



(12) 发明专利申请

(10) 申请公布号 CN 104715004 A

(43) 申请公布日 2015. 06. 17

(21) 申请号 201410742932. 3

(22) 申请日 2014. 12. 05

(30) 优先权数据

14/105, 693 2013. 12. 13 US

(71) 申请人 柯尼卡美能达美国研究所有限公司

地址 美国加利福尼亚州

(72) 发明人 嘉堵璠码

(74) 专利代理机构 北京市柳沈律师事务所

11105

代理人 李芳华

(51) Int. Cl.

G06F 17/30(2006. 01)

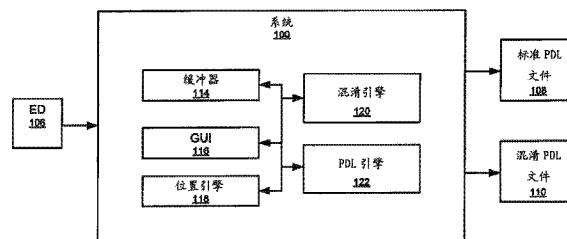
权利要求书2页 说明书8页 附图5页

(54) 发明名称

混淆页面描述语言输出以阻碍转换为可编辑格式

(57) 摘要

一种用于管理电子文档 (ED) 的方法, 包括: 接收用以生成用于 ED 的混淆页面描述语言 (PDL) 文件的请求; 在 ED 内识别包括多个字符的第一文本流; 计算所述多个字符在页面上的多个位置; 响应于所述请求通过对所述第一文本流施加混淆技术来生成修改的文本流; 和生成包括所述多个位置和所述已修改的文本流的混淆 PDL 文件。



1. 一种用于管理电子文档 (ED) 的方法, 包括:
  - 接收用以生成用于 ED 的混淆页面描述语言 (PDL) 文件的请求;
  - 在 ED 内识别包括多个字符的第一文本流;
  - 计算所述多个字符在页面上的多个位置;
  - 响应于所述请求通过对所述第一文本流施加混淆技术来生成修改的文本流; 和
  - 生成包括所述多个位置和所述已修改的文本流的混淆 PDL 文件。
2. 如权利要求 1 所述的方法, 进一步包括:
  - 在接收到所述请求之前向用户显示图形用户界面 (GUI), 该图形用户界面包括用于生成混淆 PDL 文件的选项和用于为 ED 生成标准 PDL 文件的选项,
  - 其中, 响应于用户选择所述用于生成混淆 PDL 文件的选项来生成所述请求。
3. 如权利要求 1 所述的方法, 其中, 所述 ED 是开放的办公可扩展标记语言 (OOXML) 文件, 且所述 PDL 是便携文档格式 (PDF)。
4. 如权利要求 1 所述的方法, 其中施加混淆技术包括:
  - 改变所述多个字符的顺序。
5. 如权利要求 4 所述的方法, 其中改变顺序包括反转所述第一文本流内的多个单词。
6. 如权利要求 1 所述的方法, 其中施加混淆技术包括:
  - 从所述 ED 中的第二文本流中删除字符、并将所述字符插入所述多个字符中。
7. 如权利要求 1 所述的方法, 其中施加混淆技术包括:
  - 将多个字符划分成多个 PDL 组。
8. 如权利要求 1 所述的方法, 其中施加混淆技术包括:
  - 将所述多个字符中的第一字符设为红绿蓝 (RGB) 颜色空间中的 (0, 0, 0); 和
  - 将所述多个字符中的第二字符设为灰色颜色空间中的 (0)。
9. 如权利要求 1 所述的方法, 进一步包括:
  - 响应于所述请求在 ED 内识别第一矢量图和第二矢量图, 其中所述第一矢量图和所述第二矢量图在页面上部分地重叠; 和
  - 生成与所述第二矢量图部分地重叠的第一矢量图的光栅表示,
  - 其中, 所述混淆 PDL 文件还包括所述光栅表示。
10. 如权利要求 1 所述的方法, 进一步包括:
  - 响应于所述请求在 ED 内识别形状以及用于该形状的填充颜色; 和
  - 基于所述填充颜色生成具有张量补丁渐变填充的阴影颜色空间,
  - 其中, 所述混淆 PDL 文件包括所述张量补丁渐变填充。
11. 一种用于管理电子文档 (ED) 的设备, 所述设备包括:
  - 显示部件, 用于向用户显示图形用户界面 (GUI), 该图形用户界面包括用于生成用于所述 ED 的混淆页面描述语言 (PDL) 文件的选项;
  - 接收部件, 用于接收用于生成所述 ED 的混淆 PDL 文件的请求;
  - 识别部件, 用于在 ED 内识别包括多个字符的第一文本流;
  - 计算部件, 用于计算所述多个字符在页面上的多个位置;
  - 第一生成部件, 用于响应于请求通过对第一文本流施加混淆技术来生成修改的文本流; 以及

第二生成部件,用于生成包括所述多个位置和所述已修改的文本流的混淆 PDL 文件。

12. 如权利要求 11 所述的设备,其中,所述第一生成部件包括:

改变部件,用于通过反转所述第一文本流中的多个单词来改变所述多个字符的顺序。

13. 如权利要求 11 所述的设备,其中,所述第一生成部件包括:

删除部件,用于从所述 ED 中的第二文本流删除字符、并将该字符插入到所述多个字符中。

14. 如权利要求 11 所述的设备,其中,所述第一生成部件包括:

第一设置部件,用于将所述多个字符中的第一字符设为红绿蓝 (RGB) 颜色空间中的 (0, 0, 0);和

第二设置部件,用于将所述多个字符中的第二字符设为灰色颜色空间中的 (0)。

15. 如权利要求 11 所述的设备,其中,所述第一生成部件还包括:

划分部件,用于将所述多个字符划分成多个 PDL 组。

16. 一种系统,包括:

计算机处理器;

缓冲器,被配置成储存包括第一文本流的电子文档,该第一文本流包括多个字符;

位置引擎,在计算机处理器上运行,并且被配置成计算所述多个字符在页面上的多个位置;

混淆引擎,在计算机处理器上运行,并且被配置成通过对第一文本流施加混淆技术来生成修改的文本流;和

页面描述语言 (PDL) 引擎,在处理器上运行,并且被配置成为生成包括所述多个位置和已修改文本流的用于 ED 的混淆 PDL 文件。

17. 如权利要求 16 所述的系统,其中,所述 ED 是开放的办公可扩展标记语言 (OOXML) 文件,并且其中所述 PDL 是便携文档格式 (PDF)。

18. 如权利要求 16 所述的系统,进一步包括:

图形用户界面 (GUI),包括用于生成混淆 PDL 的选项和用于生成 ED 的标准 PDL 文件的选项。

19. 如权利要求 16 所述的系统,其中施加混淆技术包括:

通过反转所述第一文本流中的多个单词来改变所述多个字符的顺序;和

从 ED 中的第二文本流删除字符、并将该字符插入到所述多个字符中。

20. 如权利要求 16 所述的系统,其中施加混淆技术包括:

将多个字符划分成多个 PDL 组;

将多个 PDL 组中的第一 PDL 组设为红绿蓝 (RGB) 颜色空间中的 (0, 0, 0);和

将多个 PDL 组中的第二 PDL 组设为灰色颜色空间中的 (0)。

## 混淆页面描述语言输出以阻碍转换为可编辑格式

### 技术领域

[0001] 本发明涉及信息处理领域,更具体地,涉及用于管理电子文档的方法、用于管理电子文档的设备、以及系统。

### 背景技术

[0002] 电子文档 (ED) 描述格式通常可分为两类:标示语言 (ML) 格式和页面描述语言 (PDL) 格式。ML 格式用于文档创建和编辑,并倾向于以较高级的项描述文档的外观和布局。例如,ML 可通过指明页边距、行距、字体、字号等来描述文本段落,而将确定每个字符确切位置的细节交给渲染段落以用于显示或打印的软件或设备。相对而言,PDL 格式不用于编辑。它们用于帮助忠实、有效的文档渲染。通常地,段落的 PDL 版本将相当明确地指明每个字符在文本中的位置,但不会指明高级数据,比如页边距或行距,因为如果准确渲染是唯一目的这些就是不必要的。

[0003] 因为 PDL 数据历来被认为是不可编辑的,用户经常将文档从 ML 格式转换成 PDL 格式以作为防止修改的简略方式。例如,作者将通常以开放的办公可扩展标记语言 (OOXML) 格式(一种 ML 格式)来创建和维护文档以用于编辑。然而,作者会将文件转换成便携文档格式 (PDF),一种 PDL 格式,以用于发布。这样做的主要原因是 PDF 文档的便携性,但是在某些情况下次要原因是 PDF 格式使得接收者要恶意修改文件更加困难,比如盗取内容或改变文件并将其冒充为接收者的成果。

[0004] 最近,出现了大量允许从 PDL 格式(例如 PDF)到 ML 格式(例如 OOXML)反向转换的工具。因为从 ML 格式到 PDL 格式的转换中丢失了较高层的上下文信息,从 PDL 格式转换回 ML 格式需要推断或猜测数据,因而通常充其量也是不完善的,并且在很多情况下几乎是不可用的。然而,在某些情况下,可允许创建原始文档的摹写,这将足以避开发布者关于不可修改的格式的目的。

### 发明内容

[0005] 总体而言,一方面,本发明涉及用于管理电子文档 (ED) 的方法。所述方法包括:接收用以生成用于 ED 的混淆页面描述语言 (PDL) 文件的请求;在 ED 中识别包括多个字符的第一文本流;计算所述多个字符在页面上的多个位置;响应于所述请求通过对第一文本流施加混淆技术来生成修改的文本流;生成包括多个位置和已修改的文本流的混淆 PDL 文件。

[0006] 总体而言,一方面,本发明涉及一种用于管理电子文档 (ED) 的设备。所述设备包括:显示部件,用于向用户显示图形用户界面 (GUI),该图形用户界面包括用于生成用于所述 ED 的混淆页面描述语言 (PDL) 文件的选项;接收部件,用于接收用于生成所述 ED 的混淆 PDL 文件的请求;识别部件,用于在 ED 内识别包括多个字符的第一文本流;计算部件,用于计算所述多个字符在页面上的多个位置;第一生成部件,用于响应于所述请求通过对第一文本流施加混淆技术来生成修改的文本流;以及第二生成部件,用于生成包括所述多个位

置和所述已修改的文本流的混淆 PDL 文件。

[0007] 总体而言,一方面,本发明涉及系统。所述系统包括:计算机处理器;缓冲器,被配置成存储包括第一文本流的电子文档,该第一文本流包括多个字符;位置引擎,其在计算机处理器上运行并配置成计算多个字符在页面上的多个位置;混淆引擎,其在计算机处理器上运行并配置成通过对第一文本流施加混淆技术来生成修改的文本流;以及页面描述语言(PDL)引擎,其在计算机处理器上运行并配置成生成用于 ED 的包括多个位置和已修改的文本流的混淆 PDL 文件。

[0008] 本发明的其他方面将从下列描述和所附权利要求中显现。

## 附图说明

[0009] 图 1 示出了依照本发明的一个或多个实施例的系统。

[0010] 图 2 示出了依照本发明的一个或多个实施例的流程图。

[0011] 图 3A 和图 3B 示出了依照本发明的一个或多个实施例的示例。

[0012] 图 4 示出了依照本发明的一个或多个实施例的计算机系统。

## 具体实施方式

[0013] 现在将参照附图来详细描述本发明的具体实施例。为保持一致性,不同图中的相同元件用相同的参考符号来表示。

[0014] 在本发明实施例的下列详细描述中,详尽阐述了许多具体细节以提供对本发明更为深入全面的理解。然而,对于本领域普通技术人员而言显而易见的是,没有这些具体细节本发明也可实行。在其他情况下,为了避免不必要地使描述复杂化,众所周知的特征没有详细描述。

[0015] 总体而言,本发明的实施例提供用于管理包括一个或多个文本流的 ED 的系统和方法。所述 ED 可以是开放的办公可扩展标记语言(OOXML 格式或任何其他 ML 格式。作为接收到生成用于 ED 的混淆 PDL 文件的用户请求的响应,计算文本流的字符的位置(例如坐标)。然后,对 PDL 数据(例如文本流、剪贴画、图像、形状等)应用一项或多项混淆技术以生成修改的 PDL 数据。例如,将混淆技术应用于文本流以生成修改的文本流。混淆 PDL 文件包括已修改的文本流和计算出的位置。混淆 PDL 文件还可包括 ED 中的任意矢量图的光栅表示。混淆 PDL 文件可以是 PDF 或任何其他 PDL 格式。和标准 PDL 文件一样,混淆 PDL 文件帮助 ED 的忠实渲染。然而,在应对被设计用来将 PDL 文件转换回原始 ML 格式(例如 OOXML)或任何其他可编辑/可修改格式的工具上,混淆 PDL 文件比标准 PDL 文件更加复原力。换句话说,任何这类工具对混淆 PDL 文件的操作的输出将和所述 ED 有很小类似,减少了将所述输出作为原件忠实且轻易可修改的复制品的功用。

