一种鲁棒性的结构未知表格分析方法*

李星原[1] 高 文[1,2]

[1](哈尔滨工业大学计算机科学与工程系 哈尔滨 150001)

[2](中国科学院计算技术研究所 北京 100080)

摘要 模型未知表格的分析是表格识别中文本分析阶段的一个重要且具有挑战性的问题.目前的一般方法仅能容忍表格线的微小断线.文章提出一种基于抽取表格线的分析结构未知表格的策略.利用抽取的表格线的特征知识和局部约束可以选择一些有效边.在扫描水平和垂直表格线时,如果环绕边都有效,则产生一个矩形块,引入迭代可以更好地利用全局信息并使抽取结果满足约束关系.这种矩形块的抽取可以容忍表格线大的断线或不合适的分割,可以处理诸如嵌入矩形块的复杂结构.矩形块被抽取后,表格的其他部件可以通过搜索剩余的部分来抽取.表格测试实验证明,该方法在表格质量很差时仍可以很好地工作.

关键词 图像分析,文本分析,图像分割,表格分析,矩形提取,递归算法.

中图法分类号 TP391

正表格被广泛用于各种场合.对表格的自动阅读可以减轻人们将表格信息输入计算机的繁琐工作,具有重要的实用价值.最早研究普通纸张表格分析的是日立[1]和IBM公司[2].

对给定的一个表格图像,如果我们关于它的知识越多,对其分析就越容易.很多分析系统采用表格模型来表示一类表格的结构,依此来提高表格分析的正确率.表格模型的建立可以是手工的、半自动的和全自动的.手工建立模型非常繁琐.有时建立一个模型可能比直接输入表格还要费时.而且,手工模型对于表格重新打印时的内部位置调整无法对应,所以使用面窄.因此,面向半自动和自动表格模型建立的结构模型未知的表格分析就更有挑战性和重要意义.

同一般文本分析一样,目前表格图像分析所面临的主要问题是如何排除噪音的干扰问题.如果抽取的表格线是完全理想的,即没有断线和多余的线,则在仅由表格线构成的图像中跟踪它内部的矩形就可以得到全部的矩形块.但是,实际中总会抽取到一些来自字符或图的假表格线段,同时,真实表格线也可能在预处理后产生断线.

在此之前的研究已经提出了很多分析未知表格的方法.丁晓青和吴佑寿等人采用基于投影的方法检测表格线位置,然后用在相应位置寻找角的办法来抽取矩形块[3].Fan等人采用细化后再取特征点的方法[4],但细化会产生歧变,而且没有好方法把字符中和表格线上的特征点区分开来.Wang和Srihari先用连接块分析去掉一些孤立的字符,然后用行相邻图(line adjacent graph)检测表格线段,再从图像搜索关键点,由关键点搜索单元[5].Garris用相关run的方法检测表格单元的关键点[6].Belaid等人用Hough变换抽取表格线段,然后将其组成一个图,再在图上进行搜索[7].Yu和Jain用块相邻图抽取表格框架但不是单元[8].尽管Wang和Srihari的方法可以处理字符和表格线粘连的情况,但是,上述4种方法都没有考虑表格线断线的情况.Fujisawa和Nakano先对水平和垂直方向的run滤波,然后用表格线增强来填补小的断线,最后用轮廓跟踪来抽取矩形块的内环[1,9].它

* 本文研究得到国家自然科学基金、国家863高科技项目基金、国家教育部跨世纪人才基金和中国科学院"百人计划"基金资助.作者**李星原**,1964年生,博士,副教授,主要研究领域为文本分析与识别,机器学习,神经网络.**高文**,1956年生,博士,教授,博士生导师,主要研究领域为智能计算机接口,多媒体技术.

本文通讯联系人:高文,北京 100080,中国科学院计算技术研究所

本文1998-05-12收到原稿,1998-11-23收到修改稿

可容忍很小的断线但不适用于字符与表格线粘连或交叉的情况.Hori 和 Doermann 结合了从原图像和缩小后图像得来的轮廓[10].由于一些在原图像中跟踪不到的矩形块的内环会出现在缩小后的图像中,这种方法对小的断线效果较好,但是当断线超过相邻字符行距离的一半或表格线与相邻字符距离的一半时,连接断线的缩小比例会连接相邻行字符或者表格线与相邻字符,从而在缩小后的图像中丢失矩形块.在文献[10]中给出的结果证实,这种方法可以克服 0.32mm 的断线.这在很多情况下是不够的.Shinjo 等人概述了一个连接表格线交叉点的克服断线的方法[11],它需要先去掉非真实表格线上的线段而留下真实表格线上的线段,这是很难做到的.

