在符号识别中，除横线和圆点用经验规则判断外，大部分字形可以采用粗分类、细分类到模板匹配的流程，接着利用位置和识别结果合并分体符号的各个字形，对于特殊符号则可用动态生成模版方法匹配。在结构分析中，利用自底向上从局部到整体逐步进行合并的方法，已经支持角标、帽子、分式、根式、矩阵、多行表达式等主要的数学公式结构，并可继续扩充。实验表明此法可以对于高质量图片给出不错的识别效果。

关键词： 数学公式识别 字符识别 结构分析

一 引言

现存科技文献中的大量数学公式，保存于不便于再次利用的形式，导致了很多繁杂且容易出错的重复输入工作，为了整合和盘活数学公式资源，有必要建立一种有效机制把现存的数学公式转换为一种统一、便于重用的形式。这样将节省重复输入数学公式的繁琐工作，同时可为数学公式的搜索和相应的进一步处理提供可能的基础，这对科技文献电子化有重要意义。

在输入方面，既然纸质文档和电子文档一般分别可以通过扫描和格式转换方法转换为图片，不失一般性，可以集中考虑识别以图片为载体的数学公式。同时，由于手写数学公式（特别是脱机的）有时连人眼都不能有把握地辨认，而且当前的实际需要主要是处理较成熟的材料，故进一步把问题限制为印刷体数学公式识别。在输出方面，识别结果应为简单、规范且通用的数学公式表示， \LaTeX 排版语言是当前一个合适的选择。

虽然数学公式识别这个问题至少可以追溯到 1968 年 [1]，国内外过往也发表过不少相关论文，留下了很多创造性的想法。然而，现在专门的数学公式识别系统极为罕见，大多数主流的文档识别系

统也未有对数学公式的专门支持，就作者所知声称有这功能的文档识别系统只有 InftyReader[2] 和赛酷文档秘书 [3]。这说明进行数学公式识别系统的设计与实现工作仍是挑战与机遇并存的。

数学公式识别大致有两个主要组成部分：符号识别和结构分析 [4]。其中符号识别即识别出公式中各个组成符号是什么和在什么位置，而结构识别即利用符号信息重构数学公式。

符号识别方面，由于数学公式中存在更多形状高度相似的符号，一些符号还可以局部延长（例如根号和矩阵的定界符），直接应用常规字符识别技术于数学符号的效果不是很理想，据说识别率会至少下降 5 至 10 个百分点左右 [4, 5]。数学符号的识别较少使用结构方法，多使用统计方法 [6]。统计方法大致分为特征提取和分类两步，其中可用于分类的特征包括模板 [7]、图象变换的系数 [7]、投影直方图 [7]、规范化模板 [7]、几何矩 [7]、Zernike 矩 [7]、高宽比 [1]、穿线数 [1] 和孔洞数 [1]，而常用分类器包括最近邻分类器 [7]、人工神经网络 [7] 和支持向量机 [8]。

在结构分析方面，按分析方向可分为自顶向下方法、自底向上方法和双向方法。按分析手段，基于文法的分析通过建立数学公式的文法描述，利用文法引导结构分析的进行，可用的文法有多种 [6]，但建立一种包容一切数学公式的文法是不切实际的；基于结构的分析则基于各符号的大小和相对位置等几何信息确定公式的结构，具体方法有通过对符号分类后分别应用相应的合并规则 [1]、递归地向两轴投影进行切分 [9]、为符号间可能的连接赋予权值然后应用最小生成树算法 [10]、估计基线结构 [11] 等等。

二 图形预处理

待识别图片由于原始图形自身的瑕疵和输入过程中产生的失真，质量往往并不理想。为了准确地还原数学公式，有必要使用数字图像处理技术进行预处理，以尽可能排除因墨点、纸张纹理、渗透、水印、光照、纸张弯曲等原因造成的干扰，为后续识别过程创造良好的条件。由于要处理的是单个公式图片而非整个页面，这里暂不考虑倾斜校正和卷曲校正问题。

