基于笔划提取的脱机手写数学公式识别研究

Stroke Extraction for Offline Handwritten Mathematical Expression Recognition

陈颂光

二零二零年十月

基于笔划提取的脱机手写数学公式识别研究

摘要

手写数学公式识别为数学公式的电子化提供了一条自然的途径，进而可为数学公式检索和其它基于数学公式的应用创造条件。由于数学公式具有紧凑的二维结构，识别它们对机器而言是一个有挑战性的任务，对于手写体尤其如此。手写体识别传统上被分为联机识别和脱机识别，前者识别动态的笔迹而后者识别静态的图片。与联机识别相比，由于缺乏动态信息和存在背景噪声，脱机识别的准确度通常较低，同时往往占用更多资源。于是，把脱机识别问题转化为联机识别问题是一个有吸引力的想法。本文的目的就是探讨基于笔划提取的脱机手写数学公式识别的可行性和实用性。

首先，本文表明了通过把一个不完全准确的笔划提取器与现有的联机手写数学公式识别系统相结合，可以得到颇为准确的脱机手写数学公式识别系统。为了自动地从位图提取笔划，本文提出了一个过切分算法：首先把黑白图像的骨架分解为一些线段和端点，然后通过自底而上的聚类把线段合并成笔划，最后用递归投影切分和拓扑排序确定笔顺。结合一个现成的联机手写数学公式识别系统，本方法分别正确地识别了联机手写数学公式识别竞赛（CROHME）2014、2016 和 2019 数据集中 58.22%、65.65% 和 65.22% 的脱机手写数学公式，由此获得了 2019 年脱机识别任务的季军。这说明了本文方法可用于构造准确性有竞争力的脱机手写数学公式识别系统。为了克服笔划提取器单点故障问题，本文还建议用提取出的笔划代替原始笔划去重新训练所基于的联机手写数学公式识别系统。实验结果表明了利用这方法可以构造一个准确度与之相若的脱机手写数学公式识别系统。这不仅指出了分别开发联机和脱机两套识别系统所需的重复工作可能并不值得，而且说明了联机识别与脱机识别这两个问题间的难度差距没有人们过往想像中大。更进一步，通过在混合数据上进行训练，可以使一个模型同时适用于书写笔划和提取出的笔划。

其次，本文表明了基于笔划提取的脱机手写数学公式识别系统具有低资源占用的优势。注意到图像预处理阶段容易成为性能瓶颈，本文为一大类局部自适应二值化算法提供了内存高效的快速实现。与基于积分图像的标准实现相比，本文提出的实现在大幅降低内存占用的同时略为提高了速度。特别地，本文给出了 Bernsen 方法和 Niblack 型方法如 Sauvola 方法的新串行实现，它们的时间复杂度与输入图像的大小成正比且与窗口大小无关，而辅助空间关于输入图像大小次线性。实验结果表明了，基于笔划提取的脱机手写识别系统的性能明显优于原生的脱机手写识别系统，而且速度与所基于的联机手写识别系统相近。值得注意的是，整个脱机识别流程可以在内存受限的手机上完成。

最后，本文讨论了如何在应用中利用不总是准确的数学公式识别结果，这对

于手写数学公式识别技术的实用化而言是至关重要的。虽然不能期望手写数学公式识别完全准确，但识别结果与正确结果间往往有一些相似性。针对要求高准确性的场合，例如作为计算机代数系统的输入时，本文提出了一个高可用性的公式编辑器设计，它充分地利用了手写识别系统提供的信息以便让用户可以高效地发现和修正错误。针对不要求高准确性的场合，例如数学公式检索，本文提出了一种基于编辑距离的容错排序方法，以便减低识别错误对搜索结果的影响。

关键词： 脱机手写数学公式识别，光学字符识别，笔划提取，局部自适应二值化，数学公式检索

Stroke Extraction for Offline Handwritten Mathematical Expression Recognition

ABSTRACT

Handwritten mathematical expression recognition provides a natural way to digitize mathematical content, which enables formula retrieval and other applications. Recognizing mathematical expressions mechanically is challenging because of their compact two-dimensional structure, especially for the handwritten ones. Handwriting recognition is traditionally divided into online recognition and offline recognition. The former recognizes dynamic trajectories whereas the latter recognizes static images. Offline handwritten mathematical expression recognition is often considered much harder than its online counterpart due to the absence of temporal information and the presence of background noises. In order to take advantage of the more mature methods for online recognition and save resources, stroke extraction is an interesting idea. The goal of this study is to explore the feasibility and practicability of offline handwritten mathematical expression recognition via stroke extraction.

The main point is that a good offline handwritten mathematical expression recognition system can be obtained by combining a stroke extractor with an existing online handwritten mathematical expression recognition system, even though the stroke extractor may not be perfect. In order to recover strokes from textual bitmap images automatically, an oversegmentation approach is proposed. The proposed algorithm first breaks down the skeleton of a binarized image into junctions and segments, then segments are merged to form strokes, finally stroke order is normalized by using recursive projection and topological sort. Given a ready-made state-of-the-art online handwritten mathematical expression recognizer, the proposed procedure correctly recognized 58.22%, 65.65%, and 65.22% of the offline formulas rendered from the datasets of the Competitions on Recognition of Online Handwritten Mathematical Expressions (CROHME) in 2014, 2016, and 2019 respectively. It was ranked third in the main offline task of CROHME 2019. The results indicate that stroke extraction can be applied to build offline recognition systems with competitive accuracy. In order to prevent the stroke extractor from becoming a single point of failure, it is suggested to retrain the underlying online recognition system with extracted strokes. Given a trainable online recognition system, the suggestion resulted in an offline recognizer with the same level of

accuracy. The results not only indicate that developing independent recognition systems for online and offline handwriting may no longer be necessary, but also imply that the essential difficulty between online recognition and offline recognition may be smaller than it was thought. Meanwhile, a model that work well on both written strokes and extracted strokes can be obtained by training it on mixed data.

In addition, stroke extraction based offline handwritten mathematical expression recognition systems are fairly lightweight. Since image preprocessing is likely the performance bottleneck, memory-efficient and fast implementations of a large class of local adaptive binarization methods are proposed. Compared with the standard implementations which rely on integral images, memory footprint is reduced dramatically, whereas the speed also increased slightly. In particular, the technique leads to new sequential implementations of Bernsen's method and Niblack inspired methods such as Sauvola's method. Their time complexity is linear to the size of input image and independent of the window size, whereas auxiliary space is sublinear to the size of input image. Experimental results show that the speed of the proposed system was comparable to that of the underlying online recognition system, which was much faster than native offline recognition systems. It should also be noted that the entire pipeline was able to facilitate on-device recognition on mobile phones with limited memory.

Last but not least, the use of possibly incorrect recognition results in application is discussed, it is a serious issue that has not yet received much attention. Although one cannot expect a handwritten mathematical expression recognizer to work perfectly, there are usually some similarities between the recognition result and the correct result. For applications which require exact input, such as computer algebra systems, a design of equation editor with usability in mind is proposed, it can make full use of the information provided by the recognition system, so that users can discover and fix errors efficiently. For other applications, such as mathematical formula retrieval, an error-tolerant ranking method based on edit distance is proposed, so that search results would be less sensitive to recognition errors in queries.

Key Words: Offline handwritten mathematical expression recognition, optical character recognition, stroke extraction, local adaptive binarization, mathematical formula retrieval

目录

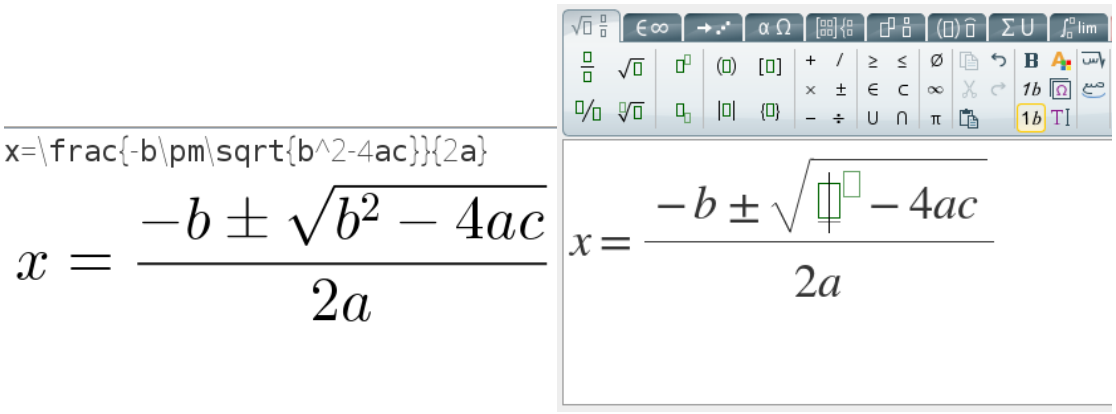
第一章　引言

1.1　问题背景

数学公式是在各种手写和印刷文档中常见的信息载体，如何充分地利用数学公式中的价值是一个有意义的问题。由于紧凑的二维结构有助于简明地表达一些概念，数学公式在经管、科学、工程和其它领域中都十分重要。为了提高数学公式的可发现性，把数学公式电子化是很有必要的，因为这有助于长时间保存和传播。进一步把数学公式结构化 [65] 将容许有效的数学公式检索 [36]，从而使问答系统 [98]、抄袭检测 [67] 和其它自动化任务成为可能，它们在教育、科研和其它方面都有很多潜在的应用。

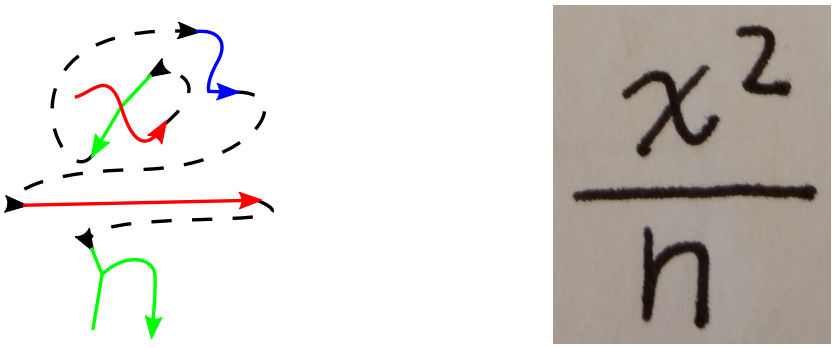
与普通文本相比，数学公式更不方便于输入。传统的输入设备如键盘是为单方向的字符串而设计的，虽然符号间的平面位置关系可以用 TEX[47] 或 MathML [7] 之类的代码表示（如图1-1a所示），但用计算机语言表达数学公式对新手而言非常不友好，因为他们不仅要另外学习一门语言和记忆一系列命令，还要与令人困惑的错误打交道。另一方面，用图形化的公式编辑器（如图1-1b所示）输入数学公式对老手而言效率太低，因为重复地从工具箱中选取结构元素和符号的效率不高。

既然人们已经习惯了在纸上或黑板上书写数学公式，用相同的方式向计算机输入数学公式就是最理想的。识别系统的目标就是把手写的数学公式转化为



(a) 编写 LaTeX 代码　　**(b)** 使用公式编辑器 MathType

图 1-1 常见的数学公式录入方法的例子



(a) 笔迹 (b) 位图

图 1-2 联机与脱机手写识别系统分别接受的输入的例子

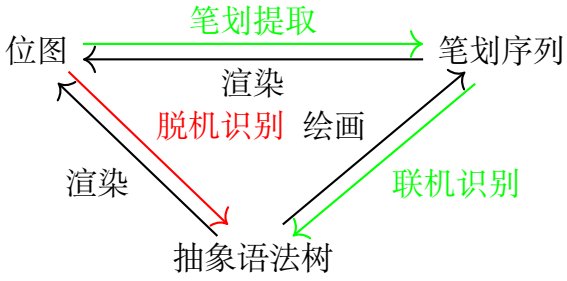


图 1-3 脱机识别与联机识别的关系图

方便机器处理的语法树。手写识别又分为联机识别和脱机识别 [89]，两者间的区别在于联机识别系统接受动态笔迹为输入（如图1-2a所示），而脱机识别则接受位图为输入（如图1-2b所示）。联机识别容许人们通过在触摸设备上书写或拖动鼠标来输入数学公式，适合用于记笔记或与计算机代数系统交互。脱机识别则容许人们录入图片中的数学公式，适合用于电子化手稿的扫描版或者黑板的照片。

联机识别的准确率一般较高。比如说在 2019 年联机手写数学公式识别竞赛（Competitions on Recognition of Online Handwritten Mathematical Expressions，CROHME）中，名列前三的联机识别系统分别正确地识别了 80.73%、79.82% 和 79.15% 的数学公式，而名列前三的脱机系统仅分别正确地识别了 77.15%、71.23% 和 65.22% 的数学公式 [63]。表面上的原因在于脱机识别不仅能利用书写结果的信息，还能利用书写过程的信息。事实上，通过简单地把笔迹渲染为图片，我们可以轻易地把联机识别问题转化为脱机识别问题。也就是说，给定一个脱机识别系统，我们总可以构造一个和它一样准确的联机识别系统。反过来，如果能够从位图中恢复笔迹信息，联机识别系统也能被用于脱机识别 [82]。图1-3表达了两类识别之间的区别与联系，脱机识别可以分解为笔划提取和联机识别。

本文的目的就是研究在不依赖于专门设计的联机识别系统的前提下，通过笔划提取识别脱机手写数学公式的可行性和实用性。在笔划提取完全准确的情况下，给定一个联机识别系统，我们也可以得到一个和它一样准确的的脱机识别

系统。不过，因为准确的笔划提取相当困难 [89]，所以需要把重点放在防止单点故障上。也就是说，要基于较不可靠的笔划提取算法来搭建较可靠的脱机识别算法。由于书写习惯的多样性已经影响到联机识别系统，它们已经能够容忍一些输入错误。进一步，如果将错就错，用提取出的笔划来训练联机识别系统，它就应该能适应，于是笔划提取的准确率对脱机识别的准确率的影响就不一定是决定性的。

1.2 选题意义

1.2.1 理论价值

一直以来，手写识别被分为联机识别和脱机识别两个子问题，两者之间的关系自然是一个基本的问题。一般认为，动态的笔迹比静态的位图包含严格更多的信息，但额外的信息对识别的准确率到底有多大的影响却在很大程度上是未知的。本文的方法提供了探讨这个问题的一个思路。

假如有一种方法可以基于任意联机识别系统构建一个与之一样准确的脱机识别系统，就能够说明脱机识别并不比联机识别困难。笔划提取正是一种可以基于任意联机识别系统构建脱机识别系统的方法。诚然，过往也有个别研究者进行过类似的实验，但他们都有这种或那种的局限性，例如大都依赖于针对提取出的笔划专门设计的联机识别系统，又或是仅考虑较简单的识别问题如单字符识别。相反，我们考虑的是极具挑战性的数学公式识别问题，而且基于在工业界和学术界有代表性的联机识别系统，它们被设计时根本没有考虑过会被用于我们的目的。因此，我们的实验能更客观地比较，而我们能做到的原因在于具强大学习能力的联机识别系统开始成熟。

即使做不到一样准确，在设法让脱机识别系统的准确性逼近所基于的联机识别系统方面付出足够多的努力后，两者的准确率的差距将给出联机识别与脱机识别间难度差异的量化比较。特别地，用提取出的笔划重新训练联机识别系统是在这个方向上的一个一般的方法，适用于现代所有主流的联机识别系统，因为它们都基于机器学习。当然，改进笔划提取的准确性仍然是另一个方法。本文在这两个方法上都作出了有益的探索。

1.2.2 实践含意

本研究对手写识别系统的开发可能有指导性意义。

在宏观上，笔划提取提供了基于常规联机识别系统搭建脱机识别系统的一般途径。由于联机识别系统通常无论在准确性还是资源占用上都占优，这方案可能导致准确而轻量的脱机识别系统。同时，笔划提取使得图片也可被用作联机手写识别的训练数据，有利于增大训练集，从而可能可提高联机手写识别系统的健

壮性。此外，这方案可以减少由于分别开发联机识别系统和脱机识别系统造成的重复劳动，有助于集中力量攻克核心的技术障碍。因此，这方案在工程上有一定的吸引力，尤其是对原有的联机识别系统厂商而言，它们可以复用既有资产而轻易地进军脱机识别市场，达到一石二鸟的效果。

在微观上，对笔划提取算法的消融研究对日后同类系统的开发有启发性。通过分析笔划提取的各个子问题对最终准确率的影响，可以了解笔划的不同特征对联机识别系统的重要性，从而为改进笔划提取算法指明了方向，并有助于更直观地了解联机识别系统的特性。

值得一提的是，数学公式是一类有代表性的二维记号，可以被用来识别手写数学公式的技术也可能可以被用于识别手写的表格 [31]、化学结构式 [30]、乐谱 [9]、电路图 [15]、流程图 [43] 等等，又或是对手写文档进行版面分析。这些任务和数学公式识别同样受符号切分、符号识别和结构分析问题困扰，也和手写数学公式识别一样有实际的用途，但针对它们的专门研究不多，所以借助笔划提取同时做联机和脱机识别是一种可以有效地利用仅有的关注的方法。

1.3 本文的主要工作

为了研究基于笔划提取的脱机手写数学公式识别的可行性和实用性，我们不但设计了一个笔划提取算法来把位图转换为笔划序列，而且提出了一系列技术来使构造出的脱机识别系统尽可能保留联机识别系统的优点。我们还探索了如何应用不总是准确的识别结果。我们通过一系列的实验检验了这些技术的有效性，有关的实验代码已经被公开以便让其它研究者可以方便地重现实验结果。本文的主要工作可以概括如下：

- 为了实现用联机手写数学公式识别系统识别脱机手写数学公式的想法，我们提出了一个针对数学公式的启发式的笔划提取算法。一般来说，任意给定一个联机手写数学公式识别系统，可以如图1-4所示那样把它包装成一个脱机手写数学公式识别系统。通过不加改动地直接套用一个市面上的联机手写数学公式识别系统去识别提取出的笔划，我们在公开数据集上验证了基于笔划提取的脱机数学公式识别系统的准确度与其它前沿的脱机识别系统相比是有竞争力的，特别是取得了 CROHME 2019 竞赛中脱机项目的季军。同时，消融实验表明了删去该笔划提取算法中的任何一个主要步骤都会使准确率明显下滑，这说明了它们的必要性。
- 为了进一步提高这类脱机识别系统的准确性，我们提出通过重新训练来解决笔划提取器的单点故障问题。实验结果显示，通过用提取出的笔划重新训练一个联机手写识别系统，可以得到一个准确率与之相若的脱机手写识别系统。这说明了联机识别系统能够很好地学会提取出的笔划的含义，从而限制了笔划提取错误对识别结果的影响。进一步的实验结果显示，通过

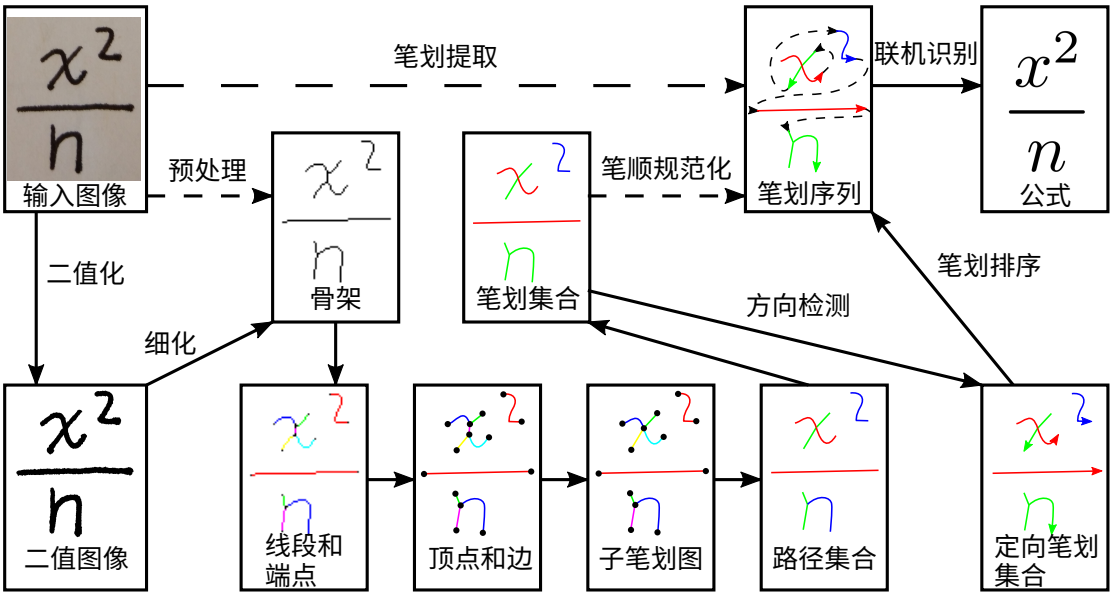


图 1-4 本文提出的脱机手写数学公式识别流程

同时使用书写笔划和提取出的笔划作为训练数据，不但可以进一步提高准确性，而且可以使同一个模型同时在两类输入上取得良好的表现。这实际上说明了不完全准确的笔划提取可以作为扩充训练集的一种方法。

- 为了进一步提高这类脱机识别系统速度和降低内存占用，我们提出了一类局部自适应二值化方法不用积分图像的线性时间实现。由于图像预处理阶段要对完整的输入图像进行操作，它最可能成为笔划提取器的性能瓶颈。实验结果验证了基于笔划提取的脱机识别系统的速度与内存占用一般优于原生的脱机识别系统，而与所基于的联机识别系统可比。这是因为联机识别系统一般比原生的脱机识别系统轻量，同时笔划提取算法也比较简单。
- 针对手写数学公式识别做不到完全准确的问题，我们考虑了应用应该如何应对。对于要求输入准确的场合如计算器，只有在用户可高效地修正错误时手写识别才是有用的，为此本文提出了一种可加速人手修正的公式编辑器设计，该设计充分地利用了来自识别系统的信息。对于对输入不太敏感的应用如数学公式检索来说，识别结果中的错误对结果影响有限，本文还给出了适当的排序方法来进一步控制这种影响。

1.4 本文的组织结构

本文共分为八章，各章节安排如下：

第一章交代了本文的研究思路。先是指出了脱机手写数学公式识别问题的重要性和挑战性，再是说明了笔划提取是一个值得尝试的想法，最后是列出了我们为实现这想法所做的工作。

第二章回顾了国际上相关研究的现状和趋势。对于本文关心的手写数学公

式识别问题，本章分别介绍了针对联机情况和脱机情况的识别方法，并对比了两类方法间的异同。对于本文用以联系联机识别和脱机识别的笔划提取方法，本章先介绍了针对普通文本的做法，再说明了应用于数学公式时所需的变更。

第三章描述了数学公式图片的预处理过程。虽然图像预处理方法是标准的，但本章专门探讨了如何高效地实现之，从而显著地加快整个脱机识别流程和减低内存占用。特别地，本章提出了一类局部自适应二值化方法的内存高效的快速实现，适用于 Bernsen 方法和包括 Sauvola 方法在内的 Niblack 型方法。

第四章提出了一个适用于数学公式的笔划提取算法。本章详细介绍了把骨架转换为笔划序列过程中的各个步骤及其动机。本章还介绍了如何指导笔划提取算法的开发。

第五章介绍了主要实验中需要用到的联机识别系统及其配置。本章刻意不去设计新颖的联机识别系统，因为这与本文的思路背道而驰。

第六章用实验结果检验了基于笔划提取的脱机手写数学公式识别系统的准确性和性能。本章验证了不加修改地套用现成的联机识别系统已经可以得到一个良好的脱机识别系统。本章还验证了通过用提取出的笔划重新训练一个联机识别系统，可以得到一个准确性与之相当的脱机识别系统；通过混合来自笔迹和图片的信息来训练一个模型，可以让联机识别系统与脱机识别系统共用模型。另外，本章用消融研究分析了笔划提取算法，由此指明了设计同类算法时要注意的事项。

第七章探索了一个不完全准确的手写数学公式识别系统的潜在应用。一方面，本章提出了一个高可用性的公式编辑器设计，以便让用户轻松而快捷地修正识别错误。另一方面，本章提出了一种容错的数学公式匹配方法，以便降低识别错误对数学公式检索的影响。

第八章总结了本文工作并指出了一些可以进一步发展的方向。

第二章　国内外研究现状

2.1　概述

本章首先对现有的数学公式识别方法进行了总结，不但介绍了传统的模块化方法和近年开始流行的基于编码器-解码器框架的方法，而且分析了它们分别被用于识别笔迹和图片时的共通点和不同之处。本章接着对现有的笔划提取方法进行了回顾，并指出了它们的启发性意义和局限性。

2.2　数学公式识别

2.2.1　简介

由 Anderson 在 1967 年的工作 [6] 开始，数学公式识别就一直是一个长期以来未有被完全解决的问题。数学公式识别可以分为印刷体数学公式识别和手写数学公式识别，其中印刷体数学公式由于相对标准化而较容易识别，因而早已产品化 [111]。手写数学公式识别又分为联机手写数学公式识别和脱机手写数学公式识别，其中联机手写数学公式由于带时序信息而被认为较容易识别，相反，针对脱机手写数学公式识别的研究直到近几年才开始活跃。虽然在三类识别中能用的信息不同，但相应的识别方法有不少共通点，存在把它们的共同部分抽象出来的尝试 [62]。大体来说，目前主流的数学公式识别方法可以分为模块化方法和基于编码器-解码器框架的方法，第2.2.2小节和2.2.3小节分别介绍了它们。

公开的数学公式数据集的发布促进了对数学公式识别方法的研究，因为只有在同一数据集上测试不同的数学公式识别系统才能公平地评判它们的优劣。对于印刷体数学公式，具影响力的数据集有 InftyCDB 系列数据集 [112] 和 Im2latex-100k 数据集 [24]，前者的图片由扫描得到并经人手对其中所有符号和符号间的位置关系进行了标注，后者的图片通过渲染 LaTeX 代码得到。对于联机手写数学公式，毋庸置疑的标杆是联机手写数学公式识别竞赛（Competitions on Recognition of Online Handwritten Mathematical Expressions，CROHME）[75] 的数据集，该竞赛先后在 2011 [71]、2012 [72]、2013 [74]、2014 [73]、2016 [70] 和 2019 [63] 年举行。对于脱机手写数学公式识别，常用的数据集仍然是 CROHME 系列数据集，只是把笔迹渲染成了图片。另外，新近的脱机手写数学公式识别与发现竞

赛（Offline Recognition and Spotting of Handwritten Mathematical Expressions, OffRaSHME) [117] 数据集也受到了注意，其中包含扫描版的手写数学公式。

另外，对于特定的应用场景，可能要求特别的联机手写数学公式识别系统。针对课堂上老师一边讲课一边写黑板的情况，Medjkoune 等人 [66] 提出结合语音来提高识别的准确性，因为形状相似的符号的发音往往是不同的。针对要求提供实时预览或其它对响应时间有较高要求的应用，Phan 等人 [87] 提出增量式的识别方法。

2.2.2 模块化方法

模块化方法把数学公式识别分解为若干个明确定义的子任务，然后各个击破。传统上，数学公式识别被分为符号识别和结构分析两个子任务 [18, 32, 130]。

符号识别的目标是把图片或笔划序列转换为符号集合。一些符号识别方法需要先作符号切分再作符号分类，即首先把笔划或前景像素分为若干组使每组分别构成一个符号，然后区分每组笔划或前景像素分别对应的符号类别。不过，一些方法可以同时进行切分和分类，例如 Winkler 和 Lang [120] 所用的隐 Markov 模型与 Zhelezniakov 等人 [143] 所用的递归神经网络。Awal 等人 [8] 则建议生成多个切分假设，并用可拒识的符号分类器来选取切分假设。

由于数学公式中的符号间可以有复杂的平面位置关系，不像普通文本行那样单方向地书写，符号切分也相对困难。无论是 Okamoto [84] 所用的递归投影切分还是 Suzuki 等人 [111] 所用的连通域分析，面对粘连字符、断裂字符或由多个连通体组成的字符（如 “i”）时都需要特殊处理。对于联机手写数学公式，符号切分可以通过假设每条笔划最多属于一个符号而简化，Shi 等人 [101] 进一步假设了不存在延迟的笔划，但存在一些不满足这些假设的手写数学公式。

符号分类是典型的分类问题，标准的做法是直接进行模板匹配或者先提取特征再使用分类器。对于静态的图片和动态的笔迹，适用的模板匹配方法和特征提取方法不尽相同，一般来说联机识别能用的工具更多。对于图片，Berman 和 Fateman [10] 使用基于 Hausdorff 距离的模板匹配。对于笔迹，Simistira 等人 [102] 提出对比 Freeman 链码, MacLean 和 Labahn [61] 提出用动态时间调整来进行模板匹配，Golubitsky 和 Watt [34] 提出对比参数曲线表示中的系数。特征也可分为静态特征和动态特征，前者仅与形状有关而后者还与书写方式有关，静态特征比较稳定但动态特征有助于区分一些相似符号（如 “p” 与 “ρ”）。常用静态特征有高宽比、长度、矩、穿线数、投影、网格特征、方向直方图等等。常见的动态特征有起/终点、坐标、速度、曲率等等。常见的分类器有最近邻分类器 [111]、决策树 [23]、人工神经网络 [79] 等等。一些系统通过结合多种分类器来提高准确率，例如 Malon 等人 [64] 建议在需要时用支持向量机区分容易混淆的符号。

结构分析的目标是利用符号间的平面位置关系来重构数学公式，大部分结

构分析方法同时适用于印刷体公式、联机手写公式和脱机手写公式。结构分析问题常常被归结为解析某种推广的上下文无关语法的问题，有关语法的产生式带有右端项间位置关系的信息。Anderson [6] 提出了一个自顶而下的解析算法，Chang [19] 提出了一个自底而上的解析算法，Chan 和 Yeung [17] 则提出了一个带回溯的解析算法。另外，也存在一些不依赖于语法的结构分析方法。Okamoto [84] 提出通过递归投影切分自顶而下地分析结构。Lee 和 Lee [56] 提出利用操作符的作用范围来进行分析。Eto 和 Suzuki [27] 提出借助找最小生成树来得出结构。Zanibbi 等人 [128] 提出通过递归地分析基线结构来识别数学公式。

