[19] 中华人民共和国国家知识产权局

[12] 发明专利说明书

专利号 ZL 01135944.7

[51] Int. Cl[7]
G06K 9/20
G06K 9/34
G06T 3/40
G06T 11/60
H04N 1/387

[45] 授权公告日 2005 年 9 月 21 日

[11] 授权公告号 CN 1220162C

[22] 申请日 1996.9.4 [21] 申请号 01135944.7
分案原申请号 96111897.0
[30] 优先权
[32] 1995. 9. 6 [33] JP [31] 229508/1995
[32] 1995.12.28 [33] JP [31] 341983/1995
[71] 专利权人 富士通株式会社
地址 日本神奈川
[72] 发明人 胜山裕 直井聪
审查员 田 竞

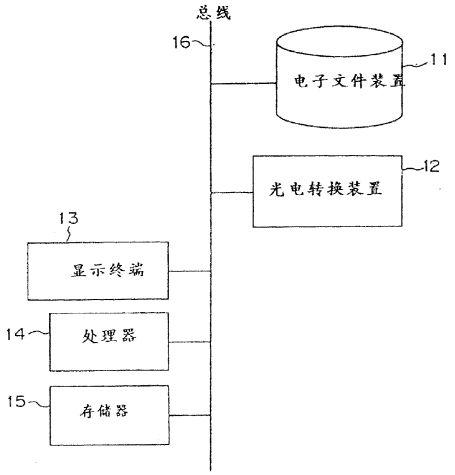
[74] 专利代理机构 中国国际贸易促进委员会专利商标事务所
代理人 吴丽丽

权利要求书 3 页 说明书 46 页 附图 77 页

[54] 发明名称 用于从文档图象抽取标题的标题抽取设备及方法

[57] 摘要

一种标题抽取装置扫描文档图象中的黑色象素并抽取外接黑色象素连接区域的矩形区域作为字符矩形。 此外，该标题抽取装置一致化邻接的多个字符矩形并抽取外接字符矩形的矩形区域作为字符串矩形。 然后，该标题抽取装置利用对应于每一字符串矩形的诸如下划线属性、框架属性、和定界线属性，文档图象中的字符串矩形的位置，以及相互位置关系的属性作为标题的似然性而计算点数，并抽取带有最高点数的字符串矩形作为标题矩形。在表格式文档的情形下，该标题抽取装置可从表的内部抽取一个标题矩形。 从标题矩形所抽取的字符通过字符识别处理用作为文档图象的关键字。



ISSN1008-4274

1. 一种用于从已经转换为图象数据的文档的文档图象抽取所需局部区域并用于进行识别的标题抽取设备，包括：

字符区域产生装置，用于产生包含由文档图象的连接的黑色象素所组成的黑色象素连接区域的字符区域；

字符串区域产生装置，用于将由所述字符区域产生装置所产生的一个或者多个字符区域一体化，并用于产生包含一个或者多个字符区域的字符串区域；

标题抽取装置，用于根据由所述字符串区域产生装置所产生的多个字符串区域的属性而抽取所述多个字符串区域中的一个特定字符串区域作为标题区域；

区段抽取装置，用于将字符串区域的内部水平划分为多个局部区域，从每一局部区域抽取具有预定的黑色象素占有率的局部区段区域，将具有超过预定阈值的高度的每一水平连接的局部区段区域一体化，并抽取该一体化的区段区域，其中使用一个区段区域抽取标题区域；

其中所述字符区域产生装置用于获得黑色象素连接区域的外接矩形作为字符区域，

其中所述字符串区域产生装置用于获得与从由所述字符区域产生装置所获得的外接矩形作为参照的第一外接矩形邻接的第二外接矩形，产生表示第一外接矩形和第二外接矩形的连接关系的连接关系表，利用连接关系表向第一外接矩形和第二外接矩形指定相同的标识信息，并将第一外接矩形和第二外接矩形一体化为一个字符串区域。

2. 如权利要求1中所述的标题抽取设备，

其中所述字符串区域产生装置用于在连接关系表中存储至少一个从第一外接矩形向第二外接矩形移动的第一指针和一个从第二外接矩形向第一外接矩形移动的第二指针。

3. 如权利要求1中所述的标题抽取设备，

其中所述字符串区域产生装置用于在框线排布在第一外接矩形和第二外接矩形之间时使第一外接矩形和第二外接矩形不被连接。

4. 如权利要求1中所述的标题抽取设备，

其中所述区段抽取装置用于将字符串区域的内部划分为多个重叠的局部区域。

5. 如权利要求1中所述的标题抽取设备，

其中所述区段抽取装置用于抽取具有与字符串区域的宽度相等的长度的区段区域。

6. 如权利要求1中所述的标题抽取设备，

其中所述标题抽取装置用于在区段区域排布在字符串区域的下面部分时从一个下划线判定该区段区域并将该字符串区域作为标题区域的一个备择。

7. 如权利要求1中所述的标题抽取设备，

其中所述区段抽取装置用于从字符串区域抽取具有相同的左边缘坐标和右边缘坐标的两个区段区域，在左边缘坐标邻域中垂直方向上产生黑色象素的第三直方图，在右边缘坐标邻域中垂直方向上产生黑色象素的第四直方图，并在第三和第四直方图的峰高度等于这两个区段区域之间的距离时判定框线排布在该字符串区域之中。

8. 一种用于从已经转换为图象数据的文档的文档图象抽取所需局部区域并用于进行识别的标题抽取方法，包括:

字符区域产生步骤，用于产生包含由文档图象的连接的黑色象素所组成的黑色象素连接区域的字符区域;

字符串区域产生步骤，用于将由所述字符区域产生步骤所产生的一个或者多个字符区域一体化，并用于产生包含一个或者多个字符区域的字符串区域;

标题抽取步骤，用于根据由所述字符串区域产生步骤所产生的多个字符串区域的属性而抽取所述多个字符串区域中的一个特定字符串区域作为标题区域；

区段抽取步骤，用于将字符串区域的内部水平划分为多个局部区域，从每一局部区域抽取具有预定的黑色象素占有率的局部区段区域，将具有超过预定阈值的高度的每一水平连接的局部区段区域一体化，并抽取该一体化的区段区域，其中使用一个区段区域抽取标题区域；

其中所述字符区域产生步骤用于获得黑色象素连接区域的外接矩形作为字符区域，

其中所述字符串区域产生步骤用于获得与从由所述字符区域产生步骤所获得的外接矩形作为参照的第一外接矩形邻接的第二外接矩形，产生表示第一外接矩形和第二外接矩形的连接关系的连接关系表，利用连接关系表向第一外接矩形和第二外接矩形指定相同的标识信息，并将第一外接矩形和第二外接矩形一体化为一个字符串区域。

用于从文档图象抽取标题的标题抽取设备及方法

本申请是申请号为“96111897.0”，发明名称为“用于从文档图象抽取标题的标题抽取装置及其方法”，申请日为1996年9月4日的分案申请。

技术领域

本发明涉及图象数据识别过程，特别涉及用于从作为文档数据所获得的文档图象抽取标题区域的标题抽取装置及其方法。

背景技术

用于从文档图象，即通过诸如扫描仪等光电转换装置从普通文档所获得的图象数据，抽取诸如文档标题之类的局部区域这样的相关技术的对照有：

（1）从带有固定区域的一文档抽取一标题（如Japanese Patent Laid-Open Publication No. 64-46873中所透露的）。

（2）利用诸如颜色标记或者轮廓线这种特定的标记手段标记文档的标题部分。通过扫描仪扫描文档并抽取标题部分（如Japanese Patent Laid-Open Publication No. 01-150974中所透露的）。

（3）诸如文档的字符串或者照片的物理结构表示为树结构等等。通过对树结构作为逻辑结构匹配，物理结构被标记有“标题”、“作者姓名”等等（如同Japanese Patent Laid-Open Publication No. 01-183784, 05-342326等中所透露的）。

（4）指定文档图象部分的区域。对区域内部进行投影并产生黑色象素的直方图。获得被投影的黑色象素数值在两个预定的阈值之间连续的区域。连续的部分的长度超过另一预定阈值的部分被抽取为标题（如同Japanese Patent Laid-Open Publication No. 05-274471中所透露的）。

此外，用于从包括一个表的文档图象抽取诸如标题的部分区域的以下相关技术对照是已知的。

（5）从包括一个表的格式化文档中抽取一个标题（如同Japanese Patent Laid-Open Publication No. 07-093348中所透露的）。

（6）对文档图象进行投影并产生黑色象素的直方图。从直方图的分布抽取轮廓线。由轮廓线所围绕的字符串被抽取作为标题（如同Japanese Patent Laid-Open Publication No. 05-274367中所透露的）。

(7) 识别文档图象中的所有字符区域的字符。对于所获得的字符代码按语言及逻辑进行诸如关键字对照和模式基本分析的知识处理。从知识处理的结果抽取看上去是标题的字符串（如同Japanese Patent Laid-Open Publication No. 03-276260中所透露的）。

(8) 由文档图象中白色象素连接部分所围绕的区域被抽取作为表部分。从表的内部抽取定界线。获得由该定界线所围绕的区域。在所获得的区域中的一个图象与预定的字符串（模板）进行模板匹配。于是，抽取相同的字符串作为标题（如同Japanese Patent Laid-Open Publication No. 03-74728中所透露的）。

然而，这些相关技术对照具有以下问题。

在方法 (1) 和 (5) 中，只能处理格式化的文档。当格式改变时，要被抽取的部分的赋值也将改变。

在方法 (2) 中标记原始文档是麻烦的。

在方法 (3) 中，要准备一个以三种结构等表示的逻辑结构的辞典。当文档的逻辑结构不包含在该辞典中时，则标题不能被精确地抽取。

在方法 (4) 中，如果这一方法用于文档图象的所有区域，虽然该方法用于分配文档图象的区域是不清楚的，诸如表或者字符大块象素部分不能正确地被抽取作为标题。而且，在仅包含字符的文档中，大字模的字符串是不总是标题。于是标题可能不能正确地被抽取。

在方法 (6) 中，如果包含标题的表由单定界线围绕，则标题能够被抽取。然而由于一个表包含复杂的定界线，标题区域就不能被精确地区分。

在方法 (7) 中，当前可用的字符识别过程占用的时间长。于是，这一方法基本上用作为批处理。此外，由于识别率不是100 %，除非使用标题位置的信息，否则会抽取不正确的部分作为标题。

在方法 (8) 中，对于图象的模板匹配处理要用时间。此外，该处理受到模板中所使用的字模的形状和规格的不良影响。并且这一方法中，只能抽取预定的字符串作为标题。于是在这一方法中，可处理的文档类型是有限制的。

于是在传统的标题抽取方法中，需要特定的准备或者特定的操作。此外，可由这些方法处理的文档和标题是有限制的。

发明内容

本发明的一个目的是提供易于从文档图象抽取标题部分的标题抽取装置及其方法。

本发明提供一种用于从已经转换为图象数据的文档的文档图象抽取所需局部区域并用于进行识别的标题抽取设备，包括：字符区域产生装置，用于产生包含由文档图象的连接的黑色象素所组成的黑色象素连接区域的字符区域；字符串区域产生装置，用于将由所述字符区域产生装置所产生的一个或者多个字符区域一体化，并用于产生包含一个或者多个字符区域的字符串区域；标题抽取装置，用于根据由所述字符串区域产生装置所产生的多个字符串区域的属性而抽取该多个字符串区域中的一个特定字符串区域作为标题区域，区段抽取装置，用于将字符串区域的内部水平划分为多个局部区域，从每一局部区域抽取具有预定的黑色象素占有率的局部区段区域，将具有超过预定阈值的高度的每一水平连接的局部区段区域一体化，并抽取该一体化的区段区域，其中使用一个区段区域抽取标题区域，其中所述字符区域产生装置用获得黑色象素连接区域的外接矩形作为字符区域，其中所述字符串区域产生装置用获得与从由所述字符区域产生装置所获得的外接矩形作为参照的第一外接矩形邻接的第二外接矩形，产生表示第一外接矩形和第二外接矩形的连接关系的连接关系表，利用连接关系表向第一外接矩形和第二外接矩形指定相同的标识信息，并将第一外接矩形和第二外接矩形一体化为一个字符串区域。

本发明提供一种用于从已经转换为图象数据的文档的文档图象抽取所需局部区域并用于进行识别的标题抽取方法，包括：字符区域产生步骤，用于产生包含由文档图象的连接的黑色象素所组成的黑色象素连接区域的字符区域；字符串区域产生步骤，用于将由所述字符区域产生步骤所产生的一个或者多个字符区域一体化，并用于产生包含一个或者多个字符区域的字符串区域；标题抽取步骤，用于根据由所述字符串区域产生步骤所产生的多个字符串区域的属性而抽取该多个字符串区域中的一个特定字符串区域作为标题区域，区段抽取步骤，用于将字符串区域的内部水平划分为多个局部区域，从每一局部区域抽取具有预定的黑色象素占有率的局部区段区域，将具有超过预定阈值的高度的

每一水平连接的局部区段区域一体化，并抽取该一体化的区段区域，其中使用一个区段区域抽取标题区域，其中所述字符区域产生步骤用获得黑色象素连接区域的外接矩形作为字符区域，其中所述字符串区域产生步骤用获得与从由所述字符区域产生步骤所获得的外接矩形作为参照的第一外接矩形邻接的第二外接矩形，产生表示第一外接矩形和第二外接矩形的连接关系的连接关系表，利用连接关系表向第一外接矩形和第二外接矩形指定相同的标识信息，并将第一外接矩形和第二外接矩形一体化为一个字符串区域。

于是，标题、地址、及发送者信息的区域易于从各个文档图象中抽取而无需进行特别的操作及使用辞典等等。从图象数据所抽取的字符串等等可用作为图象数据的关键字。

如同附图所示，本发明的这些及其它目的、特定和优点通过以下对其最佳实施方案的详细说明将变得更为显而易见。

附图说明

图1是表示本发明的理论的框图；

图2是表示本发明的一个系统的结构的框图；

图3是表示用于从文档图象抽取标题的标题抽取过程的操作流程图；

图4是表示文档图象数据的示意图；

图5是表示字符串抽取过程的操作流程图；

图6是表示对其进行了标喷处理的外接矩形的示意图；

图7是表示高度的直方图示意图；

图8是表示用于获得高度的最大频率数值的直方图的示意图；

图9是表示矩形高度表的图示；

图10是表示对应于矩形高度表的内容的直方图的图示；

图11是表示从一个大矩形抽取的段矩形的图示；

图12是表示局部段矩形的图示；

图13A、13B与13C是表示连接的局部段矩形的图示；

图14是表示框矩形的图示；

图15是表示重叠的外接矩形的的图示;

图16是表示嵌套的外接矩形的的图示;

图17是表示等腰三角形的直方图的图示;

图18是表示已经除去重叠和嵌套的外接矩形的图示;

图19是表示矩形连接关系的图示;

图20是表示连接关系表的图示;

图21是表示字符串矩形的图示;

图22是表示字符串矩形抽取过程的图示;

图23是表示所抽取的字符串矩形的图示;

图24是表示字符串矩形形成过程的操作流程图;

图25是表示清除了噪声的字符串矩形的图示;

图26A、26B、26C和26D是表示字符串矩形一致化过程的图示;

图27是表示已经一致化的字符串矩形的图示;

图28是表示文档区域的图示;

图29是表示下划线矩形的图示;

图30是表示对其框属性、定界线属性、及下划线属性已经核实的字符串矩形的图示;

图31是表示区段抽取过程的操作流程图;

图32是表示存在通配符的的情形的区段矩形的图示;

图33是表示通配符的图示;

图34是表示用于区段抽取过程（No.1）的编码的表;

图35是表示用于区段抽取过程（No.2）的编码的表;

图36是表示用于区段抽取过程（No.3）的编码的表;

图37是表示区段抽取过程（No.1）的详细的操作流程图;

图38是表示区段抽取过程（No.2）的详细的操作流程图;

图39是表示区段抽取过程（No.3）的详细的操作流程图;

图40是表示标题/地址/发送者信息抽取过程的操作流程图;

图41A和41B是表示与一个间隔重叠的字符串矩形的图示;

图42是表示第一地址抽取过程的操作流程图;

图43是表示第二地址抽取过程的操作流程图；

图44是表示标题、地址和发送者信息的第一种布局的图示；

图45是表示标题、地址和发送者信息的第二种布局的图示；

图46是表示标题、地址和发送者信息的第三种布局的图示；

图47是表示标题、地址和发送者信息的第四种布局的图示；

图48是表示多个地址和发送者信息的图示；

图49是表示标题、地址和发送者信息抽取的结果的一例的图示；

图50是表示标题、地址和发送者信息抽取的结果的另一例的图示；

图51是表示列表的文档的图示；

图52是表示表内标题抽取过程的操作流程图；

图53是表示列表的文档图象数据的图示；

图54是表示列表的文档标记的结果的图示；

图55是表示表矩形抽取过程的操作流程图；

图56是表示列表的文档的字符串矩形的图示；

图57是表示第一字符串划分过程的操作流程图；

图58是表示字符串矩形中字符矩形的顺序的图示；

图59是表示已经排序的字符矩形的顺序的图示；

图60是表示包含垂直定界线的字符串矩形的图示；

图61是表示被划分的字符串矩形的图示；

图62是表示第二字符串划分过程（No. 1）的操作流程图；

图63是表示第二字符串分割过程（No. 2）的操作流程图；

图64是表示字符串矩形列表的的结果的图示；

图65是表示已经进行划分过程的字符串矩形的图示；

图66是表示字符矩形及其字符数目的关系的图示；

图67是表示一个表矩形中的表外字符串矩形的图示；

图68是表示一个表矩形中的字符串矩形的图示；

图69是表示上定界线核实过程的的操作流程图；

图70是表示表外字符串矩形清除过程的操作流程图；

图71是表示第一查找范围的图示；

图72是表示第二查找范围的图示；

图73是表示第三查找范围的图示；

图74是表示表外字符串矩形已经被清除的字符串矩形的图示；

图75是表示标题交换输出过程的操作流程图

图76是表示字符串矩形上左边界的坐标的图示；

图77是表示表内标题的抽取结果的图示；

具体实施方式

图1是一框图表示根据本发明的标题抽取装置的理论。

图1所示的标题抽取装置包括字符区域产生装置1，字符串区域产生装置2，以及标题抽取装置3。

字符区域产生装置1产生包含作为换转为图象数据文档的文档图象中的连接黑色象素所组成的黑色象素连接区域的字符区域。

字符串区域产生装置2至少一致化由字符区域产生装置1所产生的字符区域之一，并产生包含该字符区域的一个字符串区域。

标题抽取装置3从多个字符串区域抽取一个特定的字符串区域作为对应于由该字符串区域产生装置2所产生的多个字符串区域的属性的标题区域。

字符区域产生装置1扫描文档图象中的黑色象素并抽取外接于黑色象素连接区域的矩形区域作为字符区域。于是很多字符区域对应于文档中的很多字符而产生。

然后，字符串区域产生装置2使得邻接的多个字符区域一致化并抽取外接这些字符区域的矩形区域作为字符串区域。所抽取的字符串区域例如是与水平书写的文档的一行的字符串相一致的。

标题抽取装置3评价标题对应于每一所产生的字符串区域的诸如下划线属性、框架属性、及定界线属性的似然性并抽取具有作为标题的最大似然性的特定的字符串区域作为标题区域。

下划线属性表示一条线配置在一个当前字符串区域之中或者之下。下划线属性由一个下划线标志等表示。框架属性表示字符串区域由一个框线所围绕。框架属性是由框架标志等来表示的。定界线属性表示字符串区域与垂直定界线或者水平定界线一致。定界线属性是由定界线标志等表示

的。带有下划线属性和框架属性的字符串区域常常是文档的标题。另一方面，带有定界线属性的的字符串区域常常不会是标题。于是标题的似然性可对应于这些属性而自动地被评价。

标题抽取装置3抽取包含预定规格或者更大的黑色象素的连接区域的表区域并从该表区域中的多个字符串区域抽取特定的字符串区域作为标题区域。

例如，一个表区域为外接一个黑色象素连接区域并具有预定的阈值的矩形区域。该标题抽取装置3对于作为标题的似然性评价表区域之中的字符串区域的位置与其字符数目之间的关系，并抽取具有作为标题的最大的似然性的特定字符串区域作为标题区域。

例如，最靠近表区域的左上边缘的一个字符串区域被处理为标题。此外，带有大字模的字符串区域被处理为标题。

根据本发明的标题抽取装置，无需标记原始文档及使用包含结构的特定的辞典，无论所使用的字模的规格如何，对于包括列表文档的各种文档图象都能够精确地进行标题抽取过程。此外，根据本发明的标题抽取装置，包含在所抽取的标题区域中的字符区域的 字符被识别，并且所识别的结果可被用作为文档图象的关键字。

以下，将参照附图详细说明本发明的一个实施例。

近年来，已经进行以电子装置代替传统的纸张方式存储信息的努力。电子装置的一个例子是电子文档系统。在电子文档系统中，书写在纸张上的文档由诸如图象扫描器这种光电转换装置转换为图象。关键字和管理信息分配给图象并存储在光盘或者硬盘上。

在这种方法中，由于文档是作为图象数据而存储的，所需的存储图象数据的盘的存储容量要大于通过字符识别方法对文档中书写的所有字符进行编码并然后进行存储的这种方法中的容量。然而前一方法具有胜过后一种方法的许多优点，诸如易操作性，快速的处理速度，以及包括图象和表格的非字符的存储。然而，在前一方法中，为了检索存储的信息，诸如关键字和数码这类管理信息应当与文档图象一同指定。在传统的系统中，要占用长时间设计关键字。这样，传统的系统不是用户友好的。

为了解决这一问题，可使用一种方法来自动抽取文档中的标题部分作为关键字，识别标题部分的字符，对所识别的字符进行编码，并与文档图象一同存储编码的结果。

当前可能达到的字符识别处理的速度最多为每秒钟几十个字符。当一张A4的文档被处理时，需要从30秒钟到几分钟的范围的处理时间。于是为了增加标题抽取处理的速度，从一图象仅抽取一个标题部分并识别其字符的方法是有效的。

在用于识别文档中所有的字符并抽取一个标题的方法中，图象中标题部分的位置关系是不考虑的。于是由于不正确的识别和节的连接可能不能准确抽取一个标题代码。

为了有效地操作电子文档系统，可以说重要的是从一个文档图象直接抽取标题部分（区域）的技术。以下就电子文档系统，将说明根据本发明的标题抽取技术。

图2是表示根据本发明的一个实施例的标题抽取系统的结构的一个框图。该标题抽取系统包括一个电子文件装置11，一个光电转换装置12，一个显示终端13，一个处理器14，以及一个存储器15。这些装置由总线16连接。

图1中所示的字符区域产生装置1，字符串区域产生装置2，以及标题抽取装置3是与根据本发明的该实施例（图2中所示）的处理器14相一致的。

电子文件装置11具有诸如硬盘或者光盘之类的存储装置。光电转换装置12为诸如扫描器之类的一个光学读取装置。该光电转换装置12将文档、图片、照片等等转换为图象数据。所得到的图象数据存储在电子文件装置11或者存储器15之中。显示终端13是一个具有显示装置和诸如键盘及鼠标的输入装置的操作者的终端。

对应于从显示终端13所输入的命令，处理器14从光电转换装置12所读取的以及存储在存储器15之中的文档图象抽取诸如标题之类特别的区域。另外，处理器14从电子文件装置11抽取这样一个区域。然后，处理器14识别包含在所抽取的区域中的字符。应当注意，字符识别过程可由除了标题抽取系统之外的其它系统进行。

图2所示的标题抽取系统从图4所示的文档图象获得图6中所示的字符的外接矩形，使得该字符的外接矩形一致化，并获得如图27所示的字符串矩形。然后，该标题抽取系统判定每一字符串矩形在该文档中是否被强调。

例如，由图14中所示的框线所围绕的字符串被处理为强调的字符串。该强调的字符串似乎就是标题。于是强调的字符串被抽取作为标题选择对象。并且，带有下划线和大规格的字符串被处理为被强调的字符串并被抽取为标题选择对象。字符串在文档中的位置以及当前字符串与相邻的字符串的位置关系可用作为区分标题字符串的重要信息。

于是，由于作为标题选择对象的字符串的选择要根据字符串是否被强调以及对应于表观信息，具有作为标题的高的似然性的区域可易于从文档图象中抽取。这一抽取方法比识别文档中所有的字符并抽取其标题的方法要快。此外，由于文档的类型没有限制，故本发明的抽取方法比传统方法更为一般地适用。而且，通过应用两个或者多个类型的表观信息的组合，比使用传统方法可更为精确地区分标题区域。

图3是表示图2中所示的标题抽取系统的标题抽取过程的的流程图。作为先决条件，图3所示的标题抽取过程是对于水平书写的文档所应用的。然而这一过程对于垂直书写文档可同样适用。应当注意，在垂直书写的文档中字符区域和字符串区域的高度和宽度之间的关系是与水平书写文档中的关系相反的。

在图3中，当该过程开始时，一个文档由光电转换装置12阅读并作为图象数据（文档图象）存储在存储器15之中（在步骤S1）。在这一点，为了增加处理的速度，读取的原件图象在垂直和水平两个方向被压缩到1/8。该被压缩的图象存储在存储器15之中。

当图象被压缩时，逻辑OR压缩方法用来防止区段的分开。换言之，即使由原件图象的8x8象素所组成的每一区域之一为黑色象素，当前区域的象素也被处理为黑色象素。当在当前区域中没有黑色象素时，则当前区域的象素被处理为白色象素。

然后，字符串由处理器14从文档图象中抽取。获得字符串的外接矩形（即字符串矩形）。外接矩形的坐标被存储在存储器15之中（在步骤

S2）。然后，具有小的宽度的矩形和大的高度的矩形作为噪声矩形从存储的字符串矩形除去（在步骤S3）。而且，除去不象是字符串的矩形。这样判定了一个文档区域（在步骤S4）。

然后，其余的字符串矩形在垂直方向上被排序（Y坐标）（在步骤S5）。包括框架的矩形（即框架矩形）被抽取。在框架矩形中的字符串矩形被标记为带框架的矩形（在步骤S6）。此外，包含下划线的矩形被抽取。恰好分布在下划线之上的字符串矩形被标记为下划线矩形（在步骤S7）。

然后，计算作为标题的似然性点数。带有高点数的字符串矩形被抽取作为标题（在步骤S8）。对应于这一结果，抽取该文档的地址和发送者的信息（在步骤S9和S10）。然后，识别所抽取的标题，地址和发送者的信息（在步骤S11）。其结果是完成了标题的抽取过程。

然后，对于一般的业务文档详细说明标题抽取过程。一个一般的业务文档包括诸如“标题”、“地址”、“发送日期”、“发送者部门”、“发送管理号码”、以及“文档（包括表和/或图）”这样的区域。这些区域以各种各样的组合分布。在本实施例中，一个标题，一个地址，和一个发送者的信息（包括发送日期，发送者的部门，以及发送管理号码）从带有这种格式的文档中抽取。

图4是表示由扫描器15读取的文档图象的一例的图示。图4所示的文档图象涉及一个软件促销报告的封面。文档的标题是（「ソフトウェア販推レポート送付表」）。在标题之下，排列地址和发送者信息。通过处理器14从文档图象抽取字符串。

图5是表示图3的步骤S2处的字符串抽取过程的操作流程图。

在图5之中，当该过程开始时，对应于字符的矩形从文档图象被抽取。换言之，对于文档图象使用标号方法进行黑色象素连接处理。获得了黑色象素的外接矩形并存储该结果（在步骤S21）。

在本实施例中，已经被数字化和被压缩的图象的黑色象素对应于八连接方法被扫描。当有黑色象素的连接时，相同的标号数值赋给这些黑色象素。于是产生了 黑色象素连接区域，并获得了外接矩形（字符矩形）。

对应于八连接方法的扫描方法是用于扫描每一黑色象素的八个方向（上，下，左，右，左上，右上，左下，右下）并判定是否有相邻的黑色象素。获得的外接矩形存储在文件lbtbl（lbtbl）。图6表示对于图4所示文档图象标号处理的结果。

然后，获得表示对应于标号方法的所获得的外接矩形6高度的频率分布的直方图。这样，获得了高度的最大频率数值freq（在步骤S22）。如图7所示，这一例子之中，矩形高度的直方图是以外接矩形6的一集合lbtbl作为标号方法的结果而产生的。图7中，水平轴表示外接矩形6的高度，而垂直轴表示这些高度的的外接矩形的数目（频率数值）。每一外接矩形6的高度定义为一个象素的高度的倍数。

然后，获得频率数值与带有这些频率数值的矩形的高度的关系并存储在矩形高度表高度之中。从频率数值0对矩形高度表高度进行核对。在矩形的高度变化达一个象素或者更小，矩形高度的频率数值连续改变，以及变化的频率数值的总量为9或者更多的条件下，则连续高度的最大数值定义为矩形高度的最大频率数值freq。

图8是说明表示对应于图7中所示的直方图矩形高度表高度的内容的直方图的图示。图8中，其频率数值剧烈变化的高度为最大频率数值freq。这样，当在这种方式下获得最大频率数值freq时，可以避免小于一个字符的噪声的影响。

图9是一个表，表示矩形高度表的高度的一个简单的例子。在图9中，矩形高度表高度存储了频率数值和矩形最大高度的一对数值。图10是表示矩形高度表高度的内容的直方图的一个图示。图10中所示的直方图以较低的频率数值的顺序（即，以较大的高度数值的顺序）进行核对。于是在高度10，9，和8的位置，频率数值分别变化5，5，和7。连续矩形高度差为1。变化的频率数值的总量为17。于是在高度10，9，和8，由于变化的频率数值的总量为9或者更多，故这些矩形的第一高度10定义为最大频率数值freq。

然后，为了除去框线和表的外接矩形，要指定一个用于判定大矩形的一个阈值。大于该阈值的矩形被抽取。包含框线的矩形从该大矩形被抽取并被存储（在步骤S23）。

这一例子中，大于最大频率数值freq以及具有最高频率数值的矩形的高度被定义为大矩形阈值th_large。大于阈值th_large的矩形被抽取并存储在文件盒中。

然后，为了从文件盒中的大矩形抽取框线，每一大矩形的1101的内部如图11所示被垂直划分，以便形成重叠的条形局部区域1102。在每一条形局部区域1102中获得具有一个象素高度并具有预定黑色象素占有率的水平线状区域。当两个或者多个线状区域垂直连接时，它们被一体化为一个局部区段。

图12表示图11所示的大矩形的形的条形局部区域的图示。在图12中，宽度为W的局部区域被划分为高度为1的线状区域1201。包含预定的黑色象素1204占有率并垂直连接的线状区域1201被一体化为一个局部区段矩形1202。如图12所示，两个或者多个局部区段矩形1202可能出现在一个局部区域1102中。当水平连接的局部区段区域具有八连接关系时，则它们作为一个区段被处理。图13A，13B，与13C每一表示具有八连接关系的两个局部区段矩形1202。在图11所示的情形中，从一个大矩形的上边缘部分抽取水平配置的区段矩形1103。

当区段矩形1103具有大于当前大矩形的长宽比的预定的长宽比时，该区段矩形1103被抽取为大区段矩形。当大区段矩形的两边缘长度和大矩形的两个边缘的长度之间的差在预定的范围之内，并且大区段矩形的y坐标的长度与大矩形y坐标的长度之间的差小于矩形宽度的预定的比率时，区段矩形1103被处理为分布在大矩形上或者下的水平定界线。

获得了大矩形的左边缘和右边缘的邻域中黑色象素垂直投射的频率分布的直方图。当尖峰的高度大于矩形的高度的预定的比率时，判定在左边缘和右边缘出现垂直定界线。在这一点，大矩形是作为框线的外接矩形（框矩形）被处理的。对于文件盒中的每一大矩形进行类似的处理。只有框矩形存储在文件盒中。图14表示所检测的框矩形1401，1402，和1403。

然后，已经作为框矩形和表（这些矩形被称为表矩形）被处理的矩形从由标号方法所获得的外接矩形集合lbtbl被除去（在步骤S24）。这一例子中，存储在文件盒中的框矩形从集合lbtbl被除去。然后与以下条件之一一致的矩形被假定是表矩形。这些表矩形从集合lbtbl被除去。

（a）当矩形大于整个文档图象的高度的1/3时，

（b）当矩形高度大于三倍的最大频率数值freq并具有小于0.4的长宽比时，以及

（c）当矩形具有高度大于三倍的最大频率数值freq并且宽度大于整个文档图象宽度的1/3时。

所得的矩形集合作为newtbl处理。字符串的外接矩形从矩形集合newtbl中抽取。

矩形集合newtbl中的矩形包括重叠的矩形和嵌套的矩形。这些矩形最好一致化以便澄清矩形位置的关系并有效地抽取字符串。于是，重叠的矩形和嵌套的矩形从矩形集合newtbl中被除去，而结果被存储在文件lbtbl2之中（在步骤S25）。

图15是表示两个重叠矩形的例子的图示。在图15中，矩形21和22表示倾斜区段24和25的外接矩形。矩形21和22在阴影部分26重叠。这种情形下，矩形21和22被一致化为一个矩形23。这种情形下，被嵌套的部分消除。图16是表示多个嵌套的矩形的图示。在图16中，矩形25，26，和27完全包封在矩形24之中。换言之，矩形25，26，和27嵌套在矩形24之中。这种情形下，嵌套的矩形25，26，和27被除去。这样就只有矩形24留下来。

有两种方法用于搜索矩形集合newtbl中与另一矩形重叠或者嵌套的矩形。

（d）参照一个矩形，搜索其余的矩形。

（e）在垂直或者水平方向上形成在每一矩形的一个边的中心线上具有顶点的等腰三角形。

产生了按这种方式所形成等腰三角形的直方图。此外，记录了形成频率数值的的峰值的矩形的集合（组）。相邻的峰值与直方图上的的峰值之间的距离小于预定的阈值的矩形被一致化。同时相关的矩形集合也被一致

化。矩形集合定义为一个搜索范围。当该集合中的一个矩形用作为一个参照时，该集合被搜索。另外，矩形集合在水平和垂直方向上重叠的部分可定义为搜索范围。

图17是表示用于方法（e）中的等腰三角形的直方图的一例的图示。图17中，矩形31和32的等腰三角形36和37投射到峰41。矩形33的等腰三角形38投射到峰42。矩形34和35的等腰三角形39和40投射到峰43。例如，当这些峰41，42和43出现在预定的 距离之内时，矩形31，32，33，34，和35被一致化为一个矩形集合。另外，至于矩形31和32，当矩形的等腰三角形投射到一个峰时，这些矩形可能被一致化为一个矩形集合。

根据方法（e），由于搜索在限制范围内的矩形，故该处理比方法（d）可被更快地进行。图18表示其重叠的矩形和嵌套的矩形已经被除去的外接矩形的1801的一个图示。

然后，在重叠的矩形和嵌套的矩形已经被除去之后，获得了文件lbtbl2中的矩形的高度的直方图并获得了其最大频率数值freq2（在步骤S26）。用于产生高度的直方图的方法和用于获得最大频率数值freq2的方法与步骤22的方法类似。

然后，从文件lbtbl2中抽取定界线矩形并然后该矩形被标记（在步骤S27）。这例子中，在文件lbtbl2中其高度小于1/2的最大频率数值freq，其宽度大于三倍的最大频率数值freq，并且其长宽比小于0.1的矩形被标记为定界线矩形。

然后，获得在文件lbtbl2中矩形的相互关系以便寻找包含在一个字符串中的多个字符，并然后存储在连接关系表connect之中（在步骤S28）。这例子之中，最接近一个特定矩形的矩形从文件lbtbl2在上，下，左，和右方向被搜索。结果被存储在连接关系表connect之中。矩形的相互关系表示从一个参照矩形向在上，下，左，右方向上的矩形移动的指针，在其相反的方向上移动的指针，以及从该参照矩形到上，下，左，右方向上的矩形的每一个的距离。

图19是表示一个矩形被指定为参照矩形50c的情形的矩形的连接关系的图示。在图19中，上面的矩形50a表示在上方邻接参照矩形50c的一个矩

形。上面的矩形50a由指针51和52连接到参照矩形50c。下面的矩形50d表示在下方邻接参照矩形50c的一个矩形。下面的矩形50d由指针53和54连接到参照矩形50c。左面的矩形50b表示在左方邻接参照矩形50c的一个矩形。左面的矩形50b由指针55和56连接到参照矩形50c。右面的矩形50e表示在右方邻接参照矩形50c的一个矩形。右面的矩形50e由指针57和58连接到参照矩形50c。

图20是表示存储这种指针的连接关系表connect的结构的一个图示。图20所示的连接关系表存储了参照矩形50c的表数值，到上面矩形50a的指针，从上面矩形50a来的的指针，到下面矩形50d的指针，从下面矩形50d来的的指针，到左面矩形50b的指针，从左面矩形50b来的的指针，到右面矩形50e的指针，以及从右面矩形50e来的的指针。此外，连接关系表connect还存储了从参照矩形50c到向上面，向下面，向左面，向右面邻接参照矩形50c的每一个矩形50a, 50b, 50c, 50e的距离。

连接关系表connect的产生方式是使得框架矩形的连接关系在四边断开。这样就防止了抽取从框线出来的字符串。当一个矩形最接近参照矩形50c时，也可以使用在步骤S25所使用的两个方法（d）和（e）。

然后，区分开由于扫描器的读错误所产生的噪声矩形并断开与其它矩形的水平关系（在步骤S29）。这种情形下，其高度和宽度小于最大频率数值freq2的1/4或者其长宽比小于0.1或者大于10的矩形，以及其在上和下矩形之间的距离大于预定数值的矩形被判定为噪声矩形。这些矩形之间的水平指针被除去以便断开连接关系。

然后，当相邻矩形之间的距离大或者当相邻矩形的规格的差别大时，这些矩形的连接关系被断开（在步骤S30）。这一例子中，当参照矩形与以下条件之一一致时，则与相邻矩形的连接关系被断开。

