



(12) 发明专利

(10) 授权公告号 CN 101689203 B

(45) 授权公告日 2013.09.18

(21) 申请号 200880022834.5

(22) 申请日 2008.06.23

(30) 优先权数据

172737/2007 2007.06.29 JP

(85) PCT申请进入国家阶段日

2009.12.29

(86) PCT申请的申请数据

PCT/JP2008/061781 2008.06.23

(87) PCT申请的公布数据

W02009/005019 EN 2009.01.08

(73) 专利权人 佳能株式会社

地址 日本东京都

(72) 发明人 金津知俊 榎本诚 山崎妙子

(74) 专利代理机构 北京魏启学律师事务所

11398

代理人 魏启学

(51) Int. Cl.

G06F 17/30 (2006.01)

(56) 对比文件

JP 特开 2005-275863 A, 2005.10.06, 说明书摘要, 说明书第【0013】段—第【0036】段, 说明书附图 1—4.

JP 特开平 11-232276 A, 1999.08.27, 全文.

JP 特开 2005-259017 A, 2005.09.22, 全文.

CN 1752992 A, 2006.03.29, 全文.

审查员 徐淑娴

权利要求书2页 说明书16页 附图16页

(54) 发明名称

图像处理设备和图像处理方法

(57) 摘要

一种图像处理设备,包括:字符识别单元,用于对文档图像中的多个字符图像进行字符识别,以获取与各个字符图像相对应的字符代码;以及生成单元,用于生成电子文档,其中,该电子文档包括文档图像、由字符识别单元所获取到的多个字符代码、多个字形和表示对各个字符代码进行绘制时所使用的字形的数据;其中,在绘制与由字符识别单元所获取到的多个字符代码相对应的字符时,不同的字符代码基于所述数据共享及使用多个字形中的各个字形。

```
601<?xml version="1.0"?>
602<Document>
603  <Page Width="1680" Height="2376" Dpi="200" Number="1">
604    <Image X="0" Y="0" Width="1680" Height="2376"
605      Data="ACA189w68q34rs6b23c8ax...a3E52zAD"/>
606    <Text X="238" Y="272" Font="Font01" Size="90"
607      Color="0,0,0,255" String="0x2359,0x2359,0x2341"
608      CGlyphId="1,6,3"/>
609    <Text X="238" Y="472" Font="Font01" Size="90"
610      Color="0,0,0,255" String="0x2359,0x2342,0x2363"
611      CGlyphId="0,2,5"/>
612  </Page>
613  <Page Width="1680" Height="2376" Dpi="200" Number="2">
614    <Image X="0" Y="0" Width="1680" Height="2376"
615      Data="s8DAN2q7s823Bast2z...sAq9d02dDJ"/>
616    <Text X="138" Y="372" Font="Font01" Size="90"
617      Color="0,0,0,255" String="0x242b,0x242d,0x242f"
618      CGlyphId="1,1,8"/>
619  </Page>
620  <Font Id="Font01">
621    <Glyph Id="0" Path="M0,0 V-1024 H1024 V1024 F"/>
622    <Glyph Id="1" Path="M0,0 V-1024 H998 V1024 F"/>
623    <Glyph Id="2" Path="M0,0 V-1024 H768 V1024 F"/>
624    <Glyph Id="3" Path="M0,0 V-1024 H640 V1024 F"/>
625    <Glyph Id="4" Path="M0,0 V-1024 H512 V1024 F"/>
626    <Glyph Id="5" Path="M0,0 V-1024 H384 V1024 F"/>
627    <Glyph Id="6" Path="M0,0 V-1024 H256 V1024 F"/>
628    <Glyph Id="7" Path="M0,0 V-1024 H128 V1024 F"/>
629  </Font>
630</Document>
```

1. 一种图像处理设备,包括:

字符识别单元,用于对文档图像中的多个字符图像进行字符识别,以获取与多个字符图像相对应的字符代码;

选择单元,用于基于所述文档图像中的各个字符图像的字符宽度和字符行高度或者基于所述文档图像中的各个字符图像的字符间距和字符行高度,从多个字形中选择各个字符图像要使用的字形;以及

生成单元,用于生成电子文档,其中,所述电子文档包括所述文档图像、由所述字符识别单元所获取到的多个字符代码、表示绘制与由所述字符识别单元所获取到的多个字符代码相对应的字符所要使用的形状的多个字形、以及指定绘制与由所述字符识别单元所获取到的多个字符代码相对应的字符所要使用的字形的描述数据;

其中,基于所述选择单元针对各个字符图像所选择的字形来描述所述描述数据,以及

在绘制与由所述字符识别单元所获取到的多个字符代码相对应的字符时,不同的字符代码共享及使用所述多个字形中的各个字形。

2. 根据权利要求1所述的图像处理设备,其特征在于,所述多个字形分别具有不同的宽度。

3. 根据权利要求1所述的图像处理设备,其特征在于,所述多个字形分别表示简单形状。

4. 根据权利要求1所述的图像处理设备,其特征在于,所述多个字形分别表示矩形形状。

5. 根据权利要求1所述的图像处理设备,其特征在于,所述多个字形分别表示空白字符的形状。

6. 根据权利要求1所述的图像处理设备,其特征在于,所述多个字形绘制在字符行的下部。

7. 根据权利要求1所述的图像处理设备,其特征在于,所述生成单元生成还包括字符之间距离的平均值的电子文档。

8. 根据权利要求1所述的图像处理设备,其特征在于,所述描述数据还包括如下描述:该描述用于在与所述文档图像中的各个字符图像相对应的位置处,以透明颜色绘制针对所述多个字符代码中的各个字符代码所指定的字形。

9. 根据权利要求1所述的图像处理设备,其特征在于,以可扩展标记语言格式生成所述电子文档。

10. 根据权利要求1所述的图像处理设备,其特征在于,以可扩展标记语言纸张规范格式生成所述电子文档。

11. 根据权利要求1所述的图像处理设备,其特征在于,还包括压缩单元,所述压缩单元用于对用于存储在所述电子文档中的所述文档图像进行压缩。

12. 根据权利要求11所述的图像处理设备,其特征在于,所述压缩单元分析所述文档图像中所包括的区域,并且自适应地进行压缩。

13. 根据权利要求1所述的图像处理设备,其特征在于,还包括搜索单元,所述搜索单元用于使用所输入的关键字对所生成的电子文档进行搜索,并突出显示与所输入的所述关键字相匹配的、绘制了与所述字符代码相对应的字形的部分。

14. 根据权利要求 13 所述的图像处理设备,其特征在于,所述搜索单元通过反转所述部分的颜色来突出显示所述部分。

15. 一种图像处理方法,包括:

通过对文档图像中的多个字符图像进行字符识别,获取与多个字符图像相对应的字符代码;

选择步骤,用于基于所述文档图像中的各个字符图像的字符宽度和字符行高度或者基于所述文档图像中的各个字符图像的字符间距和字符行高度,从多个字形中选择各个字符图像要使用的字形;以及

生成电子文档,其中,所述电子文档包括所述文档图像、通过进行所述字符识别所获取到的多个字符代码、表示绘制与通过进行所述字符识别所获取到的多个字符代码相对应的字符所要使用的形状的多个字形、以及指定绘制与所述多个字符代码相对应的字符所要使用的字形的描述数据;

其中,基于在所述选择步骤中针对各个字符图像所选择的字形来描述所述描述数据,

在绘制与通过进行所述字符识别所获取到的多个字符代码相对应的字符时,不同的字符代码共享及使用所述多个字形中的各个字形。

16. 根据权利要求 15 所述的图像处理方法,其特征在于,所述多个字形分别具有不同的宽度。

17. 根据权利要求 15 所述的图像处理方法,其特征在于,所述多个字形分别表示简单形状。

18. 根据权利要求 15 所述的图像处理方法,其特征在于,所述多个字形分别表示矩形形状。

19. 根据权利要求 15 所述的图像处理方法,其特征在于,所述多个字形分别表示空白字符的形状。

20. 根据权利要求 15 所述的图像处理方法,其特征在于,所述多个字形绘制在字符行的下部。

21. 根据权利要求 15 所述的图像处理方法,其特征在于,所生成的电子文档还包括字符之间距离的平均值。

22. 根据权利要求 15 所述的图像处理方法,其特征在于,所述描述数据还包括如下描述:该描述用于在与所述文档图像中的各个字符图像相对应的位置处,以透明颜色绘制针对所述多个字符代码中的各个字符代码所指定的字形。

图像处理设备和图像处理方法

技术领域

[0001] 本发明涉及用于将纸质文档的扫描图像转换成可电子搜索的数据的技术。

[0002] 背景技术

[0003] 近年来,扫描器和例如硬盘等的大容量存储装置的广泛使用引起了对纸质保存的文档进行扫描并将扫描的数据存储为电子文档。另外,还可以对通过扫描纸质文档所获得的图像数据进行字符识别,从而读取文档中所包括的字符信息并将其与图像相关联地存储。因此,用户可以使用搜索关键字来搜索这类与字符信息相关联了的电子文档。在要从大量的存储文档中快速搜索想要的文档时,可以如上所述对扫描的图像进行关键字搜索是非常重要的。

[0004] 例如,日本特开 2000-322417 讨论了在用户如上所述对与字符信息相关联的电子文档进行关键字搜索的情况下,突出显示文档图像中含有搜索关键字的部分。突出显示该部分以使得用户可以识别含有搜索关键字的部分。因此,即使在文档中存在多个含有同一关键字的部分的情况下,用户也可以通过切换页面图像来有效地识别含有关键字的部分。

[0005] 另一方面,还有一种用于将字符识别的结果作为透明文本(即将透明颜色指定为绘制颜色的字符代码)嵌入在图像文件中的技术。然后以便携式文档格式(PDF)存储该图像文件。当显示这类 PDF 文件时,将透明文本绘制在文档图像中的字符图像上。因此,当用户进行关键字搜索时,搜索透明文本。然而,由于用户不能看见透明文本,因而看起来好象在搜索图像。结果,基于通过能够绘制图像和字符的页面描述语言来描述格式的文件,可以绘制可通过关键字搜索的图像。

[0006] 在以例如 PDF 或可缩放矢量图形(SVG)等的页面描述语言描述的电子文档中,绘制字符需要各字符的字符形状信息,即字体数据的字形。然而,由于字体数据的大小通常较大,因而在电子文档中通常是指定字体的类型而不是存储字体数据,以使电子文档的大小保持较小。结果,在使用应用程序来绘制字符时,可以使用个人计算机(PC)中所安装的字体。

[0007] 另一方面,存在期望将字体数据存储于电子文档中的情况。例如,如果在一不同的 PC 中没有安装电子文档中所使用的字体数据,则在该 PC 上不能正确地打开通过使用文档创建应用程序所创建的电子文档。换句话说,如果将字体数据本身存储在电子文档中,则没有安装指定的字体数据的 PC 或应用程序也可以正确地再现电子文档。

[0008] 此外,根据用途,存在希望将字符绘制中所使用的字体数据存储于电子文档中的情况。例如,PC 中默认安装的字体可能因操作系统(OS)的改变而改变。因此,希望将字体数据存储于长期存储文件中。

[0009] 此外,存在需要将字体数据存储于电子文档中的格式。例如,当以可扩展标记语言(XML)纸张规范(XPS)格式存储文本数据时,需要利用文本数据来存储字体数据。

[0010] 然而,当将字体数据存储于电子文档中时,电子文档的大小增大。如果电子文档的文件大小增大,则在网络上发送电子文档要占用更长的时间,或者在存储电子文档时需要更大的存储容量。

[0011] 因此,对于使用存储在电子文档中的字体数据来绘制字符的文件格式的电子文档,希望避免文件大小的增大。特别地,希望在下面的情况下避免文件大小的增大:将扫描的图像、作为字符识别结果的文本数据和在文本绘制中要使用的字体数据一起存储在电子文档中。如果由于格式或系统上的限制而需要将字体数据存储在电子文档中,则文档大小的增大会成为问题。

[0012] 此外,在字符识别结果将作为透明文本而嵌入文档图像的情况下,希望使透明文本的绘制位置与文档图像中相应字符图像的位置正确匹配。通过位置匹配,在搜索文本时,搜索到的文本的位置与字符图像的位置相匹配。为了实现这一正确匹配,必须针对每一字符详细指定透明文本的绘制位置(例如,字符的位置坐标、字符宽度或字符间距)。然而,如果对于所有字符分别描述各字符的位置,则尤其在存在大量字符的情况下,要生成的电子文档的文件大小变大。

发明内容

[0013] 本发明的实施例提供图像处理设备,该图像处理设备使透明文本的绘制位置与文档图像中与该透明文本相对应的字符图像的位置相匹配,并且创建文件大小减小了的电子文档。

[0014] 根据本发明的一方面,一种图像处理设备,其包括:字符识别单元,用于对文档图像中的多个字符图像进行字符识别,以获取与各个字符图像相对应的字符代码;以及生成单元,用于生成电子文档,其中,所述电子文档包括所述文档图像、由所述字符识别单元所获取到的多个字符代码、多个字形以及表示绘制各个所述字符代码时所要使用的字形的数据;其中,在绘制与由所述识别单元所获取到的多个字符代码相对应的字符时,不同的字符代码基于所述数据共享及使用所述多个字形中的各个字形。

[0015] 根据本发明的典型实施例,可以通过选择绘制文本时所使用的字形来使透明文本的绘制位置和文档图像中的各个字符图像的位置相匹配。此外,准备不同字符宽度的简单字符形状(例如,矩形)的多个字形,并共享及使用所述多个字形以绘制各种字符类型。因此,要描述的数据量变小,并且即使在需要将字体数据存储在电子文档中的情况下,也可以降低电子文档的文件大小。