由于考虑问题的基点不同,现有模型独立的表格分析方法都没有充分利用全局信息.这体现在两个方面:(1) 表格线抽取没有把表格线作为一个整体来对待;(2) 仅由局部的区域来抽取矩形块单元,当出现断线和假表格线段时,在原图像上局部的搜索就很难找到正确的结果.到目前为止,尚没有一种可以带噪音的表格线段抽取表格单元的算法.文献[10,12]甚至认为基于表格线的方法常常不被使用于带断线的表格.

作者在文献[13]中提出了一种利用全局信息的表格线抽取方法.本文在此基础上提出一个分析结构未知表格的方法,它包括矩形块抽取方法和其他单元抽取方法.矩形块抽取以抽取的表格线为导引,并参考原图像信息.为充分利用全局信息,我们先分析矩形块之间必须满足的一些约束条件,然后用一个基本抽取算法的迭代来抽取矩形块.

1　基本定义与约束关系

矩形块抽取问题的输入为一个水平表格线集合、一个垂直表格线集合和表格的原图像,输出为一个矩形块集合.一个矩形块表示为($PTopLeft$,$PTopRight$,$PBottomLeft$,$PBottomRight$,$LLeft$,$LRight$,$LTop$,$LBottom$),其中 $PTopLeft$,$PTopRight$,$PBottomLeft$,$PBottomRight$ 分别是矩形块的左上、右上、左下和右下角位置,用(x,y)表示,它们描述矩形块的绝对位置.$LLeft$,$LRight$,$LTop$,$LBottom$ 是包围该矩形块的左、右、上、下表格线的序号,描述矩形块的相对位置.

由于抽取的表格线可能有断线,也可能有多余的线或线段,因此,矩形块并不总是完全地被输入水平表格线集合和垂直表格线集合所封闭.另一方面,矩形块之间不是相互独立的,它们之间存在着很强的依赖关系,也就是说,被抽取的几个矩形块可能会决定另一个矩形块是否应该被抽取到.例如,在图 1(b)中,如果抽取了矩形块 1,3,4 和 5,则矩形块 2 也应该被抽取.因为既然矩形块 1,3,4 和 5 有效,这些矩形块的四边就都有效,则矩形块 2 的四边也有效.如果我们用抽取算法得到了矩形块 1,3,4 和 5,而没有抽取到矩形块 2,就是一个不合理的结果.

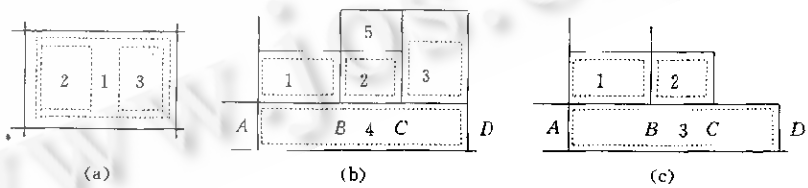


图1　矩形块和它们的边之间的约束

关于矩形块之间必须满足的基本约束,我们总结出以下 3 条约束关系.

约束 1. 如果一个矩形块完全被它内部的其他矩形块所覆盖,则无效.如图 1(a)中的矩形块 1.

约束 2. 如果一个矩形块的一条边完全被其他有效的矩形块所覆盖,则该边有效.如图 1(b)中,AD 有效,因为它被有效矩形 1,2,3(对应线段 AB,BC 和 CD)所覆盖.

约束 3. 如果一个矩形块的一条边部分地被其他有效矩形块所覆盖,而且未被覆盖的那一部分足够长,则该边有效.在图 1(c)中,AD 有效,因为矩形块 1(对应线段 AB)和矩形块 2(对应线段 BC)有效,而 CD 足够长.