[0016] 图 1 示出了依照本发明一个或多个实施例的系统(100)。如图 1 所示,系统(100)具有多个组件,包括缓冲器(114)、图形用户界面(116)、位置引擎(118)、混淆引擎(120)和 PDL 引擎(122)。每个组件(114、116、118、120、122)可位于相同的硬件设备(例如,个人计算机(PC)、桌面计算机、主机、服务器、电话机、自助服务机、电缆箱、个人数字助理(PDA)、电子阅读器、智能电话机、平板计算机等)或使用具有有线和/或无线网段的网络所连接的不同硬件设备上。在本发明的一个或多个实施例中,系统(100)输入 ED(106),并输出用于

ED(106) 的混淆 PDL 文件 (110)。系统 (100) 还可输出用于 ED(106) 的标准 PDL 文件 (108)。

[0017] 在本发明的一个或多个实施例中,ED(106) 包括一个或多个文本流。每个文本流可具有任意数量的字符,因而可具有任意数量的单词。文本流可对应于句子、段落、文本列、注脚、图片说明、尾注、章节、篇章等。每页可有多个文本流。文本流可跨越多个页面。ED(106) 还可包括要在一页或多页上显示或跨越一页或多页显示的图形特征(例如照片、矢量图、剪贴画、形状等)。两个或多个图形特征可能部分重叠。使用 ML 格式(例如,开放文档格式(ODF)、OOXML 等)来表示/定义 ED(106)。相应地,文本流、图形特征以及文本流的属性和图形特征可作为 ML 格式标签中的属性来记录/识别。要正确地渲染(例如,显示、打印) ED(106),文本流、图形特征和属性是必要的。

[0018] 如上所述,ED(106) 是可编辑/可修改的。而且,ED(106) 可通过用户应用来创建和/或修改,所述用户应用例如包括字处理应用、电子制表应用、桌面发布应用、图形应用、照片打印应用、网络浏览器、幻灯片生成应用、表格生成器等。

[0019] 在本发明的一个或多个实施例中,标准 PDL 文件 (108) 是 PDL 格式(例如,PDF、XPS 等)的 ED(106)。标准 PDL 文件 (108) 帮助 ED(106) 的忠实渲染。相应地,和 ED(106) 一样,标准 PDL 文件 (108) 包括文本流和图形特征。然而,不同于 ED(106),标准 PDL 文件 (108) 包括每个文本流的每个字符和每个图形特征的明确位置(例如,x、y 坐标,偏移等)。而且,不同于 ED(106),标准 PDL 文件 (108) 不容易被修改。

[0020] 在本发明的一个或多个实施例中,混淆 PDL 文件 (110) 是 PDL 格式(例如 PDF、XPS 等)的 ED(106)。和标准 PDL 文件 (108) 一样,混淆 PDL 文件 (110) 帮助 ED(106) 的忠实渲染并包括明确的位置。换句话说,通过渲染(例如打印、显示)标准 PDL 文件 (108) 或混淆 PDL 文件 (110) 将生成基本相同的输出。然而,不同于标准 PDL 文件 (108),混淆 PDL 文件 (110) 包括一个或多个文本流或其他数据(下面将讨论)的已修改版本。而且,不同于标准 PDL 文件 (108),混淆 PDL 文件 (110) 可包括 ED(106)(下面将讨论)中的任意图形特征(例如,向量图形等)的光栅表示。和标准 PDL 文件 (108) 一样,混淆 PDL 文件 (110) 也不容易被修改。

[0021] 掌握这些详细描述的本领域技术人员将理解存在将 PDL 格式的文件转换为 ML 格式的工具,从而使文件可编辑。因为至少有文本流的已修改版本和图形特征的光栅表示,所以混淆 PDL 文件 (110) 能比标准 PDL 文件 (108) 更有复原力地应对这些工具。换句话说,任何这类工具对混淆 PDL 文件 (110) 操作的输出将和所述 ED(106) 有很小类似,使得很难对混淆 PDL 文件进行有用的修改。

[0022] 在本发明的一个或多个实施例中,系统 (100) 包括 GUI(116)。GUI(116) 可从用于生成或修改 ED(106) 的用户应用(未示出)中调用。具体地,GUI(116) 可在将 ED(106) 从 ML 格式转换成 PDL 格式的请求之后调用。GUI(116) 可具有任意数量的窗口部件(例如单选框、复选框、下拉列表、按钮等)。通过操作一个或多个窗口部件,用户可指明是否要基于 ED(106) 生成标准 PDL 文件 (108) 和/或混淆 PDL 文件 (110)。

[0023] 在本发明的一个或多个实施例中,系统 (100) 包括缓冲器 (114)。缓冲器 (114) 可对应于任意类型的存储器或长期存储器(例如硬盘)。缓冲器 (114) 被配置成在生成标准 PDL 文件 (108) 和/或混淆 PDL 文件 (110) 的请求之后存储 ED(106)。

[0024] 在本发明的一个或多个实施例中,系统 (100) 包括位置引擎 (118)。位置引擎 (118) 被配置成计算 ED(106) 中的每个文本流的每个字符的位置。位置引擎 (118) 还被配

置成计算 ED(106) 中的每个图形特征的位置。在一个或多个实施例中,每个位置被指定为页面上的坐标对(例如,x 分量、y 分量)。在一个或多个实施例中,每个位置被指定为对于参考坐标对的偏移。

[0025] 在本发明的一个或多个实施例中,系统(100)包括混淆引擎(120)。混淆引擎(120)被配置成通过对每个文本流或其他内容应用一项或多项混淆技术来生成文本流的修改版本。有很多可应用于文本流或其他内容的可能混淆技术。

[0026] 在本发明的一个或多个实施例中,一项混淆技术包括扰乱文本流中的字符顺序以生成修改的文本流,使得 PDL 数据中的文本顺序不同于 ML 数据中的文本顺序。例如,文本流内的随机字符可交换位置。作为另一例子,文本流内的各个单词可反向。作为又一例子,文本流的整个顺序可反向(即最后一个字符现在变成第一个而第一个字符变成最后一个)。在本发明的一个或多个实施例中,一项混淆技术包括从文本流中删除一个或多个字符,并将它们添加到不同的文本流以生成修改的文本流。

[0027] 掌握这些详细描述的本领域技术人员将理解,扰乱文本流中的字符顺序和/或从文本流删除一个或多个字符并把将它们添加到不同文本流,这并不改变所计算的字符位置。然而,它会改变 PDL 数据(例如,已修改的文本流)中的字符的位置。具体地,它使得 PDL 数据中的字符顺序与显示在屏幕或硬拷贝上的字符顺序不相关。这样做的目的是迫使反向转换工具(即 PDL 到 ML 的转换工具)仅根据它们在渲染页面上的几何结构而不是 PDL 数据的结构来尽量多地解析字符之间的关系(比如它们在文本流中的顺序,或将文档中的字符恰当地划分成一组文本流),从计算机程序的观点看 PDL 数据的结构一般更简单。