[0016] 通过以下(参考附图)对典型实施例的说明,本发明的其它特征将变得清楚。

附图说明

[0017] 包含在说明书中并构成说明书的一部分的附图,示出本发明的典型实施例、特征和方面,并与说明书一起用来解释本发明的原理。

[0018] 图1是示出根据本发明第一典型实施例的图像处理设备的结构的例子的框图。

[0019] 图2是示出根据本发明第一典型实施例的电子文档生成处理的例子的流程图。

[0020] 图3是示出根据本发明第一典型实施例的电子文档搜索和查看处理的例子的流程图。

[0021] 图4是详细示出根据本发明第一典型实施例在图2所示流程图的步骤S208中进行的电子文档数据生成处理的流程图。

[0022] 图5是详细示出根据本发明第一典型实施例在图3所示流程图的步骤S306中进

行的页面绘制处理的流程图。

[0023] 图 6 示出根据本发明第一典型实施例由图像处理设备生成的电子文档的例子。

[0024] 图 7 示出根据本发明第一典型实施例的要处理的页面图像的例子。

[0025] 图 8 示出根据本发明第一典型实施例的区域分割处理的结果的例子。

[0026] 图 9 示出根据本发明第一典型实施例作为区域分割处理的结果所生成的区域数据的例子。

[0027] 图 10 示出根据本发明第一典型实施例进行字符识别时的字符图像提取的例子。

[0028] 图 11 示出根据本发明第一典型实施例作为字符识别处理的结果所生成的字符代码串数据的例子。

[0029] 图 12 示出根据本发明第一典型实施例的字符代码序列表的例子。

[0030] 图 13 示出根据本发明第一典型实施例突出显示搜索结果的页面显示的例子。

[0031] 图 14A 和 14B 示出根据本发明第一典型实施例使用不同突出显示处理突出显示搜索结果的页面显示的例子。

[0032] 图 15 是示出根据本发明第一典型实施例的字形识别 (ID) 判断处理的流程图的例子。

[0033] 图 16 示出根据本发明第二典型实施例突出显示搜索结果的页面显示的例子。

具体实施方式

[0034] 下面参考附图详细说明本发明的各种典型实施例、特征和方面。

[0035] 第一典型实施例

[0036] 图 1 是示出根据本发明第一典型实施例的图像处理设备的结构的框图的例子。

[0037] 图像处理设备 100 将文档图像数据转换成可搜索的电子文档。图像处理设备 100 包括扫描器 101、中央处理单元 (CPU) 102、存储器 103、硬盘 104、网络接口 (I/F) 105 和用户接口 (UI) 106。

[0038] 扫描器 101 读取纸质文档上的页面信息,并且将该信息转换成文档图像数据。CPU 102 是执行计算机程序的处理单元,其中,该计算机程序用于分析图像数据并且将分析结果转换成可搜索的电子文档。存储器 103 是存储介质,其存储计算机程序的数据或正处理的数据或者用作 CPU 102 的工作区。硬盘 104 是大容量存储介质,用于存储计算机程序或例如电子文档等的的数据。网络 I/F 105 是用于连接网络 120 的接口,并且用于将例如扫描图像或转换后的可搜索的电子文档等的的数据发送至外部装置或者从外部装置接收数据。UI 106 是用于从用户接收指令的接口,并且包括例如输入键或触摸面板等的输入装置以及例如液晶显示器 (LCD) 等的显示装置。

[0039] 图像处理设备 110 允许搜索或查看由图像处理设备 100 生成的电子文档。CPU 111 执行用于搜索或查看电子文档的计算机程序。存储器 112 是用作用于执行计算机程序或临时存储数据的工作区的存储介质。硬盘 113 是用于存储计算机程序或例如电子文档等的的数据的大容量存储介质。网络接口 (I/F) 114 从外部装置接收例如电子文档等的的数据,或者向外部装置发送数据。用户接口 (UI) 115 是用于从用户接收指令的接口,并且包括例如输入键或触摸面板等的输入装置或者例如 LCD 等的显示装置。

[0040] 下面参考图 2 和 3 所示流程图来说明根据第一典型实施例的处理。

[0041] 图 2 是示出图像处理设备 100 根据通过扫描纸质文档所获得的图像数据来生成可搜索的电子文档的处理的例子的流程图。然后,图像处理设备 100 将电子文档发送至图像处理设备 110。

[0042] 在步骤 S201, CPU 102 根据用户指令操作来确定要生成的 电子文档的传输目的地和传输方法。用户通过 UI 106 给出指令。用户根据例如电子邮件和使用文件传输协议 (FTP) 的文件传输等的选项来选择传输方法。

[0043] 用户将纸质文档放置在图像处理设备 100 上并按下开始按键。在步骤 S202, CPU 102 使用扫描器 101 扫描所放置的纸质文档,生成文档图像数据,并且将该数据存储在存储器 103 中。在使用自动文档进给器输入包括多页的文档的情况下,将每一页都转换成页面图像数据并且按照输入的顺序存储在存储器 103 中。

[0044] 图 7 示出页面图像的例子。参考图 7,页面图像 701 包括字符串“X1A”702、字符串“YBc”703 和照片 704。为了易于描述,将照片 704 表示为阴影矩形。然而,照片 704 实际上是自然图像。此外,尽管图 7 所示例子仅示出了字符串 702 和 703 以及照片 704,但是页面图像还可以包括例如图形等的其它区域。

[0045] 在纸质文档为彩色文档的情况下,页面图像数据为通过 8 位 RGB 值表示灰度级的彩色图像的形式。如果纸质文档为单色文档,则页面图像数据为使用 8 位表示亮度的灰度图像或者使用 1 位表示黑色或白色的二值图像的形式。

[0046] 在步骤 S203, CPU 102 选择存储在存储器 103 中的未处理的页面图像数据作为处理对象图像。在图像包括多页的情况下, CPU 102 按照输入顺序选择一个页面图像作为处理对象。

[0047] 在步骤 S204, CPU 102 通过对处理对象图像进行分析以及对例如文本区域、图形区域、照片区域或表格区域等的不同类型的各区域进行识别来进行区域分析。然后, CPU 102 生成各所识别出的区域的区域数据,并且将该数据存储在存储器 103 中。区域数据包括各区域的外接矩形的左上位置坐标 (x、y 坐标值)、表示外接矩形的大小 (宽度和高度) 的像素数量以及所识别 出的区域的类型。

[0048] 在区域分析处理中使用传统技术 (即区域识别处理、区域确定处理或区域提取处理)。例如,可以使用日本特开平 06-68301 中所讨论的技术,从而可以从二值化文档图像数据中提取类似大小的黑色像素块垂直或水平地排成一排的区域作为文本区域。

[0049] 通过对图 7 所示的页面图像 701 进行区域分析,如图 8 所示,识别出了文本区域 801 和照片区域 802。图 9 示出通过对图 7 所示的页面图像 701 进行区域分析所获得的区域数据的例子。

[0050] 在步骤 S205, CPU 102 对通过进行区域分析所识别出的各文本区域中的字符图像进行字符识别,并且获取各文本区域的字符代码串的数据。然后, CPU 102 将该数据存储在存储器 103 中。字符代码串的数据包括字符代码信息,字符代码信息是对文本区域中所包括的各字符图像的识别结果。字符代码串的数据还包括与各字符图像的外接矩形有关的信息 (即与外接矩形的左上坐标以及宽度和高度有关的信息,以及与包括字符的行的有关的信息)。

[0051] 下面将说明字符识别处理的例子。在用于识别字符图像的处理中可以使用传统技术。

[0052] 在文档图像不是二值图像的情况下,二值化文本区域以获取文本区域的二值图像。通过计数垂直线和水平线各自上的黑色像素的数量,创建每个二值化的文本区域的直方图。然后基于垂直和水平的直方图,获得条状行图像。在直方图中,将直方图为周期性的方向指定为行方向。直方图中黑色像素的数量大于或等于预定阈值的部分构成字符行。然后针对各行图像,在与行方向垂直的方向上生成直方图,并且基于该直方图的结果切出各字符的图像。所切出的区域成为一个字符的外接矩形 信息。

[0053] 在本典型实施例中,使用通过计数黑色像素的数量所获得的直方图来确定字符区域。然而,可以使用投影来确定字符区域,其中,该投影表示在各像素行上是否存在黑色像素。

[0054] 然后通过提取边缘成分,从各字符图像的外接矩形内的图像获取特征矢量。将所获得的特征矢量与先前登记的字符识别字典中的特征矢量进行比较,以获得相似度。将具有最高相似度的字符类型的代码确定为该矩形内的字符图像的字符代码。结果,对于文本区域内的所有字符的外接矩形,获得分配有字符代码的数据,并且从各文本区域所获得的字符代码组形成字符代码串。

[0055] 在英文文本字符区域的情况下,判断在字符之间是否存在单词间的空格。例如,可以通过判断字符之间距离是否大,或者通过将作为字符图像的字符识别结果的字符串与词典进行匹配以检测单词的断开,来确定单词间的空格。如果判断出存在单词间的空格,则将该空格的字符代码插入字符代码串。

[0056] 图 10 和 11 示出对图 8 所示的文本区域 801 进行字符识别的例子。首先,从图 10 所示的文本区域 1000 切出字符行 1001 和 1002。然后从字符行 1001 切出三个字符 1011、1012 和 1013,并且对各字符进行识别。结果,获得与各字符相对应的字符代码,并且获得图 11 所示的字符代码串数据 1101。类似地,对从字符行 1002 切出的三个字符 1021、1022、1023 进行字符识别,并且生成如图 11 所示的字符代码串数据 1102。

[0057] 上述处理是一个例子,并且可以使用其它传统字符识别技术来获取字符代码行。

[0058] 在步骤 S206, CPU 102 将作为当前处理对象的页面图像数据与区域数据和字符代码行数据相关联地临时存储在存储器 103 或硬盘 104 中。

[0059] 在步骤 S207, CPU 102 判断是否存在未处理的图像数据。如果存在未处理的图像数据(步骤 S207 中为“是”),则处理返回至步骤 S203,并且 CPU 102 处理下一页面图像数据。如果没有未处理的图像数据(步骤 S207 中为“否”),则处理进入步骤 S208。

[0060] 在步骤 S208, CPU 102 按照页面顺序组合存储在存储器 103 或硬盘 104 中的所有页面的数据,并且生成包括多页的可搜索的电子文档。

[0061] 在步骤 S208 所生成的电子文档数据可以保持绘制信息和内容信息这两者,其中,绘制信息用于在显示器上电子显示各页面图像、或者用于在打印机中打印各页面图像,以及内容信息用于允许使用搜索关键字进行搜索。满足这类条件的数据格式的例子有 PDF 和 SVG。在本典型实施例中所生成的电子文档的格式还指定要嵌入的字体数据。需要这样嵌入字体数据的格式的例子有 XPS。因此,下面在假定页面描述格式规范使用 XML 表述的情况下说明本典型实施例。然而,本发明不局限于这一格式。

[0062] 图 6 是在输入了包括与两页相对应的页面图像的文档的情况下,基于本典型实施例中要说明的页面描述格式规范所生成的电子文档的页面描述的例子。如图 6 所示,本典

型实施例中的页面描述格式在一个文件中集中描述电子文档。然而,本发明不局限于这种格式。例如,页面描述格式可以是下面的格式(例如,XPS):在不同的文件中分开存储字体数据,从主文件参考字体数据文件,并且使用 ZIP 压缩将文件收集成一个电子文档。

[0063] 下面参考图 4 所示的流程图说明在步骤 S208 所进行的电子文档数据生成处理的例子。

[0064] 在步骤 S401,CPU 102 描述电子文档的开始标签。在本典型实施例的页面数据描述格式规范中,<Document> 表示电子文档的开始标签,并且<Document> 和表示<Document> 结束的标签</Document> 之间的 XML 描述是文档所包括的各个页面的描述数据。在图 6 所示的描述例子中,<Document>601 是电子文档的开始标签,并且</Document>612 是电子文档的结束标签。

[0065] 在步骤 S402,CPU 102 将仍未被描述的页面中的第一个页面的数据识别为处理对象。

[0066] 在步骤 S403,CPU 102 生成并描述表示处理对象页面数据的开始的标签。在本典型实施例中,<Page> 表示页面数据的开始,并且<Page> 和作为页面数据的结束标签的</Page> 之间的 XML 描述是页面的绘制数据和内容数据。此外,在<Page> 标签中包括 Width 和 Height 属性以及 Dpi 属性以描述页面的物理大小,其中,Width 属性表示页面的像素宽度,Height 属性表示页面的像素高度,Dpi 属性表示分辨率。此外,使用表示页码的 Number 属性来描述页码。

[0067] 在图 6 所示的描述例子中,作为 Page 元素的开始标签的<Page>602 包括下面的描述:页面的宽度(即 Width 属性值)为“1680”,高度(即 Height 属性值)为“2376”,分辨率(即 Dpi 属性值)为“200”,并且页码(即 Number 属性值)为“1”。此外,在<Page>602 和作为结束标签的</Page>606 之间(即<Image> 元素 603 到<Text> 元素 605)描述第一个页面的数据。

[0068] 在步骤 S404,CPU 102 生成表示页面所包括的数据中的图像绘制数据的标签,并且描述图像绘制数据。

[0069] 在当前所说明的页面数据描述格式规范中,一个<Image> 元素表示一个图像的绘制数据。此外,在 Data 属性中描述图像数据的内容,并使用属性 X、Y、Width 和 Height 的坐标信息来描述在页面中绘制图像的位置。如果页面中存在多个图像,则按照出现的顺序写每一图像数据。以压缩图像数据形式在 Data 属性中描述图像数据的内容。在本典型实施例中,通过 JPEG 压缩来对彩色或灰度图像进行压缩,并且通过 MMR 压缩来对二值图像进行压缩,以获取用于在 Data 属性中描述图像数据的内容的代码串。