各个模块可以依次被调用，也可以存在更复杂的交互。简单的系统可能按某个顺序调用各模块，例如可以自底而上地先提取符号再进行结构分析，也可以自顶而下地先进行结构分析再识别符号。然而，这种分步方法容易累积错误，一步出错就很可能导致结果出错。同时，各个子任务往往不是完全相互独立的，例如其它符号的大小有助于区分形状几乎相同的大小写字母，符号识别结果又有助于上下标判定。为了用结构信息区分符号或相反，一些方法被提出来以更紧密地整合符号识别和结构分析。利用 Cocke-Younger-Kasami 动态规划算法可以同时优化符号切分、符号识别和结构分析，例如 Chou [21] 提出直接从像素解析印刷体公式而 Yamamoto 等人 [125] 则提议从笔划解析联机手写数学公式，代价则是较高的计算复杂度。Álvaro 等人 [4] 给出了这类系统的统计模型。

2.2.3 基于编码器-解码器框架的方法

基于编码器-解码器框架的方法把数学公式识别系统分为编码器和解码器两部分，其中编码器和解码器一般都由人工神经网络组成。这类系统具有学习复杂依赖关系的能力，可以用于许多不同的问题，类似框架在机器翻译 [110] 与语音识别 [35] 等领域已经取得了令人瞩目的成功。同时，这些系统一般是端到端可训练的，只要有一些输入-输出对即可完成训练，这简化了训练流程和降低了对训练集标注详细程度的要求，同时容许更整体的优化。不过，这类方法的训练过程是高度计算密集的，因此直到近年才成为可行。同时，这类系统缺乏可解释性。

由于输入的类型不同，用于识别笔迹的系统和用于识别图片的系统要使用不同的编码器。相反，由于都是输出数学公式的某种表示，两者可以使用高度相似的解码器。大部分系统都预测 LaTeX 串 [24, 137, 134, 122, 53]，但也有其它系统直接预测树状结构 [138, 136]。

当需要识别印刷体数学公式或脱机手写数学公式时，编码器通常基于卷积神经网络。Deng 等人 [24] 提出的编码器结合了卷积神经网络和递归神经网络。Zhang 等人 [137] 则选取了全卷积网络作为编码器以提高尺度不变性，其后 Zhang 等人 [133] 把编码器从 VGGNet 改为 DenseNet 以更充分地利用来自不同尺度的信息。

当需要识别联机手写数学公式时，编码器通常基于多层双向递归神经网络。Zhang 等人 [135] 使用了门控递归单元（Gated Recurrent Unit，GRU），Zhang 等人 [138] 采用了长短时记忆（Long short-term memory，LSTM），Hong 等人 [39] 则使用了加入了残差连接的门控递归单元。Wang 等人 [118] 又提出使用笔划特征代替点特征来促使对齐更准确。

无论识别的是笔迹还是图片，解码器一般都由递归神经网络给出，同时带有某种机制以处理对齐。Zhang 等人 [138] 使用了局部联结主义时序分类（Connectionist Temporal Classification，CTC）。Deng 等人 [24] 则采用了注意力机制。Zhang 等人 [135, 137] 引进了基于收敛的注意力机制以记录输入中哪些部分已经翻译好。Zhang 等人 [134] 继续修改了注意力机制以便在输出一个单词时决定依赖于输入还是语言模型，从而更可靠地输出没有对应笔划的 LaTeX 单词如花括号。由于注意力机制企图定位输出对应的输入部分，故把它应用于联机识别和脱机识别时要分别实现。

由于通过渲染笔迹可以轻易地得到图片，脱机识别系统也可以用于识别笔迹。由于静态信息和动态信息的互补性，结合两类方法可能可提高准确率。Zhang 等人 [134] 提出在解码时整合联机模型和脱机模型分别预测的概率。Wang 等人 [118] 则提议在编码时就把联机特征和脱机特征结合。

除了优化网络结构，改进训练过程也是一个重要的方向。Zhang 等人 [137] 在训练时使用了神经元丢弃、权重噪声和提前停止来避免过度拟合。Zhang 等人 [134] 提出在损失函数加入一个惩罚项以促使收敛模型与实际对齐一致。Truong 等人则提议 [115] 把识别问题与判定各符号是否出现于公式中的问题联合训练。

由于基于深度学习的方法一般需要大量训练数据，而收集和标注手写数学公式需要耗费大量人力物力，因此有必要设法尽用有限的数据。数据增强可以让系统在训练阶段有机会接触到更多不同的样本，这可以通过事先按某种模式扩大数据集实现，也可以通过在需要时动态生成样本做到。注意到手写体和印刷体对人类来说的相似性，Wu 等人 [123] 建议同时使用手写体图片和对应的印刷体图片进行训练，并用对抗生成网络保证提取出的特征同时适用于两者，Le [52] 则建议通过在损失函数中加入一个惩罚项来促使系统对两者得出相同结果。注意到稍为扰动数学公式不太影响阅读，而数学公式的子公式一般仍然是数学公式，Le 等人 [53] 提出通过形变和分解产生更多的手写数学公式样本来克服训练数据不足的问题。注意到缩放不改变数学公式的含意，Li 等人 [58] 建议在每轮训练前重新缩放图片再填充至固定大小以确保系统的尺度不变性。

2.3 笔划提取

作为一种把脱机手写转换为联机手写的技术，人们曾针对脱机签名验证 [57] 和东亚字符识别 [40] 之类的应用提出过不少笔划提取方法。典型的笔划提取算

法一般是基于过切分的，即先检测子笔划再通过消歧义把它们组合成笔划。不过，有的方法 [25] 通过模糊两步间的界线来结合来自不同层次的时序信息。

提取子笔划的一个常见方法是分解细化所得的骨架，这种方法简单快捷，但对噪声较敏感且未能充分利用灰度信息 [127]。其它用于提取子笔划的方法包括用几何元素如折线和圆弧逼近 [141]、使用 Gabor 滤波器 [108] 和使用 Markov 随机场 [132]。

从子笔划重组笔划的问题通常被视为图中的搜索问题。一类方法利用局部信息对子笔划聚类，Lee 和 Pan [57] 设计了一组启发式规则来跟踪骨架，Boccignone 等人 [13] 尝试通过合并相邻而方向、长度和宽度最接近的子笔划来重新构造笔划，Su 等人 [109] 则用 Bayesian 分类器来判定子笔划间的关系。在对笔划作出某些假设时有可能可以在数学上严格地证明某个笔划提取算法的正确性，Kato 和 Yasuhara [45] 考虑了交叉点处满足特定假设的单笔划情况，Nagoya 和 Fujioka [76] 则在限制笔划相交方式的前提下把这技巧推广到了多笔划情况，不过在应用中一般不能保证有关假设总是成立。另一类方法则把问题归结为整体的优化问题，Jäger [42] 通过最小化笔划内相邻线段的总方向差来重构笔划，Lau 等人 [50] 修改代价函数以把相邻线段间的距离和方向考虑进去。不幸地，这类问题本质上是 NP 完全的旅行商问题，因此在有稍多子笔划时就难以有效地求最优解。

然而，由于本文需要提取手写数学公式中的自然笔划并把它们用作常规联机手写数学公式识别系统的输入，现有的笔划提取方法并不完全适用。首先，针对汉字设计的笔划提取方法提取的常常是线笔划而非自然笔划，而且往往没有提供笔顺信息。可是，联机手写数学公式识别系统一般接受自然笔划，而且可能对笔顺敏感。其次，不少笔划提取方法依赖于特殊的识别方法，例如基于模型 [59] 或要求能容忍各种变形 [82] 的结构匹配方法。可是，有关模型难以从单字符推广到内部有复杂结构且难以切分的数学公式，基于结构匹配的字符识别方法在手写数学符号识别中也缺乏代表性。

2.4 本章小结

手写数学公式识别长期以来一直是一个没有被很好地解决的问题。在过去，人们分别针对笔迹和图片两种输入模态提出过不少数学公式识别方法，其中的许多想法同时适用于两类输入。不过，实现上也有一些不可忽略的差异，例如模块化方法中的符号切分器和编码器-解码器框架中的编码器就难免依赖于输入模态，当识别系统各个部分间紧密耦合时还会扩大影响范围。为了统一地处理两类输入，仅仅把依赖于输入模态的部分分离出来仍然不够理想，因为最小化错误累积与最大化共同部分间存在矛盾。虽然把联机手写识别归结为脱机手写识别是容易的，但这种转换会丢失可能对识别有帮助的时序信息，因此本文采取了相反的做法。为了把脱机手写数学公式识别归结为联机手写数学公式识别，

需要一个笔划提取算法。和适用于普通文本的常规笔划提取器一样，本文使用过切分的想法来从图片中提取手写数学公式的笔划集合。与它们不同的是，本文还需要考虑手写数学公式的笔顺。

第三章　图像预处理

3.1　概述

预处理阶段的目的是消除输入图像中的噪声和其它对笔划提取而言无关重要的细节。预处理阶段的主要步骤有二值化和细化，前者用于从充满噪声的背景中分离出文本成分，后者则用于在大致保持笔划形状的前提下降低前景像素的数量，许多不同的二值化方法和细化方法都可以用于我们的目的。大部分二值化算法都假设输入图像为灰度图像，因此二值化前往往需要先对各颜色通道作加权平均以得到灰度图像。图3-1勾画了预处理阶段。

对于真实的应用场景，输入图片不仅含有数学公式，还带有许多干扰因素。因此，产品级的系统可能需要增加一些额外的步骤，但本文不会详细展开。以下是一些可选的步骤：

- 对于含有其它文本或图形的图片如图3-2a，可能需要进行数学公式定位。常规的对象检测框架即可用于此目的，例如你只看一眼检测器（You Only Look Once，YOLO）[93] 或单射击多框检测器（Single Shot MultiBox Detector，SSD）[60] 可以用于预测数学公式的外接方框，U 形网络 [83] 则可以提供连通体级别的准确性。
- 对于含有背景网格的图片如图3-2b，可能需要作直线检测以找出网格从而把其隐去。
- 对于出现笔划粘连或断裂的图片如图3-2c，可能需要用数学形态学操作以改善笔划的连通性。
- 对于出现被旋转或卷曲的公式的图片如图3-2d，可能需要通过倾斜检测找出倾斜角度以便修正，更复杂的几何模型可以处理卷曲等非线性畸变。不过对于单条的数学公式，符号数目可能不足以可靠地估计倾斜角度，而且

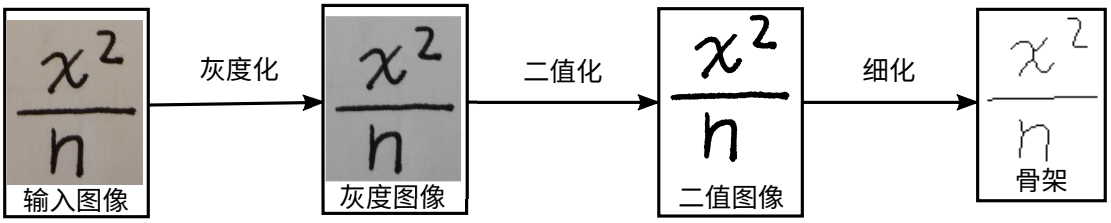
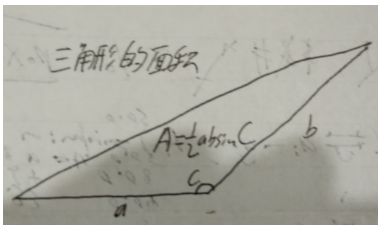
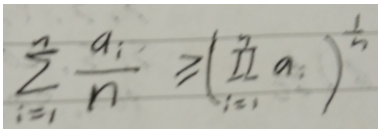


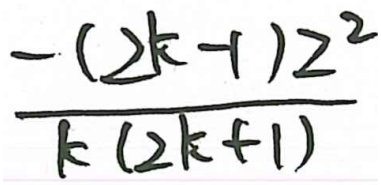
图 3-1 预处理阶段的主要步骤



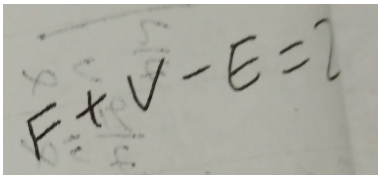
(a) 附近有其它对象的数学公式



(b) 在单行纸上写的数学公式



(c) 笔划频繁粘连的数学公式



(d) 倾斜和卷曲的纸上的数学公式

图 3-2 可能需要额外预处理步骤的公式图片的例子

符号不一定对齐到同一基线，如 "x^{x^x}" 可能会误导基于 Hough 变换等直线检测方法的倾斜角估计。

3.2 局部自适应二值化方法及其内存高效快速实现

3.2.1 常见的二值化方法

二值化是把灰度图像转化为二值图像的图像处理操作，通常用于从背景中区分出对象 [99]。特别地，二值化被广泛用作文档分析系统的预处理步骤，使光学文本识别引擎能够假设文本成分已经被提取出来。二值化的质量同时影响版面分析和字符识别的准确性，对受损文档图像 [33] 和车牌 [5] 尤其重要。

许多流行的二值化方法通过比较像素的灰度值和计算出的阀值工作。按照阀值是否因像素而异，可以把这类方法细分为局部自适应方法和整体方法。与整体方法相比，局部自适应方法对常见的受损情况如噪声、光照不均和穿透更健壮。例如单一阀值对于分离文本和咖啡迹而言可能并不足够。以下，我们把一个 $H \times W$ 灰度输入图像记为 $(I_{ij})_{0\le i<H, 0\le j<W}$，其中按惯例灰度值 $I_{ij} \in \{0, \ldots, 255\}$。相应地，我们把二值图像记为 $(B_{ij})_{0\le i<H, 0\le j<W}$，其中按惯例 $B_{ij} \in \{0, 1\}$。

整体方法对每个输入图像计算一个阀值 T，然后令

$$B_{ij} = \begin{cases} 0 & , I_{ij} \le T \\ 1 & , I_{ij} > T \end{cases}。 \tag{3-1}$$

其中最有代表性的 Otsu 方法 [85] 选择阀值使类间灰度值方差最大以顾及图像的亮度和对比度。

局部自适应方法即使对同一图像的不同像素也使用可能不同的阀值。像素

(i,j) 的阀值 T_{ij} 用它为心的邻域中像素灰度值的某些统计量来计算，然后令

$$B_{ij}=\begin{cases}0 & ,I_{ij}\leq T_{ij}\\ 1 & ,I_{ij}>T_{ij}\end{cases}。 \tag{3-2}$$

通常把邻域取为以 (i,j) 为心的 $h\times w$ 矩形窗口，常用的统计量有平均值 μ_{ij} 和标准差 σ_{ij}。特别地，Niblack [80] 取

$$T_{ij}=\mu_{ij}+K\sigma_{ij}, \tag{3-3}$$

而 Sauvola 和 Pietikäinen [96] 取

$$T_{ij}=\mu_{ij}\left(1+K\left(\frac{\sigma_{ij}}{R}-1\right)\right), \tag{3-4}$$

其中 K 为一个可调整的正参数而 R 通常取为 128，注意

$$\sigma_{ij}\leq(255-0)/2<128。 \tag{3-5}$$

按照定义

$$\mu_{ij}=\frac{1}{hw}\sum_{\substack{i-\frac{h}{2}<k\leq i+\frac{h}{2}\\ j-\frac{w}{2}<\ell\leq j+\frac{w}{2}}}I_{k\ell} \tag{3-6}$$

而

$$\sigma_{ij}=\sqrt{\frac{1}{hw}\sum_{\substack{i-\frac{h}{2}<k\leq i+\frac{h}{2}\\ j-\frac{w}{2}<\ell\leq j+\frac{w}{2}}}I_{k\ell}^2-\mu_{ij}^2}。 \tag{3-7}$$

Bernsen [11] 则取

$$T_{ij}=\begin{cases}\frac{1}{2}\left(\min_{ij}+\max_{ij}\right) & ,\max_{ij}-\min_{ij}\geq C\\ -1 & ,\max_{ij}-\min_{ij}<C\end{cases}, \tag{3-8}$$

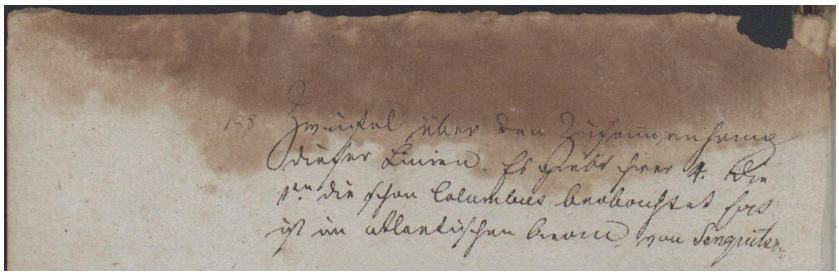
其中 C 为参数，

$$\min_{ij}=\min\left\{I_{k\ell}|i-\frac{h}{2}<k\leq i+\frac{h}{2},j-\frac{w}{2}<\ell\leq j+\frac{w}{2}\right\}, \tag{3-9}$$

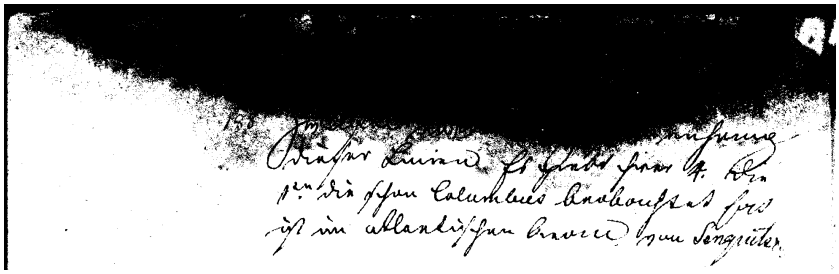
而

$$\max_{ij}=\max\left\{I_{k\ell}|i-\frac{h}{2}<k\leq i+\frac{h}{2},j-\frac{w}{2}<\ell\leq j+\frac{w}{2}\right\}。 \tag{3-10}$$

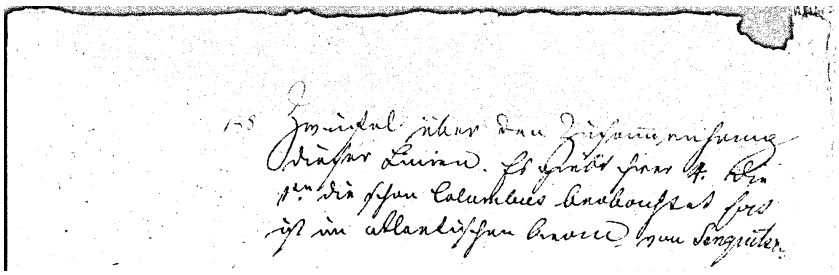
图3-3比较了一些二值化方法应用于某张样本图片时的效果。Otsu 方法把相对暗的一片背景当作了前景，这观察验证了 Otsu 方法 Otsu 对背景亮度变化不健壮，因为它是一个整体二值化方法。Bernsen 方法基本上去除了暗的背景，但一行文本也失去了，这观察验证了 Bernsen 方法在低对比度区域效果欠佳，因



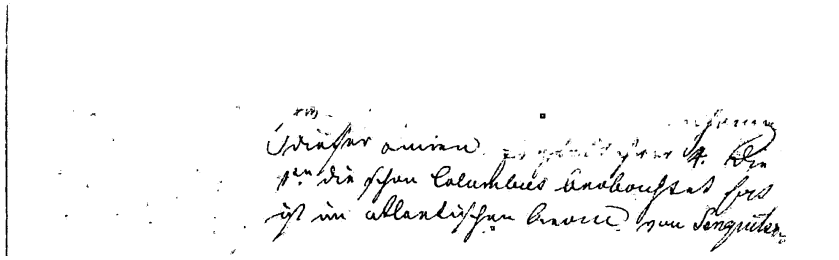
(a) 输入



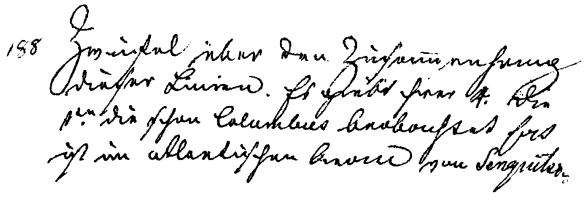
(b) Otsu 方法输出



(c) Sauvola 方法输出（$h = w = 21, K = 0.32$）



(d) Bernsen 方法输出（$h = w = 11, C = 80$）



(e) 理想输出

图 3-3 二值化的例子

为它们都被视为背景。Sauvola 方法在视觉上几乎提取了所有文本，虽然很多字符断裂了且留下了不少噪声。因此，本文系统使用 Sauvola 方法。

存在许多其它二值化方法，而且直到近期仍然有新方法被不断地提出来。除启发式方法外，还有一些基于机器学习的方法。有监督学习可以用于直接把像素分类，例如 Tensmeyer 和 Martinez [113] 设计了一个全卷积神经网络来整合来自不同尺度的信息。非监督学习可以用于对像素聚类，例如 Bhowmik 等人 [12] 用 k-均值算法去聚类受博弈论启发的特征。不过，机器学习算法通常都是计算密集的，所以当受限的计算资源是关注点时不会是首选，特别是在其它方法的结果可以接受时。

3.2.2 局部自适应二值化方法的实现

局部自适应方法现有实现及其局限性

虽然局部自适应方法一般来说比整体方法更鲁棒，但也更计算密集，现有实现要么慢要么挥霍内存。具体地说，Otsu 方法的实现只需 $\Theta(HW)$ 时间和常量辅助空间：先遍历输入图像中所有像素一遍就能得到灰度值直方图，然后就能得到阀值，最后二值图像在下一轮迭代中得到。相反，直接用定义公式计算阀值的话，Sauvola 方法的时间复杂度为 $\Theta(HWhw)$，速度随窗口变大而递减。然而，随着高分辨率图片变得普遍，大窗口正变得更有用。

一个以空间换取时间的著名方法是维护积分图像 [22]。Bradley 和 Roth [14] 用积分图像去加速窗口中灰度均值的计算。Shafait 等人 [100] 进一步用这技巧去计算标准差。注意到加速 Sauvola 方法的关键是高效地计算以下形式的量：

$$\sum_{\substack{a<k\leq b\\ c<\ell\leq d}} f(I_{k\ell}), \tag{3-11}$$

其中 f 为一个函数。令

$$J_{ij}=\sum_{\substack{k\leq i\\ \ell\leq j}} f(I_{k\ell}), \tag{3-12}$$

则

$$\sum_{\substack{a<k\leq b\\ c<\ell\leq d}} f(I_{k\ell})=J_{bd}-J_{ad}-J_{bc}+J_{ac}, \tag{3-13}$$

如图3-4所示。积分图像 $(J_{ij})_{0\leq i<H,0\leq j<W}$ 可以用以下递推关系计算：

$$J_{i,j}=\begin{cases}0 & ,i<0 \text{ 或} j<0\\ (J_{i,j-1}-J_{i-1,j-1})+f\left(I_{i,j}\right)+ & \\ J_{i-1,j} & ,\text{其它}\end{cases}。 \tag{3-14}$$

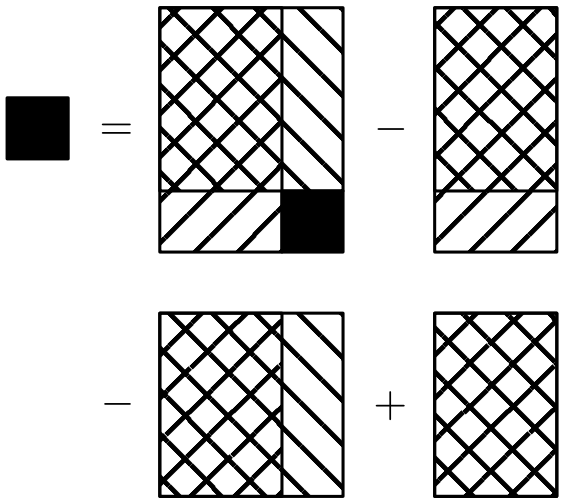


图 3-4 局部自适应二值化方法的积分图像法实现的原理

因此，Sauvola 方法的时间复杂度可以降至 $\Theta(HW)$，并且与窗口大小无关。然而，为了保存两个积分图像，过渡性内存消耗从 $\Theta(1)$ 剧增到 $\Theta(HW)$。它们要占用比输入的灰度图像更大的空间，这种内存消耗对一些应用场景而言过大。

人们还提出了硬件实现。Chen 等人 [20] 提出了一个 Sauvola 方法基于 GPU 的并行实现，但积分图像的构造未有并行化。Najafi 和 Salehi [77] 则提出了一个基于随机电路的完整硬件实现，但它局限于一个小且固定的窗口大小。

许多应用需要局部自适应二值化方法又快又省内存的实现，而且不能依赖于专门的硬件。对于实时应用场景如识别黑板上的内容，学生手中的移动设备要用有限的资源迅速地完成工作。对于批处理应用场景如大型历史文献电子化计划，高效的二值化算法将有助于节省大量时间或硬件。

Sauvola 方法的新实现

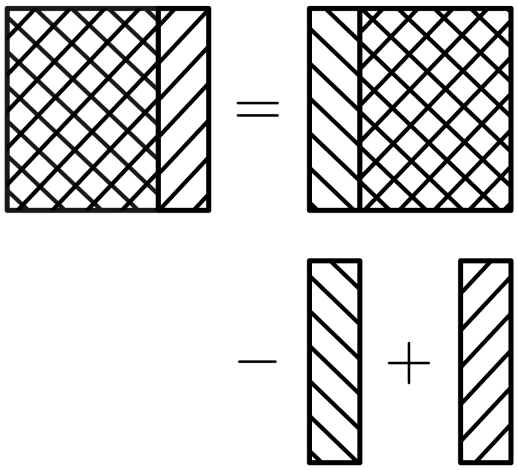
本节提出 Sauvola 方法的一个内存高效且快速的实现，它不要求任何特殊的硬件。在加速 Sauvola 方法的同时尽可能节约内存的想法来源于两个观察。

基本的观察是两个相近的窗口内含有许多公共的元素，因此灰度值的 f 在一个窗口内的总和可以被重用来计算另一个窗口内的总和。形式地，对整数 a, b, c 和d，

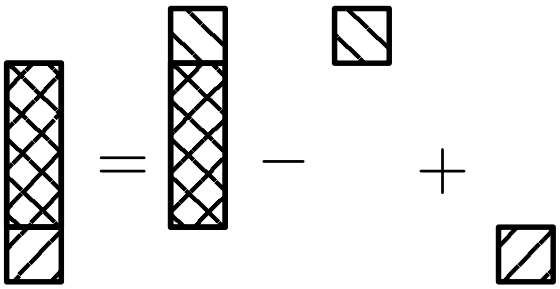
$$\sum_{\substack{a<k\leq b\\ c<\ell\leq d}} f(I_{k\ell}) = \sum_{\substack{a<k\leq b\\ c-1<\ell\leq d-1}} f(I_{k\ell}) + \sum_{a<k\leq b} f(I_{kd}) - \sum_{a<k\leq b} f(I_{kc}), \tag{3-15}$$

如图3-5a所示。类似地，对整数 a, b 和e，

$$\sum_{a<k\leq b} f(I_{ke}) = \sum_{a-1<k\leq b-1} f(I_{ke}) + f(I_{be}) - f(I_{ae}), \tag{3-16}$$



(a) 窗口级



(b) 列级

图 3-5 局部自适应二值化方法的本文实现的原理

如图3-5b所示。因此，若按行主序计算各像素的阀值，实现只需维护量

$$\sum_{i-\lfloor\frac{h+1}{2}\rfloor<k\leq i+\lfloor\frac{h}{2}\rfloor} f(I_{k\ell}), \tag{3-17}$$

其中 $\ell = 0, \ldots, W-1$，而不需要积分图像。由公式（3-15）和（3-16）可见每个像素只需四次加/减法，而由公式（3-13）和（3-14）可见积分图像法中每个像素需要五次加/减法。由于一旦有了一个像素的阀值就能立即二值化该像素，实现不用记忆阀值。

另一个观察是我们不用显式地计算阀值 T_{ij} 以达到二值化的目的。注意到在 K 非负的假设下，

$$I_{ij} \leq \mu_{ij}\left(1 + K\left(\frac{\sigma_{ij}}{R} - 1\right)\right) \tag{3-18}$$

等价于

$$I_{ij} + \mu_{ij}(K-1) \leq 0 \tag{3-19}$$

或

$$\left(I_{ij} + \mu_{ij}\left(K-1\right)\right)^2 \leq \frac{K^2\mu_{ij}^2\sigma_{ij}^2}{R^2}。 \tag{3-20}$$

因此，开根号运算可以代之以远比它快的乘法运算，从而达到强度削减的效果。

结合两个观察，Sauvola 方法可以按算法3.1实现。为简单起见，我们假设输入图像和输出图像在内存隔离，否则 $I_{i-o,j}$ 可能被意外地改写。可以修改算法以容许原地二值化，但需要额外空间以保存 oW 个灰度值。

算法3.1需要 $\Theta(W)$ 辅助空间，不包括输入和输出图像占用的空间。通过调换两个坐标轴，可以实现需要 $\Theta(H)$ 辅助空间的变种。因此，通过在运行时因应高宽比选择一个变种可以使辅助空间需求降至 $\Theta(\min\{H,W\})$，它明显低于积分图像法所需的 $\Theta(HW)$。事实上，

$$\min\{H,W\} \leq \sqrt{HW} \ll HW。 \tag{3-21}$$

另外，更小的中间量意味着可以用更短的整数类型保存。在通常应用中，灰度值从 0 到 255，窗口大小不超过 257，由直接计算

$$255 \times 257 = 65535 = 2^{16} - 1 \tag{3-22}$$

而

$$255 \times 255 \times 257 < 2^{32} - 1, \tag{3-23}$$

可见数组 P 的每个元素只需占用一个无符号 16 位整数而数组 Q 的每个元素只需占用一个 32 位整数。作为对比，一页 A4 文档按 600DPI 扫描后有多达

$$8.27 \times 11.7 \times 600^2 = 34833240 > 2^{32}/255 \tag{3-24}$$

个像素，所以积分图像的每个元素需要占用一个 64 位整数以防溢出。综上所述，积分图像法需要约 $16HW$ 个字节的辅助空间，而本算法只需要约 $6\min\{H,W\}$ 个字节的辅助空间。

算法3.1的时间复杂度为 $\Theta(HW)$ 且与窗口大小无关，和积分图像法一样在量级上最优，Sauvola 方法不可能有次线性时间的串行实现。因为更低的内存占用意味着更高的缓存命中率，加上加减法次数的减少，速度还可能有所提升。当有多个处理器时，算法可以通过用若干个矩形覆盖输入图像来并行化。