（f）当参照矩形与相邻矩形之间的距离大于三倍最大频率数值freq2时，

（g）当参照矩形或者其相邻的矩形的规格三倍大于最大频率数值freq2时，以及

（h）当相邻矩形大于两倍最大频率数值freq2时。

然后，从字符矩形集合lbtbl2和连接关系表connect抽取字符串并存储该字符串的外接矩形（字符串矩形）（在步骤S31）。这例子中，没有向右移动的指针的矩形（即，不向左边邻接另一个矩形的矩形）被作为起始的矩形处理。然后，该起始矩形的标识号码（例如标记数值）顺序地送往分布在该起始矩形的右边的另一个矩形。多个带有相同标识号码的矩形被一致化。一致化了的矩形的外接矩形定义为字符串矩形。在这一点，起始矩形的标识号码作为被抽取的字符串的标识号码（标记号码）存储在文件line_lab之中。当没有在右边连接的矩形时，停止传送标识号码。

图21是表示按以上方式抽取的字符串矩形211的一例的图示。在图21中，向四个水平排布字符矩形212到215指定标号数值L1并作为一个字符串矩形211被一致化。这例子中，字符串矩形211的标号数值也是L1。

当参照矩形右边的矩形的标识号码与文件line_lab中的字符串标识号码一致时，则右边的矩形的字符串的标识号码以所传送的矩形集合的标识号码代替。老的字符串标识号码从文件line_lab中除去。

此后，没有向右移动的指针的矩形被检测为参照矩形。当矩形分布在参照矩形的左边时，左边的矩形具有已经抽取的字符串的标识号码。这样，只要在参照矩形的右侧排布着一个矩形，则这一标识号码就传送到参照矩形右侧排布的矩形。老的标识号码由所传送的标识号码代替。老的标识号码从文件line_lab中除去。

例如，如图22所示，假定字符串67出现在另一个字符串矩形66之中。当不具有向左移动的指针的矩形64被指定为参照矩形时，矩形61排布在参照矩形64的左边。由于矩形61具有标记数值L0，标记数值L0被传送到矩形64和65。矩形64和65的标记数值以标记数值L0代替。这样，标记数值L5从文件line_lab中除去。于是，矩形61，62，63，64和65作为一个字符串矩形66被一致化。

在上述处理过程中，每一已经被标识为相同字符串的矩形具有相同的字符串标识号码。所有的矩形被扫描，并且从带有相同字符串标识号码的多个矩形的坐标获得左边缘，右边缘，上边缘，下边缘。它们作为构成一

个文件行中的字符串矩形周边而被存储。此外，被抽取的字符串的数目存储在文件maxline之中。

结果，完成了字符串抽取处理。图23是表示按上述方式抽取的字符串矩形231的图示。

然后，由处理器14进行与图3的步骤S3到S7一致的字符串矩形形成处理过程。在字符串矩形形成处理过程中，抽取并记录每一字符串矩形231的诸如下划线属性，框架属性，以及定界线属性。在以下将要说明的点数计算处理过程中，高点数指定给具有下划线属性和框架属性的字符串矩形。另一方面，低点数指定给具有定界线属性的字符串矩形231。

图24为表示字符串矩形形成处理过程的操作流程图。

在图24中，当该处理过程开始时，带有小宽度的字符串矩形和带有大高度的的字符串矩形作为噪声字符串矩形被除去，而结果被存储（在步骤S41）。这一例子中，其宽度小于最大频率数值freq的1/4的字符串矩形以及其高度小于最大频率数值freq的1/4和其长宽比大于0.1的字符串矩形被作为噪声字符串矩形处理。这些噪声字符串矩形被除去并且其余的字符串矩形存储在文件line2之中。图25是表示已经从其除去了噪声字符串矩形的字符矩形251和252的图示。

然后，生成表示文件line2中的字符串矩形连接关系的连接关系表str_conn（在步骤S42）。这一连接关系与图19所示的字符矩形的连接关系是相同的。该连接关系表str_conn具有与图20中所示的连接关系表具有相同的结构。

然后，满足诸如位置关系和高度的预定条件的两个或者多个字符串被一致化并获得一个长的字符串。该结果被存储（在步骤S43）。这一例子中，当以下条件之一被满足时，字符串矩形被一致化为大字符串矩形。

(i) 当字符串矩形之间的距离小于其高度时，

(j) 当字符串矩形水平重叠并且字符串矩形的高度几乎相同时，

(k) 当字符串矩形具有几乎等于最大频率数值freq的高度并且字符串矩形完全包含在另一个字符串矩形之中时，

(l) 当分布在三个相继的字符串矩形的两边的矩形的y坐标数值几乎相同并且只有中间的字符串矩形的y坐标数值不同于其它字符串矩形的y坐标数值时。

图26A，26B，26C和26D是表示满足方法（i），（j），（k），（l）的情形下已经被一致化的字符串矩形261到264的例子的图示。字符串矩形形成处理过程反复进行直到字符串矩形的数目不变化为止。其余的字符串矩形存储在文件line3之中。图27是表示字符串矩形已经一致化的结果的图示。当比较图25与图27所示的图示时，很清楚，字符串矩形（「ソフトウェア販推レポート」）251与（「送付表」）252被一致化为字符串矩形（「ソフトウェア販推レポート　送付表」）271。

然后，产生字符串高度的直方图以便获得字符串高度的最大频率数值str_freq（在步骤S44）。这例子中，字符串矩形高度的直方图是以图7所示的相同的方式产生的。以直方图，获得大于最大频率数值freq2和最大频率数值的高度。结果所得的数值定义为字符串矩形高度的最大频率数值（这一数值由str_freq表示）。当获得了多个与最大频率数值一致的高度时，采用与最大频率数值freq2比较接近的高度。在字符串矩形高度直方图中，有两个位置，在该处最大频率数值str_freq的两侧的频率数值都是0。在频率数值即将变为0之前的位置处的较小的高度表示为st_h，而其较大的高度表示为en_h。

然后，获得了其中除去了噪声字符串矩形的文档区域。该区域的坐标被存储（在步骤S45）。这例子中，即使字符串矩形在文档图象的左边缘和右边缘扩展到预定的的区域，这些字符串矩形也被排除。一个区域其中出现字符串矩形，区域的高度为st_h或者更大以及en_h或者更小，其宽度为最大频率数值str_freq或者更大，其长宽比小于0.5，则定义为文档区域。该区域的左边缘的x坐标，其上边缘的y坐标，其右边缘的x坐标，以及其下边缘的y坐标分别存储为st_x, st_y, en_x, en_y。当B5规格的书的一页的图象以A4规格的图象区域读取时，则左边缘和右边缘预定区域的被忽略以便将相邻的书页的字符串矩形作为噪声字符串矩形除去。图28是表示按上述方式所获得的文档区域281的图示。

另外，在文件line3中的字符串矩形垂直地（按y坐标）排序（在步骤S46）。

然后，产生表示文件line3中字符串矩形的连接关系的连接关系表str_conn2（在步骤S47）。在这一点，字符串矩形不应当从框架矩形突出。

然后判定每一字符串矩形是否完全包含在框架矩形之中。当当前的字符串矩形包含在框架矩形之中时，向该字符串矩形设置框架标志（在步骤S48）。这例子中，当文件line3中的每一字符串矩形完全含在存储在文件盒中的框架矩形中时，该字符串矩形作为有框架的矩形处理。向该字符串矩形设置框架标志。作为框架矩形的条件，

（1）框架矩形中的所有字符串矩形作为带框架的矩形处理。

（2）当框架矩形的坐标数值没有由一个阈值与字符串矩形的坐标数值分开时，该字符串矩形作为带框架矩形处理。

然后，当文件line3中的字符串矩形作为定界线矩形处理时，向定界线矩形设置一个定界线标志（在步骤S49）。在这例子中，一个字符串矩形，其规格为最大频率数值str_freq的1/2或者更小，并且其长宽比小于0.8或者大于12.5，则被作为定界线矩形处理。这种情形下，向该字符串矩形设置定界线标志。

然后，核实文件line3中的字符串矩形。当在字符串矩形之下有定界线矩形（这一定界线矩形称为下划线矩形）时或者当一个下划线出现在字符串矩形中时，向字符串矩形设置一个下划线标志（在步骤S50）。这例子中，在字符串矩形分布在定界线上，其间的距离小于最大频率数值str_freq，以及上述的字符串矩形和定界线矩形的左边缘和右边缘的长度的差为最大频率数值str_freq或者较小的条件下，向上述字符串矩形设置下划线标志。图29是表示下划线矩形的一例的图示。在图29中，由于已经设置了定界线标志的扁平的定界线矩形72分布在字符串矩形71之下，这一定界线矩形72作为下划线矩形处理。于是向字符串矩形71设定下划线标志。

此外，通过稍后将说明的方法，从其宽度或者高度为1/2或者大于最大频率数值str_freq的字符串矩形中抽取一个区段。这一区段在以下条件下作为字符串矩形中的下划线处理：从字符串矩形所抽取的区段出现在从字符串矩形的左边缘和右边缘测量的预定的象素位置，该区段的高度为该矩形的高度的WAKUTHIN倍（例如0.3倍）或更小，该区段底部的y坐标低于比该矩形的顶部的y坐标高1/2最大频率数值str_freq，该区段顶部的y坐标与该矩形顶部的y坐标之间的 差大于（最大频率数值str_freq－2），以及该区段底部的y坐标与该矩形底部的y坐标之间的 差小于该区段顶部的y坐标与该矩形顶部的y坐标之间的的差。向该字符串矩形设置下划线标志。

结果完成了字符串形成处理过程。图30是表示已经向其设置了框架标志，定界线标志和下划线标志的字符串矩形的图示。在图30中，L0到L54表示指定给相关字符串矩形的标号数值。在这些字符串矩形中，带有标号数值L1，L2，L16的字符串矩形301，302，303为有框架的矩形。

然后，将详细说明在图24中所示的步骤S50用于从字符串矩形抽取一个区段的方法。图31是表示区段抽取过程的操作流程图。

在图31中，当该过程开始时，一个字符串矩形被处理器14划分为带有预定象素宽度w的条形局部区域（在步骤61）。如图11所示的过程那样，每一区域的一半与相邻的区域的一半重叠。

然后，每一局部区域的内部对于每一1象素（h）x w象素（w）的线状区域向下校验。当一个特定的线状区域中的黑色象素的数目大于预定的的阈值时，则该线状区域的内部作为全黑色象素处理。这一区域作为黑色区域处理。当另一个黑色区域恰好排布在该特定的黑色区域之下时，则判定这两个黑色区域为连续的并作为一个黑色区域处理（局部区段矩形）（在步骤S62）。换言之，作为黑色区域的坐标，其左和右为局部区域的左和右的坐标；其顶部表示从白色区域变化的黑色区域的y坐标；而其底部是变化到白色区域的黑色区域的y坐标。于是，就一个局部区域，可获得一个或者多个黑色区域的坐标。这一过程对于所有的局部区域进行，以便获得一个黑色区域集合。

其高度大于阈值的黑色区域称为一个通配符（在步骤S63）。例如，当字符串矩形的字符被损坏而作为一团黑色时，则发生一个通配符。图32是表示字符串矩形322（划分为局部区域321）和分布在其中的通配符323的图示。图33是表示一个局部区域321中的线状区域331和通配符323的图示。在图33中，局部区域321由15个线状区域331组成。在它们之中，上面的12个线状区域331具有通配符323。

然后扫描黑色区域集合。重叠的黑色区域或者邻接的黑色区域被一致化为一个平直的矩形区域（在步骤S64到S69）。首先从黑色区域集合选择一个黑色区域。对该选择的黑色区域进行校验（在步骤S64）。当这一黑色区域不是通配符矩形时，在该黑色区域的上和下边缘的坐标以及在其左和右边缘的坐标被作为平直的矩形区域的坐标存储。向该黑色区域设置校验标志。这样，从该集合所抽取的每一黑色区域不再被使用。

然后，从该黑色区域集合抽取一个黑色区域。当黑色区域没有被校验时，其坐标与所存储的平直矩形的坐标进行比较并判定该黑色区域在平直矩形的右边是否邻接或者与其重叠。满足这样关系的黑色区域被选择（在步骤S65）。然后，判定该黑色区域是否为通配符（在步骤S66）。当黑色区域为通配符时，该黑色区域在水平方向上被一致化，而忽略其高度（在步骤S67）。在这一点，在所存储的平直矩形的右边缘的坐标以通配符矩形右边缘的处的坐标代替。

当在平直矩形的右边上邻接或者与其重叠的黑色区域不是通配符时，对于在两个矩形的上边缘和下边缘的坐标都进行比较。当差在预定的阈值之内时，这些区域在垂直和水平方向上都被一致化（在步骤S68）。在这一点，不是通配符的右侧黑色区域的上和下边缘处的坐标作为新的平直的矩形区域的上和下边缘处的坐标处理。此外，在黑色区域的右边缘处的坐标作为在平直矩形的右边缘处的坐标处理。然后，判定所有的黑色区域是否已经被校验（在步骤S69）。当有一个黑色区域待校验时，则流程返回步骤S65。此外，待校验的黑色区域变为另一个黑色区域（当在步骤S70所判定的结果为NO时）。流程返回步骤S64。在所有的黑色区域已经被抽取后，完成了区段抽取处理过程。

于是在图31所示的区段抽取过程中，矩形的内部被划分为重叠的（预定长度的）垂直带条。从一个带条中抽取带有预定的黑色象素占有率的一个部分并表示为局部区段矩形（黑色区域）。该黑色区域被存储。这些步骤与图11中所示的区段抽取方法的步骤相同。在这一点，所存储的局部区段矩形可能是下划线的一部分的小矩形。然而，当字符被损坏并接触了下划线时，局部区段矩形可能是如图32所示的带有大的高度的矩形。这些局部区段矩形在水平方向上被扫描并作为一个长的区段矩形被抽取。在图32中，忽略了字符串矩形中的通配符的高度。这些通配符与其它局部区段矩形被一致化。在字符串矩形的下边缘处抽取一个平直的矩形。

图34，35，和36表示区段抽取过程的程序代码的一例。图35表示图34中所示的程序代码的α（C1）部分。图36表示图34中所示的程序代码的β（C2）部分。图37，38，和39是分别用于说明图34，35，和36中所示的过程的操作流程图。在图34所示的过程中，损坏的字符的大量的黑色象素作为一个通配符矩形处理。具有上述八连接关系的平直的矩形被校验。与通配符矩形一同排布并具有八连接关系的矩形被一致化。这样作为一个备择的区段的一个定界线而获得了一个平直矩形。然后，参照图37，38，和39，将说明该实际的过程。

在图37中，当开始区段抽取过程时，由处理器14校验每一局部区段矩形的高度（在步骤S71）。当局部区段的高度为（（字符串矩形的高度）x 0.3）或者更大时，它被标记为一个通配符矩形（在步骤S72）。在这一点，设置use=9（其中use为局部区段矩形的标识变量）以便以一个通配符记号标记局部区段矩形。另一方面，对于其它局部区段矩形设置use=0作为常规的的矩形（标准矩形）（在步骤S73）。然后，判定是否所有的局部区段矩形已经被标记（在步骤S74）。当有待标记的局部区段矩形时，该流程返回到步骤S71。

在所有的局部区段矩形已经标记之后，抽取一个矩形作为当前的矩形i。设置（在步骤S75）xlf＝当前矩形i的左边缘的坐标，xr＝当前矩形i的右边缘的坐标，yup＝当前矩形i的上边缘的坐标，ybl＝当前矩形i的下边

缘的坐标，line_starty＝yup，以及line_endy＝ybl。然后，判定当前的矩形i的标识变量use是否为0或者9（即，use＝0或者9）（在步骤S76）。

在当前的矩形i的标识变量use为0或者9时，判定是否use＝0（在步骤S77）。当use＝0时，则设置（在步骤S78）standard_st＝yup，standard_en＝ybl，b_use＝0，use＝1，以及height＝ybl－yup＋1。当b_use为0时，它表示当前的矩形i不是通配符而是一个标准的矩形。当use＝1时，它表示当前的矩形i已经被使用。当在步骤S77处use不是0时，则（在步骤S79）设置standard_st＝0，standard_en＝0，b_use＝9，以及height2＝ybl－yup＋1。当b_use＝9时，它表示当前的矩形i是通配符而不是一个标准的矩形。

然后，抽取另一个局部区段矩形作为当前矩形k。设置（在图38所示的步骤S80）rxlf＝当前矩形k左边缘处的坐标，rxr＝当前矩形k右边缘处的坐标，ryup＝当前矩形k上边缘处的坐标，以及rybl＝当前矩形k下边缘处的坐标。然后，判定当前矩形i是否已被设置为标准矩形（即判定是否b_use＝0）（在步骤S81）。在b_use＝0时，判定当前矩形k的标识变量use是否为9（在步骤S82）。当use＝9时，这表示当前矩形i是一个标准矩形并且当前矩形k是一个通配符。

当use＝9时，判定是否有xr＋1≥rxlf，xr＜rxr，ybl＋1≥ryup，以及yup－1≤rybl（在步骤S83）。当满足这些关系时，它们表示当前矩形k排布在当前矩形i的右边，并且在水平和垂直方向上以一个象素（点）或者更多的象素重叠。当满足这些条件时，设置xr＝rxr以便向当前矩形k的右边缘扩展当前矩形i的右边缘（在步骤S84）。

当在步骤S82处use不是9时，判定是否use＝0（在步骤S85）。当use＝0时，这表示当前矩形i是一个标准矩形而当前矩形k不是一个通配符。当use＝0时，判定是否满足xr＋1≥rxlf，xr＜rxr，ybl＋1≥ryup，以及yup－1≤rybl，以及当前矩形k的高度是否在预定的范围（在步骤S86）。

当满足这些条件时，设置xr＝rxr，yup＝ryup，ybl＝rybl，use＝2，height＝rybl－ryup＋1（在步骤S87）。它们表示当前矩形i的右边缘向当前矩形k的右边缘扩展，而当前矩形i的上边缘和下边缘处的坐标由当前矩

形k的上边缘和下边缘处的坐标代替。当use＝2时，它表示当前的矩形k已经被使用。然后，判定是否满足ryup＜line_starty（在步骤S88）。当这关系满足时，设置line_starty＝ryup（在步骤S89）。然后，判定是否满足rybl＞line_endy（在步骤S90）。当这关系满足时，设置line_endy＝rybl（在步骤S91）。

然后判定是否满足b_use＝9（在图39所示的步骤S92）。当在步骤S81不满足关系b_use＝0时或者在步骤S83，S85，S86，S88和S90的判定结果为NO时，则流程进到步骤S92。

当满足b_use＝9时，判定当前矩形k的标识变量use是否为9（在步骤S93）。当满足关系use＝9时，这表示当前矩形i和当前矩形k两者都是通配符。当满足关系use＝9时，判定是否满足xr＋1≥rxlf以及xr＜rxr（在步骤S94）。当这些关系满足时，则当前矩形k排布在当前矩形i的右边，并且它们以一个点或者更多的点在水平和垂直方向上重叠。在这种情形下，设置xr＝rxr以便使得当前矩形i的右边缘向当前矩形k的右边缘扩展（在步骤S95）。

当在步骤S93不满足关系use＝9时，判定是否满足关系use＝0（在步骤S96）。当满足关系use＝0时，这表示当前矩形i是通配符，而当前矩形k却不是通配符。当满足关系use＝0时，判定是否满足关系xr＋1≥rxlf以及xr＜rxr（在步骤S97）。当满足这些条件时，设置xr＝rxr，yup＝ryup，ybl＝rybl，use＝2，b_use＝0，line_starty＝ryup，line_endy＝rybl，height＝rybl－ryup＋1，standard_st＝ryup，以及standard_en＝rybl（在步骤S98）。它们表示当前矩形i的右边缘向当前矩形k的右边缘扩展，而当前矩形i的上边缘和下边缘处的坐标由当前矩形k的上边缘和下边缘处的坐标代替。当满足关系use＝2时，这表示当前矩形k已经被使用。

然后，判定是否所有的局部区段矩形已经被抽取作为当前矩形k（在步骤S99）。当在步骤S92不满足关系b_use＝9时或者在步骤S94，S96，S97的判定结果为NO时，则流程进到步骤S99。当在步骤S99有局部区段矩形待处理时，流程返回步骤S80。在该过程对于所有的局部区段矩形完成之后，判定关系b_use＝9是否被满足（在步骤S100）。当关系b_use＝9满足

时，设置height＝height2（在步骤S101）。这一条件表示当前矩形i和所有与之连接的矩形都是通配符。

然后，判定是否所有的局部区段矩形已经被抽取作为当前矩形i（在步骤S102）。当有一个局部区段矩形待处理时，则流程返回步骤S75。在当前矩形i的标识变量use即不是0也不是9时，则这表示所抽取的局部区段矩形已经被使用。这种情形下，流程进到步骤S102。在步骤S102，抽取下一个局部区段矩形。

在所有的局部区段矩形已经被处理之后，xlf, xr, line_starty, 以及line_endy作为所抽取的区段矩形在左边缘，右边缘，上边缘和下边缘的坐标分别被存储在文件yokoline之中（在步骤S103）。结果完成了处理过程。文件yokoline与用于存储从一个字符串矩形所抽取的一个或者多个区段矩形的存储区域是一致的。

在图24的步骤S50，按上述方式从字符串矩形抽取区段。当区段与下划线矩形一致时，则向该字符串矩形设置下划线标志。在字符串矩形形成处理过程已经完成之后，由处理器14在图3的步骤S8到S10进行标题/地址/发送者信息抽取处理。图40为表示标题/地址/发送者信息抽取处理过程的操作流程图。

在图40之中，当处理过程开始时，利用相对位置，高度，以及字符串矩形的框架/下划线信息，计算表示作为标题的似然性的点数（在步骤S111）。对应于以下条件针对每一字符串矩形指定作为标题的似然性点数。

（m）正点数

字符串的属性（框架和下划线）：高点数

字符串的规格（高度和宽度）：点数正比于字符串的规格

字符串的形状（长宽比）：当字符串的长宽比大于预定的数值时指定矩形点数。

字符串的相互位置关系（垂直间隔和存在左侧矩形）：点数正比于字符串的隔离特性。

文档中的位置（中心，顶部，等等）：当字符串排布在文档的中心或者顶部位置时，点数高。当字符串的位置之间的差别小时，则点数的差别相对也小。

（n）负点数

字符串的属性（当字符串矩形由一个字符矩形组成时）：大的负点数

字符串相互位置的关系（当上面的字符串靠近下面的字符串时，当两个字符串重叠时，当一个上面的矩形和一个下面的矩形左侧对齐时，或者当一个上面的矩形和一个下面的矩形重叠时）：大的负点数

文档中的位置（右侧）：大的负点数

对应于上述条件，当对于每一个字符串矩形满足以下条件时，则指定随后的点数。

(o) 定界线矩形的点数是0。

(p) 当字符串矩形的高度小于1/2的最大频率数值str-freq时，点数是0。

(q) 当长宽比（即宽度/长度的比率）小于3时，点数为0。

(r) 当字符串矩形的宽度小于4倍的最大频率数值str-freq时，点数是0。

(s) 当字符串矩形与上述条件（o），（p），（q）和（r）不一致时，则根据以下条件对该字符串矩形指定点数。

[#1]长宽比：当长宽比为3时，指定20个点数。

[#2]垂直接近字符串矩形：除非字符串矩形重叠，在当前字符串矩形与上面的字符串矩形之间的间隔以及当前字符串矩形与下面的字符串矩形之间的间隔的每一个为（str-freq/2）或者更小时，则指定－40点数。

[#3] 在一侧垂直接近字符串矩形：在当前的字符串矩形靠近上面的字符串矩形或者下面的字符串矩形并且其间的间隔为16点或者更多时，指定－20个点。

[#4] 上面的字符串矩形与下面的字符串矩形之间的间隔：当上面的字符串矩形与下面的字符串矩形之间的间隔大于最大频率数值str-freq时，指定20个点数。

[#5] 重叠：在当前的字符串矩形与另一个字符串矩形重叠时，指定－40个点数。

[#6] 中心：当字符串矩形的水平方向上（x方向）的中心坐标在（文档区域的中心坐标）±（40%的文档区域宽度）之内时，指定30个点数。

[#7] 右侧：当字符串矩形的中心位置的坐标排布在离开文档区域的左边缘60%的位置的右边以及当数值（文档区域中心位置处的坐标－字符串矩形左边缘处的坐标）为（文档区域宽度的1/6）或者更小时，则指定30个点数。

[#8] 高度1：当字符串矩形的高度在最大频率数值str-freq的0.5到1.5倍的范围内时，指定20个点数。

[#9] 高度2：当字符串矩形的高度在最大频率数值str-freq的1.5到3倍的范围内时，指定30个点数。

[#10] 高度3：当字符串矩形的高度大于最大频率数值str-freq的3倍时，指定40个点数。

[#11] 高度4：当字符串矩形的高度大于最大频率数值str-freq的3倍，并且字符串矩形下面位置的坐标排布在距离文档区域的上面的边缘1/3的位置处时，指定10个点数。

[#12] 水平宽度：当字符串矩形的宽度大于0.4倍的文档区域宽度时，指定10个点数。

[#13] 下划线：当向一个字符串矩形设置下划线标志时，指定30个点数。

[#14] 框架：当向一个字符串矩形设置框架标志时，指定最多30个点数。该点数对字符串矩形的长宽比成比率的降低。

[#15] 无左侧矩形：当一个字符串矩形不以类似的坐标排布在当前字符串矩形的左侧，或者当其宽度小于最大频率数值的3倍的一个字符串矩形排布在当前字符串矩形的左侧时，指定20个点数。

[#16] y坐标：当字符串矩形排布在文档的顶部时，指定20个点数。每当字符串矩形向下移动1行点数减少1。

[#17] 左对齐：当字符串矩形在当前字符串矩形上的的排布方式使得当前字符串矩形的左边缘靠近另一字符串矩形左边缘时，指定－30个点数。

[#18] 重叠：当一个字符串矩形排布在当前字符串矩形之上其方式使得当前字符串矩形的左边缘或者右边缘靠近另一个字符串矩形的左边缘或者右边缘，或者当另一个字符串矩形的左边缘和右边缘比当前字符串矩形更加靠近文档区域的边缘时，指定－30个点数。

[#19] 黑色区域：当大的字符串矩形由一个黑色象素的连接区域组成时，指定－40个点数。

图41A和41B是表示如同条件[#18]的重叠的字符串矩形的图示。在图41A中，一个上面的字符串矩形411的左和右边缘靠近一个下面的字符串矩形412的左和右边缘。在图41B中，一个上面的字符串矩形413的左边缘和右边缘比下面的字符串矩形414更加靠近文档区域的边缘。这种情形下，认为对于下面的字符串矩形414作为标题的似然性是低的。

对于每一个字符串矩形计算对应于条件（o），（p），（q），（r）和（s）所获得的点数的总和并存储在存储器15之中。

然后，按较高的点数的顺序抽取标题的备择并存储结果（在标志S112）。这例子中，文件line3中所有的字符串矩形按较高的点数的顺序排序并存储该结果在文件title中。所有的字符串矩形按较高标题备择的顺序（即从第一标题备择）存储在该文件title之中。于是，第一标题备择作为标题矩形被抽取。

然后，利用对应于第一标题备择的相对位置的关系的信息抽取地址字符串矩形（地址矩形），并存储结果（在步骤S113）。此外，利用相对位置的关系的信息和对应于地址矩形的相对位置关系的信息抽取发送者信息的字符串矩形（发送者信息矩形），并存储结果（在步骤S114）。结果，完成了该处理过程。发送者信息包括文档发送的日期，发送者的姓名，报告的号码等等。

在步骤S113，获得了第一标题备择字符串矩形的y方向的位置。当第一标题备择的位置为最高位置时，进行第一地址抽取过程。否则进行第二地址抽取过程。图42是表示第一地址抽取过程的一个操作流程图。图43是表示第二地址抽取过程的一个操作流程图。

以下说明第一地址抽取过程。在图42中，当该过程开始时，由处理器14从排布在标题矩形下面的字符串矩形抽取一个关键地址矩形。存储所抽取的关键地址矩形（在步骤S121）。这例子中，抽取排布在标题矩形之下的字符串矩形作为关键地址矩形，其高度在0.6倍的st_h到1.4倍的en_h的范围，其在x方向的中心坐标出现在标题矩形的中心坐标的左边，而其长宽比大于3。正如排布在关键地址矩形上面并且其在x方向的中心坐标出现在标题矩形的中心坐标的右边的字符串矩形那样，当具有作为发送者的似然性信息的字符串矩形不出现在排布在关键地址矩形上面的字符串矩形之中时，所抽取的关键地址矩形存储在文件to之中。

然后，排布在关键地址矩形右侧的字符串矩形作为一个地址矩形添加（在步骤S122）。这例子中，排布在关键地址矩形右侧并且其y坐标出现在（关键地址矩形的y坐标）±（高度×0.2）的字符串矩形是作为地址矩形处理的。所抽取的地址矩形存储在文件to之中，其存储的方式独立于关键地址矩形。

然后，排布在上地址矩形和下地址矩形之间的字符串矩形作为地址矩形处理（在步骤S123）。这例子中，获得了所抽取的和存储在文件to之中的地址矩形的平均值（平均高度）。具有以下特征的一个字符串矩形作为一个地址矩形存储到文件to，它排布在标题矩形之下，不是尚未抽取的一个地址矩形，排布在一个地址矩形之上或者之下，其左边缘的坐标与上面或者下面的地址矩形左边缘坐标在一预定的误差范围内是一致的，其高度小于两倍的平均高度或者它到上面或者下面的地址矩形的距离小于平均高度的1/2。这一处理过程重复直到地址矩形的数目不再变化为止。

结果，完成了第一地址抽取处理过程。并抽取了文件to中的字符串矩形作为地址矩形。

以下说明第二地址抽取处理过程。图43之中，当该过程开始时，从排布在标题矩形上面的字符串矩形中抽取一个关键地址矩形并且存储（在步骤S131）。在这例子中，抽取排布在标题矩形上面的一个字符串矩形作为关键地址矩形，其高度在0.6倍的st_h到1.4倍的en_h的范围，其在x方向的

中心坐标排布在标题矩形的中心坐标的左边，而其长宽比大于3。所抽取的关键地址矩形存储在文件to之中。

然后，排布在关键地址矩形右侧的字符串矩形作为一个地址矩形添加（在步骤S132）。这例子中，以预定的距离排布在关键地址矩形右侧并且其y坐标在（关键地址矩形的y坐标）±（0.2×高度）范围内的字符串矩形是作为地址矩形处理的并存储在文件to之中，其存储的方式独立于关键地址矩形。

然后，排布在上面的地址矩形和下面的地址矩形之间的字符串矩形作为地址矩形添加（在步骤S133）。这例子之中，获得了所抽取的和存储在文件to之中的地址矩形的平均高度。排布在标题矩形之下具有以下特征的一个字符串矩形作为一个地址矩形添加到文件to，它没有被抽取为地址矩形，它排布在一个地址矩形之上或者之下，它在左边缘的坐标与上和下地址矩形的左边缘处的坐标在预定的误差之内是一致的，并且其高度小于两倍的平均高度或者它到上面或者下面的地址矩形的距离小于平均高度的1/2。这一处理过程重复直到地址矩形的数目不再变化为止。

结果，完成了第二地址抽取过程。并抽取了存储在文件to中的字符串矩形作为地址矩形。

在图40所示的步骤S114，获得了标题矩形的y方向的位置。当这一位置为最高位置时，进行第一发送者信息抽取处理过程。否则，进行第二发送者信息抽取处理过程。

在第一发送者信息抽取处理过程中，抽取排布在不是地址矩形的标题矩形下面的一个字符串矩形作为地址矩形并存储在文件from之中，该字符串矩形的高度在0.6倍的st_h到1.4倍的en_h的范围，并且它在x方向的中心坐标出现在标题矩形的右边。在第二发送者信息抽取处理过程中，抽取排布在不是地址矩形的标题矩形上面的一个字符串矩形作为地址矩形并存储在文件from之中。这样，抽取了文件from中的字符串矩形作为发送者信息矩形。

于是，第一和第二发送者信息抽取过程要比第一和第二地址矩形抽取过程简单。然而，与地址抽取过程同样，满足一预定的条件的另一个字符串矩形可作为发送者信息矩形添加。

图44是表示标题441，地址442，以及发送者信息443的第一布局的图示。在图44中，由于标题矩形441排布在文档的顶部，故进行第一地址抽取过程和第一发送者信息抽取过程。图45，46，47是分别表示标题451，461和471，地址452，462，472以及发送者信息453，463，473的第二，第三和第四布局的图示。在这些布局之中，由于标题矩形451，461和471不是排布在这些文档的顶部，故进行第二地址抽取过程和第二发送者信息抽取过程。图48是表示多个地址482a到482f和多个发送者信息483a和483b的图示。在图48之中，进行第二地址抽取过程和第二发送者信息抽取过程。

在图45，47和48所示的布局之中，当进行第二发送者信息抽取过程时，没有抽取排布在标题矩形之下的发送者信息矩形。为了避免这一问题，可以使用一种结构，其中即使标题矩形不是排布在文档的顶部，也进行第一发送者信息抽取过程。另一方面，第一和第二发送者信息抽取过程都可进行。

图49是表示由标题/地址/发送者信息抽取过程所产生并存储在文件title,to,和from的内容的图示。在图49中，字符串矩形（ 「ソフトウェア販推レポート 送付表」 ）491被抽取作为标题矩形。排布在标题矩形491之下的左对齐字符串矩形492被抽取作为多个地址矩形。排布在该文档底部的一个数码被抽取作为发送者信息493。

图50是表示标题/地址/发送者信息抽取过程的另一个抽取结果的图示。在图50中，字符串矩形（ 「外部発表の受付状況について（送付）」 ）被抽取作为标题矩形501。在标题矩形501左上位置排布的一个字符串矩形被抽取作为地址矩形502。在标题矩形501右上位置排布的多个字符串矩形被抽取作为发送者信息503。

标题矩形501，地址矩形502，和发送者信息矩形503作为字符串通过图3所示的步骤S11的识别过程被抽取和识别。在这一点，字符从一个待处理

和识别的矩形被逐个抽取。例如，识别的结果用作为电子文件系统11中的图象文件的关键字。

在上述实施例中，图31所示的区段抽取过程可被用于图3所示的步骤S6从一个大矩形抽取水平区段的过程以及在图24的步骤S50的下划线抽取过程。于是，水平区段矩形可被抽取而不论大矩形中的通配符高度如何，并且其框线可被识别。

在参照图3都50所述的实施例中，说明了用于从一个表的外部的区域抽取标题的技术。当标题排布在表的内部时，由于表矩形从图5所示的步骤S24的过程排除，故表中标题不能被抽取。

一般而言，在包含表的文档中，文档的标题常常排布在表的外部。然而，在公司内部分配的文档中，例如格式化的业务文档，标题可能排布在表的内部。即使标题排布在表的外部，当标题是诸如“会议录”之类标准的标题时，用于在电子文档系统中区分所需的文档的关键字的标题可能排布在表的区域中。

这例子之中，需要用于有效和快速抽取表内的标题部分的方法，而无需使用象传统的字符识别方法那样要占用长时间的方法。以下将说明一个实施例，用于抽取表示区域名称的条目部分，例如象作为表中的“标题”和“公司名称”之类的标题，以及表示该条目的实际内容的标题部分。

图51是表示一个列表式的公司文档的图示。在图51中，排布在由表的定界线511所围绕的一个表中的左上部分处的（「表題」）是一个条目部分，而排布在其右边的（「マルチメディアとパターン認識シンポジウム」）是一个标题部分。在水平书写文档的情形下，表中的标题部分通常是排布在条目部分的右边。

图52是一流程图，表示由图2所示的标题抽取系统所进行的表内标题抽取过程。在图52所示的过程中，作为一个先决条件，处理的是水平书写的文档。然而，如同图3所示的过程那样，垂直书写的文档也是可以处理的。

图52中，当过程开始时，一个文档由光电转换装置12扫描。被扫描的文档作为文档图象存储在存储器15之中（在步骤S141）。这例子中，如同

图3中所示的步骤S1那样，原件图象被压缩并然后被存储。图53是表示图51中所示的被压缩的文档图象的一个图示。

然后，由处理器14对文档图象进行标记处理。获得了矩形高度的最大频率数值。利用该最大频率数值，抽取大的矩形（在步骤S142）。在步骤S142所进行的处理过程与图5所示的在步骤S21，S22，S23所进行的过程类似。然而在这例子中，不抽取框架矩形。文件盒中存储的矩形要比预定的阈值th_large大。图54是表示图53中所示对文档图象进行标记的结果的一个图示。

然后，从大的矩形抽取一个围绕一个表的矩形80（这一矩形称为表矩形）（在步骤S143）。从表矩形80选择一个包含标题的矩形（在步骤S144）。在这例子中，例如选择了带有最大面积的表矩形80。以下的过程是对于所选择的表矩形80的内部进行的。

从表矩形80的内部抽取字符串（或行）矩形并获得字符串的外接矩形（这些外接矩形称为字符串矩形）。所获得的字符串矩形的坐标存储在存储器15之中（在步骤S145）。然后，其宽度小而其高度大的矩形作为噪声矩形从所存储的字符串矩形中除去（在步骤S146）。两个或者多个所得的字符串矩形被一致化（在步骤S147）。

在步骤S145的过程基本上与步骤S25到S31的过程类似。在步骤S146的处理过程与在步骤S41的处理过程类似。在步骤S147的过程与在步骤S42到S44的过程类似。

在上述过程中，对已经从表的内部抽取的字符串矩形进行排布。然而由于这些字符串矩形可能包含部分表定界线，故定界线要从字符串矩形中抽取。字符串矩形由定界线划分（在步骤S148）。

然后，计算每一字符串矩形中的字符数以便抽取象是标题的字符串矩形（在步骤S149）。字符数目在步骤S152的过程中用作为字符串矩形的一个属性。

在步骤S148的过程中，抽取对于由表的定界线所环绕的各个区域的字符串矩形。然而，当表的形状不是矩形时，排布在表外部的字符串矩形可能不被处理。于是，校验上定界线（在步骤S150）。当表的定界线不是排

布在在当前字符串矩形之上时，它作为分布在表外部的字符串矩形处理。这种字符串矩形被除去。

然后，分布在表内的字符串矩形按靠近表矩形的左上边缘的位置的顺序被排序（在步骤S151）。在当前字符串矩形的字符数目满足预定的条件时，字符串矩形作为条目部分或者标题部分被抽取（在步骤S152）。结果，完成了表内标题抽取过程。在这一点，满足预定的条件并分布在较靠近表矩形的左上边缘位置的字符串矩形被选择为较高标题的备择。