[0070] 根据图 6 所示的<Image> 元素 603 中的描述例子,要在整个页面上绘制文档的第一个页面的扫描的图像。在<Image> 元素 603 中将该图像的位置和大小描述为:X=“0”、Y=“0”,Width=“1680”,Height=“2376”。此外,将通过对图像进行 JPEG 压缩所生成的代码串的字符串描述为 Data 属性值。在图 6 中,Data 属性的字符串被部分简化以使图示简单。这样,描述了<Image> 元素 603。在进行 JPEG 压缩之后,可以在存储扫描的图像之前改变该图像的分辨率。例如,可以将以 600dpi 分辨率扫描的图像改变成 300dpi 分辨率并进行存储。

[0071] 在步骤 S405,CPU 102 生成表示页面所包括的数据中的文本的绘制数据的描述。

[0072] 在当前所说明的页面描述格式规范中,一个<Text>元素表示一行中的字符的绘制数据。<Text>元素中所描述的属性数据包括“Direction”、“X”、“Y”、“Font”、“Size”、“Color”、“String”和“CGlyphId”。

[0073] <Text>元素中所描述的“Direction”属性表示是垂直地还是水平地书写字符串。如果没有描述“Direction”属性,则使用(例如,水平的,从左到右的)默认方向。“X”、“Y”属性指定字符开始位置的坐标。

[0074] “Font”属性指定绘制字符代码所使用的字体数据 ID。“Size”属性指定字体大小。“Color”属性通过一组四个值来指定绘制字符时的字符颜色,这四个值包括 R 成分值、G 成分值、B 成分值 和表示透明度的阿尔法通道值。

[0075] “String”属性指定字符串(即字符代码串)的内容。“CGlyphId”属性指定在绘制“String”中的各字符时所使用的字符形状数据的 ID,即字形的 ID。

[0076] <Text>元素数据所包括的字符代码串使用在图 2 所示步骤 S205 中生成的字符代码串。还将在步骤 S205 中生成的字符代码串分割成各字符行,即垂直或水平地排列的字符组。

[0077] 在图 6 所示的描述例子中,两个<Text>元素 604 和 605 是用于绘制第一个页面上的字符的描述。<Text>元素 604 和 605 分别对应于图 11 所示的字符代码串数据 1101 和 1102。例如,在与图 11 所示的水平书写的字符串“X1A”的字符代码串数据 1101 相对应的<Text>元素 604 中,指定如下所述的属性。

[0078] 在字符代码串数据 1101 中,为作为围绕这三个字符的外接矩形的左上坐标的 X、Y 属性指定 X = “236”以及 Y = “272”。

[0079] 此外,为表示字体类型的“Font”属性指定“Font01”。此外,向表示通过根据字符行中字符的高度进行类推所获得的字体大小的 Size 属性指定“97”像素。为表示绘制字符时的字符颜色的 Color 属性指定 R 成分值 = G 成分值 = B 成分值 = 0 以及阿尔法通道 = 255(即透明颜色)。

[0080] 此外,为表示字符串(即与各字符相对应的字符代码的串)的内容的“String”属性指定“0x2358、0x2332、0x2341”。

[0081] 为<Text>元素中所描述的“CglyphId”属性指定作为各字符的字符形状数据而使用的字形的 ID。在本典型实施例中,基于在步骤 S205 中获得的各字符的宽度信息来指定字形的 ID。由于在扫描的图像上利用透明颜色来绘制字符的字符形状,因而用户不能看见该字符。因此,从多个例如矩形等的简单形状的字符形状数据(例如,八种字形类型)中选择要使用的字符形状数据(即字形),而不使用作为字符图像本身的形状的字符形状。也就是说,进行控制以从矩形宽度不同的多个字形(或者纵横比不同的多个字形)中选择适于各字符的字形。因此,在本典型实施例中,基于字符所在行的高度和字符宽度之间的比,选择字形 ID。

[0082] 图 15 是示出由图像处理设备 100 所进行的用于选择各字符图像的字形 ID 的操作的例子的流程图。在步骤 S1501 ~ S1507,将字符宽度和行高度之间的比与 7/8、6/8、5/8、4/8、3/8、2/8 和 1/8 进行比较。作为比较的结果,处理进入步骤 S1508 ~ S1515 中之一,以从字形 ID0 ~ 7 中选择字形 ID。

[0083] 在判断出该比(字符宽度 / 行高度) > 7/8 的情况下,选择字形 ID = 0。如果判

断出 $7/8 \geq (\text{字符宽度} / \text{行高度}) > 6/8$, 则选择字形 ID = 1。如果 $6/8 \geq (\text{字符宽度} / \text{行高度}) > 5/8$, 则选择字形 ID = 2。如果 $5/8 \geq (\text{字符宽度} / \text{行高度}) > 4/8$, 则选择字形 ID = 3。如果 $4/8 \geq (\text{字符宽度} / \text{行高度}) > 3/8$, 则选择字形 ID = 4。如果 $3/8 \geq (\text{字符宽度} / \text{行高度}) > 2/8$, 则选择字形 ID = 5。如果 $2/8 \geq (\text{字符宽度} / \text{行高度}) > 1/8$, 则选择字形 ID = 6。如果 $(\text{字符宽度} / \text{行高度}) \leq 1/8$, 则选择字形 ID = 7。在本例子中, 字形的矩形宽度随着字形 ID 编号的减小而增大。

[0084] 例如, 在图 11 所示的字符串 1101 中, 各字符图像的 (字符宽度 / 行高度) 为 “0.82”、“0.14” 和 “0.57”。因此, 根据图 15 所示的选择处理, 将字形 ID 指定为 “1、6、3”。类似地, 对于英文句子中的单词间空格, 将空格的宽度当作为字符宽度, 因而基于该空格所在的字符行的高度与空格宽度的比选择字形 ID。下面将详细说明字形形状。

[0085] 上述属性值是一个例子, 并且可以通过在定义上类似的其它值来描述属性。例如, 代替像素数量, 可以通过例如基于像素高度和图像分辨率的点的数量等的值来描述表示字体大小的 Size 属性。

[0086] 在步骤 S406, CPU 102 描述表示页面的结束的 $\langle /Page \rangle$ 。

[0087] 在步骤 S407, CPU 102 判断是否存在未被描述的页面。在存在未被描述的页面的情况下 (步骤 S407 中为 “是”), 则处理返回至步骤 S403, 从而使得下一页面成为要处理的页面图像。另一方面, 如果不存在未被描述的页面 (步骤 S407 中为 “否”), 则处理进入步骤 S408。

[0088] 在本典型实施例中, CPU 102 对与第二个页面相对应的图像进行步骤 S404 ~ S406 的处理, 并描述如在图 6 所示的描述例子那样的 Page 元素 607 和 $\langle /Page \rangle$ 610 之间的元素。

[0089] 在步骤 S408, CPU 102 描述包括在对电子文档中的字符串进行绘制时所使用的所有字形的字体数据的内容。

[0090] 在本典型实施例的页面数据描述格式规范中, 在 $\langle Font \rangle$ 和 $\langle /Font \rangle$ 之间描述包括在字体数据中的字形数据作为 Glyph 元素。 $\langle Font \rangle$ 元素包括表示字体类型的 ID 属性。Glyph 元素包括表示字形类型的 ID 属性和表示与该 ID 相对应的字形 (字符形状) 的 Path 属性。描述 Path 属性以在绘制矩形内使用直线或曲线来表示字形, 其中, 绘制矩形的原点在左下位置处。

[0091] 在图 6 所示的描述例子中, 在 $\langle Font \rangle$ 元素 611 中定义字体 ID = “Font01”, 其中, 定义了字形 ID = “0” ~ “7” 的 8 种字形。例如, 表示 ID = “7” 的字形的形状的 Path 属性 “M0,0V-1024 H128V1024f” 描述如下定义的字形: “移动至原点 (0,0), 向上绘制 1024 单位的垂直线, 向右绘制 128 单位的水平线, 向下绘制 1024 单位的垂直线, 从当前点向开始点绘制直线并且标记出闭合区域”。也就是说, Path 属性是表示标记出 1024×128 矩形的矩形字形的描述。其它 ID 是在水平方向上逐步为 ID = “7” 的字形的整数倍的长方形字形。例如, ID = “0” 是表示标记出 1024×1024 的正方形字体的描述。

[0092] 图 6 所示的 $\langle Font \rangle$ 元素 611 中的描述是例子, 并且可以定义例如三角形、圆或直线等的简单字符形状, 或者可以将空白 (即空格) 定义为字形 (字符形状)。

[0093] 在步骤 S409, CPU 102 描述表示电子文档结束的 $\langle /Document \rangle$, 并且结束对电子文档的生成。将所生成的电子文档作为文件存储在图像处理设备 100 中的存储器 103 或硬盘 104 中。在存储电子文档时, 可以使用传统的文本压缩技术来压缩电子文档。

[0094] 返回参考图 2,在步骤 S209,CPU 102 使用指定的发送方法将在步骤 S208 生成的电子文档发送至在步骤 S201 所指定的发送目的地(例如图像处理设备 110)。由于该数据发送处理使用传统技术,因而省略说明。

[0095] 作为发送目的地的图像处理设备 110 接收通过网络 I/F 114 发送的电子文档,并且将该电子文档存储在硬盘 113 中。由于该数据接收处理使用传统技术,因而省略说明。

[0096] 用于在硬盘中对存储在设备内的电子文档进行识别的识别信息(如文件名等)可以是任意信息。例如,可以将与接收电子文档时的时间有关的字符串分配为识别信息。此外,可以选择并自动分配唯一编号,或者在生成电子文档时由用户指定。

[0097] 下面参考图 3 所示的流程图说明对电子文档进行搜索和查看的处理的例子。在本典型实施例中,图像处理设备 110 进行搜索。然而,本发明不局限于该实施例,并且图像处理设备 100 也可以进行搜索。

[0098] 在步骤 S301,用户输入搜索关键字,假定该搜索关键字包括在用户想要进行搜索的电子文档的文本中。用户从 UI 115 输入该关键字,以从存储在图像处理设备 110 中的电子文档中搜索电子文档中的字符串。在本典型实施例中,将所输入的字符串的长度表示为 k。

[0099] 在步骤 S302,CPU 111 判断在图像处理设备 110 的硬盘 113 中所存储的所有电子文件中是否存在未被搜索的电子文档文件。在存在未被搜索的电子文档文件的情况下(步骤 S302 中为“是”),CPU 111 识别未被搜索的电子文件中的一个电子文件。如果所识别的电子文件是压缩的,则 CPU 111 展开该文件。处理进入步骤 S303。另一方面,如果不存在未被搜索的电子文档(步骤 S302 中为“否”),则处理进入步骤 S312。在步骤 S312,CPU 111 通知用户结束了对所有电子文档的搜索。

[0100] 在步骤 S303,CPU 111 准备搜索在步骤 S302 中所识别的电子文档中的文本数据。CPU 111 将该文档内的文本(字符代码)排成一行,并且初始化搜索开始位置 n,即设置 $n = 0$ 。

[0101] 下面说明在步骤 S303 中进行的处理的例子。

[0102] 通过 XML 解析器解析电子文档数据,并且如果存在 <Text> 元素,则获取在 String 属性中描述的字符代码串。然后,根据在 String 属性中所描述的字符代码串,针对每一字符将字符代码和电子文档数据中描述字符代码值的位置成对地添加至字符代码序列表中。字符代码值的描述位置是由从电子数据的头开始计数的字符数量所表示的、电子文档数据中用以描述字符代码的字符代码串的头的位置。图 12 示出根据图 6 所示的电子文档所生成的字符代码序列表的例子。

[0103] 例如,在图 6 所示电子文档的 <Text> 元素 604 中的 String 属性描述了三个字符代码“0x2358”、“0x2332”和“0x2341”。从如下位置开始描述各字符代码:从电子文档的头开始分别为 1093 字符、1100 字符和 1107 字符的位置。类似地,基于 <Text> 元素 605 和 609 获得其余六个字符代码的描述位置,从而生成如图 12 所示的字符代码序列表。在图 12 所示的表中,按照从 0 开始的顺序分配字符串编号。

[0104] 在步骤 S304,CPU 111 通过将搜索开始位置 n 设置为原点,判断字符代码序列表是否与搜索关键字的字符代码串相匹配。当检测到匹配部分时(步骤 S304 中为“是”),则将匹配部分的变量 n 设置作为匹配字符串的头位置,并且处理进入步骤 S305。另一方面,如果

没有匹配部分（步骤 S304 中为“否”），则处理进入步骤 S309。

[0105] 在步骤 S309, CPU 111 判断是否搜索了字符代码序列中的所有字符。当 CPU 111 判断出搜索了存储在字符代码序列中的所有字符代码串（步骤 S309 中为“是”）时，则处理进入步骤 S311。在步骤 S311, CPU 111 通知用户对作为当前搜索对象的电子文档的搜索结束了。另一方面，如果没有完成所有搜索（步骤 S309 中为“否”），则处理进入步骤 S310。

[0106] 在步骤 S310, CPU 111 将变量 n 加 1, 并且返回至步骤 S304。在步骤 S304, CPU 111 判断在下一搜索开始位置 n 处，字符代码序列是否与搜索关键字的字符代码串相匹配。当存储在字符代码序列中的字符代码的总数为 N 时，如果 $n < (N-k)$ ，则可以判断出尚未完成所有搜索。在步骤 S309, 如果 $n \geq (N-k)$ ，则判断出完成了该搜索。

