



[12] 发明专利申请公开说明书

[21] 申请号 01807955.5

[43] 公开日 2003 年 8 月 6 日

[11] 公开号 CN 1434960A

[22] 申请日 2001.4.17 [21] 申请号 01807955.5
 [30] 优先权
 [32] 2000. 4. 14 [33] GB [31] 0009129. 8
 [32] 2000. 10. 31 [33] US [31] 09/703,502
 [86] 国际申请 PCT/GB01/01725 2001. 4. 17
 [87] 国际公布 WO01/80044 英 2001. 10. 25
 [85] 进入国家阶段日期 2002. 10. 11
 [71] 申请人 皮克塞(研究)有限公司
 地址 英国格拉斯哥
 [72] 发明人 马希德·安瓦尔

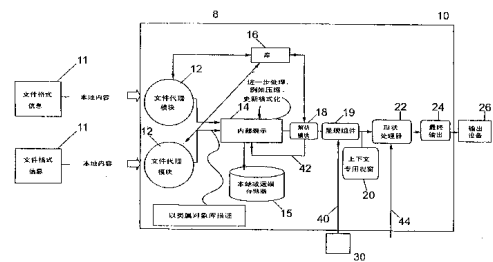
[74] 专利代理机构 永新专利商标代理有限公司
 代理人 夏青

权利要求书 6 页 说明书 26 页 附图 7 页

[54] 发明名称 用于数字文件处理的系统和方法

[57] 摘要

显示技术，将应用程序的底层的功能性与图形显示处理分离，从而消除或减少应用程序的需要以控制设备显示以及提供用于显示的图形用户界面工具与控制。另外，这样的系统减少或消除了在处理系统中当显示由应用程序创建或为应用程序创建的数据(例如文件或视频信息流)时、对提供该应用程序的需要。因而可以理解的是，在一方面中，这些系统和方法可以显示内容，包括文件、视频流或其他的内容，并且将提供用于查看所显示文件的图形用户功能，例如图像缩放、全景展开或其他此类功能，而无需在显示该内容的系统上提供底层的应用程序。这些系统和方法优越于现有技术的有益效果包括允许在同样的工作空间内在同一显示器上显示来自不同应用程序的不同类型的内容。



1. 一种数字文件处理系统，包括
 - 一应用程序调度模块，用于接收以若干预定数据格式之一表示源数据的输入字节流，并用于将所述输入字节流与所述若干预定数据格式之一相关联，
 - 一文件代理模块，用于将所述输入字节流译为所述相关联的预定数据格式的函数，并用于将所述输入字节流解析为一连串的文件对象，这些文件对象表示所述输入字节流内基本结构的内部表示，以及
 - 一个核心文件引擎，用于将所述文件对象转换为内部表示数据格式，并用于将所述内部表示数据映射到显示器上的位置。
2. 如权利要求1所述的数字文件系统，进一步包括：
 - 一形状处理器，用于处理所述内部表示数据以驱动输出设备。
3. 如权利要求1或2所述的数字文件系统，其中，所述源数据定义数字文件的内容和结构，其中所述内部表示数据用若干数据类型和参数的文件对象、独立地根据所述内容说明所述结构，所述参数定义所述文件对象的特定实例的性质。
4. 如权利要求3所述的数字文件处理系统，其中，所述定义特定实例的性质的参数包括选自于空间性质、瞬态性质和物理性质中的性质。
5. 如权利要求3或者4所述的数字文件处理系统，进一步包括一个对象类型库，所述内部表示数据基于所述库的内容。
6. 如权利要求3至5中的任一项所述的数字文件处理系统，其中，所述核心文件引擎包括解析和呈现模块，适用于根据对所述解析和呈现模块的第一控制输入，产生所述内部表示数据的至少一部分的特定视图的、基于参数和对象的表示。

7. 如权利要求6所述的数字文件处理系统,其中,所述基于参数的表示包括选自填充、路径、边界框和透明度中的参数。

8. 如权利要求5至7中的任一项所述的数字文件处理系统,进一步包括一形状处理模块,用于接收来自所述解析和呈现模块的、所述特定视图的基于对象和参数的表示,并用于将所述基于对象和参数的表示转换为适用于驱动特定的输出设备的输出数据格式。

9. 如权利要求8所述的数字文件处理系统,其中,所述形状处理模块根据所述对象的形状处理所述对象,所述形状由所述边界框、所述对象的所述数据内容和所述对象的所述透明度界定。

10. 如权利要求8或9所述的数字文件处理系统,其中,所述形状处理模块根据定义所述对象的形状处理所述对象,所述对象的所述形状由边界框界定,该边界框表示在可以其上呈现对象的显示器上的已定义区域。

11. 根据前面任一权利要求所述的数字文件处理系统,其中,所述系统采用基于色度/亮度的色彩模式来描述色彩数据。

12. 根据前面任一权利要求所述的数字文件处理系统,其中,所述系统采用通用文本编码模型。

13. 如权利要求12所述的数字文件处理系统,其中,通用的文本编码包括unicode、shift-mapping和big-5。

14. 根据前面任一权利要求所述的数字文件处理系统,进一步包括一个处理,用于通过组合具有类似属性的文件对象来压缩源文件的内部表示。

15. 根据前面任一权利要求所述的数字文件处理系统,进一步包括一个处理,用于通过组合具有类似样式属性的文件对象压缩源文件的内部表示。

16. 根据前面任一权利要求所述的数字文件处理系统,其中,所述系统适合于多重并行执行过程,用于处理来自一个或多个数据

源的源数据并产生一组或多组输出表示数据。

17. 根据前面任一权利要求所述的数字文件处理系统，进一步包括一图形用户接口，用于产生交互直观显示的内部表示，由一用户利用其来控制所述数字文件处理系统。

18. 如权利要求17所述的数字文件处理系统，包括一数据处理设备，其中合并了图形用户接口。

19. 根据前面任一权利要求所述的数字文件处理系统，具有适用于被嵌入到一设备里的平台，所述设备是从由便携式计算机、移动电话、机顶盒、传真机、复印机、嵌入式计算机系统、打印机、汽车内系统以及计算机工作站组成的组中被选择出来的。

20. 根据前面任一权利要求所述的数字文件处理系统，具有包括核心处理器系统的处理器。

21. 如权利要求20所述的数字文件处理系统，其中，所述核心处理器是RISC处理器。

22. 根据前面任一权利要求所述的数字文件处理系统，其中，所述文件代理模块包括用于以选定格式输出数据的输出处理过程。

23. 根据前面任一权利要求所述的数字文件处理系统，适用于在多处理系统上进行工作。

24. 一种用于显示内容的方法，包括：

接收表示数字内容的数据源，该数字内容具有结构和数据内容，
处理所述数据源，以识别与之相关联的文件格式，

将所述数据源作为其所识别的文件格式的函数、译为内部表示，
该内部表示包括一个第一数据结构，用于存贮关于所述数字内容的结构的信息，
以及一个第二数据结构，用于存贮关于所述数字内容中包含的数据内容的信息，