以下将详细说明表内抽取过程的各个步骤。

图55是表示图52中所示的步骤S143处表矩形抽取过程的操作流程图。当在步骤S142的过程由表矩形抽取过程跟随时，只有大于预定规格的矩形被处理。于是表矩形能够被有效地抽取。

在图55中，当过程开始时，从文件盒中大的矩形抽取大于预定阈值的矩形（在步骤S161）。在这个例子中，其高度例如大于五倍的矩形的高度的最大频率数值freq的矩形被抽取。所抽取的矩形作为表矩形存储在文件large_4bai之中。在步骤S161所抽取的表矩形用于在步骤S150的上定界线校验过程。

然后，具有宽度大于一个阈值的矩形从文件盒中大矩形抽取（在步骤S162）。这个例子中，具有宽度大于文档图象宽度的0.7倍的矩形被抽取并作为表矩形存储在文件largewide之中。

在图52中所示的步骤S144，从在步骤S162所抽取的那些矩形选择最大的表矩形。例如，从存储在文件largewide之中的那些矩形选择面积最大的表矩形。所选择的表矩形被处理。在图54所示的文档图象的情形下，只有表矩形80存储在文件largewide之中。于是，这一表矩形80被自动地处理。

然后，在图52所示的步骤S145，从所选择的表矩形的内部抽取字符串矩形。然而，符合以下条件的矩形从待处理的矩形中排除。

（t）在当前矩形为框架矩形时，

（u）在当前矩形为其高度大于3倍的最大频率数值freq并且其长宽比小于0.4的扁平矩形时，以及

（v）在当前矩形具有大于文档图象的高度的1/3的高度时。

满足条件（t）的框架矩形在图5所示的步骤S23的过程被抽取。

图56是表示通过步骤S145，S146，S147的过程一致化的字符串矩形的图示。例如在图56中，字符串矩形81，82和83的每一个包含多个由表定界线所划分的字符串。为了正确地抽取表中的字符串，字符串矩形在步骤S148由分布在每一字符矩形之间的垂直定界线划分。以下参照图57到65，说明字符串划分过程。

字符串划分方法可大概分为以下两个方法。图57是表示第一字符串划分过程的操作流程图。在第一字符串划分过程中，由处理器14判定垂直定界线是否分布在包含在每一字符串矩形中的任何两个相邻的字符矩形之间。在这一点，包含在当前字符串矩形中的字符矩形按水平方向排序以便判定黑色象素是否出现在期间。当分布有黑色象素 时，字符串矩形在黑色象素的位置被划分。这样就产生了多个新的字符串矩形。

在图57中，当过程开始时，当前字符串矩形中的字符矩形按较小的x坐标值（水平坐标值）的顺序排序（在步骤S171）。在直到步骤S147的处理过程中，串矩形中的字符矩形按较小的y坐标值（垂直坐标值）的顺序排序。换言之，水平方向的顺序不考虑。这样，就改变了存储的字符矩形的顺序。

例如，在图58所示的字符串矩形91的情形下，在字符串划分处理过程之前，字符矩形92，95，93和94按这一顺序排序并存储。然后这些字符矩形按x的坐标值的顺序重新排序。这样，字符矩形92，93，94和95按图59所示的顺序被正确地存储。

然后，一个字符串矩形的左边缘的x坐标，其右边缘的x坐标，其上边缘的y坐标，以及其下边缘y坐标，分别由sx1, sx2, sy1, sy2, 表示（在步骤S172）。此外，该字符串矩形的最左边的字符矩形定义为当前矩形（在步骤S173）。当前矩形的上边缘的y坐标，其下边缘的y坐标，其右边缘的x坐标，由cy1, cy2, cx2表示（在步骤S174）。排布在当前矩形右边的字符矩形的上边缘的y坐标，其下边缘y坐标，以及其左边缘的x坐标由ry1, ry2, 以及rx1表示（在步骤S175）。

然后，判定在由直线x = cx2, x = rx1, y = max(cy1, ry1), 以及y = min(cy2, ry2)所环绕的矩形区域中是否出现了黑色象素（在步骤S176）。这一矩形区域是排布在当前矩形与排布在当前矩形右边的字符矩形之间的一个区域。

当黑色象素出现在上述的矩形区域中时，它们被作为垂直定界线处理。由坐标x = sx1, sx2, 和y = sy1, sy2 所表示的矩形作为字符串矩形被寄存。然后设置sx1 = rx1（在步骤S177）。

然后，判定当前矩形右边的字符矩形是否排布在字符串矩形的右边缘（在步骤S178）。在否定的情形下，排布在当前矩形右边的字符矩形作为新的当前矩形处理（在步骤S179）。该流程返回步骤S174。当在步骤S176在矩形区域中没有黑色象素出现时，流程进到步骤S178。

在步骤S178，当在当前矩形的右边的字符矩形为排布在字符串矩形的右边缘的一个矩形时，由坐标x = sx1, sx2, y = sy1, sy2所表示的一个字符串矩形作为字符串矩形存储（在步骤S180）。结果，完成了第一字符串划分过程。

在第一字符串划分过程中，只要在当前矩形和其右边的矩形之间检测到一个垂直定界线，在其左边的至少一个字符矩形作为字符串矩形被存储。这样，即使原来的字符串矩形包含两个或者多个垂直定界线，字符串矩形也在其位置被划分。

例如，在如图60所示的表中的字符串矩形101的情形中，字符串矩形101包含字符矩形102，103，104，105，106，和107。一条垂直定界线分布在字符矩形102与字符矩形103之间。当对于字符串矩形101进行第一字符串划分处理时，如果字符矩形102为当前矩形，则在字符矩形102与字符矩形103之间的一个区域中检测到黑色象素（当在步骤S176的判定结果为YES时）。这样，如图61所示，包含字符矩形102的矩形作为字符串矩形108寄存（在步骤S177）。

然后，字符矩形103作为新的 当前矩形处理并进行类似的处理过程（在步骤S179）。然而没有检测垂直定界线。当字符矩形106作为新的当前矩形处理时，包含字符矩形103，104，105，106，和107的一个矩形作

为字符串矩形109寄存（在步骤S180）。结果，完成了第一字符串划分处理过程。于是，原来的字符串矩形101被划分为字符串矩形108和109。

图62和63是表示第二字符串划分过程的操作流程图。在第二字符串划分过程中，由处理器14对于每一字符串矩形的内部执行标记过程。在这一点，存储组成字符串矩形的每一字符矩形的坐标。此外，该标记过程对于字符串矩形的内部进行，以便获得字符串矩形的字符矩形的坐标。

在垂直定界线的一部分分布在字符串矩形内部的情形下，当前一组的矩形的数目与后一组矩形的数目进行比较时，它们不同是由于因为垂直定界线的原因而增加了后一组矩形的数目。于是该字符串矩形在后一组中出现不必要的字符矩形的位置被划分。

例如，在图60所示的字符串矩形101的情形，当对于字符串矩形101的内部进行标记处理时，获得了如图64所示的字符矩形。当图60所示的字符矩形组与图64所示的字符矩形组进行比较时，很明显，图64所示的字符矩形组包含一个不必要的矩形110。矩形110是与包含在字符串矩形101中的垂直定界线等同的。在这一位置，字符串矩形101被划分。

在图62中，当处理过程开始时，一个字符串矩形中的字符矩形集合由0标记（在步骤S181）。通过标记字符矩形内部所获得的字符串矩形的一个集合由N表示（在步骤S182）。集合0与N的字符矩形关于x坐标排序（在步骤S183）。该字符串矩形的左边缘的x坐标，其右边缘的x坐标，其上边缘的y坐标，和其下边缘的y坐标，分别由sx1, sx2, sy1, 和sy2表示（在步骤S184）。关于x坐标的排序过程按照图57中所示的步骤S172相同的方式进行。

然后，设置登记标志＝0。集合O的左边缘处的字符矩形由矩形OO表示。集合N的左边缘处的字符矩形由矩形NN表示。设置x2＝OO的右边缘处的x坐标以及prev＝x2（在步骤S185）。然后，登记标志具有值0或者1。

然后，判定矩形OO的左上边缘和右下边缘处的坐标是否与矩形NN的左上边缘和右下边缘处的坐标一致（在步骤S186）。当它们彼此一致时，

认为矩形OO与矩形NN是相同的。然后判定登记标志是否为1（在步骤S187）。

当登记标志为0时，排布在矩形OO右边的一个矩形作为一个新的矩形OO处理。另外，排布在矩形NN右边的一个矩形作为一个新的矩形NN处理（在步骤S188）。而且设置prev＝x2（在步骤S189）。设置x2＝OO右边缘的x坐标（在步骤S190）。然后判定矩形OO是否为排布在字符串矩形右边缘处的一个字符矩形（在步骤S191）。当另一个字符矩形排布在矩形OO的右边时，流程返回步骤S186。

当在步骤S186矩形OO的坐标与矩形NN的坐标不一致时，矩形NN作为垂直定界线处理。然后，然后判定登记标志是否为0（在图63所示的步骤S195）。当登记步骤为0时，由坐标x = sx1, prev, y = sy1, sy2所表示的矩形作为一个字符串矩形寄存（在步骤S196）。此外，设置登记标志＝1（在步骤S197）。于是，包含一个排布在矩形OO的左边的一个字符矩形的一个矩形被作为字符串矩形寄存。

然后，排布在作为垂直定界线处理的矩形NN的右边的一个矩形作为一个新的NN矩形处理（在步骤S198）。然后流程返回步骤S186。当在步骤S195登记标志不是0时，流程进到步骤S198。

当在步骤S187处登记标志为1时，矩形OO作为新一个的字符串的第一个字符处理。设置x2＝矩形OO的右边缘的x坐标以及sx1＝矩形OO的左边缘的x坐标（在步骤S192）。此外，设置prev = x2, 以及登记标志＝0（在步骤S193和S194）。然后，流程返回步骤S191。

当在步骤S191矩形OO为字符串矩形的右边缘处的一个字符矩形时，由坐标x = sx1, x2, y = sy1, sy2表示的一个矩形作为一个字符串矩形寄存（在步骤S199）。结果，完成了第二字符串划分过程。

根据第二字符串划分过程，只要检测到包含在集合N中而不包含在集合O中的一个不必要的矩形，则排布在该不必要的矩形左边的至少一个字符矩形作为一个字符串矩形寄存。然后下一个集合O的矩形排布在该字符串的左边缘。这样，不需要的一个垂直定界线从字符串矩形除去。

例如，在图64所示的字符串矩形101的情形下，集合O是由字符矩形102，103，104，105，106，和107组成的。集合N是由字符矩形102，110，103，104，105，106，和107组成的。当矩形OO是字符矩形103而矩形NN是字符矩形110时，字符矩形110作为一个垂直定界线处理（当在步骤186的判定结果为NO时）。这样，一个包含字符矩形102的矩形是作为如图61所示的字符串矩形108寄存的（在步骤S196）。

然后，字符矩形103是作为一个新的矩形NN处理的（在步骤S198）。类似的过程被重复进行。然而，对应于垂直定界线的矩形没有被检测。当字符矩形107作为矩形OO处理时，包含字符矩形103，104，105，106，和107的矩形作为字符串矩形109寄存（在步骤S199）。结果，完成了第二字符串划分过程。这样，如同第一字符串划分过程的结果那样，原来的字符串矩形101被划分为字符串矩形108和109。

当比较第一和第二字符串划分过程时，虽然这些功能基本上是相同的，但是第一字符串划分过程比第二字符串划分过程要快。图65是表示对于图56中所示的字符串矩形所执行的字符串划分过程的结果的图示。当比较图56和65中所示的结果时，很明显，原来的字符串矩形81被划分为字符串矩形111，112，和113，字符串矩形82被划分为字符串矩形114和115，而字符串矩形83被划分为字符串矩形116和117。

在字符串矩形已经划分为较小的字符串矩形之后，流程进到图52中所示的步骤S149。在步骤S149，对应于字符矩形的形状，由处理器14计算字符串矩形的字符数目。在这个例子中，字符的数目对应于每一字符矩形的长宽比而被抽取。

图66是表示字符矩形与字符数目的关系的一个图示。在图66中，每一字符矩形的高度和其宽度分别以H和W表示。认为一个字符的高度几乎等于它的宽度。这样，在字符矩形中的字符的数目可表示为[W/H]（其中[W/H]是表示实数W/H的小数被截去的一个计算符号）。

通过在步骤S148的字符串划分过程，表矩形中的字符串矩形被正确地划分。然而，表矩形可能包含实际上排布在表外部的一个字符串矩形。图67是表示排布在表矩形中的一个表外字符串矩形的例子的图示。图67中，

由于由实线表示的表的定界线124的周边不是矩形，故表矩形121包括一个排布在表外部的字符串矩形122。另一方面，象字符串矩形122那样排布在同一行上的一个字符串矩形123是表内的一个字符串矩形。

图68是表示图54中所示的表矩形80中的一个字符串矩形的图示。图68所示的多个字符串矩形的字符串矩形131是一个表外字符串矩形。为了从一个表抽取一个标题，诸如字符串矩形122和131的表外字符串矩形应当与表内字符串矩形区分开，并且表外字符串矩形应从表矩形中除去。

于是在步骤S150，判定是否有一条定界线排布在与上面的字符串矩形不邻接的字符串矩形之上。当没有定界线排布时，就除去该字符串矩形。

图69是表示上定界线校验过程的一个操作流程图。在图69中，当过程开始时，按图24所示的步骤S42相同的方式产生表示字符串矩形的连接关系的连接关系表（在步骤S201）。利用该连接关系表，获得了在其上没有其它字符串矩形排布的字符串矩形。当定界线不排布在字符串矩形上面时，它们被除去（在步骤S202）。结果完成了上定界线校验过程。

图70是表示在步骤S202的表外字符串矩形除去过程的操作流程图。在图70所示的表外字符串矩形除去过程中，参照表矩形中所有字符串矩形的连接关系表，其它字符串矩形不在其上排布的字符串矩形被抽取。每一被抽取的字符串矩形的预定的区域被搜索以求得黑色象素的字节数目。在这例子中，八象素等于一字节。字节的数目由M表示。

当总数M等于或者大于以字节表示搜索范围宽度的阈值L时，认为水平定界线出现在该范围内。字符串矩形是作为一个表内字符串矩形寄存的。当有一个字符串矩形满足关系M＜L时，认为水平定界线不排布在该字符串矩形上面。这一字符串矩形作为表外字符串矩形被除去。

在图70中，当该过程开始时，表矩形中的字符串矩形集合由表内字符串矩形集合S表示（在步骤S211）。然后，其上没有集合S的其它字符串矩形排布的字符串矩形被抽取。被抽取的字符串矩形由集合S1表示（在步骤S212）。在图67所示的表矩形的情形下，划阴影的字符串矩形122和123是集合S1的元素。

然后，在集合S1中的一个字符串矩形由SS表示（在步骤S213）。字符串矩形SS的左边缘的x坐标，其右边缘的x坐标，其上边缘的y坐标，其下边缘的y坐标，分别由sx1, sx2, sy1, 和sy2表示（在步骤S214）。

然后，获得了排布在字符串矩形SS或者不是表矩形的字符串矩形之上的一个表矩形。所获得的该表矩形的左边缘的x坐标，其右边缘的x坐标，其上边缘的y坐标，其下边缘的y坐标，由ux1, ux2, uy1, 和uy2表示（在步骤S215）。在这一点，参照了在图55所示的步骤S161所处抽取的作为其它表矩形并存储在文件large_4bai的表矩形。

然后，由直线x = max(sx1, ux2), x = min(sx2, ux2), y = sy1, 和y = uy2所围绕的矩形区域由L表示（在步骤S216）。矩形区域的宽度等同于字符串矩形SS的宽度和排布在该字符串矩形SS之上的矩形的宽度的重叠部分。在这例子中，字节数目L由以下公式给出。

$$L = \min(sx2, ux2) / 8 - \max(sx1, ux1) / 8 + 1$$

然后，获得该矩形区域中的黑色象素。获得了作为黑色象素的字节数的总和M（其中八个象素等于一个字节）（在步骤S217）。

当排布在步骤S215所获得的字符串矩形SS之上的一个矩形为字符串矩形时，在步骤S216的字节数目L以及在步骤S217的黑色象素的搜索范围如图71所示。当排布在字符串矩形SS之上的一个矩形为另一个表矩形时，在步骤S216的字节数目L以及在步骤S217的黑色象素的搜索范围如图72所示。

然后，总和M与字节数L进行比较（在步骤S218）。当总和M小于L时，认为水平定界线不排布在该字符串矩形SS之上。于是，字符串矩形SS作为一个表外字符串矩形处理。这样，字符串矩形SS从集合S中除去（在步骤S219）。

然后，判定该字符串矩形SS是否为集合S1的最后的字符串矩形（在步骤S220）。当字符串矩形SS不是最后的字符串矩形时，流程返回步骤S213。在集合S1的所有字符串矩形已经被处理之后，就完成了表外字符串矩形除去过程。

在步骤S215，当即没有表矩形，也没有字符串矩形排布在字符串矩形SS之上时，则在步骤S217搜索直到文档图象的上边缘的范围以便获得黑色象素。图73是表示搜索范围的一个图示。该宽度与字符串矩形SS的宽度一致。

通过图70所示的过程，除去图68中所示的表外字符串矩形131。图74表示其余的表内字符串矩形。标题备择对应于矩形的位置和数目的关系从所获得的表内字符串矩形抽取。

图75是表示图52所示的步骤S151和S152处执行标题备择输出过程的操作流程图。在水平书写的文档的情形下，在最接近左上边缘的位置处排布的一个字符串象是一个标题。这样，在图75中所示的标题备择输出过程中，字符串矩形是按接近表的左上边缘的位置的顺序而排序的。然后，表内标题备择是对应于该顺序以及在步骤S149所获得的字符的数目的信息而指定的。标题备择是按这一顺序输出的。

可以主要使用以下三种方法指定标题备择的优先顺序。

(w) 较接近表的左上边缘的标题备择被指定为较高的优先权。

(x) 校验每一相邻的字符串矩形的字符数目。对应于字符数目，对于字符串矩形指定标题备择的优先顺序。表示诸如“title”“subject”的标题的条目名称可能是排布在左边（或者上面）的标题。这些条目名称和标题的关系可由字符的数目来表示。当字符串由直到十的几个字符组成，以及几个字符排布在由两个到几个字符所组成的字符串右边或者下边时，可以判定，存在一个条目名称和一个标题的一对。一个较高的优先权指定给这些对子。

(y) 较高的优先权指定给这样的字符串矩形，它们由预定的字符数目组成，并具有字符数目对相邻的字符串矩形的关系是按接近于表的左上边缘的位置的顺序。

在这种情形下，在表中的字符串矩形从表的左上边缘进行校验。在当前字符串矩形的字符数目超过预定的阈值时，它作为一个条目备择来处理。此外，当另一个字符串矩形排布在当前字符串矩形的右边时，不论其字符数目如何它都作为一个标题备择来处理。

换言之，当象“item: title”这样条目和标题都排布在一个区域中时，则标题可被抽取。这样，即使一个字符串矩形包含两种元素，标题也可被抽取。此外，由许多字符组成的字符串矩形很可能是一个表内标题。即使这种字符串矩形作为一个条目备择而输出，当它象是对应于字符识别的结果的一个标题时，它可以作为一个标题处理。

在当前字符串矩形的字符数目少于阈值的情形下，当另一个字符串矩形排布在当前字符串矩形的右边并且该另一个字符串矩形的字符数超过阈值时，当前字符串矩形作为一个条目备择而被处理，而另一个字符串矩形作为条目备择处理。

方法（w），（x）和（y）对于20种文档图象实验的结果显示，方法（y）在表内标题抽取性能上优于方法（w）和（x）。在图75所示的处理过程中，优先权是对应于方法（y）指定的。

在图75中，当处理过程开始时，利用图76中的每一表内字符串矩形761的左上边缘的坐标（x1, y1），它们按（x1＋y1）的顺序排序（在步骤S221）。表内字符串矩形集合761以S表示（在步骤S222）。在集合S中具有（x1＋y1）的最小值的表内字符串矩形以当前矩形SS表示（在步骤S223）。

然后，判定当前矩形SS的字符数是否大于阈值TITLEMOJISUU（在步骤S224）。例如设定TITLEMOJISUU＝7。在当前矩形SS的字符数等于或者大于TITLEMOJISUU时，判定另一个字符串矩形是否排布在当前矩形SS的右边（在步骤S225）。在另一个字符串矩形没有排布在当前矩形SS的右边时，它作为一个条目备择而输出（在步骤S226）。然后，判定当前矩形SS是否为最后的一个字符串矩形（在步骤S227）。在当前矩形SS不是最后的一个字符串矩形时，具有下一个较小的（x1, y1）数值字符串矩形作为一个新的当前矩形SS处理（在步骤S231）。然后流程返回到步骤S224。

当在步骤S225一个字符串矩形排布在当前矩形SS的右边时，该当前矩形SS作为一个条目备择输出。排布在当前矩形SS右边的字符串矩形作为一个标题备择而输出（在步骤S230）。然后流程进到步骤S227。

在步骤S224当前矩形SS的字符数少于阈值TITLEMOJISUU时，判定另一个字符串矩形是否排布在当前矩形SS的右边（在步骤S228）。当另一个字符串矩形排布在当前矩形SS的右边时，判定字符数是否大于阈值TITLEMOJISUU（在步骤S229）。当字符数超过阈值TITLEMOJISUU时，流程进到步骤S230。

当在步骤S228另一个字符串矩形没有排布在当前矩形SS的右边时，或者在步骤S229当排布在当前矩形SS的右边的字符串矩形的字符数少于阈值TITLEMOJISUU时，流程进到步骤S227。当在步骤S227当前矩形SS为最后的字符串矩形时，完成了标题备择输出过程。

根据标题备择输出过程，满足以下三个条件的字符串矩形作为条目备择或者标题备择输出。

（α）在当前矩形字符数超过阈值并且另一个字符串矩形没有排布在当前矩形的右边时，当前矩形作为一个条目备择输出。

（β）在当前矩形字符数超过阈值并且另一个字符串矩形排布在当前矩形的右边时，当前矩形作为一个条目备择输出而排布在当前矩形右边的字符串矩形作为标题备择输出。

（γ）在当前矩形字符数小于阈值并且排布在当前矩形右边的另一个字符串矩形的字符数超过阈值时，当前矩形作为一个条目备择输出而排布在当前矩形右边的字符串矩形作为标题备择输出。

图77是表示一个表内标题第一备择的图示。在图77中，字符串矩形111为一个条目备择。字符串矩形112为一个标题备择。根据表内标题抽取过程，对于表中的条目和标题的区域可从包含各种表的文档中抽取，而无需使用特别的操作和辞典。

被抽取作为条目备择和标题备择的字符串矩形是通过与图3所示的步骤S11相同的过程作为字符串识别的。在这一点，实际上抽取作为条目备择的字符串可能包含标题字符串。于是，被识别的结果的适当部分用作为一个条目名称或者一个标题。

根据本发明，字符区域和字符串区域的形状可以不总是矩形。而是可以使用通过直线或者曲线围绕的任何形状的区域。

应当注意，本发明可用于任何模式，诸如符号和图形以及文档。

虽然本发明已经就其最佳实施例进行了说明和描述，但业内专业人士应当理解，在不背离本发明的精神和范围的情形下可在其形式和细节上作出上述的和各种其它的变化，省略，和添加。

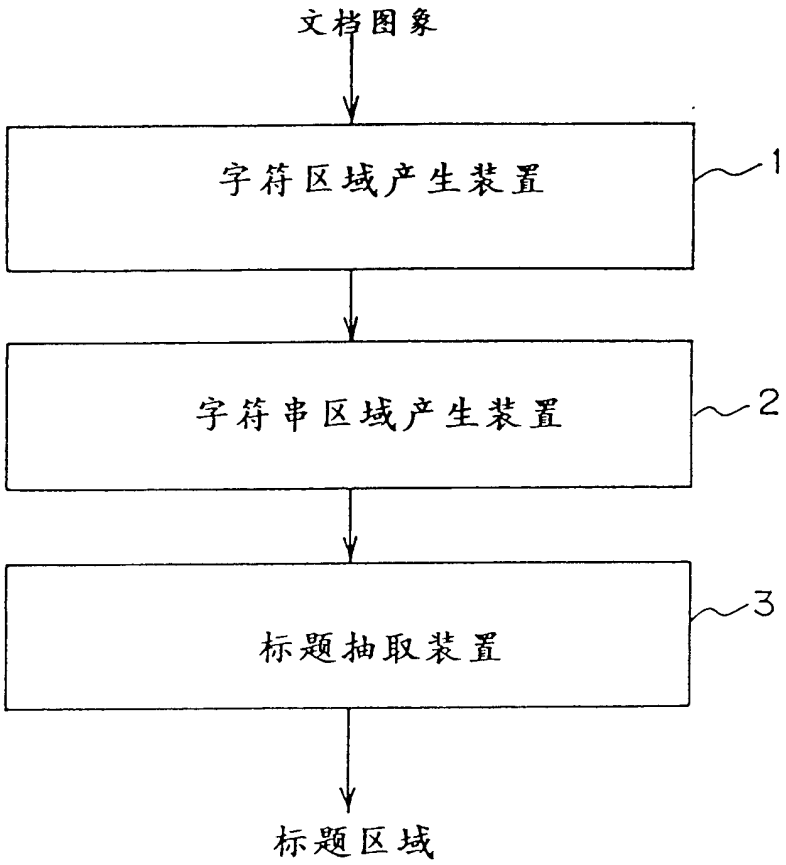


图.1

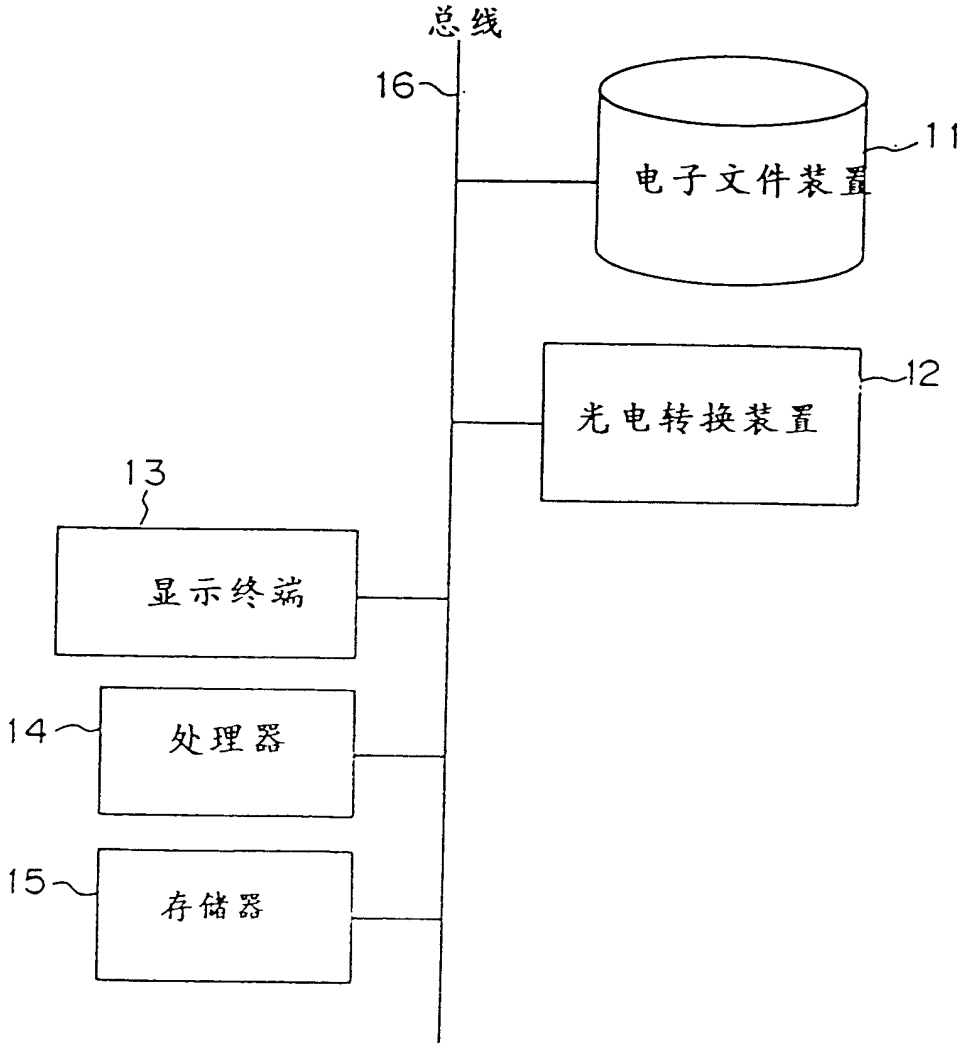


图.2

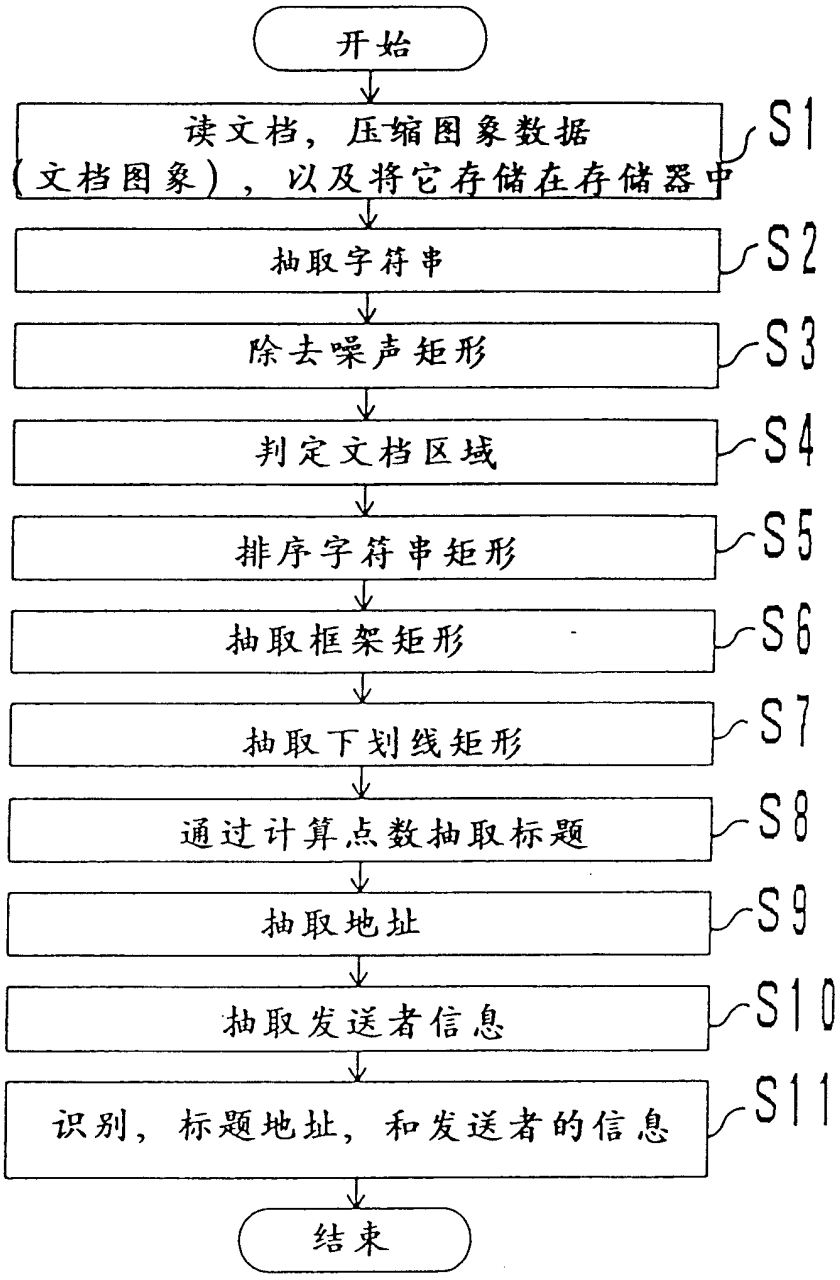

开始
读文档，压缩图象数据（文档图象），以及将它存储在存储器中
S1
抽取字符串
S2
除去噪声矩形
S3
判定文档区域
S4
排序字符串矩形
S5
抽取框架矩形
S6
抽取下划线矩形
S7
通过计算点数抽取标题
S8
抽取地址
S9
抽取发送者信息
S10
识别，标题地址，和发送者的信息
S11
结束

图.3


ソフトウェア販推レポート　送付表　【12】

新横浜

【TECHビル】

オープンシステム事業本部

[illegible]

【日本不動産ビル ⋯SSC】

ミドルウェア事業本部

[illegible]

情報処理事業推進本部

[illegible]

情報システム事業本部

[illegible]

川崎工場

社内システム開発本部

[illegible]

ミドルウェア事業本部

[illegible]

情報システム事業本部

[illegible]

オープンシステム事業本部

[illegible]

パーソナルビジネス本部

[illegible]

輸出管理本部

[illegible]

海外事業本部

[illegible]

图. 4

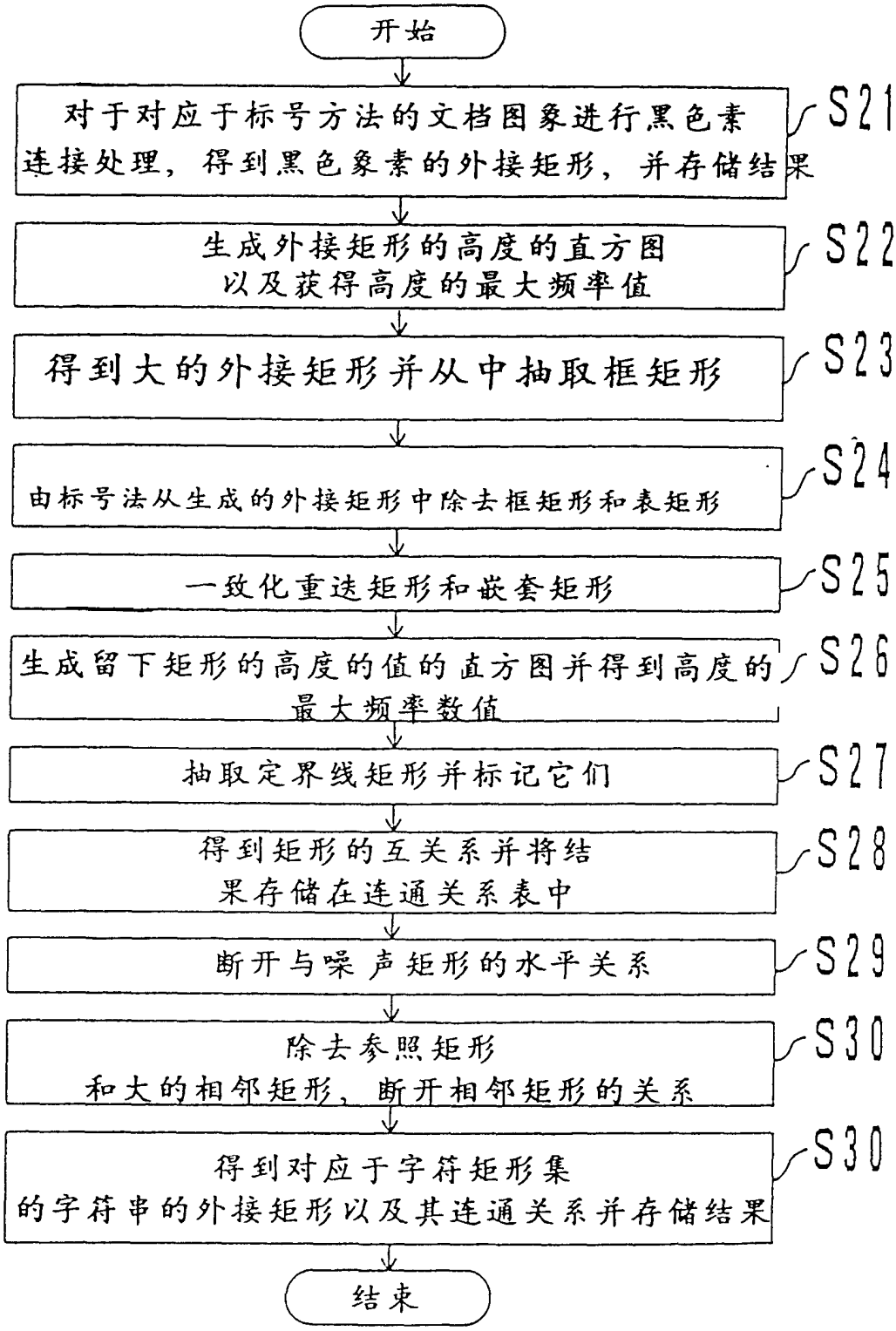

开始
对于对应于标号方法的文档图象进行黑色素连接处理，得到黑色象素的外接矩形，并存储结果
S21
生成外接矩形的高度的直方图以及获得高度的最大频率值
S22
得到大的外接矩形并从中抽取框矩形
S23
由标号法从生成的外接矩形中除去框矩形和表矩形
S24
一致化重迭矩形和嵌套矩形
S25
生成留下矩形的高度的值的直方图并得到高度的最大频率数值
S26
抽取定界线矩形并标记它们
S27
得到矩形的互关系并将结果存储在连通关系表中
S28
断开与噪声矩形的水平关系
S29
除去参照矩形和大的相邻矩形，断开相邻矩形的关系
S30
得到对应于字符矩形集的字符串的外接矩形以及其连通关系并存储结果
S30
结束