2 矩形块抽取的基本算法

2.1 算法的复杂性

设有 P 条水平和 Q 条垂直表格线.它们总共可以产生 $C_P^2 \times C_Q^2 = P \times (P-1) \times Q \times (Q-1)/4$ 个矩形块候选.这些候选只有一小部分有效.原理上说,可以通过校验这些候选来确定输出的表格块集合.然而,由于候选数目是 $O(P^2Q^2)$,要穷尽这个候选集合比较费时.如果考虑每一个候选矩形块与其他每个矩形块的关系,那么校验过程的复杂度就变为 $O(P^4Q^4)$.但是,实际上一个表格内的最大可容纳矩形块数目远远小于矩形块候选的数目.如果 P 条水平表格线和 Q 条垂直表格线都足够长,它们可以把整个表格图像分割为 $(P-1)\times(Q-1)$ 个网格,而每个矩形块至少要占一个网格,所以最多只可能有 $(P-1)\times(Q-1)$ 个矩形块.因此,我们应该可以找到一种 $O(PQ)$ 的算法.

2.2 基本算法

矩形块抽取算法的过程要求一个从上到下对水平表格线的扫描和从左到右对垂直表格线的扫描.1条水平表格线和1条垂直表格线可能形成1个交点.在每个交点处,我们计算并保存可能作为后续矩形块顶边和左边的表格线区间.同时,如果前面已经有了一条顶边和一条左边,则判断对应的底边和右边是否足够长.如果4条边均有效,就产生一个矩形块.它的时间复杂度是 $O(PQ)$.从下面的描述中可以看到,它满足约束关系1.

算法的输入是水平(垂直)表格线集合,水平(垂直)表格线从上(左)到下(右)排序,起始序号大于等于0.设 i,j 分别是扫描过程中的当前水平表格线和垂直表格线.我们用一个整数数组 $TopEdge[Q]$ 来跟踪可用作顶边来构造一个矩形块的水平表格线.也就是说,$TopEdge[j]$ 是这样一条水平表格线的序号:它在当前水平表格线 i 之上且在垂直表格线 j 和 $j+1$ 之间具有足够的长度(即有效).这样的表格线可能有多条,我们只记录最靠近当前水平表格线 i 的那一条.$TopEdge[j]$ $(0 \leqslant j < Q)$ 初始化为 -1,表示开始时这样的顶边不存在.图2列出了在对一个表格的扫描过程中,在当前水平表格线 $i=3$ 和当前垂直表格线 $j=0$ 时的 $TopEdge[]$.当 $i=1, j=0$ 时,$TopEdge[0-2]=-1$,$TopEdge[3-7]=0$.

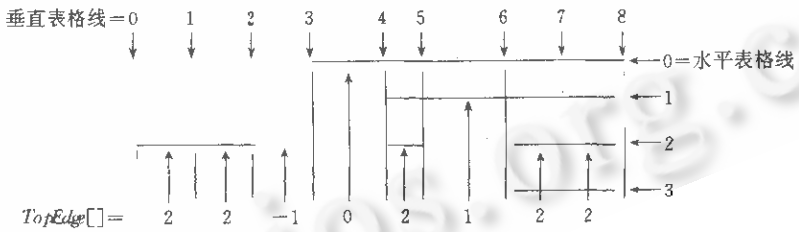


图2 一个表格结构在当前水平表格线 $i=3$ 和当前垂直表格线 $j=0$ 时的 $TopEdge[]$

算法用一个变量 vl 跟踪最近的可用来作为矩形块左边的垂直表格线.由于在每条水平表格线都对所有垂直表格线作一次扫描,我们用一个变量而不是像对顶边一样用一个数组.vl 在开始对每个水平表格线扫描时被初始化为 -1.对图2中的表格,当前水平表格线 $i=1$ 时,$vl=-1$,当 $j=0,1,2,3$;$vl=3$,当 $j=4$(正好完成对垂直表格线3的扫描之后);$vl=4$,当 $j=5$;$vl=5$,当 $j=6$;$vl=6$,当 $j=7$;$vl=7$,当 $j=8$.

下面,我们给出基本算法 RectangleExtract.其中 $IIEdgeValid(i,j1,j2)$ 用来计算是否水平表格线 i 在垂直表格线 $j1$ 和 $j2$ 之间有效,或者说,是否水平表格线 i 可以与垂直表格线 $j1$ 和 $j2$ 及另一条水平表格线构成一个矩形块.类似地,$VEdgeValid(j,i1,i2)$ 用来计算是否垂直表格线 j 在水平表格线 $i1$ 和 $i2$ 之间有效.