（一）灰度化

由于符号的颜色不是符号识别关心的，故可以把彩色输入图片转换为灰度图象以节省空间和简化处理，[12] 建议了以下公式：

$$Y = 0.309R + 0.609G + 0.082B$$

其中 R、G、B、Y 分别为红、绿、蓝、灰度值。在计算机实现时可以近似为：

$$Y = \frac{316R + 624G + 84B}{1024}$$

当图片有透明度数据时，把全透明视为白色，记透明度为 A，有以下公式：

$$Y = 255 - (255 - \frac{316R + 624G + 84B}{1024}) \times \frac{A}{255}$$

。

(二) 滤波

对图像进行滤波可以消除某些干扰，以下分别列出适用于灰度图像的两种最简单的线性和非线性滤波方法：

- 均值滤波

对每个非边缘像素，以其邻域（例如 8-连通邻域）中像素的平均灰度代替其灰度，由此对图像进行平滑化。均值滤波有助减低随机噪声，但同时使图象变得模糊 [13]。

- 中值滤波

对每个非边缘像素，以其邻域（例如 8-连通邻域）中像素的灰度中位数代替其灰度，由此去除图象中的孤立点。中值滤波对细节保持较好，但可能破坏图象的连通性 [13]。

此外，还有针对二值图形的滤波器，例如 kFill 滤波器 [14]。

(三) 二值化

把灰度图形转换为二值图形有助简化后续处理，并且为一些特征的提取提供可能。以下给出几个有代表性的方法：

- 固定阈值法

固定阈值法即预先手动设定一个阈值，像素灰度小于它时被认为是前景像素，否则为背景像素。这种方法计算速度快，但阈值选择常失之武断。除非有对待识别图象的灰度分布的知识，否则对不同图片的适应性将很差。

- Otsu 方法 [12]

Otsu 方法是一种全局阈值化方法，其基本想法为选一个全局阈值把像素分为两类使类间方差

最大。更准确地,对每个可能灰度值 t (通常为从 0 到 255 的整数),记灰度值小于 t 的像素个数为 $n_1^{(t)}$,平均灰度为 $\mu_1^{(t)}$,灰度值大于等于 t 的像素个数为 $n_2^{(t)}$,平均灰度为 $\mu_2^{(t)}$,总像素数为 $n = n_1^{(t)} + n_2^{(t)}$,总平均灰度为 μ ,则类间方差定义为 $\sigma_t^2 = \frac{n_1^{(t)}}{n}(\mu_1^{(t)} - \mu)^2 + \frac{n_2^{(t)}}{n}(\mu_2^{(t)} - \mu)^2$,选取 t 使 σ_t^2 最大,把像素按灰度值是否小于 t 分为两类,数目较大的一类被认为是背景像素,另一类被认为是前景像素。因为背景像素数量通常比较大,这种基于直方图的方法容易处理黑底白字的情况。

- Sauvola 方法 [16]

Sauvola 方法是一种局部阈值化方法,其基本想法为以一个像素为中心的窗口中的像素的平均值和标准差计算阈值。更准确地,对 (x, y) 处像素,记以之为心边长为 ω 的方邻域中平均灰度为 $m(x, y)$ 、灰度标准差为 $s(x, y)$,则该像素阈值为 $t(x, y) = m(x, y)(1 + k(\frac{s(x, y)}{128} - 1))$ (其中 k 为一个 $[0.2, 0.5]$ 中常数),该像素灰度小于 $t(x, y)$ 时被认为是前景像素,否则为背景像素。虽然可以用积分图象加速计算 [17],但计算量仍明显比全局阈值算法大,不过由于可以克服光照不均等问题有时也是值得的。

三 数学符号识别

数学符号作为数学公式的基本元素,准确地识别是对公式作为整体进行识别的基础。数学符号的识别问题基本上是被广泛研究的字符识别问题的特例,可以借用字符识别的方法,但同时应充分利用数学符号的特点以提高识别率,并顾及对后续处理的影响。

(一) 字形分割

本文系统对预处理后的二值图象进行 8 连通域分割,得到使用游程编码的各 8 连通域,事实上很多特征使用游程编码计算是方便的,而且使用游程编码有压缩效果。为了修复明显的断裂和减低待匹配对象数,把最小外接矩形明显相交的连通域合并(但不能把疑似根号与其它连通域合并)为字形,作为进行匹配的单位。连通域分割算法主要步骤如下:

1. 按行优先扫描二值图象,对每个前景像素游程:

- 若该前景像素游程上方、左上方或右上方邻近有前景像素游程,记它们所在连通域编号有 i_1, \dots, i_s ,则把此游程加入编号 i_1 的连通域,再在一个表中加入记录 $\{i_1, i_2\}, \dots, \{i_1, i_s\}$;
- 否则,创建一个新的连通域(予以编号)并把此游程加到其中。

2. 把产生的连通域编号看作无向图的顶点，而把产生的表中记录看作无向图的边，然后求无向图的连通分支，再分别把各连通分支中各个顶点对应的连通域分别为字形。

(二) 字形识别

1 常规字形识别

在字形识别过程中，用字形数据库中所有字形作为初始的候选字形集合，然后依次使用多个分类器进行匹配以逐步减少候选字形集中字形数，最后对仅有的少数余下的候选进行模板匹配。其基本流程如下：