其它方法的实现

虽然前文用 Sauvola 方法为例进行阐述，但同样的技巧也可以用于加速很多其它二值化方法。

只用修改算法3.1中 “if” 语句中的条件后即可实现大部分基于 Niblack 方法的局部二值化方法，因为它们的阀值公式也只涉及矩形窗口中的均值和方差（可能还有一些整体量），表 3-1列出了其中一些，其中 L、M 和 S 分别为整体最小值、均值和标准差，α、γ、K、K_1、K_2、p、q 和 R 均为参数，更多的例子参见综述 [97]。显然，这种修改不影响对时间和空间复杂度的分析，所以依然只需 $\Theta(HW)$ 时间和 $O(\min\{H,W\})$ 辅助空间即可二值化一个 $H \times W$ 图像。

算法 3.1: Sauvola 方法的新实现

输入: 灰度位图 $(I_{ij})_{0\leq i<H,0\leq j<W}$
输入: 窗口的宽度 w 和高度 h
输入: 正参数 K 和 R
输出: 二值位图 $(B_{ij})_{0\leq i<H,0\leq j<W}$

$o \leftarrow \lfloor \frac{h+1}{2} \rfloor$;
$u \leftarrow \lfloor \frac{h}{2} \rfloor$;
$l \leftarrow \lfloor \frac{w+1}{2} \rfloor$;
$r \leftarrow \lfloor \frac{w}{2} \rfloor$;
对于 $j \leftarrow 0$ **直到** $W-1$ **进行**
    $P_j \leftarrow 0$;
    $Q_j \leftarrow 0$;
    对于 $i \leftarrow 0$ **直到** $\min\{u-1, H-1\}$ **进行**
        $P_j \leftarrow P_j + I_{ij}$;
        $Q_j \leftarrow Q_j + I_{ij}^2$;
对于 $i \leftarrow 0$ **直到** $H-1$ **进行**
    对于 $j \leftarrow 0$ **直到** $W-1$ **进行**
        如果 $i - o \geq 0$ **则**
            $P_j \leftarrow P_j - I_{i-o,j}$;
            $Q_j \leftarrow Q_j - I_{i-o,j}^2$;
        如果 $i + u < H$ **则**
            $P_j \leftarrow P_j + I_{i+u,j}$;
            $Q_j \leftarrow Q_j + I_{i+u,j}^2$;
    $p \leftarrow 0$;
    $q \leftarrow 0$;
    对于 $j \leftarrow 0$ **直到** $\min\{r-1, W-1\}$ **进行**
        $p \leftarrow p + P_j$;
        $q \leftarrow q + Q_j$;
    对于 $j \leftarrow 0$ **直到** $W-1$ **进行**
        如果 $j - l \geq 0$ **则**
            $p \leftarrow p - P_{j-l}$;
            $q \leftarrow q - Q_{j-l}$;
        如果 $j + r < W$ **则**
            $p \leftarrow p + P_{j+r}$;
            $q \leftarrow q + Q_{j+r}$;
        $n \leftarrow$
        $(\min\{j+r, W-1\} - \max\{j-l, -1\}) \times (\min\{i+u, H-1\} - \max\{i-o, -1\})$;
        $m \leftarrow \frac{p}{n}$;
        $v \leftarrow \frac{q}{n} - m^2$;
        `/*` 实现其它 `Niblack` 型方法时修改下面这行 `*/`
        如果 $I_{ij} + m(K-1) \leq 0$ **或** $(I_{ij} + m(K-1))^2 \leq \frac{K^2 m^2 v}{R^2}$ **则**
            $B_{ij} \leftarrow 0$
        否则
            $B_{ij} \leftarrow 1$

表 3-1 Niblack 型方法的例子

方法	阀值
Niblack [80]	$\mu_{ij} + K\sigma_{ij}$
Sauvola 和 Pietikäinen [96]	$\mu_{ij}\left(1 + K\left(\frac{\sigma_{ij}}{R} - 1\right)\right)$
Wolf 等人 [121]	$\mu_{ij} - K\left(\mu_{ij} - L\right)\left(1 - \frac{\sigma_{ij}}{R}\right)$
Feng 和 Tan [29]	$\alpha\mu_{ij} + K_1\left(\frac{\sigma_{ij}}{R}\right)^{1+\gamma}\left(\mu_{ij} - L\right) + K_2\left(\frac{\sigma_{ij}}{R}\right)^{\gamma} L$
Rais 等人 [92]	$\mu_{ij} + 0.3\frac{\mu_{ij}\sigma_{ij} - MS}{\max\{\mu_{ij}\sigma_{ij}, MS\}}\sigma_{ij}$
Khurshid 等人 [46]	$\mu_{ij} + K\sqrt{\sigma_{ij}^2 + \mu_{ij}^2\frac{hw-1}{hw}}$
Phansalkar 等人 [88]	$\mu_{ij}\left(1 + pe^{-q\mu_{ij}} + K\left(\frac{\sigma_{ij}}{R} - 1\right)\right)$

注意到局部直方图也可以递推地计算，而最大值、最小值、中位数以至任何分位数都可以从直方图轻易得到，类似方法也能够加速用到相应局部统计量的局部自适应二值化方法。事实上，中值滤波器的一种快速实现 [86] 就用了这原理。特别地，Bernsen 方法 [11] 只需 $\Theta(HW)$ 时间和 $\Theta(\min\{W, H\})$ 辅助空间。对于 Bernsen 方法，由于只用到用局部最大值和局部最小值，还有一种巧妙地用最长单调序列取代直方图的实现，它的时间复杂度量级同样最优但常数因子较低。注意 Bernsen 方法的直接实现需要 $\Theta(HWhw)$ 时间和 $\Theta(1)$ 辅助空间。Otsu 方法的局部版本 [81] 也基于局部直方图，所以可以省去极度耗费内存的积分直方图（典型地每像素占用 1 KiB 内存）。

基于本法所支持的二值化方法的复合方法也会受惠。例如可以加入预处理和后处理步骤以降噪 [33]，修改阀值以消除离群值 [16]，或者应用多次局部自适应二值化来找不同大小的文本 [69, 51]。另外，一些预处理过程和后处理过程也可用同样方法加速。例如灰度值的范围可用于估计灰度图像的局部对比度 [107]，一阶矩可以用于在二值图像中寻找孤立的前景或背景像素。

一般地，本文所用技巧的应用决不限于二值化，它能被应用到图像处理中的一些其它任务以消除积分图像的若干典型用法，并且仍然有在大致保持速度的前提下节省内存的效果。比如说，双边滤镜 [90] 和导向滤镜 [37] 之类的滤镜的实现既不需要积分直方图也不需要积分图像，它们的应用包括细节增强、去雾、光照调整、样式化、联合上采样和匹配。由于它们可能需要应用在移动设备如相机上，故计算开销是一个考虑因素。通过一些小修改可得到更大的灵活性，例如滑动窗口的中心不一定为当前像素，多个滑动窗口可以同时考虑等等。然而，它并不能在所有情况下取代积分图像，因为积分图像容许随机访问任意矩形中的总和，而这技巧只能按一定顺序获取总和。

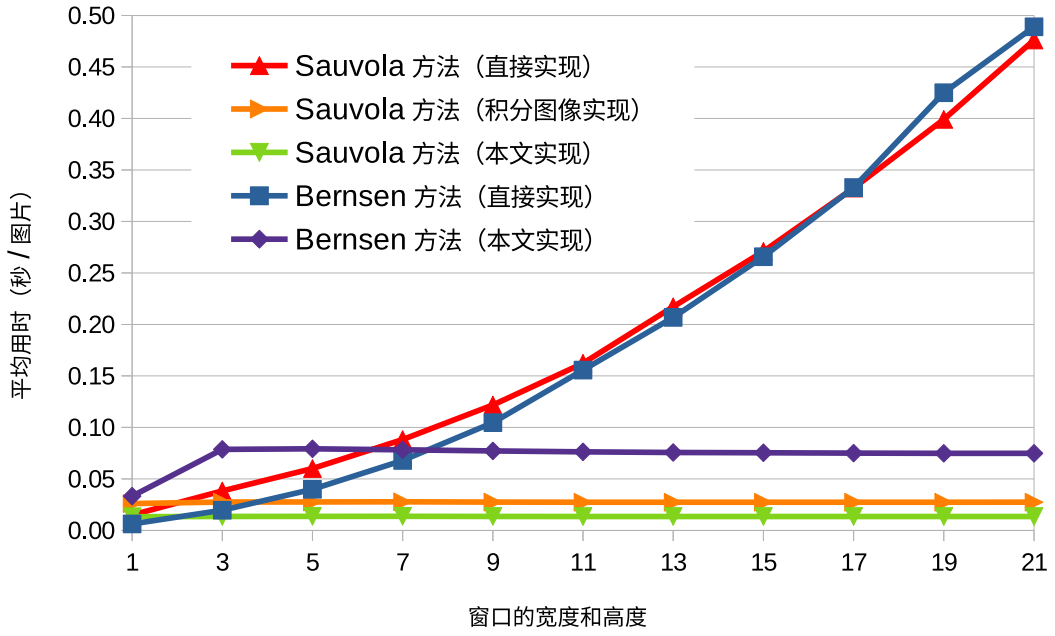


图 3-6 窗口大小与二值化运行时间的关系

速度评测

为了检验本文实现的性能，我们实现了 Sauvola 方法的三种串行算法：直接算法、基于积分图像的算法和本文提出的算法。我们也同时评测了一些其它方法。所有实验用到的程序已经公开[1]。

我们测量了二值化来自 2009 年与 2018 年间历次文档图像二值化竞赛（the Document Image Binarization COntests，DIBCO）和手写文档图像二值化竞赛（Handwritten Document Image Binarization COmpetitions，H-DIBCO）[91] 的 116 张图片所用的时间。实验在一台带 Intel® Core™ i5-7500 CPU @ 3.40GHz 和 8 GB 内存的机器上进行。不论对 Sauvola 方法还是 Bernsen 方法，图3-6都验证了本文实现的运行时间与窗口大小无关，与直接实现不同。表3-2显示了窗口大小为 21 时具体的计算时间。对 Sauvola 方法，本文实现大约比基于积分图像的实现快了一倍，但仍然比 Otsu 方法慢。值得注意的是，用于避免积分图像的增量方法与强度削减技巧都有助于加速，虽然前者主要为节省内存而引入。

[1]代码见`https://github.com/chungkwong/binarizer`。

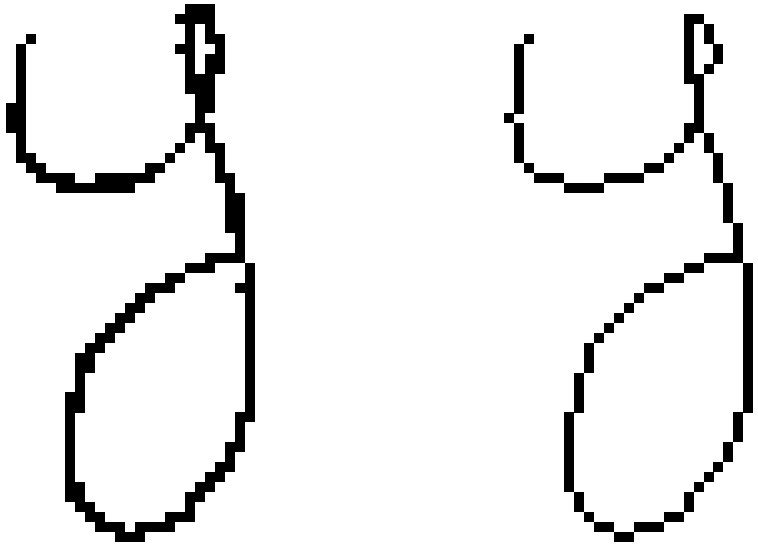
表 3-2 各个二值化方法的运行时间

方法	平均时间（秒/图片）
固定阀值	0.00096
Otsu 方法	0.00151
Sauvola 方法（直接实现）	0.47614
Sauvola 方法（积分图像实现）	0.02739
Sauvola 方法（积分图像实现加上强度削减）	0.01619
Sauvola 方法（本文实现去除强度削减）	0.02458
Sauvola 方法（本文实现）	**0.01349**
Bernsen 方法（直接实现）	0.48895
Bernsen 方法（本文实现）	**0.07483**

3.3 细化

细化把一个二值图像化为单位线宽的骨架，同时尽可能保持拓扑结构。图3-7对比了骨架化前后的样子。

本文选用 Wang 和 Zhang [119] 的细化算法，它是 Zhang 和 Suen [139] 的算法的一个较好地保持斜线形状的变种。这类细化算法的工作原理是不断寻找可以安全地删除的前景像素，直至图像中再没有这样的前景像素。其中，判断一个像素能否删除仅看它的 8-邻域内的其它前景像素的分布情况。



(a) 骨架化前 **(b)** 骨架化后

图 3-7 细化的例子

3.4 本章小结

预处理阶段的目的在于提高信噪比和规范化图像，从而简化后续处理。这阶段的必要步骤为二值化和细化，它们分别被用于去除非文本成分和最小化笔划的宽度。由于要在完整的输入图像上进行操作，这阶段可能成为笔划提取过程中相对耗时和占用较多内存的部分。因此，优化其速度和内存占用是有意义的，这对资源受限设备上的实时应用尤为重要。为此，我们为一大类常见的局部自适应二值化方法提出了内存高效的快速实现。特别地，假定输入灰度图像大小为 $H \times W$ 而窗口大小为 $h \times w$，Sauvola 方法和其它 Niblack 型方法的各种实现的复杂度如表3-3所示，本文方法在理论和实验上都明显优于流行的积分图像法，同时时空权衡比直接算法更平衡。

表 3-3 Sauvola 方法不同实现的复杂度

算法	时间复杂度	辅助空间复杂度
直接算法（Sauvola 等，2000）	$\Theta(HWhw)$	$\Theta(1)$
积分图像法（Shafait 等，2008）	$\Theta(HW)$	$\Theta(HW)$
本文方法	$\Theta(HW)$	$\Theta(\min\{H, W\})$

第四章　笔划提取算法

4.1　概述

本章提出一个适用于手写数学公式的笔划提取算法，图4-1勾画了这个算法。这个算法和一些常规的笔划提取算法一样基于过切分的思想：首先把骨架拆解为子笔划，再把子笔划自底而上地合并为笔划。在理想情况下，每条边恰好属于一条路径而笔划都是光滑的，于是一个简单的贪婪搜索即可。在现实中，一些来自噪声的边应该被忽略，而另外一些来自重笔的边应该被重复经过。因此，搜索前要先作噪声去除，而搜索后要作重笔修正。与很多其它笔划提取算法不同的是，我们的算法需要重建笔顺，而数学公式有更复杂的平面结构，不像自然语言文本那样简单地从左到右或从右到左地书写。不过通常而言，从左上到右下的书写顺序往往是可以接受的。

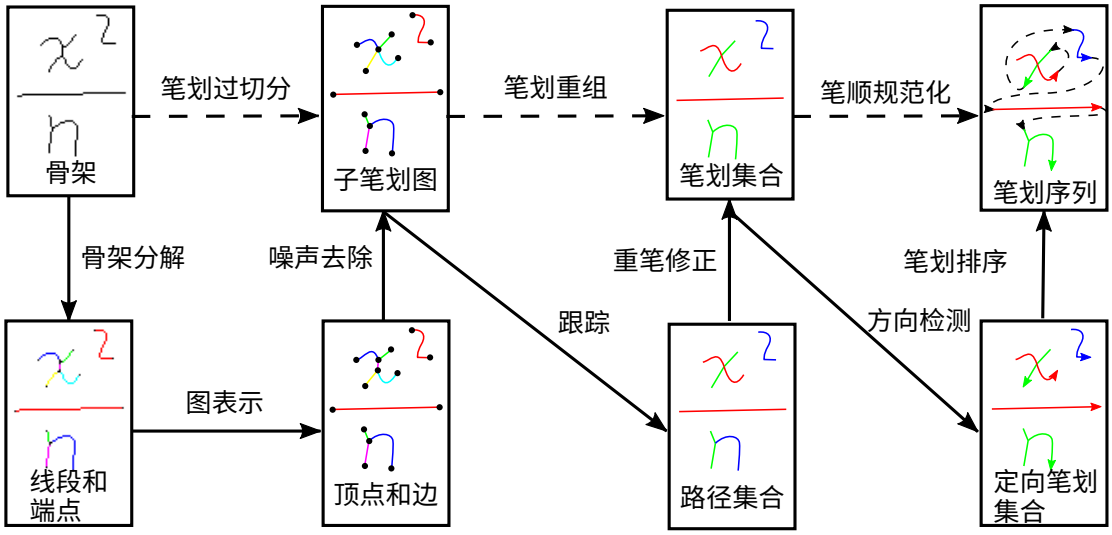


图 4-1 笔划提取的主要步骤

4.2 过切分

4.2.1 骨架分解

骨架的图表示是本文的核心数据结构，它通过把骨架分解为线段和端点而构造出来。

我们把在 8 邻域内有另外两个前景像素而这两个像素不在对方的 4 邻域内的前景像素称为线段像素，其它前景像素则称为端点像素。图4-2可视化了这规则，这个分类方式是一个局部算子。

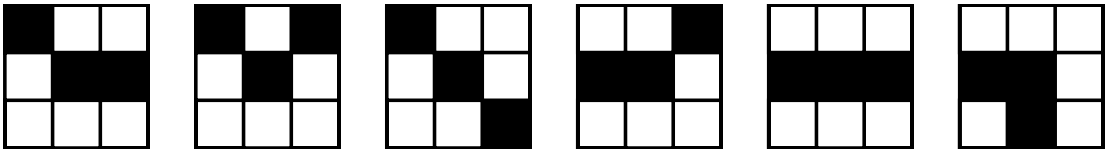
线段像素集的一个连通分支被称为线段，而端点像素集的一个连通分支被称为端点，线段的全体和端点的全体可以用任何标准的连通域分析算法 [38] 求得。图4-3展示了一个分解后的骨架，其中不同线段用不同颜色表示，而端点均用黑色表示。

对于每条线段 S_i，它的像素可以排成一列使相邻的像素在对方的 8 邻域中，形式地，

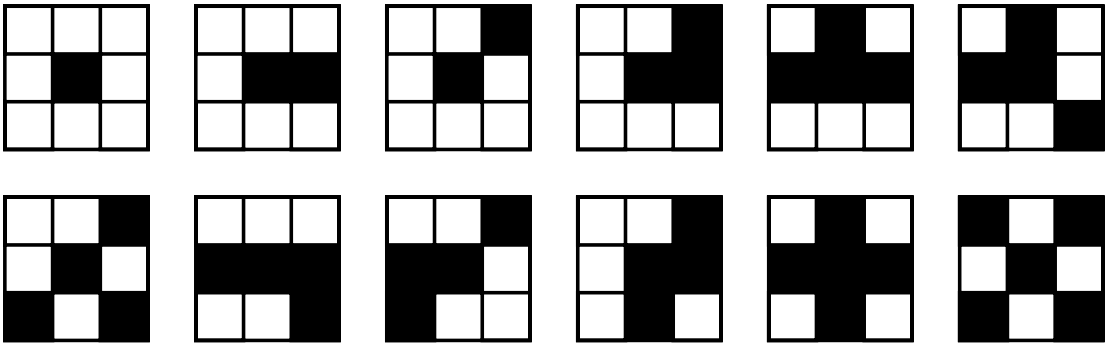
$$S_i = \{p_{i,1}, \ldots, p_{i,\ell_i}\}, \tag{4-1}$$

其中对 $k = 2, \ldots, \ell_i$，$p_{i,k}$ 在 $p_{i,k-1}$ 的 8 邻域中。若 $p_{i,1}$ 在 p_{i,ℓ_i} 的 8 邻域中，线段拓扑上为圆且不碰到任何其它线段或端点，除非 $\ell_i = 1$；否则，线段拓扑上为直线段，$p_{i,1}$ 恰好碰到一个端点而 p_{i,ℓ_i} 也如此，线段的中间像素不碰到任何其它线段或端点。

为了一致性，我们给每个圈强加一个“端点”以保证每条线段有一个起始像素和一个结束像素，它们分别碰到一个端点。这样就可以把端点看作图论意义下的顶点，把线段看作连接两个顶点（可能重合）的边。进一步，这无向图中的路径对应于可能的笔迹，图的连通分支对应于骨架的连通分支。图4-5a展示图4-3中



(a) 这些 3×3 窗口的中心为线段像素



(b) 这些 3×3 窗口的中心为端点像素

图 4-2 前景像素分类的例子

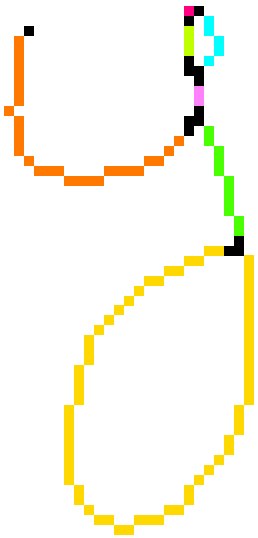


图 4-3 线段和端点的例子

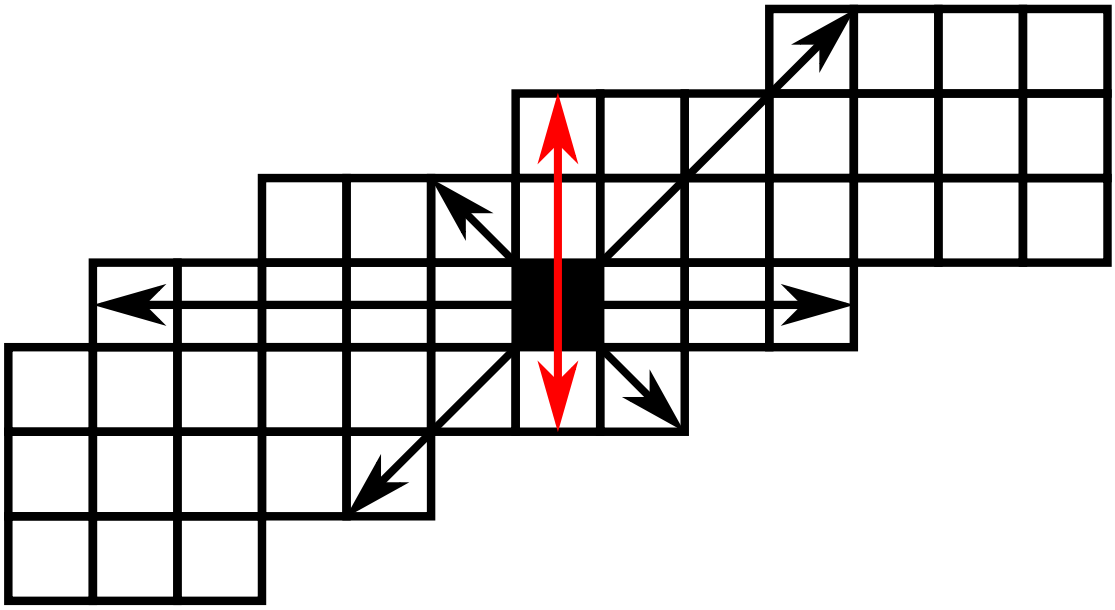


图 4-4 笔划宽度估计的例子

骨架对应的图。

4.2.2 噪声去除

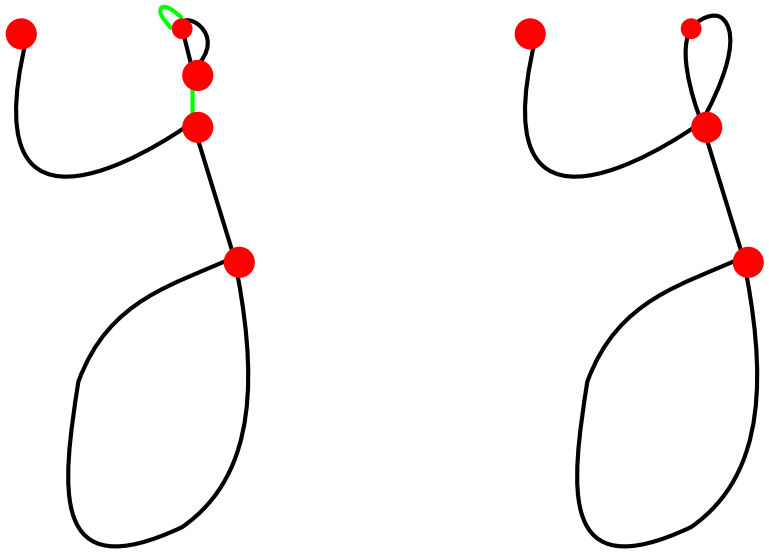
二值图像中的次要特征如椒盐噪声会影响骨架，椒噪声引致孤立的端点而盐噪声引致很短的线段，细化也可能引致形变。由于它们会干扰笔划提取和联机识别，需要设法从图中去除它们。由于绝对的阀值不适用于不同分辨率的图片，我们使用笔划宽度作为参照长度。

笔划宽度变换是一个为每个前景像素赋予笔划宽度估计值的图像算子，它被用于场景文本检测 [26]，因为笔划被视为由宽度局部近似为常数的前景像素组成。一个像素的笔划宽度可以由穿过它的四个方向游程 [28] 中最短者的长度估计，如图4-4中方格表示前景像素而箭号表示实心像素的四个方向游程，实心像素的笔划宽度被估计为红色箭号的长度，它是四者中最短的。这样，只要缓存在特定方向找到的像素数，笔划宽度变换用时相对于二值图像大小线性。

对一个像素集合，可以用其中像素的笔划宽度的最大值来估计其宽度。进而，可以用各线段宽度的平均值来估计笔尖大小。现在，用于去除噪声的规则可叙述如下：

1. 如果一条边的宽度小于笔尖大小的某个倍数，则把它从图中去除并合并与之邻接的两个顶点。
2. 如果一个孤立顶点的宽度小于笔尖大小的某倍数，则把它从图中去除。

图4-5b展示了图4-3中骨架的简化图，原图中有两条来自短线段的边，它们被用绿色标出。其中一条是自环，它在简化图中被移除。另一条关联两个不同的顶点，两者在简化图中被合并。



(a) 简化前 (b) 简化后

图 4-5 噪声去除的例子

4.3 笔划重组

4.3.1 笔划跟踪

显然，图中的孤立顶点对应于数学公式中的点，如小数点或“i”上面的点。因此，我们把各 0 度顶点分别提取为只有一个点的笔划。另外，骨架图中的路径是笔划的候选。虽然可能存在多种方式把边组合成路径，但某些路径更可能是人为的笔迹。以下是一些启发式原则：

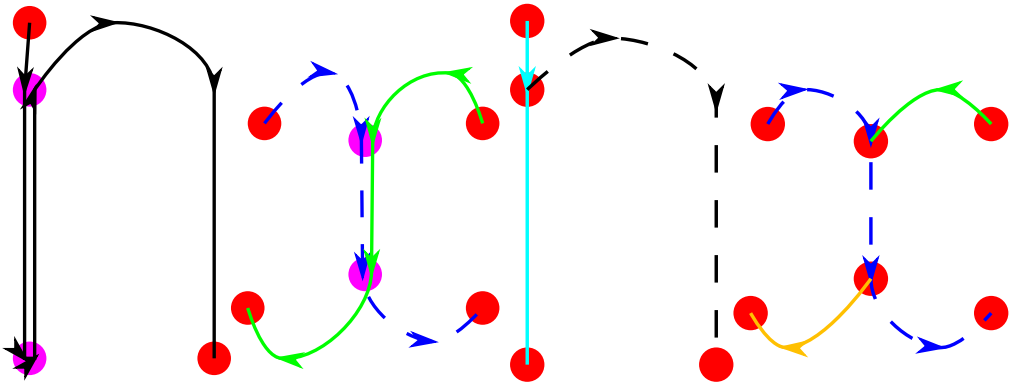
- 笔划的数目应该尽可能少。因为让笔离开纸需要额外的时间，一笔划就最好了。
- 相邻线段间的夹角应该尽可能小。因为突然拐角要减速，流畅的笔划最好是光滑的。

基于以上考虑，我们用自底而上的聚类把每条边刚好指派到一条路径。开始时，每条边单独作为路径。其后，如果有一对路径有公共的起终点，选一对夹角最小的合并。重复这过程至没有路径可以再合并。

值得注意的是，两个原则不完全一致。如果觉得笔划数更重要，可以通过把各路径和与之有公共顶点的回路合并来达到最少，就像找 Euler 路径的算法中的做法。

4.3.2 重笔修正

有时，同一条线段应该被多条笔划共享或者出现在同一笔划中多次，例如表达式“nx”如图4-6a所示的一种常见写法中，洋紫色顶点间的边被重复经过。这时，前述的跟踪方法会产生太多笔划，如图4-6b所示，使得简单的跟踪方法不能



(a) 重笔 (b) 简单笔划

图 4-6 重笔修正的例子

正确地恢复它们。

为了修正重笔，我们寻找应该被共享的线段以便重新连接被断开的笔划，这种线段应该满足以下条件：

- 线段起点与终点不同且都是图中的奇度顶点。否则把它重复后，奇度顶点的数目仍然不会下降。
- 起点与终点分别为上一节给出的路径的起终点，而且夹角不接近于 $\pi/2$。这条件可防止 “T” 的两条笔划被合并。

4.4 笔顺规范化

4.4.1 笔划方向检测

笔划跟踪的方式自然地给出了每条笔划里点的一种顺序，但相反的顺序可能更合理。我们按照人们通常由左到右、由上而下地书写的惯例来推测每条笔划的方向。设笔划首个点和最后一点的坐标分别为 $(x_{\mathrm{START}}, y_{\mathrm{START}})$ 和 $(x_{\mathrm{END}}, y_{\mathrm{END}})$，则

$$\alpha x_{\mathrm{END}} + (1-\alpha)\, y_{\mathrm{END}} < \alpha x_{\mathrm{START}} + (1-\alpha)\, y_{\mathrm{START}} \tag{4-2}$$

时颠倒其方向，其中 α 为可通过实验选取的正参数。

4.4.2 笔划排序

虽然有些数学公式识别系统不受笔划顺序影响 [8, 3]，但其它系统用笔顺来对搜索空间剪枝，因而对笔顺敏感 [103, 54]。事实上，笔顺规范化正是把一个依赖于笔顺的识别器转化为一个笔顺无关识别器的方法 [55]。

