



(12) 发明专利

(10) 授权公告号 CN 114529925 B

(45) 授权公告日 2022. 07. 15

(21) 申请号 202210425844.5

(22) 申请日 2022.04.22

(65) 同一申请的已公布的文献号
申请公布号 CN 114529925 A

(43) 申请公布日 2022.05.24

(73) 专利权人 华南理工大学
地址 510000 广东省广州市天河区五山路
381号
专利权人 人工智能与数字经济广东省实验
室(广州)

(72) 发明人 黄双萍 彭政华 杨帆

(74) 专利代理机构 东莞卓诚专利代理事务所
(普通合伙) 44754

专利代理师 朱鹏

(51) Int.Cl.
G06V 30/412 (2022.01)
G06V 10/24 (2022.01)
G06V 10/26 (2022.01)

G06V 10/28 (2022.01)

G06V 10/44 (2022.01)

G06V 10/764 (2022.01)

G06K 9/62 (2022.01)

(56) 对比文件

CN 106156761 A,2016.11.23

CN 106156761 A,2016.11.23

CN 112183038 A,2021.01.05

CN 111460927 A,2020.07.28

CN 111652204 A,2020.09.11

CN 1641632 A,2005.07.20

JP 2012059213 A,2012.03.22

JP S64561 A,1989.01.05

CN 111611783 A,2020.09.01

潘军.《复杂表格文档预处理与文本提取算
法研究》.《中国优秀博硕士学位论文全文数据库
(硕士) 信息科技辑》.2017,(第6期),I138-I313.

审查员 吴琼乐

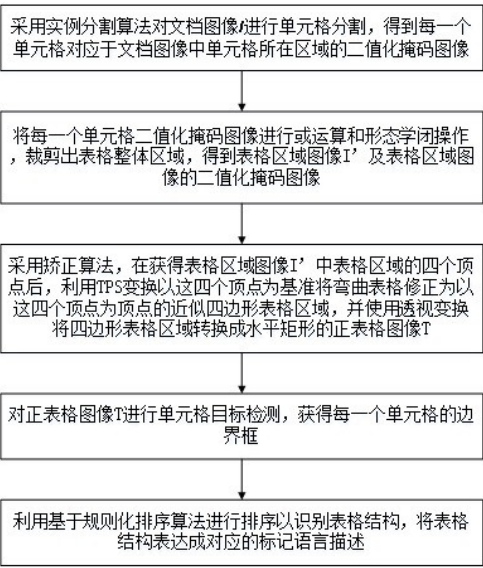
权利要求书5页 说明书12页 附图3页

(54) 发明名称

一种全线表表格结构识别方法

(57) 摘要

本发明公开了一种全线表表格结构识别方法,包括:采用实例分割算法对文档图像进行单元格分割,得到每一个单元格对应于文档图像中单元格所在区域的二值化掩码图像;将每一个单元格的二值化掩码图像进行或运算和形态学闭操作,得到表格整体区域图像及表格整体区域图像的二值化掩码图像;采用矫正算法,在获得表格整体区域图像中表格区域的四个顶点后,利用TPS变换以这四个顶点为基准将弯曲表格修正为以这四个顶点为顶点的近似四边形表格区域,并使用透视变换将近似四边形表格区域转换成水平矩形的正表格图像;进行单元格目标检测获得每一个单元格的边界框;利用基于规则化排序算法进行排序以识别表格结构,将表格结构表达成对应的标记语言描述。



1. 一种全线表表格结构识别方法,其特征在于,包括以下步骤:

步骤1,采用实例分割算法对文档图像I进行单元格分割,得到每一个单元格对应于文档图像中单元格所在区域的二值化掩码图像;

步骤2,将每一个单元格的二值化掩码图像进行或运算和形态学闭操作,裁剪出表格整体区域,得到表格整体区域图像I'及表格整体区域图像的二值化掩码图像;

步骤3,采用矫正算法,在获得表格整体区域图像I'中表格区域的四个顶点后,利用TPS变换以这四个顶点为基准将弯曲表格修正为以这四个顶点为顶点的近似四边形表格区域,并使用透视变换将近似四边形表格区域转换成水平矩形的正表格图像T;

步骤4,对正表格图像T进行单元格目标检测,获得每一个单元格的边界框;

步骤5,利用基于规则化排序算法进行排序以识别表格结构,将表格结构表达成对应的标记语言描述;

步骤5中,所述的基于规则化排序算法分别利用单元格边界框的左上角顶点和右下角顶点来代表单元格进行依行列排序和依列行排序,以获得每个单元格中的逻辑位置,求取每个单元格跨行跨列属性,以此识别表格结构;

所述的依行列排序是指:每次以一行的单元格为排序对象,先完成第一行的单元格从左到右的空间位置列排序,再完成第二行的单元格从左到右的空间位置列排序,直到最后一行,即从上至下依行进行每一行中单元格的列排序获取单元格在所在行中的列号,又因为是逐行进行的排序,第一次排序的一整行内的单元格的行号即为1,第二次排序的一整行内的单元格的行号即为2,以此类推,获得单元格在表格中代表逻辑位置的行列号;

所述的依列行排序是指:每次以一系列的单元格为排序对象,先完成第一列的单元格从上到下的空间位置行排序,再完成第二列的单元格从上到下的空间位置行排序,直到最后一列,即从左至右依列进行每一列中单元格的行排序获取单元格在所在列中的行号,又因为是逐列进行的排序,第一次排序的一整列内的单元格的列号即为1,第二次排序的一整列内的单元格的列号即为2,以此类推,获得单元格在表格中代表逻辑位置的行列号。

2. 根据权利要求1所述的一种全线表表格结构识别方法,其特征在于,步骤2包括以下步骤:

步骤201,将每个单元格对应于文档图像所在区域的二值化掩码图像集合之间进行或运算;

步骤202,进行形态学闭操作去掉或运算结果中的空隙部分;

步骤203,使用最小外接水平矩形裁剪出表格区域图像,即得到精准的表格整体区域图像I'及表格整体区域图像的二值化掩码图像。

3. 根据权利要求1所述的一种全线表表格结构识别方法,其特征在于,步骤3包括以下步骤:

步骤301,采用OpenCV中的轮廓求取算法,求得表格整体区域图像I'的二值化掩码图像中的表格掩码轮廓,然后获取表格掩码轮廓的最小外接矩形,从而获取表格的倾斜角度,在得到的倾斜角度的基础上,分别构造倾斜角度加上 45° 的两条直线与加上 135° 的两条直线共四条直线,分别位于表格掩码轮廓四个对角方向之外,将表格掩码轮廓中距离四条直线最近的点作为表格轮廓的四个顶点;

步骤302,分别在左上、右上顶点连线间以及左下、右下顶点连线间等距离取点,作为

TPS变换目标点,两两连接目标点,得到等距垂直直线系,直线系与表格轮廓的交点作为初始点且与目标点两两匹配,然后进行TPS变换,使初始点尽可能拉近目标点,得到以轮廓的四个顶点为顶点的去弯曲后的近似四边形表格图像;