[0028] 在本发明的一个或多个实施例中,一项混淆技术包括将文本流划分成多个 PDL 组(例如,PDF 组、XPS 组等)以生成修改的文本流。例如,可将文本流的每第二个字符放置于第一 PDL 组,而可以将文本流的其余字符放置于第二 PDL 组。换句话说,有意将内容的外部分组引入 PDL 数据,而隐藏原始 ML 数据中可能已存在的任何分组。这样做的意图是误导依赖 PDL 数据中这种分组结构来推断高层信息(比如将文本内容恰当地划分为文本流)的反向转换工具(即 PDL 转换成 ML 的工具)。该混淆技术可与任何其他混淆技术结合使用。

[0029] 在本发明的一个或多个实施例中,一项混淆技术包括使用功能相同而句法不同的结构来表示在 ML 数据中相关联的对象,以掩盖它们之间的关联。例如,假设存在文本流,其中的字符都应该涂成黑色。可通过如下操作创建修改的文本流:将所述字符一个子集的颜色空间设置为 RGB 且颜色值设为(0,0,0)并将其余字符的颜色空间设置为灰色(Gray)且颜色值设为(0)。这将不会影响渲染的输出(即 RGB(0,0,0)和灰色(0)在屏幕和硬拷贝上都是黑色的),但是有可能使得简单化的反向转换工具(即 PDL 到 ML 的转换工具)因为不同的颜色空间而相信这些字符不属于同一文本流。相同的技术可应用于非文本数据,比如路径填充或路径绘制。

[0030] 在本发明的一个或多个实施例中,混淆引擎(120)还被配置为操作 ED(106)中的图形特征。例如,混淆引擎(120)可生成 ED 中的矢量图的光栅表示。作为另一例子,混淆引擎(120)可生成多个重叠的图形特征的单个(即合成)光栅表示。一般地,PDL 到 ML 的转换工具从光栅表示中分析并提取高层信息比从矢量图中更加困难。

[0031] 在本发明的一个或多个实施例中,混淆引擎(120)被配置成故意使用复杂的 PDL

特有结构来表示数据。例如,假设 ED(106) 包括要被涂成蓝色的矩形,且要创建的 PDL 格式是 PDF。PDF 表示可以并不是简单地将颜色设为蓝色,而是创建带有张量补丁渐变填充的阴影颜色空间,当估值时其导致恒定的蓝色。因为张量补丁阴影不是标准 ML 格式的特征,且确定张量补丁公式导致固定的颜色有些困难,所以很可能 PDL 到 ML 的转换工具将不能以 ML 格式重建所述矩形的原始简单表示。

[0032] 掌握这些详细描述的本领域技术人员将理解混淆引擎 (120) 仅用于生成混淆 PDL 文件 (110) 而不是标准 PDL 文件 (108)。掌握这些详细描述的本领域技术人员还将理解,因为需要生成修改的文本流、光栅表示等,所以生成混淆 PDL 文件 (110) 将比生成标准 PDL 文件 (108) 需要更长的时间。类似地,渲染混淆 PDL 文件会比混淆标准 PDL 文件花费更长的时间。

[0033] 在本发明的一个或多个实施例中,系统 (100) 包括 PDL 引擎 (122)。PDL 引擎 (122) 被配置成生成标准 PDL 文件 (108) 和混淆 PDL 文件 (110) 二者。标准 PDL 文件 (108) 和混淆 PDL 文件 (110) 二者都包括由位置引擎 (118) 计算的位置。然而,混淆 PDL 文件 (110) 包括已修改的文本流、光栅表示和混淆引擎 (120) 的任何其他创建 (例如张量补丁渐变填充)。

[0034] 尽管图 1 示出具有特定数量和排列的组件 (114、116、118、120、122) 的系统 (100), 掌握这些详细描述的本领域技术人员将理解其他系统配置也是可能的。

[0035] 图 2 示出了依照本发明的一个或多个实施例的流程图。图 2 所示的过程例如可由以上参照图 1 所讨论的一个或多个组件 (例如,位置引擎 (118)、混淆引擎 (120)、PDL 引擎 (122)) 来执行。在多一个组件被配置成软件模块的情况下,计算机程序代码存储在系统 (100) 的存储器中,所述过程由读取程序代码并执行代码的处理器来实施。图 2 示出的一个或多个步骤可被省略、重复和 / 或在本发明的不同实施例中以不同的顺序执行。相应地,本发明的实施例不应被认为是限于图 2 中所示步骤的特定数量和排列。

[0036] 首先,显示 (步骤 202) 具有用于生成混淆 PDL 文件的选项的 GUI。所述 GUI 可作为对生成将 ML 格式的 ED 转换到 PDL 格式的用户请求的响应而显示。所述 GUI 可具有多个窗口部件,包括单选框、复选框、下拉框、按钮等。用户可操作一个或多个窗口部件来调用选项,包括生成混淆 PDL 文件而不是标准 PDL 文件的选项。

[0037] 在步骤 205 中,接收到生成混淆 PDL 文件的请求。换句话说,用户已指定要为 ED 生成混淆 PDL 文件 (不是标准非混淆文件)。所述请求还可指定 PDL 文件的类型 (例如,PDF、XPS 等)。

[0038] 在步骤 210 中,选择 ED 内的文本流。ED 的文本流可通过解析 ED 来识别 (例如,当 ED 存储在缓冲器 (114) 中时)。在解析期间,可以在文本流出现时选择该文本流。如上所述,每个文本流可含有任意数量的字符,因而可含有任意数量的单词。文本流可对应于句子、段落、文本列、注脚、图片说明、尾注、章节、篇章等。每页可有多个文本流。文本流可跨越多个页面。

[0039] 在步骤 215 中,计算文本流中的每个字符的位置。位置可包括用于每个字符的坐标对 (例如, x 分量、y 分量)。另外或可替代地,位置可包括相对于参考坐标对的偏移。

[0040] 在步骤 220 中,通过对文本流应用一项或多项混淆技术来生成修改的文本流。如上所述,可能的混淆技术包括扰乱文本流中的字符的顺序,从文本流中删除字符并将所述



字符添加到另一文本流,将相同文本流中的不同字符设置为不同的颜色空间等。

[0041] 在步骤 225 中,确定 ED 中是否存在另外的文本流。当确定存在另外的文本流时,过程回到步骤 210。否则,当确定不存在另外的文本流时,过程继续到步骤 230。

[0042] 在步骤 230 中,生成 ED 中的图形特征(例如,矢量图)的光栅表示。如果两个或多个图形特征重叠,可为重叠的图形特征生成单个(即合成)光栅表示。如果 ED 中没有出现图形特征,步骤 230 可被省略。