通过处理第一数据结构以确定用于所述内容文件的一部分的结构，
以及通过处理第二数据结构以确定用于所述内容文件的相应部

分的数据内容，产生内容文件，它表示将要呈现给用户的内容的内部表示。

25. 如权利要求24所述的方法，其中，接收数据源包括接收来自一数据源的一连串输入数据。

26. 如权利要求25所述的方法，其中，所述数据源是从由数据文件、由外围设备产生的字节流以及由数据文件产生的字节流组成的组中选出来的。

27. 如权利要求25或者26所述的方法，其中，处理所述数据源包括将关于所述数据源的信息提供给若干文件代理模块，每一文件代理模块能够将已知文件格式的数据源译为内部表示。

28. 如权利要求24至27中任一项所述的方法，其中，将所述数据源译为内部表示包括处理所述数据源以识别出其中的数据，并将识别的数据映射至一组对象类型，该组对象类型表示出现在数据源中的内容之类型。

29. 如权利要求28所述的方法，其中，映射包括将被识别的数据映射至一组适用于翻译源数据的对象类型，该源数据表示从由数字文件、音频/视频表示、音乐文件、交互脚本、用户界面文件和图像文件组成的组中被选择出来的内容。

30. 如权利要求24至29中任一权利要求所述的方法，其中，映射包括将被识别的数据映射至一组对象类型，该组对象类型包括位图对象类型、矢量图形对象类型、视频类型、动画类型、按钮类型、手写体类型和文本对象类型。

31. 如权利要求24至30中任一权利要求所述的方法，其中，翻译所述数据源包括滤除部分所述源数据，以便创建所述源文件的过滤后的内部表示。

32. 如权利要求24至31中任一权利要求所述的方法，其中，翻译所述数据源包括改变所述第一数据结构以调整所述数字内容的结

构。

33. 如权利要求24至32中任一权利要求所述的方法，其中，翻译所述数据源包括进一步的步骤：用第二数据结构替代数据内容，从而修改在所述内部表示内出现的内容。

34. 如权利要求24至33中任一权利要求所述的方法，其中，翻译所述数据源包括将所述数据源翻译为一组已知对象类型的文件对象，其中的文件对象包括一组定义所述文件对象的空间特性、瞬态特性和物理特性的参数。

35. 如权利要求24至34中任一权利要求所述的方法，其中，所述处理适用于在多处理器上运行。

36. 如权利要求24至35中任一权利要求所述的方法，其中，所述处理提供文本编码处理，用于以从由unicode、shift-mapping和big-5组成的组中被选择出来的一种格式进行编码。

37. 如权利要求24至36中任一权利要求所述的方法，其中，产生内容数据文件包括对一组具有相关联的参数的文件对象进行解析，以定义所述内容数据文件的结构和内容。

38. 如权利要求37所述的方法，进一步包括处理所述内容数据文件的结构和内容，以创建一组对象，该组对象定义所述内容数据文件并能够被呈现在输出设备上。

39. 如权利要求37或者38所述的方法，其中，处理所述文件对象包括处理所述相关联的参数，用于使内容流入由所述文件对象定义的结构里。

40. 如权利要求38或者依赖于权利要求38的权利要求39所述的方法，其中，所述输出设备包括显示器，该显示器选自于由视觉显示器、音频扬声器、视频播放机、电视显示器、打印机、磁盘驱动器、网络以及嵌入式显示器组成的组。

41. 一种用于与数字文件中的内容交互的系统，包括：

一文件代理模块，用于将所述数字文件中的内容转换为一组表示基本结构的内部表示的文件对象，以及

一核心文件引擎，用于呈现所述文件对象，以产生表示所述数字内容的显示，

一用户接口，用于检测输入信号，该输入信号表示用于修改所述数字文件的内容的输入，以及

一个处理模块，用于改变所述内容的所述内部表示为所述输入信号的函数，以修改所述数字内容的显示。

42. 如权利要求41所述的系统，其中，所述用户接口包括从由鼠标、触摸板、触摸屏、操纵杆、遥控装置以及键盘组成的组中被选择出来的输入设备。

用于数字文件处理的系统和方法

相关申请

本申请要求以前于2000年 4月14日申请的英国专利申请号0009129.8和2000年 10月31日申请的美国专利申请序号09/703502的优先权，两者的发明人皆为 Majid Anwar，这两个申请的内容在此合并作为参考。

发明领域

本发明涉及数据处理系统，更具体来讲，涉及用于处理数字文件以产生源文件的输出表示、作为直观显示、硬备份或以其它显示格式表现的方法和系统。

背景技术

本文所用的术语“数字文件”用来描述由数据处理系统处理的任一类型数据的数字表示，它最终以某种形式完全或部分地输出给人类用户，一般是通过被显示或直观再现（例如，借助于直观显示单元或打印机），或者是通过文本-语音转换等等。数字文件可以包括任何能够表示的特征，包括但不限于：文本；图形图像；动画图形图像；全动视频图像；交互图标、按钮、菜单或超链接。数字文件还可以包括非直观单元，例如音频（声音）单元。

数据处理系统，例如个人计算机系统，一般要求处理“数字文件”，这些文件可能来源于许多本地或远端源中的任何一个，并且可能以多种数据格式（“文件格式”）中的任何一种存在。为了产生该文件的一输出版本，例如无论作为一直观显示或打印副本，都

需要该计算机系统解释该原始数据文件并产生与相关的输出装置（例如监视器或其他直观显示设备或打印机）兼容的输出。通常，该处理涉及用于解释数据文件的应用程序、计算机的操作系统、专用于该期望输出设备的软件“驱动器”以及在某些情况下（特别是对于监视器或其他直观显示装置）作为扩展卡形式的辅助硬件。

用于处理数字文件以产生输出的传统方法，就硬件资源、软件开销以及处理时间而言效率很低，而且完全不适用于包括无线远程通信系统在内的低功率、便携式数据处理系统，也不适用于低成本数据处理系统，例如网路终端等等。在传统的数字文件处理系统中还会遇到其他问题，包括需要配置多个系统组件（包括硬件和软件组件）以便以期望的方式进行交互，以及在通过不同系统（例如在格式化、色彩再现等方面不同）对同一原始资料的处理中的不一致性。此外，用于数字文件处理的传统方法不能利用文件格式组件的通用性和/或再可用性。

发明概述

本发明的目的是提供数字文件处理方法和系统，以及结合该方法和系统的设备，它们消除或减轻了上述的传统方法和系统的缺陷。

此处所述的系统和方法提供一种显示技术，将一应用程序的底层的功能性与图形显示处理分离，从而消除或减少了应用程序对控制设备显示以及为显示提供图形用户界面工具与控制的需要。此外，这样的系统减少或消除了在处理系统上、当显示由应用程序创建或为应用程序创建的数据（例如文件或视频信息流）时、对该应用程序的需要。因而可以理解的是，在一方面中，此处所述的系统和方法可以显示内容，包括文件、视频信息流或其他的内容，并且将为查看所显示文件提供图形用户功能，例如图象缩放、全景展开或其

他此类功能，而无需在显示该内容的系统上存在其底层的应用程序。此处所述的系统和方法优越于现有技术的有益效果包括允许在同样的工作空间内、在同一显示器上、显示来自不同的应用程序的不同类型的内容。很多的有益效果对于本领域中普通技术人员是显而易见的，而且本领域中普通技术人员还能够领会使用本发明的底层技术的很多方式，用以创建另外的系统、设备和应用程序。这些修改的系统 and 替代的系统与实际应用应该被理解为在本发明的范围之内。