步骤303,将去弯曲后的近似四边形表格图像中表格的四个顶点在图像中的坐标与图像边界的四个顶点坐标相结合计算透视变换矩阵参数,对去弯曲后的近似四边形表格图像进行透视变换,得到平铺满图片的正表格图像T;

透视变换操作如下:

$$\begin{cases} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \\ \begin{pmatrix} x'' \\ y'' \\ 1 \end{pmatrix} = \frac{1}{z'} * \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} \end{cases}$$

式中的 (x, y) 和 (x'', y'') 分别为变换前后的像素坐标,变换参数矩阵中的元素 a_{**} 由透视变换前后表格四个顶点的坐标联立方程解得, (x', y', z') 为假设处于 $z=1$ 平面内的原图上的点 $(x, y, 1)$ 乘以变换参数矩阵进行三维映射至三维空间后的坐标。

4. 根据权利要求1所述的一种全线表表格结构识别方法,其特征在于,步骤4具体包括以下步骤:利用CBNetV2算法或Faster RCNN算法对步骤3中的正表格图像T进行单元格目标检测,分别获得每一个单元格的边界框,边界框的形式为:

$$p = \{(p_{top-left}^x, p_{top-left}^y), (p_{lower-right}^x, p_{lower-right}^y)\}$$

即单元格边界框的左上角与右下角在正表格图像T中的空间位置坐标 $(p_{top-left}^x, p_{top-left}^y)$ 和 $(p_{lower-right}^x, p_{lower-right}^y)$,将检测结果中的每一个单元格边界框的左上角组成左上角点集,将每一个单元格边界框的右下角组成右下角点集。

5. 根据权利要求1所述的一种全线表表格结构识别方法,其特征在于,步骤5中,所述的分别利用单元格边界框的左上角顶点和右下角顶点来代表单元格进行依行列排序和依列行排序,包括以下步骤:

步骤501,利用单元格边界框的左上角顶点来代表单元格进行依行列排序,获得以边界框左上角顶点代表单元格的行号 $row_{top-left}(c_i)$, c_i 为单元格;

步骤502,利用单元格边界框的右下角顶点来代表单元格进行依行列排序,获得以右下角顶点代表单元格的行号 $row_{lower-right}(c_i)$,则单元格的跨行属性值为:

$$rowspan(c_i) = row_{lower-right}(c_i) - row_{top-left}(c_i) + 1;$$

步骤503,利用单元格边界框的左上角顶点来代表单元格进行依列行排序,获得以边界框左上角顶点代表单元格的列号 $col_{top-left}(c_i)$;

步骤504,利用单元格边界框的右下角顶点来代表单元格进行依列行排序,获得以边界框右下角顶点代表单元格的列号 $col_{lower-right}(c_i)$,则单元格的跨列属性值为:

$$colspan(c_i) = col_{lower-right}(c_i) - col_{top-left}(c_i) + 1;$$

步骤505,利用每个单元格在表格中的逻辑位置、跨行跨列属性,依据HTML语法结合即得到描述文档图像中表格结构的HTML格式字符串。

6.根据权利要求5所述的一种全线表表格结构识别方法,其特征在于,步骤5中,利用单元格边界框的左上角顶点来代表单元格进行依行列排序的过程,包括以下步骤:

步骤50101,在每一行的排序开始前需要进行初始化,首先需要获取进行排序的一整行顶点中位置最高的顶点,在数字图像坐标系中,原点位于图像左上角且纵轴y轴正方向向下,横轴x轴正方向向左,因此获取一行顶点中位置最高的点,相当于获取y轴分量即 $p_{top-left}^y$ 最小的点,将该点作为点 P_{top} 与点 P_t ,其中 P_{top} 为这一行顶点中位置最高顶点, P_t 为算法最近寻找到的顶点,若某一行中存在具有相同最小y轴分量的顶点不唯一的情况,则在具有相同最小y轴分量的顶点中随机取点作为点 P_{top} 与点 P_t ;

步骤50102,然后向右寻找与 P_t 同处一行的相邻后继顶点:首先在左上角点集中获取所有满足下式条件的顶点:

$$\begin{cases} |p_t^y - p_n^y| < 0.4 * \min\{b_h\} \\ -90^\circ < \arg \overrightarrow{p_t p_n} < 18^\circ \end{cases}$$

其中点 p_t^y 为算法最近寻找到的顶点的y轴分量, p_n^y 为左上角点集中剩余顶点的y轴分量, b_h 为单元格边界框的高, $\arg \overrightarrow{p_t p_n}$ 为向量 $\overrightarrow{p_t p_n}$ 与横坐标轴正方向的夹角;

步骤50103,在获取到的满足条件的所有顶点中选取距离点 P_t 最近的顶点,将这一顶点更新为点 P_t ;

步骤50104,重复步骤50102和步骤50103,直到这一行 P_{top} 右侧的后继顶点全部找齐;

步骤50105,将 P_{top} 重新更新为 P_t ,开始依据同理的方法寻找 P_{top} 左侧的前继顶点,此时顶点的筛选条件为:

$$\begin{cases} |p_t^y - p_n^y| < 0.4 * \min\{b_h\} \\ -90^\circ < \arg \overrightarrow{p_n p_t} < 18^\circ \end{cases}$$

直至 P_{top} 所在行的顶点全部找齐,获得每一个单元格在该行中代表先后关系的列号,该行排序结束,与此同时,得到该行内的单元格对应的行号;

步骤50106,在左上角点集中去除已经排序过的顶点,对左上角点集中剩余的顶点都进行上述步骤50101至步骤50105的排序过程,直到左上角点集为空,则完成每个单元格的行排序与列排序,获得每一个单元格在表格中的逻辑位置;

利用单元格边界框的右下角顶点来代表单元格进行依行列排序的过程与利用单元格边界框的左上角顶点来代表单元格进行依行列排序的过程相似;

利用单元格边界框的左上角顶点来代表单元格进行依行列排序的过程,包括以下步骤:

步骤50301,在每一列的排序开始前需要进行初始化,首先需要获取进行排序的一整列顶点中位置最靠左的顶点,在数字图像坐标系中,原点位于图像左上角且纵轴y轴正方向向下,横轴x轴正方向向左,因此获取一列顶点中位置最靠左的顶点,相当于获取x轴分量即 $P_{top-left}^x$ 最小的顶点,将该顶点作为点 $P_{left-most}$ 与点 P_t ,其中 $P_{left-most}$ 为这一列顶点中位置最靠左的顶点, P_t 为算法最近寻找到的顶点,若某一列中存在具有相同最小x轴分量的顶点不唯一的情况,则在具有相同最小x轴分量的顶点中随机取点作为点 $P_{left-most}$ 与点 P_t ;