同一公式可能可以用不同的笔顺写出。比如说，一些人先写根号而其它人先写被开方数。因此，不能期望可精确地恢复原始笔顺，可以做的只是给予一个合

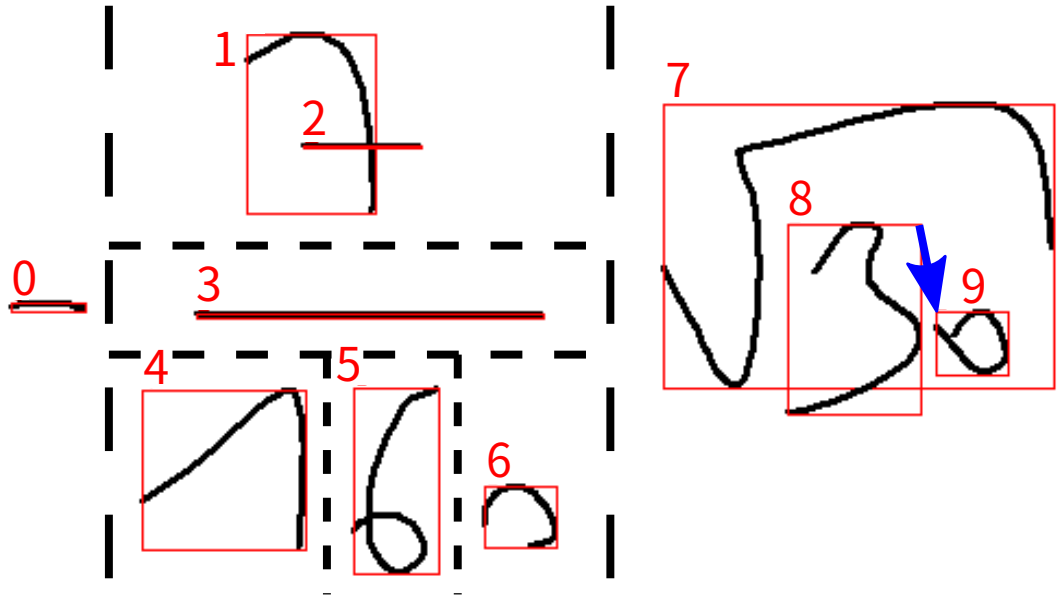


图 4-7 笔划排序的例子

理的笔顺。有时，笔划提取器的错误也可以视为一种修正，可能可以通过消除罕见的笔顺来提高准确度 [116]。

我们按照人们通常由左到右、由上而下地书写的惯例来推测笔划顺序，虽然存在一些例外如人们通常先写求和号再写上限。我们使用两步走的方法对笔划排序。首先，通过递归投影对笔划分组，并由左到右、由上而下地对各组排序。然而，根号内的笔划的顺序不能这样确定，所以我们再对各组内的笔划进行拓扑排序。笔划 T_i 先于 T_j 的条件是以下之一成立：

- T_i 在 T_j 的左方并且两者到 y 轴（但不是 x 轴）的投影相交；
- T_i 在 T_j 的上方并且两者到 x 轴（但不是 y 轴）的投影相交。

进一步的歧义可以用外接矩形左上角的坐标解决，或者使用别的什么启发式规则。图4-7展示了对笔划排序的方式，图中水平或垂直的虚线分隔了由递归投影切分生成的分组，箭头表示拓扑排序中的“先于”关系，红色数字标示了得出的笔划的顺序。

4.5 笔划提取的准确度

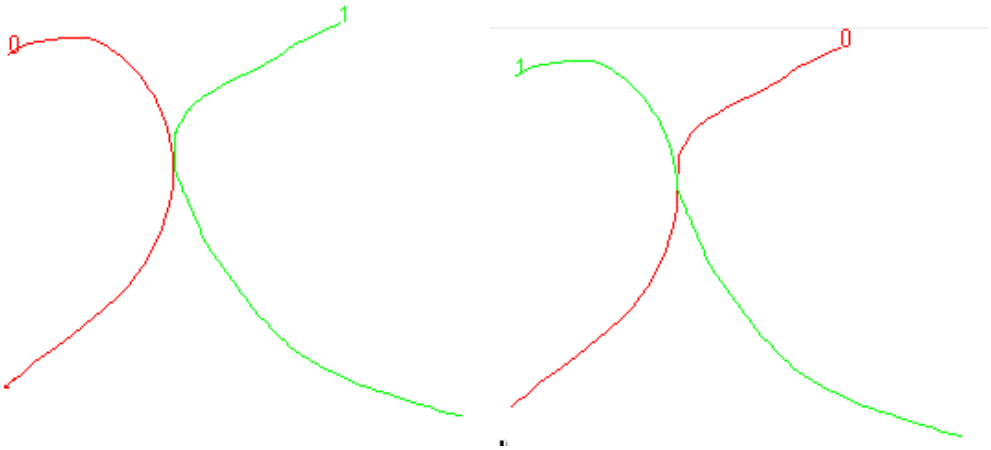
我们在 CROHME 系列数据集上测试了前面描述的笔划提取算法。

首先，我们考虑忽略笔顺时笔划提取的准确度。测试方法是对比原始笔划集和渲染后再提取得到的笔划集，其中当且仅当两条笔划间的 Hausdorff 距离小于用于渲染的笔划宽度的 4 倍时认为它们匹配。实验结果见表4-1，超过一半表达式的笔划集能被本文算法正确地恢复。值得注意的是，我们看到有部分错误其实是可以接受的，比如把“x”的一种写法变为另一种写法，参见图4-8。

接着，我们测试笔划方向检测的准确率。测试方法为对数据集中每条笔划，

表 4-1 忽略笔顺时笔划提取的准确率

CROHME 数据集	笔划		表达式
	召回率（%）	精度（%）	准确率（%）
2014 测试集	91.23	90.75	53.25
2016 测试集	92.41	92.45	58.41
2019 测试集	90.24	90.43	50.29



(a) 原始写法 **(b)** 提取结果

图 4-8 可以接受的笔划提取错误的例子

表 4-2 笔划方向检测的准确率

α	各 CROHME 数据集上的准确率（%）		
	2014 测试集	2016 测试集	2019 测试集
0.0	77.70	79.32	77.61
0.1	85.91	88.82	86.24
0.2	91.66	94.09	91.74
0.3	93.74	95.26	93.31
0.4	**94.38**	**95.46**	**93.84**
0.5	94.04	94.99	93.63
0.6	92.16	93.12	92.01
0.7	89.79	90.28	89.72
0.8	86.45	86.61	86.83
0.9	80.91	80.47	81.70
1.0	73.99	72.37	76.09

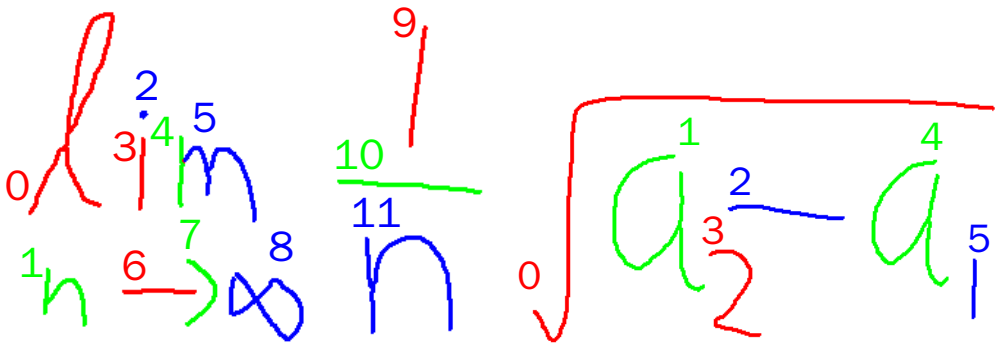
表 4-3 笔划排序的准确率

方法	准确率（%）		
	2014 测试集	2016 测试集	2019 测试集
随机	2.64	1.05	1.00
从左到右	27.59	27.72	26.11
本文方法	**36.11**	**39.58**	**39.12**

检查确定出的方向是否与真实方向相同。表4-2列出了 α 取不同值时的实验结果，大部分笔划的方向是可以准确确定出来的。基于这个实验，后面章节的实验中都把 α 固定为 0.4。

然后，我们考虑笔划排序的准确率。测试方法为对每条公式，（伪）随机地打乱笔划顺序再检查笔划排序结果是否与原来的顺序相同。作为对比的基准，我们也测试了两种平凡的排序方法，其中一种什么都不干，而另一种按笔划外接边框左上角的位置从左到右地排。表4-3表明本文方法明显优于基准方法。同时，我们注意到大部分错误其实也可以接受，比较严重的错误可以分为两类：

- 出现由递归投影切分导致的错误分组。例如算法认为图4-9a中公式 "$\lim_{n\to\infty}\frac{1}{n}$" 的下标 "$n$" 在字符 "$i$" 之前，因为它与 "$l$" 被分到同一个水平组。
- 出现一个符号在另一个的右上方的情况，这时左到右与上到下的惯例冲突。例如算法认为图4-9b中公式 "$\sqrt{a_2-a_1}$" 的下标 "2" 在运算符 "$-$" 后才写，因为 2 在减号的下方。



(a) 带错误的分组的例子 (b) 带错误“先于”关系的例子

图 4-9 典型笔划排序错误的例子

当同时考虑笔划及其顺序时，本文算法分别准确地提取了 CROHME 2014、2016 和 2019 数据集中 23.43%、26.16% 和 23.94% 公式的笔划序列。诚然，即使考虑到一些情况下提取结果与原始的不同而仍然在可以接受的范围内，这个笔划提取准确率也不算高。幸而，笔划提取的准确率对整个脱机识别系统的准确率来说不一定是关键的。

4.6 本章小结

仿照普通笔划提取算法，我们采用过切分再重组的策略。为了优化笔划提取的准确率，我们增加了步骤以去除噪声和修正重笔。由于不少联机手写数学公式识别系统对笔顺敏感，最后还要进行笔划方向检测和笔划排序。在设计笔划提取算法期间，我们是先确定评价指标，再用试验来微调算法以改善有关指标的。

第五章　联机手写数学公式识别

5.1　概述

基于笔划提取的脱机手写数学公式识别的最后一个步骤是识别被提取出来的笔划，如图5-1所示。这就需要一个联机手写数学公式识别系统来把笔划序列转换为公式的语法树。为了说明本方法不依赖于非常规的联机识别系统，本章不会设计新的联机手写识别方法，只会说明如何把现有的系统用于本文的目的，而它们在被设计出来时压根没有考虑过会被这样用。在笔划提取足够准确时，原封不动地使用现成的联机手写数学公式识别系统即可得到理想的准确性。在笔划提取不太准确时，重新训练联机识别系统以让其熟悉提取出的人工笔划有利于提高准确性。

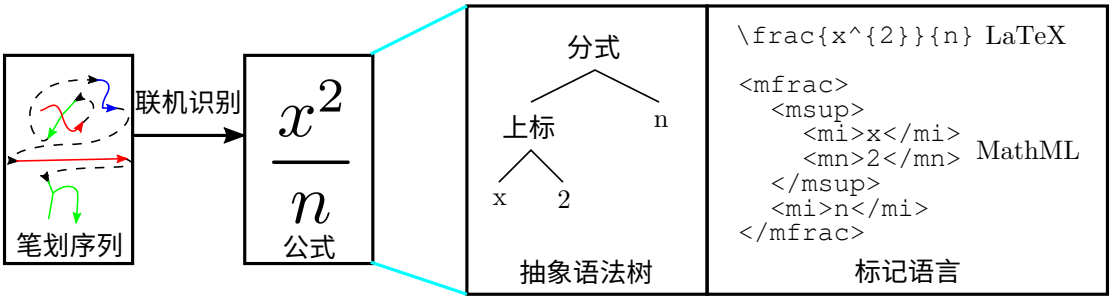


图 5-1 联机手写数学公式识别

5.2 现成的系统

MyScript 是目前最顶尖的联机手写数学公式识别系统之一。它和它的前身 Vision Objects 接连 CROHME 2012 [72]、CROHME 2013 [74]、CROHME 2014 [73] 和 CROHME 2016 [70] 的联机数学公式识别项目都以压倒性的准确率取得冠军，它在 CROHME 2019 的联机数学公式识别项目中也提交过准确率最高的结果，仅因为竞赛期间因操作失误撤回了它而失去了竞赛的冠军。MyScript 分析公式中各部分间的位置关系以找出符合语法的切分假设。MyScript 的符号识别器结合了两个人工神经网络，一个是处理若干静态特征（如投影）和动态特征（如方向和曲率）的深度多层感知器，另一个则是直接处理输入笔划的双向 LSTM 递归神经网络，其中后者用了 CTC 损失函数去训练。同时，MyScript 结合了一些统计语言模型和基于 LSTM 的语言模型来利用上下文信息以微调符号识别结果。虽然 MyScript 的结构分析模块对笔顺不太敏感，但符号识别模块对笔顺是敏感的。

MyScript 提供了软件开发工具包，可供开发者有限度地免费使用。本文主要实验中使用发布于 2019 年 3 月的 MyScript Interactive Ink 版本 1.3[1]，它提供了 Windows、iOS 和 Android 平台上的原生版本，另有云版本。本文主要实验使用的是适配 Android 平台的。MyScript 提供了一些配置选项，我们把大部分选项维持在默认值，而只使用了定制的语法以消除不出现在 CROHME 数据集中的符号和构造，该语法如表5-1所示，其中 “< start >” 为开始符号。

虽然 MyScript 是一个优秀的联机数学公式识别系统，但它对一般开发者来说很大程度上是个黑盒。灵活性的不足使一些较深入的实验难以进行，例如那些需要对联机识别系进行重新训练的实验。重新训练使得我们可以报告本文方法在公开数据集上的表现，以便公平地与其它脱机识别系统相比较。同时，这也使我们能探讨数据集工程方法。

[1]这个识别器的文档见https://developer.myscript.com/docs/interactive-ink/1.3/overview/about/。

表 5-1 针对 CROHME 数据集的定制语法

< start >	::=	< term >
< term >	::=	< operand >
	|	< symbol2 >
	|	< symbol4 >
	|	< term >< term >
	|	$\frac{\text{<term>}}{\text{<term>}}$
	|	$\text{< operand >}_{\text{<term>}}$
	|	$\text{< operand >}^{\text{<term>}}$
	|	$\text{< operand >}_{\text{<term>}}^{\text{<term>}}$
	|	< symbol3 > < term >
	|	< term > < symbol4 > < term >
<operand>	::=	< symbol1 >
	|	$\sqrt{\text{< term >}}$
	|	$\sqrt[\text{<term>}]{\text{< term >}}$
	|	< symbol5 >< term >< symbol6 >
< symbol1 >	::=	'0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'
	|	'A' | 'B' | 'C' | 'E' | 'F' | 'G' | 'H' | 'I' | 'L'
	|	'M' | 'N' | 'P' | 'R' | 'S' | 'T' | 'V' | 'X' | 'Y'
	|	'a' | 'b' | 'c' | 'd' | 'e' | 'f' | 'g' | 'h' | 'i' | 'j' | 'k'
	|	'l' | 'm' | 'n' | 'o' | 'p' | 'q' | 'r' | 's' | 't' | 'u'
	|	'v' | 'w' | 'x' | 'y' | 'z' | 'Δ' | 'α' | 'β' | 'γ'
	|	'θ' | 'λ' | 'π' | 'σ' | 'φ' | 'ϕ' | 'μ' | '<' | '>' | '!'
	|	'/' | '∞' | 'sin' | 'cos' | 'tan' | 'lim' | 'log'
< symbol2 >	::=	'+' | '-' | '$\pm$' | '$\times$' | '$\div$' | '$\cdot$' | '=' | '$\prime$' | ',' | '.'
	|	'...' | '$\rightarrow$' | '$\exists$' | '$\in$' | '$\neq$' | '$\leq$' | '$\geq$' | '$\forall$'
< symbol3 >	::=	'$\sum$' | '$\int$' | 'lim'
< symbol4 >	::=	'$\sum$' | '$\int$'
< symbol5 >	::=	'(' | '[' | '{' | '|'
< symbol6 >	::=	')' | ']' | '}' | '|'

5.3 端到端可训练的系统

5.3.1 序列到序列模型

跟踪、注意和解析（Track, Attend, and Parse, TAP）[134] 是一个开源的端到端可训练联机手写数学公式识别系统[2]。也就是说，只要有一些数学公式的笔划序列和对应的 LaTeX 代码即可进行训练。这不仅可简化训练过程，例如不用像传统的模块化系统那样要分别训练符号切分器、符号识别器和位置关系分类器等组件，而且可减低人手标注数据的工作量，例如不一定需要输入与输出的对齐信息。不过，由于端到端的人工神经网络模型对先验知识的要求低，需要收集大量数据来训练。

TAP 把联机手写数学公式识别建模为从一个不定长序列到另一个不定长序列的问题。笔划序列和数学公式分别被建模为由一些固定维数向量组成的序列，其中前者中的每个元素对应于某笔划上的某个点，后者中的每个元素对应于某 LaTeX 单词。作为输入的笔划采样点的数目因笔划序列而异，而作为输出的 LaTeX 代码的长度也因数学公式而异。

假设我们把笔划上的每个采样点分别用一个 F 维实向量表示，则笔划序列可以表示为 $\mathbf{x} = (x_k)_{k=0}^{m-1}$，其中对任何 $k = 0, \ldots, m-1$，有 $x_k \in \mathbb{R}^F$。为了说明对笔划序列进行编码的原理，例5.3.1给出了笔划序列的一种简单而完备的表示。虽然直接用这种简单的表示是可行的，但我们希望避免对现有的联机识别系统作不必要的改动，所以我们沿用了 TAP 实际使用的编码方式，它要比这例子稍为复杂一些。首先，它忽略了部分信息量不大的采样点（如与相邻采样点太近或与它们几乎成一直线的）和进行了坐标标准化 [140]。其次，它使用了带冗余的 8 维表示方式 [134]，其中有 4 维用于表示坐标的差分，剩下 1 维用于表示是否不是某笔划的最后一个采样点。

例 5.3.1 一种编码方案用 3 维向量表示一个采样点，前两维分别为横坐标和纵坐标，最后一维表示它是否某笔划的最后一个采样点。按照这方案，图5-2所示的笔划序列可以表示为

$$
\begin{aligned}
&((-1.26, -1.58, 0), (-0.67, -1.78, 0), (0.05, -0.72, 0), (0.43, -1.01, 1),\\
&(0.16, -1.96, 0), (-0.92, -0.49, 1),\\
&(1.06, -2.86, 0), (1.61, -2.56, 0), (1.29, -1.69, 0), (1.78, -1.66, 1),\\
&(-2.29, 0.51, 0), (2.21, 0.44, 1),\\
&(-0.97, 1.35, 0), (-0.95, 2.88, 0), (-0.97, 1.86, 0), (-0.03, 1.58, 0),\\
&(0.29, 1.94, 0), (0.20, 2.99, 1))
\end{aligned}
$$

[2]TAP 的原始版本见https://github.com/JianshuZhang/TAP。我们实际使用的程序、数据集和预训练模型见https://github.com/chungkwong/mathocr-tap。

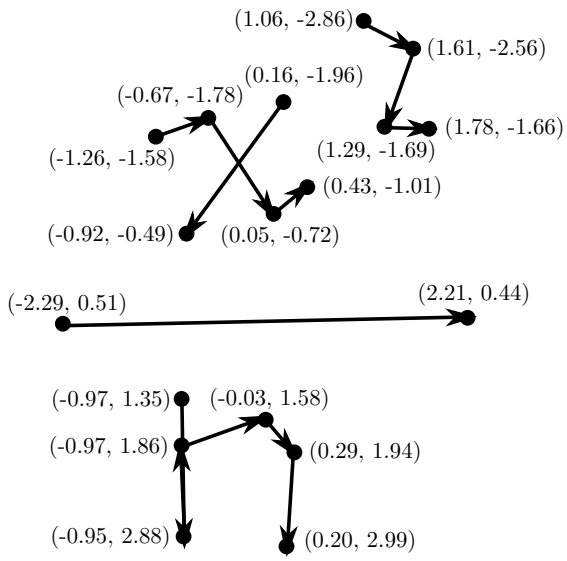


图 5-2 笔划序列采样的例子

再假设我们所考虑的公式的 LaTeX 代码不涉及 S 个 LaTeX 单词 $t_1, \ldots, t_S$ 外的其它单词，我们按标准的单词嵌入方法把这些单词分别用独热码表示为 $\mathbb{R}^{S+1}$ 中的向量 $e_1 = (0, 1, 0, \ldots, 0), \ldots, e_S = (0, 0, 0, \ldots, 1)$，而 $e_0 = (1, 0, \ldots, 0)$ 则用作表示结束符 t_0。这样，LaTeX 代码 $t_{i_0} \cdots t_{i_{n-1}}$ 可看作序列 $\mathbf{y} = (y_j)_{j=0}^{n} = \big(e_{i_0}, \cdots, e_{i_{n-1}}, e_0\big)$。例5.3.2说明了如何对输出编码。

例 5.3.2 一种可能的单词嵌入字典如表5-2所示。注意到 LaTeX 代码“`\frac{x^{2}}{n}`”依次由单词“`\frac`”、“`{`”、“`x`”、“`^`”、“`{`”、“`2`”、“`}`”、“`}`”、“`{`”、“`n`”和“`}`”组成，它们的编号依次为 79、53、27、71、53、61、106、106、53、75 和 106，所以上述 LaTeX 串对应于序列

$$\big(e_{79}, e_{53}, e_{27}, e_{71}, e_{53}, e_{61}, e_{106}, e_{106}, e_{53}, e_{75}, e_{106}, e_0\big)\ 。$$

简单的序列至序列模型由编码器和解码器两部分组成，前者把不定长的输入映射为固定维数的中间量，后者则把中间量映射为不定长的输出。编码器和解码器又分别用隐 Markov 模型进一步拆解为多次应用从固定维数空间到固定维数空间的映射，编码器每步把输入序列中的一个元素提供的信息加到中间量中，而解码器每步则从中间量中提取信息以预测输出序列的一个元素。实际的 TAP 模型要更复杂一些，它的预测流程如算法5.1所示，其中 f 和 g 是已经设计好的函数，而 Θ 则是在训练阶段确定的。

表 5-2 LaTeX 单词嵌入字典的例子

单词	编号	单词	编号	单词	编号	单词	编号
结束符	0	x	27	\Delta	54	\div	81
\geq	1	|	28	\neq	55	!	82
\sqrt	2	\gamma	29	\forall	56	\phi	83
\leq	3	l	30	\alpha	57	)	84
/	4	'	31	\times	58	-	85
\infty	5	+	32	\lim	59	1	86
\cos	6	\theta	33	.	60	5	87
,	7	3	34	2	61	9	88
0	8	7	35	6	62	=	89
4	9	\in	36	\lambda	63	A	90
8	10	\int	37	>	64	E	91
<	11	\sin	38	B	65	\beta	92
\pm	12	C	39	F	66	I	93
\log	13	G	40	\exists	67	M	94
\pi	14	\ldots	41	N	68	Y	95
H	15	S	42	R	69	]	96
L	16	[	43	V	70	\mu	97
P	17	_	44	^	71	a	98
T	18	c	45	b	72	e	99
X	19	\sigma	46	f	73	i	100
d	20	g	47	j	74	\sum	101
h	21	k	48	n	75	m	102
\}	22	o	49	r	76	q	103
\{	23	s	50	v	77	u	104
p	24	w	51	z	78	y	105
t	25	(	52	\frac	79	}	106
\tan	26	{	53	\rightarrow			

算法 5.1: TAP 识别算法框架

输入: 笔划序列的表示 $\mathbf{x} = (x_k)_{k=0}^{m-1}$
输入: 集束候选数上限 k
输出: 候选与置信度对的集合 Y
数据结构: 固定维数向量 z，记录对后续单词预测有用的信息
数据结构: 特征序列 $\mathbf{C}$，每项表示对应于若干个样本点的特征
数据结构: 收敛向量 α，用于标记哪些样本点已经有对应的输出单词
数据结构: （未完成）候选的表示 $\mathbf{y}$
数据结构: （未完成）候选的最后一个单词的表示 y
数据结构: （未完成）候选预测概率的负对数 s，越低表示候选越可靠
数据结构: 下一单词的预测概率分布 $p = (p_i)_{i=0}^{S}$，其中 p_i 表示下一单词为 t_i 的预测概率

$Y \leftarrow \emptyset$;
$(z, \mathbf{C}) \leftarrow f(\mathbf{x}; \Theta)$;
$H \leftarrow \left\{\left(\lambda, 0^{S+1}, 0, z, 0^{|\mathbf{C}|}\right)\right\}$;
当 真 **进行**
    $H' \leftarrow \emptyset$;
    对于 $(\mathbf{y}, y, s, z, \alpha) \in H$ **进行**
        $(p, z', \alpha') \leftarrow g(y, \mathbf{C}, z, \alpha; \Theta)$;
        $H' \leftarrow H' \cup \{(\mathbf{y} * (e_j), e_j, s - \log p_j, z', \alpha') \mid j \in \{0, \ldots, S\}\}$;
    $H' \leftarrow H'$ 中按第三分量排首 k 个元素组成的集合;
    $H \leftarrow \emptyset$;
    对于 $T = (\mathbf{y}, y, s, z, \alpha) \in H'$ **进行**
        如果 $y = e_0$ **则**
            $Y \leftarrow Y \cup \{(\mathbf{y}, s)\}$;
            $k \leftarrow k - 1$;
        否则
            $H \leftarrow H \cup \{T\}$;
    如果 $k \leq 0$ **则**
        跳出;

5.3.2 训练方法

本文的主要实验中，训练方法仿照 TAP 原来的实验。训练阶段使用的优化器为 adadelta。目标函数主要跟标注单词的表示与预测概率分布间的交叉熵有关，在有对齐标注可用时也跟它与收敛向量间的交叉熵有关。每批一般由 8 条数学公式组成。开始时学习率为 10^{-8}，当进行 15 轮优化后在检验集上的单词错误率都没有再改进时就把学习率降为原来的十分之一，直到再次出现 15 轮优化都没有改善的情况。接着，引入参数噪声再重复上述过程，只是容许学习率下降两次。

为了提高应用到提取出来的笔划时的准确性，应当使笔划提取器的输出与联机识别器的预期输入更接近。于是在我们面前有两条路，一是改进笔划提取器使之更忠实地还原书写的笔划，二是重新训练联机识别系统使之适应提取出的人工笔划。改进笔划提取器看似是最明显的选择，但这是困难的。加入更多启发式规则很快会导致严重的可维护性问题，而边际效益却在递减。因此，许多研究者已经放弃了这一方向。开发数据驱动的笔划提取器是有希望的，但如果用卷积神经网络和递归神经网络之类笨重的模型去预测笔迹，何不直接预测数学公式本身？可见，重新训练联机识别系统是一个务实的选择。只要识别器在训练阶段见过笔划提取器生成的人工笔划，它就应该能学会它们的含义。利用这个技巧，人工笔划和书写笔划可以解耦，笔划提取器因而不用过于关注边角的情况。

5.3.3 形式语法模型

把几何模型和语言模型耦合在一起也有其缺点。首先，产生的语言模型往往太弱，系统甚至不时会预测出一些不符合 LaTeX 语法的串，因为手写数学公式样本的收集和标记相当耗时，导致训练数据不足以让模型学会语法。相反，从 arXiv[3]或维基百科[4]之类的网站可以轻易地收集到海量的数学公式，可以用于训练强大的语言模型。其次，模型的灵活性较低，因为应用到另一领域时不能单独地修改语言模型。分领域训练的话则会使每个领域的训练数据都不足，从而导致更差劲的几何模型。因此，把单独的语言模型整合到端到端系统中仍然是有意义的。

在编译器中广泛应用的形式语言理论 [1] 给出了定义语言模型的一种方法，限制输出序列严格地符合特定的语法在应用中是重要的。首先，这可以用于保证输出的 LaTeX 代码合法且对应于一条有意义的数学公式，从而可以安全地用作其它软件的输入以对数学公式进行后续的自动化处理。例如表5-3给出的 $LL(1)$ 语法定义了数学模式 LaTeX 的一个子语言，它排除了一些明显不合法（如花括号不配对或带没有分母的分式）或不合理（如有空分组）的串。其次，这使得在

[3]参见https://arxiv.org/。

[4]参见https://en.wikipedia.org/wiki/Main_Page。

不重新训练的情况下定制系统用于识别某个子语言变得容易，就像一些商用的手写识别系统可以通过设置不同的字典来识别不同类型的文档。比如说，简单的化学反应式集和四则运算式集构成数学公式的子集，因此可以在大型手写数学公式数据集上进行训练以学会更鲁棒的特征，同时防止在识别小学生的算术作业时把 “6” 误识为 “σ” 之类的情况。

把常见的形式语言解析器集成到算法5.1中并不困难，只用为未完成候选维护解析状态。算法5.2演示了如何集成针对 $LL(1)$ 语法的表驱动预测分析器，稍作修改后也可以用于解析 $LL(k)$ 语法。更一般地，可以集成针对 $LR(k)$ 语法的移入-归约分析器，通过维护动态规划用表甚至可以集成适用于任意上下文无关语法的 CYK 分析器。

5.3.4 组合模型

组合多个模型是提高分类器准确率的有效方法 [94]。在上述框架下容易通过加权平均多个模型预测出的概率向量来组合模型。更准确地说，给定已确定参数的模型 $(f_1, g_1), \ldots, (f_\ell, g_\ell)$ 和权重 $\beta_1, \ldots, \beta_\ell$ 使

$$\sum_{i=1}^{\ell} \beta_i = 1\,, \tag{5-1}$$

令

$$f\left(\mathbf{x}\right) = \left(\left(z_1, \ldots, z_\ell\right), \left(\mathbf{C}_1, \ldots, \mathbf{C}_\ell\right)\right), \tag{5-2}$$

其中

$$\left(z_i, \mathbf{C}_i\right) = f_i\left(\mathbf{x}\right), \tag{5-3}$$

又令

$$g\left(y, \left(\mathbf{C}_1, \ldots, \mathbf{C}_\ell\right), \left(z_1, \ldots, z_\ell\right), \left(\alpha_1, \ldots, \alpha_\ell\right)\right) = \left(\sum_{i=1}^{\ell} \beta_i p_i, \left(z'_1, \ldots, z'_\ell\right), \left(\alpha'_1, \ldots, \alpha'_\ell\right)\right), \tag{5-4}$$