[0043] 在步骤 235 中,为 ED 中具备填充颜色的任何形状创建具有张量补丁渐变填充的阴影颜色空间。如果 ED 中没有形状和/或如果生成的 PDL 文件类型不是 PDF,步骤 235 可被省略。如上所述,张量补丁渐变填充阴影是 PDF 特有的特征而不是 ML 格式的标准特征。而且,任何 PDL 到 ML 的转换工具将能够估算张量补丁渐变填充并确定它实际对应于简单填充颜色,这是极小可能的。

[0044] 在步骤 240 中,生成混淆 PDL 文件,其具有已修改的文本流、字符的计算位置、光栅表示和阴影颜色空间。混淆 PDL 文件可分发至任意数量的用户。因为至少有文本流的修改版本和图形特征的光栅表示,混淆 PDL 文件比标准 PDL 文件更加有复原力地应对 PDL 到 ML 的转换工具。换句话说,任意这类工具对混淆 PDL 文件操作的输出与 ED 有很小类似,这防止了所述混淆 PDL 文件变成可修改的。

[0045] 尽管在上面提及的示例性实施例中,至少一项混淆技术被应用于每个文本流,在本发明的其他实施例中,该技术仅可被应用于某些(即非全部)文本流或用户提前选择的文本流。例如,在步骤 202 中,ED 的预览可显示在 GUI 上,用户可选择至少一个他/她想混淆的文本流。在这种情况下,仅为在步骤 220 中选择的文本流生成修改的文本流。

[0046] 图 3A 和图 3B 示出了依照本发明一个或多个实施例的示例。在图 3A 中,存在 ED(302)。ED(302)可对应于以上参照图 1 讨论的 ED(106)。ED(302)是 OOXML 格式,因此是可编辑的。所述 ED 包括多个文本流:文本流 A(312A)和文本流 B(312B)。每个文本流(312A、312B)具有多个单词,因而有多个字符。所述 ED 还包括两个矢量图:矢量图 A(314A)和矢量图 B(314B)。

[0047] 图 3A 还示出渲染的 ED(304)。换句话说,渲染的 ED(304)是当 ED(302)被显示或打印时的输出。如图 3A 所示,文本流 A(312A)近乎跨越渲染的 ED(304)的页面宽度,而文本流 B(312B)排列成渲染的 ED(304)的列。而且,两个矢量图(314A、314B)在渲染的 ED(304)中重叠(即星星位于大象的顶上)。

[0048] 图 3B 示出标准 PDL 文件(306)和混淆 PDL 文件(308)。标准 PDL 文件(306)和混淆 PDL 文件(308)可对应于以上参照图 1 讨论的标准 PDL 文件(108)和混淆 PDL 文件(110)。两个 PDL 文件(306、308)都可以处于 PDF。而且,两个 PDL 文件(306、308)都可帮助 ED(302)的忠实渲染。换句话说,渲染标准 PDL 文件(306)或混淆 PDL 文件的输出与已渲染的 ED(304)基本相同。

[0049] 如图 3B 所示,标准 PDL 文件(306)包括文本流 A(312A)和文本流 B(312B)。每个文本流仅有一部分在图 3B 中再现。具体地,仅示出文本流 A(312A)中对应于“quick”的字符和文本流 B(312B)中对应于“lemon”的字符。更重要地,标准 PDL 文件(306)包括每个字符的位置。例如,文本流 A(312A)中的字符“q”具有位置(x1, y1)。作为另一例子,文本流 B(312B)中“lemon”的字符“o”具有位置(x9, y9)。而且,标准 PDL 文件(306)包括矢量

图 A(314A) 和矢量图 B(314B) 二者的位置。

[0050] 图 3B 还示出混淆 PDL 文件 (308)。和标准 PDL 文件 (306) 一样,混淆 PDL 文件 (308) 也具有每个字符的位置。然而,不同于标准 PDL 文件 (306),混淆 PDL 文件 (308) 具有已修改的文本流:已修改文本流 A(322A) 和已修改文本流 B(322B)。仅示出了部分的已修改文本流。通过对 ED(302) 的文本流 B(312B) 施加混淆技术来生成修改的文本流 B(322B)。具体地,通过反转文本流 B(312B) 中的每个单词并删除“lemons”中的“m”来生成修改的文本流 B(322B)。换句话说,在反转之后“lemons”变成“snome1”,然后在删除“m”之后变成“snoel”。通过对 ED(302) 中的文本流 A(312A) 应用多种混淆技术来生成修改的文本流 A(322A)。具体地,通过如下操作来生成修改的文本流 A(322A):反转文本流 A(312A) 中的所有单词,插入来自文本流 B(312B) 的“m”,然后将文本流划分成两个 PDF 组:PDF 组 I(326) 和 PDF 组 II(328)。换句话说,反转之后“quick”变成“kciuq”,然后在插入“m”之后变成“kcmiuq”,再然后在分组之后变成“kcmi”和“uq”。混淆 PDL 文件 (308) 还包括重叠的矢量图 A(314A) 和矢量图 B(314B) 的单个合成光栅表示 (325)。

[0051] 掌握这些详细描述的好处本领域技术人员将理解,混淆 PDL 文件 (308) 能比标准 PDL 文件 (306) 更有复原力地应对将 PDL 格式转换成 ML 格式的工具。具体地,已修改的文本流 (322A、322B) 使得这样的工具格外难以准确地将字符分配给文本流并确定文本流中字符的顺序。而且,合成光栅表示 (325) 使得这样的工具格外难以(如果不是不可能)提取两个分离的矢量图像。换句话说,已修改的文本流 (322A、322B) 和合成光栅表示 (314) 确保混淆 PDL 文件 (308) 保持为不可修改。

[0052] 本发明的实施例可具有一个或多个下列优点:防止 PDL 文件变得易于被修改的能力;生成修改的文本流的能力;生成重叠矢量图的合成光栅表示的能力;生成能抵抗 PDL 到 ML 的转换工具的 PDL 文件的能力等。