步骤50302,然后向下寻找与 P_t 同处一列的相邻下侧顶点:首先在左上角点集中获取所有满足下式条件的顶点:

$$\begin{cases} |p_t^x - p_n^x| < 0.4 * \min\{b_w\} \\ 72^\circ < \arg \overrightarrow{p_t p_n} < 180^\circ \end{cases}$$

其中点 p_t^x 为算法最近寻找到的顶点的x轴分量, p_n^x 为左上角点集中剩余顶点的x轴分量, b_w 为单元格边界框的宽, $\arg \overrightarrow{p_t p_n}$ 为向量 $\overrightarrow{p_t p_n}$ 与横坐标轴正方向的夹角;

步骤50303,在获取到的满足条件的所有顶点中选取距离点 P_t 最近的顶点,将这一顶

点更新为点 P_t ;

步骤50304,重复步骤50302和步骤50303,直到这一列点中处于 $P_{left-most}$ 下侧的顶点全部找齐;

步骤50305,将 $P_{left-most}$ 重新更新为 P_t ,开始依据同理的方法寻找 $P_{left-most}$ 上侧的顶点,此时顶点的筛选条件为:

$$\begin{cases} |p_t^x - p_n^x| < 0.4 * \min\{b_w\} \\ 72^\circ < \arg \overrightarrow{p_n p_t} < 180^\circ \end{cases}$$

直至 $P_{left-most}$ 所在列的顶点全部找齐,获得每一个单元格在该列中代表上下位置关系的行号,该列点的上下位置行排序结束,与此同时,得到该列内的单元格对应的列号;

步骤50306,在左上角点集中去除已经排序过的顶点,对左上角点集中剩余的顶点都进行上述步骤50301至步骤50305的排序过程,直到左上角点集为空,则完成每个单元格的依列行排序,获得每一个单元格在表格中的逻辑位置;

利用单元格边界框的右下角顶点来代表单元格进行依列行排序的过程与利用单元格边界框的左上角顶点来代表单元格进行依列行排序的过程相似。

7. 根据权利要求1所述的一种全线表表格结构识别方法,其特征在于,步骤5中,所述的实例分割算法采用CBNetV2实例分割算法或CascadeMaskRCNN实例分割算法。

一种全线表表格结构识别方法

技术领域

[0001] 本发明属于图像处理与人工智能技术领域,特别是涉及一种全线表表格结构识别方法。

背景技术

[0002] 表格,不仅仅是一种可视化的交流模式,也是一种组织整理数据的方式。在过去几十年里,随着信息技术、电子计算机科技的快速发展,智能手机、扫描仪等电子设备得到了广泛的应用,越来越多的客户通过这些设备对文档进行拍照、扫描和上传,使得数字文档的生产和存储呈指数级增长。因此,对于提取这类非结构化文档图像(如零售收据、保险索赔表格和金融发票等)中的信息的需求正变得越来越迫切。从这些文档中提取信息的一个关键任务是提取表格的结构信息,但是,依赖人工从如此大规模且多样化的文档图像库中准确地提取表格结构信息是不现实的。具体地,表格文档图像手工处理,劳动力成本高,数据处理时间长等。随着计算机技术的发展,研究如何利用自动化技术加速表格文档图像的信息抽取,以便更好地提取文档图像中的表格信息,具有十分重大的现实意义。通常,这些文档包含存储在表格中的数据,不仅仅表格在布局和视觉外观上有多种变化,而且除了表格之外,文档图像中还有许多其他页面对象(如文本和图形),这些页面对象在外观上也有很大差异。表格结构的变化,用于可视化分离表格组件的图形元素的变化,以及在文档图像中存在的大量残缺、破损、遮挡等噪声,特别是由纸质文档等拍摄、扫描而得的表格图像在生成过程中由于纸张自然弯曲、拍摄角度等造成的表格不规则形变,进一步加大了图像中提取表格结构信息难度,使得从这些表格文档图像中抽取信息成为一个非常具有挑战性的问题。

[0003] 得益于深度学习技术的发展,现有的基于深度学习的表格结构识别方法大体上可以分为三类。

[0004] 第一类,受分割和检测任务的启发,一些研究采用深度学习技术进行表格中单元格的检测或分割,并结合相应的后处理算法实现表格结构识别。虽然这类方法能在一定程度上解决单元格空间位置上的定位问题,但他们只解决了单元格的定位问题,而忽略了单元格在表格中的逻辑位置。

[0005] 第二类,从图像-序列模型角度来看,其他研究试图将表格图像直接转换为标记序列(如LaTeX或HTML)。此类模型虽然能够恢复出单元格的逻辑位置,但是模型没有显示地利用单元格之间、单元格与表格整体的空间位置关系信息,从而限制了此类模型的性能。

[0006] 第三类,考虑到不同单元格逻辑位置之间的关系,一些研究开始使用图神经网络来建模不同单元格之间的关系,把结构识别问题转换为图神经网络中边的分类或节点的分类问题。这类算法在构建图网络之前往往需要采用文本检测或者关键点检测等单元格检测网络得到表格中单元格的空間位置等信息,而非实现端到端,并且易受到检测网络性能的影响。

[0007] 除此之外,现有技术中对表格结构的识别方法较为复杂,且极度依赖于表格形状

的规整程度,因为拍摄对象本身的不平整、拍摄角度以及光照等因素造成的弯曲、倾斜等不规则形状的表格文档图像的结构识别效果不佳。

发明内容

[0008] 有鉴于此,有必要针对上述技术问题,提供一种简单有效的基于规则的表格结构识别方法,所述方法直接通过单元格空间位置上的排序确定单元格的逻辑位置进而计算单元格的跨行跨列属性,在一定程度上适应表格图像的不规则形变,另外所述方法还设计了针对不规则表格进行处理的表格矫正算法,能在一定程度上减小表格图像弯曲、倾斜等不规则形变,进而提高不规则形状表格的结构识别性能。

[0009] 本发明公开了一种全线表表格结构识别方法,包括以下步骤:

[0010] 步骤1,采用实例分割算法对文档图像I进行单元格分割,得到每一个单元格对应于文档图像中单元格所在区域的二值化掩码图像;

[0011] 步骤2,将每一个单元格的二值化掩码图像进行或运算和形态学闭操作,裁剪出表格整体区域,得到表格整体区域图像I'及表格整体区域图像的二值化掩码图像;

[0012] 步骤3,采用矫正算法,在获得表格整体区域图像I'中表格区域的四个顶点后,利用TPS变换以这四个顶点为基准将弯曲表格修正为以这四个顶点为顶点的近似四边形表格区域,并使用透视变换将近似四边形表格区域转换成水平矩形的正表格图像T;

[0013] 步骤4,对正表格图像T进行单元格目标检测,获得每一个单元格的边界框;