Procedure RectangleExtract

1. 初始化 $TopEdge[k]=-1(k=0,1,\ldots,Q)$.
2. **For** 每条水平表格线 $i(i=0,1,\ldots,P$,从上到下) {
3. $vl=-1$;
4. **For** 每条垂直表格线 $j(j=0,1,\ldots,Q$,从左到右) {

5. **If** $j1 = 0$ and $HEdgeValid(i, j-1, j)$,置 $TopEdge[j-1]=i$.

6. **If** $vl != -1$ and $TopEdge[j-1] != -1$ and $VEdgeValid(j, TopEdge[j-1], i)$ and $HEdgeValid(i, vl, j)$/* 校验右边和底边 */{

7. **If** 对 $vl \leqslant k < j$ 所有 $TopEdge[k]$相等,

产生一个由水平表格线 $TopEdge[vl]$和 i,垂直表格线 vl 和 j 所包围的矩形块;

else{

8. 设 H_{min}和 H_{max}分别为 $TopEdge[k]$在 $vl \leqslant k < j$ 的最小和最大值.

9. **For** h_1 从 H_{min}到 $H_{max}-1$,

if $HEdgeValid(h_1, vl, j)$ and $VEdgeValid(vl, h_1, i)$/* 校验顶边和左边 */{

10. 产生一个由水平表格线 h_1 和 i,垂直表格线 vl 和 j 所包围的矩形块;**break**.

}

}

11. **If** 在 Step7 或 Step10 中产生了一个矩形块

For $(k=vl; k<j; k++)$

$TopEdge[k]=i$;

/* 置该矩形块的底边的所有子区间为有效顶边 */

}

12. **If** $TopEdge[j] != -1$ and $VEdgeValid(j, TopEdge[j], i) == TRUE$,

置 $vl=j$;/* 校验左边 */

}

}

在 Step7,我们检查是否当 $vl \leqslant k < j$ 时所有 $TopEdge[k]$都相等.如果相等,水平表格线 $TopEdge[k]$在所有从 vl 到 $vl+1, \ldots$,从 k 到 $k+1, \ldots$,从 $j-1$ 到 j 的最小子区间都是有效的.我们约定边有效的判定条件满足:如果一条表格线在一个区间的每个最小子区间都有效,该表格线在该区间有效.事实上,这个约定是完全符合实际要求的.这样,$TopEdge[k]$在从 vl 到 j 的区间有效.因为 vl 在从 $TopEdge[vl]$到 i 的区间有效,而且在 Step6 中已经校验过右边和底边,由 $vl, j, TopEdge[vl]$和 i 所包围的矩形块有效.如果不是对 $vl \leqslant k < j$ 所有 $TopEdge[k]$都相等(如图 3 虚矩形框所示),则需要校验左边 vl.这是因为尽管 vl 在从 $TopEdge[vl]$到 i 的区间有效,$TopEdge[vl]$可能不是矩形块的顶边.类似地,我们需要校验顶边.

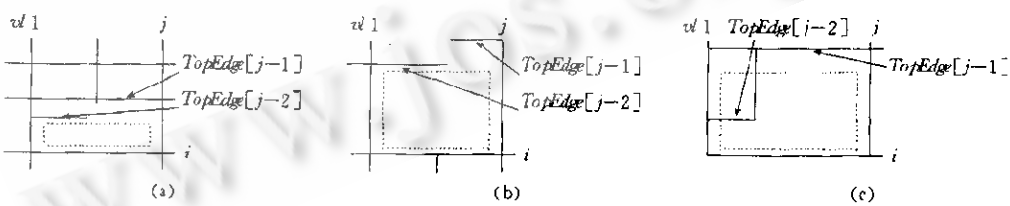


图 3　对于 $vl \leqslant k < j$,不是所有 $TopEdge[k]$都相等的情况

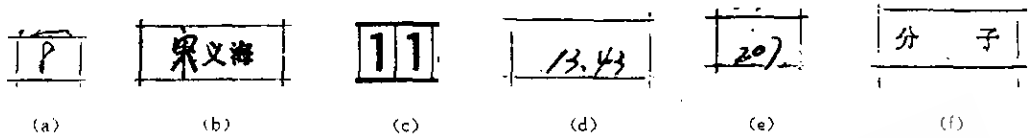
2.3　边的校验

判断一个矩形块是否有效,最基本的是要判断它的条边是否有效,也就是如何计算矩形块抽取基本算法中的 $HEdgeValid(i, j1, j2)$和 $VEdgeValid(j, i1, i2)$.本节只讨论 $VEdgeValid(j, i1, i2)$,水平边的校验与此类似.