更具体来讲，此处所述的系统和方法包括一数字内容处理系统，它包括一个应用程序调度模块（dispatcher），用于接收以若干预先确定的数据格式之一表示源数据的输入字节流，并用于将该输入字节流与该预先确定数据格式之一相关联。该系统还可以包括一文件代理模块（agent），用于将该输入字节流解释为该相关联的预先确定数据格式的函数，并用于将该输入字节流解析为文件对象流，该文件对象流提供在该输入字节流内的基本结构的内部表示。该系统还包括一个核心文件引擎，用于将这些该文件对象转换为一个内部表示数据格式，并用于将该内部表示映射到显示器上的一个位置。该系统内的一个形状处理器处理该内部表示数据，以便驱动一输出装置以通过该内部表示表达的方式呈现该内容。

现在将参照附图描述本发明的实施例，并仅仅作为示例。

附图的简短说明

根据下面参照附图的进一步说明，本发明的上述及其他目的和有益效果将更清楚，其中：

图 1 是示出根据本发明的数字文件处理系统的一个实施例的方框图。

图 2 的方框图以更多细节呈现图 1 中描述的系统；

图 3 是一个文件代理模块的流程图；

图 4 示意地描述了可以被图 1 中的系统处理的类型的示范性文件；

图 5 描述了被用来减少文件的内部表示中的冗余的两个示范性处理的流程图；而

图 6 - 8 描述了用于存储一处理后的源文件的内部表示的示范性数据结构。

图示实施例的详细说明

此处所述的系统和方法包括计算机程序，这些计算机程序运行以便处理输出流或输出文件，其中的输出流或输出文件是通过应用程序为了在输出装置（例如视频显示器）上提供该输出而产生的。根据本发明的应用程序可以处理这些流以便创建该输出的内部表示，并且可以进一步处理该内部表示，以便产生新的输出流，该新的输出流可以作为本发明的应用程序产生的输出被显示在输出装置上。因此，本发明的系统将应用程序与显示处理分离，从而使得该应用程序不用必须在一特定的显示设备上显示其输出，并进一步消除了当为了显示该应用程序的输出而处理该输出时、提供该应用程序的需要。

为了图示这种操作，图 1 提供了一系统 10 的高层功能方框图，该系统允许若干应用程序（集中以单元 13 一起示出）将他们的输出流传送到一计算机处理过程 8，该计算机处理过程 8 处理那些输出流，并产生由那些流创建的集合输出的表示，以用于显示在设备 26 上。该应用程序 13 的集合输出在图 1 中通过输出打印机设备 26 来说明，该输出打印机设备 26 呈现由不同应用程序 13 产生的输出内容。本领域中普通技术人员可以理解，该输出装置 26 呈现由计算机处理过程 8 产生的输出，而这些输出集中传送多个应用程序 13 的内容。

在由图 1 提供的图示中，所呈现的内容包括若干图像，并且该输出设备 26 是一个显示器。然而对于本领域中普通技术人员显而易见的是，在其他实际应用中，该内容可以用除图像以外的格式传送，例如听觉的、触觉的或适合于将信息传达给用户的任何其他格式或格式的组合。此外，本领域中普通技术人员可以理解的是，输出设备 26 的类型将根据该应用程序而变化，并且可以包括用于呈现音频内容、视频内容、打印内容、绘制内容或任何其他类型内容的设备。为了说明起见，此处所述的系统和方法将主要被示为通过显示设备显示图形内容，然而可以理解的是，这些示范性的系统仅仅是为了说明起见，而无论如何不能被理解为限制性的说明。因此，由该应用程序 13 产生的输出由计算机处理过程 8 处理和集合，以便创建一个单一显示，其中包括由单独的应用程序 13 产生的全部内容。

在所述实施例中，显示器 26 上显现的每一表示输出为一文件，并且每一所述文件可以与应用程序 13 之一相关联。应该理解的是，这里所用的术语文件将包含文件、流式视频、流式音频、网页以及任何其他可以由计算机处理过程 8 处理和显示的数据形式。计算机处理过程 8 产生一个单一的输出显示，包括在其中显示有应用程序 13 产生的一个或多个文件。所显示的文件集合表示由该应用程序 13 产生的内容，而这一内容被显示在由计算机处理过程 8 产生的程序窗口内。用于计算机处理过程 8 的程序窗口还可以包括一组图标，表示配置有图形用户接口的工具，并能够使得用户能够在显示的情况下、控制在程序窗口中出现的文件的操作。

与此相反，传统方法让每个应用程序形成其自己的显示，这将导致在显示设备 26 上形成包括若干程序窗口的表示，一般每个应用程序 13 对应一个窗口。此外，每个不同类型的程序窗口将包括一组不同的工具，用于对显示在窗口中的内容进行操作。因此，本发明的系统 10 具有这样的优点：提供一个一致用户界面，并且仅仅要求

了解一组用于显示和控制不同文件的工具。此外，该计算机处理过程 8 对应用程序 13 的输出进行操作，从而仅仅要求该输出来创建出现在程序窗口内的文件。因此，没有必要让该应用程序 13 驻留在与处理过程 8 相同的机器上，也不需要应用程序 13 与计算机处理过程 8 协同操作。该计算机处理过程 8 仅仅需要这些应用程序 13 的输出，而这个输出可以源自于由该应用程序 13 在早些时候创建的已存储数据文件。然而，此处所述的系统和方法，可以作为系统的一部分使用，其中一应用程序能够呈现其自己的内容、控制至少显示器 26 的一部分并在一个与所述应用程序相关联的程序窗口内呈现内容。在这些实施例中，本发明的系统和方法可以作为单独应用程序来工作，这些单独应用程序出现在显示器上并在为其使用而设的显示器的一部分内。

更具体来讲，图 1 描述了若干应用程序 13。这些应用程序可以包括文字处理程序，例如 Word、WordPerfect 或其他任何类似的文字处理程序。它还可以进一步包括这样的程序：产生 HTML 文件的 Netscape Composer、处理 PDF 文件的 Adobe Acrobat，传送 XML 或 HTML 的网络服务器、产生音频-视频数据流的流式服务器、电子邮件客户机或服务、数据库、电子制表软件或以一文件、数据流或者任何其他种类的应用程序，而该其他种类的应用程序传送输出作为文件、数据流或以某些适合于由计算机处理过程使用的其它格式。在图 1 的实施例中，每一应用程序 13 将其输出内容提交给计算机处理过程 8。操作中，这个过程可以通过让应用程序处理过程 13 将其输出流作为输入字节流引导给计算机处理过程 8 来实现。数据流的使用对于本领域中普通技术人员是公知的，并在文献中已经说明了，例如包括 Stephen G. Kochan 撰写的“Programming in C”，Hayden Publishing (1983)。可选择的是，应用程序 13 可以创建诸如 Word 文件的数据文件，该文件可以通过单独的应用程序或者通过计算机

