中山大學
SUN YAT-SEN UNIVERSITY

本科生毕业论文

题　　目：印刷体科技文档识别技术实践研究

院　　系：数学与计算科学学院

专　　业：数学与应用数学

学生姓名：陈 颂 光

学　　号：11336019

指导教师：黎培兴（副教授）
（职称）

二〇一五年五月

表一　毕业论文（设计）开题报告

论文 (设计) 题目: 印刷体科技文档识别技术实践研究

(简述选题的目的、思路、方法、相关支持条件及进度安排等)

目前，大量书籍和其它出版物依然只有纸质版或扫描版可得，造成重用上的困难。一方面，纸质版或扫描版的材料不便于进行检索，导致分散在大量文献中的信息不容易被发现，从而得不到充分利用。另一方面，引用这些材料涉及繁琐和易误的重新输入工作。因此，为了整合和盘活文献资源，有必要建立可以把现存的文献自动化地转换成便于重用形式的识别系统，这对科学技术传播有现实意义。

经过多年的发展，光学图像采集、版面分析、光学字符识别等技术已经实用化，形成了一些文档识别系统。然而，数学公式识别技术仍未获广泛使用，加上版面分析侧重于物理版面分析而非逻辑版面分析，对科技文档的电子化带来了不便。因此，科技文档识别系统的开发工作应当把数学公式识别和逻辑版面分析作为突破口。

虽然系统的开发有着一定的挑战性，但借助已有的研究成果和过去的实践经验，采用曳光弹式开发方法，优先选用简单方法，及早建立可运行的半成品，经过反复测试和修改逐步改进，可以期望最终可以形成一个基本完整的科技文档识别系统。

本选题有很强的实践性, 主要支持条件为投入到具体开发的时间。此外, 充足的参考文献和自由软件资源也是不可或缺的。扎实的数学和计算机功底则是开发顺利进行的重要保障。

计划在 2014 年 12 月开始建立程序框架和图像预处理模块, 在 2015 年 1 月开始建立版面分析模块, 在 2015 年 2 月开始建立字符识别模块, 在 2015 年 3 月进行数学公式识别系统的整合工作, 在 2015 年 4 月进行测试和论文写作工作。

指导教师意见:

1. 第一部分需要介绍当前的数学公式的识别技术与国内外研究现状与进展。
2. 其他部分按照分析 -设计 -实现，几个部分做章节划分。
3. 文中尽量体现自己的思路、方法、实现，使用多种方式描述自己的解决方案并做分析、讨论和比较。

1、同意开题 ( ) 2、修改后开题 ( ) 3、重新开题 ( )

指导教师签名:　　年　　月　　日

表二　毕业论文（设计）过程检查情况记录表

指导教师分阶段检查论文的进展情况（要求过程检查记录不少于 3 次）:

第 1 次检查

学生总结：

1. 已经建立一个具备基本功能的文档分析系统。
2. 已经写出论文大致完整的草稿，但一些地方不够清晰。
3. 正在试验一个新的数学公式结构分析算法，有待完善和测试。

指导教师意见：

1. 删除第二页中的关于可用数学方法罗列的参考文献。只保留有引用关系或直接参考意义的文献。
2. 需要精炼表述论文内容，合并和删除雷同部分，雷同部分可作为附录。
3. 与工程文档不同，毕业论文只需介绍框架和关键处理部分即可，其他非重要部分忽略或删除。
4. 需要突出的自己的总结和分析内容，删除大量的罗列内容。
5. 必须是论文结构紧凑，章节需要过渡段落。

第 2 次检查

学生总结：

1. 继续改进数学公式结构分析算法，特别是重新支持了矩阵和多重下标，但仍有不小的改进空间。
2. 针对中心不够突出的问题，对论文作了一些局部的精简、补充和修改。

指导教师意见：

1. 需要调整内容和章节结构，按照“提出问题，分析问题，解决问题”的思路写论文，不适宜采用技术文档的描述方式。
2. 章节结构建议：
   第一章 ZZ 应用问题
   第二章 ZZ 技术的最新研究进展
   第三章 YY 方法及其应用
   第五章 基于 YY 的 ZZ 应用
   第六章 小结
3. 使用“总 -分 -总”的形式，表述自己的思想方法和处理步骤。
4. 突出关键内容，删除雷同或非主要不分明。
5. 论文是议论文，着重分析和讨论过程。不需要面面俱到，其他内容作为附录。

第 3 次检查

学生总结：

1. 继续改进数学公式结构分析算法，准确率有所提升。
2. 加入了对版面分析模块的一些测试结果，并扩充了对数学公式识别模块的测试规模。
3. 进一步突出论文的主要思路，移走过细的分支材料以提高可读性。

指导教师意见：

1. 将每一次的导师指导意见填写到表格中。
2. 删除过多的空白（页）或说明，摘要前的内容精简，突出自己的主要工作。
3. 关键突出实现部分，将自己的分析、处理、结论和技术（代码）实现细节更好地分析和讨论。
4. 建议程序的部分内容，作为处理步骤和过程，部分放在正文。
5. 主要程序作为附录。附录无需非核心内容，如记号变量表等。
6. 使用总 -分 -总结构，使用更多的讨论和总结将内容整合。

第 4 次检查

学生总结：

指导教师意见：

学生签名：　　年　　月　　日

指导教师签名：　　年　　月　　日

总体完成情况	指导教师意见： 1、按计划完成，完成情况优（ ） 2、按计划完成，完成情况良（ ） 3、基本按计划完成，完成情况合格（ ） 4、完成情况不合格（ ） 指导教师签名：　　年　　月　　日

表三　毕业论文（设计）答辩情况登记表

答辩人	陈颂光	专业	数学与应用数学
论文（设计）题目	印刷体科技文档识别技术实践研究		
答辩小组成员			

答辩记录：

记录人签名：　　　　　　　　　　　　　　　　　　　　年　　月　　日

学术诚信声明

本人所呈交的毕业论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料均真实可靠。除文中已经注明引用的内容外，本论文不包含任何其他人或集体已经发表或撰写过的作品或成果。对本论文的研究作出重要贡献的个人和集体，均已在文中以明确的方式标明。本毕业论文的知识产权归属于培养单位。本人完全意识到本声明的法律结果由本人承担。

本人签名：　　日期：

印刷体科技文档识别技术实践研究

[**论文摘要**]

由于数学公式识别和逻辑版面分析功能缺失等原因，文档识别在教育、科研等方面的应用受到了限制，不利于科学技术传播。因此，研究印刷体科技文档识别技术并培育一个可用系统有现实意义。

本文以实用的观点对科技文档识别技术进行了系统的探索，形成了基本完整的技术链，足够建立一个整体设计。其中，在版面分析方面，提出了一种面向行的逻辑版面分析方法，能生成段落、标题和列表项目等逻辑块；在字符识别方面，提出了一种以字形为基本识别单位的字符识别方法，能够识别处于复杂位置关系的字符和可局部伸缩的特殊符号；在数学公式识别方面，提出了一种基于二维结构和符号类型的数学公式定位方法以及基于符号邻接图的数学公式结构分析方法，能识别上下标、帽子、分式、根式和矩阵等多种数学公式类型。这些技术在准确性、效率和可扩展性间取得了一个平衡。

系统的实现以自由软件形式提供，本文还基于公开的数据集对倾斜校正、字符识别、版面分析和数学公式结构分析等部分进行了量化的性能评估。总体上，虽然系统还有点粗糙，但已具备各种基本功能，经过更多的工作有望可以投入日常使用。

[**关键词**]　文档识别；版面分析；数学公式识别；字符识别

A Practical Study on Printed Scientific Document Recognition

[**Abstract**]

Due to the absent of functionality of mathematical formula recognition and logical layout analysis, the use of optical document recognition in education and scientific fields is limited which is a lost for the spread of advance in science and technology. Therefore, building a complete printed scientific document recognition system is of practical interest.

In this paper, key aspects of scientific document recognition are discussed systemically from a practical point of view, applicable algorithms for each stage are given in detail. Accuracy, efficiency and extensibility are considered. By combining all these works, one can outline a design of a real system.

A logical layout analysis method based on text line is proposed, it can find out paragraphs, headings, list items and more. A new character recognition framework based on glyph matching is suggested, it can identify character in complicated context, special symbol can also be recognized using dynamically generated template. A mathematical formula detection algorithm based on 2-dimensional structure detection and symbol type, in addition to a structural analysis method for mathematical formula based on symbol adjoin graph is described, scripts, hats, fractions, radical expressions, matrices are supported.

An implementation, MathOCR, is available as a free software. Quantitative evaluation shows that the system has achieved impressive performance. Although the system comes with the most essential features, more work is needed before becoming capable for daily use.

[**Key Words**] document recognition; layout analysis; mathematical formula recognition; optical character recognition

目录

第一章　引言

1.1　背景

目前，大量书籍和其它出版物依然只有纸质版或扫描版可得，造成重用上的困难。一方面，纸质版或扫描版的材料不便于进行检索，导致分散在大量文献中的信息不容易被发现，从而得不到充分利用。另一方面，对这些材料进行进一步处理涉及繁琐和易误的重新输入工作。因此，为了整合和盘活文献资源，有必要建立一种有效机制把现存的文献转换为一种统一、便于重新利用的形式，这对科学技术传播有现实意义。

经过多年的发展，光学图像采集设备、版面分析、光学字符识别等技术已经实用化，形成了一些文档分析系统。然而，数学公式识别技术仍未获广泛使用，加上版面分析侧重于物理版面分析而非逻辑版面分析，为科技文档的电子化带来了不便。

有鉴于此，开发一个可跨平台自由使用的科技文档识别系统是有必要的，其中数学公式识别是其中一个亮点。这样的系统具有明确的用户定位，预期用户包括需要少量识别文献的教师、学生和研究人员，还有需要批量识别文档的电子图书馆管理员，有填补市场空隙的应用潜力。

在文档识别的相关领域虽然已经发表过许多方法，但在大多数方面并未有公认最好的方法。特别是，数学公式识别技术尚不成熟，现成产品的缺少就说明了其难度。不过，科技文档版面相对有规律，不像中文报纸那样有很复杂的版面，这也降低了识别的难度。可见，虽然系统的开发有着一定的困难性，但由于已有许多研究成果可供参考，而且有着过往的实践经验，因而开发仍然是完全可行的。

综上所述，科技文档识别系统的开发工作是挑战和机遇并存的。

1.2　相关领域的研究现状

目前的文档识别系统在架构上已经大致定型，典型文档识别系统的主要工作包括图形预处理、版面分析和文字识别 [1]。而作为科技文档识别系统，还需要加上数学公式识别这一环节。

图像预处理的目的是使图像更能反映文档的原貌，它的两个主要任务分别是噪声去除和倾斜校正。在噪声去除方面，针对边缘噪声、背景噪声、椒盐噪

声和不规则噪声，已经分别有相应的去除方法被提出，一个综述参见 [2]。在倾斜校正方面，也已经提出了许多不同类型的方法，它们又衍生出一些以增强健壮性、降低计算量或提高准确性为目的的变种，一个综述参见 [3]。然而，各个方法的作者即使进行了性能评估，也往往是基于各自的数据库，且这些数据库通常不可获取，难以仅仅基于文献中的测试数据作出客观的比较。

版面分析的目的是把握文档的总体结构，它的两个主要任务分别是物理版面分析和逻辑版面分析。在物理版面分析方面，文献 [4] 比较了投影切分法、背景分析法、游程平滑法、文档光谱法、基于 Voronoi 图的方法和受限文本行寻找方法等版面分割方法的性能，文献 [5] 比较了一些统计特征和分类方法在物理块分类的效果，文献 [6] 提出了一个基于拓扑排序的阅读顺序确定方法。在逻辑版面分析方面，讨论相对较少，文献 [7] 提出了一些用于生成段落和区分各类逻辑块的经验规则。应该指出，这些方法都有一定的误识率。

字符识别的目的是辨认出文本块中的每个字符，它的典型流程为字符切分、特征提取、分类到后处理。字符切分方面的基本手段为投影和连通域分析 [8]，特征提取方面已有很多选择被提出 [8, 9, 10]，分类方面同样有最近邻分类器、人工神经网络和支持向量机等多种分类器被提出 [8, 10]，后处理方面也已提出一些基于词典或随机文法等的语言模型以识别词等比字符更高层的语言单元 [11]。不过，这些方法大多未有考虑到应用于数学公式中的符号识别时会产生的额外困难。

数学公式识别的目的是辨认文档中的数学公式，它的任务包括数学公式定位、数学符号识别和结构分析。在数学公式定位方面，用于定位行内公式的方法有基于识别信息的方法 [9] 和基于二维结构检测的方法 [12]，用于定位独立行公式的方法则有基于版式特征的方法 [9, 12]。在数学符号识别方面，已提出的方法与一般的字符识别方法类似，但准确率往往会降低 [13]。在结构分析方面，现有方法可分为基于结构的方法和基于文法的方法，一个综述参见 [14]。现有大部分方法的一个局限性在于，局部和整体信息未能同时加以充分运用。

关于上述领域的研究现状，在后续章节还将有更细节性的讨论。在这里仅指出，众多方法的存在除了说明了问题受到重视的程度外，也说明了问题的困难性 —未发现一个方法全面优于其它方法。

1.3 本文工作

本文的目的很明确，就是培育一个科技文档识别系统。按照引擎与接口相分离的原则，本文系统根据子功能作结构划分如图1.1所示。在接口方面，包括交互式图形用户界面（适合用于需人手保证正确性的应用场境）、批处理式命令行界面（适合用于图书馆大规模电子化纸质文献的应用场境）和应用编程接口（适合用于嵌入到一个知识管理系统作为子系统的应用场境）。在引擎方面，与

