摘要 - 本文档介绍了一种在图表元素之间进行逻辑链接结构识别的方法。 应用的方法直观地模仿了人类的识别方式，它依赖于将已经发现的连接器合并为更复杂的连接器。 这个过程由我们的方法建模，其中简单明了的连接器逐渐扩展到更复杂的结构。 每次迭代都可能导致修改到目前为止获得的连接器集。 修改由描述连接器结构应满足的逻辑和图形约束的规则集管理。 如果扩展导致违反规则定义的约束，则不执行修改。 通过这种方式，公认的图表结构与假定的原则一致。 该方法使用来自三个域的一组图表进行实验验证。 总之，讨论了方法的优点和缺点。

1介绍

在过去几年中，对类似内容图像检索有很大兴趣。这个领域有很多研究。 [1]中提出了深入的调查。类似的内容图像检索在自动图像注释，故事插图，复制检测，网络图像搜索和艺术图像分析中非常有用。如果它们包含图像，它也可以帮助搜索类似的文本文档。通常，图像说明了文档的内容。它们包含浓缩形式的文档中包含的信息，并且对表达的想法进行明确比较的问题较少。这解释了其对类似专利检索应用的浓厚兴趣，以加快专利申请程序和保护知识产权。对于自动查询，有必要将图像中的信息转换为高级描述。通常，在技术文档的情况下，图像表示以图表形式显示的工程图，图表，算法，操作和过程。这种图像呈现由某些元素和它们之间的连接组成的结构。自动识别连接结构是进一步自动分析或甚至机器理解图像的第一步。问题很重要且具有挑战性，因为文档具有高度复杂的结构，嵌入了表格和图形信息，并且它们包含相互矛盾的技术术语。处理嵌入的图像可以帮助解决问题。通常，为了应用这种方法，有必要在整个文档中找到图像。然后，处理图像以将它们分类为各种类：图表，图表，方案，流程图，图表和照片。接下来，应用专用于给定类图像的方法，以便识别特定元素及其互连。值得注意的是，这种图像解释允许以电子形式写入图像的内容，这便于其存储和进一步处理和比较。在我们的研究中，我们关注图表和流程图中元素的连通性。专用于这种图形的方法必须找到：a）所示元素的类型（各种描绘的形状），b）不属于找到的形状的线段，以及c）由这些段创建的连接。由于图表通常包含嵌入图表元素中的文本，因此还需要检测文本区域，通过应用OCR技术识别它，最后将识别的文本分配给图形图元素。本文描述的研究目的是找到一种能够检索图中描述的元素之间的逻辑链接的方法。然后，该逻辑结构在XML文件中表示。当我们考虑在两个以上的元件之间可能存在一个连接并且构成连接器的线段可以相交时，这是一项艰巨的任务。

本文由六部分组成。 下一节将介绍相关工作。 第三节提出要解决的问题。后面的部分介绍了所开发的方法。 第五节通过实验验证了我们的方法。 最后，介绍了与进一步工作有关的一些结论和建议

2相关工作

在[2]中描述了该领域作品的早期调查。 作者写道，图表识别面临许多挑战，包括图解符号的多样性，以及识别过程中噪声和模糊性的存在。 尽管从本出版物发布的那一年开始流逝，但所有上述特征图表解释的特征现在不断引起图表识别问题。

该论文[3]回顾了过去十年的研究。 作者介绍了图表识别的整个过程：图表分割，图表分类，图表解释和讨论现有解决方案。

相当多的作品致力于在线流程图识别。在[4]中，给出了用于标记流程图的每个笔划并根据它们所属的符号对笔划进行分组的分析。文献[5]中描述了相同的研究领域。在本文中，对输入的合适解释的搜索被公式化为包含最大和问题的组合优化任务。识别管道包括两个主要阶段。首先，对可能表示草图符号（符号候选）的笔划组进行分段，并检测它们之间的关系。其次，通过求解优化问题来选择最适合输入的符号候选的组合。这项工作[6]也涉及在线图表，但重点是使用2D动态编程的手绘电路图识别。论文[7]提出了另一种手绘组织图的方法，该方法基于贝叶斯条件随机场（BCRF），它共同分析所有绘图元素，以便结合上下文线索。每个对象的分类会影响其邻居的分类。 BCRF允许灵活和相关的功能。图的在线识别主要用于自动检查学生考试。