处理过程 8 以流方式进入计算机处理过程 8。

所述计算机处理过程 8 能够处理各种输入流，以创建在显示设备 26 上显示的集合显示。为此，并将在以下更详细说明，计算机处理过程 8 处理这些输入流，以产生每一输入流的内部表示。在实际应用中，该内部表示被指定为看起来尽可能接近相应的应用程序 13 的输出流。然而其他的实施例中，该内部表示可被创建为对于由相应的应用程序 13 产生的输出流具有经过挑选的、简化的或局部的相似性。另外并且可选的是，此处所述的系统和方法还可对所述被解释内容采用过滤器，从而允许从所显示的内容中除去所述内容的某些部分，或者相反来呈现所述内容的某些部分。进一步来讲，此处所述的系统和方法可以允许源文件结构的变更、允许在一文件内复原内容、重排所述文件的结构或仅仅选择某些类型的数据。同样，在一可选实施例中，可以在解释处理期间添加内容，包括链接到网络站点的有效内容。不论是哪种情况，由该计算机处理过程 8 创建的内部表示都可以由计算机处理过程 8 进一步处理，以驱动所述显示设备 26 创建图 1 中表示的集合图像。

现在来看图 2，图 2 呈现了图 1 中的所述系统的一个更详细的表示。具体来讲，图 2 描述了包括所述计算机处理过程 8、所述源文件 11、和一显示设备 26 的系统 10。所述计算机处理过程 8 包括若干文件代理模块 12、一内部表示格式文件和处理过程 14、缓冲存储器 15、类属对象库 16、核心文件引擎（在该实施例中包括解析模块 18 和呈现（rendering）模块 19）、一内视图 20、一形状处理器 22 和一终端输出 24。图 2 进一步描述了一可选的输入设备 30，用于将用户输入 40 传输给计算机处理过程 8。所述实施例包括一个处理过程 8，其中包括一形状处理器 22。然而，对于本领域中普通技术人员显而易见的是，所述处理过程 8 仅仅是示范性的，所述处理过程 8 可以通过替代的处理和体系结构来实现。例如，所述形状处理器 22 可以

选择性地被实现为硬件部件，例如一半导体器件，它支持所述处理过程 8 的其他组件的操作。此外，很清楚的是，尽管图 2 以包括单个系统的功能方框图呈现处理过程 8，处理过程 8 也可能是分布在许多不同的平台上，还可能是所述组件在不同的时间运行，而来自处理过程 8 的一个组件的输出随后才被作为输入传送给处理过程 8 的下一组件。

如上所述，每个源文件 11 与文件代理模块 12 相关联，它能够将输入的文件翻译成所述源文件 11 的内容的内部表示。为了确定适当的文件代理模块 12 来处理源文件 11，图 1 中的所述系统 10 包括一应用程序调度模块（没有示出），它控制应用程序和所述系统 10 之间的接口。在一实际应用中，外部应用编程接口（API）的使用是通过所述应用程序调度模块解决的，它传递数据、调用适当的文件代理模块 12、或者相反实现由应用程序提出的请求。为了给一特定的源文件 11 选择适当的文件代理模块 12，所述应用程序调度模块将所述源文件 11 通知给所有加载的文件代理模块 12。然后这些文件代理模块 12 以与它们的特定适配性相关的信息为响应，以便翻译所公布的源文件 11 的内容。一旦所述文件代理模块 12 已经作出响应，所述应用程序调度模块选择一个文件代理模块 12，并将一个指针（例如所述源文件的一个 URI（通用资源识别码））传递给所选定的文件代理模块 12。

在一实际应用中，所述计算机处理过程 8 可以作为一项业务运行，经由该业务可以创建若干进程活动，借以支持多个文件源 11 的多重处理。在其他实施例中，所述处理过程 8 并不支持多进程活动，而是在当前进程活动中将调用由所述应用程序调度模块所选定的文件代理模块 12。

很清楚的是，图 2 中的该示范性的实施例提供了用于处理不同文件格式的输入数据流的一个灵活和可扩展的前端。例如可选的是，

如果该应用程序调度模块确定该系统缺少一个适用于翻译该源文件11的文件代理模块12，该应用程序调度模块可以发信号给相应的应用程序13，指出该源文件的格式未被识别。可选的是，该应用程序13可以选择允许源文件11的重新格式化，例如通过将由该应用程序13产生的源文件11从它的现在的格式转换为另一个由所述应用程序13支持的格式。例如一个应用程序13可以确定该源文件11需要以一个不同的格式（例如该文件格式的一个早期版本）存储。结果，应用程序13支持所述格式，该应用程序13可以用这些支持的格式重新存储该源文件11，以便由该系统10提供的一个文件代理模块12能够翻译该源文件11。可选的是，该应用程序调度模块一旦检测到所述系统10缺少一适当的文件代理模块12，就可以向用户指示可能需要一新的特定类型的文件代理模块用于翻译当前源文件11。为此，该计算机处理过程8可以指示该用户需要将一个新的文件代理模块载入该系统10，并且可能将该用户指引到一个可以从中下载该新的文件代理模块12的位置，例如一个网络站点。可选的是，该系统可以自动地取出该代理程序而无需询问该用户，或者可以识别一个类属代理程序12，例如一个可以抽取源文件表示文本的部分的类属文本代理程序。进一步来讲，还可以提供代理程序，来提示用户在翻译处理期间的输入和指令。

在一个更进一步的可选实施例中，一个应用程序调度模块和该文件代理模块12结合在一起作为一个输入模块工作，用于根据各种标准中的任何一个来识别源文件11的文件格式，例如该文件内部的一个明确的文件类型标识符，根据文件名，包括文件名扩展部分，或者根据已知的特定文件类型的内容特征。该字节流被输入给该文件代理模块12，指定为该源文件11的文件格式。

尽管以上的说明已经论述了由一个流或者计算机文件提供输入数据，但应该能被本领域中普通技术人员理解的是，该系统10还可

以被施加从一输入设备（诸如数字式摄象机或者扫描仪）接收的输入，以及从可以直接地将它的输出流给该处理过程 8 或者通过一个操作系统将它的输出流给该处理过程 8 的应用程序接收的输入。在该情况下，输入字节流可以直接来源于该输入设备而不是来源于一个源文件11。然而，该输入字节流仍将是适用于由所述系统10处理的数据格式，并且为了本发明的目的，从这样一个输入设备接收的输入可以被认为是一个源文件11。

如图 2 所示，所述文件代理模块12采用标准对象库16来产生内部表示14，以按照在所述库16中定义的一类类型的文件对象的集合、连同定义所述文件内部的各种文件对象的特定实例的性质的参数，来描述所述源文件的内容。因此，所述库16提供了一组所述文件代理模块12、所述解析程序18以及所述系统10已经了解的若干类型的对象。例如，所述内部表示14中采用的所述文件对象可以包括：文本、位图图形以及矢量图形文件对象，这些矢量图形文件对象可能是活动的、或者可能不是活动的，并且可能是两维或者三维的：视频、音频以及各种类型的交互对象（例如按钮和图标）。矢量图形文件对象可能是具有规定的填充和透明度的类似PostScript（页面描述语言）的路径。位图图形文件对象可能包括一组子对象类型，例如JPEG、GIF和PNG对象类型。文本文件对象可表示一个特殊文本的区域。所述区域可以包括一个文本段落，一般被理解为一组字符，出现在两个分隔符（类似一对回车符）之间。每个文本对象可以包括一连串字符和所述字符串的样式信息，包括一个或多个相关联的字体、标点及其他此类的样式信息。