通常的文档分析系统类似，包括图像预处理、版面分析和字符识别模块，此外还有一个不那么常见的数学公式识别模块。本文系统的基本工作流程如图1.2所示。

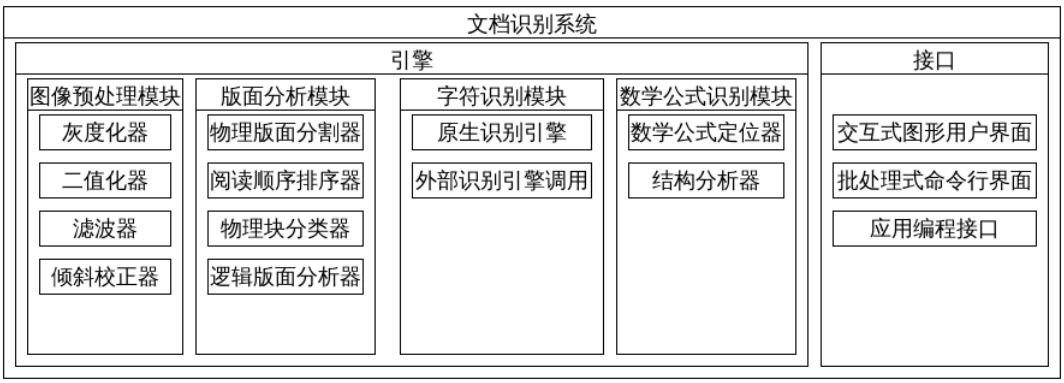


图 1.1: 本文系统的主要组成部分

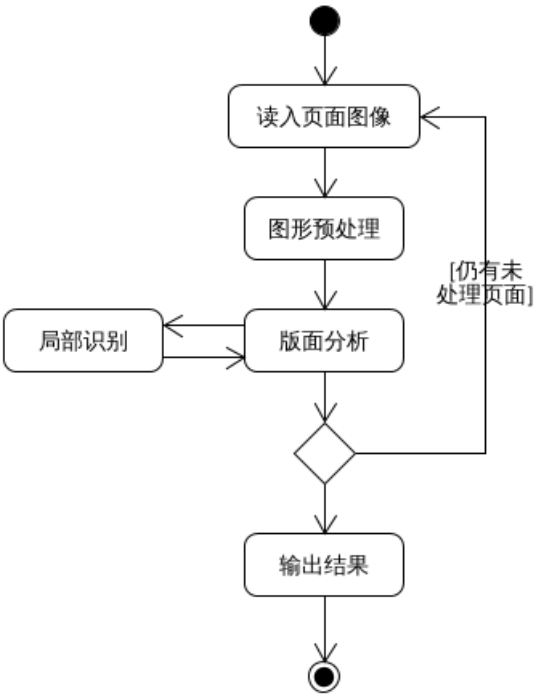


图 1.2: 本文系统的基本工作流程

出于设计本文系统的需要，有必要对各个工作流程面临的问题作出系统的探索，这些讨论构成了本文的主体部分。

在图像预处理方面，本文介绍了去除噪声和倾斜的主要手段。对于噪声去除问题，由于缺乏客观的比较标准，本文并不企图对不同的噪声去除方法作量化的比较。对于倾斜校正问题，本文利用一个公开的数据库对七种倾斜检测方法进行了性能评估。

在版面分析方面，本文选择了使用投影切分法进行物理版面分割，然后按启发式规则把物理块分为文本块和非文本块，接着使用基于拓扑排序的方法确定阅读顺序，以完成物理版面分析。对于文本块，本文提出一种利用行的对齐方式和识别结果生成段落、列表项目和标题等逻辑块的有效算法，从而达到逻辑版面分析的效果。

在字符识别方面，本文给出一个使用字形为基本识别单位的字符识别方法，其中常规字符采用层层筛选再作模板匹配的识别策略来识别，特殊字形则通过动态生成模板来识别。本文利用多种特征分别构造了匹配器，并以实验比较了多种匹配器组合的性能。

在数学公式识别方面，由于文本行中的字符已被识别，只余下数学公式定位和结构分析两个问题。对于数学公式定位问题，本文建议先利用空白和符号类型转变位置把文本行分为若干列，把含有数学符号或二维结构的列认为是数学公式，然后把相邻的同类列合并。对于结构分析问题，本文的解决方案为先综合利用局部信息和整体信息来构造符号邻接图，然后利用一些直观的规则进行图改写，同时生成排版代码。本文还基于公开的数据库检验了结构分析算法的准确性。

把所有这些工作结合起来足以提供一个科技文档识别系统基本完整的总体设计，本文系统就是它的一个具体实现。

第二章　图像预处理

2.1　问题的提出和分析

由于原始图像自身的瑕疵和转换过程中产生的失真，待识别图像往往存在一些质量问题。为了准确地识别文档，有必要使用数字图像处理技术对输入图像进行预处理，以尽可能排除因各种原因造成的干扰，使图像更能反映页面的原貌，为后续识别过程创造良好的条件。

在输入图像中，最常见的质量问题分别为噪声和倾斜。

噪声就是对识别目的而言无价值的信息，它们容易对后续识别过程造成不良影响。以下列出一些常见的噪声类型并简单讨论相应的去除方法：

- 边缘噪声
  边缘噪声是指图像超出当前页面范围以外的部分，例如进入扫描范围的纸张外区域（通常表现为大片深色区域）以及相邻页内容。去除边缘噪声的基本想法是检测出页面内容的范围，然后把其外的范围从图像中裁掉。页框检测可以人手进行，也已经提出一些自动化的方法 [15]。由于在不理解内容的情况下自动页框检测不完全可靠，故提供交互式地确定页框的机制也许更为有用。
- 背景噪声
  背景噪声即存在于文档背景的噪声，例如由背景不均匀、反渗、水印、起折或低对比度造成的。由于背景噪声与前景像素相比还是有明显区别的，去除背景噪声的基本想法是对图像进行二值化 [16]。
- 椒盐噪声
  椒盐噪声是指孤立的小型黑区域或白区域，例如由墨点、缺墨、纸张瑕疵或二值化造成的。由于椒盐噪声与实际内容相比是次要的细节，去除椒盐噪声的基本想法是通过滤波方法去除图像中的高频成分，而保留图像中的低频成分。
- 不规则噪声
  不规则噪声是指不属于以上类型的噪声，例如附在笔画上的噪声和人为的手写标记或印章（例如图书馆的印章）。由于不容易制订规则直接去除，一种应对方法是用机器学习方法构造用于判定噪声的分类器 [2]。注意到不规则噪声范围广泛且难以明确界定，还是提供交互式地消除不规则噪声的

机制比较实际。

倾斜则是由于扫描时纸张摆放不正造成的，也会影响版面分析和字符识别的准确程度。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。在倾斜检测方面，已经发表过许多方法 [3]。由于倾斜检测是一个比较活跃的专门研究领域，本文不拟提出新方法，但为了对各种方法作出客观的比较，本文将使用一个公开的数据库对一些倾斜检测方法进行性能评估，由此决定哪个方法值得默认采用。

应当指出，虽然噪声和倾斜是较常见的质量问题，去除它们将大大提高图像质量，但它们还不是全部的质量问题，例如扫描厚书籍得到的图像就可能由于卷曲而有非线性几何畸变，可见图像预处理还有其它工作可做。

2.2 问题的解决方法

2.2.1 基于二值化和滤波的噪声去除方法

正如已经指出的，二值化和滤波分别为去除背景噪声和椒盐噪声的重要手段，以下讨论具体方法。

二值化就是把彩色图像转换为二值图像，这样不但有助去除背景噪声，而且可以在保留对识别有用的主要信息的条件下压缩数据。二值化的基本流程为先把彩色图像化为灰度图像，再把灰度图像化为二值图像。

虽然程序接受的输入图像可能基于不同的颜色模型进行编码，但不同的颜色模型间大多有标准的转换公式，因此这里可以假定输入图像基于 ARGB 颜色模型，每个分量用一个字节表示，这对于文档识别的目的而言应是相当足够的。为统一记号，以下明确彩色图像、灰度图像和二值图像的定义：

定义 2.1 设 $m,n\in\mathbb{Z}$，则称

$$\begin{aligned} D\colon m\times n &\to 256\times 256\times 256\times 256\\ (i,j)&\mapsto (A(i,j),R(i,j),G(i,j),B(i,j)) \end{aligned}$$

为一个高度为 m 而宽度为 n 的彩色图像，对 $(i,j)\in m\times n$，称 $D(i,j)$ 为 D 在 (i,j) 处的像素值，并分别称 $A(i,j)$、$R(i,j)$、$G(i,j)$、$B(i,j)$ 为 D 在 (i,j) 处的不透明度、红色分量、绿色分量、蓝色分量。①

定义 2.2 设 $m,n\in\mathbb{Z}$，则称 $D:m\times n\to 256$ 为一个高度为 m 而宽度为 n 的灰度图像，对 $(i,j)\in m\times n$，称 $D(i,j)$ 为 D 在 (i,j) 处的灰度值。

定义 2.3 设 $m,n\in\mathbb{Z}$，则称 $D:m\times n\to 2$ 为一个高度为 m 而宽度为 n 的二值图像，对 $(i,j)\in m\times n$，若 $D(i,j)=0$ 则称 (i,j) 为 D 的一个前景像素，否则称 (i,j) 为 D 的一个背景像素。把 D 的所有前景像素组成的集合称为 D 的前景像素集。

作为把彩色图像转换为二值图像的一个中间步骤，彩色图像先被转换为灰度图像。对于 RGB 图像，转换为灰度图像的一个自然的想法是对三个分量取加权平均值作为灰度值，记一像素的红色分量、绿色分量、蓝色分量分别为 R、G、B，则 [17] 建议了一个灰度值 $\lfloor 0.309R + 0.609G + 0.082B \rfloor$。回到 ARGB 图像，因为不透明度为 0（即全透明）的像素应被认为是背景像素，而不透明度为 255（即全不透明）的像素对应灰度值应与去除透明度后 RGB 图像中对应灰度值一致，再注意到除以 1024 可以用右移 10 位代替和浮点运算较为耗时，于是作出如下的定义：

定义 2.4 设

$$\begin{aligned} D\colon m \times n &\to 256 \times 256 \times 256 \times 256 \\ (i,j) &\mapsto (A(i,j), R(i,j), G(i,j), B(i,j)) \end{aligned}$$

为一个高度为 m 而宽度为 n 的彩色图像，则称

$$\begin{aligned} D_0\colon m \times n &\to 256 \\ (i,j) &\mapsto \left\lfloor 255\left(1 - \frac{A(i,j)}{255}\right) + \frac{316R(i,j) + 624G(i,j) + 84B(i,j)}{1024}\frac{A(i,j)}{255} \right\rfloor \end{aligned}$$

为 D 的灰度化图像，记为 $\mathrm{gray}(D)$。

这个定义给出了求灰度化图像的直接方法。这是一个点运算，各点的灰度值可以相互独立地并行计算。对于高度为 m 而宽度为 n 的彩色图像，求灰度化图像的时间复杂度为 $\Theta(mn)$。

接下来讨论把灰度图像转换为二值图像的方法。一种明显方法为对每个像素设置阈值，把灰度值小于等于它的像素认为是前景像素，否则为背景像素。

定义 2.5 设 D 为一个高度为 m 而宽度为 n 的灰度图像，$T : m \times n \to \mathbb{Z}$，则称

$$\begin{aligned} D_0\colon m \times n &\to 2 \\ (i,j) &\mapsto \begin{cases} 0 & D(i,j) \le T(i,j) \\ 1 & D(i,j) > T(i,j) \end{cases} \end{aligned}$$

为 D 以 T 为阈值的二值化图像，记为 $\mathrm{threhold}_T(D)$。特别地，若 T 为恒取 t 的常值函数，则称 $\mathrm{threhold}_T(D)$ 为以 t 为全局阈值的二值化图像，也记为 $\mathrm{threhold}_t(D)$。

这个定义给出了求二值化图像的直接方法。这是一个点运算，各点是否前景像素可以相互独立地并行计算。对于高度为 m 而宽度为 n 的灰度图像，求二值化图像的时间复杂度为 $\Theta(mn)$。

除非有对待识别图像的灰度分布的先验知识，否则全局阈值选择常失之武断，对不同图片的适应性不理想。因此有必要给出根据具体图像计算对应全局

阈值的方法，Otsu 方法是一个被认为较优的全局阈值化方法 [17]，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。

当饱和度不均匀或存在背景噪声时，全局阈值化方法并不能满足需要 [16]，这时对每个像素计算一个局部阈值的局部阈值化方法就是值得采用的，Sauvola 方法是一个被认为较优的局部阈值化方法 [18]，其基本想法为按一个窗口中像素灰度值的均值和标准差决定阈值。

顺带指出，由于在文档中背景像素数通常明显大于前景像素数，因此前景像素多于背景像素时很可能意味着出现黑底白字的情况，这时可能应该交换前景像素和背景像素。

滤波器主要用于去除椒盐噪声，被用于文档图像的低通滤波方法包括均值滤波 [19]、中值滤波 [19]、kFill 滤波 [20] 和二值化后处理 [21]，其中后两个方法只适用于二值图像。这些滤波器不仅在去除椒盐噪声的能力上有所不同，在细节和连通性的保留情况上也有所不同。

由于倾斜校正、连通域分割、版面分析和字符识别中的算法很多都基于二值图像，故二值化是一个必要的过程，而滤波则是一个可选的过程。

2.2.2 典型的倾斜校正方法

倾斜校正的关键在于倾斜检测，综述 [3] 列出了七种有代表性的倾斜检测方法：分片填涂方法、分片覆盖方法、投影方法、交错数方法、Hough 变换方法、行间相关方法和最近邻聚类方法，其中既有比较传统的方法，也有近年发表的方法。在沿用文献中基本思路的前提下，实现时在一些细节上可以有所不同，例如：

- 把分片填涂方法中最终倾角估算从文献 [22] 中的众数改为中位数以避免离散化造成的问题。
- 把分片覆盖方法、投影方法、交错数方法、Hough 变换方法的搜索策略统一为二级搜索，步长依次为 $\frac{\pi}{60}$ 和 $\frac{\pi}{900}$。

假定倾斜校正算法得到从图像横轴到文档基线的有向夹角为 θ（顺时针方向为正），为作出纠正，只用将图像按逆时针方向旋转 θ。设一点 P 坐标②为 $(x, y) = r(\cos\phi, \sin\phi)$，则 P 绕原点按逆时针方向旋转 θ 后所得点 P' 的坐标为 $(x', y') = r(\cos(\phi-\theta), \sin(\phi-\theta)) = r(\cos\phi\cos\theta + \sin\phi\sin\theta, \sin\phi\cos\theta - \cos\phi\sin\theta) = (x\cos\theta + y\sin\theta, y\cos\theta - x\sin\theta)$。但因图像坐标的分量应为非负的，还需要进行一个平移。设图像高度为 h 而宽度为 w，则成立 $x' \geq \min\{0, h\sin\theta, w\cos\theta, w\cos\theta + h\sin\theta\}$ 和 $y' \geq \min\{0, -w\sin\theta, h\cos\theta, -w\sin\theta + h\cos\theta\}$，把两式右端分别记为 x_0, y_0，令平移所得点 P'' 的坐标 $(x'', y'') =$

$(x' - x_0, y' - y_0)$。两个变换复合后有

$$\begin{pmatrix} x'' \\ y'' \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \tag{2.1}$$

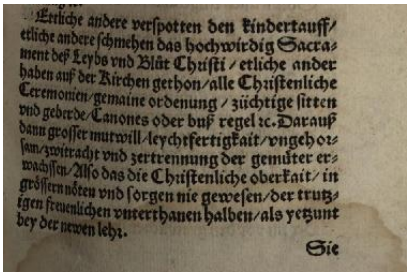
$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \left(\begin{pmatrix} x'' \\ y'' \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \right) \tag{2.2}$$