[0107] 例如，在通过搜索关键字“YB”搜索图 12 所示的字符代码序列的情况下，从字符代码序列的头开始扫描搜索关键字“YB”的字符代码串“0x2359”、“0x2342”，以搜索匹配部分。在这种情况下，重复步骤 S304、S309 和 S310 的处理，并且提取 $n = 3$ 作为第一个匹配字符串的字符串编号。

[0108] 在步骤 S305, CPU 111 识别电子文档中与字符串编号 n 相对应的字符串数据所属的页面。

[0109] 例如，如果在解析电子文档时识别了对 $\langle \text{Text} \rangle$ 元素进行描述的 Page 元素，则可以通过 Number 属性识别页码。因此，可以通过从图 12 所示的表中获取与在步骤 S305 中识别出的位置 n 相对应的字符串的描述位置，来识别该字符串所属的页面。然后可以确定包括该描述位置的 Page 元素。如果在步骤 S303 解析电子文档数据时识别了对各 $\langle \text{Text} \rangle$ 元素进行描述的 Page 元素，以及在图 12 所示的字符代码序列中存储了位置，则基于字符串编号可以容易地识别页码。本发明不局限于上述例子。可以以其它方式进行步骤 S304 中的检测匹配字符串和步骤 S305 中的识别页码。

[0110] 在步骤 S306, CPU 111 根据在步骤 S305 确定的页面绘制描述来绘制页面，并且将结果显示在 UI 115 上。此时，如果要绘制字符串编号 n 和 $n+k-1$ 之间所包括的字符，则 CPU 111 通过添加突出显示效果来绘制该字符，从而使得用户可以容易地识别该字符位于哪里。下面将详细说明向与搜索关键字相匹配的部分添加突出显示效果。

[0111] 参考图 5 所示的流程图说明步骤 S306 中所进行的页面绘制处理。

[0112] 在步骤 S501, CPU 111 根据与所识别的页码相对应的 Page 元素中所包括的属性 Width 和 Height 的值，确定要作为绘制结果而获得的页面图像的大小。

[0113] 在步骤 S502, CPU 111 分配可以存储与页面图像有关的像素信息的存储器。

[0114] 在步骤 S503, CPU 111 从 Page 元素的子元素提取未被处理的元素，并且确定未被处理的元素的类型。如果未被处理的元素为 $\langle \text{Image} \rangle$ 元素（步骤 S503 中为 $\langle \text{Image} \rangle$ ），则处理进入步骤 S504，并且，如果未被处理的元素为 $\langle \text{Text} \rangle$ 元素（步骤 S503 中为 $\langle \text{Text} \rangle$ ），则处理进入步骤 S505。在处理了 Page 元素中的所有子元素的情况下（步骤 S503 中为“无”），处理进入步骤 S517。

[0115] 在步骤 S504, CPU 111 光栅化在 $\langle \text{Image} \rangle$ 元素中被描述为 Data 属性值的压缩图像。此外，CPU 111 缩放光栅化后的图像，以使得该图像完全符合通过 X、Y、Width 和 Height 属性中的每一个所定义的页面图像中的绘制矩形区域。然后，CPU 111 将该图像写入在步骤 S502 中所分配的页面图像存储器中的区域的上，并且处理返回至步骤 S503。

[0116] 在步骤 S505, CPU 111 从作为处理对象的 <Text> 元素所描述的每一属性中, 获取字符开始位置 (X, Y)、字符字体 ID (F)、字符大小 (S) 和字符颜色 (C)。此外, CPU 111 获取在 <Text> 元素中所描述的字符的数量 (N)。

[0117] 在步骤 S506, CPU 111 分配用于生成字形图像的存储器。在本典型实施例中, CPU 111 分配用于存储 124×1024 像素的二值图像的存储器。

[0118] 在步骤 S507, CPU 111 将表示正被处理的字符的计数器 i 初始化成 1。

[0119] 在步骤 S508, CPU 111 判断是否 $i > N$, 并且如果 $i \leq N$ (步骤 S508 中为“否”), 则处理进入步骤 S509。相反, 如果 $i > N$ (步骤 S508 中为“是”), 则 CPU 111 判断出完成了对 <Text> 元素的处理, 并且返回至步骤 S503。

[0120] 在步骤 S509, CPU 111 从 <Text> 元素中的 String 属性获取第 i 个字符的字符代码 (P), 并且从 CglyphId 属性获取第 i 个字符的 GlyphId (Q)。

[0121] 在步骤 S510, CPU 111 在电子文档中对字体 Id 为 (F) 的 元素描述进行定位。此外, CPU 111 根据 元素描述的子元素从字形 Id 为 (Q) 的 Glyph 元素中获取 Path 属性。

[0122] 在步骤 S511, CPU 111 根据在步骤 S510 中获得的 Path 属性值, 在步骤 S506 所分配的字形图像生成存储器中生成字形二值图像。字形二值图像是这样的图像: 通过 1 表示要绘制的部分, 并且通过 0 表示不要绘制的部分。在本典型实施例中, 后面利用透明颜色绘制要绘制的部分 1。

[0123] 在步骤 S512, CPU 111 将字形二值图像的大小缩放成根据字符大小属性 (S) 的值的矩形的大小。获取要绘制的缩放后的字形二值图像的部分 1 的宽度, 作为缩放后的字形宽度 W_i 。

[0124] 在步骤 S513, CPU 111 在页面图像存储器中的基于坐标位置 (X, Y) 的矩形区域上, 绘制在步骤 S512 中缩放后的字形二值图像。按如下所述定义当在页面图像上绘制二值图像时的各像素的像素值。在该定义中, 绘制字形之前的页面图像中的各像素值为 (r, g, b), 而绘制字形之后的像素值为 (r', g', b')。

[0125] 字形二值图像中像素值为 0 的像素: $(r', g', b') = (r, g, b)$

[0126] 字形二值图像中像素值为 1 的像素: $(r', g', b') = (F(r, Cr), F(g, Cg), F(b, Cb))$

[0127] 在上述定义中, $F(r, Cr) = (r \times A + Cr \times (255 - A)) / 255$, $F(g, Cg) = (g \times A + Cg \times (255 - A)) / 255$, $F(b, Cb) = (b \times A + Cb \times (255 - A)) / 255$ 。

[0128] 此外, A 是字符颜色 C 的阿尔法通道值, 并且 Cr、Cg 和 Cb 分别是字符颜色 C 的 R、G 和 B 值。在指定 255 作为阿尔法通道值的情况下, 字形二值图像是透明的。因此, 当字形二值图像的像素值为 1 时, 对于像素来说, $(r', g', b') = (r, g, b)$ 。

[0129] 在步骤 S514, CPU 111 使用例如图 12 所示的字符代码序列表, 判断正被处理的第 i 个字符是否是 n 和 n+k-1 之间所包括的字符。更具体地, 由于 CPU 111 可以根据该序列表识别 n 和 n+k-1 之间的各字符的描述开始位置, 因而 CPU 111 基于正被处理的字符 i 的开始位置是否与那些位置中之一相匹配来进行判断。如果第 i 个字符是 n 和 n+k-1 之间所包括的字符 (步骤 S514 中为“是”), 则处理进入步骤 S515。相反, 如果第 i 个字符不在 n 和 n+k-1 之间 (步骤 S514 中为“否”), 则处理进入步骤 S516。

[0130] 在步骤 S515, CPU 111 进行突出显示处理, 以表示正被处理的字符在被检测为搜

索字符串的部分之内。更具体地,从页面图像存储器中的位置 (X,Y) 开始,对于与所绘制的字符串的区域相对应的矩形区域内的每一像素,将各像素值 (r, g, b) 按如下所述改变成像素值 (r', g', b')。

[0131] $(r', g', b') = (G(r), G(g), G(b))$

[0132] (其中, $G(r) = 255-r$, $G(g) = 255-g$, $G(b) = 255-b$)。

[0133] 上述反转颜色的突出显示处理是一个例子,并且可以进行其它突出显示处理。例如,可以进行下面的处理:不改变字形二值图像中像素值为 0 的像素,并且如上所述,将字形二值图像中像素值为 1 的每一像素值 (r, g, b) 改变成 (r', g', b')。

[0134] 在步骤 S516, CPU 111 确定下一字符的绘制开始位置 X, 将 i 增大 1 (即, $i = i+1$), 并且返回至步骤 S508。通过将缩放后的字形宽度 W_i 和字符之间距离与当前字符的绘制开始位置相加, 计算出下一字符的绘制开始位置 X。在本典型实施例中, 没有存储例如字符间距和字符之间距离等的的数据以减少数据量。通过假定字符之间距离为缩放后的字形宽度 W_i 的 10% 这样来计算绘制开始位置 X。因此, 通过 $X = X + 1.1 \times W_i$ 来计算下一字符的绘制开始位置 X。不一定要按如上所述计算字符之间距离, 并且字符之间距离可以是字符大小的 10% 或者预定的恒量。

[0135] 在步骤 S517, CPU 111 将与一个页面相对应的绘制结果, 即绘制了 Page 元素中的 <Image> 和 <Text> 元素的描述的页面图像存储器的内容, 传输至 UI 115 中的显示缓冲器。然后 CPU 111 显示该结果。

[0136] 如下所述, 作为与绘制图 6 所示电子文档中第一个页面的描述有关的例子, 进行图 5 所示的流程图的处理。

[0137] 在进行步骤 S501 的处理时, CPU 111 基于图 6 所示的第一个页面的 Page 元素 602 中所描述的属性值 Width = “1680” 和 Height = “2376”, 确定该页面的图像大小为 1680×2376 像素。

[0138] 在步骤 S502, CPU 111 在将页面图像表示为 R、G、B24 位彩色图像的情况下, 分配 1680×2376×3 字节的存储器。

[0139] 在步骤 S504, 将如图 6 所示 <Image> 元素 603 的 Data 属性值所描述的压缩代码光栅化成图像, 并且将该图像写在页面图像存储器的整个区域上。在本典型实施例中, 由于图像数据的像素大小为与原始页面的大小相同的 1680×2376 像素, 因而不进行缩放处理。

[0140] 在步骤 S505, CPU 111 从 <Text> 元素 604 所包括的各属性中获取 X = “236”、Y = “272”、字符数量 N = “3”、字符字体 ID = “Font01”、字符大小 = “97” 和字符颜色 = “0, 0, 0, 255”。

[0141] 在步骤 S509, CPU 111 从 <Text> 元素 604 获取 String 属性 = 0x2358 和 Glyph ID = “1” 的第一个字符。

[0142] 然后在步骤 S510, CPU 111 获取用于在步骤 S511 生成字形二值图像的、与所获取到的字符字体 ID = “Font01” 相对应的字形 Path 数据。在本典型实施例中, CPU 111 获取图 6 所示的描述例子中 元素 611 内的 Glyph 元素中的 Id = “1” 的 Path 属性。

[0143] 然后在步骤 S511, CPU 111 基于所获取到的与 Glyph 元素中 Id = “1” 相对应的 Path 属性数据, 生成字形图像。更具体地, 根据 Path 属性的描述, 利用 1 标记出 1024×896 像素的整个矩形区域。因此, 通过标记出如下矩形生成字形图像: 从被分配为用于生成字

形图像的存储器的 1024×1024 像素的区域的左端开始, 该矩形在垂直方向上为 1024 像素以及在水平方向上为 896 像素。

[0144] 在步骤 S512, CPU 111 基于字符大小“97”, 将 1024×1024 像素的字形图像生成存储器缩放成 97×97 像素。因此, 要标记出的区域变成 97×85 像素 (即缩放后的字形宽度 $W_i = 85$)。

[0145] 在步骤 S513, 在页面图像上从位置 $(X, Y) = (236, 272)$ 开始的 97×97 矩形区域中, 绘制缩放后的字形的字符图像。在图 6 所示的描述例子中, 由于字符颜色 $= 0, 0, 0, 255$, 即阿尔法通道值 A 为 255, 因而, 即使与字形二值图像相对应的像素值为 1, 对于所有情况来说, $(r', g', b') = (r, g, b)$ 。也就是说, 页面图像中矩形区域内的像素值在进行步骤 S513 的处理前后不变。

[0146] 在步骤 S514, CPU 111 基于字符代码序列列表判断图 6 所示 $\langle \text{Text} \rangle$ 元素 604 中的第一个字符是否包括在字符串编号 n 和 $n+k-1$ 之间。

[0147] 在本典型实施例中, 假定根据图 6 所示的电子文档生成了图 12 所示的字符代码序列列表。还假定在步骤 S304 中被确定为与搜索关键字相匹配的字符串在字符串编号 3 和 4 之间。由于图 6 所示 $\langle \text{Text} \rangle$ 元素 604 的第一个字符代码不在字符串编号 3 与 4 之间, 因而处理进入步骤 S516。 $\langle \text{Text} \rangle$ 元素 604 中第一个字符代码描述的首字符位置为 1903, 首字符位置 1903 与字符代码序列列表中在字符串编号 3 和 4 之间的字符描述位置不匹配。因此, 可以判断出 $\langle \text{Text} \rangle$ 元素 604 中第一个字符不是在字符串编号 3 与 4 之间的字符。

[0148] 另一方面, 在 CPU 111 在步骤 S514 判断图 6 所示 $\langle \text{Text} \rangle$ 元素 605 中的第一个字符是否包括在字符串编号 n 和 $n+k-1$ 之间的情况下, CPU 111 判断出第一个字符代码与字符代码序列列表中在字符串编号 3 与 4 之间的字符的开始位置相匹配。因此进行步骤 S515 中的突出显示处理。