1. 粗分类（可选）。通过利用字形相对稳定的特征进行筛选，从而缩小候选集，提高识别速度。

可以用来粗分类的特征包括：

- 孔洞数

孔洞数即字形中背景像素的连通域个数（不算外围的），这对大多数字形而言很稳定，但仍然可能由于噪声或二值化不当而有不稳定性。

- 高宽比

高宽比即字形外接矩形的宽度与高度之比，这对大多数字形而言有一定的稳定性，而且特别易于计算，但字形高宽比悬殊时对尺度变换敏感，另外确实存在高宽比可变的字形（这类字形将在本节稍后处理）。

2. 基于距离的判别。通过与候选集中所有字形计算某种距离，去除对应较大距离的候选，逐步缩小候选集。以下列出一些可用于计算距离的特征：

- 网格特征

把像素矩阵通过网格分为 $m \times n$ 份（在本系统中取 $m = n = 3$ ），对 $i = 1, \dots, m; j = 1, \dots, n$ ，记格子 (i, j) 中前景像素个数为 N_{ij} ，而面积为 A_{ij} ，则前景像素密度为 $d_{ij} = \frac{N_{ij}}{A_{ij}}$ ，归一化后的前景像素密度组成网格特征矩阵：

$$\left(\frac{d_{ij}}{\sum_{r=1}^m \sum_{s=1}^n d_{rs}} \right)_{m \times n}$$

。

- 矩

记前景像素的坐标集合为 R , 则 (i, j) 阶几何矩定义为

$$M_{ij} = \sum_{(x,y) \in R} x^i y^j$$

(i, j) 阶中心矩定义为

$$\mu_{ij} = \sum_{(x,y) \in R} (x - \bar{x})^i (y - \bar{y})^j$$

其中, 重心定义为

$$\bar{x} = \frac{M_{10}}{M_{00}}, \bar{y} = \frac{M_{01}}{M_{00}}$$

(i, j) 阶归一化中心矩定义为

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{\frac{i+j}{2}+1}}$$

归一化中心矩在尺度变换下不变。归一化中心矩适当的函数 (如 Hu 矩) 还可以是在旋转、缩放变换下的不变量 [12]。

- 投影

投影包括横向和纵向的前景像素数分布 (其实也可考虑其它方向)。更准确地, 记像素矩阵为 $(a_{ij})_{m \times n}$, 则横向投影为 $(\sum_{k=1}^n a_{ik})_{m \times 1}$, 纵向投影为 $(\sum_{k=1}^m a_{kj})_{1 \times n}$ 。由于投影为有限维向量, 在投影维数相同时容易定义各种距离, 在投影维数不同时也容易通过分段线性插值处理。通过把二维数据化为一维数据, 可简化处理, 但无法区分一些具对称性的字形。

3. 计算相似系数

进行模版匹配, 即把待识别连通域的像素矩阵与数据库中字形的像素矩阵作对比, 计算它们间的 Hausdorff 距离 [1], 利用它的一个单调减函数作为匹配的相似系数。虽然模版匹配计算量较大, 但能全面利用信息。

这种分层方法便于实现和扩充, 有较高的运行效率, 而且容易为人类理解, 不要求掌握复杂的机器学习背景。

2 特殊字形识别

对于高宽比悬殊的横线和细小的圆点, 由于离散化造成的误差是不可接受的, 为此设计了用于判定横线和圆点的经验规则:

- 若一个连通域的宽度为其高度的五倍或以上且在其外接矩形中像素密度达到 90%，则被认为是横线。
- 若一个连通域的高宽比在 $\frac{4}{5}$ 和 $\frac{5}{4}$ 之间且在其外接矩形中像素密度达到 70%^①，则被认为是圆点。

另外，部分符号如定界符、根号、箭头和水平括号并不总能由字体模板通过缩放得到，还涉及局部延长，这些符号包括定界符（如 $|$ 、 $()$ 、 $[]$ 、 $\{ \}$ 、 $\lfloor \rfloor$ 、 $\lceil \rceil$ ）、根号（ $\sqrt{\quad}$ ）、箭头（ \rightarrow ）和水平括号（如 \frown 、 \smile ），这时基于网格特征、矩和投影等统计特征的匹配效果会变差。针对这些特殊符号，可以采用动态生成模板方法，即根据待识别字形的特点（主要是高宽比）生成各个特殊字符的模板，然后让待识别字形与所生成模板按常规方式进行匹配。应该指出，不用对每个字形进行特殊符号检测，可以只对常规识别匹配效果不佳的字形进行。