图 5

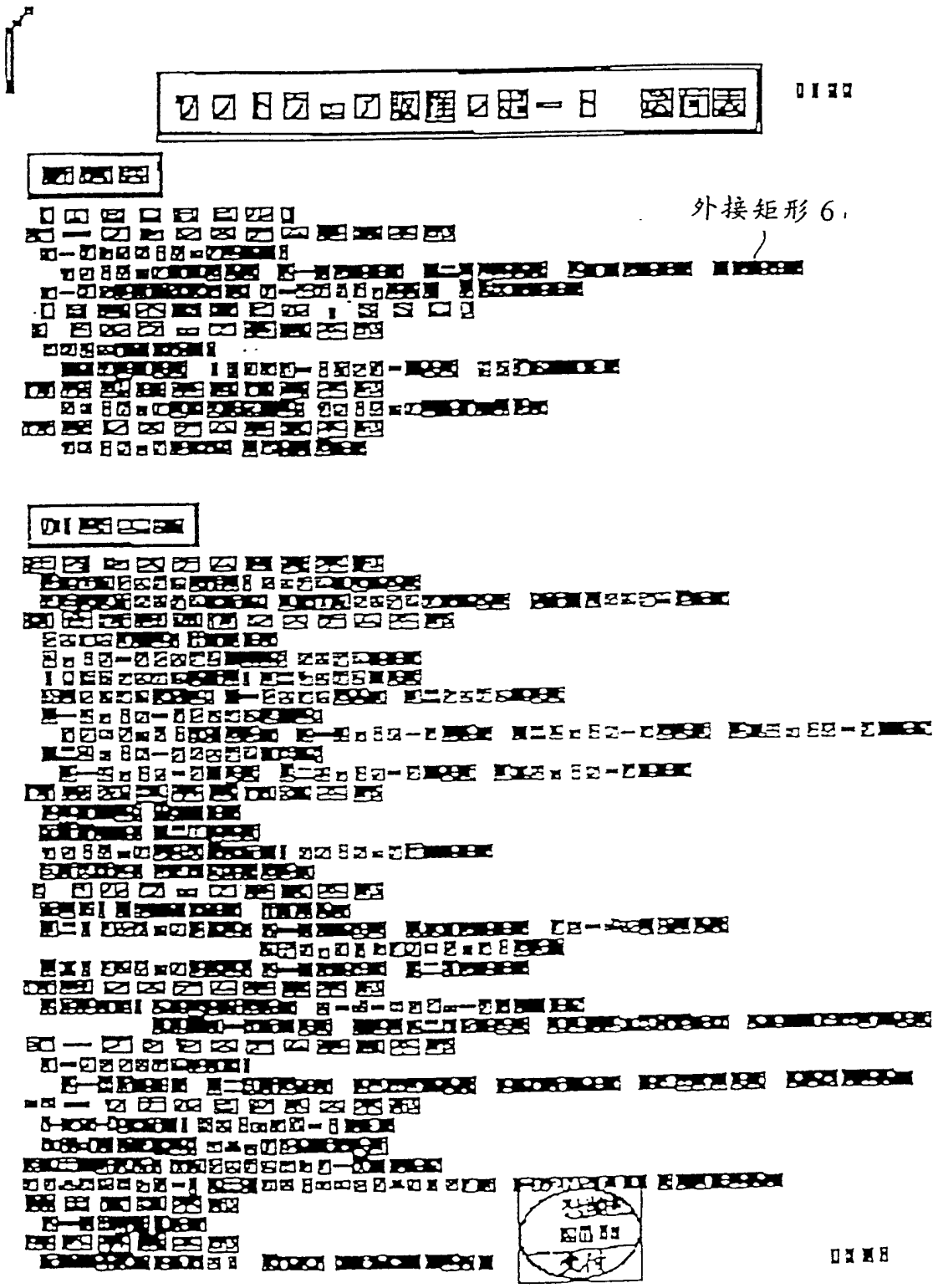
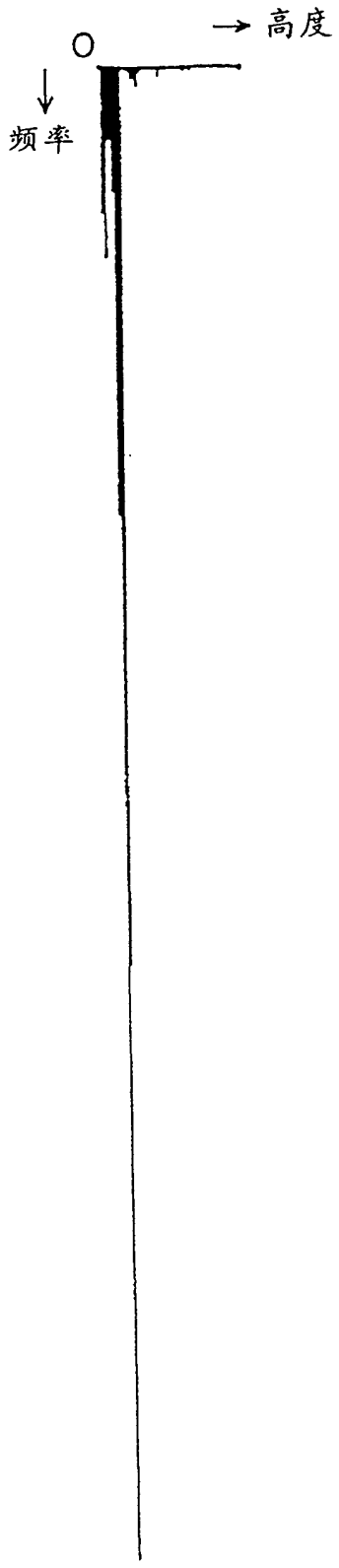


图. 6



图．7

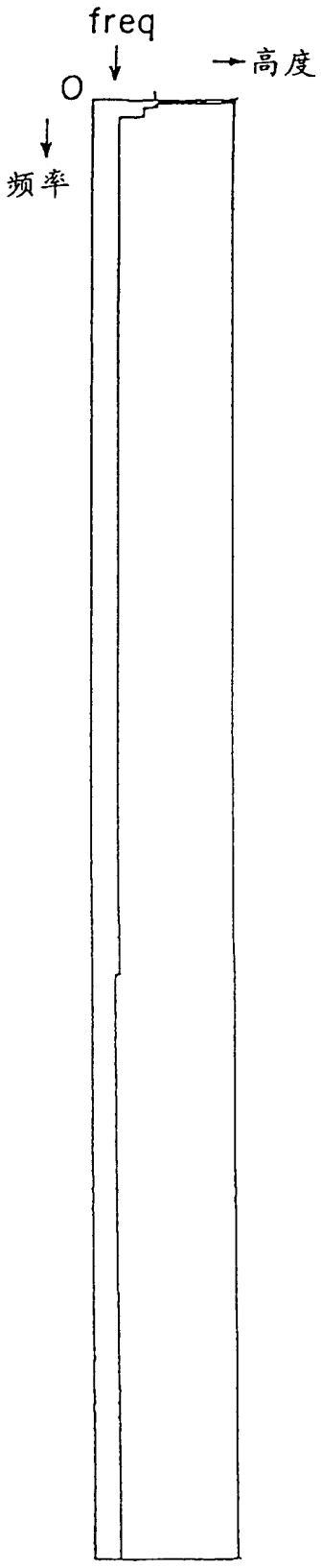


图. 8

频率	最大高度
2	15
7	10
12	9
19	8

图. 9

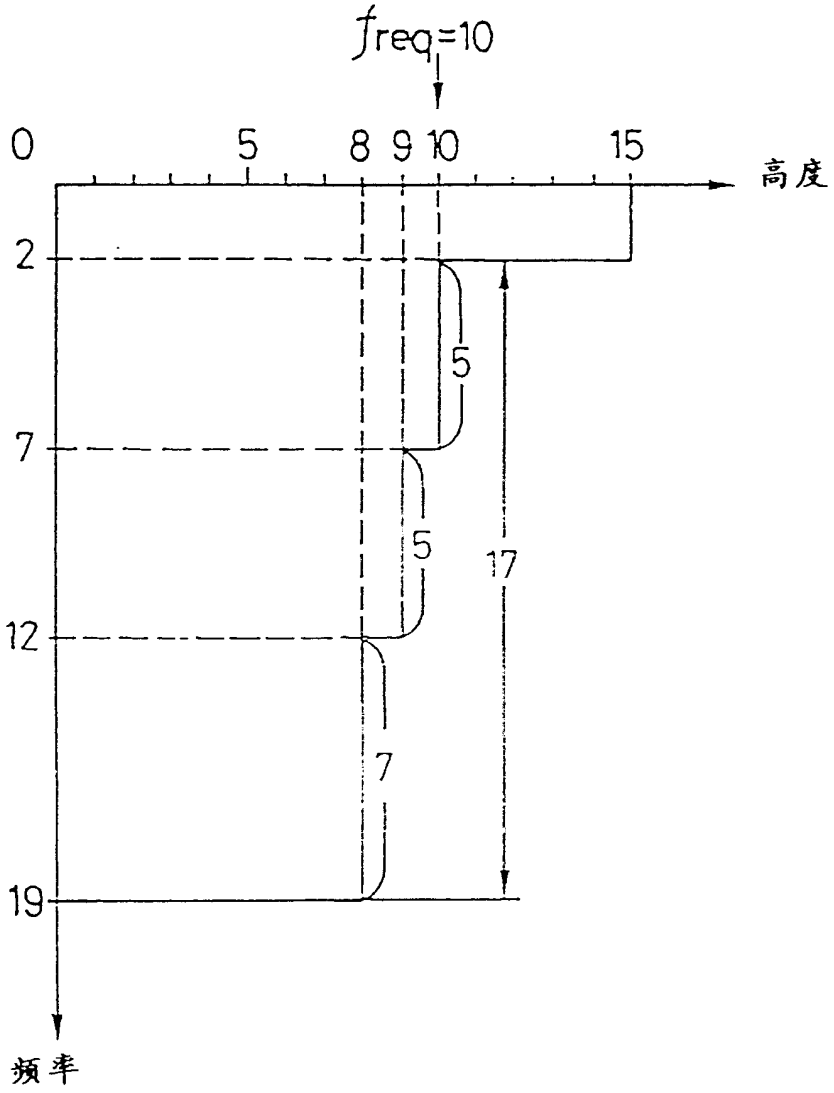


图. 10

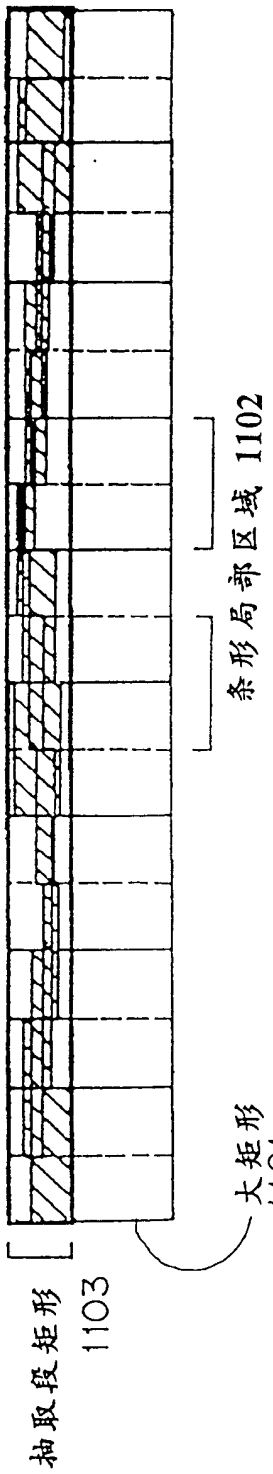


图. 11

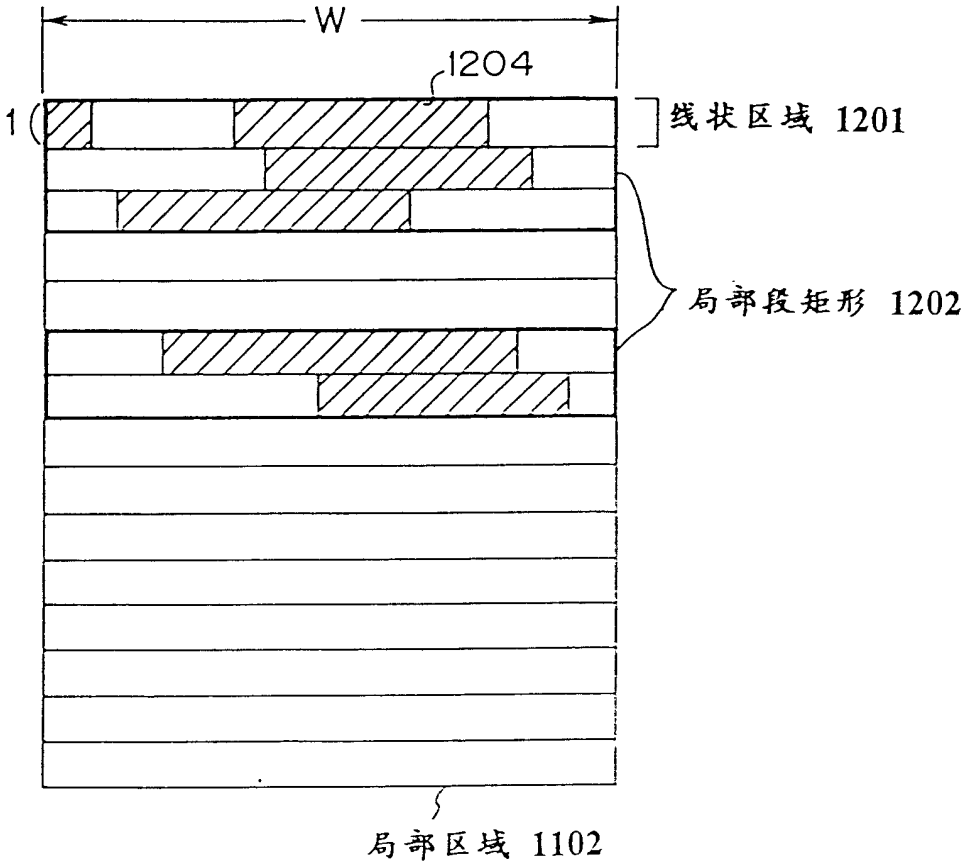


图. 12

图13A

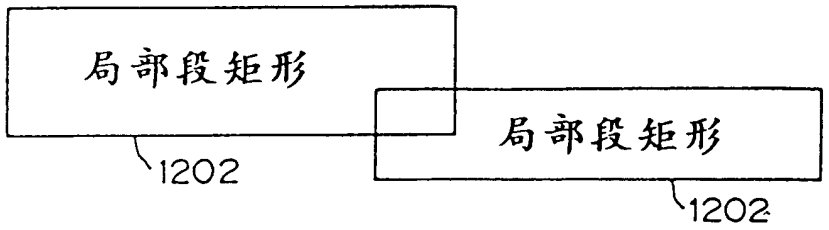


图 13B

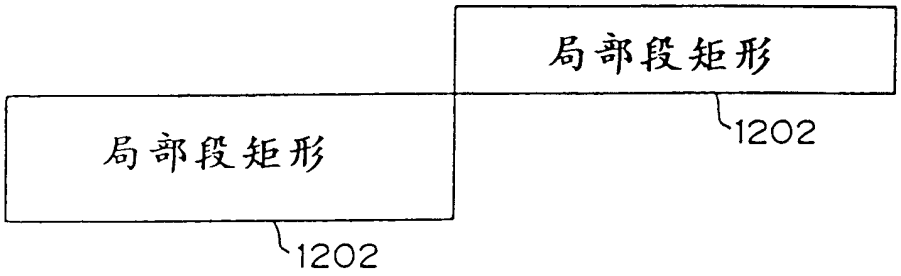
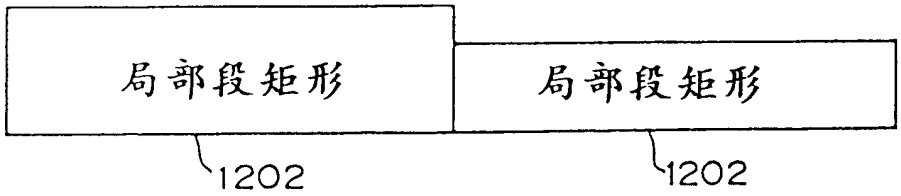


图13C



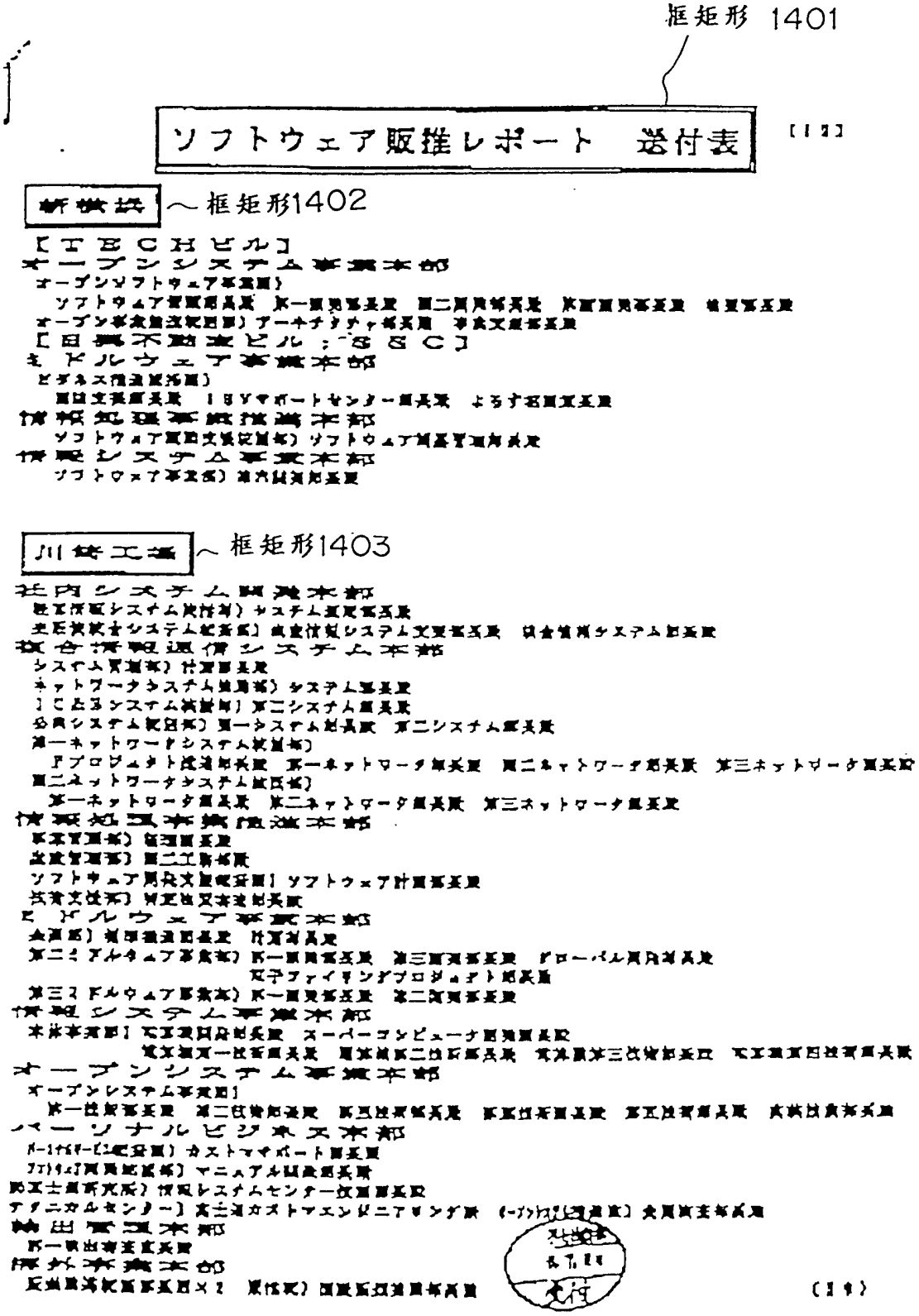


图. 14

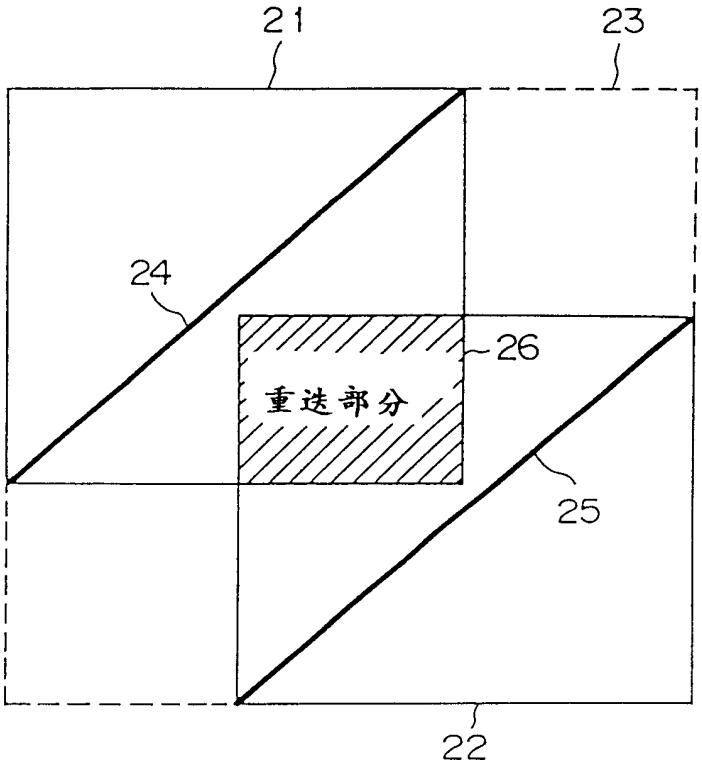


图. 15

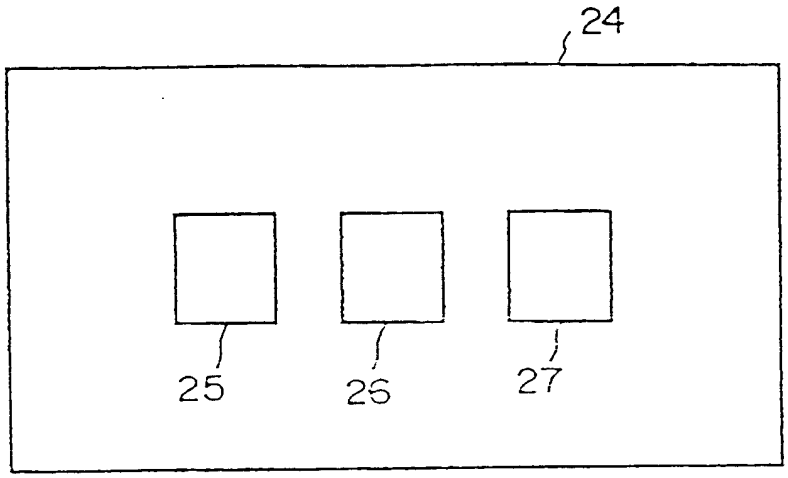


图. 16

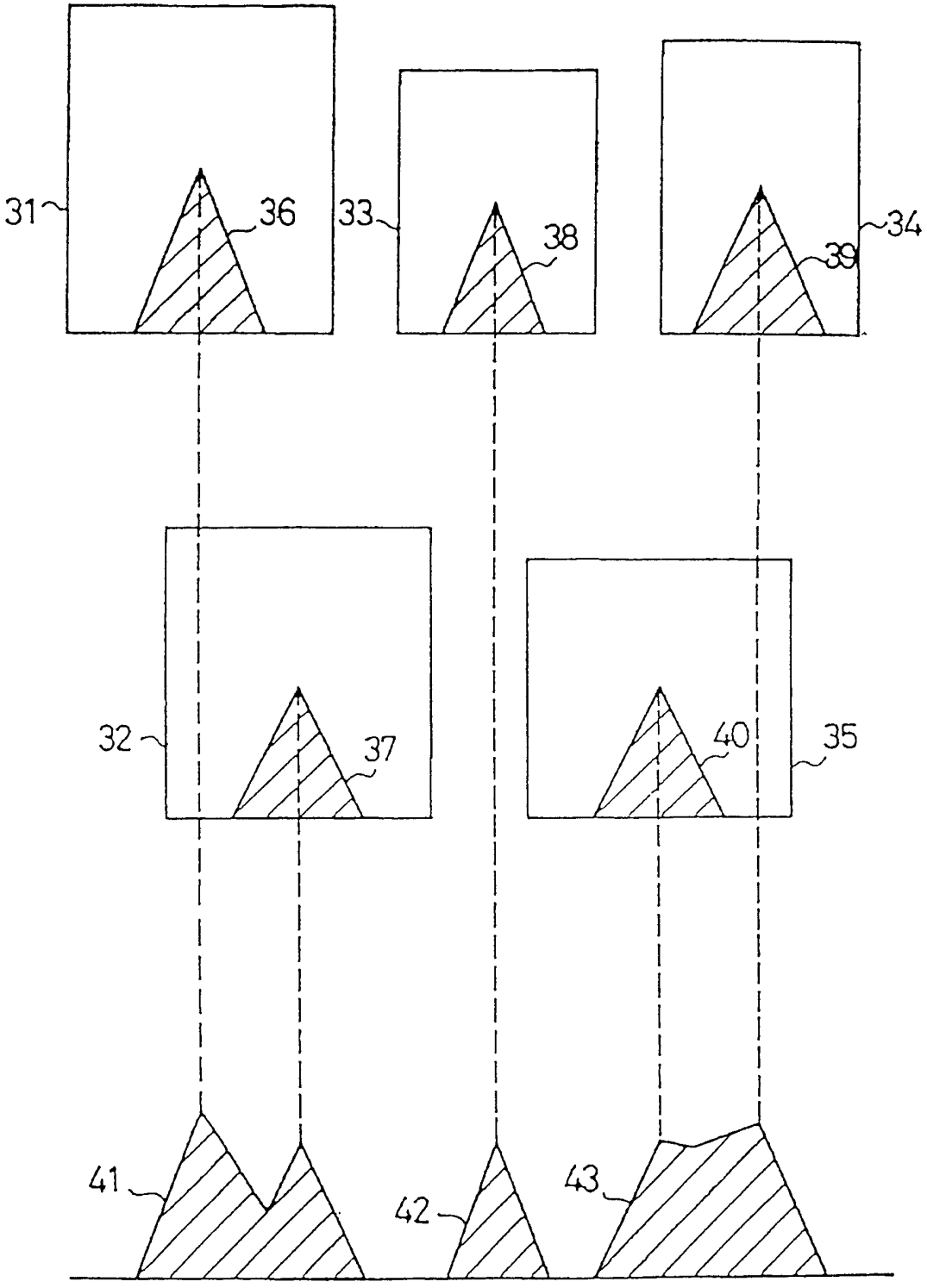


图. 17

图. 18

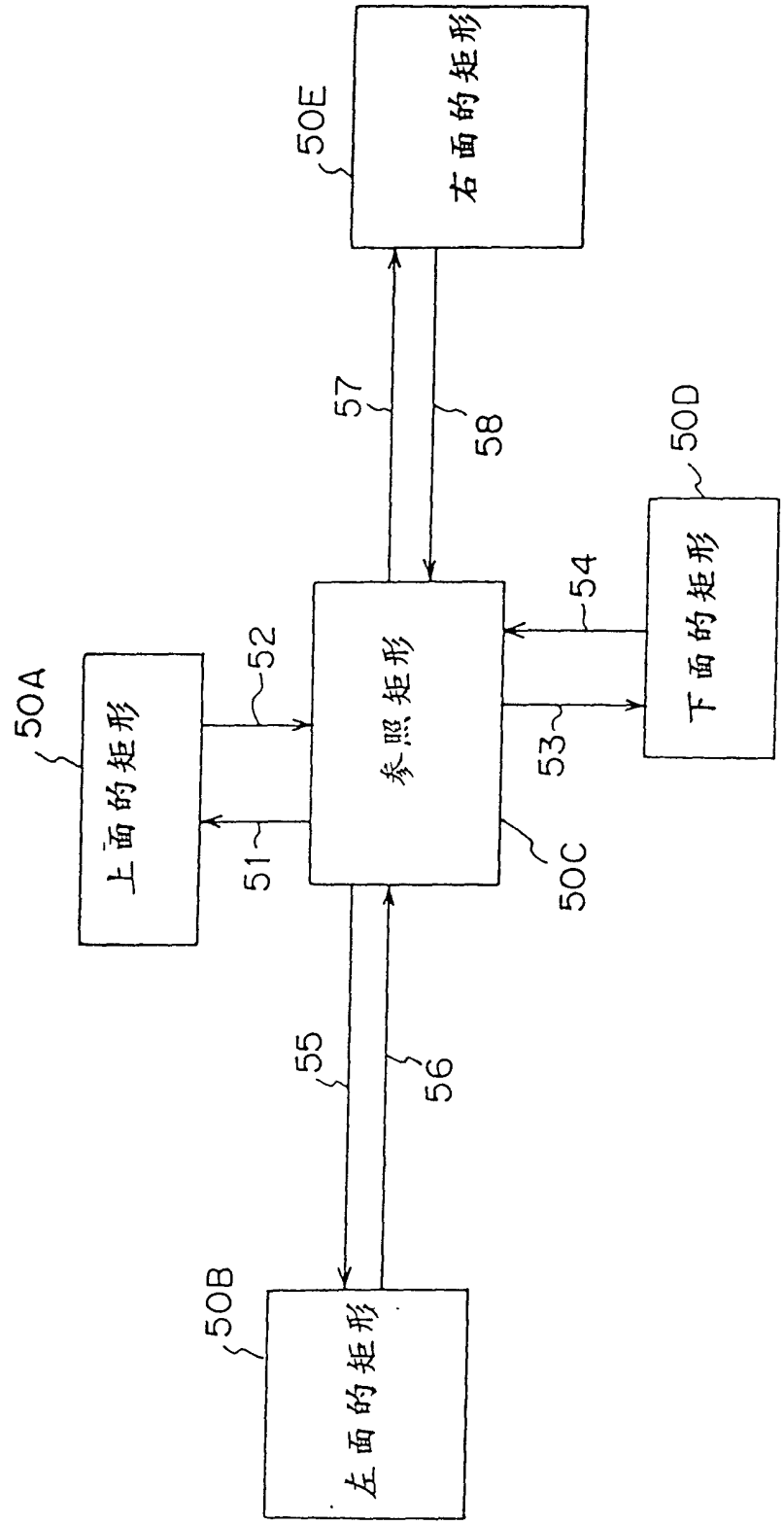

50E
右面的矩形
57
58
50D
下面的矩形
54
50A
上面的矩形
52
参照矩形
53
51
50C
55
56
50B
左面的矩形

图. 19

参照矩形的表数值
到上面矩形的指针
从上面矩形来的指针
到下面矩形的指针
从下面矩形来的指针
到左面矩形的指针
从左面矩形来的指针
到右面矩形的指针
从右面矩形来的指针

图 20

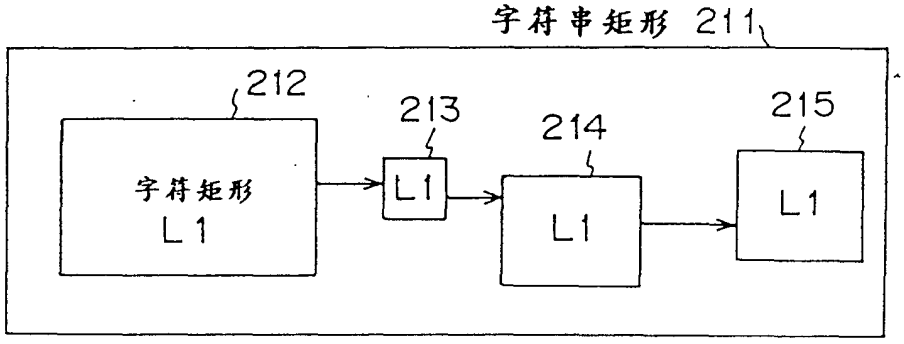


图. 21

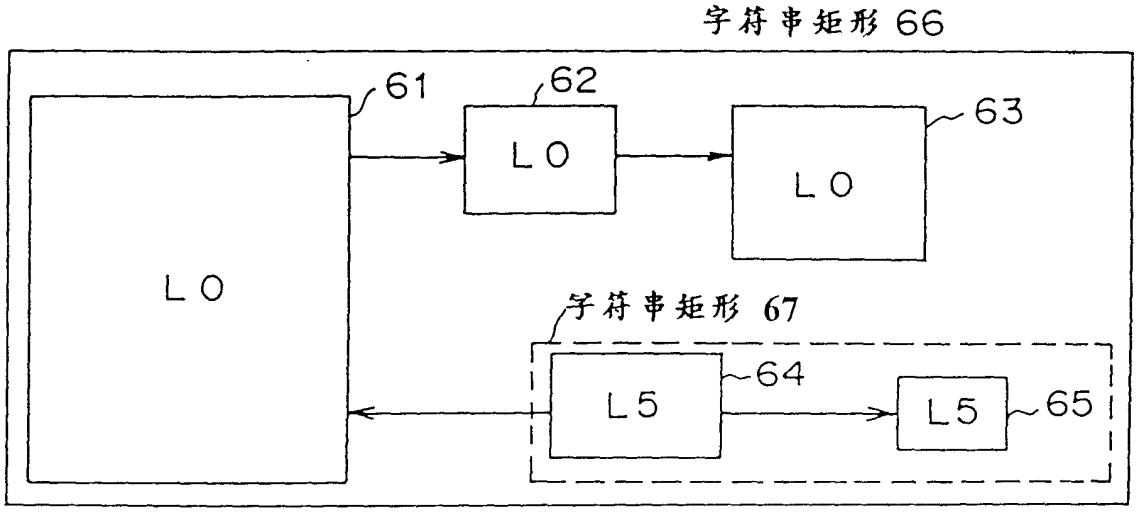


图. 22

图. 23


开始
除去噪声矩形
S41
生成表示字符串矩形的连通关系的连通关系表
S42
一致化两个或多个
字符串矩形为一个字符串矩形
S43
产生字符串矩形
高度的直方图并获得高度的最大频率数值
S44
获得除去了噪声矩形的文档区域并存储坐标
S45
垂直排序字符串矩形
S46
产生其余字符串矩形的连通关系表
S47
通过框架矩形向周围的定符串矩形设置框架标志
S48
向定界线矩形设置定界线标志
S49
当定界线矩形恰好分布在字符串矩形下面时，
向该 字符串矩形设置下划线标志
S50
结束