目前，在专利检索的背景下对流程图识别非常感兴趣。 文献[8]描述了在专利相关用例的背景下评估流程图识别方法的有效性的措施。 方法调查可以在[9]中找到。 本文介绍了一种半自动图表地面实况生成系统[10]。 使用该系统，用户能够提取多级地面实况数据。

一些作品致力于文档中的图表识别。他们应用各种分类方法。在[11]中使用了尖峰神经网络。论文[12]提出了一种用于识别大型工程图纸的系统，其特征在于符号和连接线的交替实例。所考虑的图像类包括诸如以下的域：流程图，逻辑和电路以及化学工厂图。系统的输出包括标识符号类型和互连的列表。它可以用于设计模拟或作为附图的紧凑便携式表示。该方法包括两个步骤。首先，域独立规则用于分割图形图像中连接线的符号，这些符号已经过常规方式的细化，矢量化和预处理。然后，绘图理解子系统与一组特定于域的匹配器一起工作，以对符号进行分类并自动纠正错误。他们还提出了一个交互式纠正残留错误的界面。

图识别中的另一个重要问题及其自动解释是识别出现在图表上的文本。 文本和图形层的分离通过减少大量涉及的图形元素简化了进一步的图表结构分析。 它还可以将文本信息属性附加到图表的检测图形元素。 在我们的方法中，我们使用了[13]中描述的文本分离方法。 该方法包括三个阶段。 在第一阶段，基于连通分量分析和连通分量集群的一些简单几何属性引出文本区域候选。 在第二阶段，将模式识别方法应用于该组候选者以区分真实文本区域和其他“假”候选者。 最后，将OCR应用于候选区域，并且基于对OCR识别的字符串的内容的分析来细化最终文本区域集。

3问题建模

在本文的另一部分中，我们将通过一个图表来展示一组实体 - 图表元素（DE）以及它们之间的连接。在文献中，术语“图表”与“图表”可互换使用，但图表似乎更为通用。我们将考虑使用适当的软件创建的图表图像，或使用绘图工具（标尺，绘图模板）手动精确绘制，然后通过扫描转换为光栅图像。手绘的图表超出了本文的范围，因为此类图纸中出现了很大的不准确性。我们的目标是检索图的逻辑结构，使其对应于图作者的意图。非正式地，通过图的逻辑结构，我们在这里指的是图元素之间的链接。图表元素表示出现在由图表建模的现实世界中的各种项目。它们的含义取决于应用领域。在程序流程图中，它们可以表示：语句，代码块，条件，数据源等。在逻辑电路图中，它们代表门和功能块，如：寄存器，多路复用器，触发器等。在组织结构图中，它们的元素是通常是官员或部门。虽然这里描述的方法可以应用于任何类型的图表，但我们主要关注组织图表和程序流程图。图中的元素由简单的2D几何形状描绘，如：矩形，圆形椭圆，菱形，菱形。有一些属性分配给图元素。元素的基本属性是2D形状和文本描述（通常刻入元素形状的文本）。可以从图表图像中轻松检索的其他属性包括：形状内部颜色，形状线颜色和形状边框线宽。它们在某些类型的图表中可能有意义，在其他类型的图表中它们可能被忽略。用于识别图中出现的形状并评估其属性的方法将在第IV-I节中简要描述。

图中的链接表示DE之间的逻辑关系或关联。链接由折线或交叉或连接的折线组图形表示，这些端点位于连接的DE附近。根据应用程序域，可以区分各种类型的链接。最简单的链接是一对一链接，表示一对DE之间的关联。多对多链接是将较大的DE集合关联的更复杂的情况。链接可以是无向的或定向的。在有向链接中，一些折线端点是箭头。有向链接通常表示信息流或组织从属关系。如果使用有向链接，则许多关联可以变成一对多关联，其中仅链接的单个端点没有箭头，而所有剩余的端点都是箭头。链接的图形表示将进一步称为连接器。因此，连接器是相互交叉或构成T型连接的折线组。查找已知代表一对一链接的连接器是一个简单的技术问题。同样在已知属于公共连接器的折线的连接由放置在连接点处的点（或其他图形标记）指示的情况下，问题仅在于可靠的连接标记识别。然而，如果我们不能假设属于连接器的折线的连接由连接标记以图形方式标记，则要复杂得多。在这种情况下，图表结构的识别必须基于试验来“猜测”图作者的意图。在本文的另一部分，我们将描述处理这种图的方法。