[0149] 由于与 $\langle \text{Text} \rangle$ 元素 605 中第一个字符相对应的字形 ID 为“0”, 因而通过透明颜色标记出页面图像存储器中从位置 $(236, 472)$ 开始的 92×92 像素区域。因此, 在步骤 S515, 将页面图像存储器中从位置 $(236, 472)$ 开始的 92×92 像素区域内的各像素值 (r, g, b) 改变成 $(G(r), G(g), G(b))$ 。

[0150] 例如, 在绘制 $\langle \text{Text} \rangle$ 元素 604 中的第一个字符代码的情况下 (绘制开始位置为 $(236, 272)$), CPU 111 在步骤 S516 中计算出下一字符的绘制开始位置 X 为 $236 + 1.1 \times 85 = 330$ 。因此, $\langle \text{Text} \rangle$ 元素 604 中的第二个字符的绘制开始位置变成 $(330, 272)$ 。

[0151] 结果, 当绘制了所有 $\langle \text{Text} \rangle$ 元素中的描述时, 获得图 13 所示的页面图像。对于与在步骤 S304 中已被判断为匹配搜索关键字的字符串中字符相对应的区域, 亮度被反转了。与其余字符相对应的区域保持与利用 $\langle \text{Image} \rangle$ 元素所绘制的图像数据相同。

[0152] 如上所述, 根据本典型实施例, 由于突出显示了所搜索到的字符串, 因而用户可以通过查看在步骤 S306 中显示的页面图像容易地确定页面中哪里存在搜索关键字。此外, 由于利用透明颜色绘制了与字符宽度相匹配的字形, 因而在搜索期间, 突出显示的字符的位置与文档图像中的字符图像的位置更加接近地匹配。因此, 对于用户来说更易于识别搜索关键字。

[0153] 图 14A 和 14B 示出当设置不同的突出显示方法时如何显示页面图像的例子。

[0154] 在图 14A 所示的页面绘制描述中, 定义 8 种字形作为字体数据的字形, 其中, 这 8

种字形在高度为 1024 的字符下部绘制高度为 128 且宽度为 1024 ~ 128 的被标记出的矩形。当在图 4 所示的步骤 S405 中描述 <Text> 元素的属性数据时,在相应的字符图像下部绘制与各字形相对应的低高度的矩形透明字符。在这样的页面绘制描述中,如果在步骤 S515 的突出显示处理中反转突出显示各字形的矩形区域,则生成如图 14B 所示的突出显示的页面图像。结果,用户可以看见所搜索到的部分被下划线突出显示了,并且用户可以容易地确定页面中哪里存在所搜索到的字符串。

[0155] 返回参考图 3,在步骤 S307, CPU 111 允许用户选择是结束搜索和查看处理,还是在不同搜索区域中继续搜索。如果用户选择结束该处理(步骤 S307 中为“结束”),则结束图 3 所示的处理,并且,如果用户选择继续搜索(步骤 S307 中为“继续”),则处理进入步骤 S308。

[0156] 在步骤 S308, CPU 111 将 n 定义为 $n = n+k$, 返回至步骤 S304, 并且搜索与搜索关键字相匹配的下一部分。

[0157] 如上所述,根据本发明的第一典型实施例,当将纸质文档转换成电子文档时,在电子文档中描述要以透明颜色绘制在页面图像上的从页面提取的字符。因此,用户可以通过确认被突出显示的与关键字相匹配的部分的页面显示,来搜索电子文档。

[0158] 上述电子文档存储包括许多不同字符宽度的简单字形(矩形等字符形状)的字体数据。电子文档对如下进行描述:在要绘制文档中各种字符类型的透明字符时,选择并绘制与各字符类型的宽度相匹配的简单字形(字符形状)。也就是说,相对于多个字符类型(例如上百个字符类型),共用例如具有不同宽度的八种字形(字符形状)等的许多字形(字符形状)。

[0159] 此外,即使在没有针对各字符精细描述透明文本的绘制位置(例如,利用字符的位置坐标)的情况下,透明文本的绘制位置与文档图像中各字符图像的位置也大致匹配。因此,即使在需要在电子文档中存储电子文档所使用的字体数据时,也可以降低电子文档的文件大小(即数据量)。

[0160] 第二典型实施例

[0161] 在第一典型实施例中,当在图 4 所示的步骤 S405 中描述文本元素中字形 ID 的属性数据时,基于各字符的宽度和高度信息确定与各字符相对应的字形。然而,本发明不局限于该实施例。

[0162] 例如,通过利用进行步骤 S205 的字符识别而获得的各字符图像的位置信息来获得主题字符的左端与下一字符的左端之间的距离(字符间距),并且基于该距离与字符行高度的比,可以选择字形 ID。在主题字符是字符行中最后一个字符的情况下,使用该字符的字符宽度作为该距离。在这种情况下,由于该距离可能大于字符行高度,因而可以使用宽度大于该高度(如宽度 1152 或 1280)的矩形字形。此外,在这种情况下,在图 5 所示的步骤 S516 中,按照 $X = X+W_i$ 来获得下一字符的绘制开始位置 X 。

[0163] 因此,当对通过基于如上所述的字符间距选择字形 ID 而生成的电子文档进行搜索处理时,还突出显示与关键字相匹配的字符串中字符之间的间隔。

[0164] 如上所述,根据第二典型实施例,即使在未对所有字符描述透明文本的绘制位置(如字符的位置坐标)的情况下,透明文本的绘制位置与文档图像中各字符图像的位置也大致匹配。此外,由于要存储的字形的总数是有限数量(例如,10 种字形),因而可以降低字

体数据的数据量。此外,由于简化了字形的形状,因而可以降低字形数据(字符形状数据)本身的数据量。

[0165] 第三典型实施例

[0166] 本发明的第三典型实施例说明不同于上述典型实施例中图 5 所示的步骤 S516 的、用于确定下一字符的绘制开始位置 X 的处理。

[0167] 在本典型实施例中,CPU 111 基于通过进行上述典型实施例中图 2 所示流程图的步骤 S205 中的字符识别所识别出的各字符图像的位置信息,计算字符之间距离的平均值。当然然后在图 4 所示的步骤 S405 中描述 <Text> 元素时,将字符区域中的字符之间距离的平均值描述为属性数据(AvC)。在步骤 S516,CPU 111 可以利用所描述的字符之间距离的平均值(AvC)来确定下一字符的绘制开始位置。在这种情况下,下一字符的绘制开始位置为 $X = X + W_i + AvC$ 。

[0168] 第四典型实施例

[0169] 在上述典型实施例中,生成了这样的电子文档:在 <Image> 元素中描述通过利用 JPEG 压缩对扫描的图像进行压缩而获得的完整图像,并且在 <Text> 元素中描述透明文本。然而,本发明不局限于上述实施例。

[0170] 例如,代替描述利用 JPEG 压缩进行压缩后的整个扫描图像,可以将文本区域和图形区域转换成针对各种颜色的二值图像并且利于 MMR 压缩进行压缩,而可以利于 JPEG 压缩来压缩其它区域。可以将这类描述存储在 <Image> 元素中。日本特开平 07-236062 和日本特开 2002-077633 讨论了分析并适当压缩文档图像中所包括的区域的方法。通过将用于降低在绘制透明文本中所使用的字体数据量的处理和上述图像压缩处理相结合,可以生成进一步高度压缩的电子文档。

[0171] 此外,还可以仅将字符区域、图形区域、表格区域或照片区域等区域与位置数据存储在一起来,而不是整个图像。

[0172] 第五典型实施例

[0173] 在上述典型实施例中,当突出显示与搜索结果相对应的部分时,反转图像的颜色(r, g, b)。然而,可以使用的颜色不局限于上述颜色。例如,可将用于指定搜索结果的预定颜色(如黄色)绘制成半透明(如阿尔法通道 128)。此外,可以使用字符颜色(Cr、Cg、Cb)来确定突出显示颜色。

[0174] 第六典型实施例

[0175] 在上述典型实施例的图 3 和 5 所示的流程图中,从文档的开头开始顺序搜索与关键字相匹配的字符串,并且突出显示首先搜索到的字符串。如果存在用以“搜索下一串”的指令,则顺序搜索并突出显示与关键字相匹配的下一字符串。然而,本发明不局限于按照从文档的开头开始的顺序来搜索与搜索关键字相匹配的字符串以及突出显示所搜索到的与搜索关键字相匹配的字符串。例如,可以将包括在电子文档中的所有字符串与搜索关键字进行比较,并且可以指定并同时突出显示所有匹配字符串。

[0176] 其它典型实施例

[0177] 还可以通过向系统或设备提供存储用于实现上述典型实施例的操作的软件程序代码(计算机程序)的计算机可读存储介质来实现本发明的上述典型实施例。此外,通过由系统或设备的计算机(CPU 或微处理单元(MPU))读取并执行存储在存储介质中的程序代

码（计算机程序），来实现上述典型实施例。

[0178] 该计算机程序实现包括在上述典型实施例的流程图中的各步骤。也就是说，该计算机程序是与用于使计算机执行流程图中所包括的各步骤的各处理单元相对应的程序。在这种情况下，从计算机可读存储介质读出的计算机程序本身实现上述典型实施例的操作，并且用于存储该计算机程序的存储介质构成本发明。

[0179] 此外，提供该计算机程序的存储介质可以是例如软盘、硬盘、磁带等磁性存储介质、非易失性半导体存储器、ROM、以及例如磁光盘（MO）、致密盘（CD）、数字多功能光盘（DVD）、CD 只读存储器（CD-ROM）和可记录光盘（CD-R）等的光学存储介质 / 磁光存储介质。

[0180] 此外，运行在计算机上的 OS 等也可以根据该计算机程序的指令进行部分或全部处理，并且实现上述典型实施例的功能。

[0181] 在上述典型实施例中，CPU 与存储器、硬盘和显示装置等共同执行流程图中的各步骤。然而，本发明不局限于上述结构，并且专业电子电路可以代替 CPU 进行各流程图中所述各步骤的部分或全部处理。

[0182] 尽管参考典型实施例说明了本发明，但是应该理解，本发明不局限于所公开的典型实施例。所附权利要求书的范围符合最宽的解释，以包含所有这类修改、等同结构和功能。