图. 24

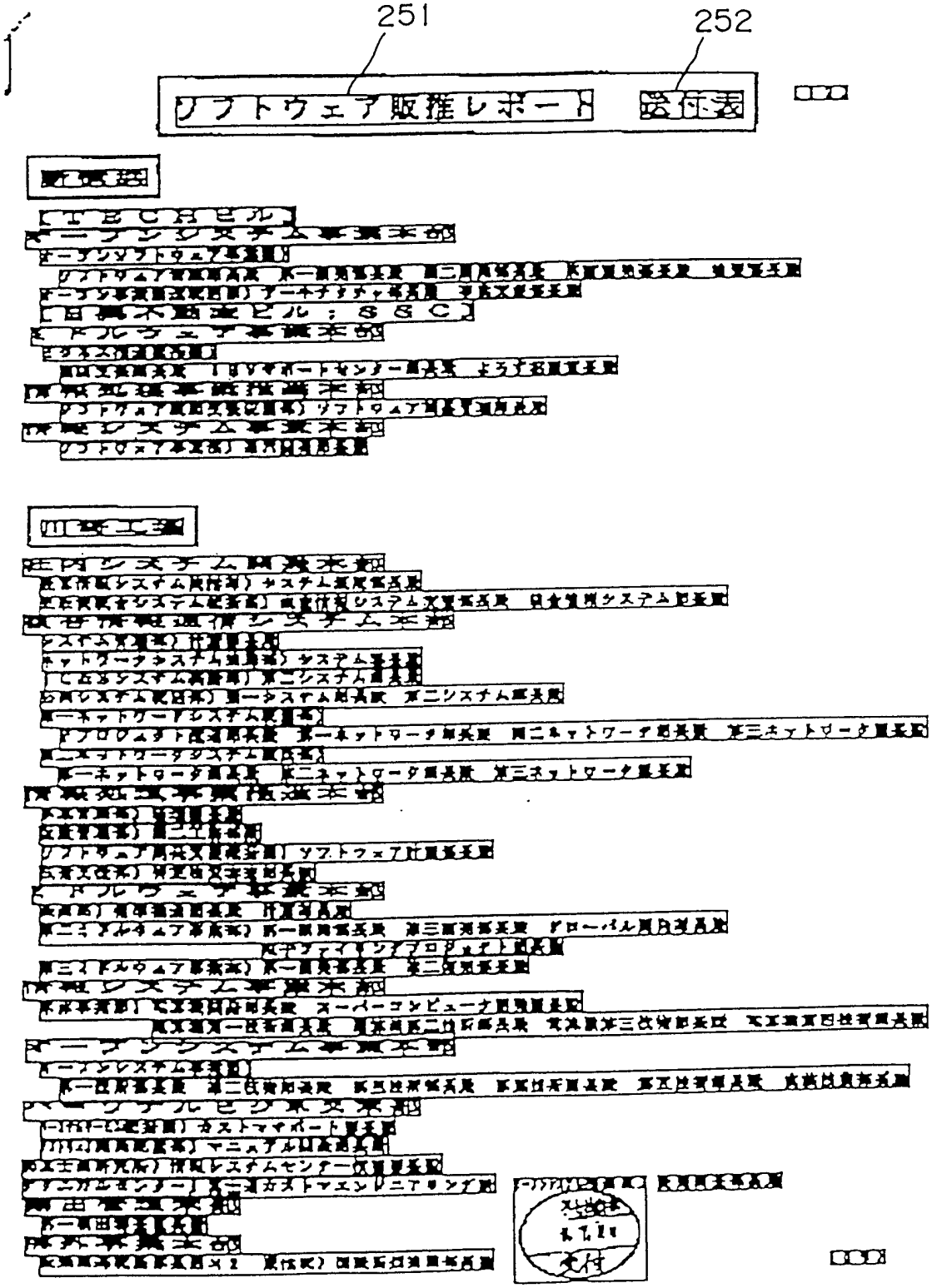


图. 25

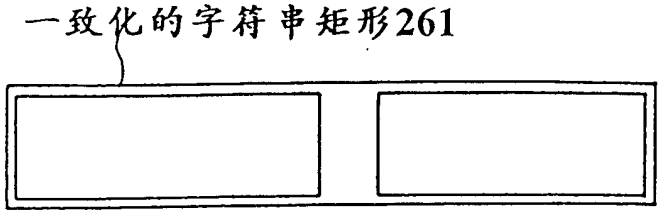


图 26A

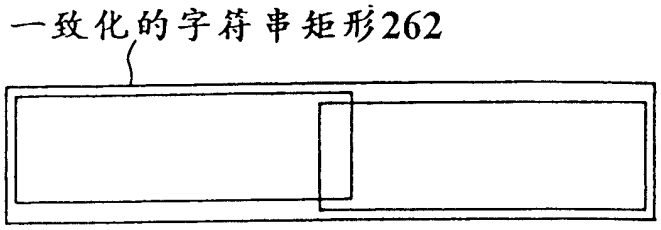


图 26B

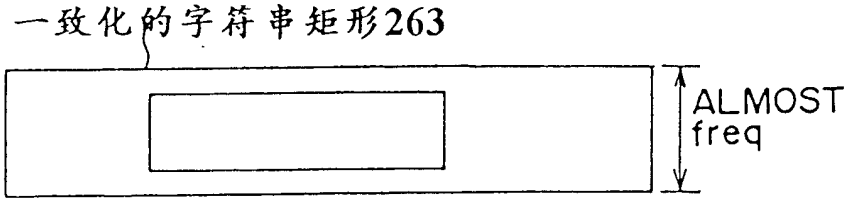


图 26C

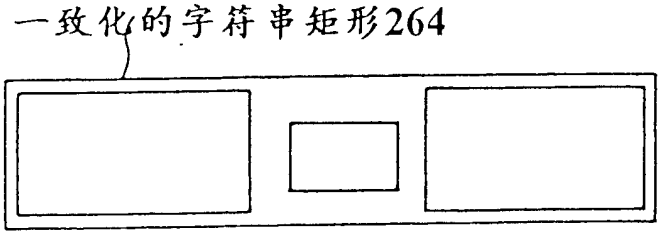


图 26D

图. 27

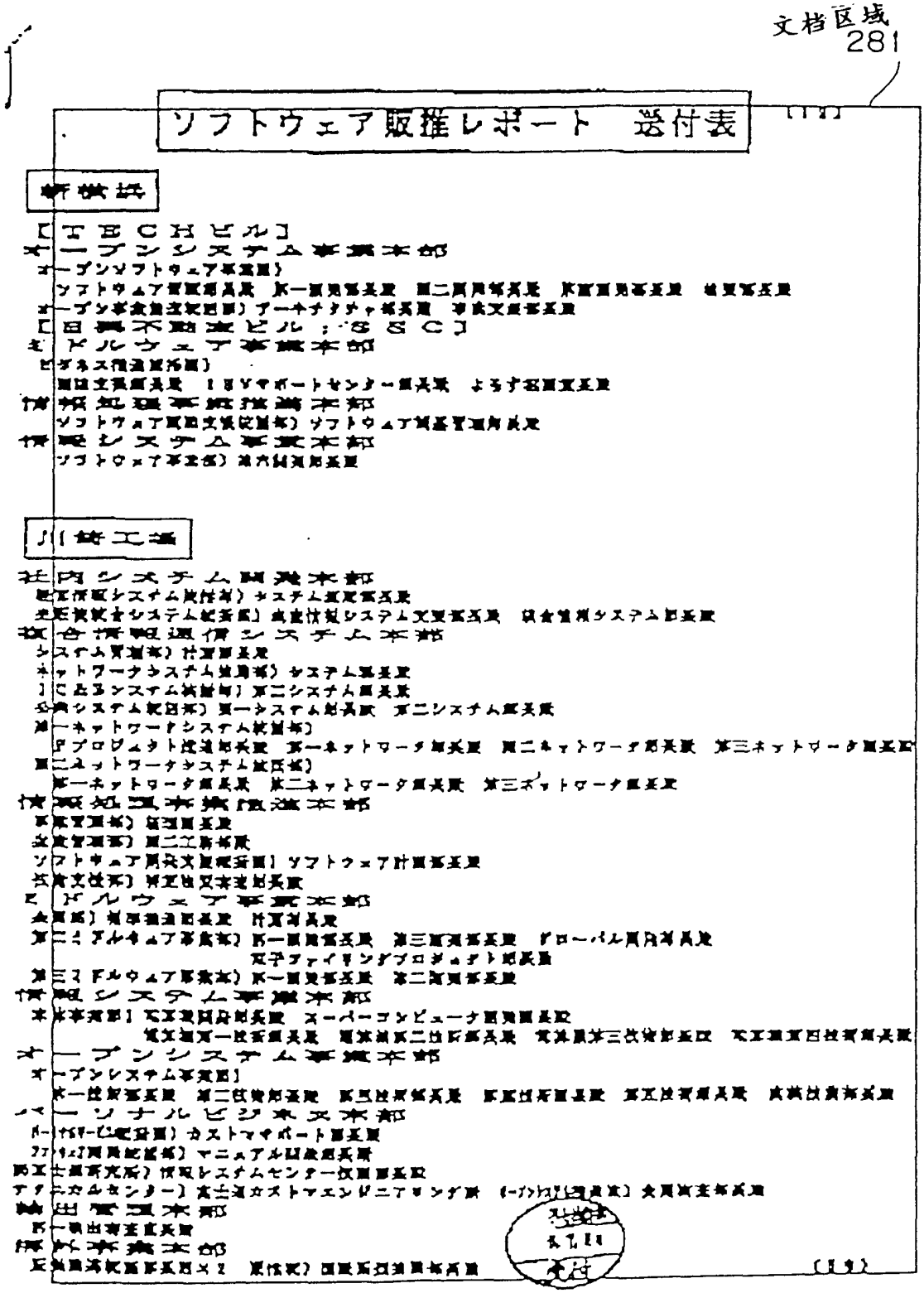

文档区域
281
ソフトウェア販推レポート 送付表
新横浜
[ＴＥＣＨビル]
オープンシステム事業本部
オープンソフトウェア事業部
[日興不動産ビル；ＳＳＣ]
ドルウェア事業本部
情報処理事業推進本部
情報システム事業本部
川崎工場
社内システム開発本部
情報システム技術部
情報処理事業推進本部
ソフトウェア開発支援室
ドルウェア事業本部
情報システム事業本部
オープンシステム事業本部
ソナルビジネス本部
輸出管理本部
海外事業本部

图. 28

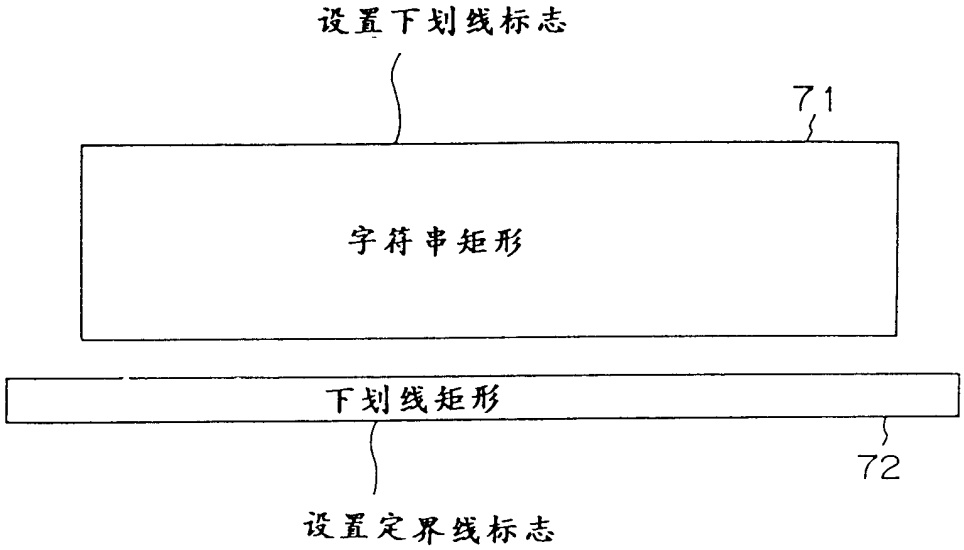


图. 29

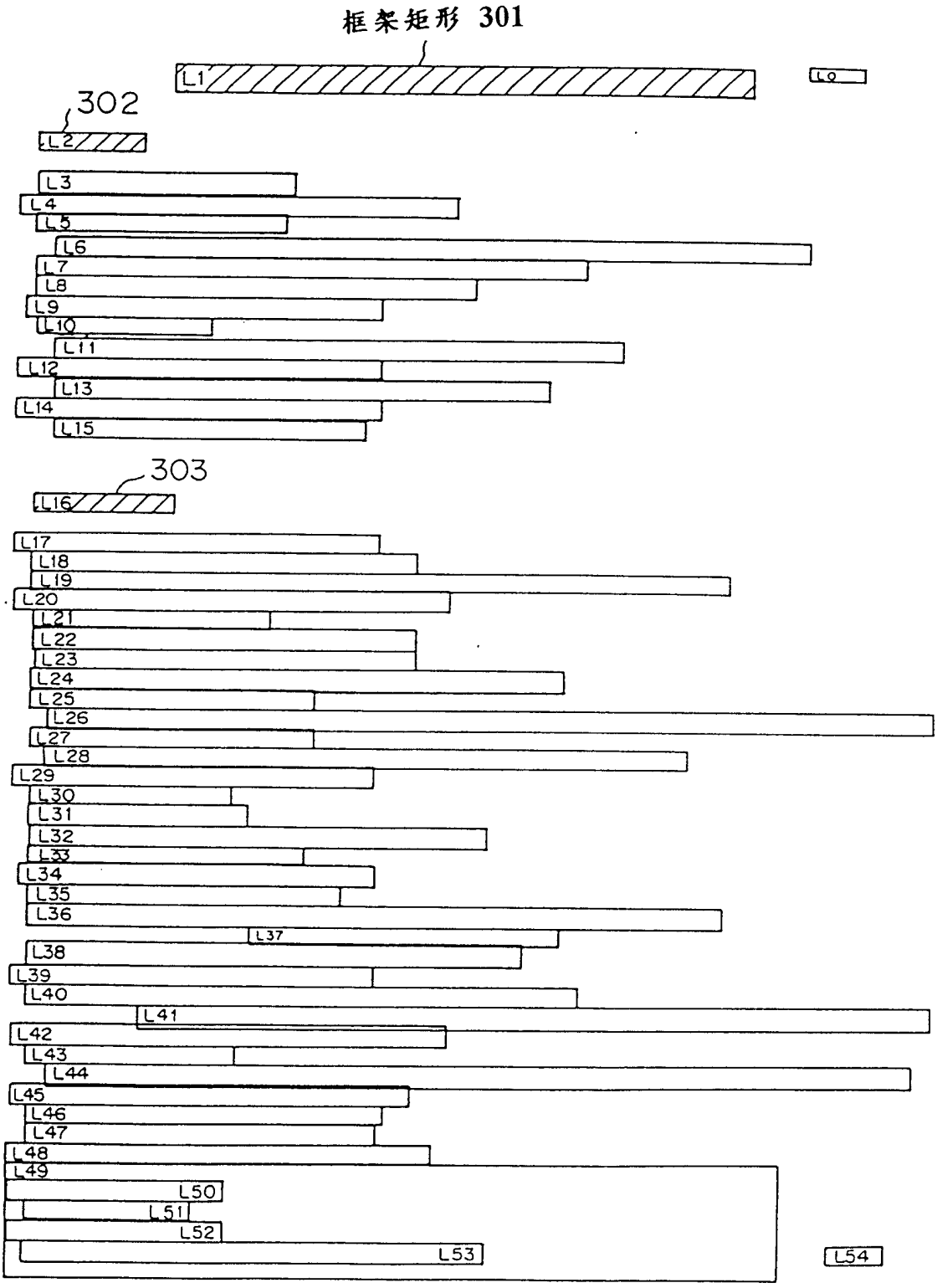

框架矩形 301
L1
L2
302
L3
L4
L5
L6
L7
L8
L9
L10
L11
L12
L13
L14
L15
303
L16
L17
L18
L19
L20
L21
L22
L23
L24
L25
L26
L27
L28
L29
L30
L31
L32
L33
L34
L35
L36
L37
L38
L39
L40
L41
L42
L43
L44
L45
L46
L47
L48
L49
L50
L51
L52
L53
L54

图. 30

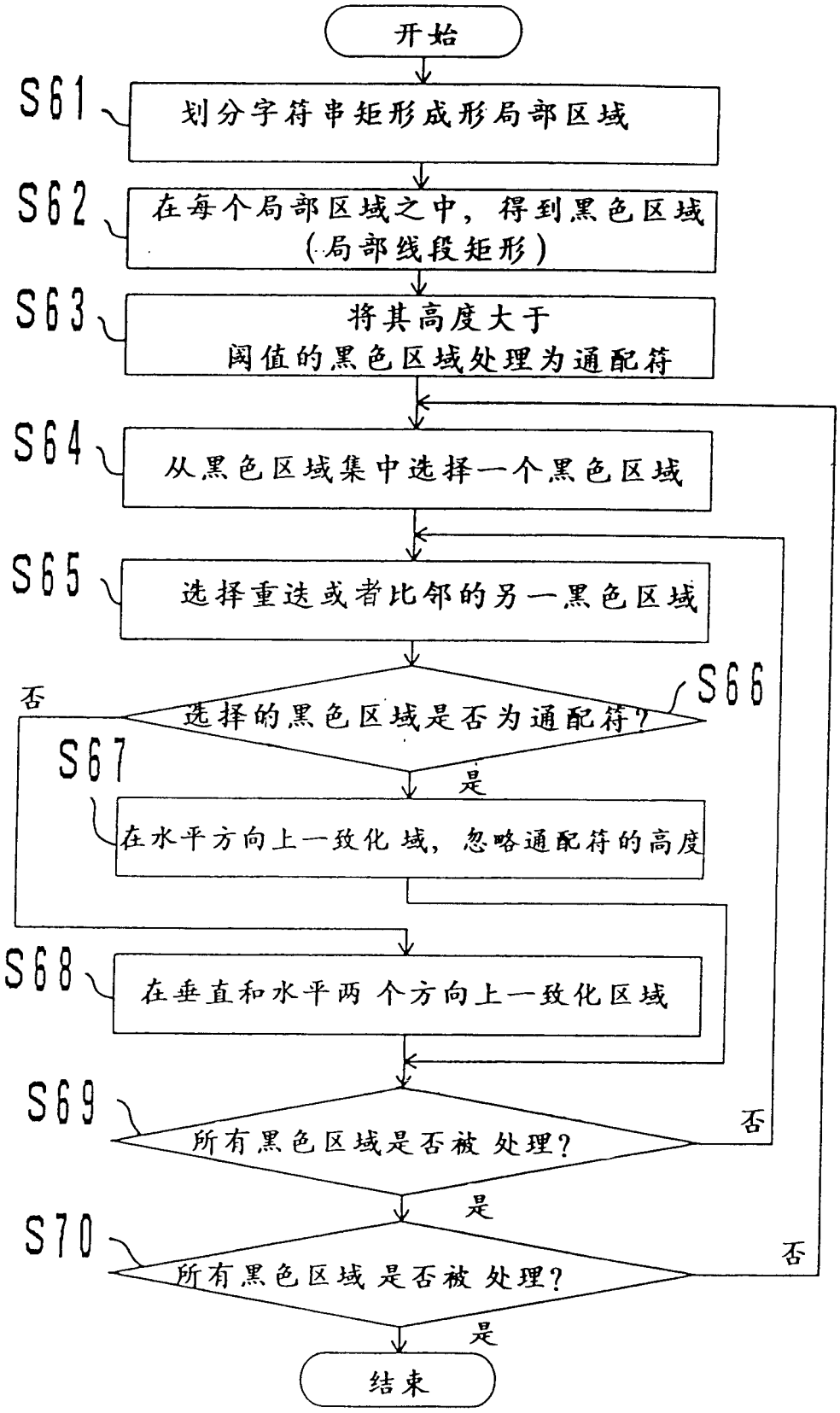

开始
S61
划分字符串矩形成形局部区域
S62
在每个局部区域之中，得到黑色区域
(局部线段矩形)
S63
将其高度大于
阈值的黑色区域处理为通配符
S64
从黑色区域集中选择一个黑色区域
S65
选择重迭或者比邻的另一黑色区域
S66
选择的黑色区域是否为通配符?
否
是
S67
在水平方向上一致化域，忽略通配符的高度
S68
在垂直和水平两个方向上一致化区域
S69
所有黑色区域是否被处理?
否
是
S70
所有黑色区域是否被处理?
否
是
结束

图. 31

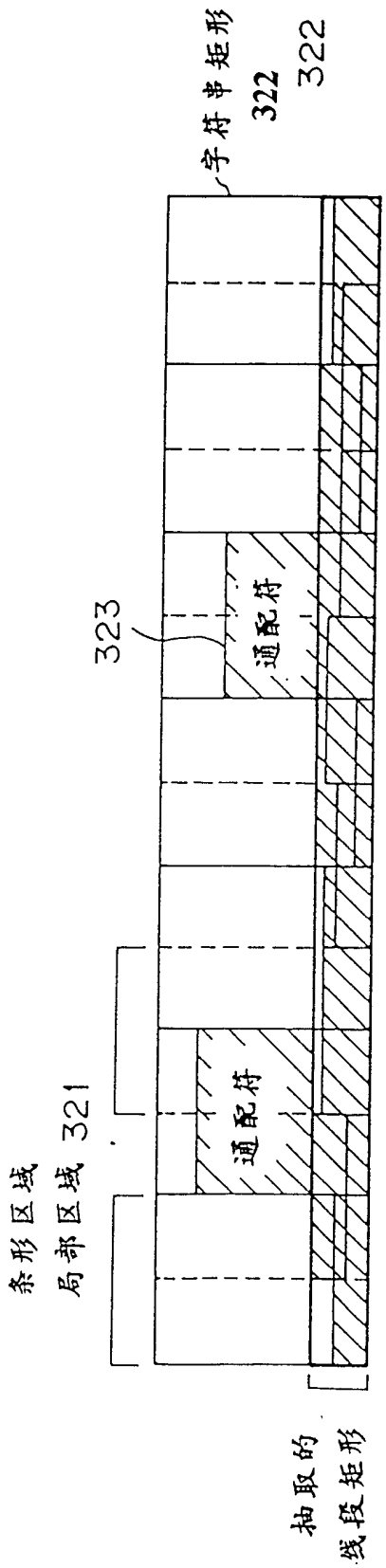


图. 32

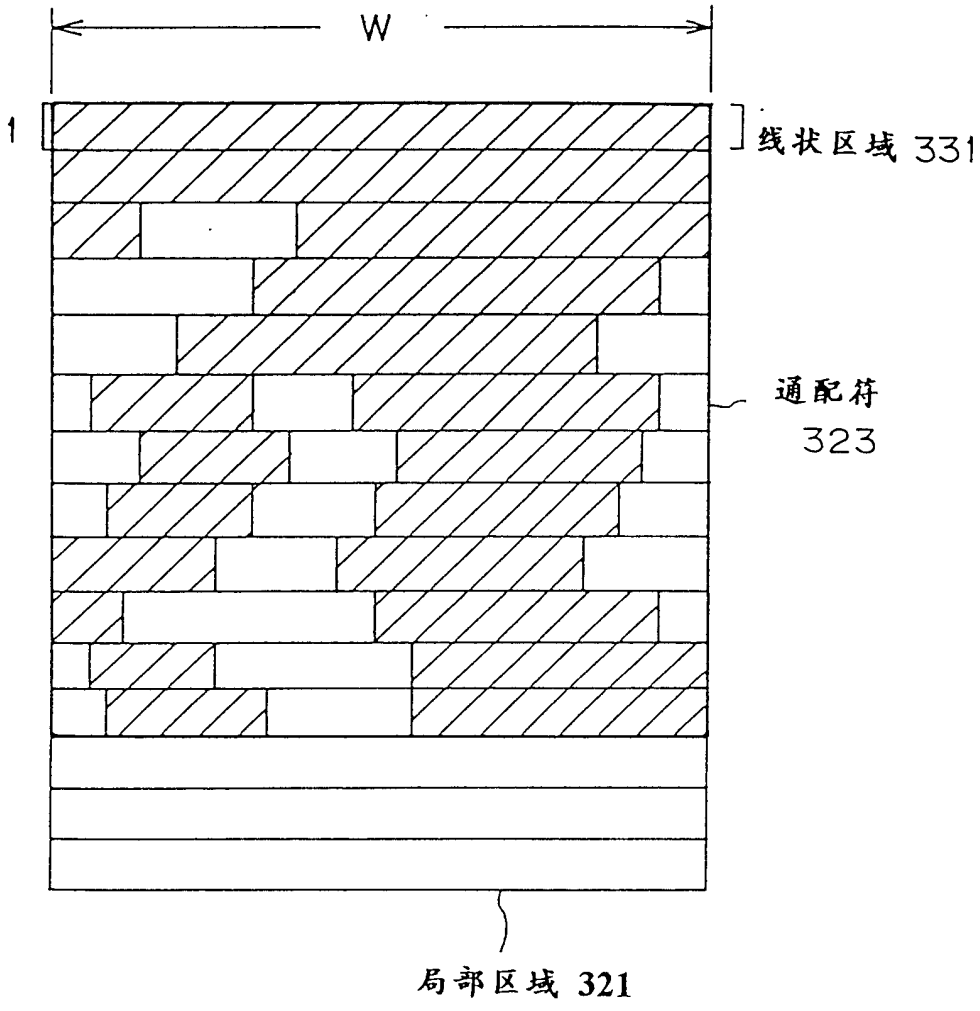


图. 33

```
MARK A PARTIAL RECTANGLE WITH A HEIGHT THAT IS (HEIGHT OF CHARACTER STRING
      RECTANGLE x0.3) OR LARGER AS A WILD CARD PARTIAL RECTANGLE (use=9).
OTHERWISE, MARK A PARTIAL RECTANLGE AS A STANDARD RECTANGLE (use=0).

1num=0

THE FOLLOWING PROCESS IS PERFORMED FOR ALL PARTIAL RECTANGLES.
    CURRENT RECTANGLE :i {

if ((use OF RECTANGLE i IS 0) OR ( use OF RECTANGLE i IS 9)) {
xlf =              COORDINATE AT LEFT EDGE OF RECTANGLE i
xr  =              COORDINATE AT RIGHT EDGE OF RECTANGLE i
yup = line_starty = COORDINATE AT UPPER EDGE OF RECTANGLE i
ybl = line_endy   = COORDINATE AT LOWER EDGE OF RECTANGLE i

if(use OF RECTANGLE i IS 0) {
  standard_st=yup;
  standard_en=ybl;
  standard_h=ybl-yup+1;
  b_use=0; /* THE FIRST SEGMENT HAS BEEN SET AS a standard RECTANGLE
                                              RATHER THAN a wild card. */
  height=ybl-yup+1;
  use OF RECTANGLE i=1;
}
else { /* IN THE CASE OF use:9 */
  standard_st=0;
  standard_en=0;
  standard_h=0;
  b_use=9; /* THE FIRST SEGMENT HAS BEEN SET AS a wild card
                                    RATHER THAN a standard RECTANGLE. */
  height2=ybl-yup+1;
  height=0;
}
THE FOLLOWING PROCESS IS PERFORMED FOR ALL PARTIAL RECTANGLES.
   CURRENT RECTANGLE :k {
C1 ⟶ α
C2 ⟶ β
}
/* IN THE CASE THAT ALL SEGMENTS ARE wild card SEGMENTS. */
if( (b_use IS 9) ) {
  /* THE HEIGHT OF THE FIRST SEGMENT IS SUBSTITUTED
                              WITH THE HEIGHT OF A LONG SEGMENT. */
  height=height2;
}

THE COORDINATE OF THE OBTAINED SEGMENT(LEFT EDGE : x1f, RIGHT EDGE : xr,
UPPER EDGE : line_starty, LOWER EDGE : line_endy) ARE STORED
                                        AT lnum IN yokoline.

lnum IS INCREMENTED BY ONE.
}
}
```

图. 34

```
        rxlr = COORDINATE AT LEFT EDGE OF RECTANGLE k
        rxr  = COORDINATE AT RIGHT EDGE OF RECTANGLE k
        ryup = COORDINATE AT UPPER EDGE OF RECTANGLE k
        rybl = COORDINATE AT LOWER EDGE OF RECTANGLE k
        rheight=rybl-ryup+1;

        /* A VALUE OF a standard RECTANGLE HAS BEEN SET. */
        if( (b_use IS 0) ) {
          /* THE CURRENT RECTANGLE IS a standard RECTANGLE.
                THE RIGHT-SIDE RECTANGLE IS a wild card. */
         if (use OF RECTANGLE k IS 9) {

            /* THE CURRENT RECTANGLE OVERLAPS WITH THE RIGHT SIDE RECTANGLE
               FOR 1 dot OR MORE IN THE HORIZONTAL AND VERTICAL DIRECTIONS. */
           if( (xr+1)>=rxlf) && (xr< rxr) ) &&
             ( (ybl+1)>=ryup) && ( (yup-1)<=rybl) ) {
             xr      = rxr;
           }
         }
         /* THE CURRENT RECTANGLE IS a standard RECTANGLE.
                THE RIGHT SIDE RECTANGLE IS NOT a wild card. */
        else if( use OF RECTANGLE k IS 0) {
        /* THE CURRENT RECTANGLE OVERLAPS WITH THE RIGHT SIDE RECTANGLE
               FOR 1 dot OR MORE IN THE HORIZONTAL AND VERTICAL DIRECTIONS. */
       if( ((xr+1) >=rxlf) && (xr <rxr) &&
         ((ybl+1)>=ryup && ((yup-1) <=rybl) &&
         (standard_h-TH_HEIGHTDOT   <= rheight) &&
         (rheight<=standard_h+TH_HEIGHTDOT) )  {
         use OF RECTANGLE k = 2;
         xr        =  rxr;
         yup       =  ryup;
         ybl       =  rybl;
         hei=rybl-ryup+1;
         if(hei>height)  {
           height=hei;
         }
         if(ryup<line_starty)
           line_starty=ryup;
         if(rybl>line_endy)
           line_endy=rybl;

         standard_h=hei;
       }
     }
   }
```

图. 35

```
/* A VALUE OF a standard RECTANGLE HAS NOT BEEN SET. */
else if((b_use ==9) ) {
  /* THE CURRENT RECTANGLE IS NOT a standard RECTANGLE.
        THE RIGHT SIDE RECTANGLE IS a wild card. */
  if (use OF RECTANGLE k IS 9) {

     /* THE CURRENT RECTANGLE OVERLAPS WITH THE RIGHT SIDE RECTANGLE
           FOR 1 dot OR MORE IN THE RIGHT SIDE RECTANGLE FOR 1 dot
              OR MORE IN THE HORIZONTAL DIRECTION. */
     if( (xr+1) >=rxlf) && (xr < rxr) ) {
        xr     =rxr;
    }
  }
  /* THE CURRENT RECTANGLE IS NOT a standard RECTANGLE.
        THE RIGHT SIDE RECTANGLE IS NOT a wild card RECTANGLE. */
  else if(use OF RECTANGLE k IS 0) {

    /* THE CURRENT RECTANGLE OVERLAPS WITH THE RIGHT SIDE RECTANGLE
          FOR 1 dot OR MORE IN THE HORIZONTAL DIRECTION. */
    if( ((xr+1) >=rxlf) && (xr < rxr) ) {

      b_use=0; /* A VALUE OF a standard RECTANGLE HAS BEEN SET. */
      use OF RECTANGLE k = 2;
      xr      = rxr;
      yup     = ryup;
      ybl     = rybl;
      hei=rybl-ryup+1;
      if(hei>height) {
        height=hei;
      }
      standard_st=ryup;
      standard_en=rybl;
      standard_h-hei;
      line_starty=ryup;
      line_endy=rybl;
    }
  }
}
```

图. 36

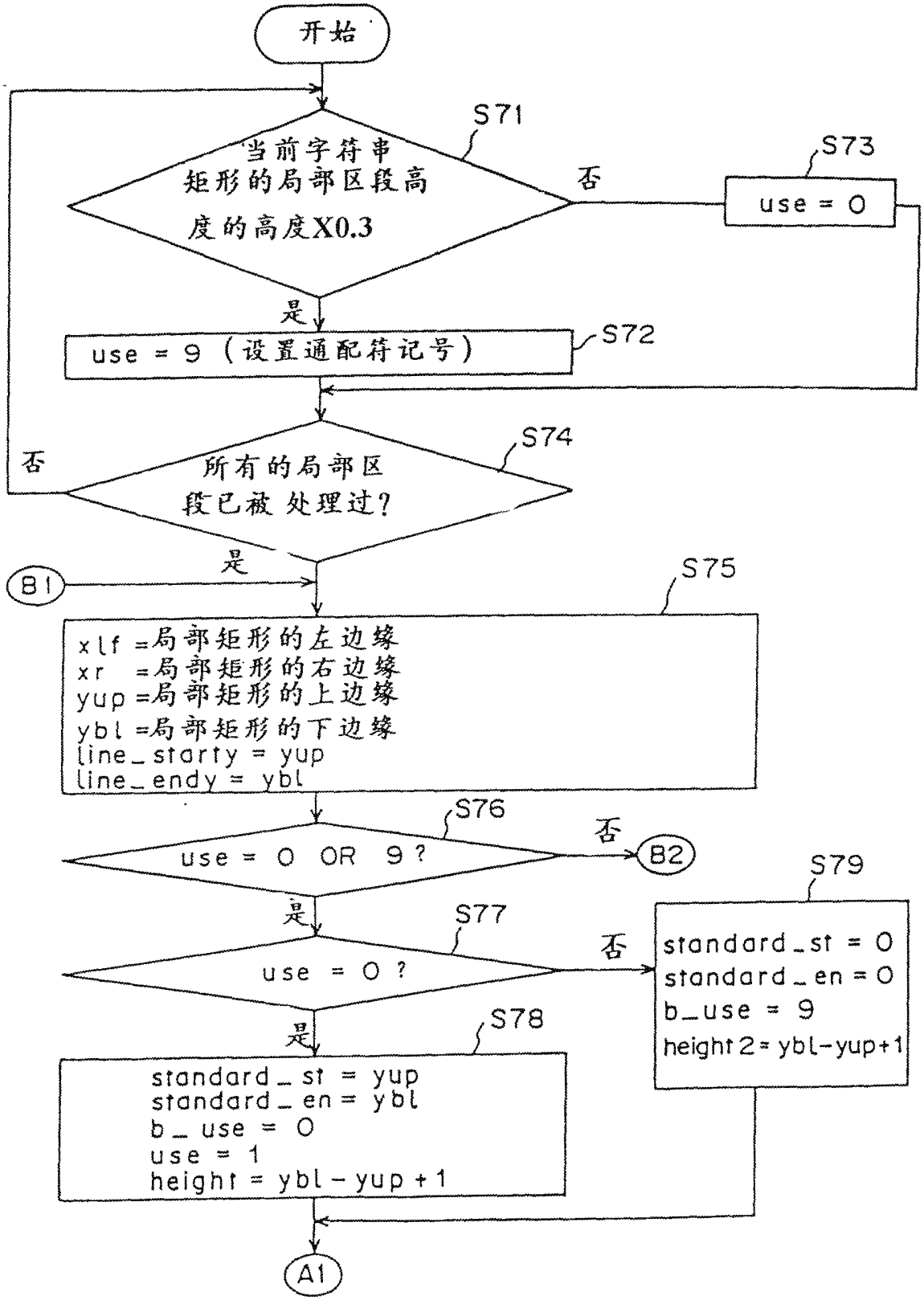

开始
S71
当前字符串矩形的局部区段高度的高度X0.3
否
S73
use = 0
是
use = 9 (设置通配符记号)
S72
S74
否
所有的局部区段已被 处理过?
是
B1
S75
xlf =局部矩形的左边缘
xr =局部矩形的右边缘
yup =局部矩形的上边缘
ybl =局部矩形的下边缘
line_starty = yup
line_endy = ybl
S76
use = 0 OR 9 ?
否
B2
是
S77
use = 0 ?
否
S79
standard_st = 0
standard_en = 0
b_use = 9
height2 = ybl-yup+1
是
S78
standard_st = yup
standard_en = ybl
b_use = 0
use = 1
height = ybl - yup + 1
A1

图 37

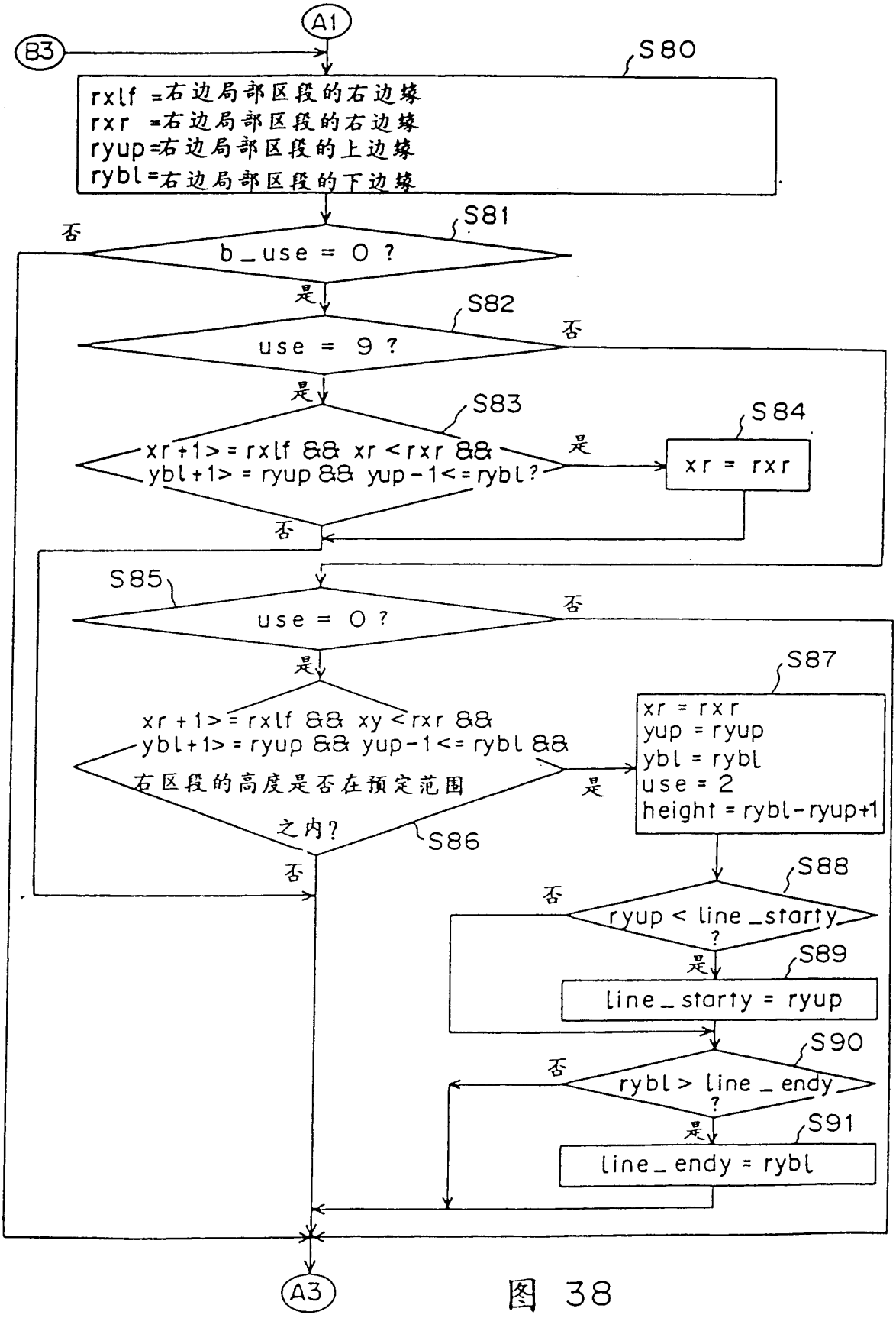

A1
B3
S80
rxlf =右边局部区段的右边缘
rxr =右边局部区段的右边缘
ryup=右边局部区段的上边缘
rybl=右边局部区段的下边缘
S81
否
b_use = 0 ?
是
S82
use = 9 ?
否
是
S83
xr+1> = rxlf && xr < rxr &&
ybl+1> = ryup && yup-1<=rybl?
是
S84
xr = rxr
否
S85
use = 0 ?
否
是
S87
xr+1> = rxlf && xy < rxr &&
ybl+1> = ryup && yup-1<= rybl &&
右区段的高度是否在预定范围
之内?
S86
是
xr = rxr
yup = ryup
ybl = rybl
use = 2
height = rybl-ryup+1
否
S88
ryup < line_starty ?
是
S89
line_starty = ryup
否
S90
rybl > line_endy ?
是
S91
line_endy = rybl
A3