（三） 字形合并

在进行字形识别后，需要利用位置和识别结果合并分体符号的各个字形。更准确地，对属分体符号的一部分的候选字形，检查该分体符号其它字形应在的位置是否有字形，如全部出现，则计算候选符号的像素矩阵和各个字形合并所得像素矩阵间的 Hausdorff 距离以代替候选的距离，再删去其它对应字形中相应于该分体字的候选；如不全出现，删去此候选。这样，系统便得到识别出的符号及其位置与大小。如有需要，还可以生成多候选，以便把一些困难的选择留待结构分析阶段处理。

此外，鉴于省略号（包括 \dots 、 $\dot{\cdot}$ 和 $\ddot{\cdot}$ ）并未作为符号包含进常见的数学字体，而人手编辑字体取得效果也不理想，故采用了经验规则合并圆点以产生省略号。

四 数学公式结构分析

结构分析是在符号识别的基础上根据符号间位置关系重组出数学公式的结构，从而生成对应的 L^AT_EX 代码。受到 T_EX 排版系统使用盒子作为排版单位 [18] 启发，本文系统的结构分析基本上采用自底向上方法，过程为先让每一个符号分别用一个盒子表示，然后逐次选择一些有特定位置关系的盒子合并为新的盒子，直至仅余下一个盒子。

^①注意一个圆与其外接正方形的面积比为 $\frac{\pi}{4} \approx 0.7854$

(一) 数据结构

盒子这个数据结构被用于表示子公式，结构分析算法将在盒子集合上施行，盒子的组成如下：

- 排版代码
生成该盒子所表示子公式所需的 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 代码（对于其它目的也可能是 MathML 等代码）。
- 基准点位置
基准点为标示盒子所表示子公式位置的点，其纵坐标为子公式的基线估计。
- 逻辑高度和逻辑宽度
子公式外接逻辑边框的高度和宽度。
- 上升和左偏移
基准点纵坐标与逻辑上边界纵坐标之差和基准点横坐标与逻辑左边界横坐标之差。
- 参照大小
用于估计子公式的相对层级的一个量。
- 基线确定与否
标记基线位置的估计否可靠。

使用面向对象方法时，不同的子公式结构可以用不同的盒子类型来表示，盒子类型可以包括但不限于^①：

- 符号型
用于表示由单个符号组成的子公式。
- 同行型
用于表示子公式通过水平排列（容许上下标）组成的子公式。
- 分式型
用于表示由分子、分母和分数线组成的子公式。
- 根式型
用于表示由被开方子公式、根号和可选的次数组成的子公式。

^①在需要时还可通过增加盒子类型支持更多样的子公式结构

- 帽子型
用于表示由帽子与其它子公式组成的子公式。
- 大型操作符型
由于表示大型操作符和上下限组成的子公式。
- 多行型
用于表示由多个行组成的子公式。

对于符号型盒子，其各个参数利用符号的像素边界和符号数据库中数据推算；而对于其它类型的盒子，其各个参数利用各组成盒子的参数推算。

(二) 算法

合并算法框架如下：

1. 初始化

对于每个符号，创建一个符号型盒子。

2. 合并

(a) 合并基线确定且一致的盒子

如有两相异盒子基线一致、水平距离较小且它们共同最小外接矩形不与其它盒子相交，则两盒子被认为是同行相邻两部分而合并，按各自基准点横坐标决定左右，然后回到合并步继续。

(b) 合并基线确定的上下标

如有两相异盒子基线不一致但差异在一定范围内、水平距离较小且它们共同最小外接矩形不与多于一个其它盒子相交，则一盒子被认为是另一盒子的上标或下标而合并，按各自基准点横坐标决定左右，按各自参照大小和基线决定上下，然后回到合并步继续。

(c) 合并基线不确定的行内盒子

与第 (a)、(b) 步类似

(d) 合并特殊符号

i. 帽子合并

对于上划线、下划线和向量箭头，如所管辖区恰有一个别的盒子，则把有关盒子合

并；对于上花括号和下花括号，如果主管辖区恰有一个其它盒子，副辖区至多有一个其它盒子，则把有关盒子合并，然后回到合并步继续。

ii. 分式合并

若分子部分和分母部分分别恰有一个别的盒子，则把有关盒子合并，然后回到合并步继续。

iii. 根号合并

若根号内部恰有一个盒子而次数位置至多有一个盒子，则把有关盒子合并，然后回到合并步继续。

iv. 大型操作符合并

若下标域恰有一个盒子而上标域至多有一个盒子，则把有关盒子合并，然后回到合并步继续。

(e) 合并行

如有两相异盒子垂直距离较小且它们共同最小外接矩形不与其它盒子相交，则两盒子被认为是相邻两行而合并，按各自基准线决定上下，然后回到合并步继续。

3. 如只剩下一个盒子，算法结束并得到整个数学公式的识别结果。否则，得到数学公式各个组成部分的识别结果。

作为一种改进，对于算法结束时剩下多于一个盒子的情况，可以利用回溯技巧以尝试消除错误的合并而继续算法。

应该指出，结构分析的输入不一定依赖于上节的符号识别技术，例如原则上可把本节方法应用于通过解析 PostScript 或 PDF 文档得到符号识别结果，以支持从电子文档不转换为图片而识别其中的数学公式。