。此外，由于图像只在离散点取值，而上面公式给出的是连续值，实现旋转时还需要进行一些插值。

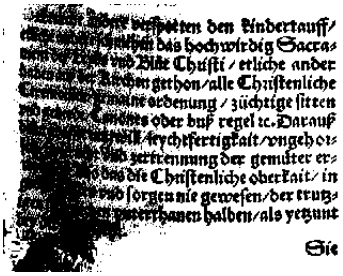
2.3 问题的解决情况

2.3.1 噪声去除

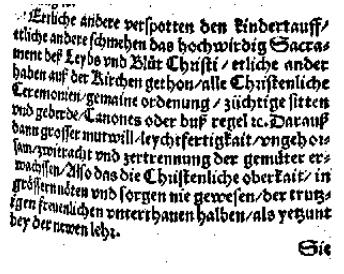
由于存在多种类型的噪声，对一些噪声类型建模本身就是一项很不平凡的工作，难以用统一的客观标准评价去噪效果，故这里不打算量化地比较各去噪手段。这里仅用一些图片直观地比较各二值化方法和滤波器的效果，见图2.1和图2.2。由于 Sauvola 方法确实往往能给出较 Otsu 方法更佳的视觉效果，本文系统采用 Sauvola 方法作为默认的二值化方法。而各种滤波器虽然可去除部分椒盐噪声，但往往不是很彻底，有时还会导致笔划断裂或粘连，所以本文系统默认不打开滤波器。



(a) 输入图像



(b) Otsu 方法二值化效果

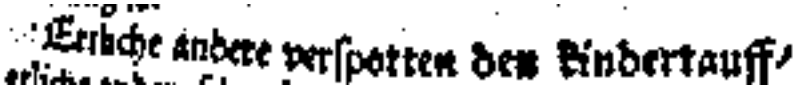


(c) Sauvola 方法二值化效果

图 2.1: 二值化的效果

(a) 输入图像

(b) 均值滤波或中值滤波效果



(c) kFill 滤波效果

(d) 二值化后处理效果

图 2.2: 滤波器的效果

2.3.2 倾斜校正

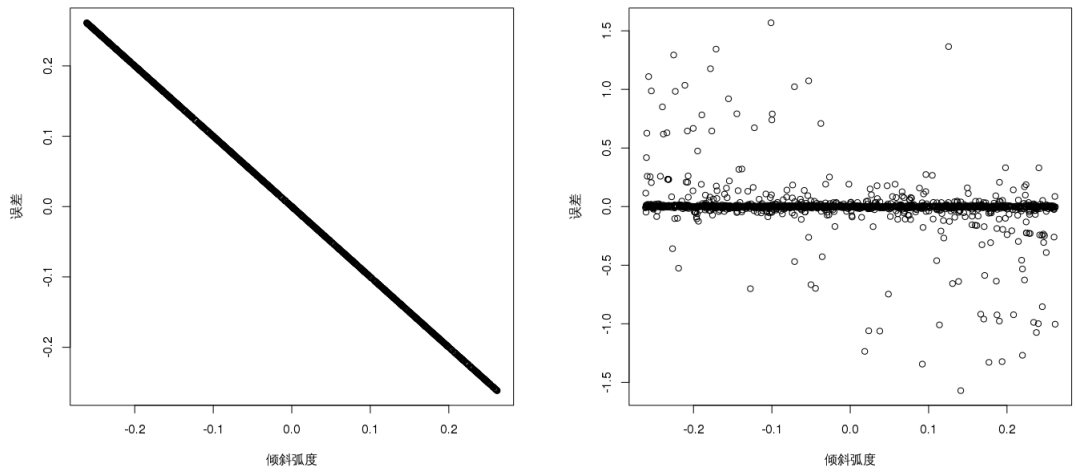
本文系统实现了七种分别基于分片填涂方法、分片覆盖方法、投影方法、交错数方法、Hough 变换方法、行间相关方法和最近邻聚类方法的倾斜检测算法。为了检验各个倾斜检测方法实现的性能，利用了 2013 年文档分析与识别国际会议（ICDAR）的文档图像倾斜检测竞赛（DISEC）基准数据集 [23] 中的 1550 个二值文档图像作为测试集进行了测试。由于测试集中倾斜角度从 $-\frac{\pi}{12}$ 到 $\frac{\pi}{12}$ 之间，基于分片覆盖方法、投影方法、交错数方法和 Hough 变换方法的算法的搜索范围设为 $[-\frac{\pi}{12}, \frac{\pi}{12}]$。在测试中，对每个图像分别调用各个倾斜检测方法，记录检测到的倾斜角度和所用时间。由于用户可容忍的倾斜检测时间有限，设置了 5 秒限时，超时将被视为未能成功返回（另一种未能成功返回的情况出现于分片填涂方法找不到足够条带时），这时倾斜检测结果也被视为 0。

在一台中央处理器为 AMD Athlon(tm) II Dual-Core M320 而内存大小为 2GB 的笔记本上，使用操作系统 Debian GNU/Linux jessie 下的 OpenJDK 1.7.0_65 编译和运行基于本文系统的测试代码时，得到实验结果的统计信息见表2.1和图2.3、2.4，其中参照方法是指总返回 0 的平凡方法。在实验中，未有发现任何一个算法在各个方面都全面优于其它算法。以下是一些观察：

- 参照方法，如所期望的那样，固然性能极高且极稳定，但基本上无用。
- 分片填涂方法有着很不错的效率，兼有合理的准确度，但在找不到足够条带时会失败。
- 分片覆盖方法有着合理的效率，准确度较高。
- 投影方法有着合理的效率，准确度较高且比分片覆盖方法可靠。
- 交错数方法的效率与投影方法基本相当，但准确度不如投影方法。

表 2.1: 各个倾斜校正方法性能对比

	成功返回率	平均误差（弧度）	均方误差（弧度）	误差中位数（弧度）	运行时间中位数（毫秒）
参照方法	**100.00%**	0.1298	0.1504	0.1286	**0**
分片填涂方法	93.48%	0.0621	0.1947	0.0061	50
分片覆盖方法	99.10%	0.0154	**0.0299**	0.0073	300
投影方法	99.35%	**0.0068**	0.0335	**0.0033**	197
交错数方法	99.35%	0.0101	0.0458	0.0035	198
Hough 变换方法	99.35%	0.1448	0.1827	0.0452	99
行间相关方法	93.81%	0.2115	0.3811	0.0346	1615
最近邻聚类方法	96.71%	0.0883	0.1207	0.0690	67



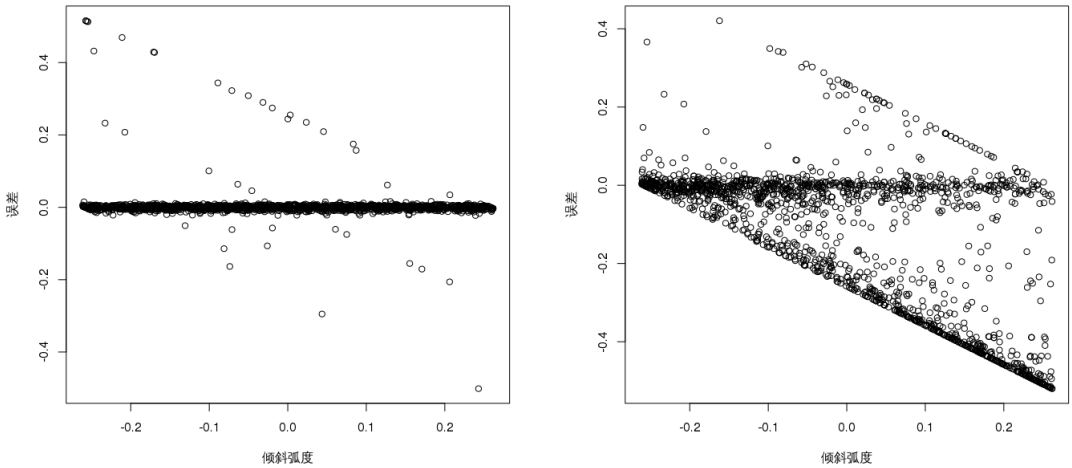
(a) 参照方法

(b) 分片填涂方法

(c) 分片覆盖方法

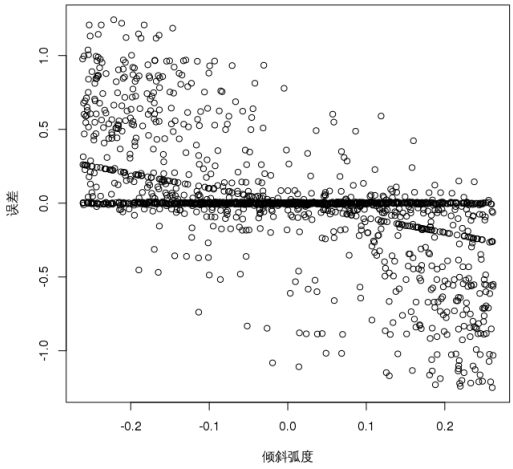
(d) 投影方法

图 2.3: 各个倾斜校正方法的残差图（上）

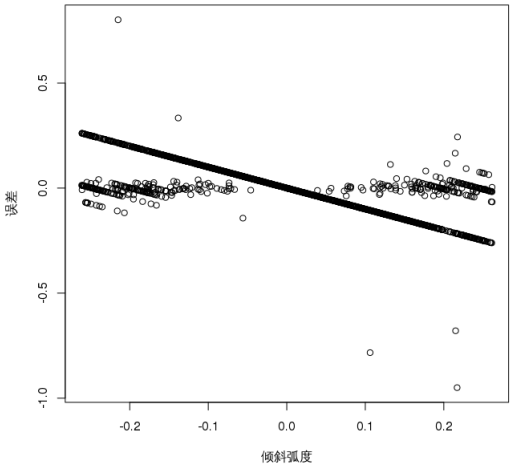


(a) 交错数方法

(b) 霍夫变换方法



(c) 行间相关方法



(d) 最近邻方法

图 2.4: 各个倾斜校正方法的残差图 (下)

- Hough 变换方法的效率比投影方法高，但准确度明显不如投影方法。
- 行间相关方法的效率明显较低，准确度也不是特别理想。
- 最近邻方法的效率不错，但准确度不是特别理想。

应该说明，由于各个倾斜检测方法的性能与所选参数和实现方式有关，这里得出的数据不足以完全代表所基于算法的优劣，特别是上述实验未能反映各种方法在大倾角时的表现。然而，根据实验结果，投影方法被选作本文系统的默认倾斜检测方法。

2.4 本章小结

本章介绍了文档图像预处理的一些过程和方法，覆盖了图像去噪和倾斜校正这两大问题，并且解释了本文系统中图像预处理模块中采用兼容并包策略的原因。对于倾斜检测问题，还给出使用一个公开数据库对七种倾斜检测算法进行性能评估的结果。

第三章　版面分析

3.1　问题的提出和分析

科技文档一般是结构化的文档，由标题、段落和图片等结构元素组成，适合用类似于 XML 的树结构表示。既然文档识别的根本目的在于重用，一个科技文档识别系统的输出不仅必须是可读的电子形式，而且应当是便于编辑的，这就要求输出能反映文档的逻辑结构。由此可见，面临的问题是把文档分解为组成它的结构元素及判断它们的类型，并决定这些结构元素的先后顺序。为统一术语，先给出一些定义。

定义 3.1 设 A 为一个集合，$\mathcal{A} \subseteq \mathcal{P}(A)$ 为一族互不相交的非空集合使 $\cup_{a\in\mathcal{A}} a = A$，则称 $\mathcal{A}$ 为 A 的一个分割，并称 $\mathcal{A}$ 中的元素为块。

本文关心的并不是一般的分割，而是就版面分割问题而言有意义的分割。一个简单观察是连成一片的前景像素应当属于同一个版面块，这让人想到连通域分割（它的准确定义和计算方法可以参考附录C）。

定义 3.2 设 $m, n \in \mathbb{Z}^+$，$F \subseteq m \times n$，$\mathcal{C}$ 为 F 的连通域分割，$\mathcal{F}$ 为 F 的一个分割，若 $\mathcal{C}$ 为 $\mathcal{F}$ 的一个加细，即对任何 $c \in \mathcal{C}$ 存在 $f \in \mathcal{F}$ 使 $c \subset f$，则称 $\mathcal{F}$ 为 F 的一个规范分割。

另一个观察是按照科技文档的排版惯例，各版面元素（如段落、标题、图片、表格等）应该分别有互不相交的外接矩形[③]。

定义 3.3 设 $F \subseteq \mathbb{Z} \times \mathbb{Z}$，$\mathcal{F}$ 为 F 的一个分割。对任何 $f \in \mathcal{F}$，记

$$R_f = \left\{ \min_{(i,j)\in f} i, \ldots, \max_{(i,j)\in f} i \right\} \times \left\{ \min_{(i,j)\in f} j, \ldots, \max_{(i,j)\in f} j \right\} \tag{3.1}$$

为 f 的物理矩形。若对任何 $f_1, f_2 \in \mathcal{F}$ 使 $f_1 \neq f_2$ 有 $R_{f_1} \cap R_{f_2} = \emptyset$，则称 $\mathcal{F}$ 为 F 的一个曼哈顿分割。

由于一次性地把文档分为其组成结构元素不好下手，可以采取两步走的策略，先进行较粗的分割，再进行细分。现在可以给出版面分割的形式化定义：

定义 3.4 设 D 为一个高度为 m 而宽度 n 的二值图像，F 为其前景像素集，把 F 的规范曼哈顿分割 P 称为 D 的一个物理版面分割，并称 P 的元素为 D（关于 P）的物理块。进一步，若 F 的规范曼哈顿分割 L 为 P 的加细，则称 L 为 D 基于 P 的一个逻辑版面分割，并称 L 的元素为 D（关于 L）的逻辑块。

3.1.1 物理版面分析

在物理版面分析阶段，目标是把文本和非文本区域分开，并把不同栏中的文本也分开，同时尽可能避免切开行。除进行分割外，还需要判定各物理块的类型，并决定它们的先后顺序。

由于科技文档的版面一般是比较规整的曼哈顿版面，文本区域与非文本区域间通常由明显的空白区分隔，且已对图像进行过倾斜校正，物理版面分析阶段也不要求很细致的划分，因此可以先考虑用容易实现且运算速度较快的自顶向下方法进行物理版面分割。

对于每个物理块，还需要判断其类型。由于物理版面分割本来就不是分得很细，这里只分为两类：

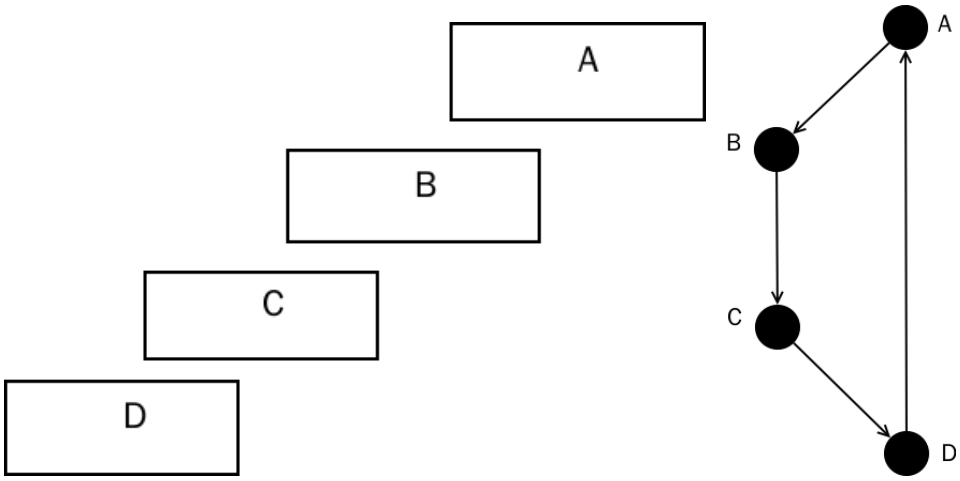
- 文本
  文本区域完全由字符组成，可以有内部结构，它们的内部结构将在逻辑版面分析阶段被进一步分析。
- 非文本
  非文本区域包括图像和表格，它们的内部结构在本文系统中不会再被进一步分析，本文系统只会把输入图片中对应区域保存为一个图片（这样输入文档中的彩色图像可以在输出中保留为彩色的），留待最终生成排版代码时引用。