其中

$$\left(p_i, z'_i, \alpha'_i\right) = g\left(y, \mathbf{C}_i, z_i, \alpha_i\right), \tag{5-5}$$

则 (f, g) 即为组合模型。

我们可以组合几个同质的模型。从几个相互独立地随机初始化的参数出发优化 TAP 模型，一般会得到几个不同的 TAP 模型，每个都分别抓住了稍为不同的方面，于是把它们组合起来往往会得到一个更全面的模型。

我们也可以再组合一些异质的模型。特别地，组合在大量 LaTeX 代码上训练过的语言模型时，可以通过利用上下文信息来提高识别的准确性，使结果更符合现实中数学公式的惯用法。在传统的光学字符识别中，语言模型对提高准确性起了关键性的作用。如果相互独立地识别每个字符，则准确率会随着文本长度增

表 5-3 数学模式 LaTeX 子语言的 $LL(1)$ 语法的例子

```
    <start>  ::=  <beginning> <remaining>
<beginning>  ::=  <operand>
               |  <symbol2>
               |  '\frac' <group> <group>
<remaining>  ::=  λ
               |  <term> <remaining>
     <term>  ::=  <beginning>
               |  '_' <group>
               |  '^' <group>
    <group>  ::=  '{' <start> '}'
  <operand>  ::=  <symbol1>
               |  '\sqrt' <sqrt>
               |  '(' <start> ')'
               |  '[' <start> ']'
               |  '\{' <start> '\}'
     <sqrt>  ::=  <group>
               |  '[' <start> ']' <group>
  <symbol1>  ::=  '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'
               |  'A' | 'B' | 'C' | 'E' | 'F' | 'G' | 'H' | 'I' | 'L'
               |  'M' | 'N' | 'P' | 'R' | 'S' | 'T' | 'V' | 'X' | 'Y'
               |  'a' | 'b' | 'c' | 'd' | 'e' | 'f' | 'g' | 'h' | 'i' | 'j' | 'k'
               |  'l' | 'm' | 'n' | 'o' | 'p' | 'q' | 'r' | 's' | 't' | 'u'
               |  'v' | 'w' | 'x' | 'y' | 'z' | '\Delta'
               |  '\alpha' | '\beta' | '\gamma' | '\theta' | '\lambda'
               |  '\pi' | '\sigma' | '\phi' | '\mu' | '<' | '>' | '!'
               |  '/' | '\infty' | '\sin' | '\cos' | '\tan' | '\log'
               |  '\lim' | '\sum' | '\int'
  <symbol2>  ::=  '+' | '-' | '|' | '\pm' | '\times' | '\div'
               |  '=' | ''' | ', ' | '.' | '\ldots' | '\rightarrow' | '\exists'
               |  '\in' | '\neq' | '\leq' | '\geq' | '\forall'
```

算法 5.2: 带 $LL(1)$ 语法约束的 TAP 识别算法框架

输入: 笔划序列的表示 $\mathbf{x} = (x_k)_{k=0}^{m-1}$
输入: 集束候选数上限 k
输入: $LL(1)$ 语法 G 的预测分析表 T，
$T(X, a)$ 表示解析非终结符 X 时若序列首个符号为 a 的话要用的产生式
输出: 候选与置信度对的集合 Y
数据结构: 固定维数向量 z，记录对后续单词预测有用的信息
数据结构: 特征序列 $\mathbf{C}$，每项表示对应于若干个样本点的特征
数据结构: 收敛向量 α，用于标记哪些样本点已经有对应的输出单词
数据结构: （未完成）候选的表示 $\mathbf{y}$
数据结构: （未完成）候选的最后一个单词的表示 y
数据结构: （未完成）候选预测概率的负对数 s，越低表示候选越可靠
数据结构: 预测分析栈 $\mathbf{t}$，用于记录解析状态
数据结构: 下一单词的预测概率分布 $p = (p_i)_{i=0}^{S}$，其中 p_i 表示下一单词为 t_i 的预测概率

$Y \leftarrow \emptyset$;
$(z, \mathbf{C}) \leftarrow f(\mathbf{x}; \Theta)$;
$H \leftarrow \left\{\left(\lambda, 0^{S+1}, 0, z, 0^{|\mathbf{C}|}, (t_0, <\texttt{start}>)\right)\right\}$;
当 真 **进行**
  $H' \leftarrow \emptyset$;
  对于 $\left(\mathbf{y}, y, s, z, \alpha, \mathbf{t}\right) \in H$ **进行**
    $(p, z', \alpha') \leftarrow g(y, \mathbf{C}, z, \alpha; \Theta)$;
    $H' \leftarrow H' \cup \left\{\left(\mathbf{y} * (e_j), e_j, s - \log p_j, z', \alpha', \mathbf{t}\right) \mid j \in \{0, \ldots, S\}\right\}$;
  $H' \leftarrow H'$ 中按第三分量排首 k 个元素组成的集合;
  $H \leftarrow \emptyset$;
  对于 $T = \left(\mathbf{y}, y, s, z, \alpha, \mathbf{t}\right) \in H'$ **进行**
    当 真 **进行**
      $X \leftarrow \mathbf{t}$ 的栈顶;
      如果 X 为终结符 **则**
        如果 $X = y$ **则** 从 $\mathbf{t}$ 弹出栈顶 **否则** $\mathbf{t} \leftarrow (<\texttt{error}>)$;
        跳出;
      否则
        如果 $T(X, y)$ 为某产生式 “$X \leftarrow X_1 \cdots X_\ell$” **则**
          从 $\mathbf{t}$ 弹出栈顶;
          依次把 $X_\ell, \cdots, X_1$ 压入栈 $\mathbf{t}$;
        否则
          $\mathbf{t} \leftarrow (<\texttt{error}>)$;
          跳出;
    如果 $y = e_0$ **则**
      如果 $\mathbf{t}$ 为空 **则** $\mathbf{y}$ 对应的序列符合语法 G
        $Y \leftarrow Y \cup \{(\mathbf{y}, s)\}$;
        $k \leftarrow k - 1$;
    否则
      如果 $\mathbf{t} \neq (<\texttt{error}>)$ **则** $\mathbf{y}$ 对应的序列可能是符合语法 G 的序列的前缀
        $H \leftarrow H \cup \left\{\left(\mathbf{y}, y, s, z, \alpha, \mathbf{t}\right)\right\}$;
  如果 $k \leq 0$ **则**
    跳出;

加而迅速递减。语言模型则利用自然语言中的冗余性来纠错，从而使得长文本的识别更为可靠。例如识别系统光从形状上看很难区分“周曰”和“周日”，但是它会根据后者在语料库中出现频率更高而更倾向于后者。数学公式虽然更紧凑，但仍然存在冗余，这种冗余有助于区分一些仅从形状难以区分的符号和空间位置关系。

语言模型可以被视为前述框架在编码器退化时的情况。在经典的自然语言处理中，常常用前面 n 个词来预测下一个词，这种简单的统计语言模型不太适合存在远距离依赖的情况（如存在多层嵌套的括号时）。基于递归神经网络的模型则可以利用所有过往历史，部分实验中将使用小型的单层 LSTM 语言模型。

在主要实验中，同质模型会使用相同的权重，而 TAP 模型与语言模型的相对权重则通过在检验集上进行实验来确定。

5.4 本章小结

本章介绍了主要实验中需要用到的联机手写数学公式识别系统及其用法。对于现成的联机识别系统如 MyScript，可以直接应用。对于可供开发者重新训练的联机识别系统如 TAP，用提取出的笔划进行训练有望进一步提升用于脱机识别时的准确性。为了确保基于笔划提取的脱机识别系统不依赖于专门为此设计的联机识别系统，我们仅使用现有且具代表性的联机识别系统。因为避免分别开发联机和脱机系统所需的重复劳动正是笔划提取方案的其中一个主要的优点，我们坚决抵制了设计新型联机识别系统的冲动。

第六章　实验结果

6.1　概述

为了检验基于笔划提取的脱机手写数学公式识别系统能否保留联机手写数学公式识别系统高准确性和低资源占用的优点，我们在权威的公开数据集上用公认的评价指标进行了一系列的实验。首先，我们原封不动地用了一个已经产品化的联机手写数学公式识别系统来识别提取出的笔划，进而确定本法的准确性是否与其它脱机手写数学公式识别方法可比。其次，我们用提取出的笔划重新训练了一个联机手写数学公式识别系统，从而检查该系统能否适应提取出的笔划，以及一个模型能否同时适用于书写笔划和提取出的笔划。另外，我们在类型广泛的设备上分别进行了性能测试，由此检验本法是否具有识别速度快和低内存占用的特性。最后，我们通过消融研究检验了笔划提取算法中各步骤的作用。

6.2　数据集和评价方法

6.2.1　数据集

我们在两个公开数据集上测试过本文提出的系统。

CROHME 系列数据集 [75] 是手写数学公式识别领域最权威的数据集。竞赛组织者通过要求一些志愿者在笔式平板电脑、指式平板电脑或电子白板上书写指定数学公式来收集对应的笔迹，然后用半自动化的方法标注笔划与符号间的对应关系。表6-1总结了这系列数据集的规模。虽然这些数据集原来被用于评测联机识别系统，但由于可以通过渲染笔迹得到数学公式图像，这数据集近年也常常被用于评价脱机识别系统。在 CROHME 2019 的任务 2（脱机手写公式识别）[63]，竞赛组织者官方提供了一个把手写数学公式渲染为 1010×1010 像素图片（其中每边有 5 像素填充）的脚本，我们的实验也会沿用这个约定。

OffRaSHME 数据集 [117] 是一个专门用来评测脱机手写数学公式识别系统的数据集。竞赛组织者要求志愿者在纸上书写指定的数学公式然后通过扫描得到图片。训练集有 19749 条公式的图片和对应的符号标签图，其中 9749 条数学公式同时提供了对其中各符号位置的标注，测试集则有 2000 条公式的图片。

两个数据集的难度不能直接比较。一方面，CROHME 数据集中的图片笔划分明且接近等宽（如图6-1所示），OffRaSHME 数据集中的图片则往往带有更多

表 6-1 CROHME 数据集中的公式数目

年份	训练集	验证集	测试集
2011	921	-	348
2012	1336	-	488
2013	8836	-	671
2014	8836	-	986
2016	8836	986	1147
2019	9993	986	1199

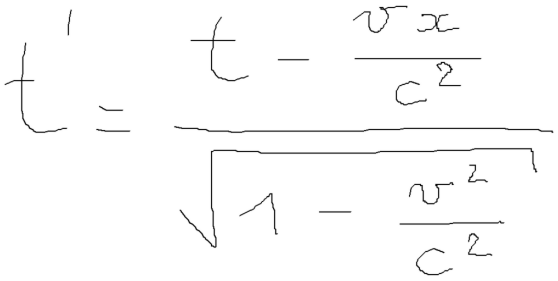
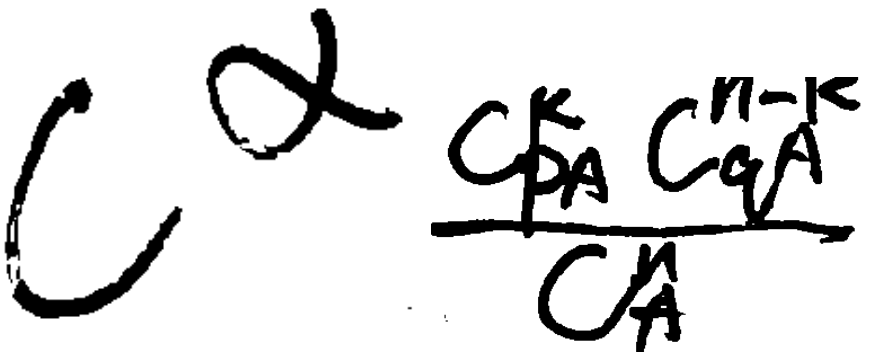


图 6-1 CROHME 数据集中数学公式图片的样本

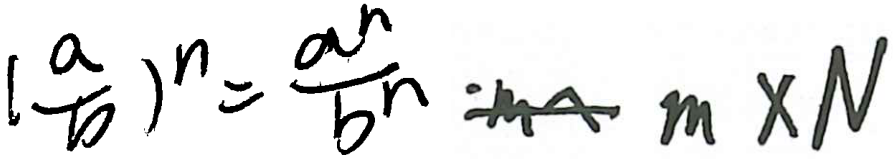
噪声，这使正确提取笔划变得困难。例如 OffRaSHME 数据集常有如图6-2a所示的断裂笔划、如图6-2b所示的粘连笔划、如图6-2c所示的椒盐噪声和图6-2d所示的删改。另一方面，CROHME 数据集的训练集较小，同时其中的公式总体而言更复杂一点，这增添了 CROHME 竞赛的难度。例如 OffRaSHME 测试集中公式的规范化 LaTeX 表示平均有约 13.4 个单词，而 CROHME 2016 测试集中公式的规范化 LaTeX 表示平均有约 17.7 个单词。

6.2.2 评价方法

为了与其它手写数学公式识别系统作出有意义的比较，不但要使用相同的测试集，而且有必要使用相同的评价指标。目前脱机手写数学公式识别系统评测用的主流指标是基于符号标签图计算的表达式级度量，它们同时被 CROHME 2019 和 OffRaSHME 2020 两个主要竞赛官方采用。为了叙述的便利，以下假设数据集中的公式只有行、上下标、分式和根式这些结构元素，从而符合表6-2所



(a) 一张带断裂笔划的图片 **(b)** 一张带粘连笔划的图片



(c) 一张带椒盐噪声的图片 **(d)** 一张带删除线的图片

图 6-2 OffRaSHME 数据集中数学公式图片的样本

示的语法。这个假设对于 CROHME 数据集和 OffRaSHME 数据集中的公式都是成立的，但以下内容也容易推广到涉及其它结构元素的情况。

首先说明如何把一条数学公式表示为符号标签图。符号标签图中的每个顶点对应于数学公式中的一个符号（包括分数线和根号），顶点被赋予相应符号类别作为值（标签），顶点按相应符号在公式中的绝对位置标识。符号标签图中的每条有向边对应于数学公式中的一对有特殊位置关系的符号，有向边被赋予相应符号间的位置关系类别作为值（标签）。数学公式中符号间的位置关系对应于符号标签图中的一条有向边，边被赋予相应的位置关系作为值（标签）。算法6.1形式地给出了符号标签图准确的构造方法：若

$$G(\mathcal{E},\mathtt{0})=(V_{\mathcal{E}},E_{\mathcal{E}},L_{\mathcal{E}}), \tag{6-1}$$

则属性图

$$(V,E,v_V,v_E)=(\{v|(v,l)\in V_{\mathcal{E}}\},\{e|(e,l)\in E_{\mathcal{E}}\},V_{\mathcal{E}},E_{\mathcal{E}}) \tag{6-2}$$

称为数学公式 $\mathcal{E}$ 的符号标签图，其中 V 和 E 分别为顶点集和边集，而 v_V 和 v_E 分别为顶点和边的赋值函数。

例 6.2.1 图6-3给出了二次方程求根公式 "$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$" 的符号标签图。每个方框表示一个顶点，顶点的值被写在方框内；每个箭号表示一条有向边，边的值被写在箭号旁。在图中每个顶点都有一个惟一的名字，由从根到它的路径决定，例如符号 "c" 对应顶点的名字为 "ORRAboveRRRInsideRRRR"。

接着说明如何定义两条数学公式之间的距离。给定一条数学公式，假设标注公式和识别结果的符号标签图分别为 $(V_1,E_1,v_{V,1},v_{E,1})$ 和 $(V_2,E_2,v_{V,2},v_{E,2})$，则

表 6-2 涵盖 CROHME 数据集中公式的一种图语法

$$
\begin{aligned}
< \texttt{start} > &::= < \texttt{mrow} > \\
< \texttt{msub} > &::= < \texttt{mrow} >_{<\texttt{mrow}>} \\
< \texttt{msup} > &::= < \texttt{mrow} >^{<\texttt{mrow}>} \\
< \texttt{msubsup} > &::= < \texttt{mrow} >_{<\texttt{mrow}>}^{<\texttt{mrow}>} \\
< \texttt{munder} > &::= \sum_{<\texttt{mrow}>}^{<\texttt{mrow}>} \mid \int_{<\texttt{mrow}>}^{<\texttt{mrow}>} \mid \lim_{<\texttt{mrow}>} \\
< \texttt{munderover} > &::= \sum_{<\texttt{mrow}>} \mid \int_{<\texttt{mrow}>} \\
< \texttt{mfrac} > &::= \frac{<\texttt{mrow}>}{<\texttt{mrow}>} \\
< \texttt{msqrt} > &::= \sqrt{< \texttt{mrow} >} \\
< \texttt{mroot} > &::= \sqrt[<\texttt{mrow}>]{< \texttt{mrow} >} \\
< \texttt{mrow} > &::= < \texttt{msub} > \mid < \texttt{msup} > \mid < \texttt{msubsup} > \\
&\quad \mid < \texttt{munder} > \mid < \texttt{munderover} > \mid < \texttt{mfrac} > \\
&\quad \mid < \texttt{msqrt} > \mid < \texttt{mroot} > \mid < \texttt{symbol} > \\
< \texttt{symbol} > &::= \text{`0'} \mid \text{`1'} \mid \text{`2'} \mid \text{`3'} \mid \text{`4'} \mid \text{`5'} \mid \text{`6'} \mid \text{`7'} \mid \text{`8'} \mid \text{`9'} \\
&\quad \mid \text{`}A\text{'} \mid \text{`}B\text{'} \mid \text{`}C\text{'} \mid \text{`}E\text{'} \mid \text{`}F\text{'} \mid \text{`}G\text{'} \mid \text{`}H\text{'} \mid \text{`}I\text{'} \mid \text{`}L\text{'} \\
&\quad \mid \text{`}M\text{'} \mid \text{`}N\text{'} \mid \text{`}P\text{'} \mid \text{`}R\text{'} \mid \text{`}S\text{'} \mid \text{`}T\text{'} \mid \text{`}V\text{'} \mid \text{`}X\text{'} \mid \text{`}Y\text{'} \\
&\quad \mid \text{`}a\text{'} \mid \text{`}b\text{'} \mid \text{`}c\text{'} \mid \text{`}d\text{'} \mid \text{`}e\text{'} \mid \text{`}f\text{'} \mid \text{`}g\text{'} \mid \text{`}h\text{'} \mid \text{`}i\text{'} \mid \text{`}j\text{'} \mid \text{`}k\text{'} \\
&\quad \mid \text{`}l\text{'} \mid \text{`}m\text{'} \mid \text{`}n\text{'} \mid \text{`}o\text{'} \mid \text{`}p\text{'} \mid \text{`}q\text{'} \mid \text{`}r\text{'} \mid \text{`}s\text{'} \mid \text{`}t\text{'} \mid \text{`}u\text{'} \\
&\quad \mid \text{`}v\text{'} \mid \text{`}w\text{'} \mid \text{`}x\text{'} \mid \text{`}y\text{'} \mid \text{`}z\text{'} \mid \text{`}\Delta\text{'} \mid \text{`}\alpha\text{'} \mid \text{`}\beta\text{'} \mid \text{`}\gamma\text{'} \\
&\quad \mid \text{`}\theta\text{'} \mid \text{`}\lambda\text{'} \mid \text{`}\pi\text{'} \mid \text{`}\sigma\text{'} \mid \text{`}\varphi\text{'} \mid \text{`}\phi\text{'} \mid \text{`}\mu\text{'} \mid \text{`}<\text{'} \mid \text{`}>\text{'} \mid \text{`!'} \\
&\quad \mid \text{`}/\text{'} \mid \text{`}\infty\text{'} \mid \text{`sin'} \mid \text{`cos'} \mid \text{`tan'} \mid \text{`lim'} \mid \text{`}\textstyle\sum\text{'} \mid \text{`}\textstyle\int\text{'} \mid \text{`log'} \\
&\quad \mid \text{`}+\text{'} \mid \text{`}-\text{'} \mid \text{`}\pm\text{'} \mid \text{`}\times\text{'} \mid \text{`}\div\text{'} \mid \text{`}\cdot\text{'} \mid \text{`}=\text{'} \mid \text{`}\prime\text{'} \mid \text{`,'} \mid \text{`.'} \\
&\quad \mid \text{`}\ldots\text{'} \mid \text{`}\rightarrow\text{'} \mid \text{`}\exists\text{'} \mid \text{`}\in\text{'} \mid \text{`}\neq\text{'} \mid \text{`}\leq\text{'} \mid \text{`}\geq\text{'} \mid \text{`}\forall\text{'} \\
&\quad \mid \text{`(}\text{'} \mid \text{`[}\text{'} \mid \text{`\{}\text{'} \mid \text{`|}\text{'} \mid \text{`\}}\text{'} \mid \text{`]}\text{'} \mid \text{`)}\text{'}
\end{aligned}
$$

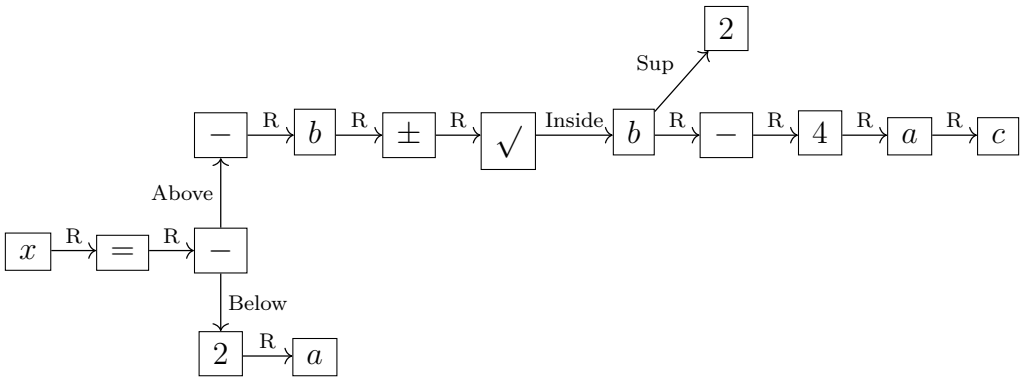


图 6-3 数学公式的符号标签图的例子

算法 6.1: 符号标签图构造函数 G 的实现

输入: 数学公式 $\mathcal{E}$, 基线上最左顶点 P

输出: 顶点赋值函数 $V_\mathcal{E}$, 边赋值函数 $E_\mathcal{E}$, 基线上最右顶点 $L_\mathcal{E}$

如果 $\mathcal{E}=\mathcal{X}^\mathcal{Y}$ **则**
  $(V_\mathcal{X},E_\mathcal{X},L_\mathcal{X})\leftarrow G(\mathcal{X},P)$, $(V_\mathcal{Y},E_\mathcal{Y},L_\mathcal{Y})\leftarrow G(\mathcal{Y},L_\mathcal{X}\texttt{Sup})$;
  $V_\mathcal{E}\leftarrow V_\mathcal{X}\cup V_\mathcal{Y}$, $L_\mathcal{E}\leftarrow L_\mathcal{X}$;
  $E_\mathcal{E}\leftarrow E_\mathcal{X}\cup E_\mathcal{Y}\cup\{((L_\mathcal{X},L_\mathcal{X}\texttt{Sup}),\texttt{Sup})\}$;
否则如果 $\mathcal{E}=\mathcal{X}_\mathcal{Y}$ **则**
  $(V_\mathcal{X},E_\mathcal{X},L_\mathcal{X})\leftarrow G(\mathcal{X},P)$, $(V_\mathcal{Y},E_\mathcal{Y},L_\mathcal{Y})\leftarrow G(\mathcal{Y},L_\mathcal{X}\texttt{Sub})$;
  $V_\mathcal{E}\leftarrow V_\mathcal{X}\cup V_\mathcal{Y}$, $L_\mathcal{E}\leftarrow L_\mathcal{X}$;
  $E_\mathcal{E}\leftarrow E_\mathcal{X}\cup E_\mathcal{Y}\cup\{((L_\mathcal{X},L_\mathcal{X}\texttt{Sub}),\texttt{Sub})\}$;
否则如果 $\mathcal{E}=\mathcal{X}_\mathcal{Y}^\mathcal{Z}$ **则**
  $(V_\mathcal{X},E_\mathcal{X},L_\mathcal{X})\leftarrow G(\mathcal{X},P)$, $(V_\mathcal{Y},E_\mathcal{Y},L_\mathcal{Y})\leftarrow G(\mathcal{Y},L_\mathcal{X}\texttt{Sub})$, $(V_\mathcal{Z},E_\mathcal{Z},L_\mathcal{Z})\leftarrow G(\mathcal{Z},L_\mathcal{X}\texttt{Sup})$;
  $V_\mathcal{E}\leftarrow V_\mathcal{X}\cup V_\mathcal{Y}\cup V_\mathcal{Z}$, $L_\mathcal{E}\leftarrow L_\mathcal{X}$;
  $E_\mathcal{E}\leftarrow E_\mathcal{X}\cup E_\mathcal{Y}\cup E_\mathcal{Z}\cup\{((L_\mathcal{X},L_\mathcal{X}\texttt{Sub}),\texttt{Sub}),((L_\mathcal{X},L_\mathcal{X}\texttt{Sup}),\texttt{Sup})\}$;
否则如果 $\mathcal{E}=\frac{\mathcal{X}}{\mathcal{Y}}$ **则**
  $(V_\mathcal{X},E_\mathcal{X},L_\mathcal{X})\leftarrow G(\mathcal{X},P)$, $(V_\mathcal{Y},E_\mathcal{Y},L_\mathcal{Y})\leftarrow G(\mathcal{Y},L_\mathcal{X}\texttt{Below})$;
  $V_\mathcal{E}\leftarrow V_\mathcal{X}\cup V_\mathcal{Y}$, $L_\mathcal{E}\leftarrow L_\mathcal{X}$;
  $E_\mathcal{E}\leftarrow E_\mathcal{X}\cup E_\mathcal{Y}\cup\{((L_\mathcal{X},L_\mathcal{X}\texttt{Below}),\texttt{Below})\}$;
否则如果 $\mathcal{E}=\frac{\mathcal{Z}}{\mathcal{Y}}^\mathcal{X}$ **则**
  $(V_\mathcal{X},E_\mathcal{X},L_\mathcal{X})\leftarrow G(\mathcal{X},P)$, $(V_\mathcal{Y},E_\mathcal{Y},L_\mathcal{Y})\leftarrow G(\mathcal{Y},L_\mathcal{X}\texttt{Below})$,
  $(V_\mathcal{Z},E_\mathcal{Z},L_\mathcal{Z})\leftarrow G(\mathcal{Z},L_\mathcal{X}\texttt{Above})$;
  $V_\mathcal{E}\leftarrow V_\mathcal{X}\cup V_\mathcal{Y}\cup V_\mathcal{Z}$, $L_\mathcal{E}\leftarrow L_\mathcal{X}$;
  $E_\mathcal{E}\leftarrow E_\mathcal{X}\cup E_\mathcal{Y}\cup E_\mathcal{Z}\cup\{((L_\mathcal{X},L_\mathcal{X}\texttt{Below}),\texttt{Below}),((L_\mathcal{X},L_\mathcal{X}\texttt{Above}),\texttt{Above})\}$;
否则如果 $\mathcal{E}=\frac{\mathcal{X}}{\mathcal{Y}}$ **则**
  $(V_\mathcal{X},E_\mathcal{X},L_\mathcal{X})\leftarrow G(\mathcal{X},P\texttt{Above})$, $(V_\mathcal{Y},E_\mathcal{Y},L_\mathcal{Y})\leftarrow G(\mathcal{Y},P\texttt{Below})$;
  $V_\mathcal{E}\leftarrow V_\mathcal{X}\cup V_\mathcal{Y}\cup\{(P,-)\}$, $L_\mathcal{E}\leftarrow P$;
  $E_\mathcal{E}\leftarrow E_\mathcal{X}\cup E_\mathcal{Y}\cup\{((P,P\texttt{Above}),\texttt{Above}),((P,P\texttt{Below}),\texttt{Below})\}$;
否则如果 $\mathcal{E}=\sqrt{\mathcal{X}}$ **则**
  $(V_\mathcal{X},E_\mathcal{X},L_\mathcal{X})\leftarrow G(\mathcal{X},P\texttt{Inside})$;
  $V_\mathcal{E}\leftarrow V_\mathcal{X}\cup\{(P,\surd)\}$, $L_\mathcal{E}\leftarrow P$;
  $E_\mathcal{E}\leftarrow E_\mathcal{X}\cup\{((P,P\texttt{Inside}),\texttt{Inside})\}$;
否则如果 $\mathcal{E}=\sqrt[\mathcal{Y}]{\mathcal{X}}$ **则**
  $(V_\mathcal{X},E_\mathcal{X},L_\mathcal{X})\leftarrow G(\mathcal{X},P\texttt{Inside})$, $(V_\mathcal{Y},E_\mathcal{Y},L_\mathcal{Y})\leftarrow G(\mathcal{Y},P\texttt{Above})$;
  $V_\mathcal{E}\leftarrow V_\mathcal{X}\cup V_\mathcal{Y}\cup\{(P,\surd)\}$, $L_\mathcal{E}\leftarrow P$;
  $E_\mathcal{E}\leftarrow E_\mathcal{X}\cup E_\mathcal{Y}\cup\{((P,P\texttt{Inside}),\texttt{Inside}),(P,P\texttt{Above},\texttt{Above})\}$;
否则如果 $\mathcal{E}=\mathcal{X}\mathcal{Y}$ **则**
  $(V_\mathcal{X},E_\mathcal{X},L_\mathcal{X})\leftarrow G(\mathcal{X},P)$, $(V_\mathcal{Y},E_\mathcal{Y},L_\mathcal{Y})\leftarrow G(\mathcal{Y},L_\mathcal{X}\texttt{R})$;
  $V_\mathcal{E}\leftarrow V_\mathcal{X}\cup V_\mathcal{Y}$, $L_\mathcal{E}\leftarrow L_\mathcal{Y}$;
  $E_\mathcal{E}\leftarrow E_\mathcal{X}\cup E_\mathcal{Y}\cup\{((L_\mathcal{X},L_\mathcal{X}\texttt{R}),\texttt{R})\}$;
否则
  $V_\mathcal{E}\leftarrow\{(P,\mathcal{E})\}$, $L_\mathcal{E}\leftarrow P$;
  $E_\mathcal{E}\leftarrow\emptyset$;

