图片中印刷体数学公式的自动识别

陈颂光 (11336019)
中山大学数学与计算科学院 2011 级数学与应用数学

2014 年 11 月 28 日

摘要

现存科技文献中的大量公式被保存于不便于再次利用的形式，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。这样将节省输入公式的繁琐工作，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。

在回顾国内外相关工作的基础上，本文沿用把数学公式识别系统分为字符识别系统和结构分析系统两个主要部分的基本框架，对其中每个环节均予以讨论并给出可能的解决方案。

在符号识别方面，我们以字形为基本识别单位，因而在识别前进行连通域分割再作简单的合并。对于大多数字形，利用从来自字体文件的信息，构造多个分类器，由粗到细层层筛选以得一个或多个候选；对于横线和圆点，则利用经验规则进行判断。接着，利用字形识别结果和字形间几何位置关系作出可能构成完整符号的合并，再以基于豪斯多夫距离的模版匹配方法决定合并的优劣。此外，对于若干不能由字体模版缩放而得的特殊符号则用动态生成模版的方法处理。

在公式结构分析方面，我们主要采用自底向上方法，从局部到整体逐步进行合并和生成对应的 LaTeX 代码。我们的系统已经支持角标、帽子、分式、根式、矩阵、多行表达式等主要的数学公式结构。

本文设计的系统已经有 Java 语言的实现 MathOCR，它对于高质量的图片取得了值得注意的识别效果。虽然当前仍然处于试验阶段，但经过更多的工作，存在把系统投入到日常工作环境的可能性。

〔**关键词**〕 公式识别; 字符识别; 结构分析

目录

部分 1

背景

1.1 问题

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 TEX 等数学排版工具重用的形式。这样将节省重复输入数学公式的繁琐工作，同时可作为数学公式的搜索和相应的进一步处理可能的基础，这对科技文献电子化有重要意义。

具体地，我们注意到数学公式的不便于再次利用的形式包括：

纸质文献 书籍、期刊、会议论文、学位论文等印刷材料和笔记、草稿等手写材料中存在大量数学公式。这些公式只能通过抄录、手动输入电子计算机或扫描为图片而重用。

电子文档 目前许多在线论文数据库以便携文件格式（简称 PDF）、PostScript（简称 PS）等格式向读者提供学术论文下载，这些格式的电子文档能提供较佳的阅读质量，但其中的数学公式不便于修改且一般不能通过直接复制而加以重用。

网页图片 由于兼容性的考虑，网页中的数学公式往往以图片的形式呈现给读者。然而，这些图片进行缩放后质量没有保证，而且也不便于修改。

既然上述各种数学公式载体都可以转换为图片，只要我们能把图片中数学公式识别出来并转换为便于重用的格式，其它形式数学公式的识别转换问题也就解决了，因此我们将集中研究图片中数学公式的自动识别问题。同时，由于（脱机）手写数学公式有时连人眼都不能有把握地辨认，更不能指望机器能做到，而且当前的实际需要主要是处理较成熟的材料，故我们的工作仅针对印刷体数学公式进行。

在数学公式识别领域中，与通常的字符识别和文档分析相比，面临着更多的困难：

- 由于存在大量数学符号，而且需要区分字体，这使得分类数较大。
- 数学公式中经常存在许多形状极为相似甚至相同的符号（例如分数线、减号、上划线、下划线形状相同），难以区分。
- 数学公式是一种平面结构，符号间可能存在多种位置关系，并且局部误识容易导致全局错误。
- 不同领域、不同作者有不同的符号体系，这限制了通用识别系统中语义信息的使用。

因此，我们有必要针对数学公式设计一个专用的识别系统。

1.2 相关研究的历史与现状

Anderson 在他于 1968 年的博士论文已经提出了数学公式识别问题 [1]，但在其后二十余年仅有少量论文发表 [2]，20 世纪 70 年代的工作集中于建立完备的文法，20 世纪 80 年代的工作则集中于特定类型数学公式的识别 [1]。20 世纪 90 年代起随着文字识别技术的成熟，公式识别的研究热度日益增加，涉及到各个方面 [1]。在国内，数学公式识别的大多数工作在 2000 年后才起步 [1]。

目前的公式识别系统设计通常有两个主要组成部分：符号识别和结构分析 [3]。

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。这方面有很多已经相对成熟并已投入实用多年的字符识别技术可供借鉴。不过，由于数学存在更多形状高度相似的字符，一些符号还可以变形（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 [3, 4]。