现有的一些块分类方法基于统计或机器学习，常被选用的特征有游程统计量 [5] 和连通域统计量 [5] 等，而被选用分类方法有最近邻方法 [5] 和线性判别法 [5] 等，这类方法的缺点在于可理解性差且依赖于训练数据。另一些块分类方法则基于图像和表格通常含有较大的连通域的特点，例如把是否存在大连通域作为判断一个区域是否非文本区域的主要标准 [24]，或是利用腐蚀运算把文本去掉而让非文本留下 [25]，它们的缺点在于大字标题、大型定界符和根号等需要特殊处理以免被误判为非文本。总体上看后一类方法比较实用，通过适当地构造判别条件应该可以免除大部分特殊处理。

由于物理块间的位置关系是二维的，而文档的排版代码却是一维，因此需要对物理块进行排序，这个排序应当与通常人阅读文档的顺序一致。由于大多数科技文档是横排的，阅读的顺序为从上向下、从左向右，容易想像可以通过制定一些规则去决定各物理块的先后顺序，[6] 提出了用以下两条规则建立严格偏序再进行拓扑排序以得到阅读顺序：

- 若两个物理块 A, B 使 A 的纵坐标小于 B 的纵坐标而横坐标范围有重叠，则 A 先于 B。
- 若两个物理块 A, B 使 A 的横坐标小于 B 的横坐标，并且不存在纵坐标介于 A, B 之间的其它物理块 C 与 A, B 分别有重叠的横坐标范围，则 A 先于 B。

然而，不难发现这两条规则实际上不总能建立严格偏序，例如在各物理块的物



(a) 各物理块的物理矩形　　(b) “先于” 关系图

图 3.1: 表明 Breuel 规则不一定给出偏序之反例

理矩形如图3.1a所示时，由这两条规则建立的先于关系如图3.1b所示（仅显示直接用上述规则生成的边），其中存在环路 $ABCDA$，因此不可能扩充为严格偏序关系。不过由于这个方法的简单性和它失败的情况看来都很特殊，本文打算沿用拓扑排序的思路，但修正生成关系的方法以保证严格偏序性。

3.1.2 逻辑版面分析

逻辑版面分析阶段的目的是得出整个文档的逻辑结构。其中，由于暂不深入分析非文本块，非文本块均被认为是逻辑块，故关键在于把文本块进一步分解为更细的逻辑单元，这个过程称为文本块版面分析。

观察一些从文本块细分出的逻辑块类型及其版式特征：

- 题名
  题名一般在页面居中对齐，在一个作品中只有一个。
- 作者
  作者一般在页面居中对齐，并且紧跟在题名下方。
- 标题
  通常以特定格式的编号开始，有时会居中对齐。
- 段落
  段首往往有缩进而段末往往有空白区或特定标点符号，除独立行公式外中间各行左右边缘分别与所在文本块的左右边缘对齐。
- 列表条目
  编号列表和非编号列表分别有特定的开始模式，而且往往有缩进。
- 图表说明
  一般以 “图”、“表”、“表格”、“Figure”、“Fig.” 或 “Table” 等字样后接一个

数字编号开首，有时居中对齐。

容易注意到各种逻辑块均以文本行作为共同的组成单位，受以上观察启发，可以利用首行对齐方式和对文字识别结果进行正则表达式匹配来抓住各逻辑块的开首，然后再借助排版习惯生成整个逻辑块，于是可以建立一个从文本行组生成逻辑块列表的一趟算法。

诚然，表格也有内部结构的，可以分割为一些单元格。对于带边框的表格，可以利用投影方法进行分析 [9]。对于不带边框的表格，则可以用基于层次聚类的方法分析 [26]。然而，为了集中精力于主要问题，本文不打算专门研究表格分析技术，只把表格作为图片处理。

进行文本块逻辑版面分析后，可以简单地用串接方法整个文档的逻辑块列表。接下来的工作就是结果表示，即生成排版代码作为输出。

3.2 问题的解决方法

3.2.1 基本的物理版面分析方法

由于物理版面分析并非科技文档识别的特有问题，以下仅简单地说明可以如何采用和改造现成的方法以提供基本的功能。

关于物理版面分割，如前所述，容易实现的自顶向下方法已经足够对通常的科技文档进行物理版面分割，故可以暂时使用基于递归投影切分的方法进行物理分割，其中阈值的选择基于页面连通域的高度分布以适应不同分辨率的页面。与大部分其它自顶向下方法类似，基于递归投影切分的方法对噪声较为敏感，但由于模块化的设计方法，如有需要的话日后可以方便地支持更多物理版面分割方法。

关于块分类问题，因为文本块中通常没有特别大的连通域，可以把物理块最大连通域高度与平均连通域高度之比是否不超过某个事先给定的阈值作为区分一个物理块是否文本块的主要标准。当然，在非文本块中不存在足够大的连通域或在文本块中存在不明的大型符号时，这个检测算法可能会作出误判。设物理块中连通域个数为 k，则此分类算法的时间复杂度为 $\Theta(k)$。

关于阅读顺序排序，为了修正 [6] 中生成关系的方法以保证严格偏序性，首先注意到仅应用第一条规则的话可以保证严格偏序性，故可以放心地先应用它生成一个关系。随后为防止严格偏序性被破坏，虽然对每一对物理块检查它们是否满足第二条规则的条件，但只有在不会破坏严格偏序性的情况下它才会被应用。整个阅读顺序排序过程如算法3.1所示。假定有 k 个物理块，则这个严格偏序关系可以在 $O(k^4)$ 时间内完成构造，而拓扑排序可以在 $O(k^2)$ 时间内完成，故整个阅读顺序排序可以在 $O(k^4)$ 时间内完成。由于一个页面中物理块数目通常不多，故这个过程实际上很快。

算法 3.1: 阅读顺序排序

输入: 物理块集合 $\{R_0, \ldots, R_{k-1}\}$

输出: 物理块列表 L

数据结构: 邻接矩阵 A(初始为 I_k), 入度数组 d(初始全 0) 和栈 s

开始

/* 使用第一条规则 */

对于每个 $i \in \{0, \ldots, k-1\}$ **进行**

  对于每个 $j \in \{0, \ldots, k-1\}$ **进行**

    如果 $R_j.left \leq R_i.right \wedge R_i.left \leq R_j.right \wedge R_j.top < R_i.top \wedge A_{ji} = 0$ **则**

      $A_{ji} \leftarrow 1, d_i \leftarrow d_i + 1;$

      对于每个 $u, v \in \{0, \ldots, k-1\}$ **进行**

        如果 $A_{uj} = 1 \wedge A_{iv} = 1 \wedge A_{uv} = 0$ **则**

          $A_{uv} \leftarrow 1, d_v \leftarrow d_v + 1;$

/* 使用修正的第二条规则 */

对于每个 $i \in \{0, \ldots, k-1\}$ **进行**

  对于每个 $j \in \{0, \ldots, k-1\}$ **进行**

    如果 $R_j.right < R_i.left \wedge A_{ij} = 0 \wedge A_{ji} = 0$ **则**

      $A_{ji} \leftarrow 1, d_i \leftarrow d_i + 1;$

      对于每个 $u, v \in \{0, \ldots, k-1\}$ **进行**

        如果 $A_{uj} = 1 \wedge A_{iv} = 1 \wedge A_{uv} = 0$ **则**

          $A_{uv} \leftarrow 1, d_v \leftarrow d_v + 1;$

  如果 $d_i = 0$ **则** $s.push(i);$

/* 进行拓扑排序 */

当 s 非空 **进行**

  $i \leftarrow s.pop();$

  $L.add(R_i);$

  对于每个 $j \in \{0, \ldots, k-1\}$ **进行**

    如果 $A_{ij} = 1$ **则**

      $d_j \leftarrow d_j - 1;$

      如果 $d_j = 0$ **则** $s.push(j);$

3.2.2 面向行的逻辑版面分析方法

从文本块生成逻辑块列表可分为把文本块分解为文本行和从文本行组合出逻辑块两个子步骤。

把文本块分解为文本行组可以用常规的横向投影方法，如以下定义所示。

定义 3.5 设 B 为一个文本块，T 为 B 的一个规范的曼哈顿分割使对任何互异的 $T_1, T_2 \in T$，有 $\mathrm{Pr}_1(T_1) \cap \mathrm{Pr}_1(T_2) = \emptyset$，则称 T 为 B 的一个文本行分割，并称 T 的元素为 B 的文本行。

为了由文本行组生成逻辑块，把每个文本行分为以下五种类型之一：

文本行类型 0 文本行在页面居中。

文本行类型 1 文本行在页面不居中但在所在文本块中居中。

文本行类型 2 文本行在所在文本块中靠左对齐。

文本行类型 3 文本行在所在文本块中靠右对齐。

文本行类型 4 文本行与所在文本块基本上同宽。

按照排版惯例，同一逻辑块中普通文本行类型的转换关系可以由以下相容矩阵表示：

$$B = \begin{pmatrix} b_{00} & b_{01} & b_{02} & b_{03} & b_{04} \\ b_{10} & b_{11} & b_{12} & b_{13} & b_{14} \\ b_{20} & b_{21} & b_{22} & b_{23} & b_{24} \\ b_{30} & b_{31} & b_{32} & b_{33} & b_{34} \\ b_{40} & b_{41} & b_{42} & b_{43} & b_{44} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

其中 b_{ij} 为 1 当且仅当容许一逻辑块内两上下相邻的两个文本行依次分别为文本行类型 i 和文本行类型 j。再结合一个逻辑块的类型信息可从首行推断的观察，即可得到文本块逻辑版面分析算法3.2，若记文本块中文本行数为 k，则其时间复杂度为 $\Theta(k)$。其中，独立行公式的判定将在后面的章节讨论。

这样，把物理版面分析得到的物理块列表中的文本块换为对它作文本块逻辑版面分析所得的逻辑块列表，即得到整个页面的逻辑块列表。其中，每个文本块生成的逻辑块从上而下排列，这是符合阅读顺序的。

由于文档往往由相互联系的多页组成，为了完整准确地重构文档的结构，需要把各页的信息结合起来。首先列出一个简单的定义：

定义 3.6 设 $(P_1, \ldots, P_s)$ 为一列彩色图像，则称之为一个页数为 s 的文档，并且对 $i = 1, \ldots, s$，称 P_i 为它的第 i 页。

给定页数为 s 的文档 $(P_1, \ldots, P_s)$，对 $i = 1, \ldots, s$，记对页 P_i 进行图像预处理和前述的版面分析后得到的逻辑块列表为 L_i，把列表 $L_1, \ldots, L_s$ 依次串接起来得到列表 L。为了把跨页或被浮动体分开的段落重新连接在一起，还需要作一点处理：若 L 的第 k 和 $\ell (k < \ell)$ 个逻辑块分别为无结束的段落和无开始的段落，并且 L 的第 $k+1$ 到 $\ell - 1$ 个逻辑块均为浮动体，则把 L 的第 ℓ 个逻辑

算法 3.2: 文本块逻辑版面分析

输入: 文本块中文本行组 $(T_0, \ldots, T_{k-1})$, 页宽 w, 已发现题名标记 f

输出: 逻辑块列表 L

数据结构: 相容矩阵 B, 文本行组对应类号组 $(t_0, \ldots, t_{k-1})$

开始
  $i \leftarrow 0$;
  当 $i < k$ **进行**
    如果 T_i 被判为独立行公式 **则**
      $curr \leftarrow Paragraph(T_i)$
    否则如果 $T_i.text$ 具有图表说明前缀 **则**
      $curr \leftarrow Caption(T_i)$
    否则如果 $t_i = 0 \wedge \neg f$ **则**
      $curr \leftarrow Title(T_i), f \leftarrow \neg f$
    否则如果 $t_i = 0 \wedge L$ 最后一个元素为题名 **则**
      $curr \leftarrow Author(T_i)$
    否则如果 $t_i = 0 \vee t_i = 1 \vee (t_i = 2 \wedge T_i.width < \frac{w}{3} \wedge T_i.text$未以结句标号符号终止) **则**
      $curr \leftarrow Heading(T_i)$
    否则如果 $T_i.text$ 具有列表前缀 **则**
      $curr \leftarrow Listing(T_i)$
    否则
      $curr \leftarrow Paragraph(T_i)$
    当 $i + 1 < k \wedge (b_{t_i,t_{i+1}} = 1 \vee T_{i+1}$被判为独立行公式) **进行**
      $i \leftarrow i + 1$;
      $curr.addTextLine(T_i)$;
    $i \leftarrow i + 1$;
    $L.add(curr)$;

块内容加到第 k 个逻辑块的末尾再把第 ℓ 个逻辑块从列表中删除。其中，一个段落被认为有开始当且仅当它的首行为为文本行类型 3，而一个段落被认为有结束当且仅当它的最后一行为文本行类型 2。这个合并过程是直接的，只用顺序扫描 L 一趟，L 以链表表示时，整个合并过程的时间复杂度为 $\Theta(|L|)$。

最后的阶段为结果表示，本文系统支持的输出格式包括便于重新出版的 LaTeX 和便于在线发布的 HTML④，生成代码的方法是直接的，词法分析和语法分析的标准技巧可以用于简化生成的代码。

3.3 问题的解决情况

为了评估本章版面分析方案的准确程度，使用 MediaTeam Oulu 文档数据库 [27] 中的部分文档进行性能评估。由于本文的目的，仅使用其中类别为文章、手册、数学、程序和地理的 274 个文档。

本文系统的递归投影切分算法对其中 164 个文档 (占总数的约 59.9%) 给出了可接受的物理版面分割 (即把文本内容和非文本内容分开，把不同栏的文本分开，但不把同一栏中的行切开)。虽然这个结果并不让人满意，但由于数据库中不少文档有复杂版面，对于真实的科技文档，可以预期本文系统会给出更好的物理版面分割效果。

在人手修正物理版面分割后，使用本文系统的块分类算法区分各物理块类型，实验结果见表3.1，文本块的识别准确率达到 98.7%，文本块的召回率达到 99.3%。可见，该算法能较好地工作，但有时也确实会作出误判。其中，导致把文本块误判为非文本块的主要原因是使用了艺术字体，而导致把非文本块误判为文本块的主要原因是不存在单一大连通域。