[0014] 步骤5,利用基于规则化排序算法进行排序以识别表格结构,将表格结构表达成对应的标记语言描述。

[0015] 具体地,所述的实例分割算法采用CBNetV2实例分割算法或CascadeMaskRCNN实例分割算法。

[0016] 具体地,步骤2具体包括以下步骤:

[0017] 步骤201,将每个单元格对应于文档图像所在区域的二值化掩码图像集合之间进行或运算;

[0018] 步骤202,进行形态学闭操作去掉或运算结果中的空隙部分;

[0019] 步骤203,使用最小外接水平矩形裁剪出表格区域图像,即得到精准的表格整体区域图像I'及表格整体区域图像的二值化掩码图像。

[0020] 具体地,步骤3包括以下步骤:

[0021] 步骤301,采用OpenCV中的轮廓求取算法,求得表格整体区域图像I'的二值化掩码图像中的表格掩码轮廓,然后获取表格掩码轮廓的最小外接矩形,从而获取表格的倾斜角度,在得到的倾斜角度的基础上,分别构造倾斜角度加上 45° 的两条直线与加上 135° 的两条直线共四条直线,分别位于表格掩码轮廓四个对角方向之外,将表格掩码轮廓中距离四条直线最近的点作为表格轮廓的四个顶点;

[0022] 步骤302,分别在左上、右上顶点连线间以及左下、右下顶点连线间等距离取点,作为TPS变换目标点,两两连接目标点,得到等距垂直直线系,直线系与表格轮廓的交点作为初始点且与目标点两两匹配,然后进行TPS变换,使初始点尽可能拉近目标点,得到以轮廓的四个顶点为顶点的去弯曲后的近似四边形表格图像;

[0023] 步骤303,将去弯曲后的近似四边形表格图像中表格的四个顶点在图像中的坐标

与图像边界的四个顶点坐标相结合计算透视变换矩阵参数,对去弯曲后的近似四边形表格图像进行透视变换,得到平铺满图片的正表格图像T。

[0024] 透视变换操作如下:

$$[0025] \quad \begin{cases} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \\ \begin{pmatrix} x'' \\ y'' \\ 1 \end{pmatrix} = \frac{1}{z'} * \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} \end{cases}$$

[0026] 式中的 (x, y) 和 (x'', y'') 分别为变换前后的像素坐标,变换参数矩阵中的元素 $a_{*,*}$ 由透视变换前后表格四个顶点的坐标联立方程解得, (x', y', z') 为假设处于 $z = 1$ 平面内的原图上的点 $(x, y, 1)$ 乘以变换参数矩阵进行三维映射至三维空间后的坐标;

[0027] 具体地,步骤4具体包括以下步骤:利用CBNetV2算法或Faster RCNN算法对步骤3中的正表格图像T进行单元格目标检测,分别获得每一个单元格的边界框,边界框的形式为:

$$[0028] \quad p = \{(p_{top-left}^x, p_{top-left}^y), (p_{lower-right}^x, p_{lower-right}^y)\}$$

[0029] 即单元格边界框的左上角与右下角在正表格图像T中的空间位置坐标

$(p_{top-left}^x, p_{top-left}^y)$ 和 $(p_{lower-right}^x, p_{lower-right}^y)$,将检测结果中的每一个单元格边界框的左上角组成左上角点集,将每一个单元格边界框的右下角组成右下角点集。

[0030] 具体地,步骤5中,所述的基于规则化排序算法分别利用单元格边界框的左上角顶点和右下角顶点来代表单元格进行依行列排序和依列行排序,以获得每个单元格中的逻辑位置,求取每个单元格跨行跨列属性,以此识别表格结构;

[0031] 所述的依行列排序是指:每次以一行的单元格为排序对象,先完成第一行的单元格从左到右的空间位置列排序,再完成第二行的单元格从左到右的空间位置列排序,直到最后一行,即从上至下依行进行每一行中单元格的列排序获取单元格在所在行中的列号,

又因为是逐行进行的排序,第一次排序的一整行内的单元格的行号即为1,第二次排序的一整行内的单元格的行号即为2,以此类推,获得单元格在表格中代表逻辑位置的行列号;

[0032] 所述的依列行排序是指:每次以一系列的单元格为排序对象,先完成第一列的单元格从上到下的空间位置行排序,再完成第二列的单元格从上到下的空间位置行排序,直到最后一列,即从左至右依列进行每一列中单元格的行排序获取单元格在所在列中的行号,又因为是逐列进行的排序,第一次排序的一整列内的单元格的列号即为1,第二次排序的一整列内的单元格的列号即为2,以此类推,获得单元格在表格中代表逻辑位置的行列号。

[0033] 更进一步地,所述的分别利用单元格边界框的左上角顶点和右下角顶点来代表单元格进行依行列排序和依列行排序,包括以下步骤:

[0034] 步骤501,利用单元格边界框的左上角顶点来代表单元格进行依行列排序,获得以边界框左上角顶点代表单元格的行号 $row_{top-left}(c_i)$, c_i 为单元格;

[0035] 步骤502,利用单元格边界框的右下角顶点来代表单元格进行依行列排序,获得以右下角顶点代表单元格的行号 $row_{lower-right}(c_i)$, 则单元格的跨行属性值为:

$$[0036] \quad rowspan(c_i) = row_{lower-right}(c_i) - row_{top-left}(c_i) + 1;$$

[0037] 步骤503,利用单元格边界框的左上角顶点来代表单元格进行依列行排序,获得以边界框左上角顶点代表单元格的列号 $col_{top-left}(c_i)$;

[0038] 步骤504,利用单元格边界框的右下角顶点来代表单元格进行依列行排序,获得以边界框右下角顶点代表单元格的列号 $col_{lower-right}(c_i)$, 则单元格的跨列属性值为:

$$[0039] \quad colspan(c_i) = col_{lower-right}(c_i) - col_{top-left}(c_i) + 1;$$

[0040] 步骤505,利用每个单元格在表格中的逻辑位置、跨行跨列属性,依据HTML语法结合即得到描述文档图像中表格结构的HTML格式字符串。

[0041] 更进一步地,利用单元格边界框的左上角顶点来代表单元格进行依行列排序的过程,包括以下步骤:

[0042] 步骤50101,在每一行的排序开始前需要进行初始化,首先需要获取进行排序的一整行顶点中位置最高的顶点,在数字图像坐标系中,原点位于图像左上角且纵轴y轴正方向向下,横轴x轴正方向向左,因此获取一行顶点中位置最高的点,相当于获取y轴分量即

$P_{top-left}^y$ 最小的点,将该点作为点 P_{top} 与点 P_t , 其中 P_{top} 为这一行顶点中位置最高顶点, P_t 为算法最近寻找到的顶点,若某一行中存在具有相同最小y轴分量的顶点不唯

一的情况,则在具有相同最小y轴分量的顶点中随机取点作为点 P_{top} 与点 P_t ;