图 38

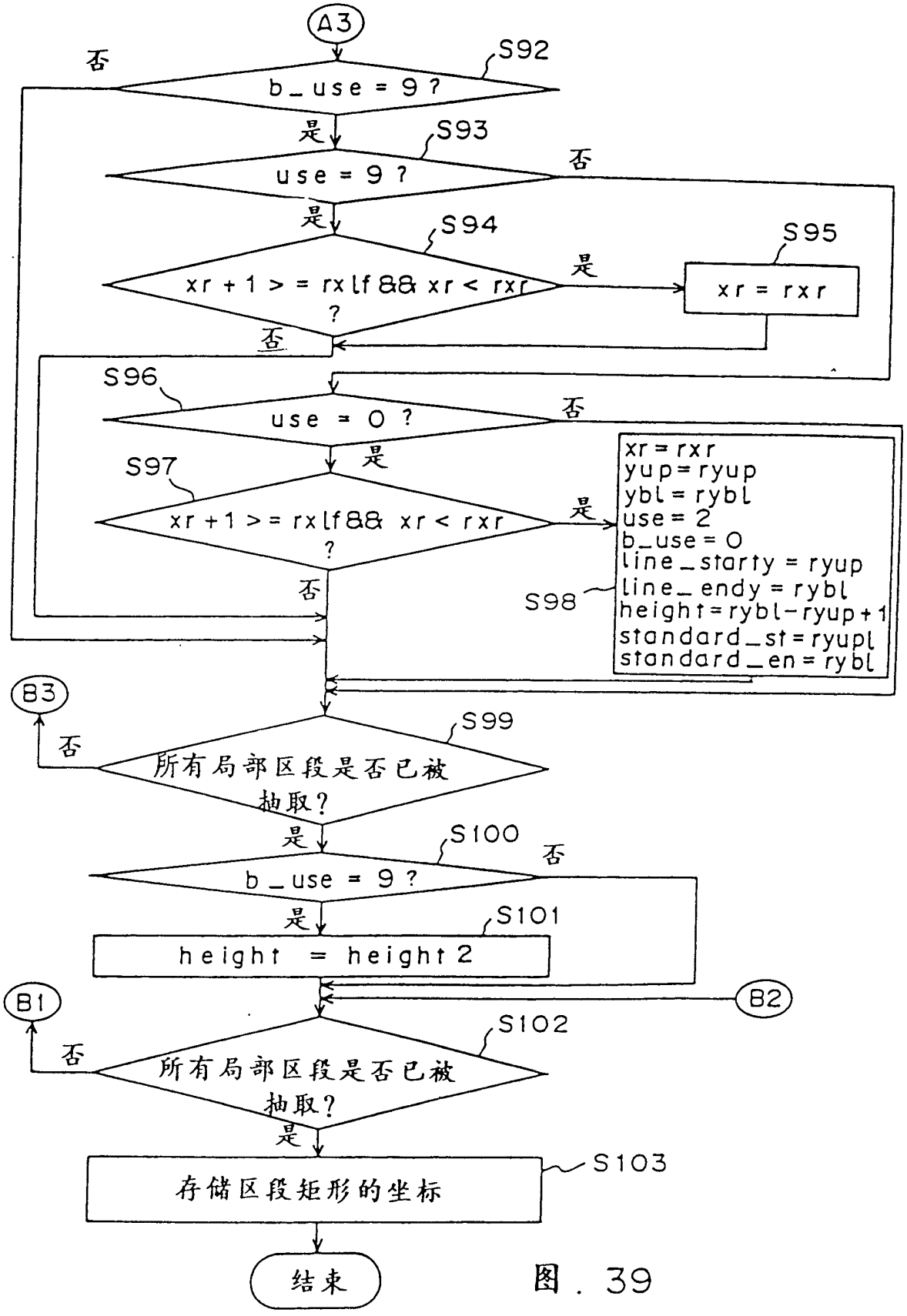

A3
S92
否
b_use = 9 ?
是
S93
use = 9 ?
否
是
S94
S95
xr + 1 > = rxlf && xr < rxr ?
是
xr = rxr
否
S96
use = 0 ?
否
是
S97
xr + 1 > = rxlf && xr < rxr ?
是
xr = rxr
yup = ryup
ybl = rybl
use = 2
b_use = 0
line_starty = ryup
line_endy = rybl
height = rybl - ryup + 1
standard_st = ryupl
standard_en = rybl
S98
否
B3
否
S99
所有局部区段是否已被抽取?
是
S100
b_use = 9 ?
否
是
S101
height = height 2
B2
B1
否
S102
所有局部区段是否已被抽取?
是
S103
存储区段矩形的坐标
结束

图 . 39

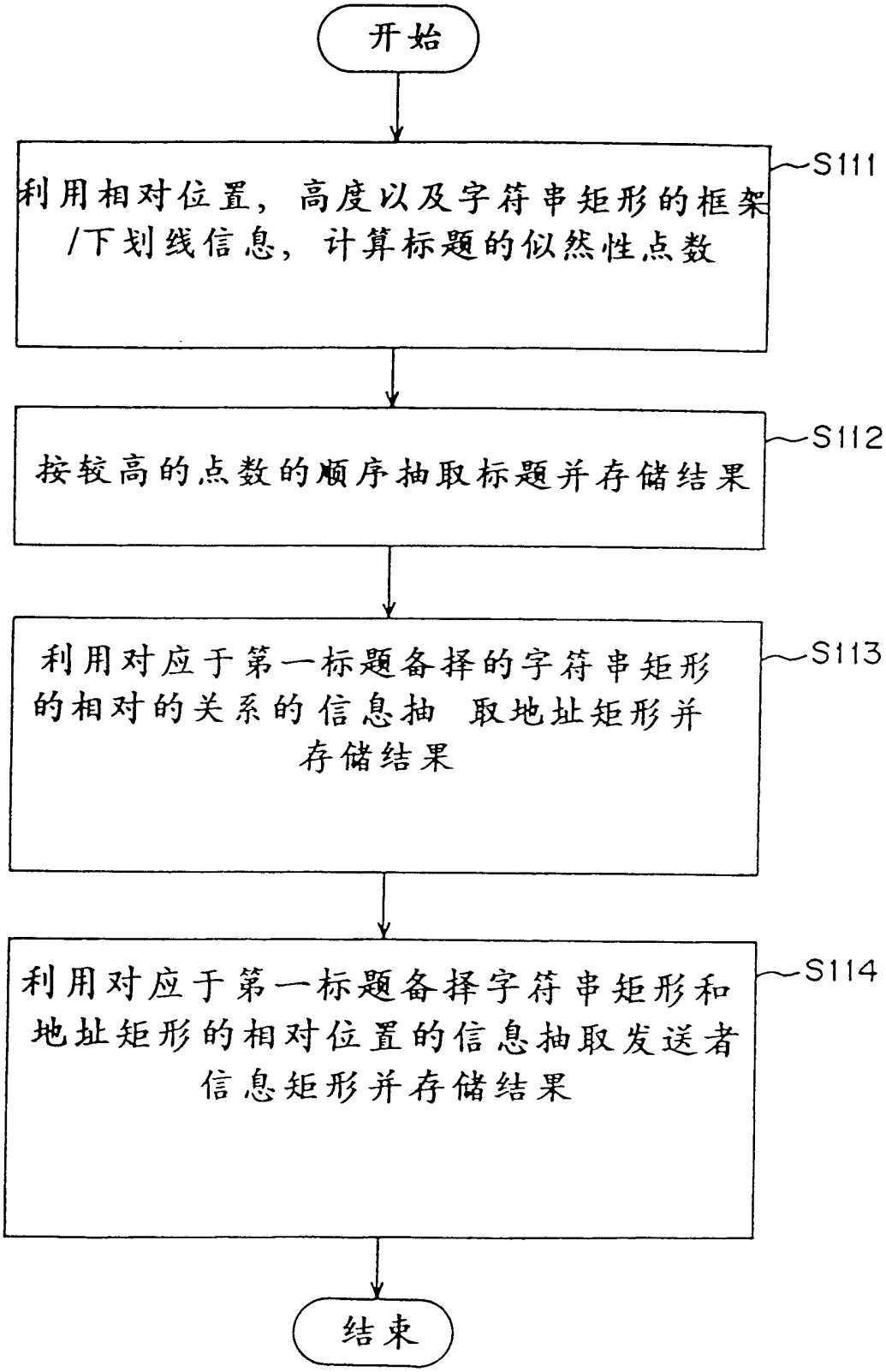

开始
利用相对位置，高度以及字符串矩形的框架/下划线信息，计算标题的似然性点数
S111
按较高的点数的顺序抽取标题并存储结果
S112
利用对应于第一标题备择的字符串矩形的相对的关系的信息抽 取地址矩形并存储结果
S113
利用对应于第一标题备择字符串矩形和地址矩形的相对位置的信息抽取发送者信息矩形并存储结果
S114
结束

图 . 40

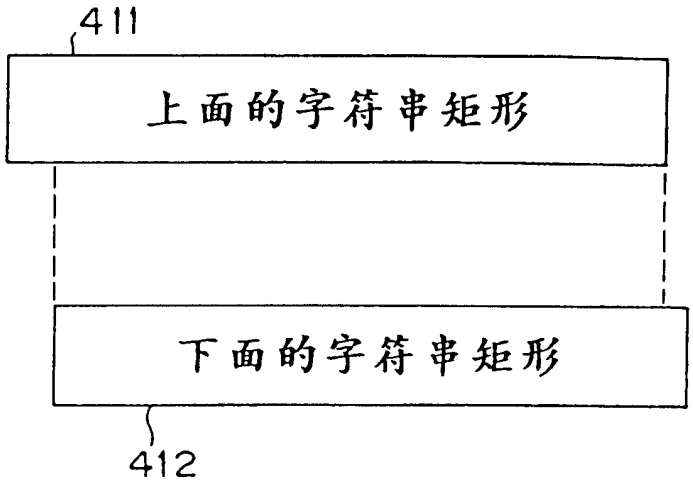


图 41A

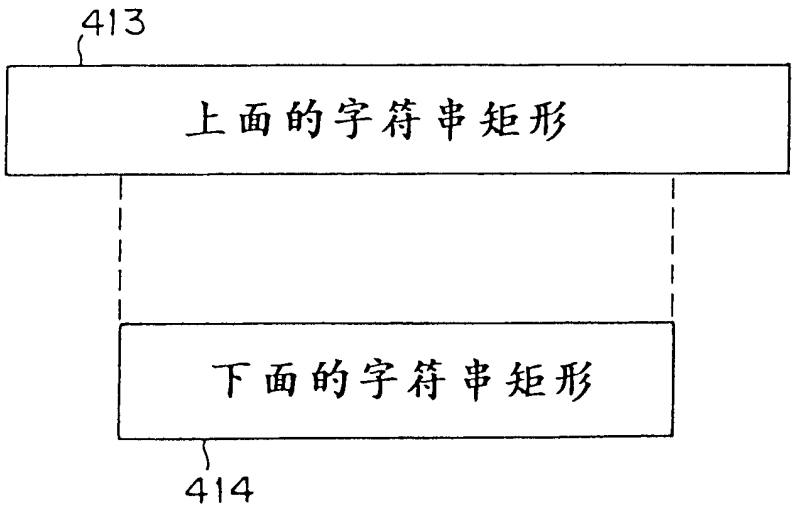


图 41B

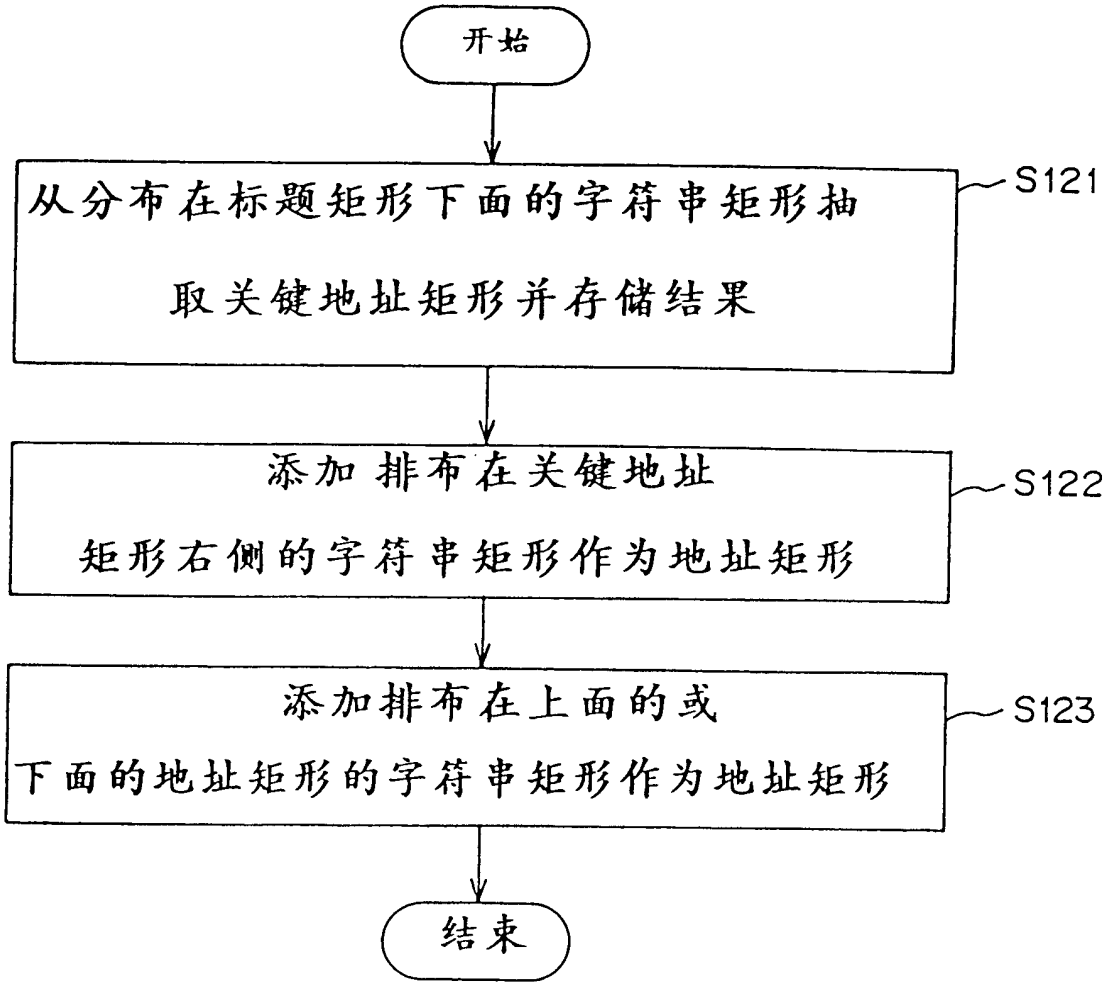

开始
从分布在标题矩形下面的字符串矩形抽取关键地址矩形并存储结果
S121
添加排布在关键地址矩形右侧的字符串矩形作为地址矩形
S122
添加排布在上面的或下面的地址矩形的字符串矩形作为地址矩形
S123
结束

图 42

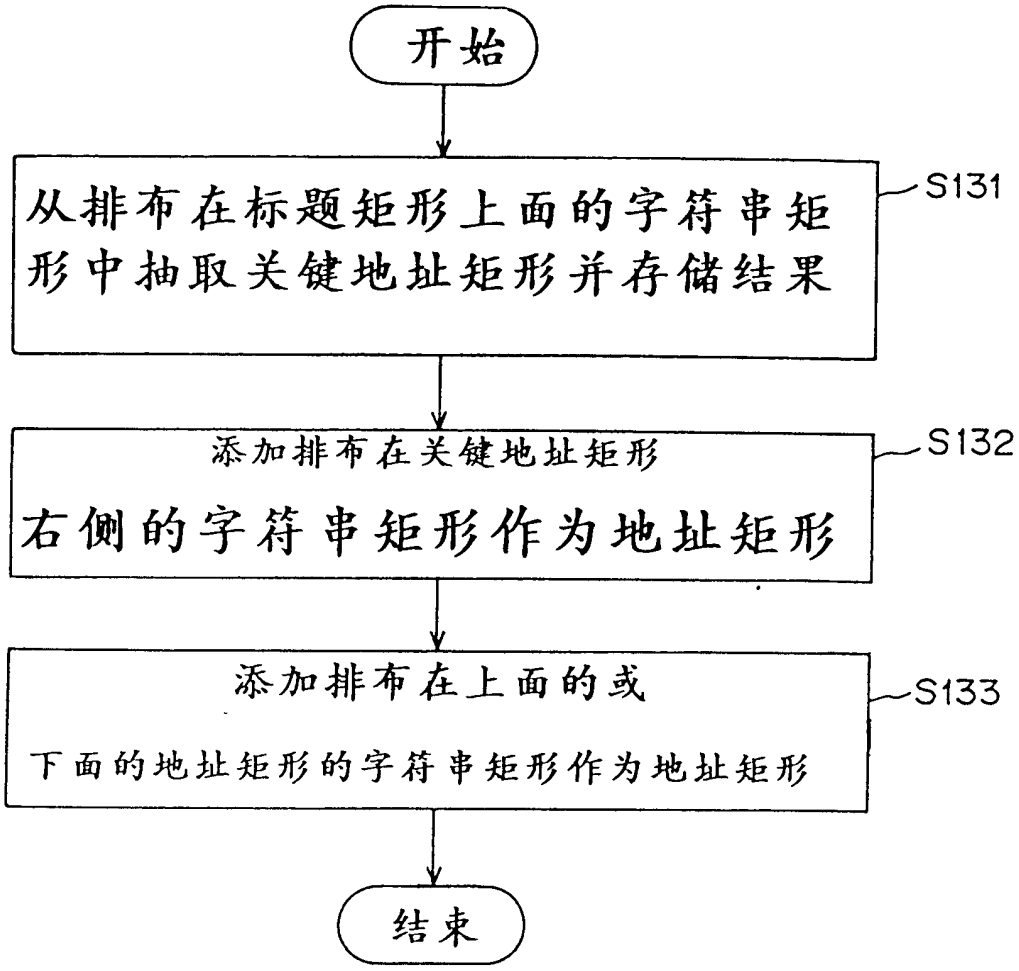

开始
从排布在标题矩形上面的字符串矩形中抽取关键地址矩形并存储结果
S131
添加排布在关键地址矩形右侧的字符串矩形作为地址矩形
S132
添加排布在上面的或下面的地址矩形的字符串矩形作为地址矩形
S133
结束

图 43

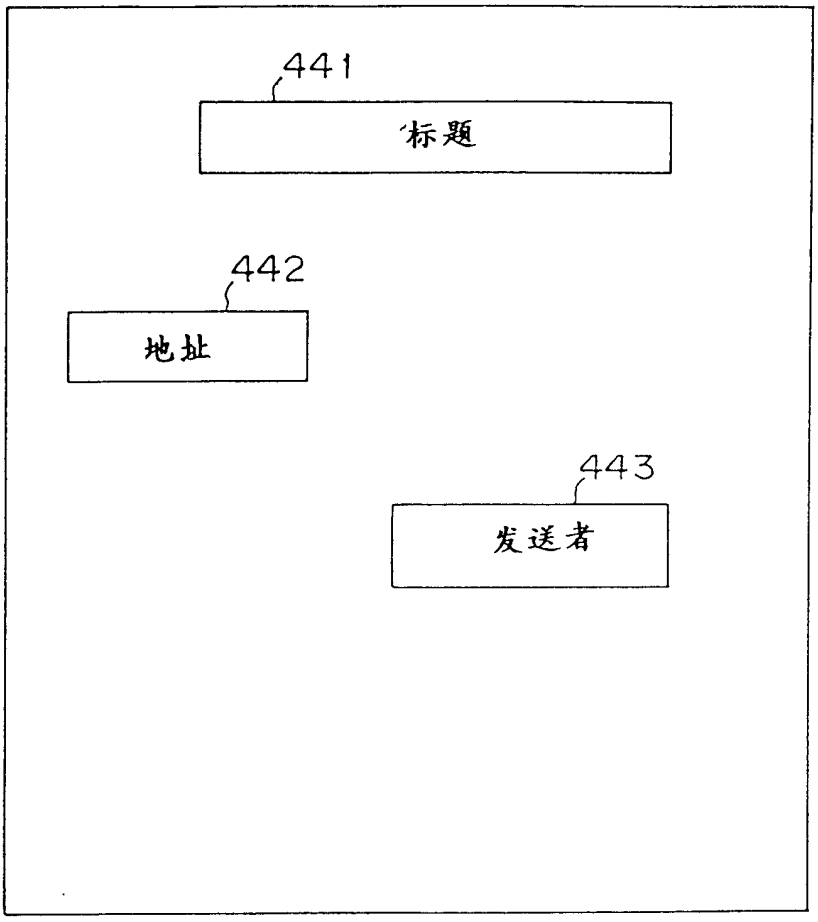


图. 44

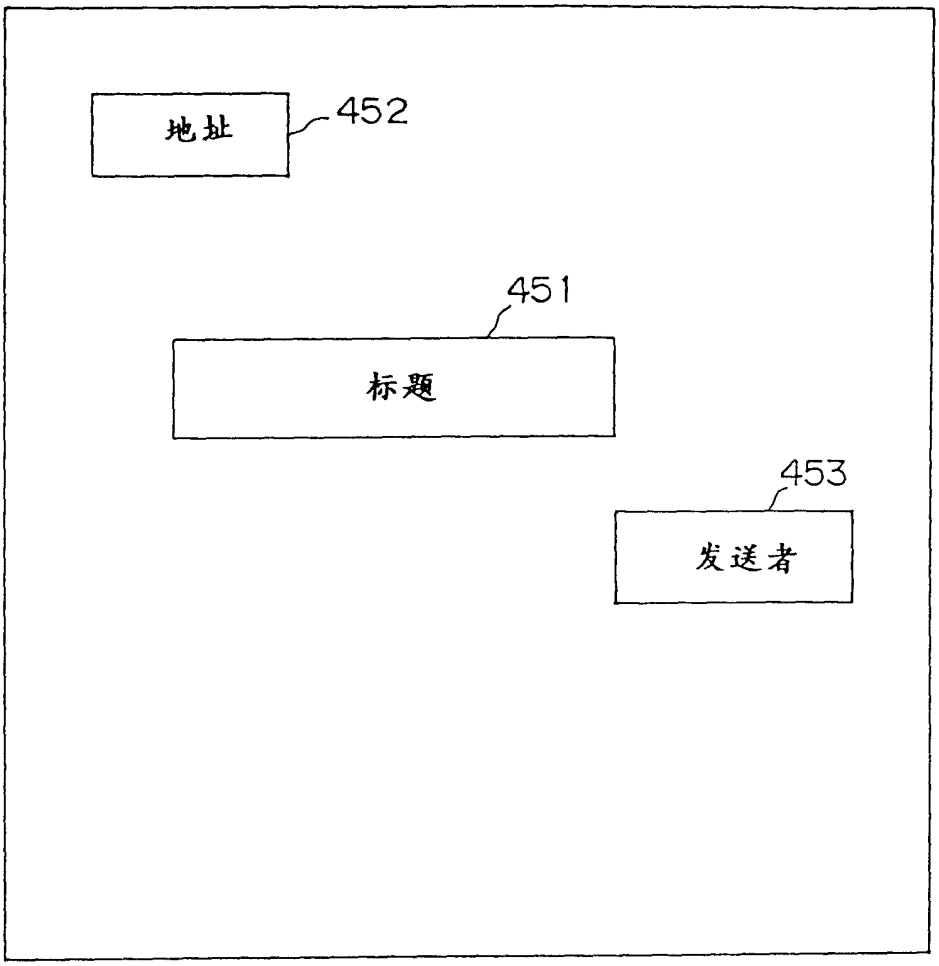


图. 45

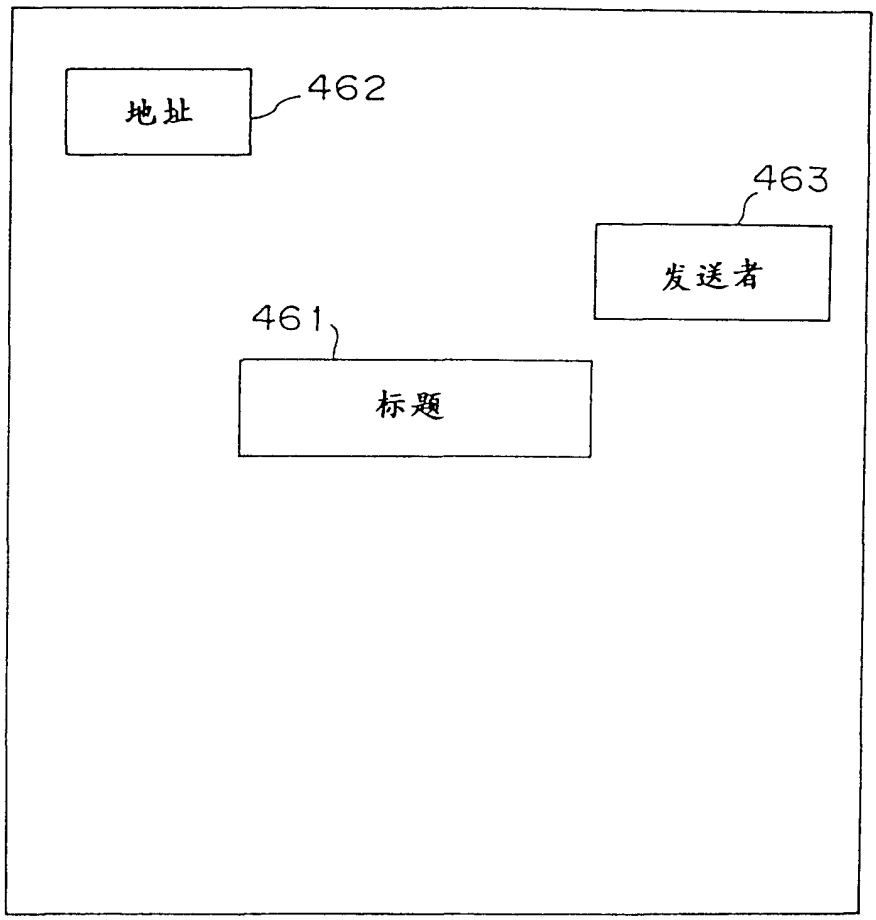


图. 46

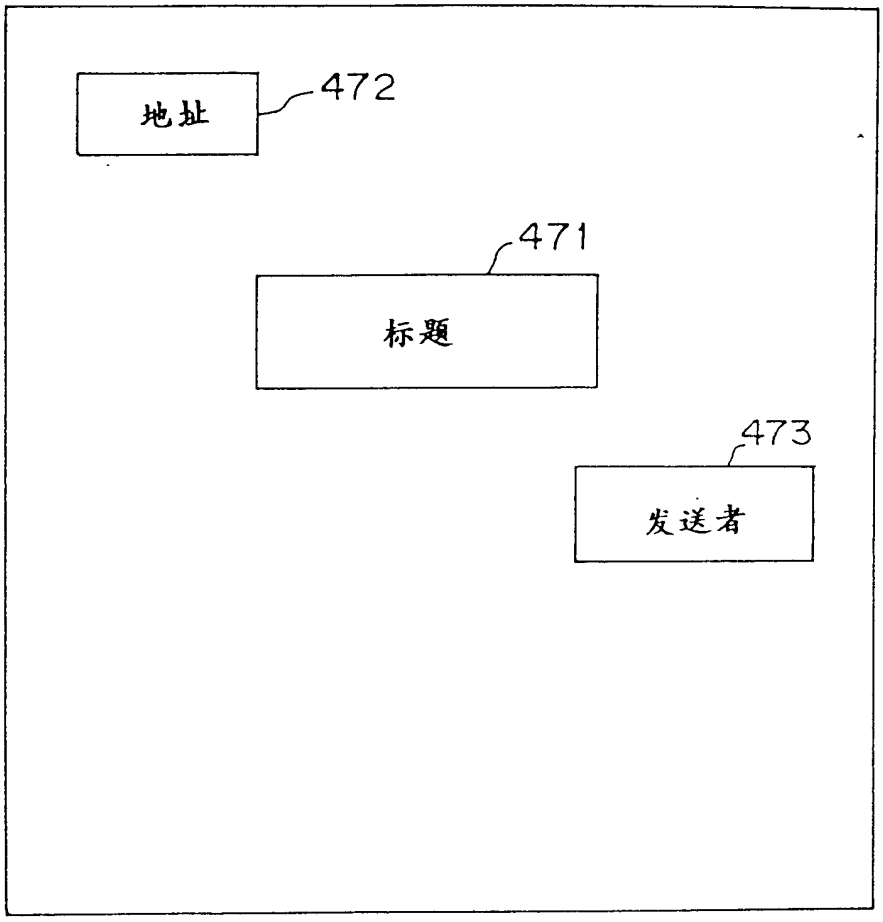


图. 47

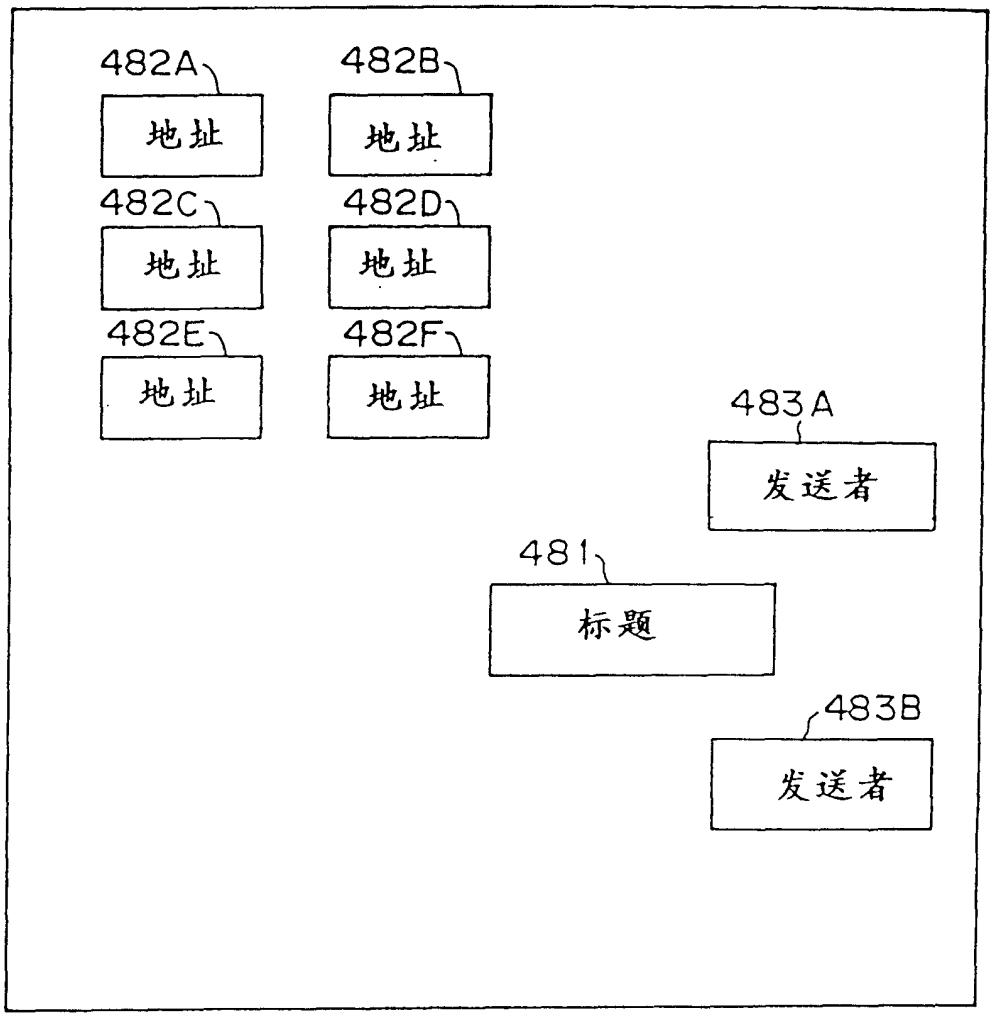

482A
地址
482B
地址
482C
地址
482D
地址
482E
地址
482F
地址
483A
发送者
481
标题
483B
发送者

图 . 48

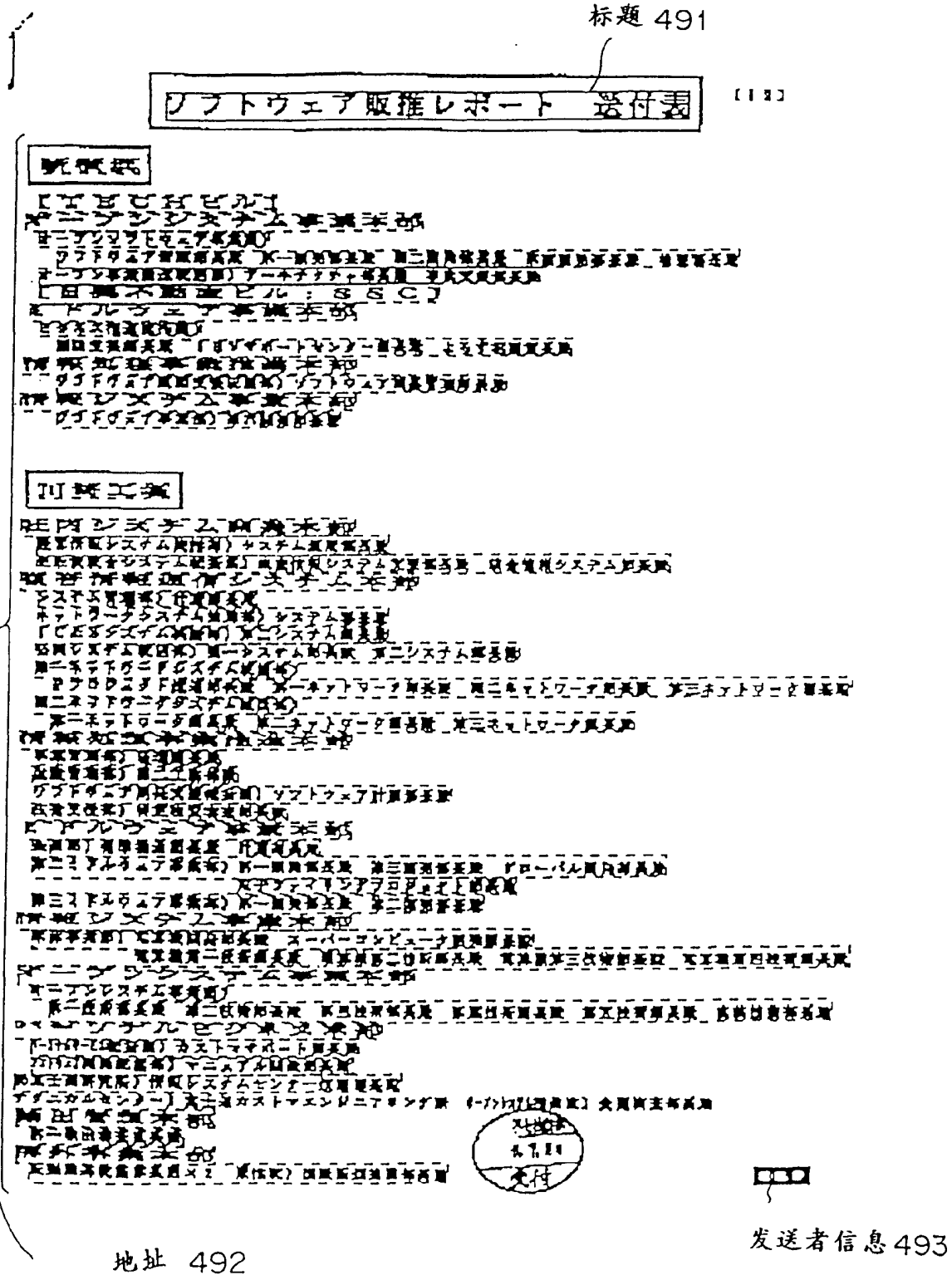


图. 49

technical_news

地址 502

[redacted]

[redacted] 发送者信息

[redacted] 503

标题 501

[redacted]

[redacted]

[redacted]

记

[redacted]

以 上

[redacted]

图. 50

表定界线 511

出張報告書

回覧

1／18

表題	ﾏﾙﾁﾒﾃﾞｨｱ とパターン認識シンポジウム	発行No.: 0123456789－1995 発行元: ○○○事業部△△△部××××課 発行日: 平成 7年12月31日

報告書				公開範囲	1: 事業部内 2: 部門内 3: 部内 4: 課内
従業員番号	氏 名	従業員番号	氏 名		
1) 111111	藤山 裕	2)		報告書の性格	
3)		4)		1: 普通 4: 海外 2: 急ぎ 5: 重要 3: 国内 6: その他	①: 速報 2: 詳細
5)		6)			
7)		8)			
9)		10)			

要約（３５０文字）：パターン認識研究の中で、文字認識は、郵便番号読み取り装置のように最も明確なニーズをもった研究対象の一つである。画像をテキストコードに変換する文字認識技術はマルチメディア時代のさまざまな新サービスに利用されようとしている。ここではこれから文字認識を研究されようとする方に、現状のサービス形態、技術レベルについて概観する。文字認識方式にはオフライン型とオンライン型の２種類がある。紙に書かれた文字をスキャナで読み取って認識する方式はオフライン文字認識と呼ばれ、郵便番号読み取り装置、伝票ＯＣＲ、文書ＯＣＲ等がある。一方、最近筆圧を読んだペン入力コンピュータによる文字認識方式はオンライン文字認識と呼ばれ、ペンの軌跡過程を時々刻々コンピュータに取り込み、その情報をもとに手書き文字を認識する。

社内分類	独自分類	通番	関連出張報告書	関連番号	関連製品名
(1) 1 2 3 4	(1) 2 3 4	(1) 0 0 0 1	(1)	(1) A A A 0 1	(1) 文書リーダ
(2) 5 6 7 8	(2) 4 5 6	(2)	(2)	(2) B B B 0 2	(2) 帳票ＯＣＲ
(3)	(3)	(3)	(3)	(3) C C C 0 3	(3)
		(4)	(4)	(4) D D D 0 4	(4)
		(5)	(5)	(5) E E E 0 5	(5)

キーワード		所属長印
(1) ＯＣＲ	(2) 文字認識	
(3) 画像処理	(4) ラベリング	
(5) 領域抽出	(6) 電子ファイリングシステム	
(7) 文書標成要素	(8) タイトル	
(9) 表	00 項目	
01	02	受付印
03	04	
05	06	

関係配布先名	コード	関係配布先名	コード	
(1) 鈴木課長殿	1111	(2) 佐藤課長殿	2222	
(3) 山本部長殿	3333	(4) 加藤課長殿	4444	
(5)		(6)		
(7)		(8)		整理番号
(9)		00		
01		02		
03		04		
05		06		
07		08		合計配布部数 4部
09		20		