表 3.1: 物理块分类的混淆矩阵

	被判为文本块	被判为非文本块
文本块	1149	8
非文本块	15	225

此外，本文系统的阅读顺序排序算法在实验中对 265 个文档 (占总数的约 96.7%) 都给出了合理的阅读顺序。这说明该算法基本可用。导致出现错误的主要原因为有行优先的表格式版面和有报纸式的复杂版面。

在文本行提取方面，实验中发现文本块中文本行总数为 15749，基于投影的行提取未能正确把两行分开的情况只出现了 129 次，而把一行切开的情况也只出现了 97 次，可见通过投影从文本块提取文本行还是很可靠的。其中，导致前一类错误的主要原因包括倾斜校正未如理想、段首字符跨行和存在行内公式，而导致后一类错误的情况包括把 “i” 或 “j” 上的小圆点分开、把帽子与字符切开和把求和号与其上下限切开。

由于版面分析由多个子过程组成，即使每个子过程都较可靠，总体上也可能表现为很不可靠。在目前还不能让版面分析自动地完美完成的情况下，容许用户交互式地参与版面分析过程是一种现实的折衷办法。即使有时可能需要人手干预，比起完全人手操作，节省的工作量也是不容低估的。

3.4 本章小结

本章以实用的角度讨论了版面分析技术，覆盖了物理版面分析和逻辑版面分析。在物理版面分析方面，本章说明了本文系统的版面分析模块在物理版面分割、阅读顺序排序和物理块分类所作的简单选择，本章特别指出和修正了 [6] 中阅读顺序排序方法中的一个缺陷。在逻辑版面分析方面，本章提出了一种利用行缩进和正则表达式匹配的文本块逻辑版面分析方法，并指出了重新连接跨页或跨浮动体段落的方法。由于目前版面分析技术还达不到高度准确的程度，因此容许人手观察中间结果和及时修正错误可能使系统更为有用。

第四章　字符识别

4.1　问题的提出和分析

由于字符为文本块的基本组成单位，字符识别是还原文档内容的关键一步，也是困难的一步。对于本文的目的而言，调用 Tesseract[28] 或 GNU Ocrad[29] 等外部字符识别引擎的能力虽然在系统开发早期对及早形成可用性有重要作用，但外部字符识别引擎并未能提供字体、上下标和缩进等对较细致地重构文档有用的信息，单方面扩充这些外部引擎则会带来维护上的困难，因此开发一个原生的字符识别系统仍是有必要的。

现有的大部分字符识别算法都要求先将各个字符分离出来，字符分割的主要手段有：

- 纵向投影 [8]
  基于相邻字符间一般有空白区分隔的观察，可以取文本行纵向投影的极小值点作为分割点。然而，纵向投影不能有效地分割斜体字符，有时会误切左右结构的字符。
- 连通域分析 [8]
  基于字符一般由一些连通域组成的观察，可以先进行连通域分割再合并。然而，连通域分析不能单独处理字符粘连、断笔等情况。
- 凹角分析 [28]
  对于粘连字符，可以利用轮廓多边形近似的凹角选取候选切分方案，再根据识别结果评价切分。

由于数学公式中常有斜体字符和二维结构，纵向投影对于数学符号识别的目的而言并不合适。而由于粘连字符的出现频率仅约 1%[30]，在字符识别系统的准确率达到 98% 前粘连字符并不足以成为提高准确率的主要障碍。因此，本文以连通域分割作为字符分割手段。注意到物理矩形明显相交的连通域一般属于同一字符（假定它们都不是根号），可以把这样的连通域合并为字形。而由于数学公式中常出现多种语言、字体和大小的字符混排的情况，难以仅用连通域的几何位置信息准确地合并为字符。因此本文选择以字形而非字符或连通域作为基本识别单位，这样既避免了早期合并为字符的困难，而待识别的基本单位个数又低于连通域的个数。

虽然以字形而非字符作为基本识别单位，但字形识别方法可参照字符识别

的方法。由于不容易定义稳定的骨架 [31]，本文并不对字符进行细化。由于轮廓是易碎的 [31]，本文也不对字符进行轮廓提取。进行模板匹配固然可以完全利用信息，适合作为验证和多候选排序的主要依据，但计算量也较大。因此，需要使用某种筛选策略把大部分候选先筛掉以降低进行模板匹配的次数，一个简单的筛选策略就是从初始候选集出发，依次应用一系列匹配器，每个匹配器把一些的候选去掉。

和常见的分类器类似，每个匹配器的标准工作流程分为特征提取和分类两步。在特征提取方面，本文选择了低阶矩 [10]、投影 [10]、孔洞数 [9]、高宽比 [9]、穿线数 [9] 和网格特征 [9] 这几个较直观、容易计算且不要求很专门的分类方案的特征进行实验，再根据实验结果决定在本文系统中默认选用什么特征进行匹配。分类器的选择一定程度上受选用的特征影响：由于孔洞数和穿线数是离散的特征，可以进行精确匹配；由于高宽比只用于粗分类，只要求比值接近于 1；由于矩和网格特征为固定维数的特征，可以自然地计算距离，从而可以用最近邻分类器。

此外，数学公式中有些符号不能仅由字体模板通过缩放得到，还涉及局部延长 [32]。这些符号包括定界符（如“|”、“(”、“)”、“[”、“]”、“{”、“}”、“⌊”、“⌋”、“⌈”、“⌉”、“⟨”、“⟩”）、根号（如“$\sqrt{}$”）、箭头（如“$\longrightarrow$”）和水平括号（如“$\overbrace{}$”、“$\underbrace{}$”）。为了识别这些符号，使用基于缩放不变特征的方法是不合适的。为了避免人手设计可以惟一标识各个特殊符号的结构特征所带来的繁琐工作和与常规字符识别方法的不协调性，本文提出动态生成模板的方法，即根据待识别字形的特点（主要是高宽比）生成各个特殊字形的模板，然后让待识别字形与各个特殊字形模板进行模板匹配。为了节省开销，可以只在待识别字形按常规方式得不到理想识别或高宽比悬殊时才启用特殊字形识别。

最后一个实际困难在于需要标号数据作为训练集，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，且容易扩充支持的字体列表。因此，本文系统暂时利用字体文件和代码点与字符名称的对应表来生成用于识别的数据。当然，由于训练集缺乏退化数据，可能产生过度拟合，这个问题只能留待日后具备条件后再通过扩充数据集解决。

4.2 问题的解决方法

4.2.1 字形匹配器的构造

按照本文的筛选策略，需要用到一些匹配器以去除部分候选。以下给出用于匹配各种特征的匹配器的具体构造，它们以待识别字形 G 和候选字形集 $\mathcal{G}$ 作为输入，并以 $\mathcal{G}$ 的一个子集作为输出。

- 孔洞数匹配器

孔洞数即字形中背景像素集的连通域个数（不算外围的），这是一个拓扑不变量。孔洞数的匹配采取精确匹配，记 $\mathcal{G}_h$ 为与 G 有相同孔洞数的所有已知字形的集合，则以 $\mathcal{G}_h \cap \mathcal{G}$ 作为输出。

- 穿线数匹配器
  对每个 $i \in \mathbb{Z}$，令

$$c_i = |\{j \in \mathbb{Z} | (i,j) \in G \wedge (i,j+1) \notin G\}| \tag{4.1}$$

  ，把序列 $(c_i)_{i=-\infty}^{\infty}$ 中相邻的重复元素都删除掉即得到 G 的横向穿线数序列。对称地有纵向穿线数序列。穿线数序列的匹配采取精确匹配，记 $\mathcal{G}_c$ 为与 G 有相同穿线数序列（包括纵向和横向的）的所有已知字形的集合，则以 $\mathcal{G}_c \cap \mathcal{G}$ 作为输出。
- 高宽比匹配器
  对于字形 C，令其高宽比为

$$ar(C) = \frac{\max_{(i,j)\in C} i - \min_{(i,j)\in C} i + 1}{\max_{(i,j)\in C} j - \min_{(i,j)\in C} j + 1)} \tag{4.2}$$

  ，则以

$$\left\{ G' \in \mathcal{G} \middle| \frac{4}{5} ar(G) \le ar(G') \le \frac{5}{4} ar(G) \right\}$$

  作为输出。
- 网格特征匹配器
  把像素矩阵通过网格分为 mn 格（在本文系统中取 $m = n = 3$），对 $i = 1, \cdots, m; j = 1, \cdots, n$，记格子 (i,j) 中 G 点数为 N_{ij}，而面积为 A_{ij}，则前景像素密度为

$$d_{ij} = \frac{N_{ij}}{A_{ij}} \tag{4.3}$$

  ，归一化后前景像素密度为

$$d'_{ij} = \frac{d_{ij}}{\sum_{r=1}^{m} \sum_{s=1}^{n} d_{rs}} \tag{4.4}$$

  。于是得到一个 mn 维特征向量

$$D(G) = \frac{1}{mn}(d'_{11}, \ldots, d'_{1n}, \ldots, d'_{m1}, \ldots, d'_{mn}) \tag{4.5}$$

  ，可以用绝对值距离比较。输出为

$$\left\{ G' \in \mathcal{G} \middle| 1 - \|D(G) - D(G')\|_1 > \frac{9}{10} \left(1 - \min_{G'' \in \mathcal{G}} \|D(G) - D(G'')\|_1 \right) \right\}$$

  。

- 矩匹配器

  本文使用前两阶归一化中心矩构造特征向量，这特征在平移和尺度变换下不变 [17]，它们可以用带权的绝对值距离比较。其中，各分量的权值为数据库中该分量标准差的倒数，这样选取权值是为了消除各分量在数量级上的差别。根据距离生成输出的方法与网格特征匹配器类似。
- 投影匹配器

  对每个 $i \in \mathbb{Z}$，令

$$p_i = |\{j \in \mathbb{Z} \mid (i,j) \in G\}| \tag{4.6}$$

  ，序列 $(p_i)_{i=-\infty}^{\infty}$ 就是 G 的横向投景。对称地有纵向投影。投影作归一化处理后也可以定义绝对值距离。根据距离生成输出的方法与网格特征匹配器类似。

上面仅仅给出匹配器的一些可能的构造，完全可以构造别的匹配器，只要满足输入输出规范即可在这个字符识别框架下使用。

4.2.2 基于字形匹配器的字符识别方法

字符识别算法以文本行中连通域列表为输入，而以字符（作为前景像素集）与候选集的对应表作为输出。

算法首先通过合并连通域生成字形，合并规则为如果两个连通域物理矩形之交的面积大于面积较小者面积的 $\frac{1}{5}$，并且两个连通域都未有被判为根号，则这两个连通域属于同一字形。合并过程采用带路径压缩和按秩合并的不相交集合算法 [33]。

然后，每个字形按它被判为横线、圆点还是其它来分配初始候选集（这样区分是因为对于宽度或高度很小的横线和小圆点，由于离散化造成的误差是不可接受的），这种区分的依据为以下的经验规则：

- 若一个连通域的宽度为其高度的五倍或以上且在其物理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 $\frac{5}{4}$ 之间且在其物理矩形中像素密度达到 70%（注意一个圆与其外接正方形的面积比为 $\frac{\pi}{4} \approx 0.7854$），则被认为是圆点。

接着依次应用各匹配器以缩小候选集。在应用所有匹配器后，对于候选集中每个属于分体字符的字形，搜索该分体字符的其它字形是否都在相应位置出现，如全部出现则计算这些待识别字形合并后与该字符的 Hausdorff 距离；对于其它字形，则直接计算待识别字形与字形间的 Hausdorff 距离。如果最小的 Hausdorff 距离大于一个阈值，则启用特殊字形识别。特殊字形识别的方法是对于每个已知特殊字形，如果可能的话，生成它具有待识别字形的高宽比的实例，并计算与待识别字形间的 Hausdorff 距离，再在候选集中加入相应的候选。

这种识别方案不仅可识别复杂二维结构中的字符，而且可通过改变匹配器组合定制，具备相当的灵活性和可扩展性。

4.3 问题的解决情况

为了评估本章字符识别方案的效果，本文利用的 AMSFonts[34] 字体中字符作为训练集和测试集，包括 CMB10、CMBSY10、CMEX10、CMMI10、CMMIB10、CMR10、CMSY10、EUFB10、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），共包含了 1131 个符号。生成识别数据后，利用一个自动化测试程序生成完美的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，不加滤波，训练用字体大小为 40。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。

部分实验结果见表4.1和表4.2。根据实验结果，对于各种特征，有以下观察：

表 4.1: 字符识别的识别率

匹配器组合	待识别字体大小				
	10	20	30	40	50
穿线数	17.06%	42.53%	54.64%	99.20%	61.27%
网格	20.69%	75.42%	85.94%	99.20%	93.72%
矩	13.97%	76.75%	86.91%	99.29%	94.25%
投影	16.89%	78.78%	88.15%	99.20%	93.55%
网格 + 矩 + 投影	18.30%	77.72%	87.44%	99.20%	94.96%
高宽比 + 矩 + 投影	16.18%	77.90%	88.15%	99.20%	94.61%
高宽比 + 网格 + 矩	19.36%	76.22%	86.74%	99.20%	93.99%
高宽比 + 网格 + 投影	21.04%	77.90%	87.62%	99.20%	94.25%
高宽比 + 网格 + 矩 + 投影	20.51%	78.34%	87.62%	99.20%	94.96%
孔洞数 + 高宽比 + 网格 + 矩 + 投影	17.15%	64.46%	80.11%	99.20%	89.83%
穿线数 + 高宽比 + 网格 + 矩 + 投影	16.27%	42.53%	54.47%	99.20%	61.63%

- 高宽比的应用可能稍为提高或降低准确率，但通常可缩短识别用时。
- 孔洞数并不稳定，可能需要先行设法填补小孔洞。
- 穿线数序列也不稳定，不适合用于进行精确匹配，可能应考虑改用编辑距离。
- 单独应用时网格特征、矩和投影的区分力接近，但使用网格特征的用时明显较短。

表 4.2: 平均识别每个字符的用时（毫秒）

匹配器组合	待识别字体大小				
	10	20	30	40	50
穿线数	3.94	2.95	3.97	4.57	7.60
网格	4.80	6.03	7.80	5.32	14.13
矩	7.27	20.84	35.84	39.74	78.40
投影	7.32	10.53	14.28	9.95	23.35
网格 + 矩 + 投影	3.72	2.97	3.26	3.06	4.94
高宽比 + 矩 + 投影	3.69	3.71	4.24	3.49	6.74
高宽比 + 网格 + 矩	4.36	3.55	4.25	3.73	7.44
高宽比 + 网格 + 投影	3.56	2.82	2.93	3.02	4.64
高宽比 + 网格 + 矩 + 投影	3.56	2.68	2.80	2.91	4.35
孔洞数 + 高宽比 + 网格 + 矩 + 投影	3.65	2.71	2.86	2.88	4.34
穿线数 + 高宽比 + 网格 + 矩 + 投影	3.32	2.20	2.56	2.42	4.11

- 使用多个匹配器可以缩小候选集，减少较耗时的模板匹配，从而往往可缩短用时。

根据实验结果，本文系统默认启用高宽比匹配器、网格特征匹配器、矩匹配器和投影匹配器作为默认的匹配器组合，这较好地平衡了准确率和用时。但必须承认，在低分辨率条件下字符识别的准确率并不理想，这个实验未纳入中文字体也带来了局限性。