[0043] 步骤50102,然后向右寻找与 P_t 同处一行的相邻后继顶点:首先在左上角点集中获取所有满足下式条件的顶点:

$$[0044] \quad \begin{cases} |p_t^y - p_n^y| < 0.4 * \min\{b_h\} \\ -90^\circ < \arg \overrightarrow{p_t p_n} < 18^\circ \end{cases}$$

[0045] 其中点 p_t^y 为算法最近寻找到的顶点的y轴分量, p_n^y 为左上角点集中剩余顶点的y轴分量, b_h 为单元格边界框的高, $\arg \overrightarrow{p_t p_n}$ 为向量 $\overrightarrow{p_t p_n}$ 与横坐标轴正方向的夹角;

[0046] 步骤50103,在获取到的满足条件的所有顶点中选取距离点 P_t 最近的顶点,将这一顶点更新为点 P_t ;

[0047] 步骤50104,重复步骤50102和步骤50103,直到这一行 P_{top} 右侧的后继顶点全部找齐;

[0048] 步骤50105,将 P_{top} 重新更新为 P_t ,开始依据同理的方法寻找 P_{top} 左侧的前继顶点,此时顶点的筛选条件为:

$$[0049] \quad \begin{cases} |p_t^y - p_n^y| < 0.4 * \min\{b_h\} \\ -90^\circ < \arg \overrightarrow{p_n p_t} < 18^\circ \end{cases}$$

[0050] 直至 P_{top} 所在行的顶点全部找齐,获得每一个单元格在该行中代表先后关系的列号,该行排序结束,与此同时,得到该行内的单元格对应的行号;

[0051] 步骤50106,在左上角点集中去除已经排序过的顶点,对左上角点集中剩余的顶点都进行上述步骤50101至步骤50105的排序过程,直到左上角点集为空,则完成每个单元格的行排序与列排序,获得每一个单元格在表格中的逻辑位置;

[0052] 利用单元格边界框的右下角顶点来代表单元格进行依行列排序的过程与利用单元格边界框的左上角顶点来代表单元格进行依行列排序的过程相似;

[0053] 利用单元格边界框的左上角顶点来代表单元格进行依列行排序的过程,包括以下步骤:

[0054] 步骤50301,在每一列的排序开始前需要进行初始化,首先需要获取进行排序的一

整列顶点中位置最靠左的顶点,在数字图像坐标系中,原点位于图像左上角且纵轴y轴正方向向下,横轴x轴正方向向左,因此获取一系列顶点中位置最靠左的顶点,相当于获取x轴分量即 $p_{top-left}^x$ 最小的顶点,将该顶点作为点 $P_{left-most}$ 与点 P_t ,其中 $P_{left-most}$ 为这一列顶点中位置最靠左的顶点, P_t 为算法最近寻找到的顶点,若某一系列中存在具有相同最小x轴分量的顶点不唯一的情况,则在具有相同最小x轴分量的顶点中随机取点作为点 $P_{left-most}$ 与点 P_t ;

[0055] 步骤50302,然后向下寻找与 P_t 同处一系列的相邻下侧顶点:首先在左上角点集中获取所有满足下式条件的顶点:

$$[0056] \quad \begin{cases} |p_t^x - p_n^x| < 0.4 * \min\{b_w\} \\ 72^\circ < \arg \overrightarrow{p_t p_n} < 180^\circ \end{cases}$$

[0057] 其中点 p_t^x 为算法最近寻找到的顶点的x轴分量, p_n^x 为左上角点集中剩余顶点的x轴分量, b_w 为单元格边界框的宽, $\arg \overrightarrow{p_t p_n}$ 为向量 $\overrightarrow{p_t p_n}$ 与横坐标轴正方向的夹角;

[0058] 步骤50303,在获取到的满足条件的所有顶点中选取距离点 P_t 最近的顶点,将这一顶点更新为点 P_t ;

[0059] 步骤50304,重复步骤50302和步骤50303,直到这一列点中处于 $P_{left-most}$ 下侧的顶点全部找齐;

[0060] 步骤50305,将 $P_{left-most}$ 重新更新为 P_t ,开始依据同理的方法寻找 $P_{left-most}$ 上侧的顶点,此时顶点的筛选条件为:

$$[0061] \quad \begin{cases} |p_t^x - p_n^x| < 0.4 * \min\{b_w\} \\ 72^\circ < \arg \overrightarrow{p_n p_t} < 180^\circ \end{cases}$$

[0062] 直至 $P_{left-most}$ 所在列的顶点全部找齐,获得每一个单元格在该列中代表上下位置关系的行号,该列点的上下位置行排序结束,与此同时,得到该列内的单元格对应的列

号;

[0063] 步骤50306,在左上角点集中去除已经排序过的顶点,对左上角点集中剩余的顶点都进行上述步骤50301至步骤50305的排序过程,直到左上角点集为空,则完成每个单元格的依列行排序,获得每一个单元格在表格中的逻辑位置;

[0064] 利用单元格边界框的右下角顶点来代表单元格进行依列行排序的过程与利用单元格边界框的左上角顶点来代表单元格进行依列行排序的过程相似。

[0065] 与现有技术相比,本发明的有益效果在于:本发明方法设计了表格矫正的算法,能够对存在弯曲、倾斜等不规则形变的表格文档图像进行矫正处理,且本发明提出的基于规则的单元格排序算法不依赖于水平和垂直的表格框线,在方法步骤中提及的点的寻找方法不要求同行单元格必须水平,方法本身具备更好的容忍度和适应性,能够适应一定范围弯曲程度的表格,进一步提高算法对不规则形状表格图像的表格结构识别能力。

附图说明

[0066] 图1示出了本发明实施方法的流程示意图;

[0067] 图2示出了本实施例表格结构识别对象的表格示例;

[0068] 图3示出了本实施例单元格实例分割结果示意图;

[0069] 图4示出了本实施例对单元格二值化掩码图像集合之间进行或运算结果示意图;

[0070] 图5示出了本实施例对单元格二值化掩码图像或运算结果进行形态学闭操作结果示意图;

[0071] 图6示出了本实施例进行表格裁剪得到的表格图像结果;

[0072] 图7示出了本实施例表格图像结果对应的二值化掩码图像结果;

[0073] 图8示出了本实施例进行表格矫正算法前的结构示意图。

[0074] 图9示出了本实施例进行表格矫正算法后的示意图。

[0075] 图10示出了本实施例进行表格矫正算法后的表格图像的结果;

[0076] 图11示出了本实施例对矫正后的表格图像进行透视变换的结果。

具体实施方式

[0077] 为使本发明实施例的目的、技术方案和优点更加清楚,下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0078] 为了引用和清楚起见,下文中使用的技术名词、简写或缩写总结解释如下:

[0079] 单元格的逻辑位置:某单元格在表格中的(第i行,第j列)。

[0080] 单元格的空間位置:某单元格在表格文档图像上的像素点的坐标范围。

[0081] TPS(Thin Plate Spline):薄板样条变换。

[0082] HTML(Hyper Text Markup Language):超文本标记语言。

[0083] 图1示出了本发明实施例的流程示意图。一种全线表表格结构识别方法,包括以下步骤:

[0084] 步骤1,采用实例分割算法对文档图像I进行单元格分割,得到每一个单元格对应

于文档图像中单元格所在区域的二值化掩码图像；

[0085] 步骤2,将每一个单元格的二值化掩码图像进行或运算和形态学闭操作,裁剪出表格整体区域,得到表格整体区域图像I'及对应的表格整体区域图像的二值化掩码图像；

[0086] 步骤3,采用矫正算法,在获得表格整体区域图像I'中表格区域的四个顶点后,利用TPS变换以这四个顶点为基准将弯曲表格修正为以这四个顶点为顶点的近似四边形表格区域,并使用透视变换将近似四边形表格区域转换成水平矩形的正表格图像T；

[0087] 步骤4,对正表格图像T进行单元格目标检测,获得每一个单元格的边界框；

[0088] 步骤5,利用基于规则化排序算法进行排序以识别表格结构,将表格结构表达成对应的标记语言描述。

[0089] 具体地,为了获得表格文档图像中的表格结构,本发明提出采用实例分割算法对文档图像进行单元格分割提取出表格区域;构建矫正算法对表格区域整体进行矫正;采用目标检测算法对矫正后的表格进行单元格检测,并利用基于规则化排序算法进行排序以识别表格结构,并将表格结构表达成对应的标记语言描述。在一个实施例中采用超文本标记语言(HTML)对表格进行描述。本实施例采用如下步骤进行发明方法的实施。

[0090] 一种全线条表格结构识别方法,包括以下步骤:

[0091] 步骤(1),将拍摄、扫描等方法获得的表格文档图像I作为输入,利用深度学习实例分割算法对I进行单元格分割,得到每一个单元格对应于原图中单元格所在区域的二值化掩码图像;

[0092] 步骤(2),将得到的每个单元格的二值化掩码图像进行或运算、形态学闭操作并裁剪出表格整体区域,即得到精准的表格区域图像I'及其二值化掩码图像;

[0093] 步骤(3),构建表格矫正算法,首先获取图像I'中表格区域的四个顶点,利用TPS变换以这四个顶点为基准将弯曲表格修正为以这四个顶点为顶点的近似四边形表格区域,并使用透视变换将四边形表格区域转换成水平矩形的正表格图像T;

[0094] 步骤(4),对正表格图像T进行单元格目标检测,获得每一个单元格的边界框,在获得单元格边界框的基础上利用规则化的排序算法进行表格结构识别,并将其表达成对应的HTML格式字符串。

[0095] 所述步骤(1)是利用实例分割算法对表格文档图像I进行单元格分割,得到每一个单元格对应于原图中单元格所在区域的二值化掩码图像,具体为:

[0096] (1-1)为了得到更加精准的表格区域,利用CBNetV2实例分割算法对表格文档图像I进行单元格分割,表格示例如图2,得到各个单元格对应于原图中单元格所在区域的二值化掩码图像集合,如图3;

[0097] 正常情况下每一个单元格都对应一张大小与原图大小一致的单元格掩码图像。

[0098] 所述步骤(2)是将得到的每个单元格的二值化掩码图像之间进行或运算、形态学闭操作并裁剪出表格整体区域,即得到精准的表格区域图像I'及其二值化掩码图像,具体为:

[0099] (2-1)将得到的每个单元格对应于原图所在区域的二值化掩码图像集合之间进行或运算;

[0100] (2-2)接着进行形态学闭操作去掉或运算结果中的“空隙”部分,如图4白色区域内部存在的黑色区域,结果如图5所示;

[0101] (2-3)使用最小外接水平矩形裁剪出表格区域图像即得到精准的表格区域图像I'及其二值化掩码图像,如图6和图7所示;

[0102] 所述步骤(3)是构建表格矫正算法,首先获取图像I'表格掩码图像中表格区域的轮廓及其四个顶点,利用TPS变换以四个顶点为基准将弯曲表格修正为以这四个顶点为顶点的近似四边形表格区域,并使用透视变换将四边形表格区域转换成水平矩形的正表格图像T,具体为:

[0103] (3-1)、采用OpenCV中的轮廓求取算法,求得表格图像I'的二值化掩码图像中的表格轮廓,如图8中的黑色表格轮廓,然后获取表格掩码轮廓的最小外接矩形(非水平的最小外接矩形),从而获取表格的倾斜角度。在倾斜角度的基础上,分别构造倾斜角度加上45°的两条直线与加上135°的两条直线共四条直线,如图7中灰色直线,分别位于表格掩码四个对角方向之外,将表格掩码轮廓中距离四条直线最近的点作为表格轮廓的四个顶点,如图8中黑色实心圆点;

[0104] (3-2)、接着分别在左上、右上顶点连线间以及左下、右下顶点连线间等距离取点,作为TPS变换目标点,如图8中黑色圆环,两两连接目标点,得到等距垂直直线系,如图8中灰色虚线,直线系与表格轮廓的交点作为初始点(图8中灰色虚线与黑色轮廓相交的灰色点)且与目标点两两匹配,然后进行TPS变换,使初始点尽可能拉近目标点,得到以轮廓的四个顶点为顶点的近似四边形表格区域图像,如图9所示为矫正后的表格轮廓图像;

[0105] (3-3)、将去弯曲过后的表格四个顶点坐标与表格平铺满图片的正表格图片的四个顶点坐标结合计算透视变换参数,对去弯曲表格图片进行透视变换,得到平铺满图片的正表格图像T。透视变换操作如下:

$$[0106] \quad \begin{cases} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \\ \begin{pmatrix} x'' \\ y'' \\ 1 \end{pmatrix} = \frac{1}{z'} * \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} \end{cases}$$

[0107] 式中的 (x, y) 和 (x'', y'') 分别为变换前后的像素坐标,变换参数矩阵中的元素 a_{*} 由透视变换前后表格四个顶点的坐标联立方程解得, (x', y', z') 为假设处于 $z = 1$ 平面内的原图上的点 $(x, y, 1)$ 乘以变换参数矩阵进行三维映射至三维空间后的

坐标。本实施例中透视变换前后结果如图10和图11所示。

[0108] 所述步骤(4)对正表格图像T进行单元格目标检测,获得每一个单元格的边界框,接着采用规则化的排序算法进行表格结构恢复,并将结果表达成相应的HTML格式字符串,具体为:

[0109] (4-1)利用CBNetV2算法对步骤(3)中得到的正表格图像T进行单元格目标检测,分别获得每一个单元格的边界框,边界框的形式为:

$$[0110] \quad p = \{(p_{top-left}^x, p_{top-left}^y), (p_{lower-right}^x, p_{lower-right}^y)\}$$