我们采用一种基于规则的方法,先从区间内的表格线段和区间内及区间附近的原始图像中抽取一些特征,然后用一些规则判断该区间是否有效.

如果抽取到表格线覆盖全部区间,当然该区间有效.但是,在实际的表格中会碰到很多复杂的情况.图 4 是实际表格中的一些矩形块例子.在图 4(a)~(c)中,常会从字符中抽取到假的垂直表格线段.而图 4(d)~(f)中,每个矩形块都有比较短的边.甚至在一些例子中,来自字符的假表格线段占整个区间的比例比来自真表格线的

表格线段占整个区间的比例还要小.在这些情况下,仅凭区间被表格线所覆盖的长度来检验该区间的有效性就不够了.



(a)　(b)　(c)　(d)　(e)　(f)

图 4　从表格中收集的矩形块例子

我们组合以下特征来判断一个区间的有效性:(1) 区间的长度 *len*;(2) 区间被抽取到的表格线段所覆盖的长度 *llen*;(3) 区间内原图像的最大的连续空白(0)行数 *zlen*;(4) 区间内原图像垂直投影的峰值 *peak*;(5) 区间的两侧是否有类似于笔画的黑像素;(6) 这条垂直表格线的相邻上区间和相邻下区间是否有效,*uvalid*.

通常,一条垂直表格线段不会在它的两侧都有笔画与之相连,而来自字符的假表格线段则常常在它的一侧甚至两侧有笔画.我们把区间分为 3 类.第 1 类:两侧均无笔画;第 2 类:在一侧有笔画;第 3 类:两侧均有笔画.对不同的类型,采用不同的标准.也就是说,对第 1 类松,第 3 类紧.选用上下区间的有效性作为一个特征是受了盖式塔心理学的启发.如果它的上区间和/或下区间有效,且宽度相似,则人们总是倾向于把该区间看做有效.因此,在这种情况下,我们采用较松的要求.

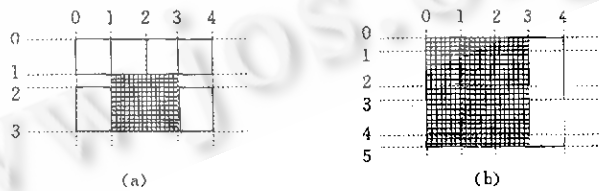
为了计算 *zlen*,*peak* 和确定该区间的两侧是否有笔画,将该区间及它周围稍作延伸的区域的原图像向水平和垂直方向投影.*zlen* 是区间在 Y 轴上投影的最大连续空白长度,而 *peak* 是区间在 X 轴上投影的峰值.在 X 轴上的投影中,如果在区间左侧(右侧)的邻域内有近似为 0 的投影,我们说在区间左侧(右侧)没有笔画,否则,说在它的左侧(右侧)有笔画.

基于以上的特征,我们采用一些规则来判断区间的有效性.

3　基本算法的扩展

3.1　组合 4 条边来确定矩形块的有效性

到目前为止,我们是独立地使用每条边的有效性来产生矩形块,这样做有时可能会产生错误.例如,在图 5(a)中,按照长度,垂直表格线 1 在水平表格线 1 和 3 之间的区间被确定为有效.同样地,垂直表格线 3 在水平表格线 1 和 3 之间的区间也被确定为有效.这样,由水平表格线 1 和 3 及垂直表格线 1 和 3 所包围的矩形块(如图 5(a)中阴影所示)就被确定为有效.



(a)　(b)

图5　表格线区间按照长度被确定为有效但有"严格白子区间"(EWS)

为避免这样的错误,对每条有效的边,我们计算它的"严格白子区间"(exact white interval,简称 EWS)的数目.所谓严格白子区间是指,表格线在它的上下(对垂直区间)或左右(对水平区间)边界上有像素,但是在其内部没有像素,也就是说,表格线正好覆盖它的两端但不覆盖它的内部.垂直(水平)区间的 EWS 可以通过搜索该区间的水平(垂直)投影得到.在图 5(a)中,垂直表格线 1 和 3 都有一个从水平表格线 1 到水平表格线 2 的 EWS.在图 5(b)中,垂直表格线 3 有 3 个 EWS.从水平表格线 0 到 1,从 2 到 3,从 4 到 5.然后我们去掉 4 条边的 EWS 之和大于等于 2 的矩形块.这样,在图 5(a)和(b)中阴影所示的矩形块就不会产生,因为它们分别有 2 和 3 个 EWS.