4.4 本章小结

本章在回顾现有字符识别技术的基础上，提出了一种以字形为基本识别单位并采取层层筛选再验证策略的字符识别方案，这种方法可处理具有复杂二维关系的字符且具有一定的可扩展性。对于个别特殊字形，则采用动态生成模板的方法加以匹配。实验表明，通过选择合适的匹配器组合，可以取得值得注意的准确率和识别速度。然而，为了达到可投入日常使用的要求，还需要更多工作。

第五章　数学公式识别

5.1　问题的提出和分析

科技文档的一个特点是经常含有数学公式且往往是其中重要的组成部分，人手输入数学公式又恰好是特别繁杂和容易出错的工作，因此正确识别数学公式对于识别科技文档有重要意义。然而，与通常的字符识别和版面分析相比，数学公式识别面临着更多的困难：

- 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），需要用语义信息来区分。
- 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。

5.1.1　数学公式定位

为了识别数学公式，传统方法要求首先把它们从普通文本中分离出来。按照排版时是否单独占用一行，数学公式可分为独立行公式和行内公式。

对于行内公式，由于缺乏明显的起止标记，定位问题较为困难。目前已提出的方法大致分为两类：

- 基于识别信息的方法
  由于一些字符类型在公式和非公式中出现频率悬殊（[30] 给出了一个量化分析），可以根据符号的识别信息区分它是否公式的特有字符，然后通过区域生长提取整个行内公式。其中，对于中文文档可以把被汉字识别系统拒识的字符视为公式符号 [9]，区域生长的方法则可以利用预期操作符位置 [35]。
- 基于二维结构的方法
  由于数学公式中常有上下标等不常见于非公式文本的二维结构，这些结构的存在可作为发现行内公式的标志。例如已提出对于每个利用纵向投影得到子串，根据基线异常字符的比例判断是否行内公式 [12]。然而，这类方法不能发现无二维结构的简单公式。

为了定位行内公式，应当综合利用符号类型转换和二维结构信息，从而分别克服上述两类方法各自的局限性。

对于独立行公式，目前已提出的提取方法大致分为两类：

- 基于版式特征的方法
  由于独立行公式往往具有行高较高、居中和带编号等版式特征，可以根据这些版式特征判断一个文本行是否独立行公式。例如已提出利用 Parzen 分类器根据行高、上下行间距、左右行缩进和公式末尾编号区分独立行公式 [12]。这类方法容易高效地实现，但对于排版方式敏感。
- 基于内容的方法
  由于独立行公式内部往往有一些不常见于其它文本行的二维结构，可以根据字符间的邻接关系区分独立行公式。例如已经提出利用连通域邻接图的特点区分独立行公式 [35]。这类方法较能把握数学公式的本质，但计算较复杂。

为了定位独立行公式，可以先用版式特征把一些明显的独立行公式区分出来。对于余下的文本行，再进行行内公式定位，如果整个文本行被判定为一个行内公式，则也把该文本行改判为独立行公式。由于行内公式定位利用了内容的信息，这样既可以在一些情况下利用基于版式特征方法的高效，又能不损失基于内容方法的优点。

5.1.2 结构分析

结构分析即把各个符号组合为用合适数据结构描述的完整数学公式，其中关键在于确定符号间的关系。

结构分析方法按分析方向可分为：

- 自顶向下方法
  自顶向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。这类方法较能把握整体结构，适合处理分式、根式和矩阵等结构。
- 自底向上方法
  自顶向下方法逐步地把符号合并为子公式，直至得到整个数学公式。这类方法较能把握局部结构，适合处理上下标等结构。

为了结合这两类方法的优势，应当采用双向方法，综合利用局部信息和整体信息，以便识别种类广泛的数学公式。

结构分析方法按使用的手段则可分为：

- 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。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 [14]。虽然这类方法可避免产生一些语法上不合理的识别结果，但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，故这类方法较适合用于识别领域特定的数学公式。

- 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 [36]、基于特征字符的合并方法 [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 [37]、估计基线结构 [38] 等等。

上述方法虽然有各自的局限性，但也能分别把握数学公式的一些特征。例如递归投影切分不适合处理上下标，但能有效地处理分式和矩阵；基于特征字符的合并可利用不同符号的特点，但不太便于扩充；对于连接赋予合理权值是困难的，但使用图表达邻接关系是有普遍意义的；直接提取各基线容易出现漏识情况，但利用基线判定上下标仍然是可取的。受这些方法启发，可以利用类似投影的方法估计分数线和大型操作符等符号的作用范围，利用图表达左右邻接关系，利用符号类型引导合并过程，利用基线信息判定上下标关系。把这些方法整合起来，就可以得到一种基于符号邻接图的数学公式结构分析方案，让各自的长处消融在符号邻接图的构造和改写之中。

由于很多被提出的结构分析方法并未给出量化的实验结果，即使有的也要么基于很小且不可得的数据库，要么仅评估各种局部结构的分析情况。为了客观地评价结构分析算法的整体可靠程度，需要使用公开的数据库进行性能评估，以求得到整个数学公式的结构分析正确率。

5.2 问题的解决方法

5.2.1 基于符号类型和二维结构的数学公式定位方法

以下给出定位行内数学公式和部分独立行公式的具体方法。

为了定位行内数学公式，首先把文本行按空白和符号类型转变位置分为若干列，把内有数学公式特有符号或存在二维结构的列认为是数学公式，然后把相邻的数学公式合并。

为了利用版式特征定位部分独立行公式，注意到仅用居中性、行高和行距等版式特征并不能有效地区分独立行公式和标题，故只用编号比较保险。对于编号的数学公式，由于编号一般在文本块右对齐或左对齐（与通常的列表项目不同），并且编号与公式内容间有较大的空白（与通常的标题不同），可以利用这两点把编号公式区分出来。

这种数学公式定位方法的主要不足在于它依赖于字符识别结果，而本文的字符识别技术并不是很成熟。

5.2.2 基于符号邻接图的数学公式结构分析方法

在数学公式结构分析过程中，需要表达子公式和它们间的关系，以便进行合并，而图正是一种适合这目的的数据结构。在符号邻接图中，每个顶点表示一个子公式，而每条边表示有左右邻接关系（含上下标关系）。为了方便操作，每个子公式记录了物理矩形、外接矩形、基准点和排版代码等信息。这样，数学公式结构分析就可以分为构造符号邻接图和改写符号邻接图两步。

为了构造符号邻接图，有一些准备工作。首先，为了区分一些在符号识别阶段未能区分的相似符号，需要利用其它符号的信息。例如，为了区分“.”与“·”、“,”与“’”等相似符号，可以匹配基线。又例如，为区分减号、上划线、下划线和分数线，可以检验其物理矩形的上方邻近和下方邻近分别是否有其它符号。

其次，为了在确定左右邻接关系时避免一些非法情况，对于分数线、上下划线、上下括号、帽子、根号和大型操作符等特别符号，有必要界定相应的作用范围，即预期的操作数位置。分数线、上下括号和大型操作符分别有上下两个作用范围，下划线只有一个上作用范围，上划线和帽子只有一个下作用范围，而根号有主作用范围和次数作用范围。具体的作用范围可以利用特别符号的位置信息和区域投影得到。为方便见，对于特别符号，把包含物理矩形与它的所有作用范围的最小矩形称为外接矩形；对于其它符号，外接矩形是指其逻辑矩形；对于一般的子公式，外接矩形是指包含所有组成符号的外接矩形的最小矩形。

为了初始化符号邻接图，自然地通过让每个符号分别作为顶点来构造初始顶点集。而为了找出一些可能的左右邻接关系来构造初始边集，对每个非特别符号，往右扫描符号集以找出与之有相交外接矩形纵坐标范围的符号，其中靠左的就很可能与之有左右邻接关系，但禁止产生跨越特殊符号的作用范围边界的边。

剩下的工作为对符号邻接图进行改写。由于整体信息的使用，本文给出的图改写规则数目远少于已被提出的 60 条 [39]。以下按优先应用顺序（大致是从较可靠到较不可靠）列出这些规则：

1. 合并简单地左右相邻的子公式
   若符号邻接图中有有向边 $e = (u, v)$，使 u 出度为 1 且 v 入度为 1，则合并 u 和 v，合并后顶点的入边集和出边集分别为 u 的入边集和 v 的出边集，而对应子公式由 u 和 v 对应子公式连接得到，其中水平和上下标相区分的主要标准为参考点的位置。
2. 合并更多上下标
   假定符号邻接图中有出度为 2 的顶点 u 使其以出边邻接的顶点 v 和 w 均有入度 1，如果 v 和 w 的出度都是 0 或者出度都是 1 且有共同的出边终点，则合并 u、v 和 w，其中 v 和 w 中参考点纵坐标较小者为上标而另一个为下标。对称地，可处理左方的上下标。
3. 按宽度从小到大处理特别符号：

- 对于分数线、上括号和下括号，如除宽度不小于它的特别符号外，分别恰有一个子公式的外接矩形与其上、下作用范围相交，则与这两个子公式合并。
- 对于上划线、下划线或帽子，如除宽度不小于它的特别符号外，恰有一个子公式的外接矩形与其作用范围相交，则与这个子公式合并。
- 对于根号，如除宽度不小于它的特别符号外，恰有一个子公式的外接矩形与其主作用范围相交，并且至多有一个子公式的外接矩形与次数作用范围相交，则与这些子公式合并。
- 对于大型操作符，如除宽度不小于它的特别符号外，恰有一个子公式的外接矩形与其下作用范围相交，并且至多有一个子公式的外接矩形与其上作用范围相交，则与这些子公式合并。

进行合并后，还需要更新符号邻接图以反映新增的左右邻接关系。

4. 合并矩阵

   对于表达式以 (、[、{、⟨ 或 | 结尾且出度至少为 2 的顶点，利用出边指向顶点的外接矩形估计行结构，然后利用投影方法估计单元格结构，最终合并出矩阵。

5. 消除部分边

   在符号邻接图中有时会存在一些对应不现实邻接关系的边，一个征兆就是符号邻接图对应无向图中存在回路。为了消除这种情况，可以检测回路，每发现一条回路就把回路中对应最长水平距离的边去除。利用不相交集合算法，记符号邻接图中原顶点数和边数个数为 n 和 m，则消除所有回路可以在时间 $\Theta((m+n)\alpha(m+n))$ 内完成，其中 α 为 [33] 中定义的一个增长极其缓慢的函数。

图改写算法的框架如算法5.1，由于除最后一次外层循环外，每次外层循环使 $(|V|, |E|)$ 按字典序严格递减，故算法必在有限步内终止。如果改写后符号邻接图只有一个顶点，则它对应的子公式即为整个数学公式，否则识别失败。注意到不但可以通过增加规则以支持更多数学公式类型，而且可以通过改写强度控制结果的可靠程度（较低的改写强度可阻止一些较不可靠规则的应用，从而可能造成更多识别失败，但对于错识较拒识会造更坏影响时这是有用的），可见这个结构分析方法具备相当的灵活性。

在上述图改写框架下，还可以作一些改进。首先，由于大部分帽子都只作用于单一个符号，可以事先把这类帽子与它所作用于的符号合并，这样不但可以通过及早降低顶点个数来节省计算量，而且有时可以预防一些误合并。其次，在发生识别失败时，可以删除符号邻接图中所有边再重新构造边集，然后再尝试进行图改写，这样做不会影响原来已成功识别的数学公式的识别结果，但可能可以通过消除一些早期的错误决策而使图改写过程得以进行下去。

算法 5.1: 符号邻接图改写

输入: 符号邻接图 $G = (V, E)$，改写强度 k

输出: 经改写的符号邻接图 G

数据结构: 修改标记 `changed`

```
1 开始
2     changed ←true;
3     当 changed 进行
4         changed ←false;
5         对于每个 i ∈ {1, ..., k} 进行
6             如果 应用图改写规则 i 导致变化 则
7                 changed ←true;
8                 break;
```

5.3 问题的解决情况

为避开字符识别而独立评估上述结构分析方法的性能，使用 Infty 项目的 InftyCDB-2 数据库 [40] 中的 4400 个公式作为测试集进行测试。这个数据库给出了公式中各个符号的位置信息及所属的字符类别，可以以这些数据作为输入运行本文系统的结构分析算法，并以人手判断结构分析结果是否正确。

实验结果如表5.1所示，可见本文系统对于各类数学公式都取得了一定的识别率，但仍有不小的改进空间。

表 5.1: 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	2692	2096	77.9%
含根号表达式	64	47	73.4%
含分式表达式	676	391	57.8%
含帽子表达式	769	450	58.5%
含大型操作符表达式	714	310	43.4%
总计	4400	3060	69.5%

更仔细的观察发现导致错误的常见原因包括：

- 上下标关系被误判。
- 作用范围估计不准确。
- 数学公式中存在系统未支持的结构。
- 数据库本身存在错误。

此外，值得注意的是，实验中对全部 4400 个数学公式进行结构分析仅用时

3041 毫秒，平均每秒处理了约 1447 个数学公式。可见，与字符识别过程相比，结构分析过程非常快速。

5.4 本章小结

本章讨论了数学公式识别中数学公式定位和结构分析两大问题。对于数学公式定位问题，本章提出了一种基于二维结构和符号类型的数学公式定位方法，它克服了仅检测二维结构的方法无法检测简单公式的不足，又克服了仅用符号识别信息的方法无法检测文字公式的不足。对于结构分析问题，本章提出了一种基于对符号邻接图进行图改写的结构分析方法，这种方法不但可全面利用局部信息和整体信息，而且具有便于扩充的特点，并有良好的运行速度。应当指出，本章提出的数学公式结构分析方法并不依赖于前述的字符识别方法，例如其输入也可以来自解析 PDF 文件。

第六章　结论

6.1　取得成果

本文从构造一个科技文档识别系统的愿望出发，立足全局地对所涉及的问题作出了系统的讨论，最终实现了原先的设想。

本文以实用的观点对科技文档识别中的各个环节都进行了一些探索，综合考虑了准确性、效率和可扩展性，形成了基本完整的技术链：

- 在图形预处理阶段，使用数字图像处理的技巧以求尽可能恢复文档的原貌。
  - 交互式地确定页框的机制被用于去除边缘噪声。
  - 二值化被用于去除背景噪声，其中 Sauvola 方法为首选，Otsu 方法为备选。
  - 滤波器被用于去除椒盐噪声，可选的滤波器有多种。
  - 交互式地修改连通域集合的机制被用于去除不规则噪声。
  - 使用传统的投影方法进行倾斜检测，实验表明它在七种倾斜检测方法中有较高的准确性，效率也不差。
- 在版面分析阶段，逐步把文档分解为各种逻辑块并决定它们的先后顺序。
  - 基于递归投影切分的方法被选作物理版面分割方法，它有较高的效率，适合于处理较简单的科技文档版面。
  - 基于连通域高度分布的方法被用于区分文本块与非文本块，实验表明这个简单方法也是快速而准确的。
  - 基于拓扑排序的方法被用于阅读顺序排序，本文改进后的关系生成方法保证了拓扑排序的合理性，实验结果也说明了其可靠性。
  - 横向投影被用于提取行，实验显示它还是比较可靠的。
  - 基于行对齐和识别结果的方法被用于文本块逻辑版面分析，这个方法不但高效，而且便于扩充。
- 在字符识别阶段，需要分割出各个字符并确定它们分别是什么。
  - 以字形而非字符为基本识别单位，适应数学公式中字符间有复杂位置关系的特点。
  - 字形的识别使用从初始候选集开始用多个匹配器依次进行筛选的策略。
  - 基于 Hausdorff 距离的模板匹配被用于多候选排序的依据。