表 6-3 评价指标的例子

序号	标注公式	识别结果	Δ_V	Δ_E	Δ_G
1	$\frac{1}{4}=-\frac{3}{4}+1$	$\frac{1}{4}=-\frac{3}{4}+1$	0	0	0
2	$\frac{9}{5}$	$\frac{7}{5}$	1	0	1
3	$x_1x_{d+1}+\ldots+x_dx_{2d}$	$X_1X_{d+1}+\ldots+X_jX_{21}$	6	0	6
4	$y=2x-\frac{c_i+c_j}{2}$	$y=2x-\frac{c_i+c}{2}$	1	1	2
5	$\sqrt{1+rm_a}$	$\sqrt{1+rma}$	2	2	4
6	$E\times\ldots\times E$	$EX..XE$	5	1	6
7	$\|\|x\|\|=\|a\|+\|b\|$	$\|x\|f=\|a\|+\|b\|$	12	1	13
8	$(2ba)b^n=2nb^n+2b^{n+1}a$	$(2ba)b^n=2nb^n+2b^{n+1}$	1	1	2
9	$y^iy^j=y^{i+j}$	$y^{iyj}=y^{i+j}$	12	10	22

符号的错误数由

$$\Delta_V=\#(V_1\setminus V_2)+\#(V_2\setminus V_1)+\#\{v\in V_1\cap V_2|v_{V,1}(v)\neq v_{V,2}(v)\} \tag{6-3}$$

给出，而位置关系的错误数由

$$\Delta_E=\#(E_1\setminus E_2)+\#(E_2\setminus E_1)+\#\{e\in E_1\cap E_2|v_{E,1}(e)\neq v_{E,2}(e)\} \tag{6-4}$$

给出，于是识别结果中总错误数为

$$\Delta_G=\Delta_V+\Delta_E。 \tag{6-5}$$

当且仅当

$$\Delta_E=0 \tag{6-6}$$

时称识别结果结构正确。

表6-3列出了一些数学公式识别结果的错误数，由此可以看出上述指标有如下特性：

- 总错误数给出了两个数学公式间的一种距离。也就是说，当且仅当总错误数为零时两条数学公式完全相同，同时对称性和三角不等式都是成立的。
- 当且仅当可以只通过修改符号识别结果把识别结果改正时识别结果结构正确，这时符号错误数就是所必须修正的符号数目。
- 关系错误数非零时符号错误数也一定非零。
- 在一个左右结构的子公式中，出现在左侧的插入或删除错误一般会比出现在右侧的同类错误更影响总错误数。

结构准确率测量忽略所有符号的标签时匹配的公式比例，表达式准确率测量容许至多若干个标签错误（符号或位置关系）时的准确率。形式地说，设测试

集中有 N 条手写数学公式，又对应识别结果的位置关系错误数和总错误数分别为 $\Delta_{E,i}$ 和 $\Delta_{G,i}$，其中 $i = 1, \ldots, N$，则

$$\text{结构准确率} = \frac{\#\{i|\Delta_{E,i} = 0\}}{N} \tag{6-7}$$

而

$$\text{表达式准确率}_k = \frac{\#\{i|\Delta_{G,i} \leq k\}}{N}。 \tag{6-8}$$

早年的 CROHME 系列竞赛也采用过一些笔划级别的指标，例如笔划分类率（被指定到正确符号类别的笔划比例）、符号切分率（被正确切分出来的符号的比例）和符号识别率（被正确切分出来的符号中被正确识别的比例），它们可用以分别评价系统在符号切分、符号识别和结构分析方面的表现。不过，脱机识别系统并不会生成笔划与符号之间的对应关系，部分端到端联机手写识别系统也不再有显式的切分步骤，因此这类指标已经淡出。

6.3 基于现成联机识别系统的情况

在本实验中，本文提出的笔划提取器[1]与顶尖的数学公式识别器 MyScript 结合成一个脱机识别器。

表6-4记录了在 CROHME 2014 测试集上的实验结果。表中前七个系统是参与了 CROHME 2014 [73] 的联机识别系统。除了 MyScript 外，其它参赛者都仅用了官方的训练数据。本文系统的准确率甚至比 MyScript 以外的参赛者都更高。由于 MyScript 本身在过去几年也在发展中，我们也在联机场景自行测试了写作时最新的 MyScript Interactive Ink 版本 1.3。另一方面，im2markup [24]、Watch, Attend, and Parse(WAP) [137]、Paired Adversarial Learning(PAL) [123]、CNN-BLSTM-LSTM [53] 和 Multi-Scale Attention with Dense Encoder(MSD) [133] 都是脱机识别系统，本文系统的准确率比它们报告的都要高。注意由于 MyScript 的使用，本文系统间接地用了额外的训练数据，im2markup 和 PAL 则使用了一些印刷体公式作为训练数据，其它脱机系统只基于官方的训练数据（但可能作了数据增强）。

表6-5记录了在 CROHME 2016 测试集上的实验结果。前五个系统是参与了 CROHME 2016 [70] 的联机识别系统。MyScript 另外用了约 30, 000 条额外手写公式来训练，Wiris 包含了一个用维基百科公式数据集去训练的语言模型，其它参赛者只用了官方的训练数据（但可能作了数据增强）。本文系统的准确率仍然比 MyScript 以外的参赛者都更高，比 MyScript 低则是理所当然。另一方面，WAP [137]、CNN-BLSTM-LSTM [53]、MSD [133] 和 PAL [123] 是脱机识别系

[1]本文提出的笔划提取器适用于 JVM 的版本和适用于 Android 的版本可以分别从`https://github.com/chungkwong/mathocr-myscript`和`https://github.com/chungkwong/mathocr-myscript-android`下载。

表 6-4 基于 MyScript 时在 CROHME 2014 测试集上的表现

系统	表达式准确率			结构准确率
	精确	≤ 1 错误	≤ 2 错误	
联机识别系统				
São Paulo	15.01	22.31	26.57	-
RIT, CIS	18.97	26.37	30.83	-
RIT, DRPL	18.97	28.19	32.35	-
Tokyo	25.66	33.16	35.90	-
Nantes	26.06	33.87	38.54	-
Politècnica de València	37.22	44.22	47.26	-
MyScript	62.68	72.31	75.15	-
MyScript 1.3	**69.47**	**78.30**	**81.03**	**82.86**
脱机识别系统				
Harvard, im2markup	39.96	-	-	-
USTC, WAP	44.4	58.4	62.2	-
CAS, PAL	47.06	63.49	72.31	-
TDTU, CNN-BLSTM-LSTM	48.78	63.39	70.18	-
USTC, MSD	52.8	68.1	72.0	-
CAS, PAL-v2	54.87	70.69	**75.76**	-
笔划提取 +MyScript 1.3	**58.22**	**71.60**	75.15	**77.38**

表 6-5 基于 MyScript 时在 CROHME 2016 测试集上的表现

系统	表达式准确率			结构准确率
	精确	≤ 1 错误	≤ 2 错误	
联机识别系统				
Nantes	13.34	21.02	28.33	21.45
São Paolo	33.39	43.50	49.17	57.02
Tokyo	43.94	50.91	53.70	61.55
Wiris	49.61	60.42	64.69	74.28
MyScript	67.65	75.59	79.86	88.14
MyScript 1.3	**73.06**	**82.30**	**87.10**	**88.58**
脱机识别系统				
USTC, WAP	42.0	55.1	59.3	-
TDTU, CNN-BLSTM-LSTM	45.60	59.29	65.65	-
USTC, MSD	50.1	63.8	67.4	-
CAS, PAL-v2	57.89	70.44	76.29	-
笔划提取 +MyScript 1.3	**65.65**	**77.68**	**82.56**	**85.00**

统，本文系统的准确率比它们报告的都要高。虽然额外训练数据的间接使用可能是部分原因，但改为与一个较弱且仅使用官方数据集来训练的联机识别系统结合后得出的准确率仍然与它们可比，详细情况见下一节。

本文系统参与了 CROHME 2019 [63] 并取得了脱机项目的季军，表6-6列出了竞赛的最终结果。虽然本文系统的完全准确率不如 PAL，但结构准确率比 PAL 更高。注意 Samsung R&D 1、PAL、MyScript 1.3 和 MathType 都用了额外的公式笔迹或公式图片作为训练数据，USTC-iFLYTEK 则包含了在外部数据集上训练过的语言模型。

虽然本文提出的脱机识别系统的准确率与其它脱机识别系统相比不错，但仍然明显低于所基于的联机识别系统。假如笔划提取完全准确，两者应该相同。另一个观察是结构准确率上的差距比表达式准确率上的差距要小一些，这说明与符号识别相比，结构分析对笔划提取错误较不敏感。

表 6-6 基于 MyScript 时在 CROHME 2019 测试集上的表现

系统	表达式准确率			结构准确率
	精确	≤ 1 错误	≤ 2 错误	
联机识别系统				
TUAT	39.95	52.21	56.54	58.22
MathType	60.13	74.40	78.57	79.15
PAL-v2	62.55	74.98	78.40	79.15
Samsung R&D 2	65.97	77.81	81.73	82.82
MyScript 1.3	77.40	85.82	87.99	88.82
MyScript	79.15	86.82	89.82	90.66
Samsung R&D 1	79.82	87.82	89.15	89.32
USTC-iFLYTEK	**80.73**	**88.99**	**90.74**	**91.49**
脱机识别系统				
TUAT	24.10	35.53	43.12	43.70
Univ. Linz	41.49	54.13	58.88	60.02
SCST-USTC	62.14	75.06	78.23	78.32
PAL-v2	62.89	74.98	78.40	79.32
笔划提取 +MyScript 1.3	65.22	78.48	83.07	84.90
PAL	71.23	80.31	82.65	83.82
USTC-iFLYTEK	**77.15**	**86.82**	**88.99**	**89.49**

6.4 基于可训练联机识别系统的情况

在本实验中，本文提出的笔划提取器与联机手写数学公式识别系统 TAP [134] 结合成一个脱机手写数学公式识别系统。虽然引进 TAP 的论文组合了三个联机模型、三个脱机模型和三个语言模型来获得最好的结果，但在我们的实验中先不用模型组合技术，因为我们的目的是检验一个纯联机模型能否适应提取出的人工笔划。

我们在 CROHME 2016 数据集上进行了实验，实验中只分别用官方的训练集和检验集来训练和检验。当用书写的笔划训练时，笔划点和对齐由数据集直接提供。当用人工笔划训练时，笔划点来自从图像提取出来的笔划，对齐则用 Hausdorff 距离估计。在两种情况下，TEX 代码都由 MathML 代码标注转换得到。虽然我们在训练集中发现过百条公式的标注有误（通过检测 TEX 代码标注和 MathML 代码标注的不一致性发现），但我们仍然不加修正地直接使用。由于数据集较小而表达式级别的准确率是整体度量，我们在相同训练集上用三组不同的初始权重分别训练了三个模型，它们中在检验集上最准确的一个被选中在测试集上测试。

表6-7记录了在 CROHME 2016 测试集上的实验结果。和用 MyScript 时观察到的情况相同，用书写笔划训练出的模型在书写笔划上的表现明显优于在人工笔划上的表现。然而，在人工笔划上训练的模型在人工笔划上的表现几乎赶上了在书写笔划上训练的模型在书写笔划上的表现，表达式准确率的差距仅约 0.78%，而 CROHME 2019 中最佳的脱机模型与最佳的联机模型间的表达式准确率的差距约 3.58%。这个结果证实了一个联机识别系统可以通过重新训练很好地适应人工笔划。这似乎也表明联机识别与脱机识别之间本质上的难度差距比过往想像中要小。另外一个有趣的事实是基于笔划提取的脱机识别系统的结构准确率超过了所基于的联机识别系统，这可能是因为笔划排序有助于简化结构分析。

虽然基于 TAP 的脱机识别器的准确性不如基于 MyScript 的，但仍然有可以改良的空间。例如语言模型、组合模型、增强数据集、扩充数据集等技巧并

表 6-7 基于在不同模态数据上训练的单个 TAP 模型时在 CROHME 2016 数据集上的表现

训练	测试	表达式准确率			结构准确率
		精确	≤ 1 错误	≤ 2 错误	
书写	书写	**45.77**	58.15	64.34	65.65
书写	提取	24.15	40.02	49.26	54.14
提取	提取	44.99	**59.11**	**66.00**	**67.92**

表 6-8 基于经修改的 TAP 时在 CROHME 2016 数据集上的表现

脱机识别系统			表达式准确率			结构
形式语法	TAP 模型	语言模型	精确	≤ 1 错误	≤ 2 错误	准确率
×	1	0	44.99	59.11	66.00	67.92
✓	1	0	45.51	59.37	66.70	68.53
✓	3	0	49.69	64.34	70.27	72.19
✓	3	3	**50.83**	**64.69**	**71.32**	**72.89**

没有被用在表6-7所记录的实验中，但其它数学公式识别系统 [63] 往往使用它们，并且确实能明显提高准确率 [134, 53]。表6-8显示了至少部分方法也适用于本文系统。

形式语言模型可以保证结果的合法性，但对提升准确率的目的而言作用十分有限。我们注意到不带形式语法模型的 TAP 识别出不少不合理的结果。事实上，CROHME 2016 的训练集、检验集和测试集中都分别有约 99.9% 的数学公式的 LaTeX 表示符合表5-3所示的语法，但用 TAP 识别出的 LaTeX 代码只有约 94.9% 符合该语法，主要问题是括号不匹配。这表明仅直接用官方数据训练未能使 TAP 完全学会 LaTeX 的语法。不过通过用该形式语法约束解码器的输出，准确率仅因而略为提高。即使再加上一个在 NTCIR-12 数据集 [129] 上训练过的单层 LSTM 语言模型，准确率也只是再稍为提升。组合模型的效果更为明显。通过组合三个 TAP 联机模型和三个单层 LSTM 语言模型，我们在 CROHME 2016 测试集上得到了比 MSD 更高的准确率，而 MSD 组合了五个脱机模型 [133]。

本文系统参加了 ICFHR 2020 OffRaSHME 竞赛的任务 1，结果如表6-9所示。参赛版本基于 TAP 并仅使用官方数据集来训练。因为过半数公式都没有提供对各符号位置的标注，而且组织者要求参赛者声明是否使用这种标注，我们没有使用有关标注。这个版本用了形式语法约束和组合模型，但没有使用数据驱动的语言模型。即使这个数据集中的公式更为真实，本文系统仍然完全正确地识别了测试集中约 61.35% 的数学公式，甚至比在 CROHME 2016 数据集上对应系统的 49.52% 明显更高一些。虽然部分原因是测试集大了约一倍，但还是可以说明基于笔划提取的脱机手写数学公式识别技术可以适应存在噪声的图片。

我们还检验了把同一个 TAP 模型同时用于联机识别和脱机识别的可行性。为此，我们把 CROHME 2019 联机手写数学公式数据集、CROHME 2019 脱机手写数学公式数据集和 OffRaSHME 2020 数据集各自的训练集合并成一个大型的混合训练集，相应地也把它们的检验集合并成一个混合检验集。表6-10表明在混合数据集上训练过的 TAP 模型在各个数据集的测试集上都取得了优良的表现，比仅在各自的训练集和检验集上训练得到的结果都要好一些。特别地，这说明了把从图片提取出的笔划用于扩充数据集可以明显地提高联机手写数学公式

表 6-9 基于 TAP 时在 OffRaSHME 2020 数据集上的表现

系统	额外数据	表达式准确率		
		精确	≤ 1 错误	≤ 2 错误
MLE	✓	46.85	61.15	65.80
笔划提取 +TAP	×	61.35	77.30	81.55
IVTOV	✓	61.90	73.00	75.95
HCMUS-I86	×	66.95	79.65	83.60
TUAT	×	71.75	82.70	85.80
SCUT-DLVCLab	×	72.90	86.05	88.80
USTC-iFLYTEK	×	79.85	89.85	92.00
USTC-iFLYTEK	✓	**81.85**	**90.50**	**92.30**

表 6-10 基于在混合数据集上训练的单个 TAP 模型（带形式语法约束）时在各测试集上的表现

竞赛	输入模态	表达式准确率			结构准确率
		精确	≤ 1 错误	≤ 2 错误	
CROHME 2016	联机手写	54.14	67.13	72.62	73.67
CROHME 2016	脱机手写	54.23	67.74	72.10	73.67
CROHME 2019	联机手写	52.71	66.47	71.56	72.23
CROHME 2019	脱机手写	53.46	67.89	73.48	73.23

识别的准确率。在原来的情况下，联机识别系统和脱机识别系统需要分别用笔迹和图片去进行训练，这限制了它们各自能用的训练数据量。而借助于笔划提取，无论笔迹还是图片都能被用于训练联机识别系统，这就使得可用的训练样本变多了，从而有望导致更准确的模型。

6.5 脱机识别的性能

本实验在从低端手机到强大的 GPU 服务器的各类设备上检验了本文系统的性能，表6-11给出了这些设备的规格。

表6-12记录了识别 CROHME 2016 数据集中 1147 条脱机手写数学公式的用时。虽然笔划提取算法仅在 CPU 上实现且没有特意优化速度，但它仍然在所有设备上明显比联机识别用时短。一台现代的低端手机（手机 2）平均用约 1 秒识别一张图片，一台明显过时的手机（手机 1）则需要约 2 秒。如果用户满意于目前本地联机识别的速度，他们也应该可以接受本地脱机手写数学公式识别的这

表 6-11 性能测试中所用设备的规格

设备	组件	规格
手机 1	CPU	Cortex-A7 4 cores @ 1.3 GHz
	RAM	512MB
手机 2	CPU	Cortex-A53 2 cores @ 2.0 GHz + 6 cores @ 1.45 GHz
	RAM	3GB
超级本	CPU	Intel® Core™ m3-6Y30 4 cores @ 0.90GHz
	RAM	4GB
台式机	CPU	Intel® Core™ i5-7500 4 cores @ 3.40GHz
	RAM	8GB
服务器	CPU	Intel® Xeon® Gold 6271C 2 cores @ 2.60GHz
	RAM	32GB
	GPU	NVIDIA® Tesla® V100 16GB

个速度。同时，能够在内存只有 512MB 的手机 1 上顺利完成测试说明本文系统的内存占用较低。

原生脱机手写识别系统 WAP[2]比我们提出的过程慢。由于它与其它现代的光学字符识别系统一样用计算密集 [143] 的卷积神经网络提取特征，它的速度高度依赖于高性能 GPU 的可用性。由于联机识别通常比原生脱机识别系统占用较少资源，本地的联机手写识别功能早已被整合到笔记软件和输入法中，光学字符识别功能则往往依赖于云端服务。因此，笔划提取提供了一种通过避免网络延迟、可用性、机密性和隐私等问题 [124] 来改善用户体验的方案。

对于需要同时支持两类手写数学公式识别的软件，采用基于笔划提取的方案能够减低软件的安装包大小和存储空间占用。这种需求是客观存在的，例如一个学生在上课时可能希望先通过拍照记下板书的内容，再通过用笔或手指在触摸设备上书写来添加注记，因而需要在脱机识别与联机识别间来回切换。虽然这可以通过同时包含一套联机识别系统和一套脱机识别系统来实现，但维护两套不同系统会造成额外的开发成本，部署两组模型还会使软件占用更多的存储空间。作为对比，采用笔划提取方案时，只需一组模型即可同时应对两类识别问题，从而可以显著地降低安装包的大小。

[2]代码见`https://github.com/JianshuZhang/WAP`。

表 6-12 各个脱机识别系统识别 CROHME 2016 测试集中手写数学公式的用时

设备	识别器	运行时间（秒）			
		笔划提取	联机识别	总计	平均
手机 1	MyScript	1025	1653	2678	2.33
手机 2	MyScript	254	1123	1377	1.20
超级本	TAP	22	6908	6930	6.04
台式机	TAP	**10**	535	544	0.47
	WAP	-	-	7845	6.84
服务器	TAP	34	**152**	**186**	**0.16**
	WAP	-	-	417	0.36

6.6 对笔划提取算法的分析

本实验旨在分析本文的笔划提取方法。为了评测其中各个组件的必要性，我们分别从系统中移除它们后重新测试，然后比较移除前后的准确率变化。移除后准确率下降得越多，表示被移除组件的重要性越大。

表6-13记录了各残缺系统在 CROHME 2016 测试集上的测试结果，其中 MyScript 被用于识别提取出的笔划。虽然笔划方向检测的规则相当简单，但它大幅度地提高了准确率。这主要是因为递归神经网络对输入的顺序敏感，故笔划的方向会影响字符识别的结果。注意到 MyScript 的符号识别器同时使用了静态和动态特征，所以在笔划方向错误的情况下有时仍然能正确识别符号。重笔修正亦对准确性有较大的作用，因为断开的笔划容易导致过切分，使字符切分更不可靠。另外，噪声去除、笔顺规范化和定制语法也对最终准确率有不可忽略的贡献。

本实验可能对日后笔划提取方法的设计有启发性。虽然本文在多个步骤中

表 6-13 基于 MyScript 时在 CROHME 2016 测试集上的消融实验结果

脱机识别系统	表达式准确率			结构准确率
	精确	≤ 1 错误	≤ 2 错误	
完整系统除去笔划方向检测	46.56	63.64	70.53	74.63
完整系统除去重笔修正	53.97	66.87	71.93	73.76
完整系统除去噪声去除	58.94	72.36	79.51	80.91
完整系统除去笔顺规范化	59.98	73.50	78.38	81.34
完整系统除去定制语法	61.99	74.72	79.69	82.65
完整系统	**65.65**	**77.68**	**82.56**	**85.00**

用了生硬的规则且它们大概不是最优的，但实验结果表明了它们都有助于提升准确率，也就是说它们都处理了正确的问题。因此，其它用于脱机手写识别的笔划提取算法的开发也应该考虑这些问题，特别是在不能重新训练底层的联机手写识别系统时。

6.7 本章小结

基于笔划提取的脱机手写数学公式识别系统不仅可以取得不错的准确率，而且往往比基于卷积神经网络的脱机识别系统更轻量。通过用提取出的笔划重新训练联机手写数学公式识别系统，有望得到一个与之几乎同样准确的脱机手写数学公式识别系统；通过用混合数据集重新训练联机手写数学公式识别系统，可以实现联机识别系统与脱机识别系统间的模型共用。此外，我们通过消融研究验证了本文提出的笔划提取算法中各主要步骤都是有正面作用的。

第七章　应用

7.1　概述

虽然手写数学公式识别的准确率正在持续提高，但和许多其它模式识别问题一样，完全准确依然是遥不可及的目标。于是在应用中，要么在完成识别后要求用户人手修正并确认，要么确保利用识别结果的程序能够容忍错误的存在。针对第一个方向，我们设计了一个高可用性的交互式数学公式编辑器，它能与手写识别紧密地配合，让新手和老手都能高效地输入数学公式。针对第二个方向，我们说明了对于从已知公式集合中选取数学公式的应用如数学公式检索，少量错误对结果的影响可能是可以容忍的，而且存在可有效地压缩这种影响的方法。本章内容同时适用于联机手写识别和脱机手写识别。

7.2　数学公式的录入与编辑

7.2.1　现有的公式编辑器

虽然手写是输入数学公式最自然的方法，但它并不适合作为提供给用户的惟一输入方法。即使目前最好的手写数学公式识别系统也只有约 80% 的准确率，也就是说，只要用户随意手写几条公式，就很可能会碰到出现识别错误的情况。如果一个应用只容许用户通过手写输入数学公式，则用户在出现识别错误时要么重写要么放弃。然而，数学公式识别系统对于最终用户而言很大程度上是个黑匣子，他们并不知道应该怎么写才能增加被正确识别的机会，所以重写后很可能也是误识并令用户感到绝望。技术的提升或许可以提高识别的准确性，但一些人眼都不一定能区分的手写公式的存在性决定了不能期望达到完全准确的地步。因此，为了避免用户流失，不应只依赖于手写识别，还需要提供一种机制以协助用户手动编辑识别结果。同时，编辑过程应该可以轻易而快速地完成。如果手写再编辑比从头开始用传统方法输入更费时失事，手写识别的用途就不大了。

虽然目前存在少数几个用于修正手写识别结果的界面设计，但它们都有不理想之处。对于模块化的手写数学公式识别系统，Smithies 等人 [104] 分别设计了用于修正源于不同模块的错误的模式，例如在用于修改符号切分结果的模式中用户可以通过用笔把属于同一符号的笔划连起来以修改笔划的分组。然而，这种设计把识别器的内部设计暴露了给用户并要求用户掌握，模式切换也容易使用户

感到困惑。对于采用编码器-解码器框架的手写数学公式识别系统，Zhelezniakov 等人 [142] 设计了一些用于插入、删除和替代子公式的手势，例如用户可以通过划掉子公式来删除它。然而，手势具有隐蔽性，而且要在提高表达能力和控制歧义间取一个平衡点。与此同时，大多数其它现有的数学公式编辑器也同样存在严重的可用性问题，其中以下问题尤其突出：

- 标记语言、所见即所得和手写三种输入方式的割裂。三种主要的数学公式输入方式各有利弊，它们间具有互补性。标记语言容许通过纯键盘操作快速地盲打，但学习曲线较陡峭且出错时难以排错，因而相对适合专业用户。所见即所得编辑器只需少量探索即可学会，但效率较低，因而相对适合新手用户。手写不需专门学习，但会不时出现难以补救的错误。许多公式编辑器仅提供一种输入方式，故不能同时满足不同用户的需要。部分公式编辑器虽然提供了多于一种输入方式，但把它们实现为不同的模式，模式之间没有足够的联系。例如一些公式编辑器为 LaTeX 代码提供了预览，但代码的光标位置没有被正确地显示出来，而且在输入过程中常常由于未完成的代码不合法（例如输入 "`\sqrt{n}`" 的过程中可能要经过不合法的 "`\sqrt{n`"）而使预览失效。又如有的公式编辑器同时支持所见即所得和手写，但不能通过手写作局部的修改，而整体重新识别则可能破坏原来已经正确识别或修正好的部分。
- 对文本编辑器的模仿不到位。由于数学公式编辑器与常规文本编辑器在用途和界面上都有一定的相似性，于是用户也会期望两者有一些共同的功能。然而，不少公式编辑器缺少这个或那个功能，这种不一致性为用户带来了困扰。例如有的所见即所得编辑器不能通过拖放选取子公式、不能撤销和重做、不能复制和粘贴或者不能使用快捷键导航。又如有的基于标记语言的公式编辑器没有提供命令的自动补全。

7.2.2 基本的公式编辑器

基本设计

为了克服现有公式编辑器的可用性问题，我们设计了一个公式编辑器，并分别在 JVM 和 Web 平台[1]上实现了它。其设计无缝地结合了标记语言和所见即所得两种输入方式，用户既可以通过用键盘直接输入 LaTeX 代码来得到相应的子公式，也可以通过从工具箱中寻找所需结构或符号来输入，甚至可以随时交替地使用两种方式。另外，我们的公式编辑器尽可能模仿常规文本编辑器以降低用户的学习成本，他们可以用他们习惯的方式导航和选区，可以通过剪切和粘贴移动结构，可以通过撤销和重做修正错误。

公式编辑器成品的核心部分显示编辑中的数学公式，如图7-1所示。数学公

[1]网页版见`http://chungkwong.cc/equation-editor/equation-editor.html`。

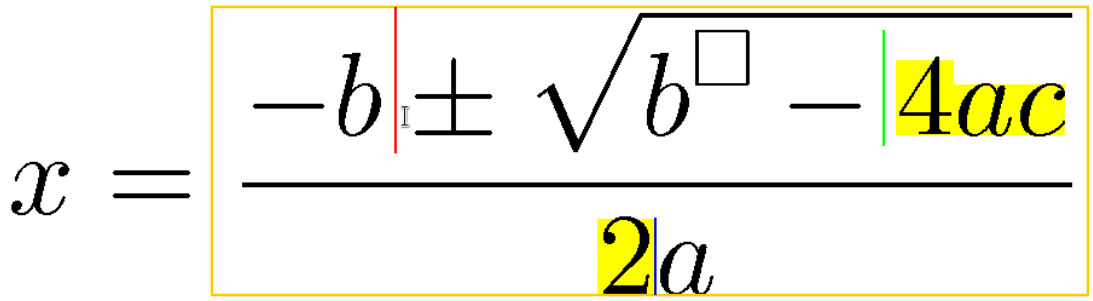


图 7-1 公式编辑区的截图

式在编辑过程中始终实时用高质量算法渲染以实现真正的即见即所得。由于我们希望公式编辑器不仅能用于从头开始创建数学公式，还要能够用于高效地修改数学公式，容许用户选取数学公式的一部分以便进行替换或复制是重要的。数学公式中可以插入子公式的位置称为插入点，一条公式中的所有插入点和符号可以按照它们在规范化 LaTeX 表示中对应的位置排成一列。所选范围由介于锚点和游标间的符号组成，其中锚点和游标分别用绿色竖线和蓝色竖线表示，所选符号则用黄色背景高亮。由于编辑操作一般仅能应用到良构的子公式，编辑器用橙色方框标出了包含所选范围的最小良构子公式。

鼠标操作

公式编辑器容许用户仅用鼠标完成各种编辑操作。用户在进行操作前可以通过拖放选取操作对象，按下鼠标位置附近的插入点将成为锚点而放开鼠标位置附近的插入点将成为游标。为了提高选取过程的准确性，与光标最近的插入点会用红色竖线标出，让用户更好地掌握按下和放开鼠标的时机。在选好操作对象后，用户通过点击如图7-2所示的工具箱中的按钮进行操作。大部分按钮用于输入符号，按下后操作对象会被替换为指定的符号，并把锚点和游标移到符号后。点击用于输入上标、下标、分式和根式的按钮后，操作对象则会被替换为相应结构，在操作对象非空时它将用作红色方框标示的部分，否则把锚点与游标移动至可输入该部分的位置。例如在所选范围非空时，点击上标和分式按钮分别使操作对象成为上标和分子。另外，也有用于剪切、粘贴和删除的按钮。按钮的工具提示给出对应的键盘快捷方式，让用户可逐渐地学会他们对应的键盘操作，从而渐进式地完成从新手到老手的过渡。

键盘操作

公式编辑器也容许用户仅用键盘完成各种操作，纯键盘操作可以避免用户在键盘和鼠标间切换时分心 [78]。它故意被设计成当用户通过键盘输入 LaTeX 代码时即可准确地得到它表示的数学公式，这使得专业用户可以高速盲打，同时继续享受实时预览的便利。我们的原型系统仅支持表5-3中语法给出的 LaTeX 数