五 实验结果

前述数学公式识别方案已获数学公式识别系统 MathOCR[19] 实现，它是一个以 Java 语言 [20] 编写的自由软件，可以在 GNU 通用公共许可证 (GPL) 版本 3 或 (按你的意愿) 更新版本有关条件下自由使用、研究、修改和散布。以下的实验将以它为基础。

(一) 符号识别实验

利用的 AMSFonts[21] 字体中的符号作为训练集和测试集, 包括 CMB10、CMBSY10、CMEX10、CMMI10、CMMIB10、CMR10、CMSY10、MSAM10、MSBM10、RSFS10, 它们是常用的数学字体并覆盖了大部分常用的数学符号 (常用缺失符号如 \neq 被人手加进字体中)。由于 Java 能比较好地处理 TrueType 字体, 先把它们用 FontForge[22] 转换为 TTF 文件, 并生成字体列表。然后, 手动把字体列表中符号的名字改为在 L^AT_EX 数学模式中生成它们的相应命令 (另外, 对于帽子、大型操作符和不完整符号, 分别加特殊标记)。接着, 用 MathOCR 的“字体训练”功能生成数据文件和测试图片, 其中训练字体大小为 40。最后, 利用一个自动化测试程序生成完美孤立符号图片供识别并统计第一候选给出正确识别结果的比例, 其中二值化方法选为 Sauvola 方法, 粗分类方法为高宽比, 距离判别器使用网格特征和投影。对于字母, 由于字体常影响含义, 必须连字体也正确识别才算正确识别; 而对于非字母的符号, 并不区分字体。

实验结果如表1所示, 对于字体大小 40 (这恰为训练样本的字体大小), 识别率还是很不错的, 事实上这时进一步分析被误识的符号发现, 一些误识出现在减号、上划线、破折号之间, 而余下的出现在小数点、乘号点、求导点之间, 这两组符号内部的符号几乎是本质上无法仅用形状区分的。不过, 随着字体大小下降, 识别率明显下降, 但仍明显优于瞎猜。一方面, 必须承认, 在未有外加噪声情况下这个识别率有很大的改进空间。另一方面, 就识别电子文档的用途而言, 由于电子文档可缩放, 字体较大的要求不难满足。通过在结构分析阶段用基线和大小匹配方法进行纠正, 还有望进一步提高符号识别的准确率。

(二) 结构分析实验

为了验证结构分析算法的准确性, 用 MathOCR 对由两本数学公式排版教程中数学公式转换得到的图片进行识别, 其中数学公式种类多样, 用人手判断识别结果是否准确。实验结果如表2所示^①, 可见对于各类数学公式均有一定的识别率, 但对于实用目的而言仍是很不足的。其中, 导致出错的主要原因包括上下标关系误判、存在符号识别错误和存在不支持的数学公式结构。

^①部分公式同时属于多于一种类型

表 1: 符号识别的识别率

字体	大小			
	40	30	20	10
CMB10	123/125	106/125	97/125	34/125
CMBSY10	126/126	120/126	104/126	55/126
CMEX10	95/95	82/95	77/95	52/95
CMMI10	110/110	98/110	77/110	3/110
CMMIB10	111/111	99/111	69/111	25/111
CMR10	126/128	99/128	62/128	5/128
CMSY10	125/126	115/126	106/126	29/126
MSAM10	114/114	105/114	93/114	24/114
MSBM10	85/85	80/85	68/85	3/85
RSF10	26/26	25/26	18/26	0/26
总计	1041/1046	929/1046	771/1046	230/1046