  - 可局部伸缩的特殊字形通过动态生成模板的方法匹配。
- 在数学公式识别阶段，需要把数学公式找出来并分析其结构。
  - 版式特征被用于快速定位编号公式，而其它公式利用二维结构检测和符号类型定位，从而结合了过往各种数学公式定位方法的长处。
  - 基于符号邻接图的方法被用于数学公式结构分析，这个方法不但高效和可扩展，而且实验表明即使对于复杂的公式也有一定的准确率。

更为重要的是，按照本文思路实现了一个演示性质的科技文档识别系统—MathOCR，它接受以 JPEG、PNG、GIF、BMP 或 PNM 等格式提供的印刷体科技文档图像作为输入，而输出为 LaTeX 或 HTML 格式的排版代码。这个软件的主要特点包括：

- MathOCR 是极少数支持逻辑版面分析和数学公式识别的文档识别系统之一。
- MathOCR 容许用户交互式地修正不同层次的识别错误。
- MathOCR 采用模块化设计，可以方便地进行定制和扩充。
- MathOCR 是一个纯 Java 软件，具备良好的跨平台特性。
- MathOCR 是一个自由软件，可以自由使用、研究、修改和散布。

6.2 未来展望

虽然已经建立了一个文档识别系统的设计并予以实现，但它在多方面仍有待改进。以下是一些想法和可能性：

- 更细致的预处理
  噪声不论对于版面分析还是字符识别都会带来不良影响，但已知的噪声去除手段似乎针对性都不强，故噪声去除仍需要更深入的研究。此外，本文系统未能纠正非线性几何畸变。
- 基于案例的版面分析
  本文使用的版面分析算法主要只用到单一页面的信息，但利用同一出版物中不同页面的版面一致性，如果能建立一个版面数据库，可以自动根据图片特征动态地选择相应的版面参数进行版面分析，将有利于提高版面分析的准确程度。
- 识别更多类型的对象
  本文系统尚不能正确识别表格、算法、代码、注释、页码引用、数学中的交换图、化学中的分子结构式和其它一些对象，加入对它们的识别支持将让系统的功能更为完善。
- 加入对粘连和断裂字形的处理
  本文系统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

的连通域合并。

- 借用成熟的字符识别技术
  本文系统的字形识别方法几乎是从零开始构造的，如果把字形识别的任务分派给专门的字符识别系统（必须是可训练的），既可以获益于较成熟的识别技术，又可以不损失以字形为基本识别单位的优点。
- 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，需要设法自动化地验证识别结果是否正确。
- 字符识别结果修正
  通过建立某种语言模型，从而发现和修正一些明显的误识，提高字符识别的准确率。
- 数学公式结构的自学习
  目前的数学公式结构分析中用于构造和改写符号邻接图的规则依赖于对数学公式结构的先验知识，如果能改为自动地从样本提取规则，将有利于提高可扩展性。
- 手写文献的自动识别
  本文只针对印刷体文档，但也有不少历史文献是手写的，把这些文献电子化也是有意义的。但手写体远比印刷体更不规范，有时用人眼辨认也有困难，且常伴有意图不明的标记，可以合理地预期，这个问题极具挑战性。

依托印刷体科技文档识别技术，通过减少重复性的输入工作使得大规模地电子化文档成为了可能，在此基础上可以构建或完善多种有实用价值的应用，例如文献检索系统、论文相似性检测系统、事实库自动生成系统、定理自动发现与验证系统等等。其中，数学公式识别技术不但有助把数学公式作为关键词搜索，而且也许能够弥补以往论文相似性检测对数学公式束手无策的遗憾。

不过，在为印刷体科技文档识别技术取得进展而感到欣慰的同时，更应该为这种技术的存在而感到悲哀。从根本上看，既然现在绝大多数科技文档本来就是用计算机排版的，只要公开原始的可编辑形式即可消除大部分文档识别的需求，它们没有被公开的原因更多在经济层面而非技术层面。最终，让文档识别技术连同产生它的需求一起消亡才是最希望看到的。

注释：

[①]在本文中的自然数均假定取自自然数的标准集合论模型，即 $0 = \emptyset$，而对 $n \in \mathbb{N}$ 有 $n + 1 = n \cup \{n\}$。

[②]按照数字图像处理的标准约定，本文所用的坐标系均为屏幕坐标系，即横轴向右为正而纵轴向下为正。

[③]本文所谈矩形的边均平行于坐标轴。

[④]在个别细节上目前可能仍未完全符合标准。

参考文献：

[1] SIMONE M. Introduction to document analysis and recognition[A]. Machine learning in document analysis and recognition[M], Berlin: Springer, 2008:1–20.

[2] FARAHMAND A, SARRAFZADEH A, SHANBEHZADEH J. Document Image Noises and Removal Methods[A]. Proceedings of International MultiConference of Engineers & Computer Scientists 2013[C], Hong Kong: Newswood Limited, 2013:1–5.

[3] REZAEI S B, SARRAFZADEH A, SHANBEHZADEH J. Skew Detection of Scanned Document Images[A]. Proceedings of the International MultiConference of Engineers and Computer Scientists Vol I.[C], Hong Kong: Newswood Limited, 2013:1–6.

[4] SHAFAIT F, KEYSERS D, BREUEL T M. Performance Comparison of Six Algorithms for Page Segmentation[A]. DOCUMENT ANALYSIS SYSTEMS VII, PROCEEDINGS[C], Berlin: SPRINGER-VERLAG, 2006:368–379.

[5] KEYSERS D, SHAFAIT F, BREUEL T M. DOCUMENT IMAGE ZONE CLASSIFICATION[A]. Proceedings of International Conference on Computer Vision Theory and Applications[C], Barcelona: INSTICC, 2007:44–51.

[6] BREUEL T M. High Performance Document Layout Analysis[A]. Proceedings of the Symposium on Document Image Understanding Technology[C], Greenbelt: University of Maryland, 2003:209–218.

[7] KAMOLA G, SPYTKOWSKI M, PARADOWSKI M, et al. Image-based logical document structure recognition[J]. Pattern Analysis and Applications, 2014:1–15.

[8] MOHAMED C, NAWWAF K, LIU C L, et al. Character recognition systems[M]. Hoboken: Wiley-Interscience, 2007.

[9] 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.

[10] TRIER D, TAXT T, JAIN A K. Feature extraction methods for character recognition - A survey[J]. Pattern Recognition, 1996, **29**(4):641–662.

[11] BREUEL T M. The OCRopus open source OCR system[A]. Proceedings of SPIE 6815, Document Recognition and Retrieval XV[C], San Jose: SPIE-IS&T, 2008:68150F–68150F–15.

[12] 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.

[13] 肖敏, 黄磊, 刘迎建. 数学公式识别系统 [A]. 第八届全国汉字识别学术会议论文集 [C], 绍兴: 中国中文信息学会基础理论专业委员会, 2002:31–37.

[14] CHAN K F, YEUNG D Y. Mathematical expression recognition: a survey[R]. Hong Kong: HKUST, 1999.

[15] SHAFAIT F, VAN BEUSEKOM J, KEYSERS D, et al. Document cleanup using page frame detection[J]. International Journal on Document Analysis and Recognition, 2008, **11**(2):81–96.

[16] SAUVOLA J, PIETIKäINEN M. Adaptive document image binarization[J]. Pattern Recognition, 2000, **33**(2):225–236.

[17] BURGER W, BURGE M J. 数字图像处理: Java 语言算法描述 [M]. 北京: 清华大学出版社, 2010.

[18] SHAFAITA F, KEYSERSA D, BREUEL T M. Efficient Implementation of Local Adaptive Thresholding Techniques Using Integral Images[A]. Proceedings of The International Society for Optical Engineering, Document Recognition and Retrieval XV[C], San Jose: SPIE-IS&T, 2008:681510–681510–6.

[19] 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.

[20] O'GORMAN L. Image and Document Processing Techniques for the RightPages Electronic Library System[A]. Proceedings of the International Conference on Pattern Recognition[C], Los Alimitos: IEEE, 1992:260–263.

[21] GATOS B, PRATIKAKIS I, PERANTONIS S. Adaptive degraded document image binarization[J]. Pattern Recognition, 2006, **39**(3):317–327.

[22] ALAEI A, NAGABHUSHAN P, PAL U, et al. A Painting Based Technique for Skew Estimation of Scanned Documents[A]. Proceedings of the International Conference on Document Analysis and Recognition[C], Washington: IEEE, 2011:299–303.

[23] Computer Vision Lab of Vienna University of Technology. Skew Estimation Datasets[DB/OL]. `http://www.caa.tuwien.ac.at/cvl/research/skew-database/`, 2015-1-15.

[24] SUZUKI M, TAMARI F, FUKUDA R, et al. INFTY —An Integrated OCR System for Mathematical Documents[A]. Proceedings of the 2003 ACM Symposium on Document Engineering[C], Grenoble: Elsevier B.V., 2003:95–104.

[25] 靳从. 中文版面分析关键技术的研究 [D]. 南京: 南京理工大学, 2007.

[26] KIENINGER T, DENGEL A. The T-Recs Table Recognition and Analysis System[A]. Proceedings of the Third IAPRWorkshop on Document Analysis Systems[C], London: Springer-Verlag, 1998: 255–269.

[27] SAUVOLA J, KAUNISKANGAS H. MediaTeam Oulu Document Database[DB/OL]. `http://www.mediateam.oulu.fi/downloads/MTDB/`, 2015-4-3.

[28] SMITH R. An Overview of the Tesseract OCR Engine[A]. Proceedings of the International Conference on Document Analysis and Recognition[C], Curitiba: IEEE, 2007:629–633.

[29] DIAZ A D. GNU Ocrad Manual[EB/OL]. `http://www.gnu.org/software/ocrad/manual/ocrad_manual.html`, 2014-12-29.

[30] UCHIDA S, NOMURA A, SUZUKI M. Quantitative Analysis of Mathematical Documents[J]. International Journal on Document Analysis and Recognition, 2005, **7**(4):211–218.

[31] SMITH R. History of the Tesseract OCR Engine: What Worked and What Didn't[A]. Proc. of SPIE-IS&T Electronic Imaging, Document Recognition and Retrieval XX[C], Burlingame: SPIE-IS&T, 2013:865802–865802–12.

[32] KNUTH D E. The TEXbook[M]. Reading: Addison-Wesley, 1986.

[33] CORMEN T H, LEISERSON C E, RIVEST R L, 等. 算法导论 [M]. 北京: 机械工业出版社, 2013.

[34] American Mathematical Society. AMSFonts 3.04[CP/DK]. http://www.ctan.org/pkg/amsfonts, 2013-1-14.

[35] ZANIBBI R, BLOSTEIN D. Recognition and retrieval of mathematical expressions[J]. International Journal on Document Analysis and Recognition, 2012, **15**(4):331–357.

[36] RAJA A, RAYNER M, SEXTON A, et al. Towards a parser for mathematical formula recognition[A]. Mathematical Knowledge Management, Proceedings[C], Berlin: SPRINGER-VERLAG, 2006:139–151.

[37] ETO Y, SUZUKI M. Mathematical formula recognition using virtual link network[A]. Proceedings of Sixth International Conference on Document Analysis & Recognition[C], Washington: IEEE Computer Society, 2001:762–767.

[38] 李永华, 王科俊, 上官伟, 等. 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.

[39] BLOSTEIN D, GRBAVEC A. Recognition of Mathematical Notation[A]. Handbook on Optical Character Recognition and Document Image Analysis[M], Singapore: World Scientific, 1997:557–582.

[40] SUZUKI M. Character and Symbol Image Database InftyCDB[DB/OL]. http://www.inftyproject.org/en/database.html, 2015-3-1.

致谢

正如马克思所指出的，“甚至当我从事科学之类的活动，即从事一种我只是在很少情况下才能同别人直接交往的时候，我也是社会的，因为我是作为人活动的。不仅我的活动所需的材料，甚至思想家用来进行活动的语言本身，都是作为社会的产品给予我的，而且我本身的存在就是社会的活动；因此，我从自身所做出的东西，是我从自身为社会做出的，并且意识到我自己是社会存在物”。在完成本科学业的四年里，我的学习和研究工作，决不是自己孤立的工作，也与无数人的努力分不开。在完成毕业论文之际，谨对他们表达诚挚的谢意。

首先，感谢导师黎培兴副教授认真的指导和教诲，正是他的鼓励让我走进了文档分析的领域，并在这里开发出一个受到注意的软件。此外，还需感谢所有曾教导过我的老师，他们的言行已经融合成今天的我所不可或缺的部分。感谢中大，学校提供了良好的支持条件，让我得以安心地学习，使我在这匆匆的四年中不管在学识还是世界观都上了一个台阶，有幸可以把自己的学习和工作作为为自己而玩。

最后，感谢所有在数学或计算机领域进行过工作的人，因为现在一切的工作都是站在前人的肩膀上的，不管他们是不是巨人。特别要感谢多年来为自由软件运动挥洒汗水的黑客们，他们不仅让大家可以用上许多优秀的软件，而且让我形成了现实的理想主义精神。对于他们的奉献，实无法回报，只好在力所能及的范围内开发自由软件和传播数学与计算机知识，把共享进步的理念传承下去。

陈颂光

2015 年 5 月 7 日

附录 A　两种图像二值化方法

正文中提到两种在全局阈值化方法和局部阈值化方法中分别被认为较优的 Otsu 方法和 Sauvola 方法，以下详细介绍这两种二值化方法并讨论如何有效地实现它们。

定义 A.1 设 D 为一个高度为 m 而宽度为 n 的灰度图像，对每个 $t \in 256$，把 $m \times n$ 中各点分为两类

$$A_1^{(t)} = \{(i,j) \in m \times n \mid D(i,j) \leq t\} \tag{A.1}$$

$$A_2^{(t)} = \{(i,j) \in m \times n \mid D(i,j) > t\} \tag{A.2}$$

，记两类中的点数分别为 $n_1^{(t)} = |A_1^{(t)}|, n_2^{(t)} = |A_2^{(t)}|$，又记两类中平均灰度和总平均灰度为

$$\mu_1^{(t)} = \frac{\sum_{(i,j)\in A_1^{(t)}} D(i,j)}{\max(n_1^{(t)},1)}, \mu_2^{(t)} = \frac{\sum_{(i,j)\in A_2^{(t)}} D(i,j)}{\max(n_2^{(t)},1)}, \mu = \frac{\sum_{(i,j)\in m\times n} D(i,j)}{m \times n} \tag{A.3}$$