3.2　矩形块被其内部矩形块部分覆盖的情况

当一个矩形块被其内部矩形块部分覆盖,内部的矩形块把它分割为一个凹多边形和内部矩形本身.由于凹

多边形的表示比较复杂，我们用其内部矩形和最小外部矩形来表示.

只要内部矩形块与外部矩形块不共底边(如图 6(a),(b)所示)，用前面所述的基本算法就可以抽取到外部和内部矩形框.这是因为算法可以得到正确的上下左右边.否则，用基本算法就抽取不到外部矩形块，因为在抽取内矩形块的时候丢失了对外矩形块的正确左边 vl.

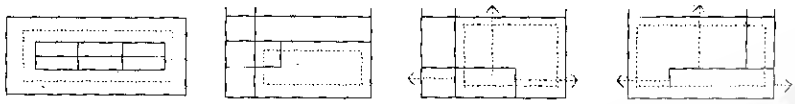


图6　一个矩形块被其内部矩形块部分覆盖

为了找到外部的矩形块，我们在基本矩形块抽取算法之后附加一个搜索过程.首先寻找可能的内部矩形块.一个矩形块可能是内部矩形块，只有当它相邻的两边(如图 6(b),(c)所示)或者相邻两边的相邻一段(图 6(d))不被其他矩形块所覆盖.然后对可能的内部矩形块，逐步向外搜索包围它的最小矩形块，直至找到一个矩形块或表格的边界为止，搜索方向如图 6(c)和图 6(d)的虚箭头所示.

4　迭代求精算法

由于表格矩形块之间的强依赖关系，很难通过一次扫描就把全部矩形块都抽取好.本节提出一个迭代算法，它是在前面基本算法的基础上进行迭代，以便有效地利用矩形块之间的关系.

对垂直区间进行校验时，我们使用了它的上下区间的有效性 $urValid$ 作为一个特征.由于基本算法是一个从上往下的扫描过程，校验一个垂直区间时已经得到了它上面的区间的有效性.下面区间的有效性的检验是以上面区间有效性检验的结果为前提的.为了检验全部矩形块的有效性，我们迭代地运行基本算法.在第 1 轮，假定每个垂直区间的下区间和每个水平区间的右区间无效，得到矩形块结果.然后从抽取的矩形块中导出水平和垂直表格线在每个最小区间的有效性.在下一次迭代时，当校验垂直(水平)区间时，使用刚导出的下(右)区间有效性.

我们用一个矩阵 $H=[h_{ij}]_{P\times(Q-1)}$ 记录水平表格线的最小区间的有效性.

$$h_{ij}=\begin{cases}1, & \text{第 } i \text{ 条水平表格线在垂直表格线 } j \text{ 和 } j+1 \text{ 之间有效;}\\ 0, & \text{否则.}\end{cases}$$

同样地，用矩阵 $V=[v_{ij}]_{Q\times(P-1)}$ 记录垂直表格线的最小区间的有效性.v_{ij} 的定义类似.

对抽取到的每个矩形块，对 $i=LTop,LBottom$ 和 $j\in[LLeft,LRight)$ 设置 $h_{ij}=1$，对 $j=LLeft,LRight$ 和 $i\in[LTop,LBottom)$ 设置 $v_{ji}=1$，就得到最小区间的有效性.

这样，$h_{ij}(0\leqslant i<P,0\leqslant j<Q-1)$ 和 $v_{ij}(0\leqslant i<Q,0\leqslant j<P-1)$ 初始时被设置为 0，在以后的迭代中设置某些 h_{ij} 和某些 v_{ij} 为 1.运行迭代过程直到抽取的矩形块集合与上次相同.

迭代过程的另一个重要作用是可以用来使算法保持第 1 节中所述的约束.

为了满足约束 2，我们在 $VEdgeValid(j,i1,i2)$ 中增加下面规则.

Rule 0. 如果对 $i1\leqslant i<i2,v_{ij}=1$，即在上次抽取的结果中垂直表格线 j 在 $i1$ 和 $i2$ 之间有效，则 $VEdgeValid(j,i1,i2)=1$.