  我们在这里考虑的是在其输入上以矢量化图形图像开始的方法。原始（光栅）图像的矢量表示是通过应用图像处理操作序列，然后是将二进制图像转换为线段（矢量）集合的矢量化过程来获得的。我们还假设已经成功找到了DE。在我们的方法中，我们使用基于矢量序列匹配的形状识别方法来匹配由规则或代数定义的基本形状，如第IV-I部分中所述。构建发现的DE的载体被鉴定并从进一步的考虑中提取。令E = {e1，e2，...，eN}表示找到的DE的集合，并且让L表示未分配给任何检测到的DE的线段集合（称为后边缘）。边缘是一对2D点，其在平面上的端部是lj =（p0j，p1j），pi =（xi，yi）。 L中的线段可能属于连接器。形式上，连接器是来自L的连接边的子集。我们在这个阶段的目标是将尽可能多的边聚集到与连接器对应的不相交的子集Ci⊆L中，同时尽可能少地保留未分配的线，即属于未分配的线。设置U.结果是子集族{C1，C2，...，CM}。连接器组的构造可以被视为一个规则化的过程，其中规则定义了一些必须满足的约束，以便将某个边缘子集放入连接器中，以及强制在单个连接器中包含一些边缘的原则。确定连接器合理构造原理的规则来自对典型图图像上出现的图结构的分析。使用来自各个领域的图表集合进行分析，我们用于测试图表分析方法。定义L的连接器Ci的结构的规则如下：

•Ci的元素是连贯的（即，对于Ci中的每对边缘，Ci中有一系列其他边缘，可能是空的，连接它们）;

•如果Ci中的边缘有一个顶点未与Ci中的其他线共享（即，它是折线的端点，其中元素在Ci内），那么它必须接近来自集合E的检测到的DE中的一个，例如顶点是终点顶点;

•Ci中没有两个边的顶点作为终点顶点并且接近相同的DE（仅由单个边组成的连接器不能将DE与自身链接）;

•由集合Ci定义的图形中没有循环，即Ci中没有与其自身相交的折线;

•如果Ci中的边la具有公共顶点，L中的另一个边lb且lb未分配给任何其他连接器Cj，则lb也必须属于Ci（没有连接符以折线中间结尾）;

•如果p是属于Ci的两条边共享的内部顶点，则它不能比ǫ更接近属于另一条连接器Cj的两条边共享的内部顶点（连接器不能相互接触，除了外壳

终端边缘端点接触另一个连接器的终端边缘端点）;

•每个边缘唯一地分配给子集Ci或U中的一个（边缘不由连接器共享）;

•连接Ci中两个终端顶点的最长路径不能长于假设阈值，即8（没有非常复杂形状的连接器）;

•Ci中两个相邻边缘之间的角度与同一连接器内的任何其他边缘不共享公共顶点的角度不小于直角（连接器的折线段中没有锐角）;

•Ci内每条边的宽度与连接器中的加权平均线宽相差不超过30％;

•两个元素ei，ej∈E不能沿着单个连接器中的多个路径连接（但是它们可以通过许多路径连接，只要它们属于不同的连接器）。

因为将连接器合并为更复杂的连接似乎是合理的，只要它不会导致违反上述规则，那么最终目标是将集合L划分为子集族{U，Ci：i = 1， ...，M）}以便最小化子集Ci的数量，其中附加约束是集合U不包含构成有效连接器的任何边缘子集。

对于与图的自动分析相关的实际目的，不仅要找到连接器（作为边缘组），而且要确定通过各个连接器连接的连接Des的集合。因此，结构识别过程的最终结果是DE系列，其中每组定义由单个连接器连接的元件。另外，因为每个连接器端点可以用箭头标记，所以还应该检索该信息并将其包括在作为识别过程的输出的数据结构中。连接器的每个端点由一对（e，t）描述，其中e∈E和t∈{true，false}指示链接到e的连接器的端点是否是箭头。由单个连接器C连接的元件可以由多集VC指定{=（（ei，ti）：ei∈E，ti∈{true，false}}。表达式t = true表示箭头外观，t = false。表示无箭头连接器端点。这里应用多字节而不是简单集，因为图元素可以与自身连接。在这种情况下，存在与同一元素相关联的两个连接器端点。它导致在VC中两次包含该元素的外观。最后，图结构识别的产物是连接器集和连接图元素的相应多集的族。