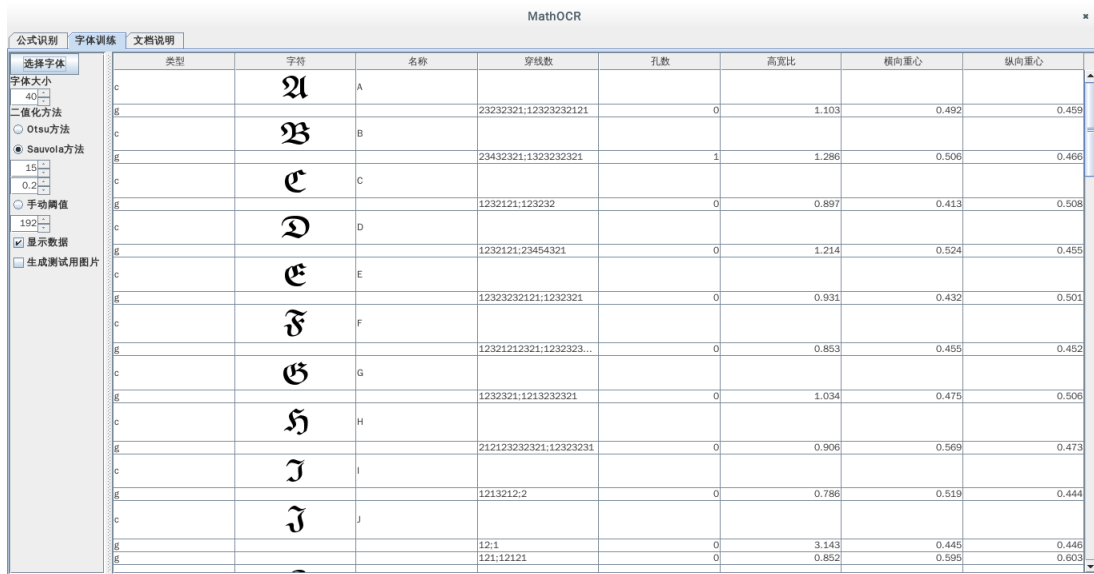
表 2: 结构分析的准确率

类型	样本个数	结构分析准确个数	准确率
简单表达式	39	30	76.9%
含根号表达式	13	11	84.6%
含分式表达式	23	19	82.6%
含帽子表达式	22	15	68.2%
含大型操作符表达式	24	16	66.7%
含矩阵表达式	20	14	70.0%
多行表达式	35	30	85.7%
总计	147	113	76.9%

六 总结和展望

(一) 取得成果

本文最重要的工作在于实现了一个演示性质的数学公式识别系统 MathOCR，并且提供了基本的图形介面（见图1和图2），它容许用户观察并修改关键步骤的结果。



类型	字符	名称	穿线数	孔数	高宽比	横向重心	纵向重心
c	A	A					
g			23232321:12323232121	0	1.103	0.492	0.459
c	B	B					
g			23432321:1323232321	1	1.286	0.506	0.466
c	C	C					
g			1232121:123232	0	0.897	0.413	0.508
c	D	D					
g			1232121:23454321	0	1.214	0.524	0.455
c	E	E					
g			12323232121:1232321	0	0.931	0.432	0.501
c	F	F					
g			12321212321:1232323...	0	0.853	0.455	0.452
c	G	G					
g			1232321:1213232321	0	1.034	0.475	0.506
c	H	H					
g			212123232321:12323231	0	0.906	0.569	0.473
c	I	I					
g			1213212:2	0	0.786	0.519	0.444
c	J	J					
g			12:1	0	3.143	0.445	0.446
g			121:12121	0	0.852	0.595	0.603