[0111] 即单元格边界框的左上角与右下角在正表格图像T中的空间位置坐标;

[0112] (4-2)将单元格边界框的空间位置坐标信息用于恢复表格的结构——单元格在表格中的逻辑位置(某单元格在表格中的第几行、第几列)以及单元格的跨行跨列属性;

[0113] 在表格的HTML语法中,矩形单元格的上边界处于同一水平直线内的单元格会被划分在同一行中,而与单元格的跨行跨列属性无关。

[0114] 受HTML等标记语言的表格描述语法启发,利用单元格边界框的左上角顶点

$(p_{top-left}^x, p_{top-left}^y)$ 与右下角顶点 $(p_{lower-right}^x, p_{lower-right}^y)$ 的空间位置坐标信息来代表单元格的空间位置信息进行单元格在空间上的排序以获得单元格在表格中的逻辑位置,这就将单元格排序简化为点集在二维空间上的空间位置排序。

[0115] 因此选择利用单元格边界框的左上角来代表单元格进行“依行列排序”,所谓“依行列排序”指每次以一行的单元格为排序对象,先完成第一行的单元格从左到右的空间位置列排序,再完成第二行的单元格从左到右的空间位置列排序,直到最后一行,即从上至下依行进行每一行中单元格的列排序获取单元格在所在行中的列号,又因为是逐行进行的排序,第一次排序的一整行内的单元格的行号即为1,第二次排序的一整行内的单元格的行号即为2,以此类推,获得单元格在表格中代表逻辑位置的行列号。“依行列排序”具体为:首先

获取点集中位置最高的点,即 $p_{top-left}^y$ 最小的顶点作为点 p_{top} 与点 p_t (在数字图像

中y轴向下,x轴向左;若一行中的所有点的y值相等,则随机取点; p_{top} 为每一行的最高

点, p_t 为算法最近寻找到的点);

[0116] 然后向右寻找与 p_t 同一行的相邻后继顶点。具体方法是首先在点集中获取所有满足下式的顶点:

$$[0117] \quad \begin{cases} |p_t^y - p_n^y| < 0.4 * \min\{b_h\} \\ -90^\circ < \arg \overrightarrow{p_t p_n} < 18^\circ \end{cases}$$

[0118] 其中点 p_t 为算法最近寻找到的点(此时为 p_{top}) , p_n 为点集中剩余顶点(算法未寻找到的顶点), b_h 为单元格边界框的高, $\arg \overrightarrow{p_t p_n}$ 为向量 $\overrightarrow{p_t p_n}$ 与横坐标轴的夹角;

[0119] 在获取到的满足上式的所有顶点中选取距离点 p_t 最近的顶点, 将这一顶点更新为点 p_t ;

[0120] 此时顶点的筛选条件为:

$$[0121] \quad \begin{cases} |p_t^y - p_n^y| < 0.4 * \min\{b_h\} \\ -90^\circ < \arg \overrightarrow{p_n p_t} < 18^\circ \end{cases}$$

[0122] 直至 p_{top} 所在行的顶点全部找齐, 获得每一个单元格在该行中代表先后关系的列号, 该行排序结束。与此同时, 也可以得到该行内的单元格对应的行号;

[0123] 接下来在点集中去除已经排序过的顶点, 对点集中剩余的顶点都进行上述的依行列排序;

[0124] 因为依行列排序是从上至下依行进行的, 每排序完一行就可以得到该行内的单元格对应的列号和行号, 所以当最后一行单元格的依行列排序完成时就相当于单元格的行排序与列排序也随即完成, 从而获得每一个单元格在表格中的逻辑位置。

[0125] (4-3) 逻辑位置推断完成后进行单元格跨行跨列属性求取。通过步骤(4-2)所述“依行列排序”算法, 每一个单元格都可得到以边界框左上角顶点代表单元格进行依行列排序所得的行号 $row_{top-left}(c_i)$ (c_i 为某一单元格)。

[0126] 然后以单元格边界框的右下角顶点来代表各单元格, 再次按照步骤(4-2)排序算法进行依行列排序, 得到以右下角顶点代表单元格进行排序的行号 $row_{lower-right}(c_i)$, 那么单元格的跨行属性值为:

$$[0127] \quad rowspan(c_i) = row_{lower-right}(c_i) - row_{top-left}(c_i) + 1.$$

[0128] 根据对称性, 同理也可根据单元格左上角顶点与右下角顶点来进行“依列行排序”来求出单元格的跨列属性, 所谓“依列行排序”指从左至右依列进行每一列中单元格上下位置的行排序获得单元格在所在列的行号;

[0129] 以单元格边界框左上角代表单元格进行依列行排序得到列号 $col_{top-left}(c_i)$,

然后以边界框右下角顶点代表单元格进行依列行排序得到列号 $col_{lower-right}(c_i)$,

则单元格的跨列属性值为:

[0130] (4-4)将单元格在表格中的逻辑位置、跨行跨列属性依据HTML语法结合即得到描述文档图像中表格结构的HTML格式字符串。

[0131] 关于本实施例的替代方案:步骤1+步骤2的设计是为了获得更精确的表格整体的掩码,另一种方法就是直接就行表格整体的分割,但在实验中发现直接进行表格整体的实例分割的边沿部分精确度不如先进行单元格实例分割再进行步骤2的高,而矫正算法依赖于边沿分割的精度,后续中若实现表格整体实例分割的高精度边沿也可以直接进行表格整体实例分割后接剩余算法部分。在步骤1中可以采用CBNetv2实例分割算法,也可以采用其他的实例分割算法例如Cascade Mask RCNN。在步骤3中寻找表格掩码四个顶点的方法也可以采用角点检测的方法来获取。在步骤4也可以采用其他目标检测算法,例如Faster RCNN。

[0132] 本领域普通技术人员可以意识到,结合本文中所公开的实施例描述的各示例的单元及算法步骤,能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本发明的范围。

[0133] 以上实施例的各技术特征可以进行任意的组合,为使描述简洁,未对上述实施例中的各个技术特征所有可能的组合都进行描述,然而,只要这些技术特征的组合不存在矛盾,都应当认为是本说明书记载的范围。

[0134] 本领域技术人员应明白,本申请的实施例可提供为方法、系统或计算机程序产品。因此,本申请可采用完全硬件实施例、完全软件实施例或结合软件和硬件方面的实施例的形式。而且,本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

[0135] 以上所述仅是本发明的优选实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明原理的前提下,还可以做出若干改进和润饰,这些改进和润饰也应视为本发明的保护范围。