为了满足约束 3，在 $VEdgeValid(j,i1,i2)$ 中增加下面规则.

Rule 1. 如果一个区间不满足其他规则的条件，但是$\exists\ i\in[i1,i2)$ 使得 $v_{ij}=1$，则把该区间中 $v_{ij}=1$ 的最小子区间去掉，然后将相邻的最小子区间合并，得到一个子区间集合.如果这些子区间全部有效，则该区间有效.

例如，在图 7(a)中，假定在上一轮中抽取了矩形块 1 和 2.对区间 AF 得到子区间$\{AB,DF\}$.如果 AB 和 DF 有效，则 AF 有效.

如果在每抽取一个矩形块之后都更新 v_{ij}，则在校验边时应用 Rule 0 和 Rule 1，不用迭代也可以使结果在很多情况下满足约束.但是，迭代仍然是必不可少的.在图 7(b)中，假定由于 DA 被设置为无效而矩形块 4 无效，但

是矩形块1,2,3有效(由于DA的两侧都有笔画,但DC,CB,BA的两侧均没有,这是可能的),我们需要第2轮来按Rule 0确认DA有效.

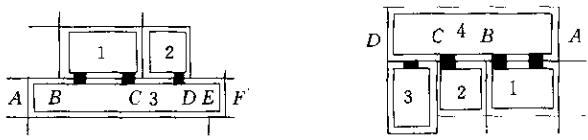


图7　利用迭代来使算法满足约束

迭代过程的另一个作用是可以处理一些关于表格结构的问题.由于关于表格结构的知识通常是用某些特殊的表格线来表示的,在应用这些知识前需确定这些特殊的表格线.迭代可以解决这个问题.首先在不应用表格结构知识的情况下抽取矩形块集合,然后确定这些特殊的表格线,再在以后的矩形块抽取中应用这些知识.

5　实验结果

到目前为止,尚没有可以用来测试表格分析方法有效性的有足够多种类型表格的图像数据库.美国国家标准与技术研究院(NIST)的Special Database 6(structured forms database 2)有5 595页图像,但是它只属于20种表格结构(12种表格,其中8种有两页).

如何衡量表格分析的精度目前尚没有一个统一的度量标准.本文定义两种计算精度的标准:单元正确率和边正确率.对于单元正确率,任何在抽取结果中被分裂、合并或者包含错误边界的一个单元都算作一个错误.对于边正确率,任何在单元抽取结果中被增加或者漏掉的一条边都算作一个错误.单元正确率反映抽取出的单元结果的可用程度,而边正确率则反映用人工来纠正抽取的错误需要付出多大的努力.

我们已经在一个大的表格图像集上对本文所提出的方法进行了测试.测试集包括属于120类表格的260张图像.大部分表格是从实际应用中收集的,很多图像的质量较差.一些是包括一个或多个表格的印刷文本,采集于国内、国际的报纸和杂志,其中包括几期计算机学报和几期IEEE Transactions on PAMI上的所有带表格的页.这些杂志没有被拆开,先复印后扫描.很多印刷文本是多栏的,一些带有双线.尽管表格只占这些印刷文本的一部分,在分析前我们没有人工框定表格的区域.我们选择这些印刷表格是由于有研究表明"大部分印刷体OCR系统对从多栏文本中分割表格有困难[14]".除此之外,我们也自己设计了一些表格,来测试算法对于各种结构和各种断线程度的适应性,其中一些是用手(不用尺)画出来的.260张测试表格中120张带有手写字符.对于在实际表格中填入模拟数据的样张,没有对书写者作出任何限制.图像至少扫描自6种不同的扫描仪,扫描分辨率从200dpi到600dpi不等.

对260张测试表格,在不提供任何关于表格类型的知识(包括不指示是否该表格是"规则表格")和不改变系统中任何参数的情况下,单元抽取正确率是98.1%,边抽取正确率是98.5%.对于一般的2000×3000大小的图像,在一台Pentium200M HZ PC上,从表格线抽取到单元分析整个时间约为2s左右.这个速度比其他方法要快,或至少是可与文献[6,9]的结果相比的.