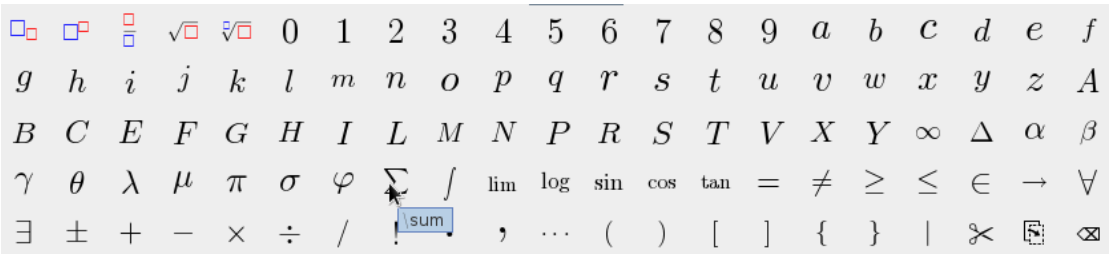


图 7-2 公式编辑器工具箱的截图

图 7-3 自动补全的截图

学模式子语言，但需要时完全可轻易地推广到更一般的情况。为了加速用户输入 LaTeX 代码的效率和降低记忆负担，编辑器提供了图7-3所示的自动补全功能。此外，编辑器支持丰富的键盘快捷键，提供与常规文本编辑器一致的导航和选区体验，表7-1比较了常见的按键组合的行为映射。

7.2.3 整合了手写识别的公式编辑器

并排呈现笔迹与识别结果

为了让用户更容易地发现错误从而能够加以修正，应当并排地显示笔迹和识别结果，如图7-4所示。用户可以在手写区书写数学公式，也可以选取图片再通过笔划提取化为笔迹。值得注意的是，识别结果中的各个符号分别用不同颜色显示，而笔划与它所属的符号同色。可视化笔划与符号之间的对应关系有助于发现和定位一些错误，例如属于同一符号的笔划被标为不同颜色意味着符号切分出错，而这很可能导致结果出错。另外，手写识别结果只替换当前选择的良构子公式，因而用户可以自如地切换编辑方式。比如说，用户正在输入 LaTeX 代码时忘记了某个符号对应命令的话，就可以通过手写输入它，然后又可以随时用回键盘。由于长公式识别的准确率通常会较低，局部重写原来通过手写输入的公式也可能是有用的。

对于联机手写的情况，为了尽快给予用户反馈，显示已写的部分的识别结果而非等待用户确认输入完毕后再识别是有益的，因为严重的局部识别错误很可能导致整体识别错误，继续写下去意义不大。增量式识别系统由于能降低响应时间而特别适合用于这场景。对于其它识别系统，只要识别在背景进行，不阻碍用户继续书写也没有大问题。

为了显示笔划与符号间的对应关系，联机手写数学公式识别系统首先需要能提供对齐信息。对于模块化的传统系统，这一般不是问题。对于采用编码器-解

表 7-1 文本编辑器与数学公式编辑器的导航相关键盘快捷键对比

按键	公式编辑器	文本编辑器
Home	把锚点和游标移到所在最小良构子公式左侧	把锚点和游标移到行首
End	把锚点和游标移到所在最小良构子公式右侧	把锚点和游标移到行末
ctrl + Home	把锚点和游标移到公式最左边	把锚点和游标移到文件开首
ctrl + End	把锚点和游标移到公式最右边	把锚点和游标移到文件结尾
←	向后移动锚点和游标一个插入点	向后移动锚点和游标一个字符
→	向前移动锚点和游标一个插入点	向前移动锚点和游标一个字符
ctrl + ←	向后移动锚点和游标一个最小非空良构子公式	向后移动锚点和游标一个单词
ctrl + →	向前移动锚点和游标一个最小非空良构子公式	向前移动锚点和游标一个单词
⇧ + Home	把游标移到所在最小良构子公式左侧	把游标移到行首
⇧ + End	把游标移到所在最小良构子公式右侧	把游标移到行末
ctrl + ⇧ + Home	把游标移到公式最左边	把游标移到文件开首
ctrl + ⇧ + End	把游标移到公式最右边	把游标移到文件结尾
⇧ + ←	向后移动游标一个插入点	向后移动游标一个字符
⇧ + →	向前移动游标一个插入点	向前移动游标一个字符
ctrl + ⇧ + ←	向后移动游标一个最小非空良构子公式	向后移动游标一个单词
ctrl + ⇧ + →	向前移动游标一个最小非空良构子公式	向前移动游标一个单词
ctrl + A	选取整条公式	选取整个文件

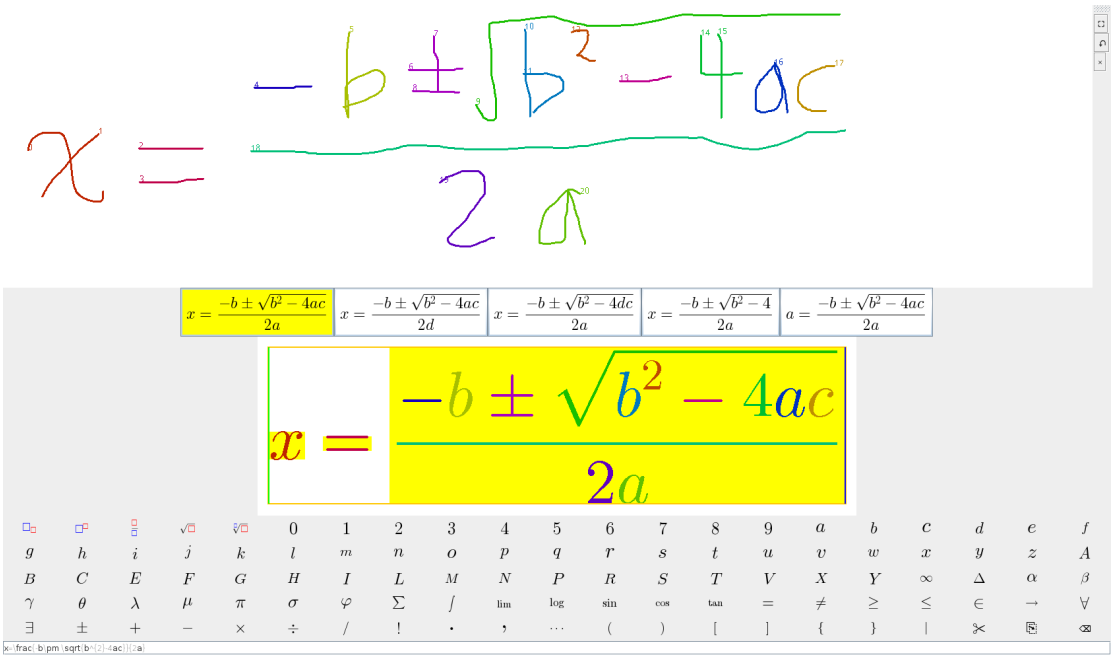


图 7-4 整合了手写识别的公式编辑器的截图

码器框架的系统，其中也往往带有为对齐建模的组件，例如我们使用注意力机制中的收敛向量变化来估计对齐。

多候选

当数学公式识别系统没有把握在若干个候选中作出选择时，让用户来选择而非勉强选取一个可能是好注意。事实上，英文联机手写识别系统一般会提供多个候选单词供用户选择，手写中文输入法也往往会提供若干个字符候选供用户选择，因此把同一做法应用到手写数学公式并不奇怪。我们的公式编辑器同时提供数学公式级的候选和符号级的候选以便让用户方便地修正不同类型的错误，如图7-5所示。当第一候选不正确时，用户可以一键把识别结果改为其它数个候选之一。当个别符号出现识别错误时，用户可以通过双击把它改为下一符号候选，也可以在上下文菜单中选取候选。

为了实现多候选功能，所基于的数学公式识别系统需要能够给出多个候选，这对常规的系统而言应该都不会是问题，因为它们的工作原理本来就是先生成大量候选然后企图寻找某指标最优的那个作为识别结果。对于模块化系统，由于有专门的符号识别器，生成符号级的候选也是自然的。对于基于编码器-解码器框架的系统，解码出一个符号时一般也能同时得到其它符号类别的预测概率。

为了说明多候选对手写数学公式编辑的重要性，表7-2列出了如有能从若干个候选中选取最佳者的谕示器，单个 TAP 模型在 CROHME 2016 数据集上所能达到的准确率。当提供 5 个整体候选时，对于联机和脱机情况，用户通过一键选择候选可分别使 15.43% 和 15.26% 的公式的识别结果从错误变为正确。同时，

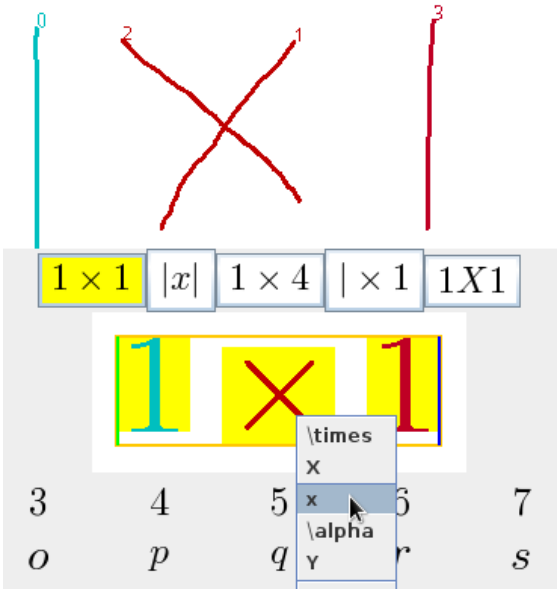


图 7-5 多候选展示的截图

对于联机和脱机情况，用户通过仅重选一个符号可分别使 12.29% 和 13.86% 的公式的识别结果从错误变为正确。更进一步，对于联机和脱机情况，用户通过修改公式级候选和重选最多一个符号可分别使 24.94% 和 26.42% 的公式的识别结果从错误变为正确。以上结果说明，整体多候选和局部多候选都有助于修正相当比例的公式的识别结果，而且两者有互补性。表7-3列出了基于三个 TAP 模型时的实验结果。可见，虽然用组合模型的话第一候选准确率要明显比单一模型高一些，但前五准确率差别不大。同时，组合模型会比单一模型慢若干倍，故采用多候选策略可能有助于降低对识别准确率的要求。

表 7-2 基于单个 TAP 模型（带形式语法约束）的手写识别系统在 CROHME 2016 测试集上结构正确且符号错误数少于特定值的公式的百分比

候选数	符号错误数					
	$=0$	≤ 1	≤ 2	≤ 3	≤ 4	不限
联机识别						
1	46.73	59.02	63.73	65.04	66.09	66.61
2	53.88	65.04	68.88	70.01	70.88	71.49
3	58.06	68.00	71.58	72.89	73.67	74.46
4	60.07	69.92	73.41	74.54	75.41	76.20
5	62.16	71.67	75.41	76.46	77.42	78.20
脱机识别						
1	45.51	59.37	65.21	66.87	67.74	68.53
2	53.88	65.91	70.36	71.67	72.54	73.23
3	57.63	69.22	72.97	73.93	74.98	75.68
4	59.37	70.79	74.46	75.50	76.46	77.16
5	60.77	71.93	75.50	76.90	77.68	78.38

表 7-3 基于三个 TAP 模型（带形式语法约束）的手写识别系统在 CROHME 2016 测试集上结构正确且符号错误数少于特定值的公式的百分比

候选数	符号错误数					
	$=0$	≤ 1	≤ 2	≤ 3	≤ 4	不限
联机识别						
1	48.56	61.90	66.43	67.48	68.44	69.22
2	57.45	68.35	72.45	74.11	74.46	75.33
3	61.20	71.14	74.89	76.37	77.16	77.94
4	63.73	72.97	76.90	78.38	78.99	79.77
5	65.65	74.46	78.38	79.77	80.30	81.17
脱机识别						
1	49.69	64.34	69.05	70.62	71.49	72.19
2	57.80	70.62	74.54	75.94	76.63	77.24
3	61.64	72.89	76.72	77.94	78.64	79.25
4	63.21	74.37	77.94	79.08	79.95	80.65
5	64.43	76.02	79.16	80.30	81.34	82.04

7.3 数学公式检索

7.3.1 数学公式检索系统

基本的数学公式检索接受一条完整的数学公式作为查询，然后寻找与之在某种意义下相似的一些数学公式，这可以用于找出一条公式的出处。不过，在一些应用场景中查询可能还需要包含通配符：

寻找定义 人们可能希望通过搜索 “$\#1\# = x \times x$” 或者 “$x^2 = \#1\#$” 找到 “$x^2 = x \times x$”。

寻找证明 人们可能希望通过查询 “$(a+b)^2-(a-b)^2 = \#1\# = 4ab$” 找到恒等式 “$(a+b)^2-(a-b)^2 = 4ab$” 的证明 “$(a+b)^2-(a-b)^2 = ((a+b)+(a-b))\,((a+b)-(a-b)) = (2a)\,(2b) = 4ab$”。

寻找例子 人们可能希望通过查询 “$1+\frac{1}{1+\frac{1}{\#1\#}}$” 找连分数的例子，比如 “$1+\cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{\ddots}}}$”。

有时，我们还希望两个通配符对应的值相同，例如由于约束变元的选取不影响表达式的意义，用户可能希望查询 “$\int \sin(\#1\#)\,\mathrm{d}\#1\#$” 匹配 “$\int \sin(x)\,\mathrm{d}x$” 和 “$\int \sin(y)\,\mathrm{d}y$”，但不匹配 “$\int \sin(x)\,\mathrm{d}y$”。虽然十二届信息获取技术评价会议（Conference on Evaluation of Information Access Technologies，NTCIR）数学信息检索任务（MathIR）的孤立数学公式检索子任务只用到上述的查询语言，一些数学公式检索系统支持更一般的查询。有的系统 [48] 支持逻辑查询，例如可指定一些子公式一定要在结果中出现或不出现。

数学公式检索的查询有时是通过手写输入的，而除极个别的检索系统 [131] 可直接对比图片外，其它系统都只接受数学公式在某种标记语言中的表示，因此需要先进行识别。由于识别结果不一定完全准确，而要求用户人手修正又会浪费时间和降低用户体验，本节探索不准确的查询对检索结果的影响。和普通全文检索系统类似，数学公式检索系统一般也是先用倒排索引找出一些候选再通过计算相似度进行排序，因此需要分别分析输入错误对这两步的影响。值得注意的是，许多分析也适用于数学公式库中的公式也不准确的情况，例如公式库也是通过手写识别得到时。

7.3.2 倒排索引

为了加快搜索过程，数学公式检索系统一般会事先对公式库进行索引工作。也就是说，公式库中的每条公式都会被拆解为一些索引项，反向索引记录每个索引项在哪些公式中出现过。对于简单的查询，搜索时只需用相同方式把查询拆解为索引项，然后就能从反向索引找出与之有共同索引项的公式作进一步处理，由此大幅收窄搜索范围。

一般来说，由于数学公式检索的目的是找相似的公式而非简单地找精确匹配的公式，两条不一样但相似的公式应当有共同的索引项。特别地，手写数学公式的识别结果和用户本来想写的公式间通常有很大的相似性，因此两者也往往有不少共同的索引项。共同索引项的多少与所用的拆解方法有关，以下对其中一些索引项的生成方法进行讨论：

- 有的数学公式检索系统用数学公式树表示中的子树 [105]、符号对 [106] 或路径 [49] 作为索引项。与仅用符号作为索引项相比，这时识别错误可能会影响更大比例的索引项，例如识别错一个符号就会把它所在的所有子树、符号对和路径都改掉。
- 有的数学公式检索系统 [68, 105] 对成立交换律的操作的操作数进行排序，例如把 “x^2+1” 规范化为 “$1+x^2$”。这种规范化要求先构建操作树，而出现识别错误时从语法树到操作树的转换并不可靠。
- 有的数学公式检索系统 [105, 49] 会对约束变量进行重命名，例如把 “$\forall n P(n)$” 规范化为 “$\forall \#1\# P(\#1\#)$”。如果约束变量名出现识别错误，就会影响这种索引项。
- 有的数学公式检索系统 [68] 会在索引或搜索时添加同义项，例如把 “sh” 规范化 “sinh”。如果 “sh” 被误识为 “sn” 之类，就不能被规范化了。不过，类似的查询扩展有助找回出现常见字符识别错误的公式 [95]，例如在碰到项 “S” 时可加入 “s”。
- 有的数学公式检索系统 [41] 会把子表达式替换成通配符，例如为 “$\frac{1}{1+x^2}$” 加入索引项 “$\frac{\#1\#}{\#2\#}$”。这样只要结构正确，即使所有符号都分错了类都不会影响这些项。

7.3.3 排序指标

由于用户一般只会检视前面几个搜索结果，数学公式检索系统要尽可能把最相关的结果排在前面。排序可以是基于特征的，也可以是基于结构匹配的。由于基于特征的排序一般较快，但基于结构匹配的排序结果往往更合理，因此一些系统先基于特征排序再对前面若干个结果用结构匹配重新排序。

在基于特征的排序中，查询和候选公式分别被表示为向量，然后计算向量间的相似度。这个向量可以是基于传统的项频率和公式频率倒数，也可能是专门为数学公式设计的特征，甚至可以用基于人工神经网络的编码器生成 [114]。至于相似度，既可以用标准的相似性度量如余弦和 BM25，也可以用定制的量度。由于我们把基于特征的排序当作一种优化手段，我们不打算详细展开。

在基于结构的匹配中，直接对比查询与候选并计算它们之间的某种相似度，顺便得到通配符对应的子公式。一类典型相似度基于某种编辑距离 [44]，这类度量往往也可以作为数学公式识别的评价指标。我们以下将考虑用规范化 LaTeX 表示间单词级别的编辑距离作为排序指标的情况。

对于精确搜索的目的而言，使用编辑距离排序的一个好处是能够事先评估纠错能力。作为编码理论的基本常识，如果在一个编码方案中任意两个不同的有效码间的距离至少为 $2k+1$，则用最短距离译码原则可以纠 k 位错误。同理，如果一个数学公式集合中任何两条不同的数学公式间的编辑距离至少为 $2k+1$，则可以容忍 k 处编辑错误。对于真实数学公式检索应用中涉及的数学公式集合，不能期望总能纠若干个错误，因为一处修改完全可能使一条数学公式变成另一条有意义的公式。然而，我们仍然可以探讨有检索价值的数据集在多大程度上可以纠错，以下是我们将检验的数据集：

NTCIR-12 MathIR 孤立数学公式检索子任务 [129] 提供了一个数学公式集，它是数学公式检索的标准数据集。该数据集提供了超过 59 万条来自维基百科的数学公式纪录，其中有超过 32 万条数学公式互不相同。为了探讨数据集大小的影响，我们还从该数据集（伪）随机地分别抽取了由 32,000 和 3,200 条互异公式组成的子集。

Wikidata 包含于 2020 年 7 月 18 日从一个开放的语义网[2]取得的 4492 条与属性“公式定义”（用于把定理或定律与其数学公式表达联系起来，编号为“wdt:P2534”）相关的纪录，其中有 4408 条互不相同的公式。由于维基数据的覆盖面和结构化特性，有关纪录有检索价值。

CROHME 2016 测试集包含 1147 条数学公式。由于有对应的手写笔迹，适合用于测试识别错误对检索的影响。

表7-4列出了在各个数学公式集中，按最短距离译码原则提供 n 个候选时，可以保证识别错误不超过 k 时正确公式有多大可能性出现在候选中。显然，在其它条件不变时，提供的候选越多，识别错误越少，正确公式就越可能出现在候选中。值得注意的是，随着数学公式集增大，虽然相应的百分比在下降，但下降的速度在下降。同时，我们看到 CROHME 2016 数据集的难度与其它数据集比并不低，因而在其上进行的进一步实验有参考价值。

为了减低识别错误对搜索结果的影响，我们提出对基于编辑距离的匹配方法进行两项修改。虽然我们仅对序列的情况进行陈述，但它们原则上也可以用于包括树编辑距离在内的其它编辑距离。给定序列 $\mathbf{x}=(x_0,\ldots,x_{m-1})$ 和 $\mathbf{y}=(y_0,\ldots,y_{n-1})$，经典的编辑距离 d 由递推公式

$$\mathrm{d}(\mathbf{x},\mathbf{y})=\begin{cases} n & ,m=0 \\ m & ,n=0 \\ \min\{\mathrm{d}(\check{\mathbf{x}},\check{\mathbf{y}}),\mathrm{d}(\mathbf{x},\check{\mathbf{y}})+1,\mathrm{d}(\check{\mathbf{x}},\mathbf{y})+1\} & ,m>0,n>0,x_{m-1}=y_{n-1} \\ \min\{\mathrm{d}(\check{\mathbf{x}},\check{\mathbf{y}}),\mathrm{d}(\mathbf{x},\check{\mathbf{y}}),\mathrm{d}(\check{\mathbf{x}},\mathbf{y})\}+1 & ,m>0,n>0,x_{m-1}\neq y_{n-1} \end{cases} \tag{7-1}$$

给出，其中 $\check{\mathbf{z}}$ 则表示从序列 $\mathbf{z}$ 中删去最后一项得到的序列。我们的第一个建议

[2]参见https://www.wikidata.org/。

表 7-4 各公式集中与第 n 条与之最近的相异公式间编辑距离不少于 $2k+1$ 的公式的百分比

n \ k	1	2	3	4	5	6
NTCIR-12 MathIR						
1	63.77	49.98	40.95	34.09	28.96	24.94
2	72.95	59.39	49.98	42.49	36.58	31.67
3	76.25	62.55	53.20	45.62	39.51	34.55
4	78.00	64.32	54.94	47.36	41.25	36.14
CROHME 2016						
1	68.60	50.53	37.72	28.68	20.61	15.61
2	77.54	59.56	47.02	35.96	26.40	19.39
3	82.11	63.60	51.23	39.12	29.91	22.54
4	85.09	66.84	54.12	42.46	32.11	24.30
NTCIR-12 MathIR-32000						
1	79.08	64.71	55.20	47.88	41.74	36.89
2	83.90	69.39	59.63	51.99	45.53	40.34
3	85.78	71.11	61.22	53.53	47.01	41.67
4	87.01	72.27	62.15	54.43	47.94	42.61
NTCIR-12 MathIR-3200						
1	88.34	75.03	65.13	57.13	50.63	44.97
2	91.38	77.94	68.03	59.91	53.16	47.53
3	93.31	79.19	69.47	61.19	54.50	48.81
4	94.16	80.06	70.25	61.88	55.16	49.34
Wikidata						
1	89.47	80.94	72.21	63.68	56.33	49.73
2	94.42	86.75	78.54	70.01	61.73	55.31
3	96.08	89.54	81.58	73.14	64.72	57.60
4	96.67	90.40	83.39	74.91	66.99	59.39

是为不同的错误赋予不同的重视程度，而不是像常规的编辑距离那样把所有错误都一视同仁。比如说，“C” 被误识为 “c” 的可能性远比被误识为 “+” 的可能性大，因此应当让从 “c” 到 “C” 的 “距离” 短于从 “+” 到 “C” 的 “距离”。这样，常见的识别错误对修改后的 “编辑距离” 的相对影响就会较低，于是对排序结果的影响也会较低。形式地说，可用

$$\overline{\mathrm{d}}\left(\mathbf{x},\mathbf{y}\right)=\begin{cases} n & ,m=0 \\ m & ,n=0 \\ \min\left\{\overline{\mathrm{d}}\left(\check{\mathbf{x}},\check{\mathbf{y}}\right)-\log p_{x_{m-1}},\overline{\mathrm{d}}\left(\mathbf{x},\check{\mathbf{y}}\right)-\log p^{y_{n-1}},\right. & \\ \qquad\left.\overline{\mathrm{d}}\left(\check{\mathbf{x}},\mathbf{y}\right)-\log p_{x_{m-1}}^{y_{n-1}}\right\} & ,m>0,n>0 \end{cases} \tag{7-2}$$

量度从识别结果 $\mathbf{x}$ 到公式库中公式 $\mathbf{y}$ 的 “距离”，其中 $p_{x_{m-1}}$ 表示 x_{m-1} 被无中生有地多识出来的概率，$p^{y_{n-1}}$ 表示 y_{n-1} 被漏识的概率，而 $p_{x_{m-1}}^{y_{n-1}}$ 表示 y_{n-1} 被识别为 x_{m-1} 的概率。我们的第二个建议是利用多个候选识别结果来扩展查询。假设对于一条手写数学公式，识别系统给出的前 ℓ 个候选分别为 $\mathbf{x}_0,\ldots,\mathbf{x}_{\ell-1}$，则可用

$$\tilde{\mathrm{d}}\left(\left(\mathbf{x}_0,\ldots,\mathbf{x}_{\ell-1}\right),\mathbf{y}\right)=\min\left\{-\log q_i+\overline{\mathrm{d}}\left(\mathbf{x}_i,\mathbf{y}\right)\mid i=0,\ldots,\ell-1\right\} \tag{7-3}$$

给出从该手写数学公式到公式库中公式 $\mathbf{y}$ 的 “距离”，其中 q_i 表示候选 i 比其它候选都准确的概率。对于一个数学公式识别系统，上述的概率都是可以通过实验估计的参数。

假设要搜索的公式库由 CROHME 2016 测试集给出，用户分别通过手写搜索其中的每条公式，而手写公式的笔迹或图片和 CROHME 数据集中的相同，我们就可以知道用识别结果作为查询时真正的公式出现在结果的第几位。实验结果如7-5所示，我们看到无论是用哪个手写识别系统，搜索的前一准确率都远比相应的识别准确率高，对于较弱的 TAP（仅使用了单个模型和形式语法约束）而言提升尤其明显。同时，可以看出利用多个识别结果扩展查询和使用错误感知排序指标都有且提高搜索的准确性，而且两项技术有互补性，其中各参数在 CROHME 2014 测试集上估计。通过使用这两项技术，当用准确率仅约 46% 的手写数学公式识别系统的识别结果作为查询时，前五搜索结果找回了约 98% 的数学公式，这对于数学公式检索的目的而言可能已经足够。

虽然我们只对精确搜索进行了实验，但查询语言可以推广为可带 glob 风格通配符的 LaTeX 代码。由于 glob 模式语言已经被广泛用于描述文件路径集，这种查询语言容易学习。同时，通配符在计算上带来的额外开销很小。支持一种可

表 7-5 CROHME 2016 测试集上前 k 个搜索结果的准确率

系统	错误感知	识别候选	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
联机识别系统							
MyScript	×	1	94.51	97.21	98.17	98.52	98.95
MyScript	✓	1	96.25	98.34	98.87	99.22	99.48
TAP	×	1	85.79	91.02	92.33	93.03	93.64
TAP	×	2	88.40	92.85	94.16	94.51	95.03
TAP	×	5	90.32	93.90	95.64	96.43	96.51
TAP	✓	1	89.28	94.33	95.55	96.34	96.60
TAP	✓	5	93.03	96.16	96.95	97.56	97.65
脱机识别系统							
MyScript	×	1	92.85	96.16	97.30	97.56	98.08
MyScript	✓	1	95.03	97.38	97.99	98.43	98.69
TAP	×	1	87.53	92.41	93.46	94.33	94.86
TAP	×	2	90.76	94.07	94.94	95.64	95.99
TAP	×	5	93.11	96.16	97.12	97.47	97.82
TAP	✓	1	91.54	95.20	96.43	97.30	97.65
TAP	✓	5	94.59	97.38	97.99	98.78	98.78

以匹配任意子序列的通配符是容易的，只要把 $\overline{\mathrm{d}}$ 的递推公式改为

$$\underline{\mathrm{d}}(\mathbf{x},\mathbf{y})=\begin{cases} m & ,n=0 \\ \underline{\mathrm{d}}(\mathbf{x},\check{\mathbf{y}})-\log p^{y_{n-1}} & ,m=0,n>0,y_{n-1}\text{非通配符} \\ \underline{\mathrm{d}}(\mathbf{x},\check{\mathbf{y}}) & ,m=0,n>0,y_{n-1}\text{为通配符} \\ \min\{\underline{\mathrm{d}}(\check{\mathbf{x}},\check{\mathbf{y}})-\log p_{x_{m-1}}, & \\ \quad \underline{\mathrm{d}}(\mathbf{x},\check{\mathbf{y}})-\log p^{y_{n-1}}, & ,m>0,n>0,y_{n-1}\text{非通配符} \\ \quad \underline{\mathrm{d}}(\check{\mathbf{x}},\mathbf{y})-\log p_{x_{m-1}}^{y_{n-1}}\} & \\ \min\{\underline{\mathrm{d}}(\check{\mathbf{x}},\mathbf{y}),\underline{\mathrm{d}}(\mathbf{x},\check{\mathbf{y}})\} & ,m>0,n>0,y_{n-1}\text{为通配符} \end{cases}。 \tag{7-4}$$

动态规划算法仍然只需用 $\Theta(mn)$ 时间和 $\Theta(\min\{m,n\})$ 辅助空间即可算出使一个长度为 m 的（不含通配符）序列匹配一个长度为 n 的（可含通配符）序列所需的编辑次数。需要时，也可以记录各通配分别符捕获的子序列，以备向用户呈现结果时用。类似地，可以加入可匹配单个任意元素的通配符，也可以加入可匹配若干个给定元素之一的通配符。

一般地，对于任何一个序列和任何一个上下文无关语法，可以考虑该序列与上下文无关语言中序列间编辑距离的最小值 [2]。用于解析上下文无关语法的动态规划算法稍作修改后即可在 $O(n^3)$ 时间内完成计算，其中 n 为序列的长度，常数因子与语法有关。修改后算法对每一个子序列和每一个非终结符，计算使该序列能由该非终结符导出所需的最少编辑次数。通过按子序列长度递增的顺序计算，最终可以得到把整个序列化为开始符号能导出的序列所需的最少编辑次数。

然而，泵引理表明即使上下文无关语言也无法表达合一的概念，例如 $\{w2w|w\in\{0,1\}^*\}$ 不是上下文无关语言 [1]。在应用中，一种近似地实现合一的方法是计算应当合一的子公式间的编辑距离再加到排序指标中。

7.4 本章小结

虽然手写数学公式识别的结果不一定完全正确，但通常都和正确的结果有很多共通点。一方面，这意味着借助一个良好的公式编辑器，用户只需一次或数次操作即可迅速地修正大部分错误。向用户提供多候选和直观的编辑操作可以加快人手修正的过程。另一方面，这意味着利用基于最短距离译码原则的方法，在多选一以至数学公式检索的应用场景中可能可自动地纠正大部分错误。通过选取识别错误感知的“编辑距离”和利用候选识别结果来扩展查询，可以减低识别错误对数学公式检索结果的影响，这方法可以推广到带通配符查询。