图 1: MathOCR 中训练字体的图形界面

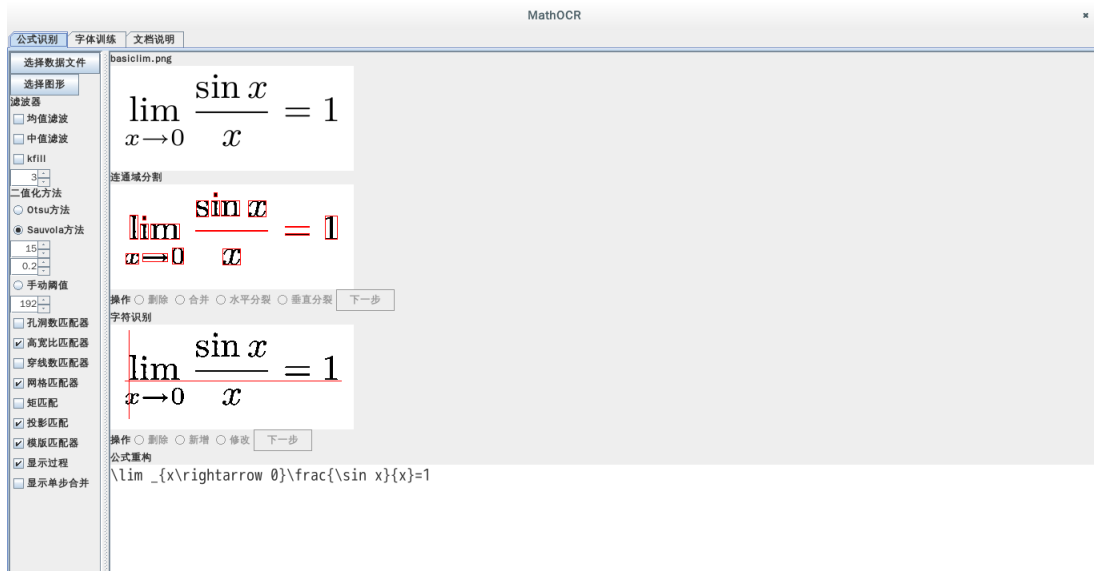


图 2: MathOCR 中公式识别的图形界面

它的设计虽然主要为组合传统技术，但也有以下的特点：

- 尽可能避免进行归一化以减少因离散化造成的失真；
- 使用动态生成模板方法把特殊字形识别问题化为常规字形识别问题；
- 采用了对基线、逻辑高度、逻辑宽度的估计；
- 避免作人为的假定，未有显式地使用任何统计模型或形式文法。

(二) 后续工作

虽然已经建立了一个令人鼓舞的公式识别系统，但它在多方面仍有待改进。为了进一步提高公式的识别率，以下是一些想法：

- 更细致的预处理

对待识别图形的预处理对识别效果有一定影响，因此有必要在预处理上做更多工作，例如加入倾斜和其它变形的检测与校正、采用更有效的噪声去除手段。

- 加入对粘连和断裂字形的处理

本文系统仅使用连通域分割作字形分割，但这不能处理粘连和断裂字形。可能的解决方法包括对未能有效识别的连通域寻找弱连结节点切分或与邻近连通域合并。

- 对公式中汉字的识别

有时候数学公式中存在描述性的文字，因而数学公式识别系统也应该把它们识别出来。虽然原则上容许直接加入任何语言的字体作为符号识别的学习集，但对于汉字这样庞大的字符集，这样做并不能取得令人满意的准确率和速度。因此，有必要针对这问题提出解决方法。一个可能方法是把匹配效果不佳的字形经聚类后提交专门的汉字识别系统处理。

- 对交换图的识别

本文系统尚不能正确识别交换图，而交换图在不少数学文献中出现，因此识别交换图也是有意义的。一个可能的解决方案为把交换图作为矩阵识别。

- 识别结果的自动验证

既然难以保证识别结果完全正确，而在应用场合却要求结果完全正确，需要设法验证识别结果是否正确，这个步骤也应该自动化，尽可能减少要求人手干预的次数。

- 识别结果修正

通过建立常见的误识列表，可望对识别结果进行修正。

(三) 应用前景

以数学公式识别系统为基础，可以构建多种有意义的应用：

- 科技文献电子化

当前文献多以纸质提供，运输和保存成本较高，限制了广泛流通。在取得适当版权许可后，通过把它们转换为电子格式可望大大降低阅读成本，促进科技文化传播。扫描方式可以容易地把文献转换为人眼可识别的图片，但图片版仍不便于进行其它处理且体积很大，有必要把内容识别出来，转换为便于利用的格式。只有把数学公式识别技术整合进传统的字符识别技术和文档分析技术，识别系统才可能对含有数学公式的大量文献作可接受的识别，其中把公式从上下文提取出来是整合的一个难点。

- 数学公式检索

在科技文献中，数学公式往往是重要组成部分，因而在文献检索中允许以数学公式将带来巨大的便利。利用公式识别系统，可以把现存各种文献中的公式转换为排版语言，对后者更容易使用现有形式语言和自然语言处理技术处理。尚需解决的问题还有排版代码统一化（多种排版代码可能描述同一数学公式）、变量名称统一化（变量名称一般不影响公式含义），更智能的系统也许可以结合计算机代数系统把仅差一个简单变形的公式视为是相似的。

此外，数学公式识别系统中所用技术还可能用于解决其它问题：

- 化学公式识别

在不少科技文献中，化学公式也是重要组成部分，因而化学公式识别有与数学公式识别相类似的地位。化学公式与数学公式同样是二维结构，它们的识别难点有共同之处，因而数学公式识别技术有可供借鉴的地方。当然，化学公式与数学公式也有不少差异，例如与数学公式相比，化学公式层次一般较少，但有机分子的结构式具有数学公式所不常具备的连接结构。

参考文献

[1] 王科俊, 冯伟兴. 中文印刷体文档识别技术 [M]. 北京: 科学出版社, 2010.

[2] Science Accessibility Net. InftyReader-Top Page[EB/OL]. [2014-08-17]. <http://www.sciaccess.net/en/InftyReader/>.