定义文件对象的特定实例的参数通常包括定义所述文件对象的物理形状、大小和位置的空间坐标系以及任何相应的用于定义性质随时间而变的文件对象的瞬态数据，从而允许所述系统处理动态文件结构和/或显示功能。例如，一个视频输入流可以由所述系统10作

为一系列以例如30帧每秒的速率变化的图形来处理。在该情况下，这种图形对象的瞬态特征指示所述图形对象每秒更新30次。如上所述，对于文本对象，所述参数一般地还包括被应用于字符串的字体和大小。对象参数还可能定义其他性质，例如透明度。很清楚的是，所述内部表示可以本地的文件格式存储/存放到所述系统，而输入给所述系统10的可能的源文件范围可以包括所述系统中的本地文件格式的文件。还可能是使用适当的转换代理程序，在需要的时候将所述内部表示14转换为一些其他文件格式中的任何一个。

图3描述了可以由一个文件代理模块12执行的一个示范性处理的流程图。具体来讲，图3描述了表示一个示例文件代理模块12的操作的一处理过程50，在该情况下，该文件代理模块12适用于将Microsoft Word文件的内容翻译为内部表示格式。具体来讲，所述处理过程50包括一个初始化步骤52，其中所述处理过程50初始化数据结构、存储空间及其他将由所述处理过程50在翻译所述源文件11期间使用的资源。在步骤52之后，所述处理过程50进行一系列步骤54、58和60，其中所述源文件11被分析并被分成子节。在图3中所述处理过程50中，步骤54、58和60在所述源文件11流入到所述文件代理模块12中时首先将其细分为小节，然后将所述小节细分为段落，然后将段落细分为组成所述段落的单独的字符。可以在一张分块表格(piece table)内识别出所述源文件11内识别出的所述小节、段落和字符，该表格包含在所述源文件11内指向不同子节的指针。本领域中普通技术人员可以理解的是，在图3中描述的该分块表格表示由MSWord采用的一构造，用于提供指向文件的不同的、所识别的子节的指针。可以进一步理解的是，分块表格或者类似结构的分块表格的使用是可选的，而且取决于使用的应用程序，包括取决于被处理的文件的类型。

当在步骤60中所述处理过程50开始识别出现在一特定段落内的

不同的字符时，该处理60可以进行到步骤62，其中一个样式被施加于在步骤60中识别出的字符或一组字符。一个样式的施加被理解为将所述识别出的字符与表示样式相关联，该表示样式与那些字符一起被使用。所述表示样式可以包括与所述字符相关联的性质，包括字体类型、字体大小，无论所述字符是被加粗、变斜或是采用其他样式。另外，在步骤62中，所述处理可以确定该字符是否被旋转，或者是否沿一曲线轨迹或者其他形状放置。另外，在步骤62中，与字符在其中的段落相关联的样式也可以被识别出并与这些字符相关联。这样的性质可以包括与段落相关联的行距、与该段落相关联的页边距、字符之间的间隔及其他此类性质。

在步骤62之后，该处理过程50进行到步骤70，其中该内部表示被建立。说明文件结构的对象在步骤64中被作为内部表示内的一个对象创建，该对象的关联样式连同其包含的字符串在步骤68在内部表示内被独立地创建。将在之后更详细解释的图6、7和8，这些图描述了由该处理过程50创建的文件结构，其中一个文件的结构由一组文件对象记录，而与这些文件对象相关联的数据被保存在一个单独的数据结构中。在步骤70之后，处理过程50进行到判定块72，其中该处理过程50确定与上次已处理字符相关联的段落是否完成。如果该段落没有完成，该处理过程50返回到步骤60，读取该段落的下一字符。或者，如果该段落是完成的，该处理过程50进行到判定块74，在其中处理过程50确定该节是否完成。如果该节是完成的，该处理过程50返回到步骤58，读取该表格的下一段落。或者，如果该节是完成的，该处理过程50进行到步骤54，其中如果存在下一节的话就从所述分块表格读取下一节，并且处理继续。一旦所述文件已经被处理完，系统8可以传输、存储、输出或者相反为后来的使用存贮所述解释后的文件。所述系统可以用一个与所述内部表示兼容的格式存贮所述解释后的文件，并且可选用其他格式以及包括与

所述源文件11的文件格式兼容的格式（可以采用没有示出的“输出文件代理模块”，能够接收内部表示数据和创建源文件数据）存贮，或者采用二进制格式，文本文件描述结构，标记文本（marked-up text），或者用任何其他适当的格式存贮；而且可以采用一通用的文本编码模型，包括unicode（统一代码）、shifmapping（移位映射）、big - 5（繁体代码）以及亮度 / 色度模型。

从上述可以看出，所述内部表示14的格式将如由所述对象类型与它们的参数说明的所述文件的“结构”（或者“布局”）与各种对象的“内容”分离；例如一文本对象的字符串（内容）与所述对象的空间参数分离；一图形对象的图像数据（内容）与它的空间参数分离。这允许以一种压缩的方式定义文件结构，并且为远端存储并只有当需要时才由系统取出的内容数据提供了选项。所述内部表示14用“高级”描述来说明所述文件和它的构成对象。

如上参照图 3 所述的文件代理模块12，能够处理由MSWord字处理应用软件创建的一数据文件，并将该数据文件翻译为一个内部表示，该内部表示由一组从所述库16中选择出来的对象类型形成，表示所述处理文件的内容。因此，所述文件代理模块12分析所述Word文件并将该文件的结构和内容翻译为一个内部表示为所述计算机处理8所知的内部表示。在图 4 中描述了可以由所述文件代理模块12处理的一种Word文件的一个例子。具体来讲，图 4 描述了由MSWord应用程序创建的Word文件32。所述文件32包括一个信息页面，其中一个页面包括两列文本34和一个图表36。图 4 进一步描述了这些文本列34和图表36是以这样一种方法位于页面38上：一列文本从页38的顶端一直排到页面38的底部，第二列文本从页面的中部附近一直排到到页面底部，而图表36被置于第二列文本34上方。

如上参照图 3 所述，所述文件代理模块12开始通过确定所述文件32包括一个页面和包含若干不同的对象来处理所述文件32。对

于由所述文件代理模块12发现的一个页面，该文件代理模块12识别所述页面的样式，例如可能是一个肖像画格式的8.5 x 11页面的页面样式。由该文件代理模块12识别出的页面样式被嵌入到该内部表示中，以供以后由解析程序18在将文本格式化和使文本流进由处理过程8创建的文件时使用。

对于图4中所描述的文件32仅仅给出一个页面。然而，很清楚的是，文件代理模块12可以处理包括若干页面的Word文件。在这种情况下，文件代理模块12将分别通过创建一个页面然后用在库中发现的类型的对象填充它来处理每个页面。因此页面样式信息可以包括：一个文件包含若干页面以及这些页面具有一特定大小。其它的页面样式信息可以由该文件代理模块12识别出，而识别出的页面样式信息可以根据该应用程序变化。从而，可以由一个能够处理Microsoft Excel文件或者一实时介质数据流的文件代理模块识别出不同的页面样式信息。