符号识别通常以字符为识别单位，但也可以以字形为识别单位而让从字形合并为字符作为结构分析的工作 [5]。在进行符号识别时，需要先进行符号分割得到待识别的单位：以字符为识别单位时常用的手段有进行连通域分割后进行合并，还有基于投影的方法；以字形为识别单位时只用进行连通域分割，每个连通域分别就是待识别的字形。不过，当字形出现断裂或粘连时需要另加处理，这些处理现在仍不完善 [4]。有的系统还会对符号作细化 [1]。

当前数学符号的识别较少使用结构方法，多使用统计方法 [6]。基于统计的符号识别方法的主要步骤如下：

1. 特征提取，即从待识别符号得到对识别目的有区分力的信息。以下是一些有着不同的不变性和可重构性的特征：
   (a) 模板 [7]
   (b) 图象变换的系数 [7]
   (c) 投影直方图 [7]

(d) 规范化模板 [7]
(e) 几何矩 [7]
(f) Zernike 矩 [7]
(g) 高宽比 [1]
(h) 穿线数 [1]
(i) 孔洞数 [1]

2. 分类，即把从待识别符号提取的特征与数据库中各已知符号的特征进行比较以决定待识别符号应被识别为哪个符号。以下是一些可用于识别的分类器：

- 最近邻分类器 [7]
- 人工神经网络 [7]
- 支持向量机 [8]

结构分析即把各个符号组合为用树、图或其它数据结构描述的完整数学公式。这些方法按分析方向可分为：

- 自顶而下方法
  递归地把数学公式分解为子公式，直至得到不用再分解的符号。
- 自底向上方法
  递归地把符号合并，直至得到整个数学公式。
- 双向方法
  分别进行自顶而下的分解和自底向上的分解以处理不同的结构。

按使用的手段则可分为：

- 基于文法的分析
  建立数学公式文法描述，利用文法引导结构分析的进行。这种文法可以是随机上下文无关文法、约束属性文法、结构说明、属性文法、图文法、描述文法等等 [6]。然而，由于数学公式高度多样化并且正变得更多样化，建立一种包容一切数学公式的文法是不切实际的，即使能建立也会因过于复杂而使识别效率很低。
- 基于结构的分析
  基于各字符的大小和相对位置等几何信息确定公式的结构。具体方法有通过递归地向两轴投影进行切分 [5]、对字符分类后分别应用相应的合并规则 [1]、为符号间可能的连接赋予权值然后应用最小生成树算法 [9]、估计基线结构 [10] 等等。

此外，与数学公式识别的研究还包括数学公式提取 [1, 11]、自动性能评估 [6, 11]、错误检测与校正 [6] 等等。

除了图片中印刷体数学公式识别外，数学公式识别还有其它方向，以下仅作简单介绍：

- 文档数学公式识别
  目前也有许多数学公式保存于不便于重用的电子文档格式中，例如便携文件格式（简称 PDF）、PostScript（简称 PS）等格式的文档中，这些文档包含了完整、准确的排版信息，因而一个想法是直接把这些格式的文档中数学公式直接转换为便于重用的格式。[12] 提出解析 PostScript 的命令，通过提取字符的名称、位置、字体识别出数学符号，再利用数学符号间位置关系进行合并以重构数学公式。

- 联机手写数学公式识别
  为方便数学公式输入，使之与用粉笔在黑板书写尽可能接近，在电子设备上书写再进行识别是一个自然的想法。由于有笔画序列信息和可以与用户交互，在字符识别和结构分析都比较容易，目前已经有一些联机手写数学公式识别程序 (例如 [13])。

1.3 现有数学公式识别程序

在国内外已经开发出一些数学公式识别系统并已嵌入到文档识别系统中，以下是一些从事这项工作的团队：

- 赛酷科技
  赛酷科技有限公司官方网站宣称其公式识别 SDK 能识别千种公式结构并有识别结果与图象自动对应校正，在 300dpi 的分辨率下，识别率达到 95%，识别速度达到 3000 字/分钟 [14]。公式识别功能已被整合进产品《赛酷文档秘书》中，但需要用户自行圈选公式区域且还不支持矩阵和行列式识别 [15]。