，令类间方差为

$$\sigma_t^2 = \frac{n_1^{(t)}}{n_1^{(t)}+n_2^{(t)}}(\mu_1^{(t)}-\mu)^2 + \frac{n_2^{(t)}}{n_1^{(t)}+n_2^{(t)}}(\mu_2^{(t)}-\mu)^2 \tag{A.4}$$

，取 $t_0 \in 256$ 使

$$\sigma_{t_0}^2 = \max_{t\in 256} \sigma_t^2 \tag{A.5}$$

，则称 t_0 为 D 的 Otsu 全局阈值。

为了计算 Otsu 全局阈值，可以先生成灰度直方图，这需要遍历所有灰度值一次，余下的计算均可仅基于灰度直方图进行，与图像的高度和宽度无关。因此，对于高度为 m 而宽度为 n 的灰度图像，计算 Otsu 全局阈值的时间复杂度为 $\Theta(mn)$。

定义 A.2 设 D 为一个高度为 m 而宽度为 n 的灰度图像，ω 为正整数，$k \in [0.2, 0.5]$，对每个 $(i,j) \in m \times n$，记

$$(D(y,x))|_{i-\frac{\omega}{2}<y\leq i+\frac{\omega}{2}, j-\frac{\omega}{2}<x\leq j+\frac{\omega}{2}}$$

的平均值和标准差分别为 $m(i,j)$ 和 $s(i,j)$，令

$$T(i,j) = m(i,j)\left(1 + k\left(\frac{s(i,j)}{128} - 1\right)\right) \tag{A.6}$$

，则称 T 为 D 的 Sauvola 阈值，记为 $\text{Sauvola}_{\omega,k}(D)$。

在 Sauvola 方法中，在窗口大小为 ω 时，计算一点的阈值需要 ω^2 个像素的灰度值，但注意到实际上只需要这些灰度值的和与平方和，因此可以用积分图像技巧简化计算 [18]，这样在一点的阈值计算的摊还时间复杂度就与窗口大小无关，从而总的时间复杂度为 $\Theta(mn)$，不过预期的运行时间还是要长于全局阈值化方法。

附录 B　应用积分图像的图像处理技巧

一些图像处理操作（例如 Sauvola 二值化方法、均值滤波和二值化后处理）需要计算图像中一个矩形区域（通常被称为窗口）中像素值之和，这时如果直接按定义计算，则涉及的像素个数与窗口面积成正比。以下介绍一种优化这种计算的方法。

定义 B.1 设有函数 $D\colon m \times n \to \mathbb{Z}$，则称

$$\begin{aligned} I\colon \mathbb{Z} \times \mathbb{Z} &\to \mathbb{Z} \\ (k, \ell) &\mapsto \sum_{\substack{i \le k \\ j \le \ell}} D(i, j) \end{aligned}$$

为 D 的积分图像。

以下命题给出了一个利用积分图像计算图像中一个窗口中像素值和的方法，它只用访问积分图像中 4 个值，进行三次加法或减法，与窗口大小无关。

命题 B.1 设 I 为 $D\colon m \times n \to \mathbb{Z}$ 的积分图像，$x_1, x_2, y_1, y_2 \in \mathbb{Z}$ 使 $x_1 < x_2, y_1 < y_2$，则

$$\sum_{\substack{x_1 < j \le x_2 \\ y_1 < i \le y_2}} D(i, j) = I(y_2, x_2) - I(y_2, x_1) - I(y_1, x_2) + I(y_1, x_1) \tag{B.1}$$

。

证明

$$\sum_{\substack{x_1 < j \le x_2 \\ y_1 < i \le y_2}} D(i, j) = \sum_{\substack{x_1 < j \le x_2 \\ i \le y_2}} D(i, j) - \sum_{\substack{x_1 < j \le x_2 \\ i \le y_1}} D(i, j) \tag{B.2}$$

$$= \left(\sum_{\substack{j \le x_2 \\ i \le y_2}} D(i, j) - \sum_{\substack{j \le x_1 \\ i \le y_2}} D(i, j) \right) - \left(\sum_{\substack{j \le x_2 \\ i \le y_1}} D(i, j) - \sum_{\substack{j \le x_1 \\ i \le y_1}} D(i, j) \right) \tag{B.3}$$

$$= \sum_{\substack{j \le x_2 \\ i \le y_2}} D(i, j) - \sum_{\substack{j \le x_1 \\ i \le y_2}} D(i, j) - \sum_{\substack{j \le x_2 \\ i \le y_1}} D(i, j) + \sum_{\substack{j \le x_1 \\ i \le y_1}} D(i, j) \tag{B.4}$$

$$= I(y_2, x_2) - I(y_2, x_1) - I(y_1, x_2) + I(y_1, x_1) \tag{B.5}$$

□

注意到 $i < 0$ 或 $j < 0$ 时有积分图像取值 $I(i, j) = 0$，以下推论给出了生成积分图像的一个有效的递推方法，其时间复杂度为 $\Theta(mn)$（只用求在 $m \times n$ 上的值）。

推论 B.1 设 I 为 $D\colon m \times n \to \mathbb{Z}$ 的积分图像，则对 $i \in m, j \in n$ 有

$$I(i, j) = I(i, j-1) + I(i-1, j) + D(i, j) - I(i-1, j-1) \tag{B.6}$$

。

证明 因为有

$$D(i, j) = I(i, j) - I(i, j-1) - I(i-1, j) + I(i-1, j-1)$$

移项即得。 □

附录 C　连通域分割的一种计算方法

连通域分割在正文中多次用到，以下给出连通域的严格定义和一种有效的连通域分割算法。

定义 C.1 设 $F \in \mathcal{P}(\mathbb{Z}\times\mathbb{Z})$，若存在 $(x_0,y_0),\ldots,(x_n,y_n)\in F$ 使 $(x_0,y_0)=(x,y),(x_n,y_n)=(z,w)$ 且对任何 $i=1,\ldots,n$ 有

$$|x_i-x_{i-1}|\le 1,|y_i-y_{i-1}|\le 1 \tag{C.1}$$

，则称 (x,y) 与 (z,w) (在 F) 连通，并称 $(x_0,y_0),\ldots,(x_n,y_n)$ 为一条从 (x,y) 到 (z,w) 的道路。进一步，称

$$\simeq=\{((x,y),(z,w))\in F\times F|(x,y)\text{与}(z,w)\text{连通}\} \tag{C.2}$$

为 F 上连通关系。

命题 C.1 $F\in\mathcal{P}(\mathbb{Z}\times\mathbb{Z})$ 上的连通关系 $\simeq$ 为 F 上的一个等价关系。

证明 按照等价关系定义，需要验证以下三点：

- 自反性
  对任何 $(x,y)\in F$，显然 (x,y) 即为一条从 (x,y) 到 (x,y) 的道路，故 (x,y) 与 (x,y) 连通，即 $(x,y)\simeq(x,y)$。
- 对称性
  若 $(x,y)\simeq(z,w)$,则存在一条从 (x,y) 到 (z,w) 的道路 $(x_0,y_0),\ldots,(x_n,y_n)$，于是 $(x_n,y_n),\ldots,(x_0,y_0)$ 是一条从 (z,w) 到 (x,y) 的道路，故 (z,w) 与 (x,y) 连通，即 $(z,w)\simeq(x,y)$。
- 传递性
  若 $(x,y)\simeq(z,w),(z,w)\simeq(u,v)$，则存在一条从 (x,y) 到 (z,w) 的道路 $(x_0,y_0),\ldots,(x_n,y_n)$,又存在一条从 (z,w) 到 (u,v) 的道路 $(z_0,w_0),\ldots,(z_m,w_m)$，于是 $(x_0,y_0),\ldots,(x_n,y_n)=(z_0,w_0),\ldots,(z_m,w_m)$ 为一条从 (x,y) 到 (u,v) 的道路，故 (x,y) 与 (u,v) 连通，即 $(x,y)\simeq(u,v)$。

□

定义 C.2 设 $m,n\in\mathbb{Z}^+$，$F\subseteq m\times n$，$\simeq$ 为 F 上连通关系，则 F 关于 $\simeq$ 的各个等价类称为 F 的连通分支，而商集 $F/\simeq$ 称为 F 的连通域分割。

命题 C.2 设 $m,n\in\mathbb{Z}^+$，$F\subseteq m\times n$，$\simeq$ 为 F 上连通关系，对 $(i,j)\in m\times n$ 或 $(i,j)=(m,0)$，令

$$\mathcal{D}_{i,j}=\{(y,x)\in F|(y,x)\prec(i,j)\} \tag{C.3}$$

$$\mathcal{C}_{i,j}=\mathcal{D}_{i,j}/\simeq_{i,j} \tag{C.4}$$

$$\mathcal{E}_{i,j}=\{C\in\mathcal{C}_{i,j}|\{(i-1,j-1),(i-1,j),(i-1,j+1),(i,j-1)\}\cap C=\emptyset\} \tag{C.5}$$

，其中 $\simeq_{i,j}$ 为 $\mathcal{D}_{i,j}$ 上的连通关系，则 $\mathcal{C}_{0,0}=\emptyset$，且对 $(i,j)\in m\times n$，令 $(k,\ell)=\begin{cases}(i+1,0) & ,j=n-1\\(i,j+1) & ,j\neq n-1\end{cases}$，则有

$$\mathcal{C}_{k,\ell}=\begin{cases}\mathcal{E}_{i,j}\cup\left\{\{(i,j)\}\cup\cup_{C\in\mathcal{C}_{i,j}\setminus\mathcal{E}_{i,j}}C\right\} & ,(i,j)\in F\\\mathcal{C}_{i,j} & ,(i,j)\notin F\end{cases} \tag{C.6}$$

。

证明 因 $\mathcal{D}_{0,0}=\emptyset$, 故有 $\mathcal{C}_{0,0}=\emptyset$。接着, 在 $(i,j)\notin F$ 时, $\mathcal{D}_{k,\ell}=\mathcal{D}_{i,j}$, 故 $\mathcal{C}_{k,\ell}=\mathcal{C}_{i,j}$。在 $(i,j)\in F$ 时, 有 $\mathcal{D}_{k,\ell}=\mathcal{D}_{i,j}\cup\{(i,j)\}$。注意到若 $(y,x)\in\mathcal{D}_{i,j}$ 在 $\mathcal{D}_{k,\ell}$ 中与 (i,j) 连通, 则从 (y,x) 到 (i,j) 的道路必经 $\{(i-1,j-1),(i-1,j),(i-1,j+1),(i,j-1)\}$ 中某点，于是 (y,x) 在 $\{(i-1,j-1),(i-1,j),(i-1,j+1),(i,j-1)\}$ 中某点在 $\mathcal{D}_{i,j}$ 所属连通分支；反之，如果 $(i-1,j-1),(i-1,j),(i-1,j+1),(i,j-1)$ 中某个属于 $\mathcal{D}_{i,j}$，则它在 $\mathcal{D}_{i,j}$ 所属连通分支中任一点均在 $\mathcal{D}_{k,\ell}$ 与 (i,j) 连通。这说明 $\mathcal{D}_{k,\ell}$ 中 (i,j) 所在连通分支为 $\{(i,j)\}\cup\cup_{C\in\mathcal{C}_{i,j},\{(i-1,j-1),(i-1,j),(i-1,j+1),(i,j-1)\}\cap C\neq\emptyset}C$。而对于 $\mathcal{D}_{k,\ell}$ 中不在此连通分支中的点 (y,x)，由它出发的道路都不经 (i,j)，从而完全落在 $\mathcal{D}_{i,j}\setminus\{(i-1,j-1),(i-1,j),(i-1,j+1),(i,j-1)\}$ 中，于是 (y,x) 在 $\mathcal{D}_{k,\ell}$ 中所属连通分支也是它在 $\mathcal{D}_{i,j}$ 中所属连通分支。 □

注意到在上述命题中，$\mathcal{C}_{m,0}$ 即为 F 的连通域分割，故它给出了一个求连通域分割的递推方法，具体操作见算法C.1。其中 $\texttt{makeset}((i,j,k-j))$ 表示在 $\mathcal{C}$ 中新增集合 $\{(i,j),\ldots,(i,k-1)\}$ 而 $\texttt{union}(C_1,C_2)$ 表示合并 $\mathcal{C}$ 中的集合 C_1 和 C_2，这些操作使用带路径压缩和按秩合并的不相交集合算法 [33]。整个算法时间复杂度为 $\Omega(mn)$ 和 $O\left(mn\alpha\left(mn\right)\right)$，其中 α 为 [33] 中定义的一个增长极其缓慢的函数。

算法 C.1: 连通域分割

输入: $F \subseteq m \times n$

输出: F 的连通域分割 $\mathcal{C}$

数据结构: $lastlt$，数组 $last$

开始
  $\mathcal{C} \leftarrow \emptyset$;
  对于每个 $i \in m$ **进行**
    $lastlt \leftarrow \emptyset$;
    对于每个 $j \in n$ **进行**
      如果 $(i,j) \notin F$ **则**
        $lastlt \leftarrow last[j], last[j] \leftarrow \emptyset$;
      否则
        $c \leftarrow lastlt$;
        $k \leftarrow j$;
        当 $k < n \wedge (i,k) \in F$ **进行**
          如果 $i \neq 0 \wedge last[k] \neq \emptyset$ **则**
            如果 $c = \emptyset$ **则**
              $c \leftarrow last[k]$;
            否则
              $\texttt{union}(c, last[k])$;
          $k \leftarrow k + 1$;
        如果 $i \neq 0 \wedge k \neq n \wedge last[k] \neq \emptyset$ **则**
          如果 $c = \emptyset$ **则**
            $c \leftarrow last[k]$;
          否则
            $\texttt{union}(c, last[k])$;
        如果 $c \neq \emptyset$ **则**
          $\texttt{union}(c, \texttt{makeset}((i,j,k-j)))$;
        否则
          $c \leftarrow \texttt{makeset}((i,j,k-j))$;
        对于 $l \in \{j, \ldots, k-1\}$ **进行**
          $last[l] \leftarrow c$;
        $j \leftarrow k - 1$;

附录 D　获取本文系统的途径

本文中，“本文系统”是指 MathOCR 0.0.3，MathOCR 是一个在 GNU 通用公共许可证（GPL）版本 3 下发布的自由软件，这个软件（包括所有源代码和训练数据）可以从项目网站http://mathocr.sourceforge.net/或后备站点https://github.com/chungkwong/MathOCR免费下载。如果使用版本控制系统 Subversion，可以用以下命令从代码仓库签出项目代码：

```
svn checkout svn://svn.code.sf.net/p/mathocr/code/
   mathocr-code
```

；如果使用版本控制系统 Git，则可以用以下命令从代码仓库签出项目代码：

```
git clone https://github.com/chungkwong/MathOCR.git
```

。

毕业论文成绩评定记录

指导教师评语：   成绩评定： 指导教师签名：　　年　　月　　日
答辩小组或专业负责人意见：   成绩评定： 签名（章）：　　年　　月　　日
院系负责人意见：   成绩评定： 签名（章）：　　年　　月　　日