进一步参照图4描述，一旦该文件代理模块12已经识别出该页面样式，该文件代理模块12可以开始将该文件32分解为若干个对象，这些对象可以被映射到为该系统所知的、并且一般保存在库16中的文件对象。例如，该文件代理模块12可以处理该文件32，以发现文本对象、位图对象和矢量图形对象。可以选择提供其他类型的对象类型，包括视频类型、动画类型、按钮类型和手写体类型。在该实际应用中，该文件代理模块12将识别出一个文本对象34，其相关的样式具有两列。在该文本对象34内出现的文本段落可以被分析，来识别每个相应的段落中的每个字符。处理过程50可以将样式性质施加于每个识别出的字符串，而在文件32内识别出的每个字符串可以被映射到该库16内列举类型的文本对象。每个字符串和施加的样式可以被理解为由文件代理模块12识别出的对象，如已经在文件32内被发现，并且已经被译为一个文件对象，在该情况下，该文件对象

是库16内列举类型的一文本对象。该内部表示对象可以从文件代理模块12流进内部表示14。该文件代理模块12可以继续将在文件32内出现的对象翻译成为系统10所知的文件对象，直到每个对象都已经被翻译。这些对象类型可能适合于该应用程序，并且可能包括适用于翻译源数据的对象类型，该源数据表示数字文件、音频/视频表示、音乐文件、交互脚本、用户界面文件和图像文件、以及任何其他文件类型。

现在来看图5，能够看出图5中描述的处理过程80允许将源文件11的内部表示中出现的相似对象进行压缩，以便减少内部表示的尺寸。举例来说。图5描述了一处理过程80，其中步骤82具有一基本库对象A，在步骤84中通过将基本对象插入到将成为源文件11的内部表示的文件中来处理该基本库对象A。在步骤88中，另一个由文件代理模块12提供的对象B被传送给该内部表示文件处理14。然后该处理过程80进行步骤序列92—98，其中将对象A的特征与对象B的特征进行比较，以确定这两个对象是否具有相同的特征。举例来说，如果对象A和对象B表示两个字符例如字母P和字母N，如果字符P和N色彩相同、字体相同、尺寸相同并且样式（例如粗体或者斜体）也相同，那么处理过程80在步骤94中将这两个对象结合在一起，并处于存贮在内部表示内的一个对象分类中。如果这些特征不匹配，那么处理过程80将他们作为两个单独的对象添加到内部表示中。

图5描述了一个处理过程80，其中该内部表示文件14将这些对象压缩为一个物理上邻近对象的相似性的函数。本领域中普通技术人员可理解的是，这仅仅是用于压缩对象的一种处理，可以采用其他方法。例如，在一可选实际应用中，该压缩过程可以包括用于压缩视觉上邻近的对象的处理。

图6、7和8描述了已经由图1和2中描述的系统处理过的文件的内部表示之结构。该文件的内部表示可以作为存储在磁心存储

器中的计算机文件或者数据来嵌入。然而，对于本领域中普通技术人员显而易见的是，所选的用于记录或者传输该内部表示的数据结构可以根据应用而变化，而且任何适当的数据结构都可以与此处所述的系统和方法结合使用，而不会脱离本发明的范围。

在下文将更详细的说明被处理文件的内部表示的结构将文件的结构与文件的内容分离开。具体来讲，文件的结构由数据结构记录，该数据结构指出组成该文件的不同文件对象以及这些文件对象相互之间排列的方式。在图6中示出结构与内容的分离，其中数据结构110记录被处理的文件的结构，并以一种与该文件相关联的内容无关的数据格式存储该结构。具体来讲，该数据结构110包括一资源表112和一个文件结构114。该资源表112提供一个用于构造该文件的内部表示的资源列表。例如该资源表112可以包括一个或多个出现在文件内部的公共结构之表格，例如字体、链接和色彩列表。可以在该资源表112内用编号引用这些公共结构。资源表112的资源涉及在文件结构114内安置的文件对象。如图6所示，文件结构114包括若干由嵌套括号组表示的存储包(container)118。在存储包118内是若干文件对象120。如图6所示，存储包118表示在正在被处理的文件内出现的文件对象的集合。图6进一步示出，存储包118还能够拥有子存储包。例如，文件结构114包括一个由外面一组标为1的括号识别出的顶级存储包，还具有三个嵌套存储包2、3和4。另外，存储包4在存储包1和存储包3内被两次嵌套。

每个存储包118表示在文件内的特征，其中这些特征可以是单独的文件对象(例如所述文件对象120)的一个集合。这样举例来说，一个文件例如图4中描述的文件32，可以包括一个表示该字符串的存储包，其中该字符串包括在列34内出现的文本。举例来说，在该字符串存储包内出现的不同文件对象120可以由该字符串内出现的不同段落表示。该字符串存储包具有一个与之相关联的样式。举例来

说，图4中描述的字符串可以包括表示该字符字体类型、字体大小、样式（例如粗体或者斜体字样式）的样式信息，以及表示该列的尺寸包括宽度和长度的样式信息，该列中存在该字符串或者该字符串的一部分。这些样式信息可以在以后被解析模块18使用，用来对上下文专用视窗20内的文本重新格式化和使之回流。存储包的另一个例子可以是一个表格，举例来说可以出现在文件32的一列34内。该表格可以是一个具有对象的存储包。存储包的其他类型和运用将根据涉及的应用而变化，而本发明的系统和方法不局限于任何特定的对象类型组或者存储包。

因此，当文件代理模块12翻译源文件11时，将遇到属于已知对象类型的对象，而该文件代理模块12将请求库16创建一个适当对象类型的对象。该文件代理模块12于是将所创建的文件对象放置到文件结构114内的适当位置，以保持源文件11的整体结构。举例来说，当文件代理模块12遇到源文件11中的图像36时，文件代理模块12将图像36（例如可能是一个JPEG图像）识别为一个位图类型的对象，而可选子类型为JPEG。如图3中步骤64和68所示，这个文件代理模块12可以创建适当的文件对象120并且可以将所创建的文件对象120放置到结构114中。另外，用于JPEG图像文件对象120的数据，或者在另一个例子中，用于字符的数据以及一字符串的相关样式，可以被存贮在图8中描述的数据结构150内。

当源文件11正在被处理时，文件代理模块12可以识别其他的存储包，其中这些其他的存储包可以表示在一个已存在的存储包（例如字符串）内出现的子特征。例如，这些子特征可以包括到参考资料的链接，或者是到出现在文件内、并且包含单独的文件对象120的集合的剪贴的直观区域或者特征的链接。该文件代理模块可以将这些文件对象120放置在一个单独的存储包内，而该存储包将被嵌套到已经存在的存储包中。图7A中以一树状结构130示出这些文件对象120