- Infty 项目
  InftyReader 是一个识别含数学公式印刷体科技文档的软件，支持分辨率为 600dpi 或 400dpi 的输入图片 [16]。文献 [17] 对此软件的设计作出了介绍，并得出数学公式部分字符识别率达到 95.18%，结构分析准确率达 89.6%。

虽然现已存在一些印刷体数学公式识别系统，但由于它们效果仍欠理想且均为私有软件，限制了其广泛使用。目前数学公式识别也仍未成为主流文档识别系统的功能。因此，开发印刷体数学公式识别系统仍是一件有广阔发展空间的工作。

1.4 小结

数学公式识别是一个挑战性伴随着实用性的课题。目前，虽然在国内外已经可以看到不少有关数学公式识别的讨论，但很少看到成品，特别是可以自由使用的。因此，当前进行数学公式识别系统的设计与实现工作是可行而且是有意义的。

部分 2

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、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。

2.1 灰度化

由于字符的颜色不是字符识别关心的，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，[19] 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:

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

2.2 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，由此对图象进行平滑化。均值滤波有助减低随机噪声，但同时使图象变得模糊 [20]。

- 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 [20]。

此外，还有针对二值图形的滤波器，例如 kFill 滤波器 [21]。

2.3 二值化

把灰度图形转换为二值图形有助简化后续处理，并且为基于结构的字符识别方法提供可能。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
  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 $\sigma_t^2 = \frac{n_1^{(t)}}{n}(\mu_1^{(t)} - \mu)^2 + \frac{n_2^{(t)}}{n}(\mu_2^{(t)} - \mu)^2$，我们选取 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
  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 [22]，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。

2.4 其它

对于通过扫描得到的数学公式，有时还有必要进行倾斜校正和卷曲校正，这里我们仅为完整性而对一些有关方法作简单介绍。

由于扫描时纸张摆放不正，得到的图片往往有微小的倾斜情况。这时，如果可以检测出倾斜角度，则通过反方向旋转该角度即可还原。以下就给出倾斜检测的一些方法：

- 投影法
  把图片向多个稍偏离横轴的方向进行投影，选择一个角度使投影图具有某种极端性质（例如空白区最长）。
- 直线检测法
  对前景象素点进行 Hough 变换（也可以先采用降维手段以减低计算量），寻找局部密度较大的点以估计倾斜角度。
- 主轴方向法
  利用图片的矩计算其主轴方向，以此估计数学公式的方向。这种方法优点在于计算量小。
- 基于识别的方法
  利用旋转不变特征对（部分）字形进行高可信度的识别，通过估计这些字形的倾斜角度估计图片的倾斜。

其中上述前三种方法虽然适用于普通文本行，但不适应数学公式的特点，对于一些数学公式会给出错误的结果，例如没倾斜的 $x^{x^{x^{x}}}$ 可能被认为是倾斜的。最后一种方法可用于各种结构的公式，但精度较低，而且在预处理阶段做识别会使系统设计混乱。

由于扫描书籍时页面靠近装订线处难以平展，得到的图片往往有变形情况。对于文本行，有以下处理方法：

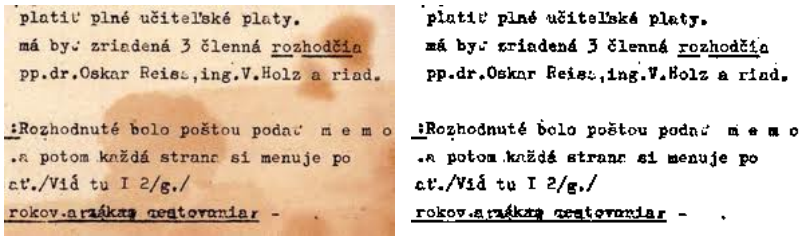
- 基于模型的方法
  利用图片中有代表性的曲线建立圆柱面模型并以此进行恢复 [2]。
- 基于图像的方法
  先把连通体聚成字进而行，然后平移每个字使其中心落在所在行用 Hough 变换得到的直线上，最后旋转各字完成校正。[23]
- 综合方法
  通过垂直投影函数、有效包围盒和标记点提取文本行中心线以估计全局几何参数，通过分片四边形映射进行校正 [24]

然而，数学公式的存在会干扰这些方法。

综上所述，我们认为倾斜校正和卷曲校正更适合对整个页面而非单个数学公式进行 (科技文档不常有整页全是公式的页面的情况，通常还有不少文本行，这些文本行可以用于估计倾斜和卷曲情况)，所以在我们实现的数学公式识别系统中也暂时不再考虑倾斜校正和卷曲校正问题。