6　结　论

本文提出了一种新的模型独立的表格单元分析方法.给出了一种矩形块抽取算法.它以抽取的表格线为基础,通过在扫描水平表格线和垂直表格线的过程中校验矩形块的边抽取矩形块.在清楚分析矩形块之间的约束关系之后,引入迭代来使抽取结果满足这些约束关系.迭代还进一步用来处理一些特殊情况和知识.校验边时兼顾表格线信息和原图像信息.通过增加一些搜索,系统可以处理很多复杂的情况.实验证明了我们的方法的有效性和对噪音的高容忍度.

参考文献

1 Nakano Y, Fujisawa H, Okada K *et al*. A document understanding system incorporating with character recognition. In: Proceedings of the 8th International Conference on Pattern Recognition. Washington D. C.: IEEE Computer Press, 1986. 801～803

2 Casey R G, Ferguson D R. Intelligent forms processing. IBM Systems Journal, 1990,29(3):435～450

3 Liu J, Ding X, Wu Y. Description and recognition of form and automated form data entry. In: Proceedings of the 3rd International Conference on Document Analysis and Recognition. Washington D. C.: IEEE Computer Press, 1995. 579～582

4 Fan K C, Lu J M, Wang J Y. A feature point approach to the segmentation of form documents. In: Proceedings of the 3rd International Conference on Document Analysis and Recognition. Washington D. C.: IEEE Computer Press, 1995. 623～626

5 Wang D, Srihari S N. Analysis of form images. In: Proceedings of the 1st International Conference on Document Analysis and Recognition. AFCET-IRISA/INRIA, Washington D. C.: IEEE Computer Press, 1991. 181～191

6 Garris M D. Correlated run length algorithm (CURL) for detecting form structure within digitized documents. In: Proceedings of the 3rd Annual Symposium of Document Analysis and Information Retrieval. Washington D. C.: IEEE Computer Press, 1994. 413～424

7 Belaid Y, Belaid A, Turolla E. Item searching in forms: application to french tax form. In: Proceedings of the 3rd International Conference on Document Analysis and Recognition. Washington D. C.: IEEE Computer Press, 1995. 744～747

8 Yu B, Jain A K. Generic A. System for form dropout. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1996,18(11):1127～1134

9 Fujisawa H, Nakano Y, Kurino K. Segmentation methods for character recognition: from segmentation to document structure analysis. Proceedings of the IEEE, 1992,80(7):1079～1092

10 Hori O, Doermann D S. Robust table-form structure analysis based on box-driven reasoning. In: Proceedings of the 3rd International Conference on Document Analysis and Recognition. Washington D. C.: IEEE Computer Press, 1995. 218～221

11 Shinjo H *et al*. A method for connecting disappeared junction patterns on frame lines in form documents. In: Proceedings of the 4th International Conference on Document Analysis and Recognition. Washington D. C.: IEEE Computer Press, 1997. 667～670

12 Watanabe T, Luo Q, Sugie N. Layout recognition of multi-kinds of table-form documents. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995,7(4):432～445

13 李星原.表格自动阅读研究[博士学位论文].哈尔滨工业大学,1997
(Li Xing-yuan. A study of automatic form reading [Ph.D. Thesis]. Harbin Institute of Technology, 1997)

14 Kanai J, Rice S V, Nartker T A *et al*. Automated evaluation of OCR zoning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995,17(1):86～90

A Robust Method for Unknown Structure Form Analysis

LI Xing-yuan[1]　GAO Wen[1,2]

[1](*Department of Computer Science and Engineering　Harbin Institute of Technology　Harbin*　150001)
[2](*Institute of Computing Technology　The Chinese Academy of Sciences　Beijing*　100080)

Abstract　The analysis of unknown forms is a challenging and important problem in document processing. Current methods can only tolerate small breaks in form lines. In this paper, a strategy is proposed for analyzing unknown structure and filled forms based on extracted lines. Individual edges are validated using knowledge of

features of the extracted lines and their local proximity. In a process of scanning the horizontal and vertical lines, candidate edges are validated and rectangles are generated if their surrounding edges and their combination are all valid. To preserve the constraints and make full use of global information, the process is recursively applied. The rectangle extraction can tolerate large breaks in form lines, ignore irrelevant segments and deal with complex configurations such as embedded rectangles. After rectangle extraction, other form components are extracted by searching the remaining segments. Experiments on a collection of forms with handwritten fields and documents with tables show that the proposed approach works well even on poor quality images.

Key words Image analysis, document analysis, image segmentation, form analysis, rectangle extraction, recursive algorithm.