以及存储包118的布局，其中单独的存储包1、2、3和4分别被示为存储包对象132、134、138和140。存储包118和文件对象120被以树状结构布置，该树状结构示出文件结构的嵌套的存储包结构以及在该存储包118内存在的不同文件对象120。图7A的树状结构还图示出该结构114记录和保持了源文件11的结构，以文件对象120的分级结构显示了源文件，其中该文件对象120包括样式信息，例如在其中出现一字符串的列的尺寸，或者是瞬态信息，例如用于流式内容的帧频。因此，通过一系列参数化的组件说明了每个文件的图形结构。在下面的表1中给出了一个例子。

表 1

参数	举例
类型	位图
边界框	400, 200; 600, 700单元（左下，右上）
填充	对象17
阿尔法	0（没有）
形状	对象24
时间	0, -1（无限）（开始，结束）

可以看出，表 1 给出了可以用来说明一个文件的图形结构的参数的一个例子。表 1 给出例如对象类型的参数的例子，在该情况下是一个位图对象类型。提供了边界框参数，并给出该文件对象在源文件11中的位置。表 1 进一步提供了采用的填充以及一个表示该对象的透明度的阿尔法因子。形状参数提供了一个关于对象形状的句柄（handle），在该情况下可以是一条定义对象轮廓的路径，包括

不规则形状的对象。表1还给出一个时间参数，它表示该对象的瞬态变化。在这个例子中，该图像是稳定的及没有随时间而变化。然而，如果该图像对象给出流式媒体，那么该参数可以包含一个指示该对象变化的速率的瞬态特性，例如该内容的可与期望帧频相比较的速率。

因此，该构造单元都是具有可流动数据内容的存储包，这些可流动数据单独保持并通过存储包的一个句柄引用。以这种方法，可以根据该文件结构远端地保持任何或者所有的数据内容。这允许文件以可以本地保持和远端保持的数据内容的混合的方式呈现文件。另外，该数据结构允许对源文件的内部表示的快速渐进的呈现，可以首先呈现较宽泛和较高层的对象，而细微特征可以在随后顺序中被呈现。因此，单独的结构和数据允许在流式数据“填充”内容的时候呈现视觉文件。另外，内容和结构的分离允许容易地编辑或者改变文件的内容。因为文件结构与内容无关，可以将不同的内容代入该文件结构。这可以通过存储包原理在存储包上完成或者对整个文件进行。文件的结构可以被传送而独立于内容，而在之后提供内容，或者是该结构被传送所至的平台上进行呈现。

此外，图7A表明源文件11的结构可以被表示为一个树状结构130。在一实际应用中，该树状结构可以被修改和编辑，以改变源文件11的表示。例如，该树状结构可以被修改以便向该树130增加另外的结构和内容。这在图7B中进行了描述，图7B示出图7A在一高层存储包下复制和呈现的原始树状结构。这样，图7B示出可以通过处理由文件代理模块12产生的树状结构130创建一个新的文件结构，从而创建一个新的表示。这就允许对一个文件内的对象的视觉位置进行改变，而不同的对象120的相对位置可以保持相同。通过调整树状结构130，此处所述的系统可以编辑和修改内容。例如，在那些其中树状结构130中的内容表示视觉内容的应用中，此处所述的系统可以编

辑该树状结构以便复制该文件的图像并且并排地呈现文件的图像。替换的方案是，该树状结构130可以被编辑并被补充，以便增加另外的视觉信息，例如通过增加一个新文件或者文件的一部分的图像。此外，通过控制树状结构改变的速率，此处所述的系统可以创建一个文件逐渐改变的错觉，例如滑过一个例如显示设备26的显示器，或者逐渐改变为一个新的文件。还可以实现其他效果，例如缩略图的创建及其他类似结果，本领域中的普通技术人员通过对此处所述的系统和方法做修改，而这样修改后的系统和方法将属于本发明的范围。

该源文件11的数据独立于结构114存贮。为此，每个文件对象120包括一个指向与对象相关联的数据的指针，而这个信息可以被安排在一个间接寻址列表内，例如图8中描述的间接寻址列表。如图8所示，在这个实际应用中，每个文件对象120被编号并创建了一个间接寻址列表152，其中每个文件对象编号154与一个偏移量158相关联。例如，由附图标记160标志出的文件对象编号1可以与由附图标记162标志的偏移量700相关联。这样，该间接寻址列表将对象编号1与偏移量700相关联。该偏移量700可以表示磁心存储器中的一个位置，或者一个文件偏移量，其中与对象1相关联的数据可以驻留。如在图8中进一步示出的，可以存在一数据结构150，其中可以存贮表示与一相应的文件对象120相关联的内容的数据。因此举例来说，位于跳转位置700的所述对象1，可以包括表示在图6中描述的存储包1的字符串中存在的字符的Unicode字符。类似地，在图8中由附图标记172描述的对象2数据，与由附图标记170识别的磁心存储器位置810相关联，可以表示JPEG位图，该位图与在图6的文件结构内引用的位映象文件对象120相关联。

本领域中普通技术人员应注意的是，因为数据与结构独立，一源文件的内容被保存在一集中式储存库（centralized repository）

中。因而，此处所述的系统允许对不同类型的数据对象压缩。这样的处理在有限资源系统中提供更大的存储灵活性。

返回到图2，很清楚的是，一旦用于压缩一个内部表示文件的内容的处理完成了对不同的对象的压缩，这些对象被传递给解析模块18。该解析模块18对在内部表示的结构部分中识别的对象进行解析，并参照与这一对象相关联的数据内容，再次将位置和样式信息应用到每个对象。呈现组件19产生一个由内部表示14表示的文件的上下文特定表示，也就是“视图”20。所要求的视图可以是所有这些文件、一个完整文件或者一个或一些文件的一部分。该呈现组件19接收视图控制输入40，该视图控制输入40定义将要被产生的特定文件视图的所视上下文以及任何有关瞬态参数。举例来说，可以要求该系统10产生一个文件的一部分的图象缩放视图，然后变为全景展开或者滚动该图象缩放视图以显示该文件的邻近部分。该视图控制输入40被该呈现组件19解释，以确定特定的视图要求内部表示的哪一部分，以及该视图怎样显示、何时显示和显示多长时间。

该上下文特定的表示/视图20是用基本形状和参数表达的。

该呈现组件19还可以在产生源文件11之所要求的视图20时，对内部表示的相关部分执行另外的预处理功能。该视图表示20被输入给一形状处理器22，用于处理以产生一个适当格式的输出，来驱动一个输出设备26，例如显示设备或者打印机。

该呈现组件19的预处理功能可以包括色彩校正、分辨率调整/增强和防叠处理（anti-aliasing）。分辨率增强可以包括图象缩放功能，当由目的输出设备显示或者再现对象的内容时保持其清晰度。分辨率调整可以上下文相关；例如当该文件视图是静态的时候，在该显示文件视图正在展开或者滚动和增加期间，特定的对象的显示分辨率可以被减少。

可选的是，在解析模块18和内部表示14之间可以有一条反馈路

径42，例如为了触发内部表示14的内容的更新，比如在由内部表示所表示的源文件11包括多帧动画的情况。

该呈现组件19的输出以基本对象表示文件。对于每一文件对象，该呈现组件19的表示至少以一物理的、矩形边界框定义该对象，而该对象的实际轮廓路径由该边界框、对象的数据内容和它的透明度界定。