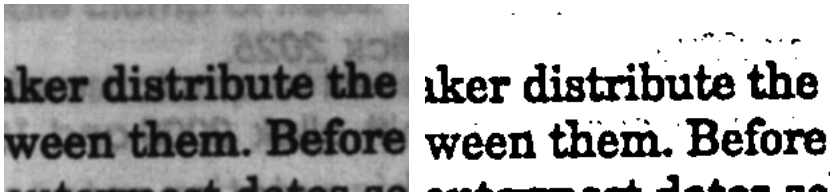
2.5 预处理效果

图2.1,2.2和2.3分别演示了使用灰度化加上 Sauvola 方法的图形预处理效果。从试验结果可见，图形预处理确实能至少部分地克服光照不均、渗透和水印等问题，对提高图形质量有帮助。



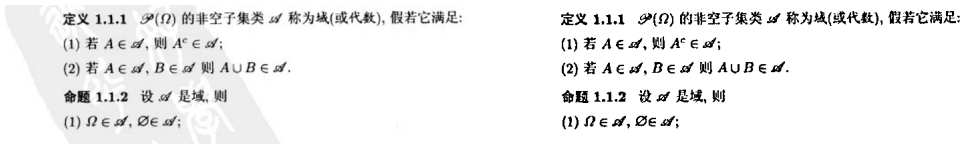
(a) 输入图形 (b) 预处理效果

图 2.1: 对背景不均图形的预处理效果



(a) 输入图形 (b) 预处理效果

图 2.2: 对有穿透情况图形的预处理效果



(a) 输入图形 (b) 预处理效果

图 2.3: 对有水印图形的预处理效果

2.6 小结

我们就公式图片中常见的问题考虑了在预处理阶段的对策以便后续使用。

首先，通过灰度化减低前景和背景颜色的影响。

其次，通过滤波手段企图去除各种噪声的干扰，而尽可能不破坏公式主体。而对于大型噪声，仍留待识别过程由于拒识而被抛弃。

为了简化处理，图形最终会被转换为二值图形。为了克服光照不均和背景纹理的问题，动态阈值法 Sauvola 方法是一个较佳的选择。

至于倾斜校正和卷曲校正，它们更适合对整个页面而非单个数学公式进行，故我们在这里不予处理。

部分 3

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。

3.1 字形分割

我们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，很多特征使用游程编码计算是方便的。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 $i_1, \cdots, i_s$，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 $\{i_1, i_2\}, \cdots, \{i_1, i_s\}$。
   - 否则，创建一个新的连通域（予以编号）并把此游程加到其中。
   - 在像素矩阵中把当前游程中像素置为所加入连通域编号的相反数。
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，然后利用常规遍历方法求无向图连通分支，再分别把各连通分支中顶点对应连通域合并，则余下的各连通域即为所求。

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) 作为字形，作为进行匹配的单位。

3.2 字形识别

在字形识别过程中，我们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缩少候选字形集合中字形数，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），这对大多数字符而言很稳定，但仍然可能由于尺度变换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，这对大多数字符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符号（这类字符将在本章稍后处理）。

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，。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 $i = 1, \cdots, m; j = 1, \cdots, n$，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 $d_{ij} = \frac{N_{ij}}{A_{ij}}$，我们用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。

- 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，计算它们间的豪斯多夫距离 [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，还容易并行化计算，而且容易为人类理解，不要求掌握复杂的机器学习背景。

此外，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，为此我们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。

3.3 字形合并

在进行字形识别后，可以利用位置关系把字形合并为字符。对每个连通域（外接矩形从大到小）：

1. 对属分体字符的一部分的候选字形，据数据库检查该分体字符其它字形应在的位置是否有字形。
   - 如全部出现，计算候选字符的像素矩阵和各个字形合并所得像素矩阵间的豪斯多夫距离从而相似系数，以之代替连通域与候选字形间的相似系数。
   - 否则，从连通域的候选字形表中去掉它。
2. 把连通域识别为相似系数最大（另外优先考虑字形数较多的字符）的候选字形所属字符。
3. 从连通域集合删去所有所选字符其它对应字形所在的连通域。