[0053] 本发明的实施例可在几乎任意类型的计算系统上实现而无论其使用什么平台。例如,计算系统可以是一个或多个移动设备(例如,便携式计算机、智能电话机、个人数字助理、平板计算机或其他移动设备),桌面计算机、服务器、服务器机架中的刀片或至少包括如下部分的任意其他类型的计算设备,即包括能执行本发明的一个或多个实施例的至少最小处理能力、存储器和输入输出设备。例如,如图 4 所示,计算系统 (400) 可包括一个或多个计算机处理器 (402)、关联存储器 (404)(例如,随机存取存储器 (RAM)、高速缓存存储器、闪存等)、一个或多个储存设备 (406)(例如,硬盘、比如光盘 (CD) 驱动器或数字多功能光盘 (DVD) 驱动器的光驱、闪存条等) 和多个其他元件和功能。计算机处理器 (402) 可以是用于处理指令的集成电路。例如,计算机处理器可以是一个或多个内核、或处理器的微内核。计算系统 (400) 还可包括一个或多个输入设备 (410),比如触屏、键盘、鼠标、话筒、触控板、电子笔或任何其他类型的输入设备。此外,计算系统 (400) 还可包括一个或多个输出设备 (408),比如屏幕(例如,液晶显示器 (LCD)、等离子显示器、触屏、阴极射线管显示器 (CRT)、投影仪或其他显示设备)、打印机、外部存储器或任何其他输出设备。一个或多个输出设备可以和输入设备相同或不同。计算系统 (400) 可经由网络接口连接(未示出)连接到网络 (412)(例如,本地局域网 (LAN)、比如互联网的广域网 (WAN)、移动网络或任何其他类型的网络)。输入和输出设备可以本地或者远程(例如,经由网络 (412))连接到计算机处理器 (402)、存储器 (404) 和储存设备 (406)。存在很多不同类型的计算系统,并且上述输入和输

出设备也可采用其他形式。

[0054] 用于实现本发明实施例的处于计算机可读程序代码形式的软件指令可完整或部分地、暂时或永久地存储在非暂时性计算机可读的介质上,比如 CD、DVD、储存设备、磁盘、磁带、闪存、物理存储器或任何其他计算机可读的储存介质。具体地,所述软件指令可对应于计算可读的程序代码,所述程序代码当被处理器执行时,被配置成实现本发明的实施例。

[0055] 此外,上述计算系统(400)的一个或多个元件可位于远程位置并与网络上的其他元件相连接。此外,本发明的实施例可在具有多个节点的分布式系统上实现,其中本发明的每部分可位于分布式系统内的不同节点上。在本发明的一个实施例中,节点对应于不同的计算设备。可替代地,节点可对应于具有关联物理存储器的计算机处理器。替换地,节点可对应于具有共享存储器和/或资源的计算机处理器或计算机处理器的微内核。

[0056] 虽然仅结合有限数量的实施例对本发明进行了描述,掌握此公开的益处的本领域技术人员将理解,可设计在本文所揭示的发明范围之内的其他实施例。相应地,本发明的范围应仅受所附权利要求的限制。