[0183] 本申请要求 2007 年 6 月 29 日提交的日本专利申请 2007-172737 的优先权，其全部内容通过引用包含于此。

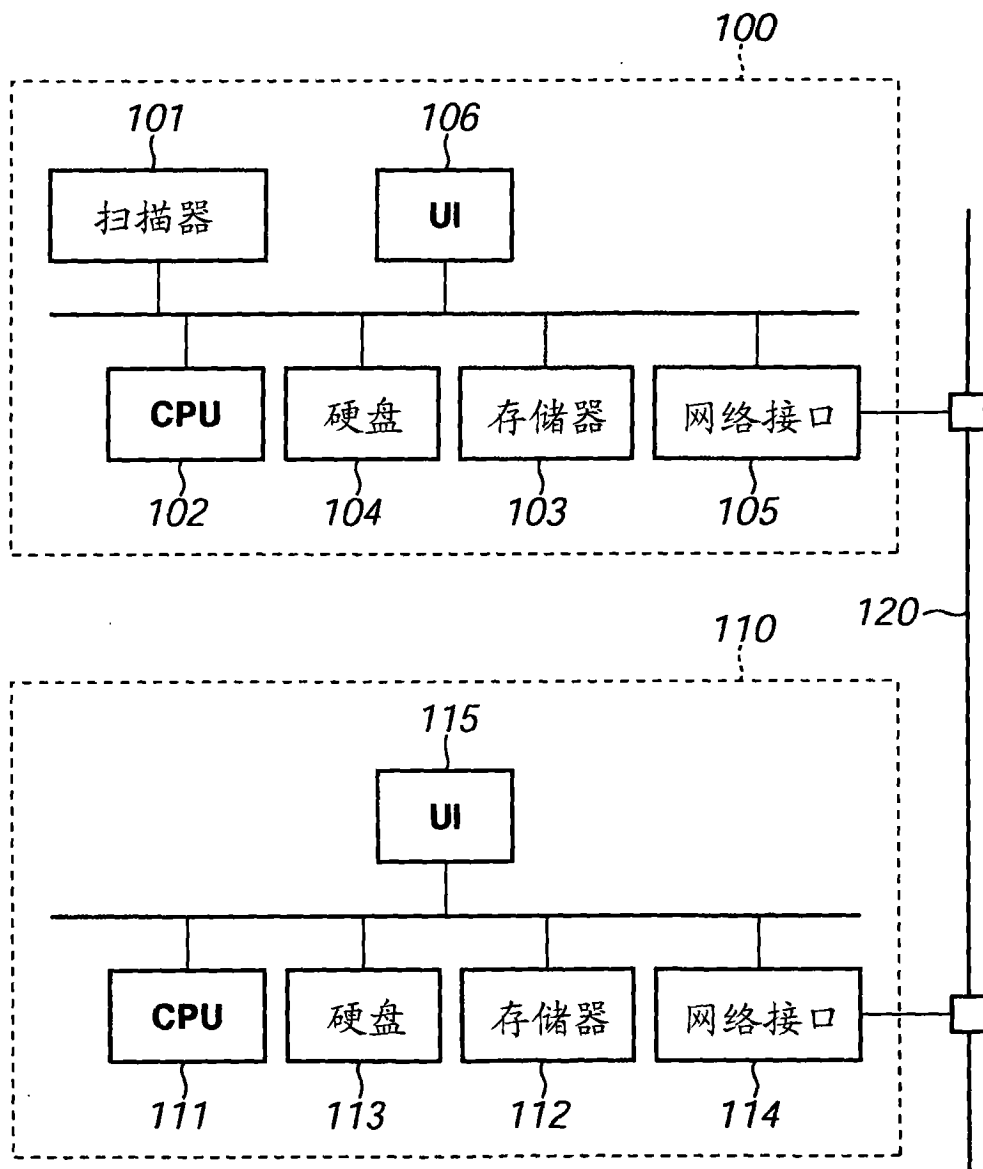


图 1

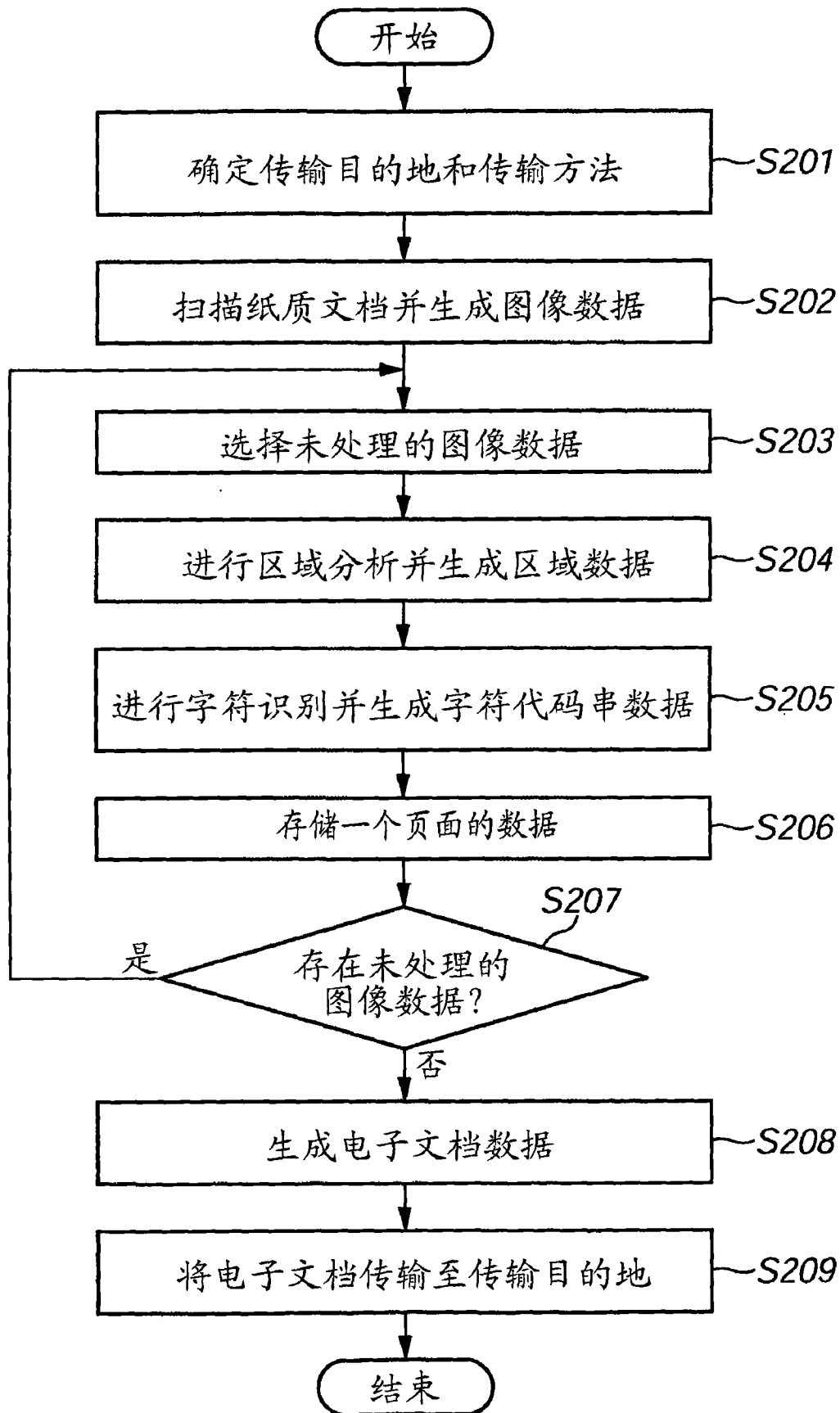


图 2

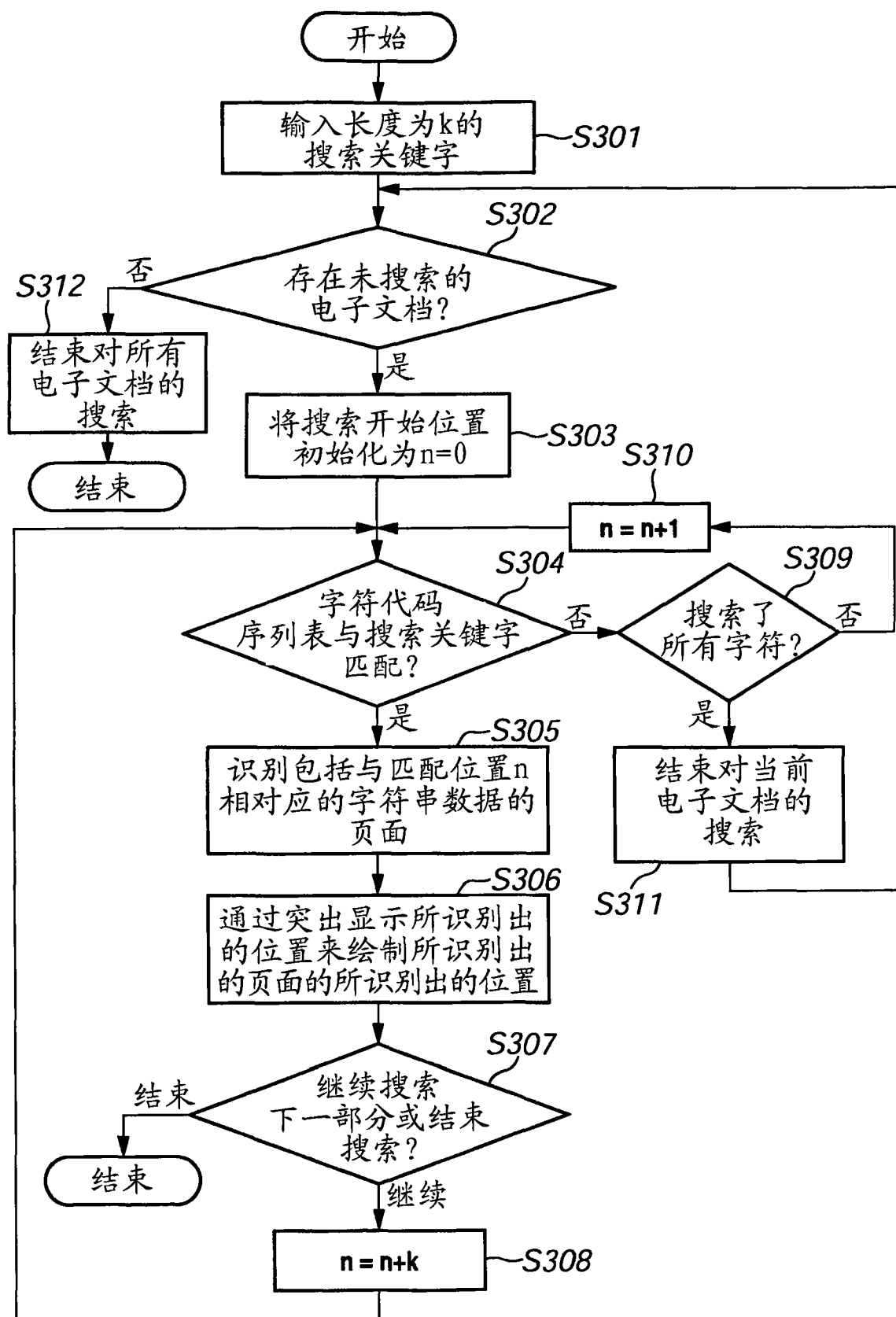


图3

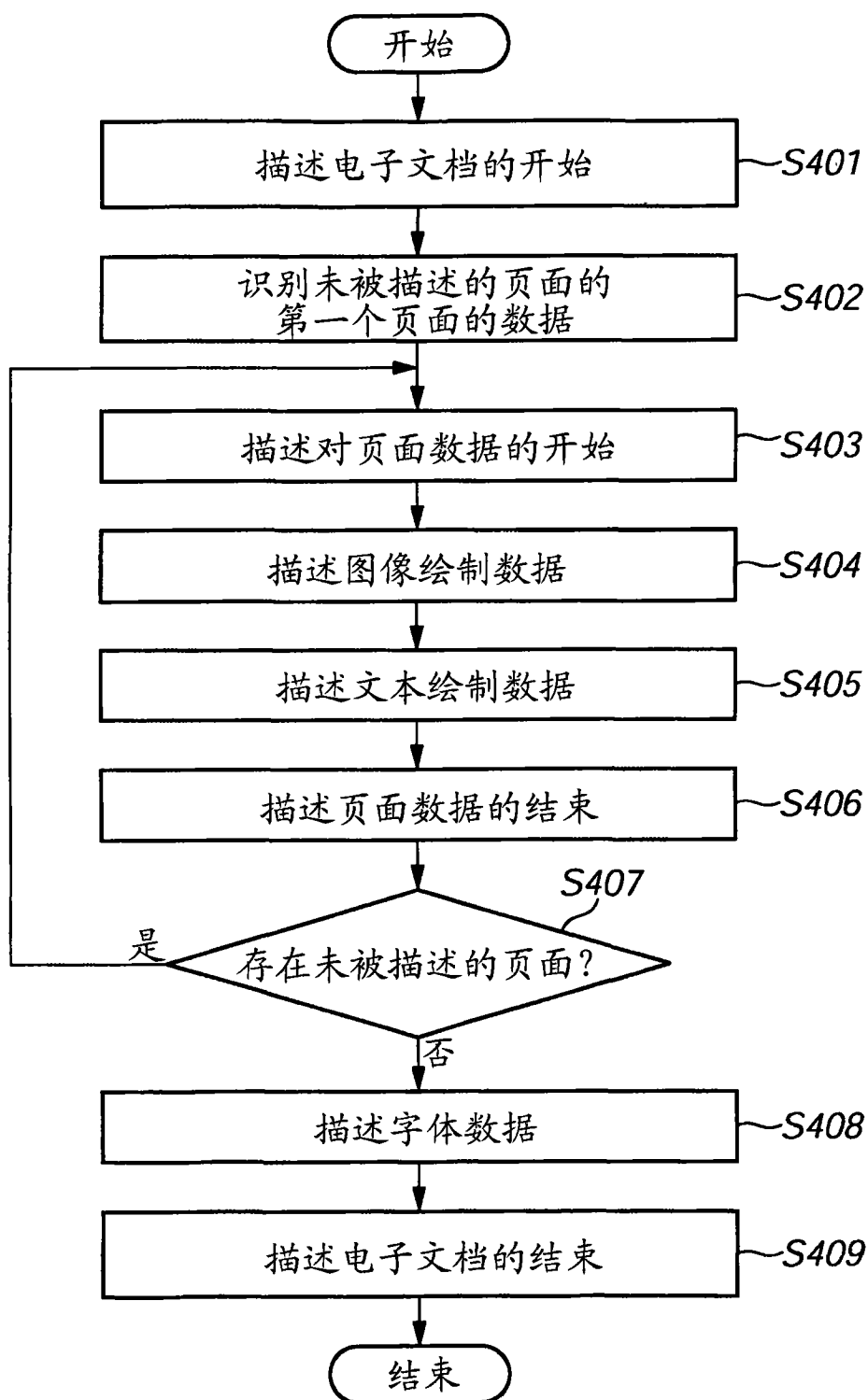


图 4

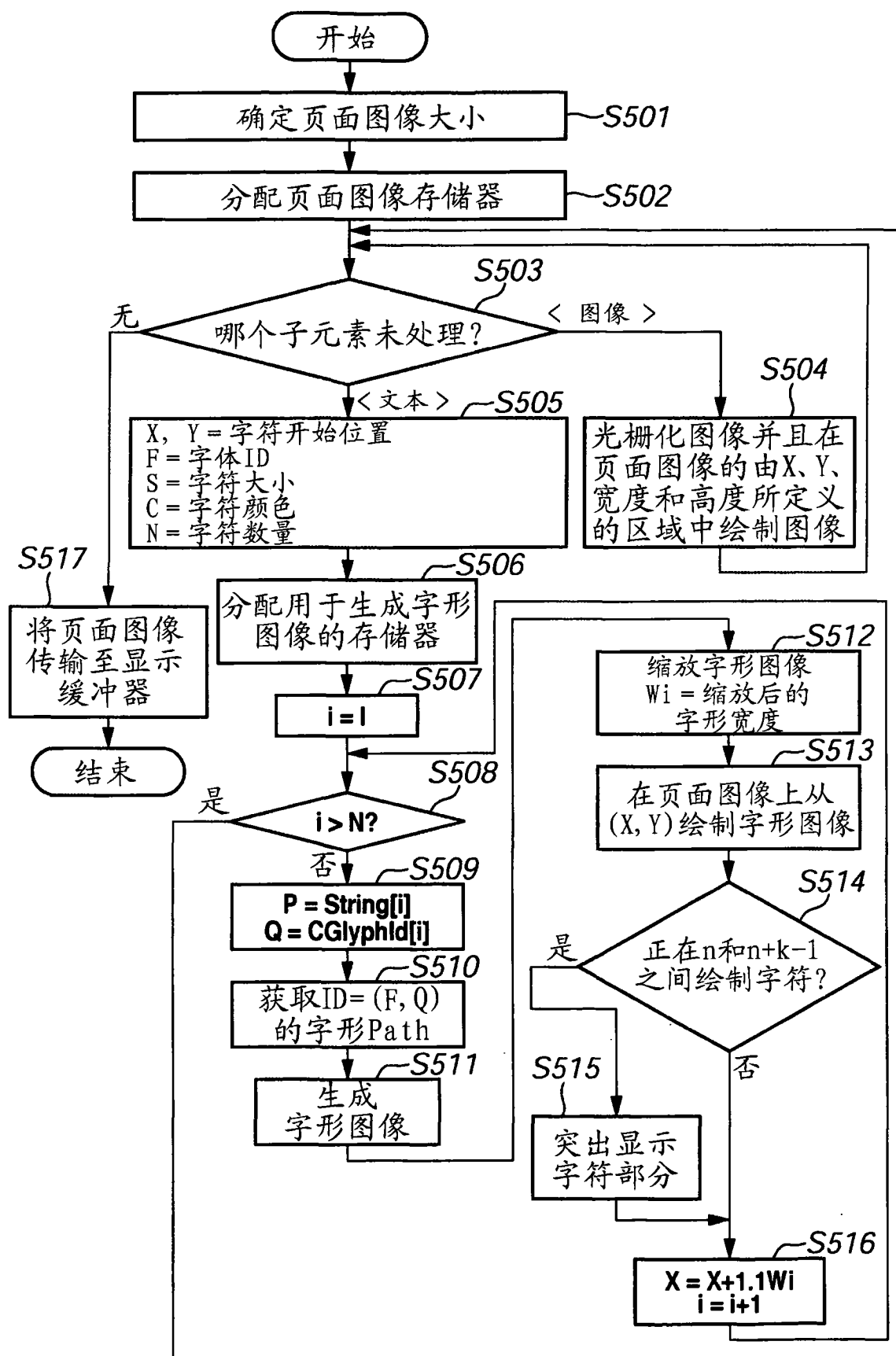


图 5


```
601 { <?xml version="1.0"?>
      <Document>
602 {   <Page Width="1680" Height="2376" Dpi="200" Number="1">
603 {     <Image X="0" Y="0" Width="1680" Height="2376"
        Data="ACAf49sw89q34rs6dbc23cfasx....a3E52zAD"/>
604 {     <Text X="236" Y="272" Font="Font01" Size="97"
        Color="0,0,0,255" String="0x2358,0x2332,0x2341"
        CGlyphId="1,6,3"/>
605 {     <Text X="236" Y="472" Font="Font01" Size="92"
        Color="0,0,0,255" String="0x2359,0x2342,0x2363"
        CGlyphId="0,2,5"/>
606 {   </Page>
607 {   <Page Width="1680" Height="2376" Dpi="200" Number="2">
608 {     <Image X="0" Y="0" Width="1680" Height="2376"
        Data="sa8OAN2qx7sa23Basd2x....aAqw8D2dDJ"/>