现在，我们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。

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

3.4 特殊符号识别

部分符号可能并不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 |、(、)、[、]、{、}、⌊、⌋、⌈、⌉）、根号（$\sqrt{}$）、箭头（$\longrightarrow$）和水平括号（如 $\overbrace{}$、$\underbrace{}$），这时基于几何矩、投影、模板匹配等统计特征的匹配效果会变差。

针对这些特殊符号，有以下识别方法：

- 利用结构特征进行识别。人手设计可以惟一标识各个特殊符号的特征，用于进行识别。
- 动态生成模版。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各个特殊字符模板按常规方式进行匹配。这将在我们的系统中使用。

应该指出，没有必要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。

此外，鉴于省略号（包括 ...、$\vdots$ 和 $\ddots$）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，我们选择了利用经验规则合并圆点以产生省略号。

3.5 数学符号数据库的建立

为了建立一个分类器，我们需要训练集，即一个已知字符归属的字符图片集合。自然的获取途径有两个：解析计算机字体文件、扫描再人手标记。前者显然要省工作量，容易扩充支持的字体列表。因此，我们决定以字体文件和代码点与字符名称对应表为输入，生成用于识别的数据。

对每个字符记录如下信息:

- 字符名称
- 基线信息
- 下属字形集合

对每个字符的每个字形记录如下信息:

- 字形位置
- 孔洞数
- 穿线数

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 3.1: 字符识别的识别率

- 网格特征
- 几何矩
- 密度
- 模版（采用游程编码）

虽然所记录特征不完全是独立的，但在建立数据库时进行预计算有助于避免重复计算，提高识别阶段的效率。

3.6 测试结果

我们利用 Tex Live[26] 中的 AMSFonts 字体中符号作为训练集和测试集，包括 CMB10、CMBSY10、CMEX10、CMMI10、CMMIB10、CMR10、CMSY10、MSAM10、MSBM10、RSFS10，这些字体在 SIL 开放字体许可证下发布，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 TTF 文件，先把它们用 FontForge[27] 转换为 TTF 文件，并生成字体列表（后缀名设为“.nam”）。然后，手动把字体列表中符号的名字改为在 LaTeX 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，我们利用一个自动化测试程序生成完美孤立字符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、投影、模版匹配。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，我们并不区分字体。

由实验结果如表3.1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。虽然缺少可供直接比较的数据，单从这些数据看来，我们的方法有过度拟合之嫌。一方面，必须承认，在没有外加噪声情况下这个识别率并不理想。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。

3.7 小结

我们设计的字符识别系统以字形为基本识别单位，先进行连通域分割再作简单合并以得到待识别字形，然后对每一个待识别字形进行识别，给出一些候选字形和相应的识别确定度，最后把字形按平面位置关系合并为字符。

在字形识别过程中，首先利用比较稳定的的特征作粗分类，然后再分别用多种方式计算与各候选字形间的接近程度以进行细分类，最后以基于豪斯多夫距离的模版匹配得到一个识别确定度。在这个框架下，容易加入各种分类器，并可以并行化，因而具有一定的可扩展性。对于不能由字体模版缩放得的特殊符号，我们采用动态生成模版的方法处理。另外，对于受离散化影响较大的圆点和横线，则制订了特殊规则予以判定。

其中，字符和字形的有关数据通过字体文件获取，使用这个方法可以容易地增加支持的字符数。当利用排版软件（例如 LaTeX）的字体文件来生成识别数据时，将对使用该软件排出的公式有较好的适应性。

实验结果表明，上述的字符识别方案可以对数学符号给出明显优于瞎猜的识别结果，但仍有很大改进空间。

部分 4

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 LaTeX 代码。我们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。

4.1 数据结构

盒子这个数据结构用于表示子公式，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 LaTeX 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、根号和次数组成的子公式。
- 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。

4.2 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，按各自参考大小和基线决定上下，然后回到合并步继续。

[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

(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，副管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。

4.3 测试结果

我们用 MathOCR 对两本数学公式排版教程中数学公式转换为图片用 MathOCR 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4.1所示[2]，我们看到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够的。其中，导致出错的主要原因包括上下标关系误判、存在字符识别错误和存在不支持的数学公式结构。

4.4 小结

在结构分析中，我们使用了自底向上方法，过程为先让每一个符号用一个盒子表示，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。

[2]部分公式同时属于多于一种类型

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%
多行表达式	35	30	85.7%
总计	147	113	76.9%