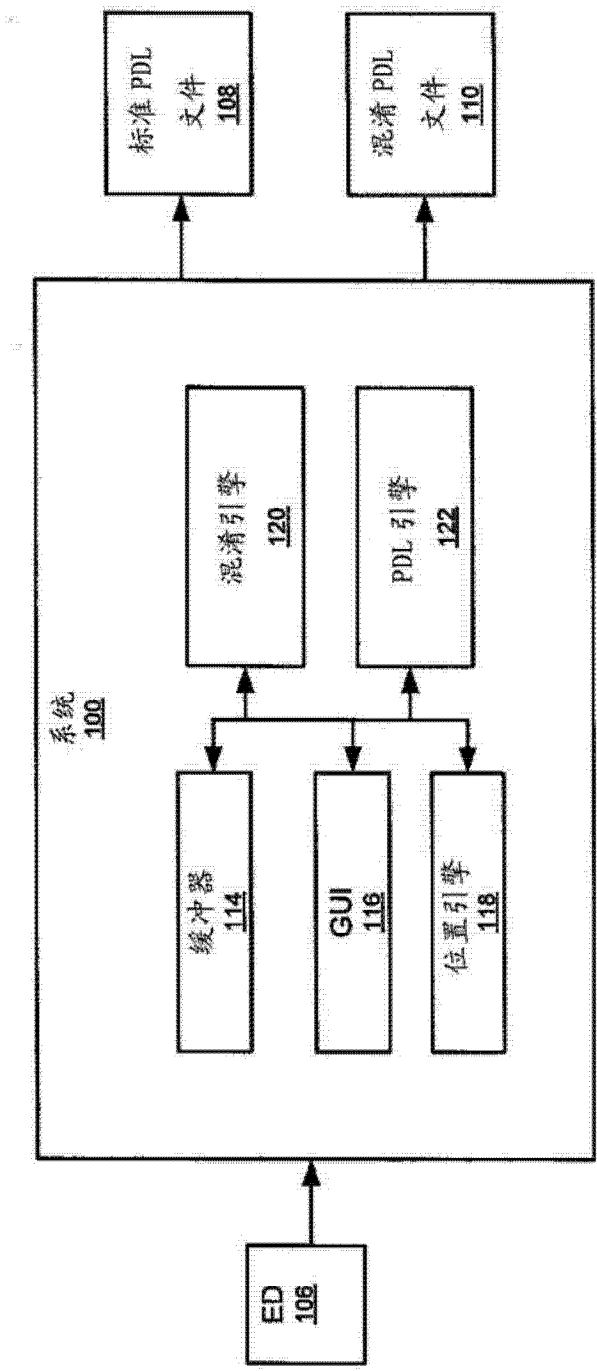


图 1

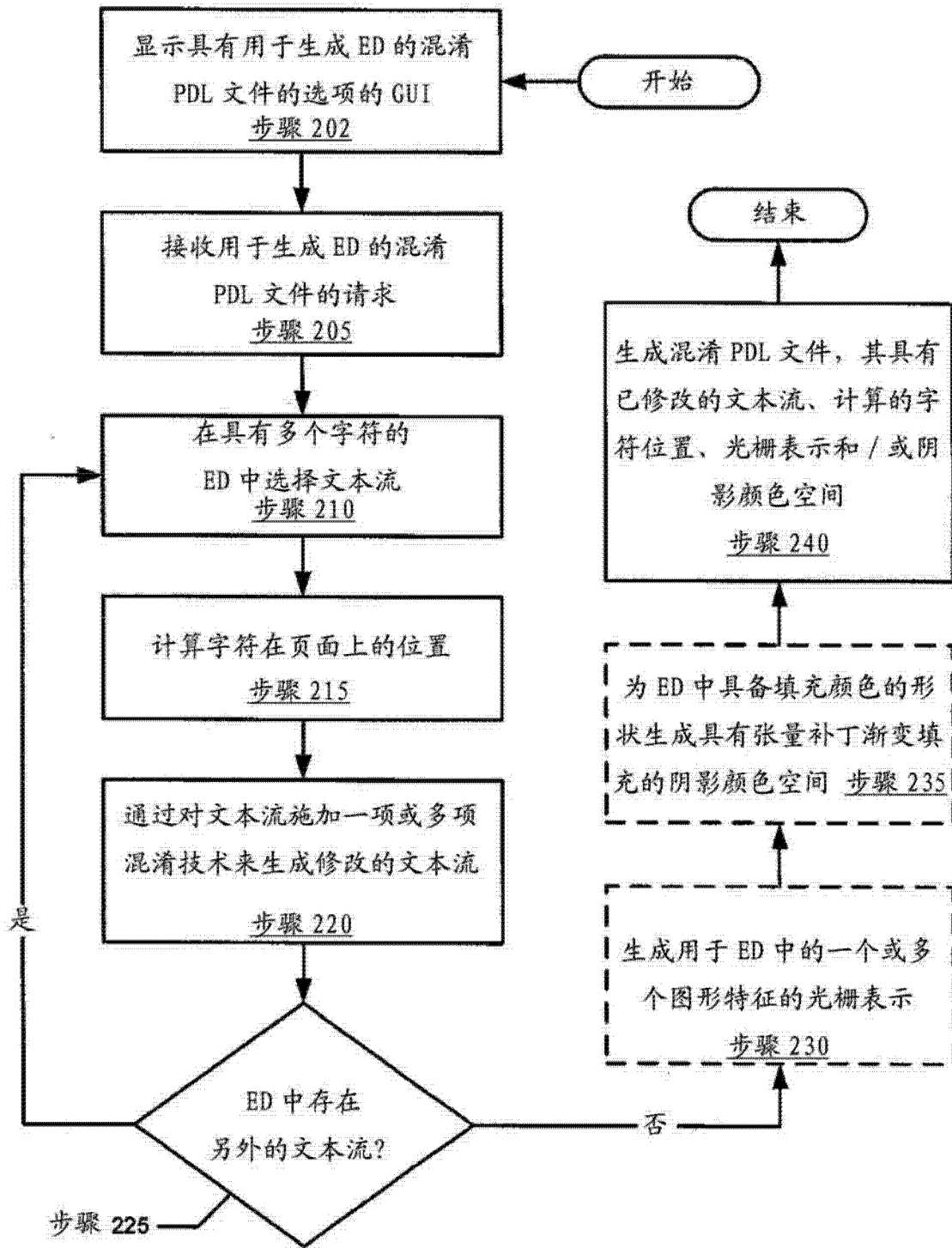


图 2

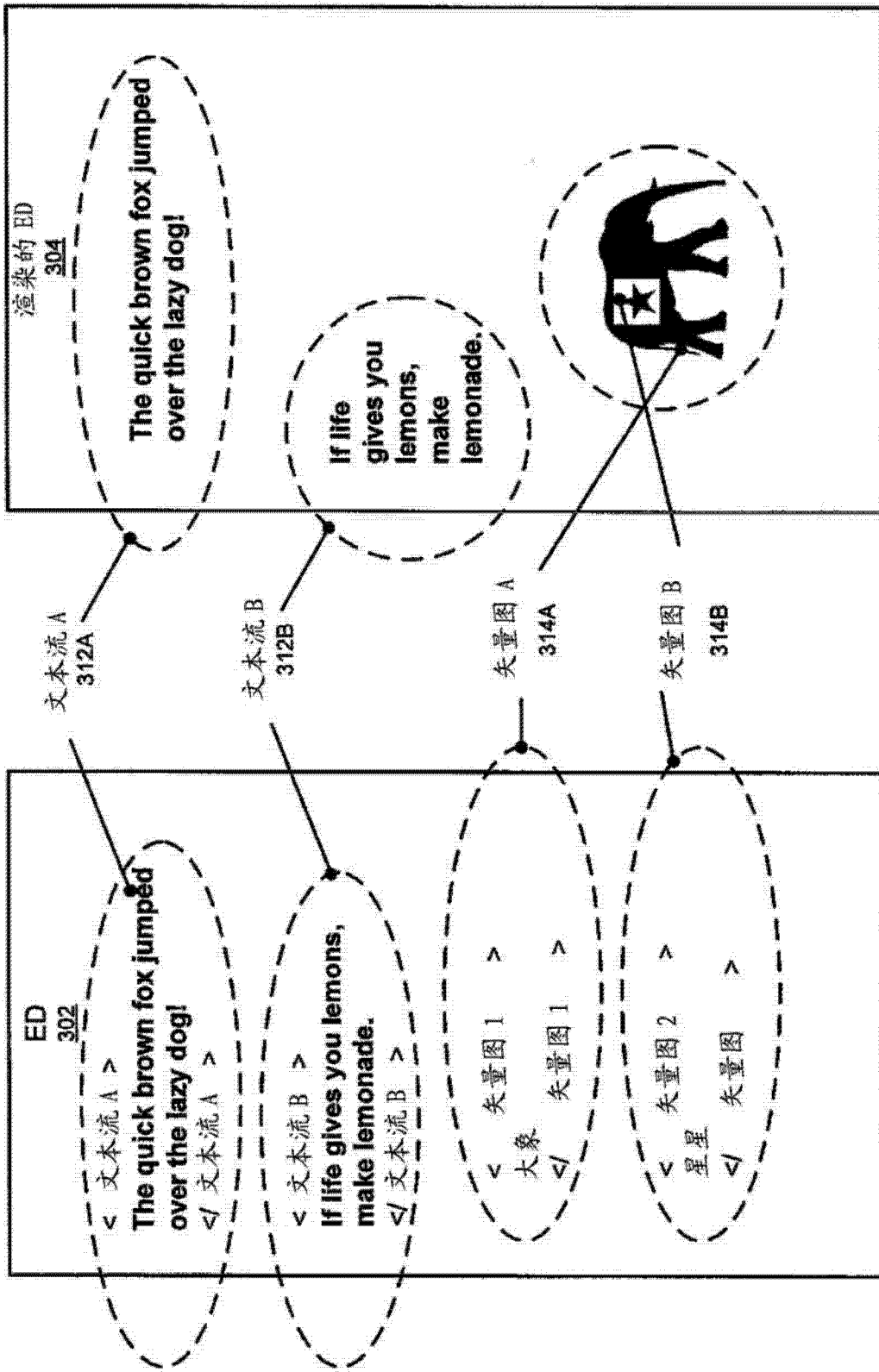