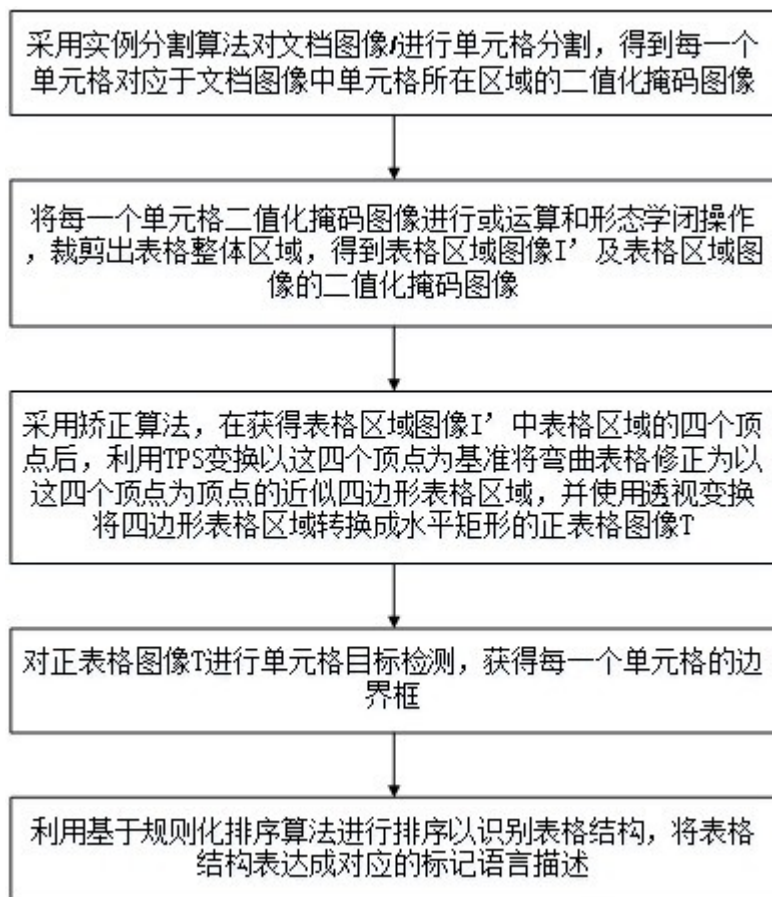


图1

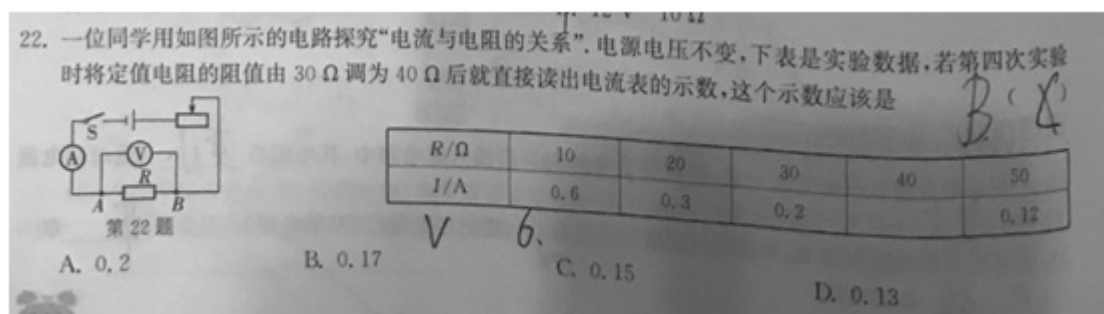


图2

—	—	—	—	—	—
单元格 (1,1)	单元格 (1,2)	单元格 (1,3)	单元格 (1,4)	单元格 (1,5)	单元格 (1,6)
—	—	—	—	—	—
单元格 (2,1)	单元格 (2,2)	单元格 (2,3)	单元格 (2,4)	单元格 (2,5)	单元格 (2,6)

图3



图4



图5

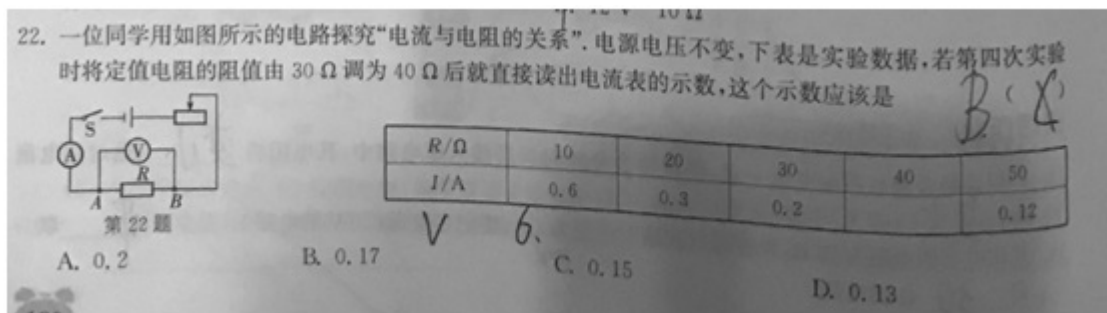


图6



图7

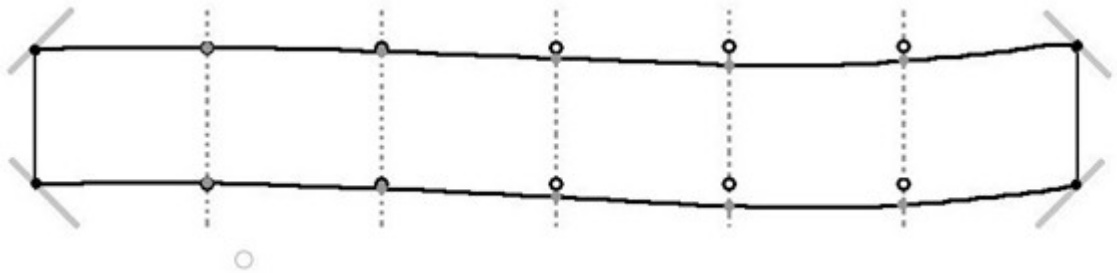


图8

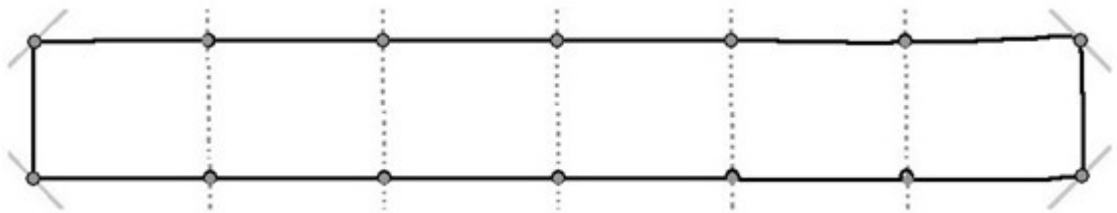


图9

R/Ω	10	20	30	40	50
I/A	0.6	0.3	0.2		0.12

图10

R/Ω	10	20	30	40	50
I/A	0.6	0.3	0.2		0.12

图11