表 4.1: 结构分析的准确率

应该指出，本章介绍的结构分析方法的输入不一定非要由前一章的符号识别技术得到，也可以是解析 PostScript 或 PDF 文档得到符号识别结果。

由实验结果可见，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够的。

部分 5

总结和展望

5.1 基本方法

在本系统开发过程中，广泛参考前人在文字识别、数学公式识别等方面的工作，并以程序员的视角作独立思考，拒绝瀑布模型，及早建立可运行原型，优先选用简单方法，经过反复测试和修改逐步改进，最后才对性能作优化。

更具体地，公式识别系统各组成部分的设计思路如下：

1. 图形预处理
   首先了解可能影响光学文字识别的图形质量问题，然后参考数字图形处理领域的现成解决方法，选择简单且被推荐的方法加以实现，并以实验效果决定适用性。

2. 符号识别
   首先选取若干特征并分别建立基于它们的分类器，通过实验所得识别率选择分类器的组合。最后，对于少数频繁出现的误判情况，尝试建立经验规则加以排除。

3. 结构分析
   首先设计一种足以表示从局部到整体识别结果的数据结构和合并算法的框架，然后逐步实现各数学公式类型的具体合并算法。

5.2 取得成果

我们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。

出于可携性和可用类库的考虑，在主流程序设计语言中，我们选择了用 Java[25] 语言进行开发，并且提供了基本的图形介面（见图5.1和图5.2），它容许用户观察并修改关键步骤的结果。这个公式识别系统暂定名为 MathOCR，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，项目网站http://sourceforge.net/projects/mathocr/提供了软件（包括所有源码）的下载链接。

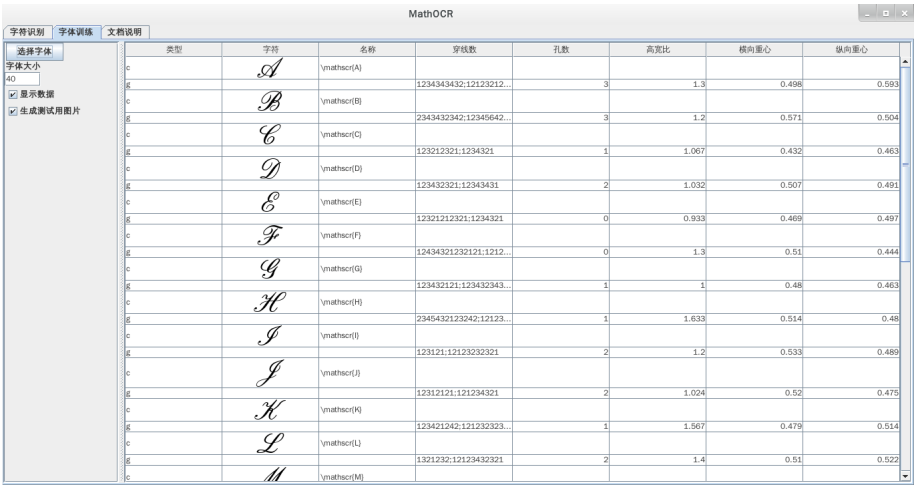


图 5.1: MathOCR 中训练字体的图形界面

它的设计虽然主要为组合传统技术，但也有以下的创新点：

- 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符识别问题化为常规字符识别问题
- 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，没有显式地使用任何统计模型或形式文法

5.3 下一步工作

虽然我们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影响很大，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
  我们的字形分割仅使用连通域分割，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点切分或与邻近连通域合并。