609 {     <Text X="136" Y="872" Font="Font01" Size="92"
        Color="0,0,0,255" String="0x242b,0x242d,0x242f"
        CGlyphId="1,1,6"/>
610 {   </Page>
611 {     <Font Id="Font01">
        <Glyph Id="0" Path="M0,0 V-1024 H1024 V1024 f" />
        <Glyph Id="1" Path="M0,0 V-1024 H896 V1024 f" />
        <Glyph Id="2" Path="M0,0 V-1024 H768 V1024 f" />
        <Glyph Id="3" Path="M0,0 V-1024 H640 V1024 f" />
        <Glyph Id="4" Path="M0,0 V-1024 H512 V1024 f" />
        <Glyph Id="5" Path="M0,0 V-1024 H384 V1024 f" />
        <Glyph Id="6" Path="M0,0 V-1024 H256 V1024 f" />
        <Glyph Id="7" Path="M0,0 V-1024 H128 V1024 f" />
        </Font>
612 {   </Document>
```

图 6

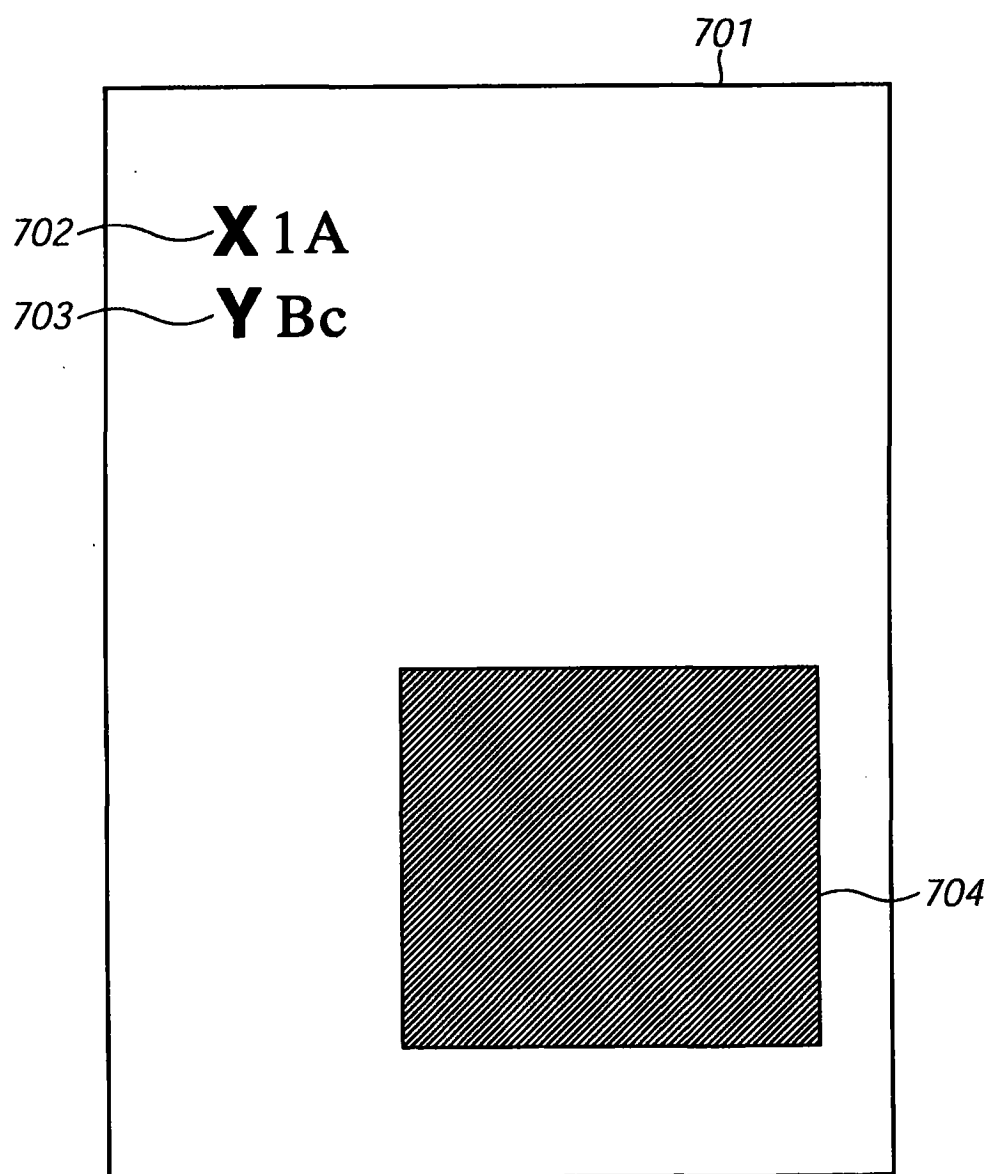


图 7

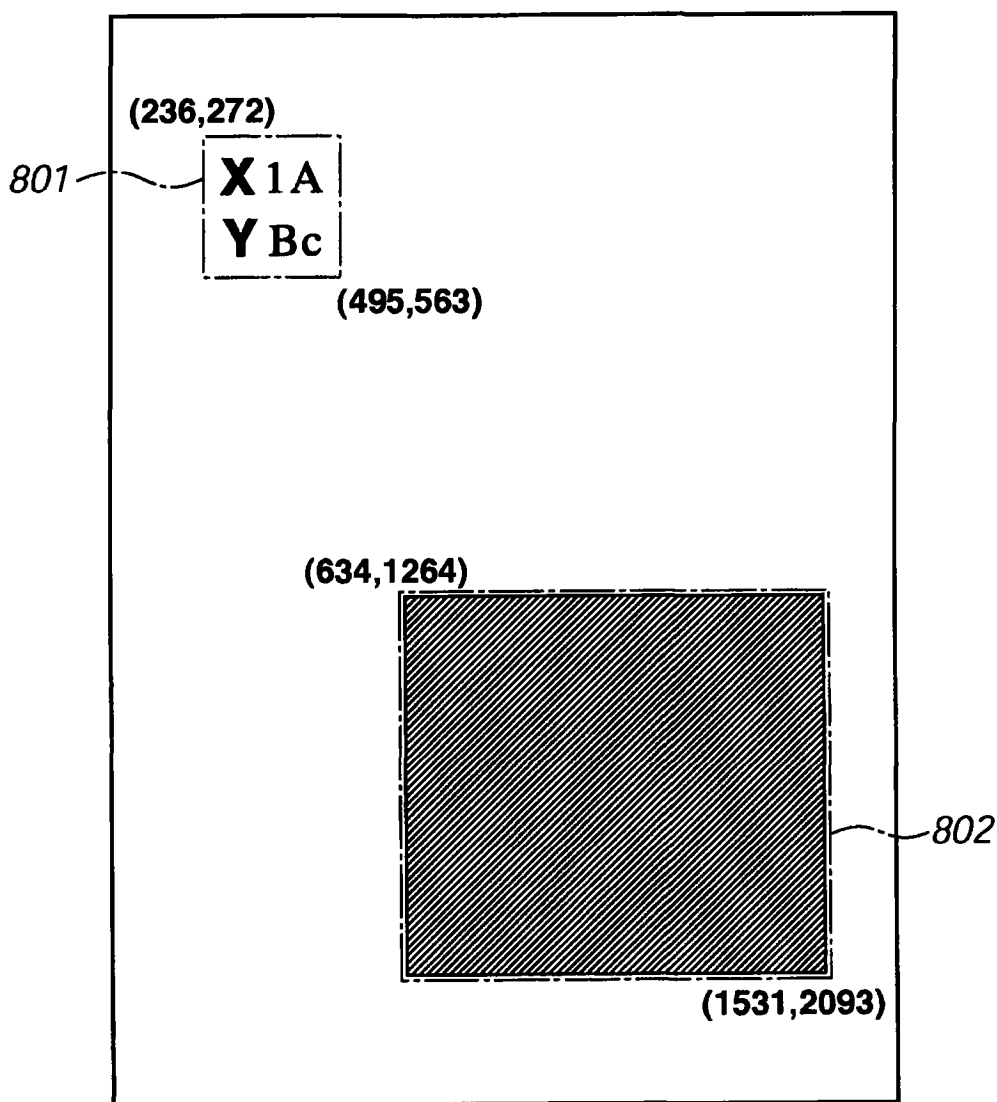


图 8

区域	x	y	宽度	高度	类型
1	236	272	260	292	文本
2	634	1264	898	830	照片

图 9

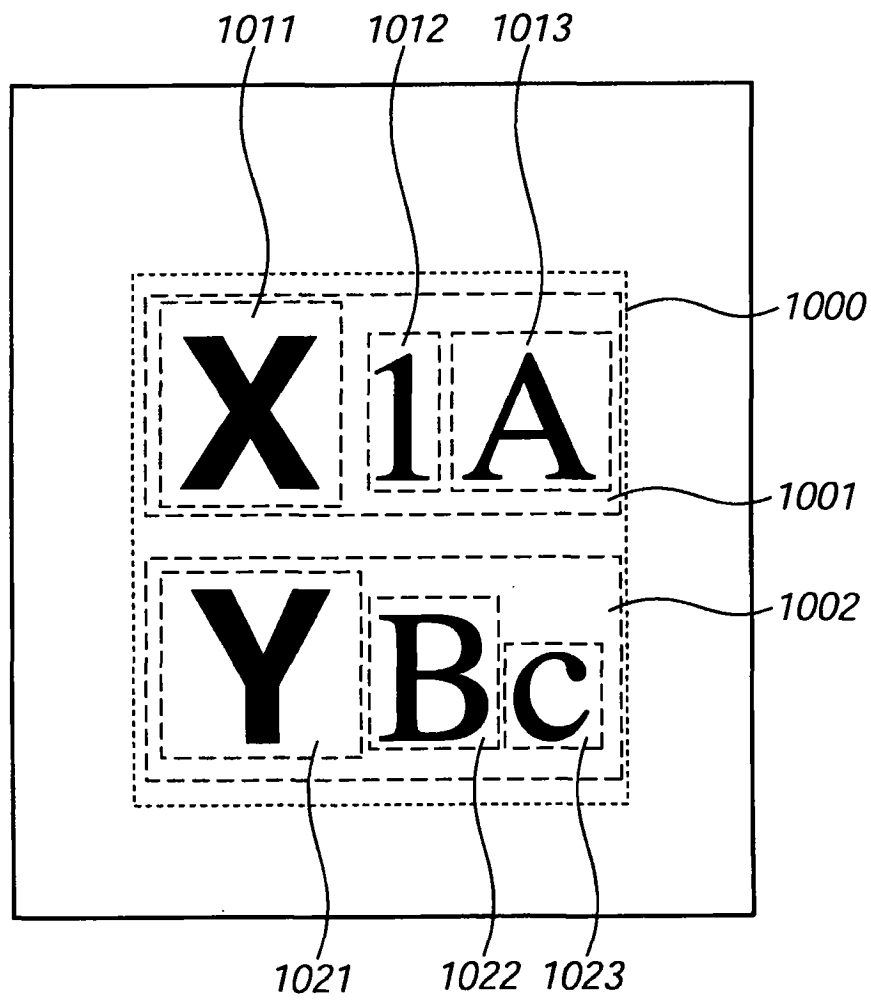


图 10

1101						
字符	x	y	宽度	高度	字符 宽度/行高度	代码
1	236	272	80	97	0.82	0x2358
2	339	284	14	75	0.14	0x2332
3	401	274	55	75	0.57	0x2341