第八章　结语

8.1　成果总结

手写数学公式识别是一个困难的实际问题，它又分为联机识别和脱机识别两个相关的子问题。联机手写数学公式识别技术相对成熟，这不仅体现在研究和产品化远比脱机情形开展得早，而且体现在目前的联机识别系统一般较准确和较节约资源。为了使得基于常规联机识别系统构建脱机识别系统成为可行，本文提出了一个针对脱机手写数学公式的笔划提取算法。实验表明，基于笔划提取的脱机手写数学公式识别方案确实能把联机识别的优点带到脱机识别。

首先，本文验证了这个方案可以达到较高的准确度，即使所基于的联机识别系统并不是专门为提取出的笔划而设计的。通过与现成的商用联机识别系统相结合，本文方法在 CROHME 数据集上取得了良好的准确率。通过用提取出的笔划重新训练联机识别系统，得到了准确性与之相若的脱机识别系统；通过用混合数据重新训练联机识别系统，得到了同时在联机和脱机数据上表现良好的模型。在理论上，这意味着脱机识别本质上可能并不比联机识别难很多，而且主要难在背景噪声的存在而非动态信息的缺失。在实践中，这意味着分别开发联机识别系统和脱机识别系统可能不再是必要的，集中精力发展联机识别而通过笔划提取让脱机识别自动跟上可能是更经济的。

其次，本文验证了这个方案特别适用于资源受限设备上的实时应用场景。因为联机识别一般而言比脱机识别快且省内存，高效的笔划提取提供了一个在手机、平板电脑和嵌入式系统上实现本地识别的可行方案。这个方案不用像云计算那样面临那么多网络延迟和信息安全等问题。作为一个副产物，本文提出了一类具代表性的局部自适应二值化方法的内存高效快速实现。

和其它脱机手写数学公式识别系统一样，本文方法也不能提供绝对的准确性。本文方法的准确性对于某些应用来说可能是足够的，例如数学公式检索系统可能可以用这技术来容许用户通过上传图片来输入查询，通过使用识别感知的排序方法能够进一步降低对识别错误的敏感度。对于因为要求输入准确而要求用户人手修正识别结果的应用，基于笔划提取的方法则有助于让公式编辑器可视化图片与识别结果间的关系，从而让用户更容易地发现和修正错误。

8.2 未来展望

纵使本文揭示了把脱机识别归结为联机识别的经典想法在深度学习时代有重生的潜力，但还是留下了很多可以继续挖掘的空间。通过进一步的工作，可望做出一系列在真实场景下可用的产品。

首先，手写数学公式定位对脱机手写数学公式识别技术的实用化来说是关键的。包括本文方法在内的已知脱机手写数学公式识别方法一般都假设每张图片有且仅有一条数学公式，而没有其它干扰因素。然而，现实中通过拍照或扫描得到的图片往往附带别的东西如旁边的文字和信纸的背景，因而需要先行把数学公式分离出来。虽然人们已经在场景文本检测方面进行了广泛的研究 [126]，但针对数学公式检测的研究还处于非常初步的阶段，而且往往只局限于找出印刷体数学公式的外接方框。同时，虽然图像预处理方面有很多用于去除噪声的滤镜如各种二值化方法，但其中一些并不适用于存在大小相差很大的符号（如小数点和根号）的数学公式。

其次，基于笔划提取的脱机手写识别方法在数学公式以外的手写对象上的应用是可以期待的。笔划提取是一个原则上可以应用于任何脱机手写识别问题的一般方法。既然把它应用于数学公式时能够取得正面的结果，那么把它应用于化学公式、乐谱、图表或其它手写对象时的效果就是值得审视的。如果基于笔划提取的脱机手写识别方法在这些手写对象上继续取得成功，将能验证其广泛适用性。同时，是否需要为不同类型的手写体分别设计不同的笔划提取器是一个有趣的问题。此外，笔划提取方法对印刷体识别的有效性也是一个可以探讨的问题。

参考文献

[1] AHO A V, LAM M S, SETHI R, et al. Compilers: Principles, Techniques, and Tools (2nd Edition). Boston: Addison-Wesley, 2006

[2] AHO A V, PETERSON T G. A minimum distance error-correcting parser for context-free languages. SIAM Journal on Computing, 1972, **1** (4): 305–312

[3] ÁLVARO F, SÁNCHEZ J A, BENEDÍ J M. Recognition of on-line handwritten mathematical expressions using 2d stochastic context-free grammars and hidden markov models. Pattern Recognition Letters, 2014, **35**: 58–67

[4] ÁLVARO F, SÁNCHEZ J A, BENEDÍ J M. An integrated grammar-based approach for mathematical expression recognition. Pattern Recognition, 2016, **51**: 135–147

[5] ANAGNOSTOPOULOS C E, ANAGNOSTOPOULOS I E, PSOROULAS I D, et al. License plate recognition from still images and video sequences: A survey. IEEE Transactions on Intelligent Transportation Systems, 2008, **9** (3): 377–391

[6] ANDERSON R H. Syntax-directed recognition of hand-printed two-dimensional mathematics. Interactive Systems for Experimental Applied Mathematics. Washington. USA. Aug. 1967. 436–459

[7] AUSBROOKS R, BUSWELL S, CARLISLE D, et al. Mathematical markup language (MathML) version 3.0 2nd edition. https://www.w3.org/TR/MathML3, 2014, 4, 10

[8] AWAL A M, MOUCHèRE H, VIARD-GAUDIN C. A global learning approach for an online handwritten mathematical expression recognition system. Pattern Recognition Letters, 2014, **35**: 68–77

[9] BARÓ A, RIBA P, CALVO-ZARAGOZA J, et al. From optical music recognition to handwritten music recognition: A baseline. Pattern Recognition Letters, 2019, **123**: 1–8

[10] BERMAN B P, FATEMAN R J. Optical character recognition for typeset mathematics. International Symposium on Symbolic and Algebraic Computation. Oxford. UK. Aug. 1994. 348–353

[11] BERNSEN J. Dynamic thresholding of grey-level images. International Conference on Pattern Recognition. Berlin. Germany. 1986. 1251–1255

[12] BHOWMIK S, SARKAR R, DAS B, et al. Gib: A *G* ame theory *I* nspired *B* inarization technique for degraded document images. IEEE Transactions on Image Processing, 2019, **28** (3): 1443–1455

[13] BOCCIGNONE G, CHIANESE A, CORDELLA L, et al. Recovering dynamic information from static handwriting. Pattern Recognition, 1993, **26** (3): 409–418

[14] BRADLEY D, ROTH G. Adaptive thresholding using the integral image. Journal of Graphics Tools, 2007, **12** (2): 13–21

[15] BUNKE H. Attributed programmed graph grammars and their application to schematic diagram interpretation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1982, **4** (6): 574–582

[16] CHAMCHONG R, FUNG C C. Comparing background elimination approaches for processing of ancient thai manuscipts on palm leaves. International Conference on Machine Learning and Cybernetics. Hebei. China. Jul. 2009. 3436–3441

[17] CHAN K F, YEUNG D Y. An efficient syntactic approach to structural analysis of on-line handwritten mathematical expressions. Pattern Recognition, 2000, **33** (3): 375–384

[18] CHAN K F, YEUNG D Y. Mathematical expression recognition: a survey. International Journal on Document Analysis and Recognition, 2000, **3** (1): 3–15

[19] CHANG S K. A method for the structural analysis of two-dimensional mathematical expressions. Information Sciences, 1970, **2** (3): 253–272

[20] CHEN X, GAO Y, HUANG Z. CUDA-accelerated fast sauvola's method on kepler architecture. Multimedia Tools and Applications, 2015, **74** (24): 11809–11820

[21] CHOU P A. Recognition of equations using a two-dimensional stochastic context-free grammar. Symposium on Visual Communications, Image Processing, and Intelligent Robotics Systems. Philadelphia. USA. Nov. 1989. 852–863

[22] CROW F C. Summed-area tables for texture mapping. SIGGRAPH Comput. Graph., 1984, **18** (3): 207–212

[23] DAVILA K, LUDI S, ZANIBBI R. Using off-line features and synthetic data for on-line handwritten math symbol recognition. International Conference on Frontiers in Handwriting Recognition. Heraklion. Greece. Sep. 2014. 323–328

[24] DENG Y, KANERVISTO A, LING J, et al. Image-to-markup generation with coarse-to-fine attention. International Conference on Machine Learning. Sydney. Australia. Aug. 2017. 980–989

[25] DOERMANN D S, ROSENFELD A. Recovery of temporal information from static images of handwriting. International Journal of Computer Vision, 1995, **15** (1): 143–164

[26] EPSHTEIN B, OFEK E, WEXLER Y. Detecting text in natural scenes with stroke width transform. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Francisco. USA. Jun. 2010. 2963–2970

[27] ETO Y, SUZUKI M. Mathematical formula recognition using virtual link network. International Conference on Document Analysis and Recognition. Seattle. USA. Sep. 2001. 762–767

[28] FAN K C, WU W H. A run-length-coding-based approach to stroke extraction of chinese characters. Pattern Recognition, 2000, **33** (11): 1881–1895

[29] FENG M L, TAN Y P. Contrast adaptive binarization of low quality document images. IEICE Electronics Express, 2004, **1** (16): 501–506

[30] FILIPPOV I V, NICKLAUS M C. Optical structure recognition software to recover chemical information: OSRA, an open source solution. Journal of Chemical Information and Modeling, 2009, **49** (3): 740–743

[31] GAO L, HUANG Y, DÈJEAN H, et al. Icdar 2019 competition on table detection and recognition (cTDaR). International Conference on Document Analysis and Recognition. Sydney. Australia. Sep. 2019. 1510–1515

[32] GARAIN U, CHAUDHURI B B. OCR of printed mathematical expressions. Digital Document Processing: Major Directions and Recent Advances. London. UK. 2007. 235–259

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

[34] GOLUBITSKY O, WATT S M. Distance-based classification of handwritten symbols. International Journal on Document Analysis and Recognition, 2010, **13** (2): 133–146

[35] GRAVES A, MOHAMED A, HINTON G. Speech recognition with deep recurrent neural networks. IEEE International Conference on Acoustics, Speech and Signal Processing. Vancouver. Canada. May. 2013. 6645–6649

[36] GUIDI F, COEN C S. A survey on retrieval of mathematical knowledge. Mathematics in Computer Science, 2016, **10** (4): 409–427

[37] HE K, SUN J, TANG X. Guided image filtering. European Conference on Computer Vision. Heraklion. Greece. Sep. 2010. 1–14

[38] HE L, REN X, GAO Q, et al. The connected-component labeling problem: A review of state-of-the-art algorithms. Pattern Recognition, 2017, **70**: 25–43

[39] HONG Z, YOU N, TAN J, et al. Residual BiRNN based seq2seq model with transition probability matrix for online handwritten mathematical expression recognition. International Conference on Document Analysis and Recognition Recognition. Sydney. Australia. Sep. 2019. 635–640

[40] HSIEH C C, LEE H J. Off-line recognition of handwritten chinese characters by on-line model-guided matching. Pattern Recognition, 1992, **25** (11): 1337–1352

[41] HU X, GAO L, LIN X, et al. Wikimirs: a mathematical information retrieval system for wikipedia. ACM/IEEE-CS Joint Conference on Digital Libraries. Indianapolis. USA. Jul. 2013. 11–20

[42] JÄGER S. Recovering writing traces in off-line handwriting recognition: using a global optimization technique. International Conference on Pattern Recognition. Vienna. Austria. Aug. 1996. 150–154

[43] JULCA-AGUILAR F, MOUCHÈRE H, VIARD-GAUDIN C, et al. A general framework for the recognition of online handwritten graphics. International Journal on Document Analysis and Recognition, 2020, **23** (2): 143–160

[44] KAMALI S, TOMPA F W. Structural similarity search for mathematics retrieval. International Conference on Intelligent Computer Mathematics. Bath. UK. Jul. 2013. 246–262

[45] KATO Y, YASUHARA M. Recovery of drawing order from single-stroke handwriting images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, **22** (9): 938–949

[46] KHURSHID K, SIDDIQI I, FAURE C, et al. Comparison of niblack inspired binarization methods for ancient documents. Document Recognition and Retrieval. San Jose. USA. Jan. 2009. 72470U

[47] KNUTH D E. The TeXbook. Reading: Addison-Wesley, 1984

[48] KOHLHASE M, SUCAN I. A search engine for mathematical formulae. Artificial Intelligence and Symbolic Computation. Beijing. China. Sep. 2006. 241–253

[49] KRISTIANTO G Y, TOPIC G, AIZAWA A. MCAT math retrieval system for NTCIR-12 MathIR task. NTCIR Conference on Evaluation of Information Access Technologies. Tokyo. Japan. Jun. 2016. 323–330

[50] LAU K K, YUEN P C, TANG Y Y. Stroke extraction and stroke sequence estimation on signatures. International Conference on Pattern Recognition. Quebec City. Canada. Aug. 2002. 119–122

[51] LAZZARA G, GÉRAUD T. Efficient multiscale sauvola's binarization. International Journal on Document Analysis and Recognition, 2014, **17** (2): 105–123

[52] LE A D. Recognizing handwritten mathematical expressions via paired dual loss attention network and printed mathematical expressions. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. Seattle. USA. Jun. 2020. 2413–2418

[53] LE A D, INDURKHYA B, NAKAGAWA M. Pattern generation strategies for improving recognition of handwritten mathematical expressions. Pattern Recognition Letters, 2019, **128**: 255–262

[54] LE A D, NAKAGAWA M. A system for recognizing online handwritten mathematical expressions by using improved structural analysis. International Journal on Document Analysis and Recognition, 2016, **19** (4): 305–319

[55] LE A D, NGUYEN H D, INDURKHYA B, et al. Stroke order normalization for improving recognition of online handwritten mathematical expressions. International Journal on Document Analysis and Recognition, 2019, **22** (1): 29–39

[56] LEE H J, LEE M C. Understanding mathematical expressions using procedure-oriented transformation. Pattern Recognition, 1994, **27** (3): 447–457

[57] LEE S, PAN J C. Offline tracing and representation of signatures. IEEE Transactions on Systems, Man, and Cybernetics, 1992, **22** (4): 755–771

[58] LI Z, JIN L, LAI S, et al. Improving attention-based handwritten mathematical expression recognition with scale augmentation and drop attention. International Conference on Frontiers of Handwriting Recognition. Dortmund. Germany. Sep. 2020. 175–180

[59] LIU C L, KIM I J, KIM J H. Model-based stroke extraction and matching for handwritten chinese character recognition. Pattern Recognition, 2001, **34** (12): 2339–2352

[60] LIU W, ANGUELOV D, ERHAN D, et al. SSD: Single shot multibox detector. European Conference on Computer Vision. Amsterdam. The Netherlands. Oct. 2016. 21–37

[61] MACLEAN S, LABAHN G. Elastic matching in linear time and constant space. International Workshop on Document Analysis Systems. Boston. USA. Jun. 2010

[62] MAHDAVI M. Query-driven global graph attention model for visual parsing: Recognizing handwritten and typeset math formulas. Ph.D. degree dissertation. Rochester Institute of Technology. 2020

[63] MAHDAVI M, ZANIBBI R, MOUCHÈRE H, et al. ICDAR 2019 CROHME + TFD: Competition on recognition of handwritten mathematical expressions and typeset formula detection. International Conference on Document Analysis and Recognition Recognition. Sydney. Australia. Sep. 2019. 1533–1538

[64] MALON C, UCHIDA S, SUZUKI M. Mathematical symbol recognition with support vector machines. Pattern Recognition Letters, 2008, **29** (9): 1326–1332

[65] MARCHIORI M. The mathematical semantic web. International Conference on Mathematical Knowledge Management. Bertinoro. Italy. Feb. 2003. 216–224

[66] MEDJKOUNE S, MOUCHERE H, PETITRENAUD S, et al. Combining speech and handwriting modalities for mathematical expression recognition. IEEE Transactions on Human-Machine Systems, 2017, **47** (2): 259–272

[67] MEUSCHKE N, SCHUBOTZ M, HAMBORG F, et al. Analyzing mathematical content to detect academic plagiarism. ACM Conference on Information and Knowledge Management. Singapore. Singapore. Nov. 2017. 2211–2214

[68] MILLER B R, YOUSSEF A. Technical aspects of the digital library of mathematical functions. Annals of Mathematics and Artificial Intelligence, 2003, **38** (1): 121–136

[69] MOGHADDAM R F, CHERIET M. A multi-scale framework for adaptive binarization of degraded document images. Pattern Recognition, 2010, **43** (6): 2186–2198

[70] MOUCHÈRE H, VIARD-GAUDIN C, ZANIBBI R, et al. ICFHR2016 CROHME: Competition on recognition of online handwritten mathematical expressions. International Conference on Frontiers in Handwriting Recognition. Shenzhen. China. Oct. 2016. 607–612

[71] MOUCHERE H, VIARD-GAUDIN C, KIM D H, et al. CROHME2011: Competition on recognition of online handwritten mathematical expressions. International Conference on Document Analysis and Recognition. Beijing. China. Sep. 2011. 1497–1500

[72] MOUCHERE H, VIARD-GAUDIN C, KIM D, et al. ICFHR 2012 competition on recognition of on-line mathematical expressions (CROHME 2012). International Conference on Frontiers in Handwriting Recognition. Bari. Italy. Sep. 2012. 811–816

[73] MOUCHÈRE H, VIARD-GAUDIN C, ZANIBBI R, et al. ICFHR 2014 competition on recognition of on-line handwritten mathematical expressions (CROHME 2014). International Conference on Frontiers in Handwriting Recognition. Heraklion. Greece. Sep. 2014. 791–796

[74] MOUCHERE H, VIARD-GAUDIN C, ZANIBBI R, et al. ICDAR 2013 CROHME: Third international competition on recognition of online handwritten mathematical expressions. International Conference on Document Analysis and Recognition. Washington. USA. aug. 2013. 1428–1432

[75] MOUCHÈRE H, ZANIBBI R, GARAIN U, et al. Advancing the state of the art for handwritten math recognition: the CROHME competitions, 2011–2014. International Journal on Document Analysis and Recognition, 2016, **19** (2): 173–189

[76] NAGOYA T, FUJIOKA H. Recovering dynamic stroke information of multi-stroke handwritten characters with complex patterns. International Conference on Frontiers in Handwriting Recognition. Bari. Italy. Sep. 2012. 722–727

[77] NAJAFI M H, SALEHI M E. A fast fault-tolerant architecture for sauvola local image thresholding algorithm using stochastic computing. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2016, **24** (2): 808–812

[78] NAKAYAMA Y. Mathematical formula editor for CAI. ACM SIGCHI Bulletin, 1989, **20** (SI): 387–392

[79] NGUYEN H D, LE A D, NAKAGAWA M. Deep neural networks for recognizing online handwritten mathematical symbols. IAPR Asian Conference on Pattern Recognition. Kuala Lumpur. Malaysia. Nov. 2015. 121–125

[80] NIBLACK W. An Introduction to Digital Image Processing. Englewood Cliffs: Prentice Hall, 1986

[81] NICOLAOU A, INGOLD R, LIWICKI M. Binarization with the local otsu filter. International Workshop on Graphics Recognition. Bethlehem. USA. Aug. 2014. 176–190

[82] NISHIDA H. An approach to integration of off-line and on-line recognition of handwriting. Pattern Recognition Letters, 1995, **16** (11): 1213–1219

[83] OHYAMA W, SUZUKI M, UCHIDA S. Detecting mathematical expressions in scientific document images using a u-net trained on a diverse dataset. IEEE Access, 2019, **7**: 144030–144042

[84] OKAMOTO M. Recognition of mathematical expressions by using the layout structures of symbols. International Conference on Document Analysis and Recognition. St. Malo. France. Oct. 1991. 242–250

[85] OTSU N. A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics, 1979, **9** (1): 62–66

[86] PERREAULT S, HEBERT P. Median filtering in constant time. IEEE Transactions on Image Processing, 2007, **16** (9): 2389–2394

[87] PHAN K M, LE A D, INDURKHYA B, et al. Augmented incremental recognition of online handwritten mathematical expressions. International Journal on Document Analysis and Recognition, 2018, **21** (4): 253–268

[88] PHANSALKAR N, MORE S, SABALE A, et al. Adaptive local thresholding for detection of nuclei in diversity stained cytology images. International Conference on Communications and Signal Processing. Calicut. India. Feb. 2011. 218–220

[89] PLAMONDON R, SRIHARI S N. Online and off-line handwriting recognition: a comprehensive survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, **22** (1): 63–84

[90] PORIKLI F. Constant time o(1) bilateral filtering. IEEE Conference on Computer Vision and Pattern Recognition. Anchorage. USA. Jun. 2008. 1–8

[91] PRATIKAKIS I, ZAGORI K, KADDAS P, et al. ICFHR 2018 competition on handwritten document image binarization (H-DIBCO 2018). International Conference on Frontiers in Handwriting Recognition. Niagara Falls. USA. Aug. 2018. 489–493

[92] RAIS N B, HANIF M S, TAJ I A. Adaptive thresholding technique for document image analysis. International Multitopic Conference. Lahore. Pakistan. Dec. 2004. 61–66

[93] REDMON J, FARHADI A. YOLOv3: An incremental improvement. https://arxiv.org/abs/1804.02767, 2018, 5, 21

[94] ROKACH L. Ensemble-based classifiers. Artificial Intelligence Review, 2009, **33** (1-2): 1–39

[95] RUSSELL G, PERRONE M P, MIN CHEE Y, et al. Handwritten document retrieval. International Workshop on Frontiers in Handwriting Recognition. Niagara on the Lake. Canada. Aug. 2002. 233–238

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

[97] SAXENA L P. Niblack's binarization method and its modifications to real-time applications: a review. Artificial Intelligence Review, 2019, **51** (4): 673–705

[98] SCHUBOTZ M, SCHARPF P, DUDHAT K, et al. Introducing MathQA: a math-aware question answering system. Information Discovery and Delivery, 2018, **46** (4): 214–224

[99] SEZGIN M, SANKUR B. Survey over image thresholding techniques and quantitative performance evaluation. Journal of Electronic Imaging, 2004, **13** (1): 146–165

[100] SHAFAIT F, KEYSERS D, BREUEL T M. Efficient implementation of local adaptive thresholding techniques using integral images. Document Recognition & Retrieval. San Jose. USA. Jan. 2008. 681510–681515

[101] SHI Y, LI H, SOONG F K. A unified framework for symbol segmentation and recognition of handwritten mathematical expressions. International Conference on Document Analysis and Recognition. Curitiba. Brazil. Sep. 2007. 854–858

[102] SIMISTIRA F, KATSOUROS V, CARAYANNIS G. A template matching distance for recognition of on-line mathematical symbols. International Conference on Frontiers in Handwriting Recognition. Montréal Canada. Aug. 2008

[103] SIMISTIRA F, KATSOUROS V, CARAYANNIS G. Recognition of online handwritten mathematical formulas using probabilistic SVMs and stochastic context free grammars. Pattern Recognition Letters, 2015, **53**: 85–92

[104] SMITHIES S, NOVINS K, ARVO J. A handwritting-based equation editor. Graphics Interface. Kingston. Canada. Jun. 1999. 84–91

[105] SOJKA P, LÍŠKA M. The art of mathematics retrieval. ACM Symposium on Document Engineering. Mountain View. USA. Sep. 2011. 57–60

[106] STALNAKER D, ZANIBBI R. Math expression retrieval using an inverted index over symbol pairs. Document Recognition and Retrieval. San Francisco. USA. Feb. 2015. 940207

[107] SU B, LU S, TAN C L. Robust document image binarization technique for degraded document images. IEEE Transactions on Image Processing, 2013, **22** (4): 1408–1417

[108] SU Y M, WANG J F. A novel stroke extraction method for chinese characters using gabor filters. Pattern Recognition, 2003, **36** (3): 635–647

[109] SU Z, CAO Z, WANG Y. Stroke extraction based on ambiguous zone detection: a preprocessing step to recover dynamic information from handwritten chinese characters. International Journal on Document Analysis and Recognition, 2009, **12** (2): 109–121

[110] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks. International Conference on Neural Information Processing Systems. Montreal. Canada. Dec. 2014. 3104–3112

[111] SUZUKI M, TAMARI F, FUKUDA R, et al. INFTY: an integrated OCR system for mathematical documents. ACM Symposium on Document Engineering. Grenoble. France. Nov. 2003. 95–104

[112] SUZUKI M, UCHIDA S, NOMURA A. A ground-truthed mathematical character and symbol image database. Technical Report of Ieice Prmu, 2005, **104**: 675–679

[113] TENSMEYER C, MARTINEZ T. Document image binarization with fully convolutional neural networks. International Conference on Document Analysis and Recognition. Kyoto. Japan. Nov. 2017. 99–104

[114] THANDA A, AGARWAL A, SINGLA K, et al. A document retrieval system for math queries. NTCIR Conference on Evaluation of Information Access Technologies. Tokyo. Japan. Jun. 2016. 346–353

[115] Truong T N, Nguyen C T, Phan K M, et al. Improvement of end-to-end offline handwritten mathematical expression recognition by weakly supervised learning. International Conference on Frontiers in Handwriting Recognition. Dortmund. Germany. Sep. 2020. 181–186

[116] VIARD-GAUDIN C, LALLICAN P M, KNERR S. Recognition-directed recovering of temporal information from handwriting images. Pattern Recognition Letters, 2005, **26** (16): 2537–2548

[117] WANG D H, YIN F, WU J W, et al. ICFHR 2020 competition on offline recognition and spotting of handwritten mathematical expressions - OffRaSHME. International Conference on Frontiers in Handwriting Recognition. Dortmund. Germany. Sep. 2020. 211–215

[118] WANG J, DU J, ZHANG J. Stroke constrained attention network for online handwritten mathematical expression recognition. https://arxiv.org/abs/2002.08670, 2018, 5, 21

[119] WANG P S P, ZHANG Y Y. A fast and flexible thinning algorithm. IEEE Transactions on Computers, 1989, **38** (5): 741–745

[120] WINKLER H J, LANG M. Online symbol segmentation and recognition in handwritten mathematical expressions. IEEE International Conference on Acoustics, Speech, and Signal Processing. Munich. Germany. April. 1997. 3377–3380

[121] WOLF C, JOLION J M, CHASSAING F. Text localization, enhancement and binarization in multimedia documents. International Conference on Pattern Recognition. Quebec City. Canada. Aug. 2002. 1037–1040

[122] WU J W, YIN F, ZHANG Y M, et al. Image-to-markup generation via paired adversarial learning. Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Dublin. Ireland. Sep. 2018. 18–34

[123] WU J W, YIN F, ZHANG Y M, et al. Handwritten mathematical expression recognition via paired adversarial learning. International Journal of Computer Vision, 2020

[124] XIAO Z, XIAO Y. Security and privacy in cloud computing. IEEE Communications Surveys Tutorials, 2013, **15** (2): 843–859

[125] YAMAMOTO R, SAKO S, NISHIMOTO T, et al. On-Line Recognition of Handwritten Mathematical Expressions Based on Stroke-Based Stochastic Context-Free Grammar. International Workshop on Frontiers in Handwriting Recognition. La Baule. France. Oct. 2006. 249–254

[126] YE Q, DOERMANN D. Text detection and recognition in imagery: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, **37** (7): 1480–1500

[127] YE X, CHERIET M, SUEN C Y. Stroke-model-based character extraction from gray-level document images. IEEE Transactions on Image Processing, 2001, **10** (8): 1152–1161

[128] ZANIBBI R, BLOSTEIN D, CORDY J. Recognizing mathematical expressions using tree transformation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, **24** (11): 1455–1467

[129] ZANIBBI R, AIZAWA A, KOHLHASE M, et al. NTCIR-12 MathIR task overview. NTCIR Conference on Evaluation of Information Access Technologies. Tokyo. Japan. Jun. 2016. 299–308

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

[131] ZANIBBI R, YU L. Math spotting: Retrieving math in technical documents using handwritten query images. International Conference on Document Analysis and Recognition. Beijing. China. Sep. 2011. 446–451

[132] ZENG J, FENG W, XIE L, et al. Cascade markov random fields for stroke extraction of chinese characters. Information Sciences, 2010, **180** (2): 301–311

[133] ZHANG J, DU J, DAI L. Multi-scale attention with dense encoder for handwritten mathematical expression recognition. International Conference on Pattern Recognition. Beijing. China. Aug. 2018. 2245–2250

[134] ZHANG J, DU J, DAI L. Track, attend, and parse (TAP): An end-to-end framework for online handwritten mathematical expression recognition. IEEE Transactions on Multimedia, 2019, **21** (1): 221–233

[135] ZHANG J, DU J, DAI L. A GRU-based encoder-decoder approach with attention for online handwritten mathematical expression recognition. International Conference on Document Analysis and Recognition. Kyoto. Japan. Nov. 2017. 902–907

[136] ZHANG J, DU J, YANG Y, et al. SRD: A tree structure based decoder for online handwritten mathematical expression recognition. IEEE Transactions on Multimedia, 2020

[137] ZHANG J, DU J, ZHANG S, et al. Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition. Pattern Recognition, 2017, **71**: 196–206

[138] ZHANG T, MOUCHÈRE H, VIARD-GAUDIN C. A tree-BLSTM-based recognition system for online handwritten mathematical expressions. Neural Computing and Applications, 2018, **32** (9): 4689–4708

[139] ZHANG T, SUEN C. Fast parallel algorithm for thinning digital patterns. Communications of the ACM, 1984, **27** (3): 236–239

[140] ZHANG X, YIN F, ZHANG Y, et al. Drawing and recognizing chinese characters with recurrent neural network. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, **40** (4): 849–862

[141] ZHANG X, SONG J, DAI G, et al. Extraction of line segments and circular arcs from freehand strokes based on segmental homogeneity features. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2006, **36** (2): 300–311

[142] ZHELEZNIAKOV D, CHERNEHA A, ZAYTSEV V, et al. Evaluating new requirements to pen-centric intelligent user interface based on end-to-end mathematical expressions recognition. International Conference on Intelligent User Interfaces. Cagliari. Italy. Mar. 2020. 212–220

[143] ZHELEZNIAKOV D, ZAYTSEV V, RADYVONENKO O. Acceleration of online recognition of 2d sequences using deep bidirectional LSTM and dynamic programming. International Work-Conference on Artificial Neural Networks. Gran Canaria. Spain. Jun. 2019. 438–449