图 3A

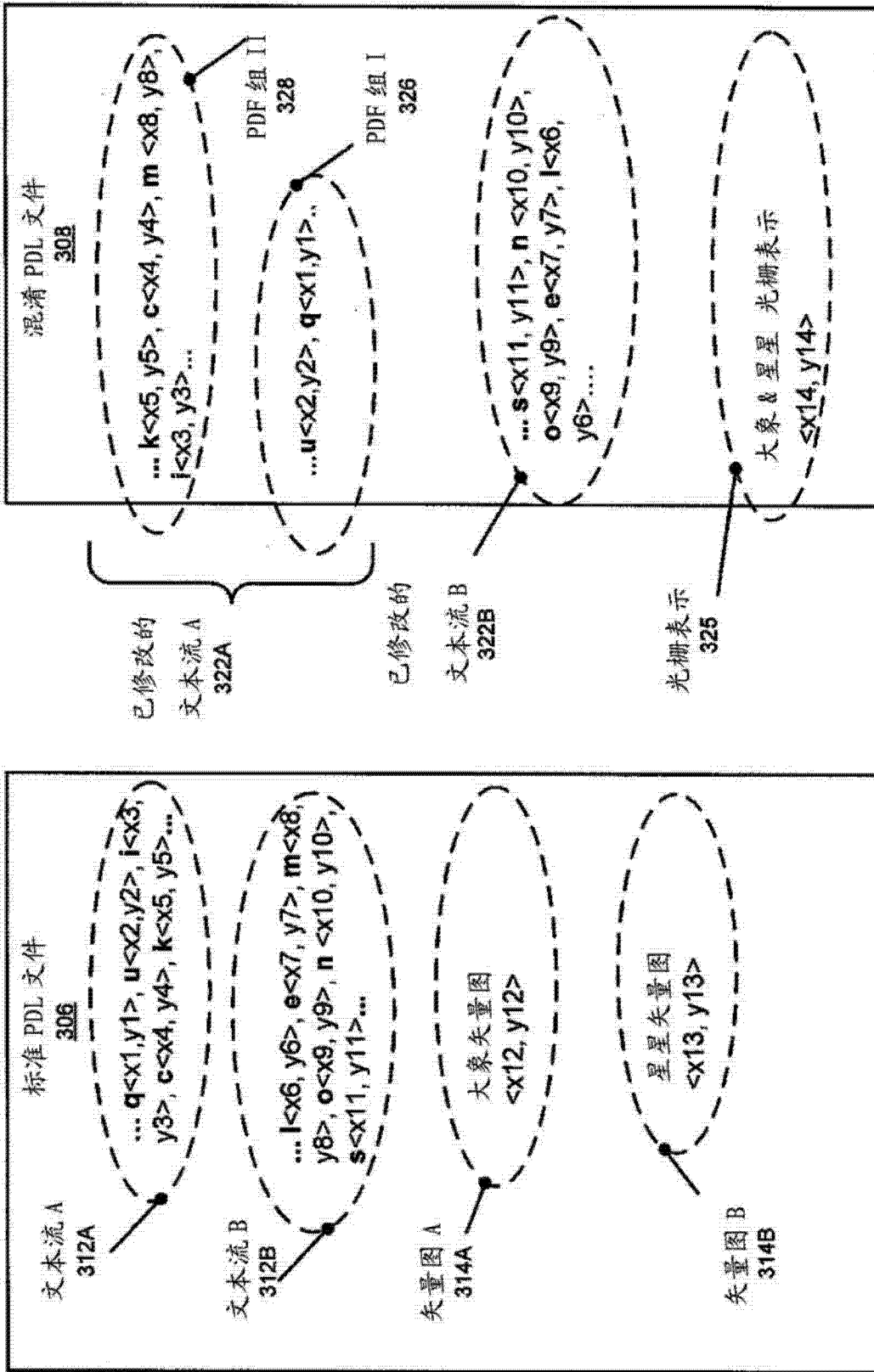


图 3B

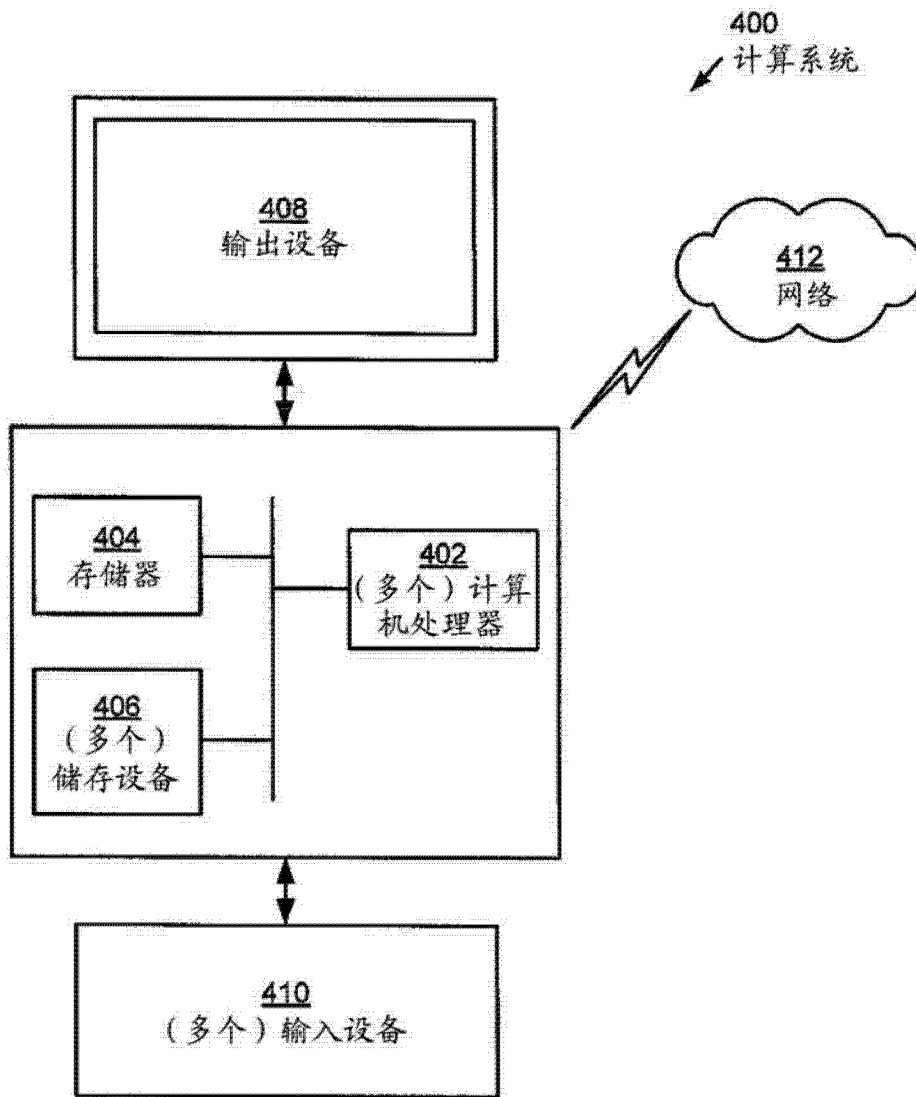


图 4