- [3] 赛酷科技有限公司. 赛酷文档秘书 (互联网版)[EB/OL]. [2014-08-17].http://www.saqtech.com.cn/saq_document01.asp.
- [4] 肖敏, 黄磊, 刘迎建. 数学公式识别系统 [C]//第八届全国汉字识别学术会议论文集. 绍兴: 中国中文信息学会基础理论专业委员会, 2002:31-37.
- [5] 靳简明, 江红英. 印刷体数学公式处理研究现状 [C]//2001 年中国智能自动化会议论文集 (上册). 昆明: 中国自动化学会智能自动化专业委员会, 2001:69-74.
- [6] Chan K F, Yeung D Y. Mathematical expression recognition: a survey[R]. Hong Kong: HKUST, 1999.
- [7] Trier Ø D, Taxt T, Jain A K. Feature extraction methods for character recognition - A survey[J]. Pattern Recognition, 1996, 29(4):641-662.
- [8] Malon C, Suzuki M, Uchida S. Support Vector Machines for Mathematical Symbol Recognition[C]// Structural, Syntactic, And Statistical Pattern Recognition, Proceedings. Berlin: SPRINGER-VERLAG, c2006 : 136-144.
- [9] Raja A, Rayner M, Sexton A, Sorge V. Towards a parser for mathematical formula recognition[C]// Mathematical Knowledge Management, Proceedings. Berlin: SPRINGER-VERLAG, c2006 : 139-151.
- [10] Eto Y, Suzuki M. Mathematical formula recognition using virtual link network[C]// Proceedings of Sixth International Conference on Document Analysis & Recognition. Washington: IEEE Computer Society, c2001 : 762-767.
- [11] 李永华, 王科俊, 上官伟, 唐立群. 数学公式基线结构分析及识别算法研究 [J]. 计算机工程与应用, 2008, 44(16):18-26.
- [12] Burger W, Burge M J 著; 黄华译. 数学图像处理: Java 语言算法描述 [M]. 北京: 清华大学出版社, 2010.
- [13] 卢晓卫. 印刷体数学公式识别系统的研究与实现 [D]. 长沙: 国防科学技术大学, 2009.
- [14] O'Gorman L. Image and Document Processing Techniques for the RightPages Electronic Library System[C]// Proceedings of the International Conference on Pattern Recognition. Los Alimitos: IEEE, c1992: 260-263.

- [15] Shapiro L G, Stockman G C 著; 赵清杰, 钱芳, 蔡利栋译. 计算机视觉 [M]. 北京: 机械工业出版社, 2005.
- [16] Sauvola J, Pietikäinen M. Adaptive document image binarization[J]. Pattern Recognition, 2000, 33(2):225-236.
- [17] Shafaita F, Keysersa D, Breuel T M. Efficient Implementation of Local Adaptive Thresholding Techniques Using Integral Images[C]// Proceedings of The International Society for Optical Engineering, Document Recognition and Retrieval XV. San Jose: SPIE-IS&T, 2008:681510–681510–6.
- [18] Knuth D E. The \TeX book[M]. Reading: Addison-Wesley, 1986.
- [19] 陈颂光. MathOCR[CP/DK]. [2014-11-29].<http://mathocr.sourceforge.net/>.
- [20] Open JDK 1.7.0_55[CP/DK]. [2014-5-16].<http://openjdk.java.net/>.
- [21] American Mathematical Society. AMSFonts 3.04 [CP/DK]. [2013/01/14].<http://www.ctan.org/pkg/amsfonts>.
- [22] Williams G. FontForge[CP/DK]. [2012-07-31]. <http://fontforge.org/>.

An Attempt on Printed Mathematical Formula Recognition

Abstract: Since optical formula recognition should be an essential part of a document analysis system but it is missing from most main-stream systems in reality, a practical solution to formula recognition is proposed. Like most existing designs, the system consist of two main parts: symbol recognition and structural analysis. For the character recognition part, the core is a glyph recognizer, coarse classification is followed by fine classification to produce candidates, and then template matching based on Hausdorff distance is being used to verify. Dynamically generated template is used to match special glyph. Empirical rules is also being used to match line and dot. Later on, some glyphs is combined to form symbol according to their recognition result and coordinates. For the structural analysis part, a bottom-up approach is applied. Scripts, fractions, radical expressions, matrices and multi-line expressions are supported, further extension is also possible. An implementation based on ideas presented in this article, MathOCR, is already available as a free software.

Although the system has not yet acquired industrial strength and robustness for daily use, it can produce impressive output using high-quality input.

Key Words: optical mathematical formula recognition; structural analysis; optical character recognition