複写の発行元確認	①: 要 2: 不要	連絡先・電話番号	氏名 藤山 裕	☎777-1111

图. 51

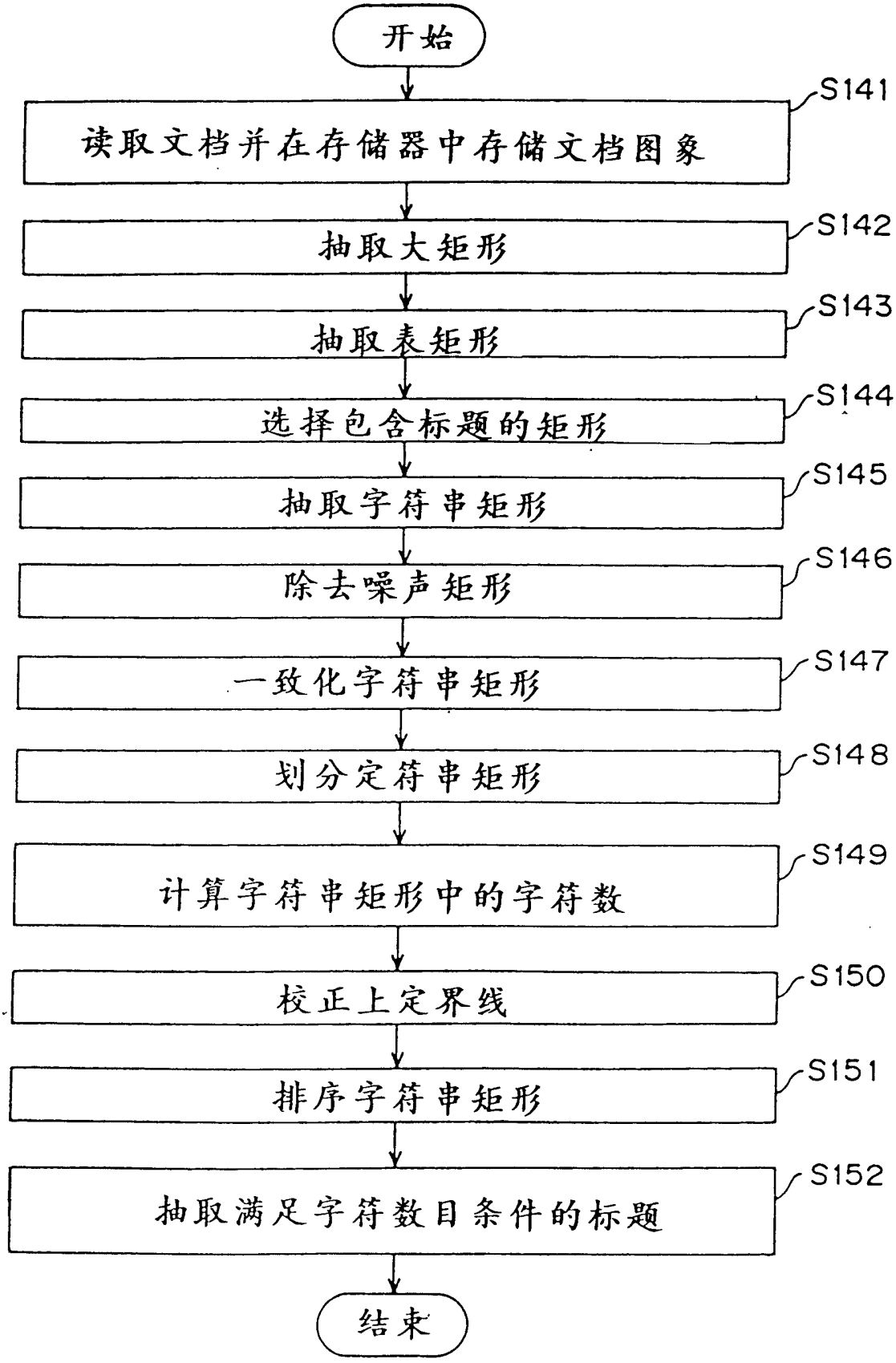

开始
S141
读取文档并在存储器中存储文档图象
S142
抽取大矩形
S143
抽取表矩形
S144
选择包含标题的矩形
S145
抽取字符串矩形
S146
除去噪声矩形
S147
一致化字符串矩形
S148
划分定符串矩形
S149
计算字符串矩形中的字符数
S150
校正上定界线
S151
排序字符串矩形
S152
抽取满足字符数目条件的标题
结束

图 . 52

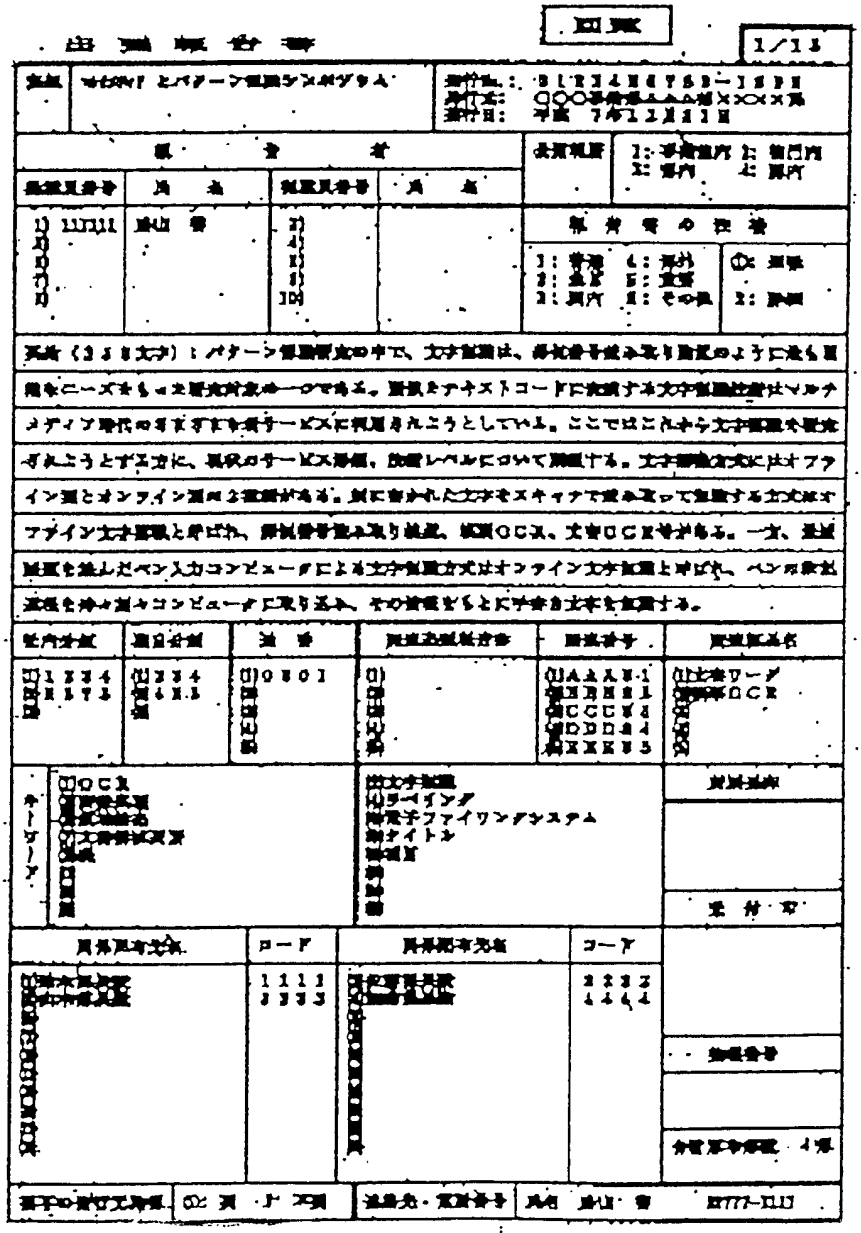


图. 53

表矩形 80

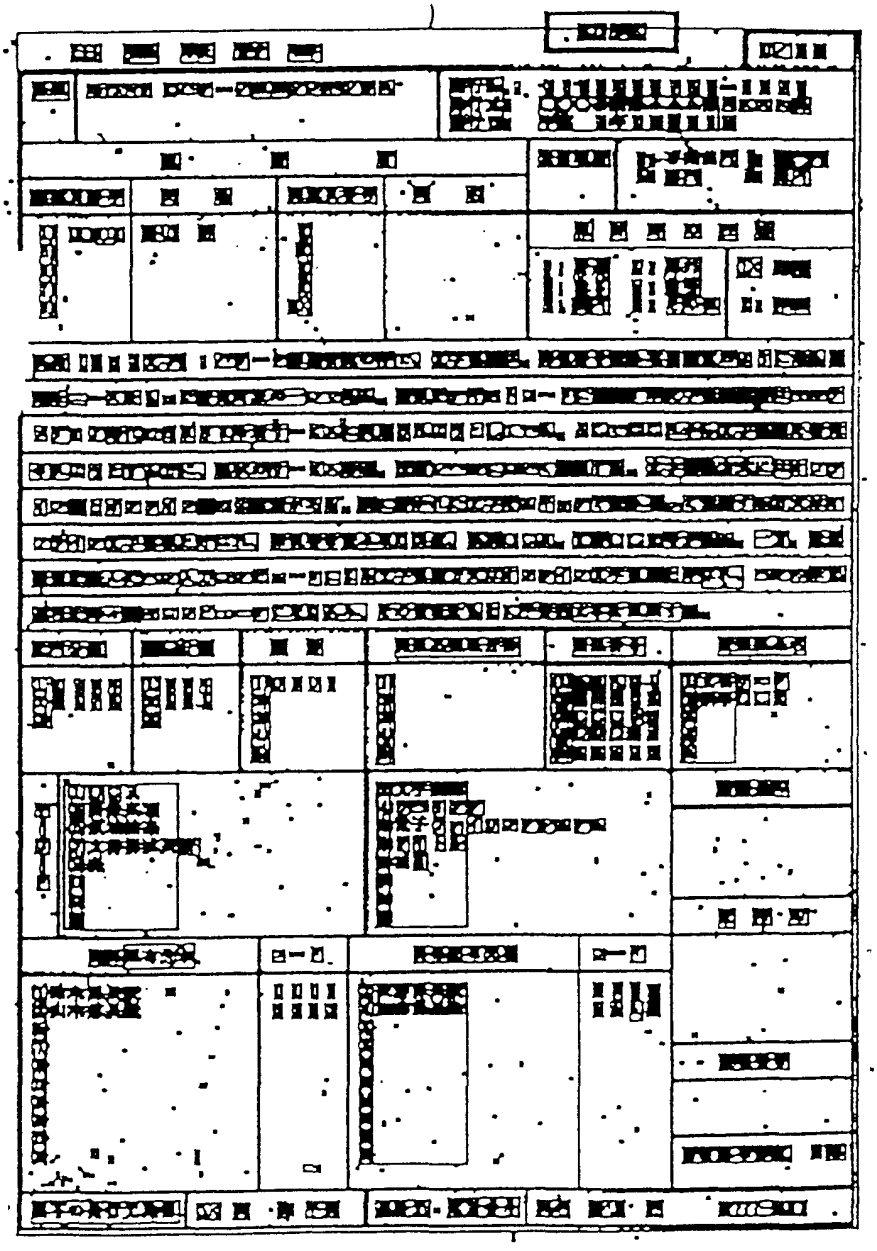


图 54

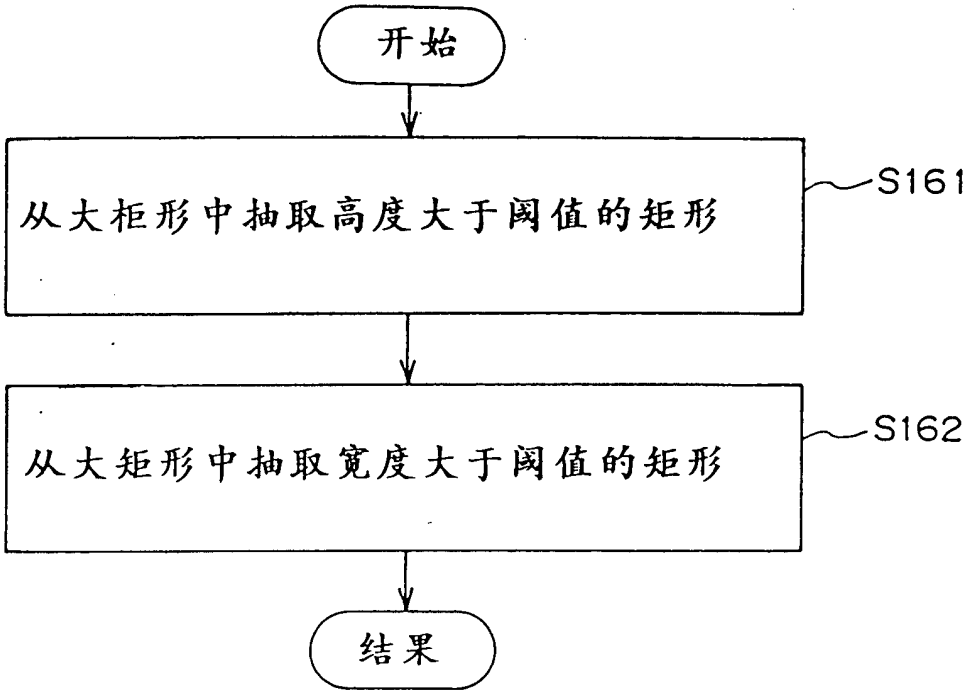


图 55

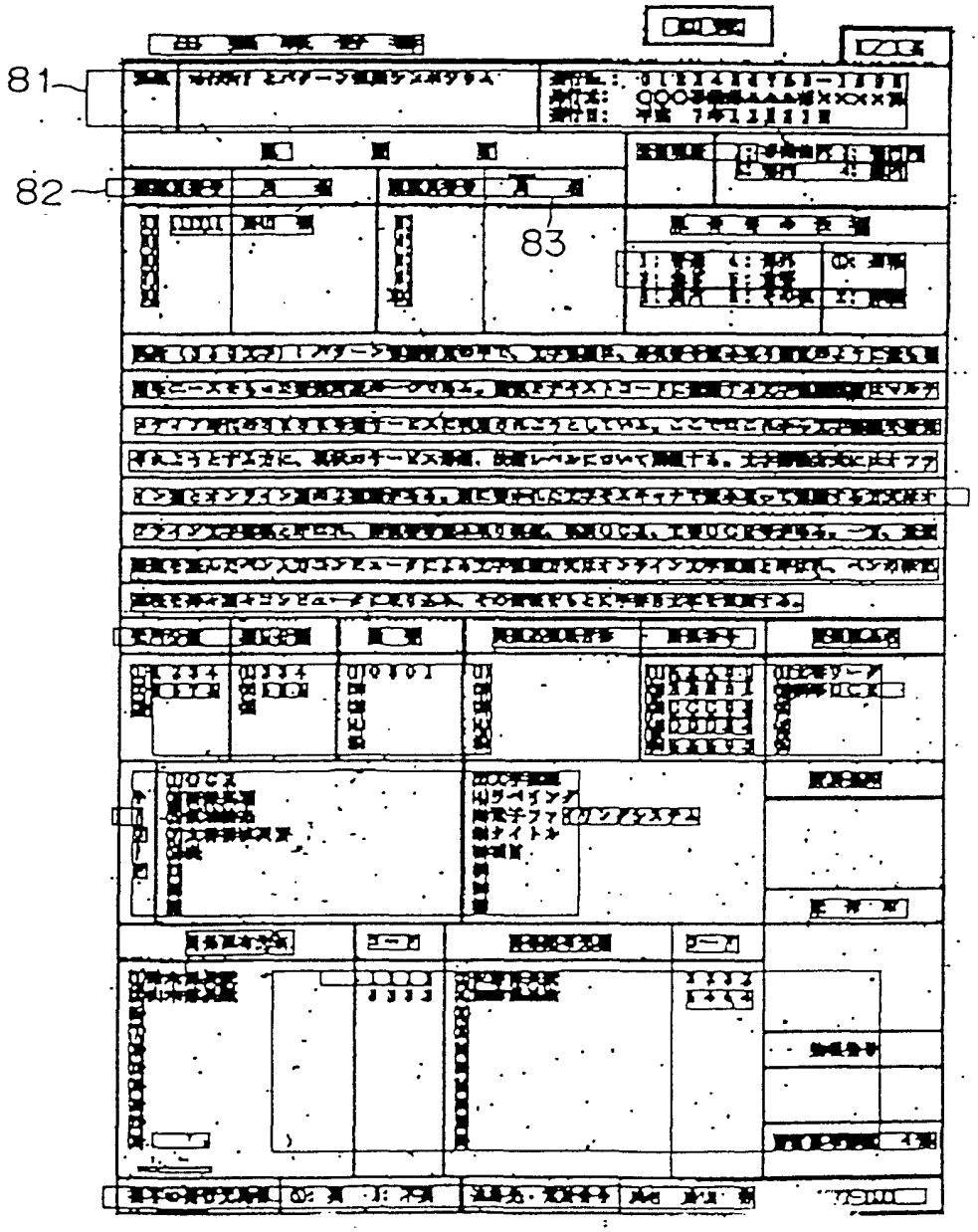


图56

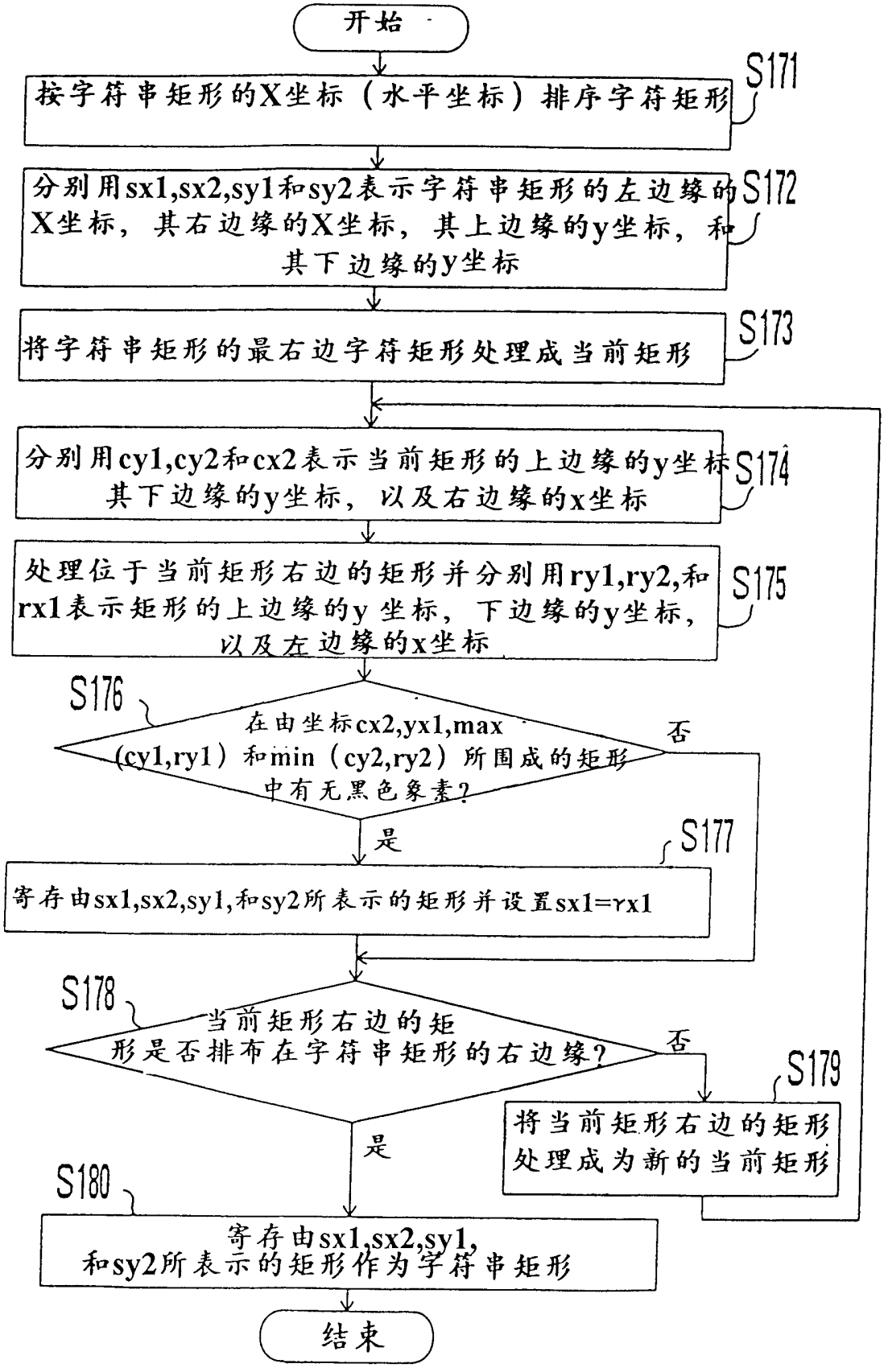

开始
按字符串矩形的X坐标（水平坐标）排序字符矩形
S171
分别用sx1,sx2,sy1和sy2表示字符串矩形的左边缘的X坐标，其右边缘的X坐标，其上边缘的y坐标，和其下边缘的y坐标
S172
将字符串矩形的最右边字符矩形处理成当前矩形
S173
分别用cy1,cy2和cx2表示当前矩形的上边缘的y坐标其下边缘的y坐标，以及右边缘的x坐标
S174
处理位于当前矩形右边的矩形并分别用ry1,ry2,和rx1表示矩形的上边缘的y 坐标，下边缘的y坐标，以及左边缘的x坐标
S175
S176
在由坐标cx2,yx1,max(cy1,ry1）和min（cy2,ry2）所围成的矩形中有无黑色象素?
否
是
S177
寄存由sx1,sx2,sy1,和sy2所表示的矩形并设置sx1=rx1
S178
当前矩形右边的矩形是否排布在字符串矩形的右边缘?
否
S179
将当前矩形右边的矩形处理成为新的当前矩形
是
S180
寄存由sx1,sx2,sy1,和sy2所表示的矩形作为字符串矩形
结束

图. 57

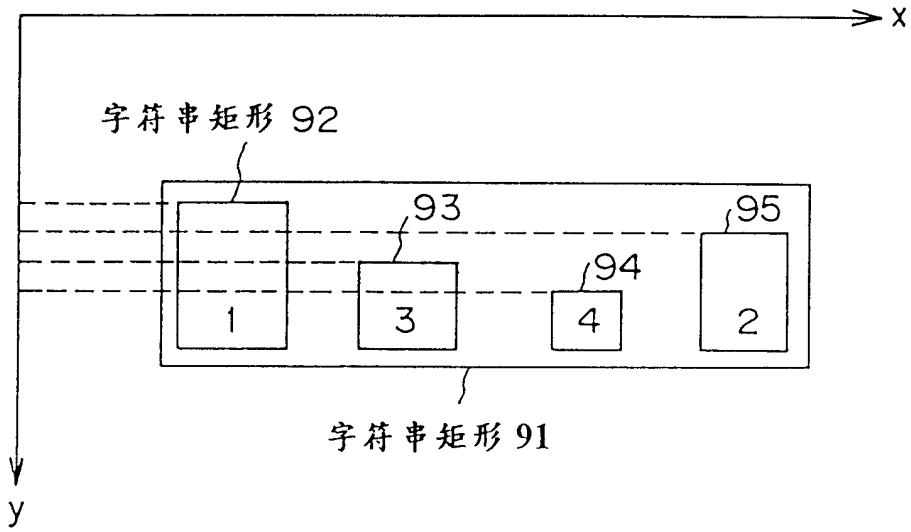


图. 58

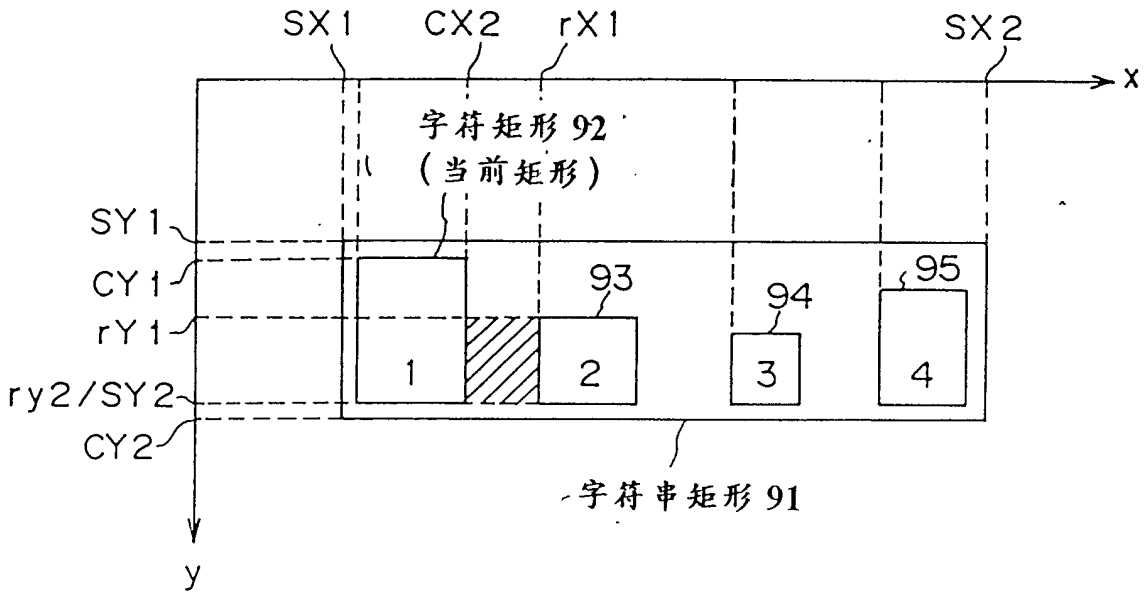

SX1
CX2
rX1
SX2
x
字符矩形 92
(当前矩形)
SY1
CY1
rY1
ry2/SY2
CY2
93
94
95
1
2
3
4
字符串矩形 91
y

图. 59

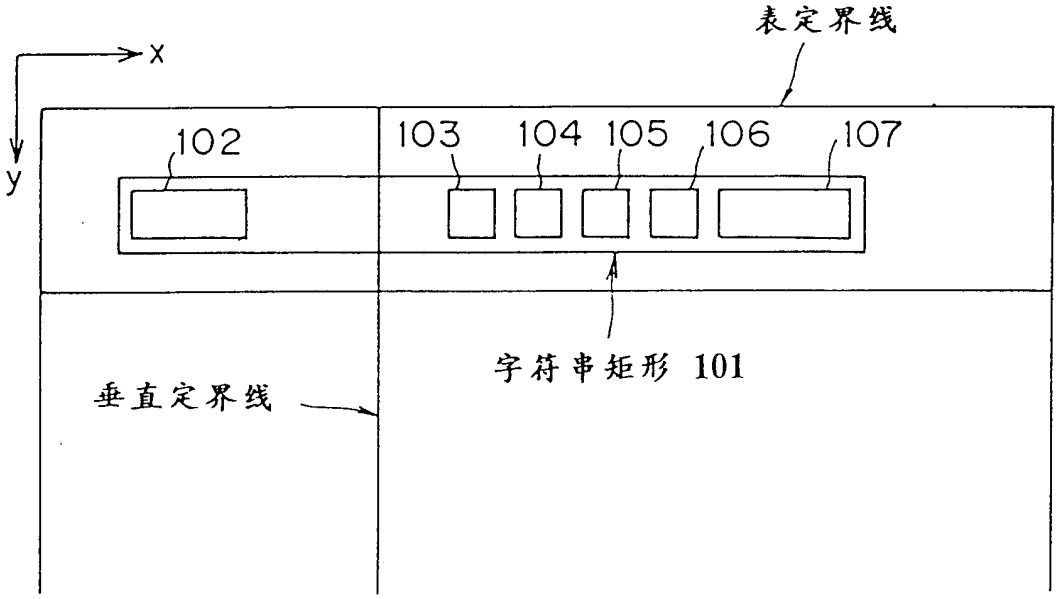


图. 60

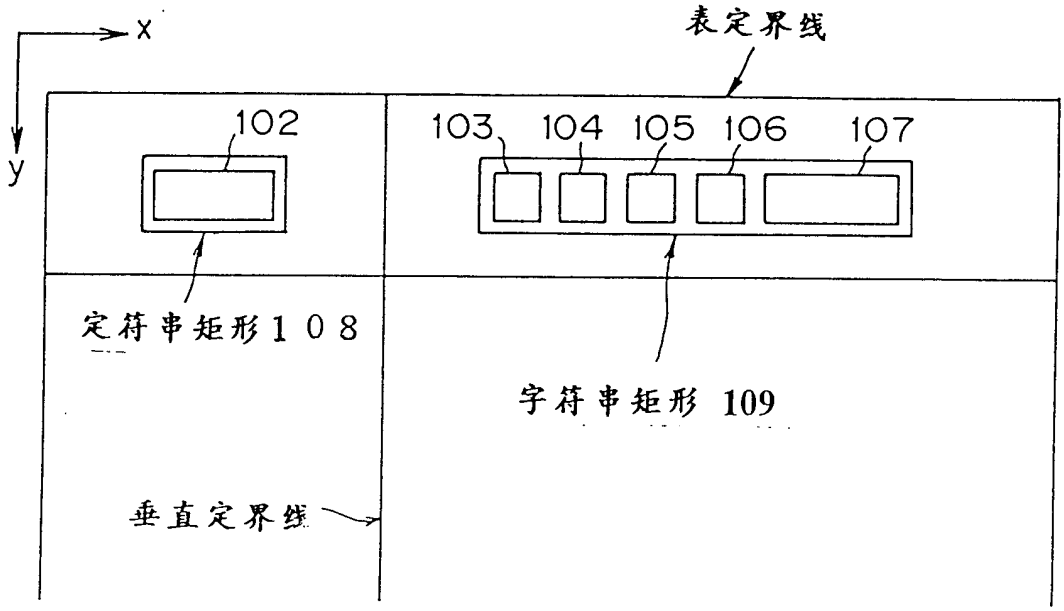


图. 61

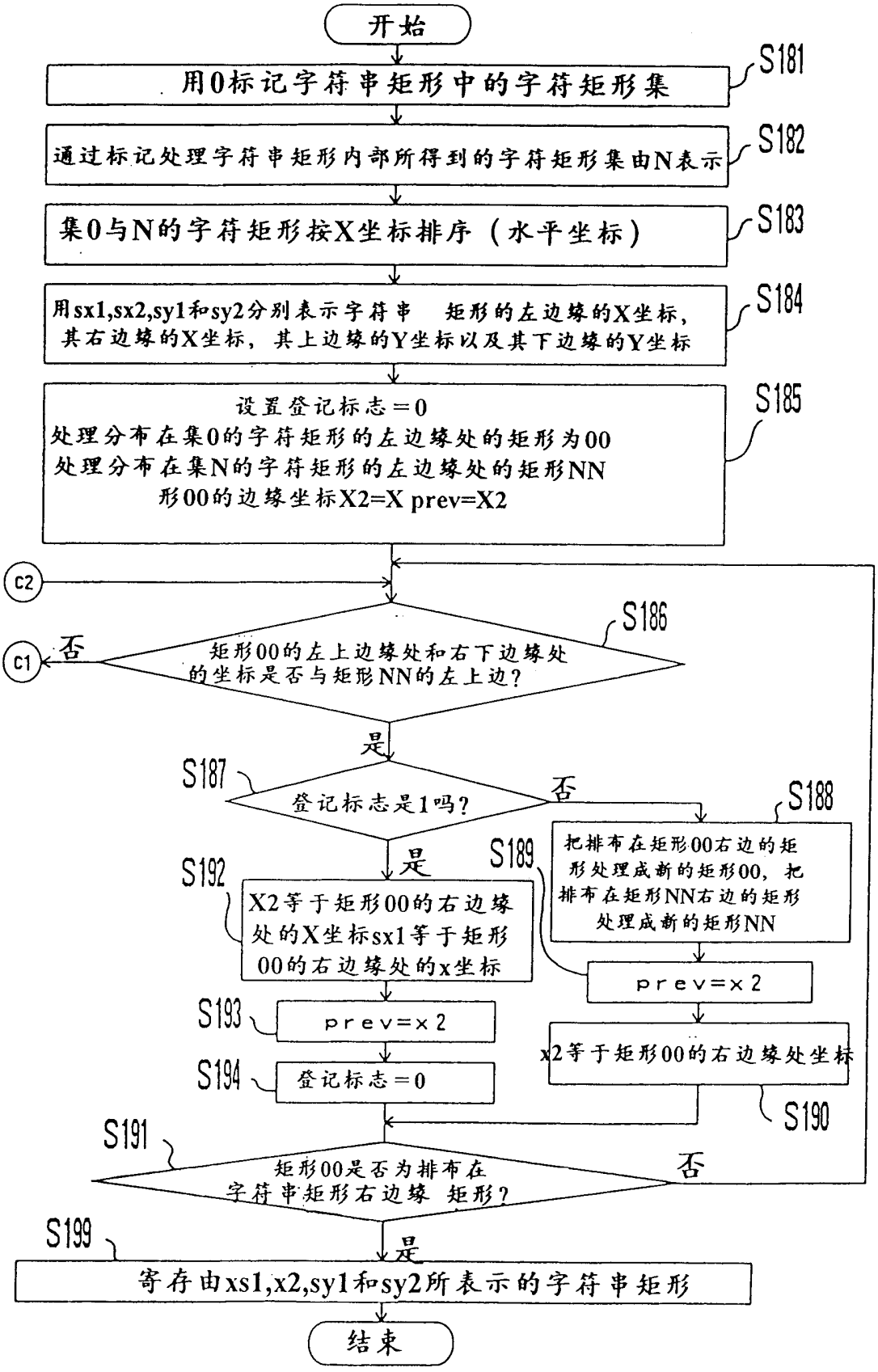

开始
用0标记字符串矩形中的字符矩形集
S181
通过标记处理字符串矩形内部所得到的字符矩形集由N表示
S182
集0与N的字符矩形按X坐标排序（水平坐标）
S183
用sx1,sx2,sy1和sy2分别表示字符串 矩形的左边缘的X坐标，其右边缘的X坐标，其上边缘的Y坐标以及其下边缘的Y坐标
S184
设置登记标志＝0
处理分布在集0的字符矩形的左边缘处的矩形为00
处理分布在集N的字符矩形的左边缘处的矩形NN
形00的边缘坐标X2=X prev=X2
S185
C2
C1
否
矩形00的左上边缘处和右下边缘处的坐标是否与矩形NN的左上边？
S186
是
S187
登记标志是1吗？
否
S188
把排布在矩形00右边的矩形处理成新的矩形00，把排布在矩形NN右边的矩形处理成新的矩形NN
S189
是
S192
X2等于矩形00的右边缘处的X坐标sx1等于矩形00的右边缘处的x坐标
prev=x2
S193
登记标志＝0
S194
x2等于矩形00的右边缘处坐标
S190
S191
矩形00是否为排布在字符串矩形右边缘 矩形？
否
是
S199
寄存由xs1,x2,sy1和sy2所表示的字符串矩形
结束

图 . 62

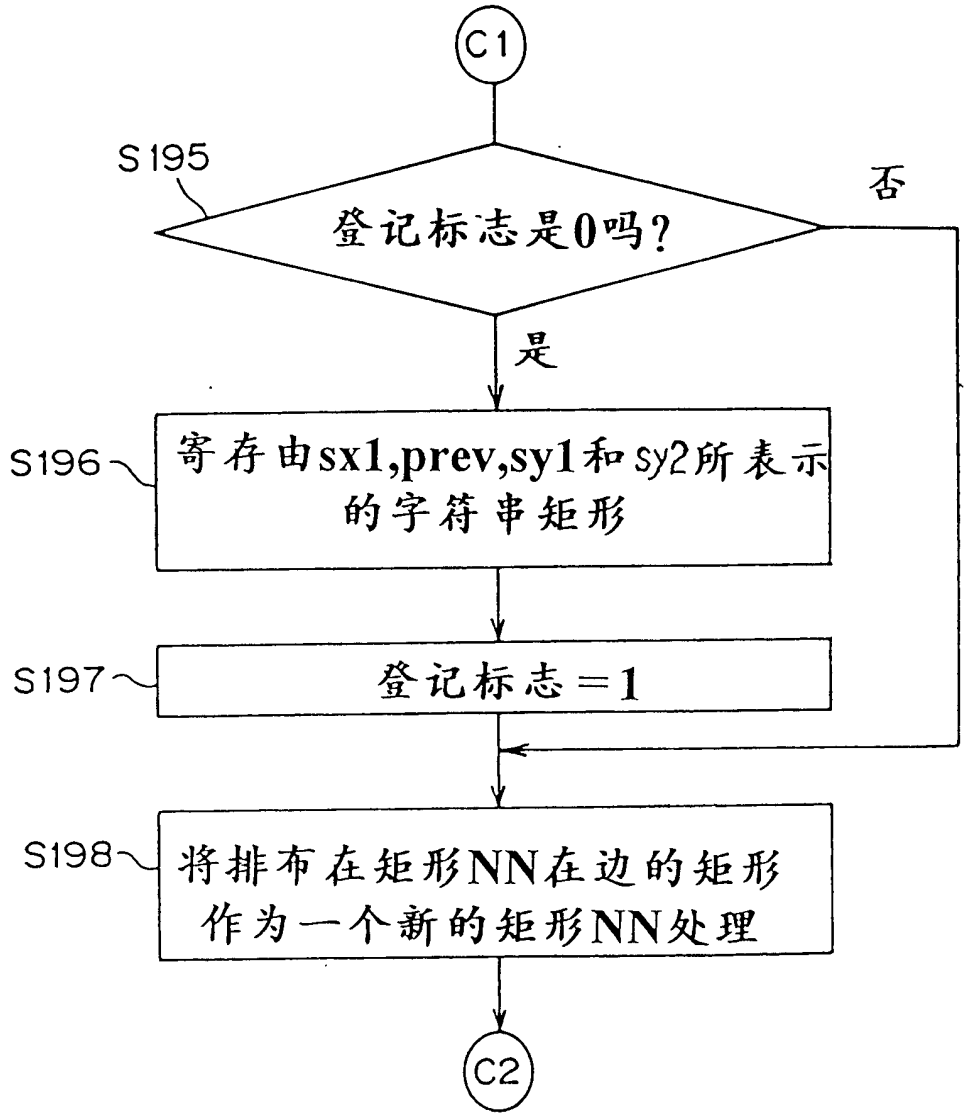

C1
S195
登记标志是0吗?
否
是
S196
寄存由sx1,prev,sy1和sy2所表示的字符串矩形
S197
登记标志=1
S198
将排布在矩形NN在边的矩形作为一个新的矩形NN处理
C2