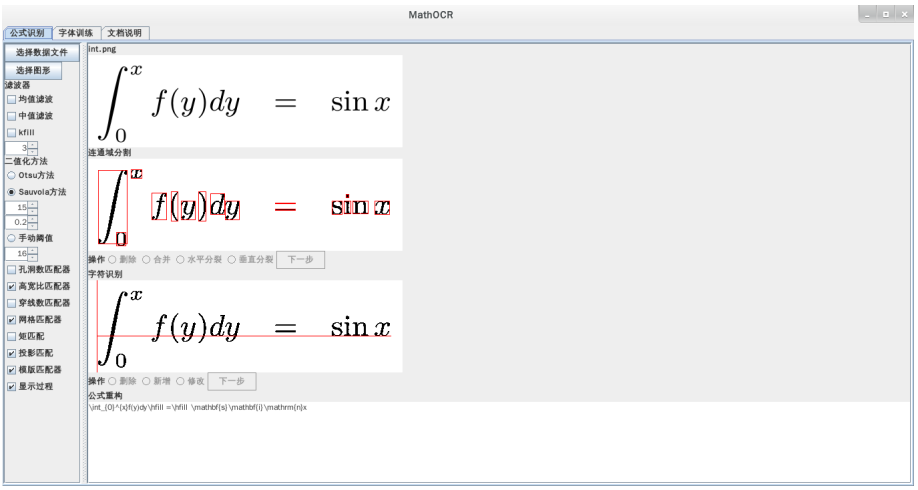


图 5.2: MathOCR 中公式识别的图形界面

- 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也应该把它们识别出来。虽然原则上我们的程序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
  我们的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。

- 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，对识别结果进行修正。

5.4 应用前景

以数学公式识别系统为基础，可以构建多种意义重大的应用：

- 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，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有类似的地位。化学公式与数学公式同样是二维结构，它们的识别难点有类似的地方，因而数学公式识别的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。

主要参考文献

[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] 田学东. 光学公式识别技术研究 [D]. 保定: 河北大学, 2007.

[3] 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.

[4] 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.

[5] 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.

[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] Yuko Eto, Masakazu Suzuki. Mathematical formula recognition using virtual link network[C]// Proceedings of Sixth International Conference on Document Analysis & Recognition. Washingington: IEEE Computer Society, c2001 : 762-767.

[10] 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.

[11] 靳简明, 江红英, 王庆人. 数学公式识别系统:MatheReader[J]. 计算机学报, 2006, 29(11):2018-2026.

[12] 张志伟. 数学表达式数字化处理中关键技术的研究 [D]. 合肥: 中国科学技术大学, 2007.

[13] The Pattern Recognition and Human Language Technology research center. Mathematical Expression Recognition[EB/OL]. [2014-08-17].http://cat.prhlt.upv.es/mer/.

[14] 赛酷科技有限公司. 商务合作 [EB/OL]. [2014-08-17].http://www.saqtech.com.cn/business.asp.

[15] 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.

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

[17] Suzuki M, Tamari F, Fukuda R, Uchida S, Kanahori T. INFTY —An Integrated OCR System for Mathematical Documents[C]// Proceedings of the 2003 ACM Symposium on Document Engineering. Grenoble: Elsevier B.V., c2003:95-104.

[18] Linda G. Shapiro, George C. Stockman 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.

[19] Wilhelm Burger, Mark J. Burge 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.

[20] 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.

[21] O'Gorman L. Image and Document Processing Techniques for the Right-Pages Electronic Library System[C]// Proceedings of the International Conference on Pattern Recognition. Los Alimitos: IEEE, c1992: 260-263.

[22] Shafaita F, Keysersa D, Breuelb 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. [Unknown]: Elsevier B.V.,c2008 : .

[23] Zhang Z, Tan C L. RESTORATION OF IMAGES SCANNED FROM THICK BOUND DOCUMENTS[C]// INTERNATIONAL CONFERENCE ON IMAGE PROCESSING. Thessaloniki: IEEE, c2001 : 1074-1077.

[24] 向世明, 赵国英, 陈睿, 贾富仓, 李华. 积厚文档扫描图像校正 [J]. 计算机辅助设计与图形学学报, 2005, 17(01): 42-48.

[25] Open JDK 1.7.0_55[CP/DK]. http://openjdk.java.net/.

[26] TeX Live 2014[CP/DK].http://tug.org/texlive.

[27] George Williams. FontForge[CP/DK]. [2012-07-31]. http://fontforge.org/.

An Attempt on Optical Formula Recognition

Chan Chung Kwong
(School of Mathematics and Computational Science
, Sun Yat-sen University, Guangzhou, Guangdong 510275, China)

Abstract: Since mathematical formula appears frequently in different kinds of documents, Optical Formula Recognition(OFR) should be an essential part of a general-purpose document analysis system. This article proposed a practical solution on the matter. Like most existing designs, the system consist of two main parts: symbol recognition and structural analysis. For the character recognition part, the core is the glyph recognizer, coarse classification is followed by fine classification to produce candidates, and then template matching based on Hausdorff distance is used to verify. Empirical rules is also 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 used. 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. Although the system has not yet acquired industrial strength and robustness for daily use, it can produce noticeable output using high-quality input.

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