1102						
字符	x	y	宽度	高度	字符 宽度/行高度	代码
1	236	474	92	85	1.00	0x2359
2	348	472	58	92	0.63	0x2342
3	428	472	28	89	0.30	0x2363

图 11

编号	字符代码	描述位置
0	0x2358	1093
1	0x2332	1100
2	0x2341	1107
3	0x2359	1250
4	0x2342	1257
5	0x2363	1264
6	0x242b	2601
7	0x242d	2608
8	0x242f	2615

图 12

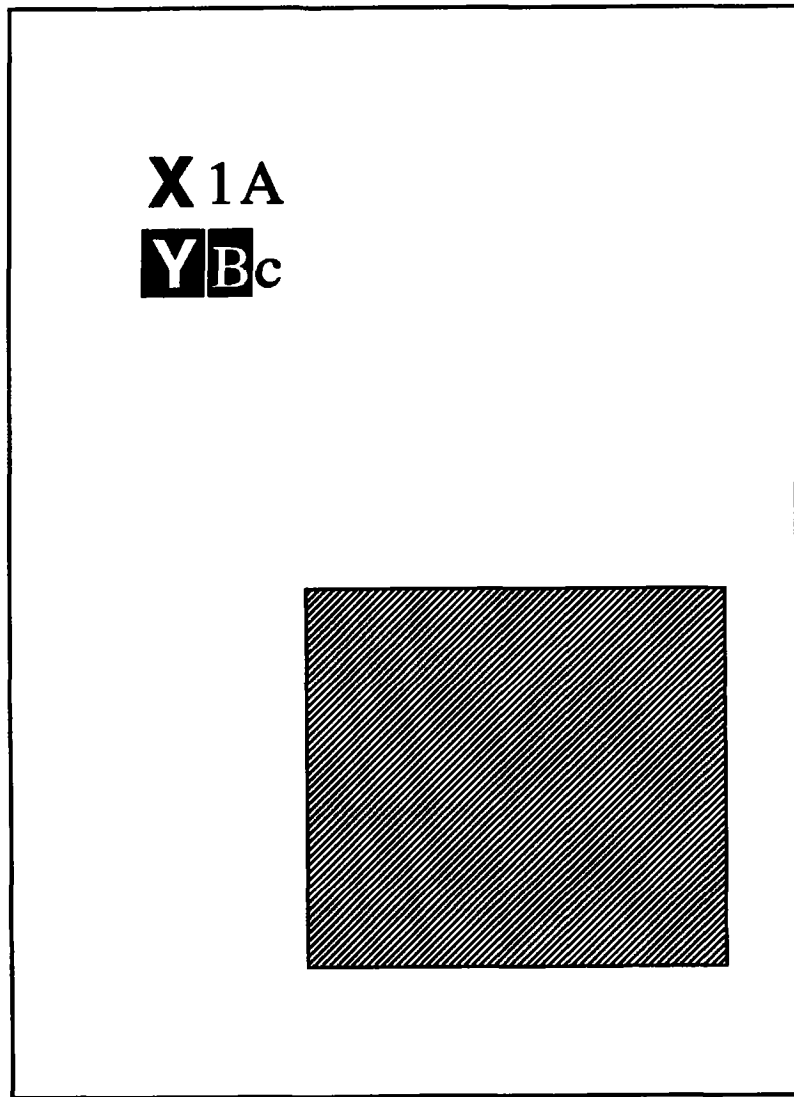


图 13

```
<?xml version="1.0"?>
<Document>
  <Page Width="1680" Height="2376" Dpi="200" Number="1">
    <Image X="0" Y="0" Width="1680" Height="2376"
      Data="ACAf49sw89q34rs6dbc23cfasx....a3E52zAD"/>
    <Text X="236" Y="272" Font="Font01" Size="97"
      Color="0,0,0,255" String="0x2358,0x2332,0x2341"
      CGlyphId="1,6,3"/>
    <Text X="236" Y="472" Font="Font01" Size="92"
      Color="0,0,0,255" String="0x2359,0x2342,0x2363"
      CGlyphId="0,2,5"/>
  </Page>
  <Font Id="Font01">
    <Glyph Id="0" Path="M-896,0 V-128 H1024 V128 f" />
    <Glyph Id="1" Path="M-896,0 V-128 H896 V128 f" />
    <Glyph Id="2" Path="M-896,0 V-128 H768 V128 f" />
    <Glyph Id="3" Path="M-896,0 V-128 H640 V128 f" />
    <Glyph Id="4" Path="M-896,0 V-128 H512 V128 f" />
    <Glyph Id="5" Path="M-896,0 V-128 H384 V128 f" />
    <Glyph Id="6" Path="M-896,0 V-128 H256 V128 f" />
    <Glyph Id="7" Path="M-896,0 V-128 H128 V128 f" />
  </Font>
</Document>
```

图 14A

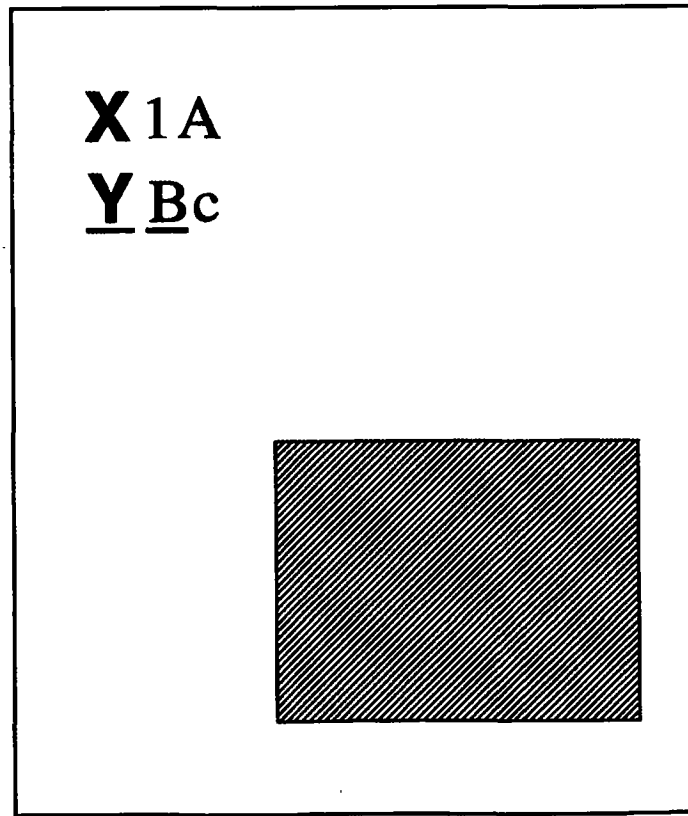


图 14B

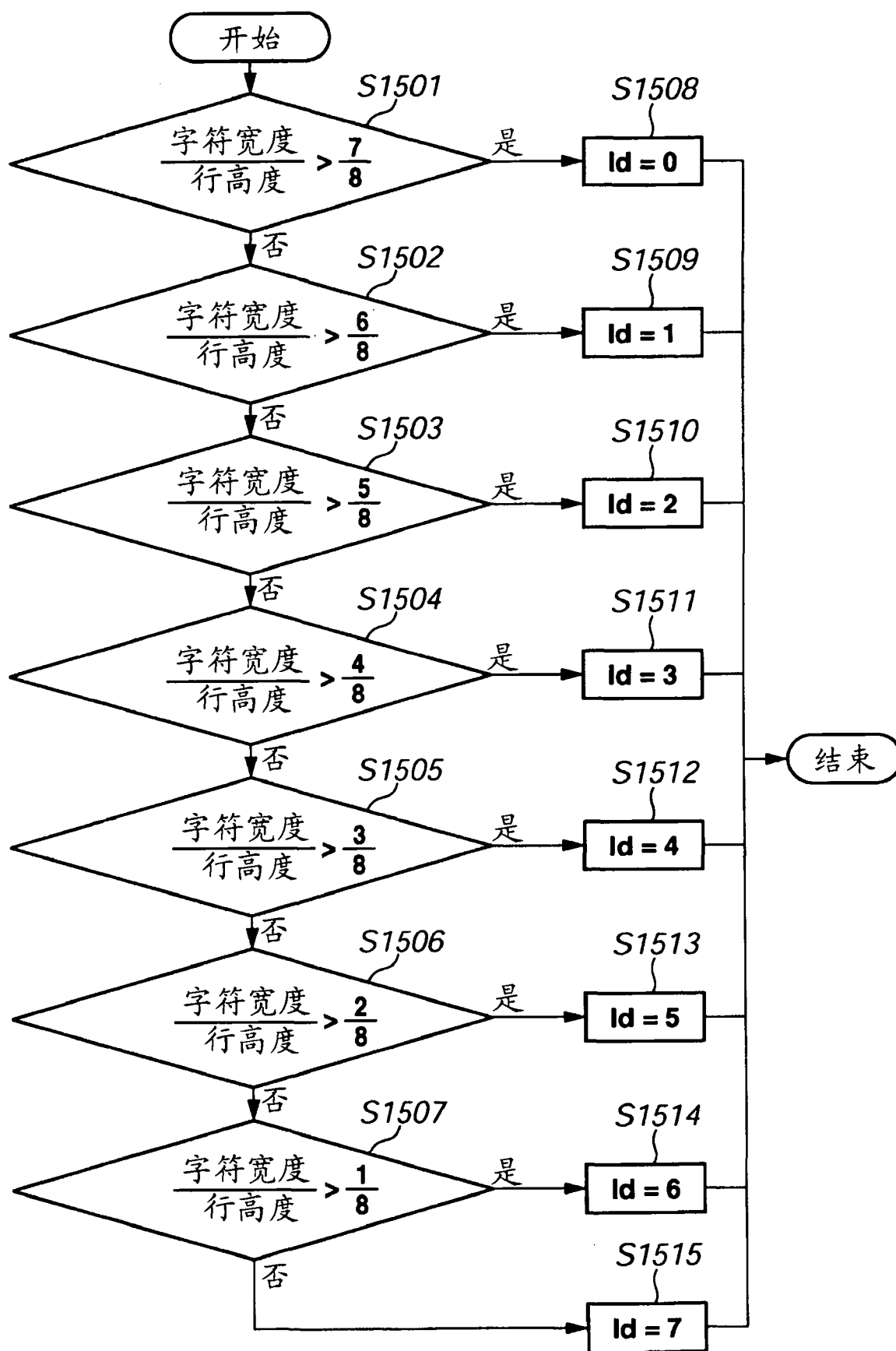


图 15

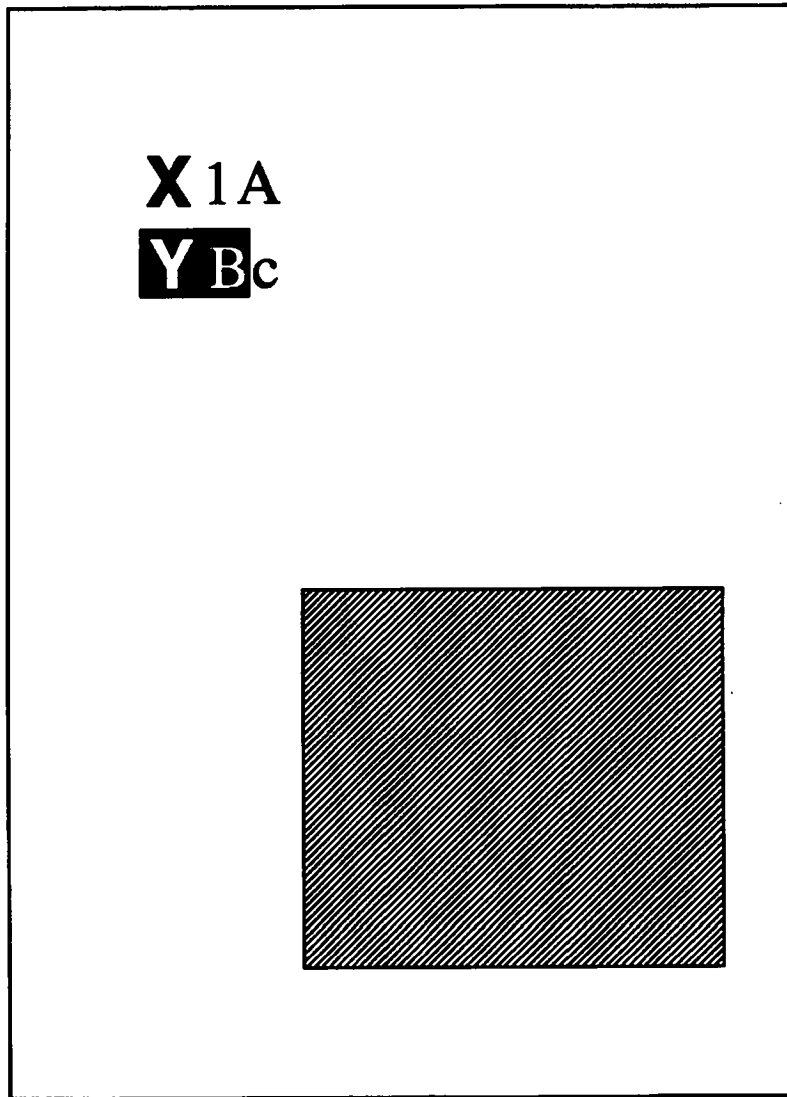


图 16