图 63

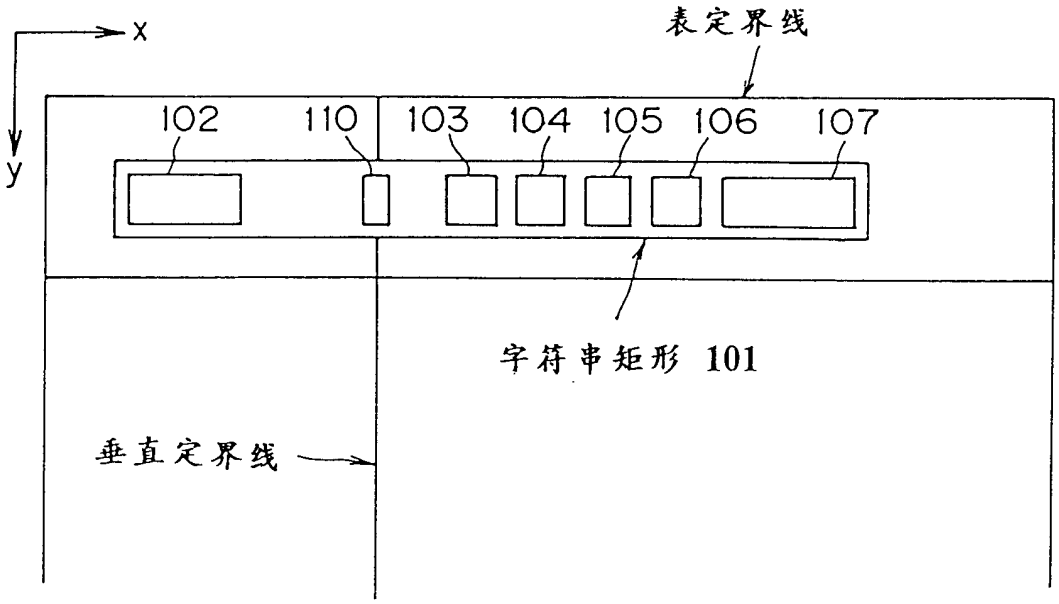


图. 64

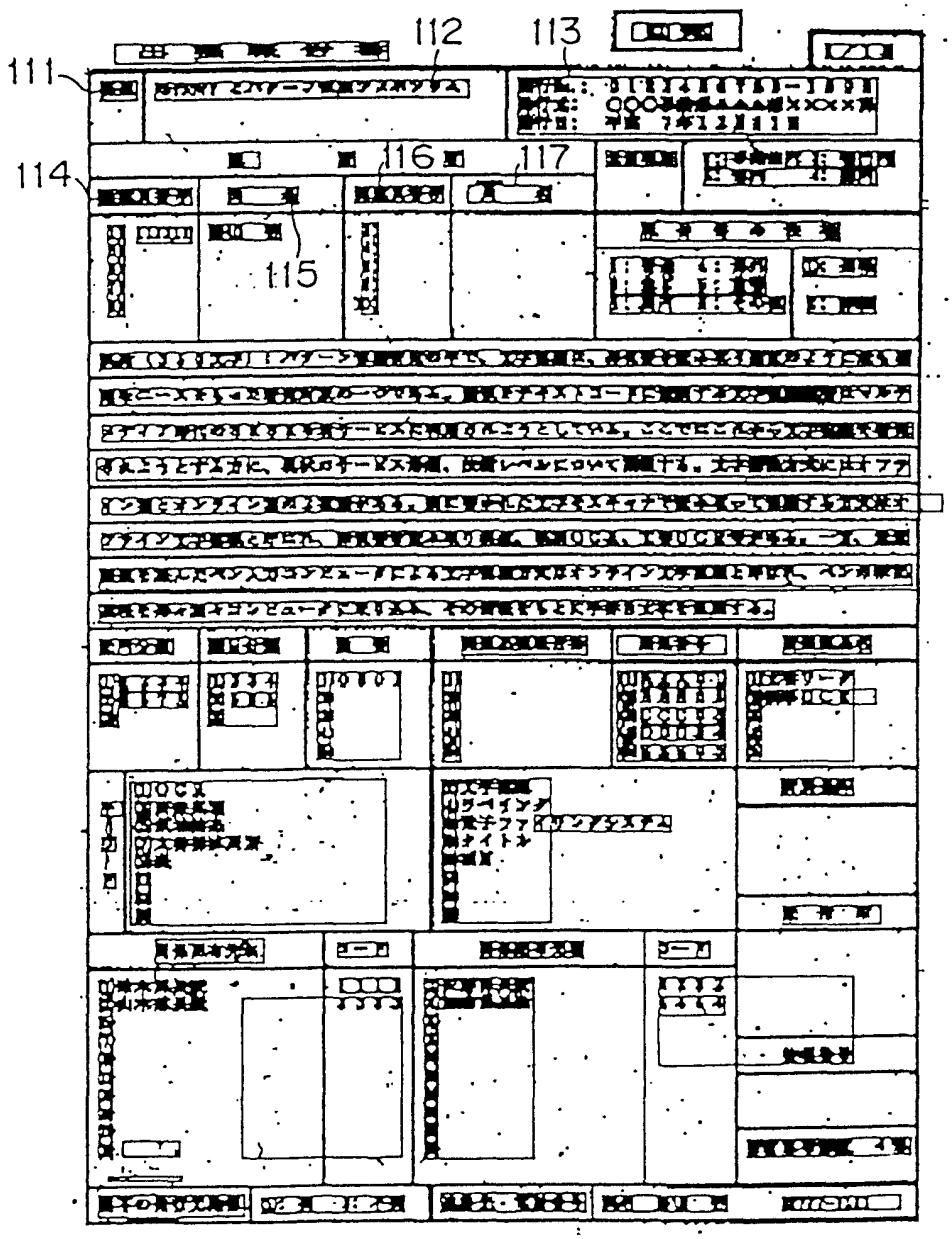


图. 65

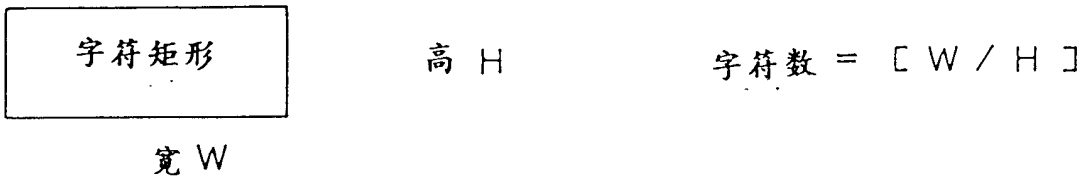


图. 66

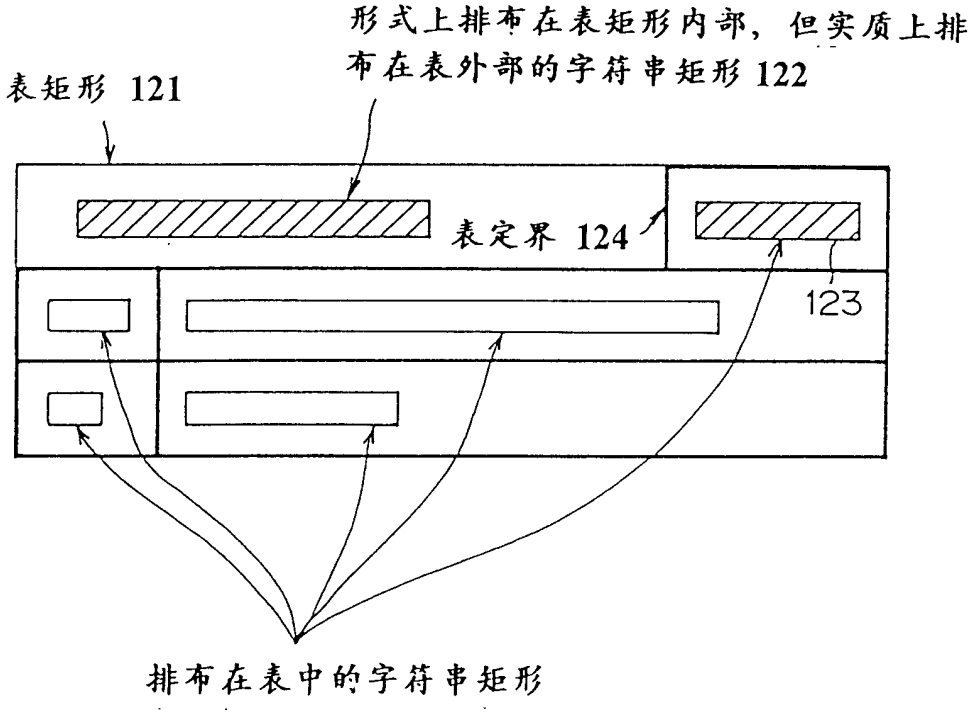

形式上排布在表矩形内部，但实质上排布在表外部的字符串矩形 122
表矩形 121
表定界 124
123
排布在表中的字符串矩形

图. 67

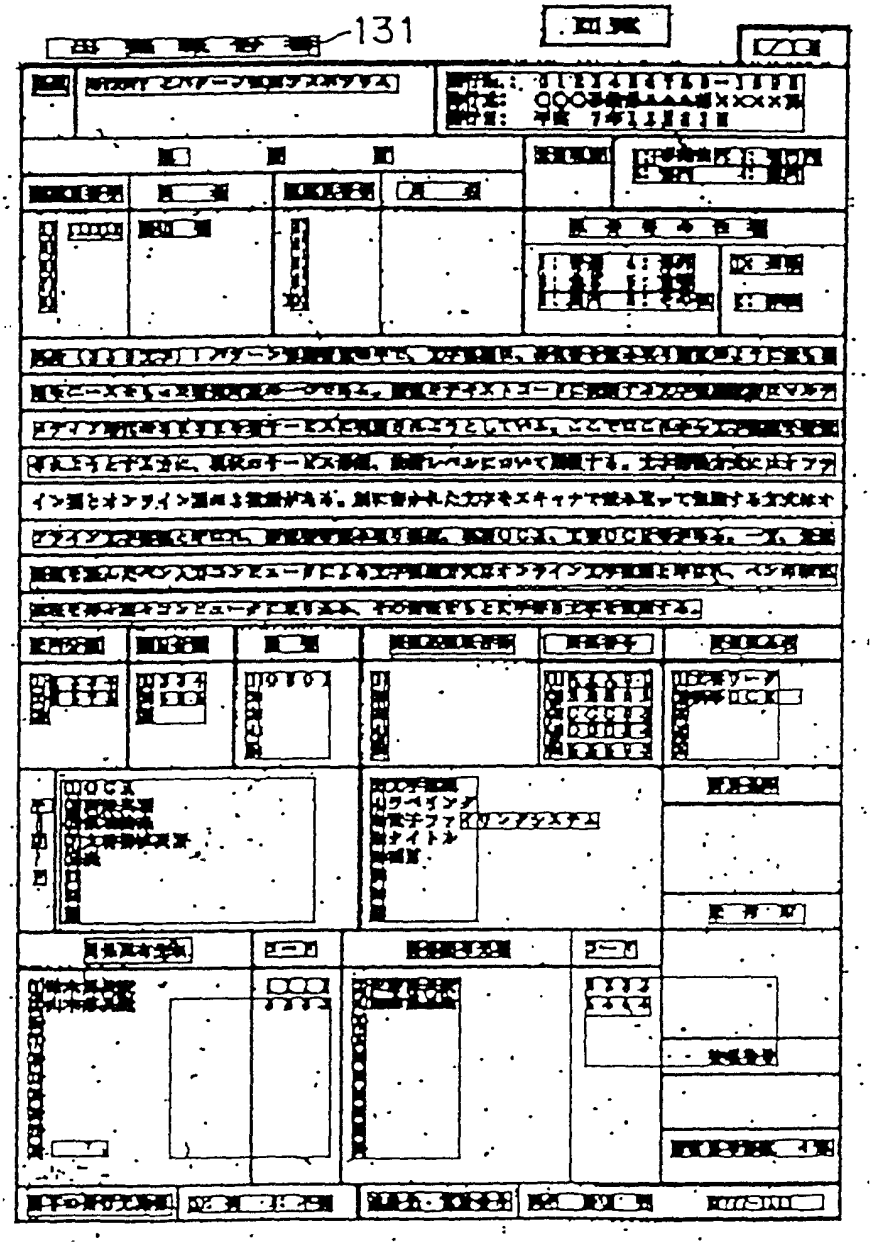


图 68

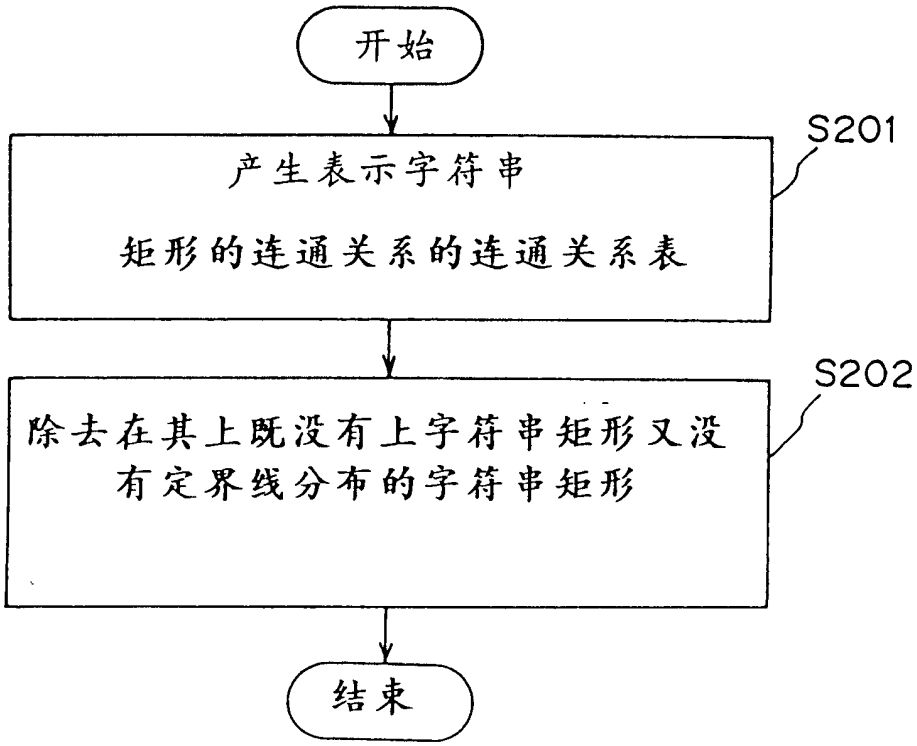

开始
S201
产生表示字符串
矩形的连通关系的连通关系表
S202
除去在其上既没有上字符串矩形又没有定界线分布的字符串矩形
结束

图 . 69

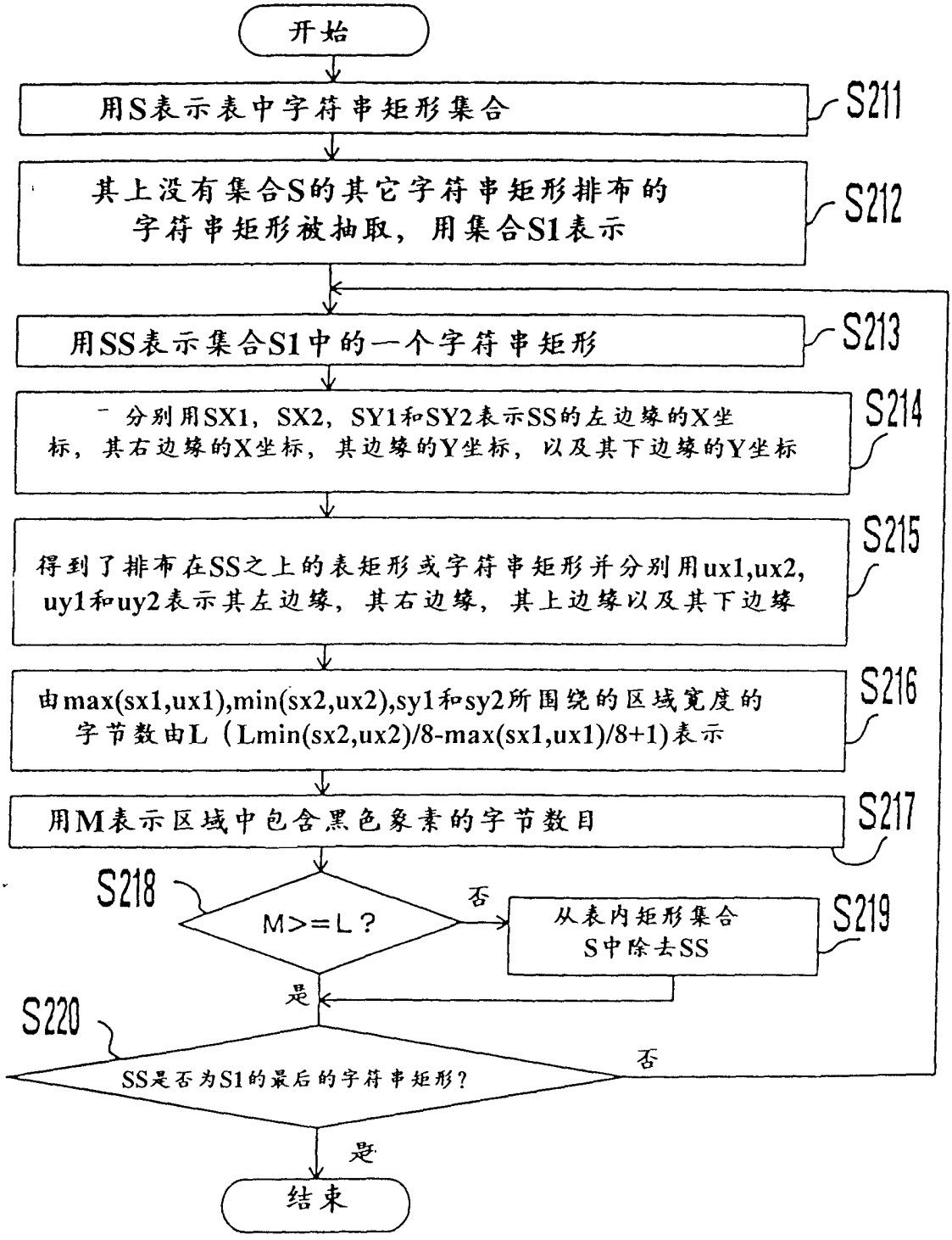

开始
用S表示表中字符串矩形集合
S211
其上没有集合S的其它字符串矩形排布的字符串矩形被抽取，用集合S1表示
S212
用SS表示集合S1中的一个字符串矩形
S213
分别用SX1，SX2，SY1和SY2表示SS的左边缘的X坐标，其右边缘的X坐标，其边缘的Y坐标，以及其下边缘的Y坐标
S214
得到了排布在SS之上的表矩形或字符串矩形并分别用ux1,ux2,uy1和uy2表示其左边缘，其右边缘，其上边缘以及其下边缘
S215
由max(sx1,ux1),min(sx2,ux2),sy1和sy2所围绕的区域宽度的字节数由L（Lmin(sx2,ux2)/8-max(sx1,ux1)/8+1)表示
S216
用M表示区域中包含黑色象素的字节数目
S217
S218
M>=L？
否
从表内矩形集合S中除去SS
S219
是
S220
SS是否为S1的最后的字符串矩形？
否
是
结束

图.70

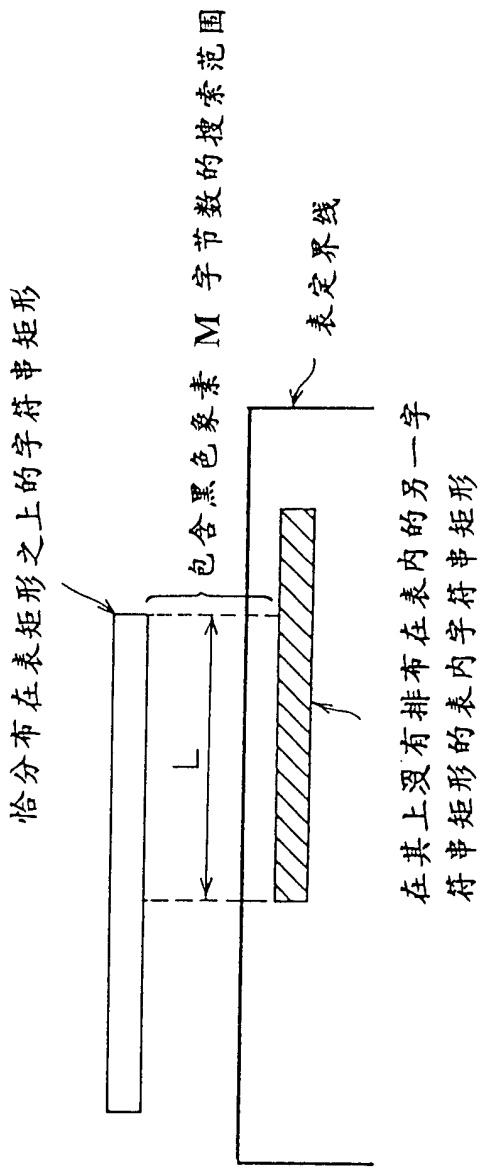

恰分布在表矩形之上的字符串矩形
包含黑色象素 M 字节数的搜索范围
表定界线
L
在其上没有排布在表内的另一字
符串矩形的表内字符串矩形

图. 71

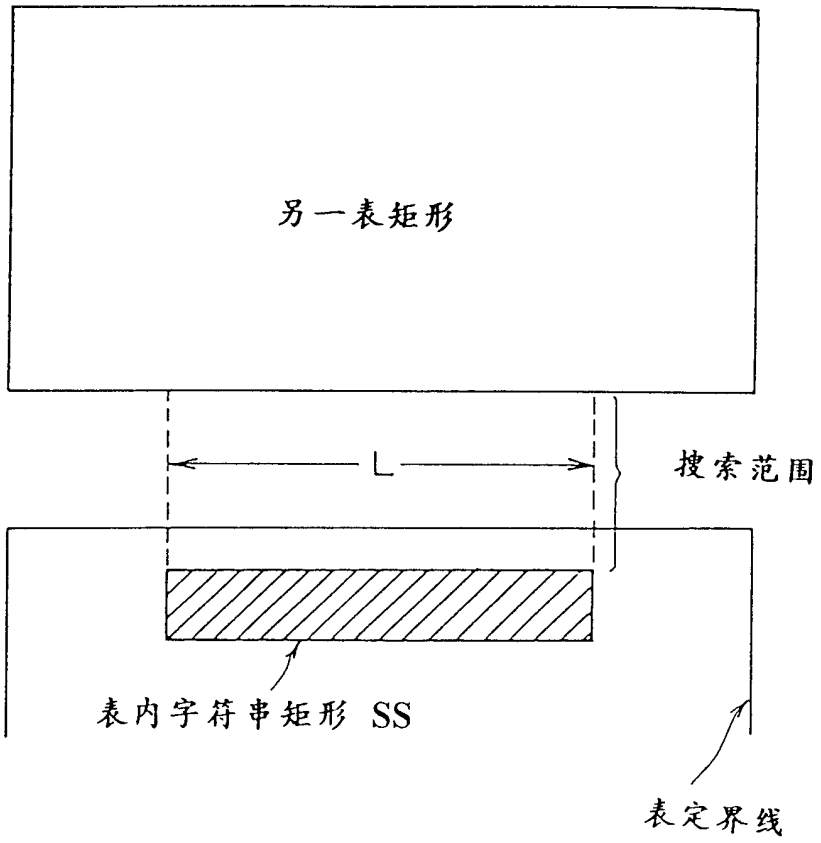


图. 72

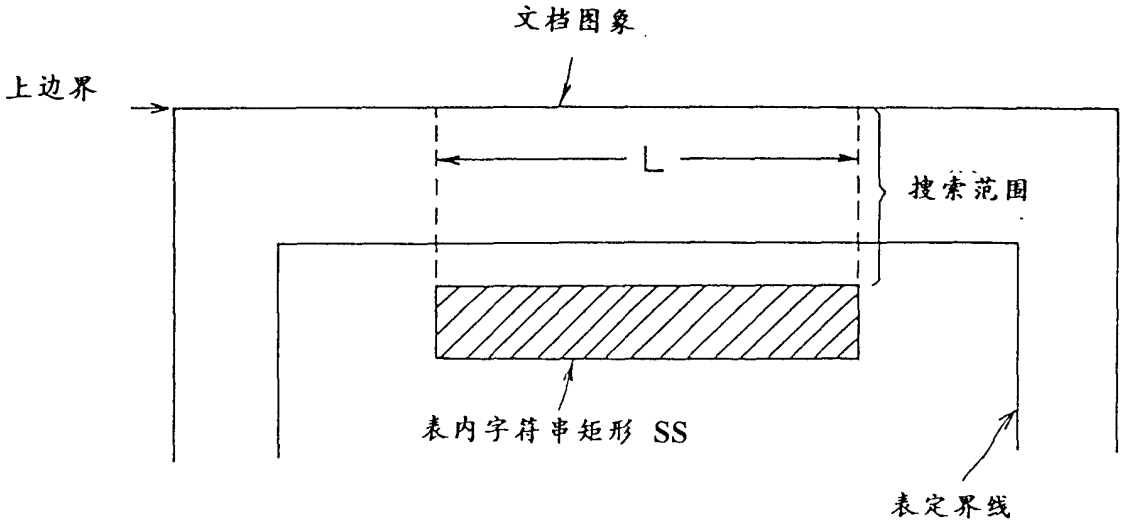


图. 73

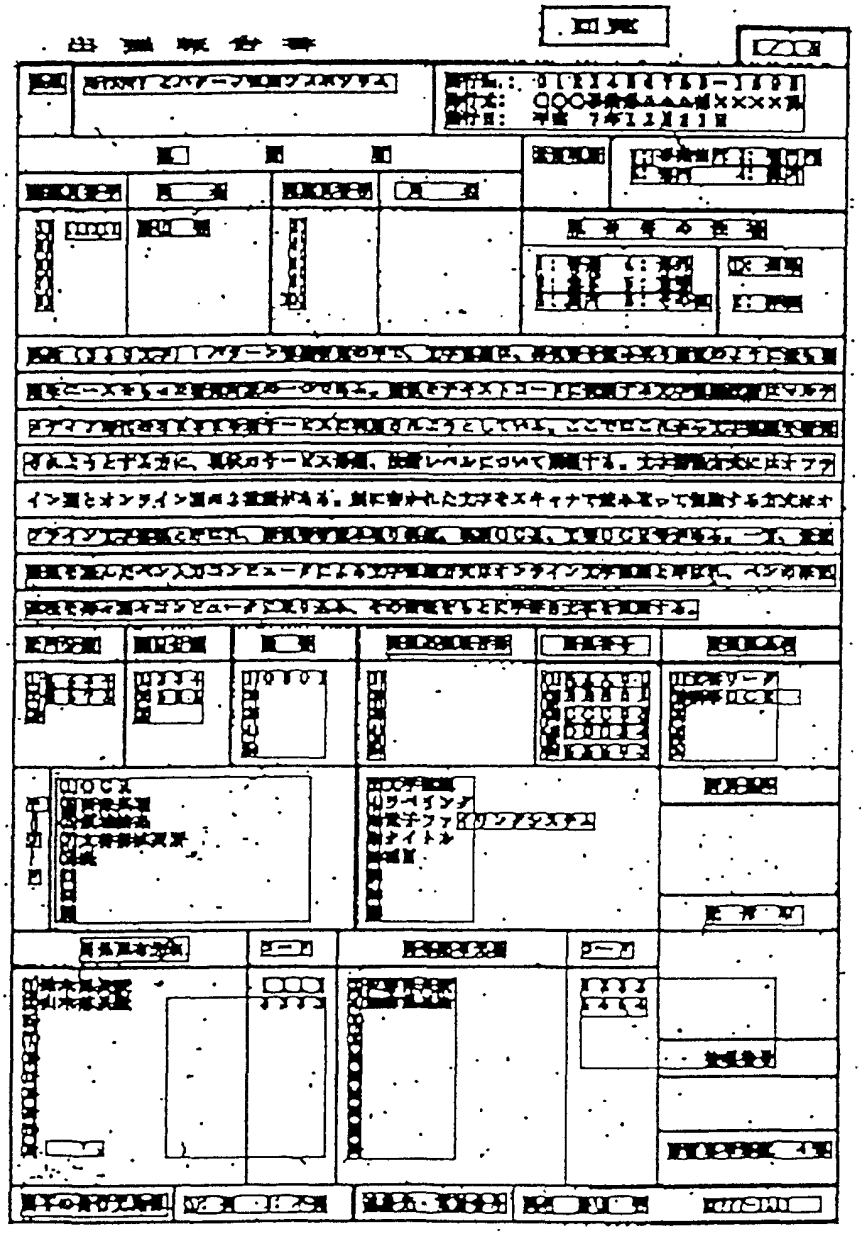


图. 74

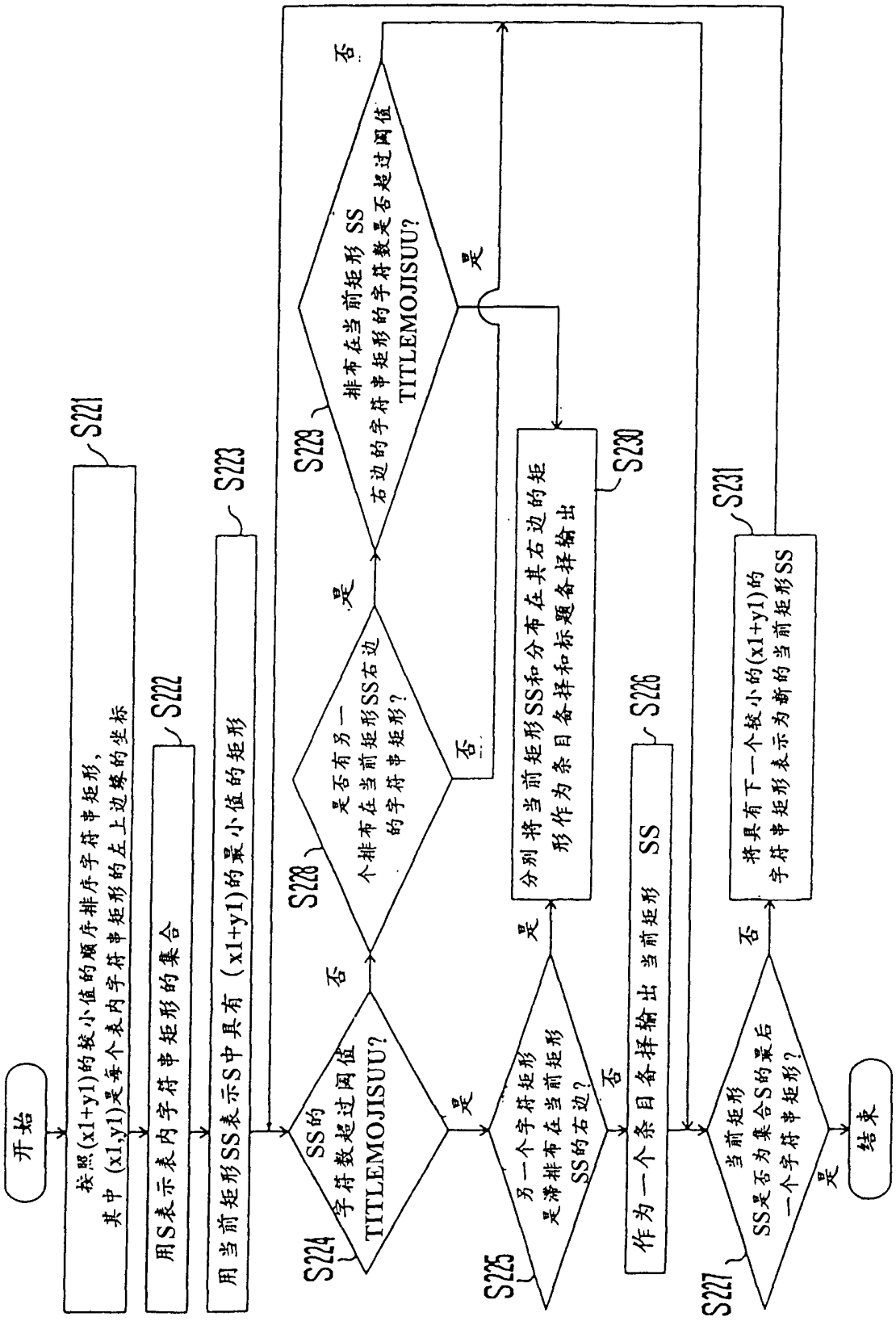

开始
按照(x1+y1)的较小值的顺序排序字符串矩形，
其中(x1,y1)是每个表内字符串矩形的左上边缘的坐标
S221
用S表示表内字符串矩形的集合
S222
用当前矩形SS表示S中具有(x1+y1)的最小值的矩形
S223
SS的
字符数超过阈值
TITLEMOJISUU?
S224
否
是
另一个字符矩形
是滞排布在当前矩形
SS的右边?
S225
是
否
是否有另一
个排布在当前矩形SS右边
的字符串矩形?
S228
是
否
排布在当前矩形 SS
右边的字符串矩形的字符数是否超过阈值
TITLEMOJISUU?
S229
是
否
分别将当前矩形SS和分布在其右边的矩
形作为条目备择和标题备择输出
S230
作为一个条目备择输出 当前矩形 SS
S226
当前矩形
SS是否为集合S的最后
一个字符串矩形?
S227
是
否
将具有下一个较小的(x1+y1)的
字符串矩形表示为新的当前矩形SS
S231
结束

图.75

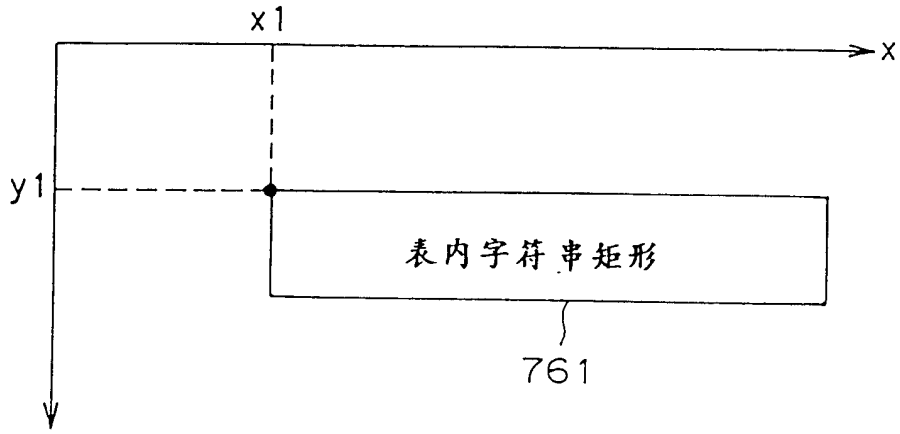


图. 76

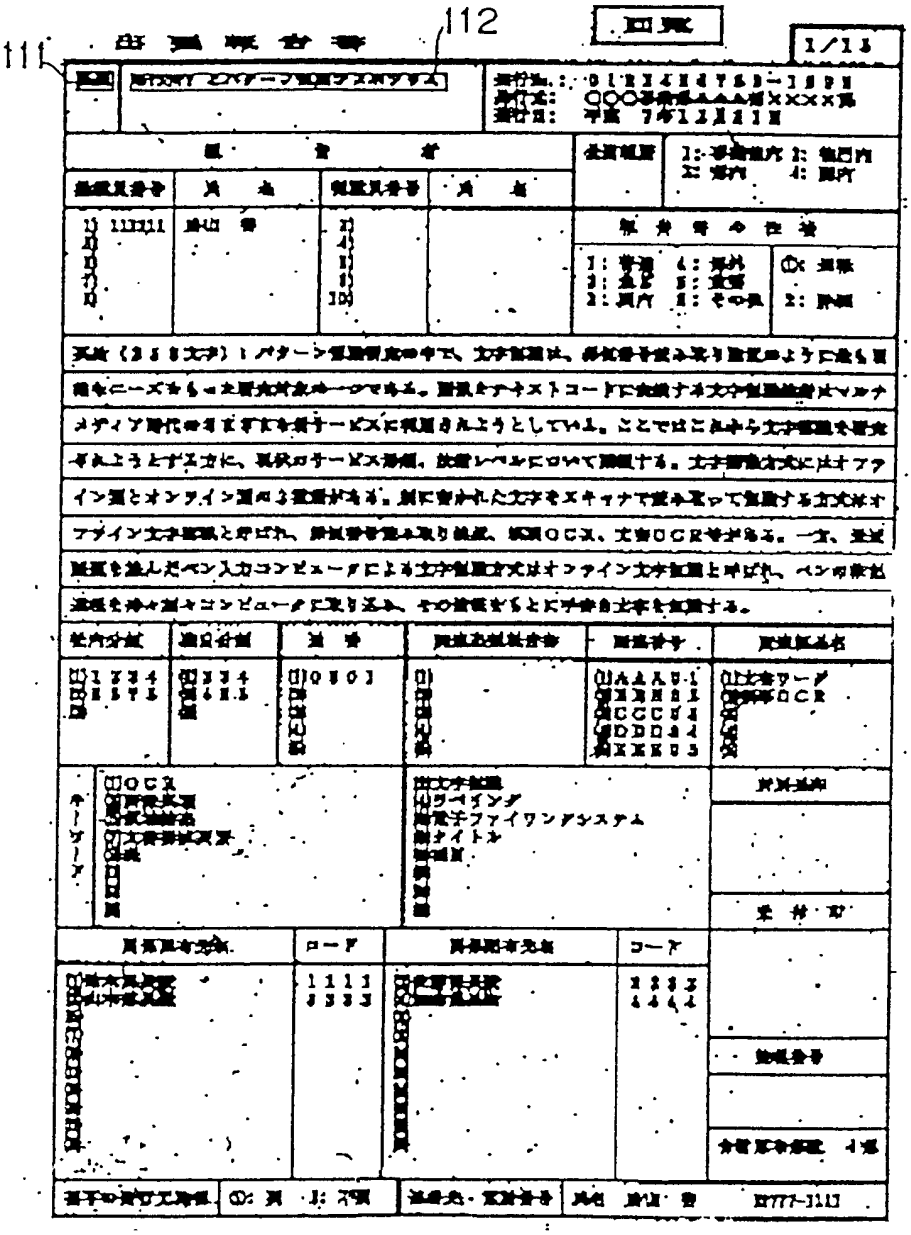


图. 77