该形状处理器22解释该基本对象并将它转换为适合于目标输出设备26的输出帧格式；例如对于打印机的点映射，对于绘图仪的向量指令组 或者对于显示设备的位图。一个对形状处理器22的输出控制输入44给该形状处理器22提供信息，以便产生适用于特别的输出设备26的输出。

该形状处理器22最好处理由视图表示20用“形状”（即对象的轮廓形状）、“填充”（对象的数据内容）以及“阿尔法”（对象的透明度）来定义的对象，执行适合于所要求的视图和输出设备的图象缩放和剪贴（一般利用通过扫描转换或类似手段得到的像素，用于大多数种类的显示设备或者打印机）。该形状处理器22可选地包括一边缘缓存器，它利用扫描转换的像素定义对象的形状，并且最好是对轮廓形状应用防叠处理。可以用一种以输出设备26的特征确定的方式，通过对对象边界施加灰度级梯度（grey-scale ramp）执行防叠处理。该方法启用储存有效的形状剪贴以及形状交叉处理，并且也是存储有效的和处理器有效的。可以采用查找表或者其他技术来定义多频声响应曲线，提供非线性的呈现控制。由形状处理器22处理的单独的基本对象被组合成合成输出帧。在题为Shape Processor（形状处理器）、与此同一日期申请的专利申请中，更详细的示出了适合与此处所述的系统一同使用的一种形状处理器的设计方案，该申请的内容合并作为参考。然而，可以采用任何适当的形状处理器系统或者处理过程，而不会脱离本发明的范围。

如上所述，图1中描述的处理过程8可以被实现为一个软件组件，在一数据处理系统上运行，所述数据处理系统比如是手持式计算机、移动电话、机顶盒、传真机、复印机或者其他办公设备、嵌入式计算机系统、Windows或者UNIX工作站或者任何其他能够完全或者部分地支持如上所述的文件处理系统的计算机/处理平台。在这些实施例中，所述系统可以用C语言计算机程序或者以任何高级语言（包括C++、Fortran、Java或者Basic）编写的计算机程序实现。此外，在采用微控制器或者DSP（数字信号处理器）的实施例中，所述系统可以使用以微代码编写的计算机程序或者是以高级语言编写并且被编译为可在所采用平台上执行的微代码的计算机程序实现。此类系统的开发对于本领域中普通技术人员是已知的，而且此类技术在 Intel® StrongARM processors SA - 1110 Microprocessor Advanced Developer 's Manual进行了阐述。此外，用于高级编程的一般技术是已知的，举例来说，在Stephen G. Kochan的 Programming in C（Hayden Publishing，1983）中就进行了阐述。应注意的是，数字信号处理器特别适合于实现信号处理功能，包括诸如通过调整对比度、边缘清晰度和亮度进行图像增强的预处理功能。用于数字信号处理器和微控制器系统的代码开发按照在本技术领域中公知的原理。

因此，尽管图1和2用图形将该计算机处理过程8描述为包括若干功能块单元，对于本领域中普通技术人员显而易见的是，这些单元可以实现为能够在数据处理平台上运行的计算机程序或者部分计算机程序，从而将该数据处理平台配置为根据本发明的系统。此外，尽管图1以一个文件处理过程8和显示设备26的集成化装置描述该系统10，对于本领域中普通技术人员显而易见的是这仅仅是一实施例，此处所述的系统可以通过其他体系结构和布局实现，包括将处理过程8的文件处理功能与由显示器26执行的文件显示操作分离的系统

体系结构。此外，很清楚的是，本发明的系统不局限于包括显示器或者输出设备的那些系统，而是本发明的系统将包含处理一个或多个数字文件、以便创建可以在一输出设备上呈现的输出的处理系统。然而，该输出可以被保存在一个数据文件中，用于随后在显示设备上表示，用于长期存储、经由网络传送或者其它不同于直接显示的目的。因此，对于本领域中普通技术人员显而易见的是，此处所述的系统和方法可以支持许多不同的文件和内容处理应用程序，而且用于特定应用程序的系统或者处理的结构将根据该应用程序及设计者的选择而变化。

根据以上所述，可以理解的是，本发明的系统可以是“硬布线的”；例如在ROM中实现和/或集成到专用集成电路（ASIC）或者其他单片系统里，或者可以实现为固件（可编程只读存储器，例如可擦除可编程只读存储器），或者被实现为软件，被本地或者远端存贮并在一特定设备要求时被取出和执行。这样的改善和修改可以被合并而不会脱离本发明的范围。

本领域中的普通技术人员仅仅使用常规实验就会知道或者能够确定，有许多等效于此处所述的实施例和实际应用的方案。举例来说，此处所述的系统和方法可以是用于处理源文件11的独立系统，但是可以选择性的是，这些系统可以许多不同的方式被合并到各种类型的数据处理系统和设备中，以及合并到外围设备里。在一通用数据处理系统（“主机系统”）中，本发明的系统可以与所述主机系统的操作系统和应用程序并列合并，或者可以被完全地或者部分地合并到所述主机操作系统里。举例来说，此处所述的系统能够在具有LCD显示器的便携式数据处理装置上快速显示各种类型的数据文件，而无需要求使用浏览器或者应用程序。采用本系统的便携式数据处理装置的例子包括“掌上型”计算机、便携式数字助理（PDA，包括片型PDA，其中主要的用户界面包括图形显示，用户通过它直接依

靠一指示笔设备进行交互)、可访问因特网的移动电话及其他通讯装置。为了便携性,这类数据处理装置要求小型、低功率的处理器。一般,这些设备采用在ASIC(专用集成电路)中设计的高级RISC型核心处理器,以便使该电子器件封装是小型的和集成的。这类设备还具有有限的随机存取存储器,并且一般没有非易失性的数据存储单元(例如硬盘)。传统的操作系统模式,例如在标准台式计算系统(PC)采用的操作系统模式,要求大功率的中央处理器和大容量存储器,用于处理数字文件并产生有效输出,是完全不适于这类数据处理装置的。具体来讲,传统的系统不提供以一集成的方式对多文件格式的处理。相形之下,此处所述的系统对所有文件格式采用公共的处理与传递途径,从而提供一个高度集成的文件处理系统,它在功率消耗和系统资源的利用方面是极其高效的。

本发明的系统可以被集成在便携式数据处理装置的BIOS级,能够用比传统系统模式低得多的系统开销进行文件处理与输出。另一方式是,这些系统可以被实现在仅仅高于传输协议栈(transport protocol stack)的最低系统级。举例来说,所述系统可以被合并到一网络设备(卡)或者系统中,以提供网络业务量的嵌入处理(例如在TCP/IP系统中的分组级工作)。

此处所述系统可以被配置为用一预定组数据文件格式和特定的输出设备运行;例如所述设备的直观显示装置和/或至少一种打印机。

此处所述的系统还可以被合并到低成本数据处理终端中,例如增强的电话和“薄的”网络客户终端(例如具有有限局部处理和存储资源的网路终端),以及可交互/可访问因特网的有线电视系统使用的“机顶盒”。所述系统还可以被合并到外围设备中,例如硬拷贝设备(打印机和绘图仪)、显示设备(例如数字投影仪)、网络设备、输入装置(摄像机、扫描仪等等)以及多功能外围设备(MFP)。

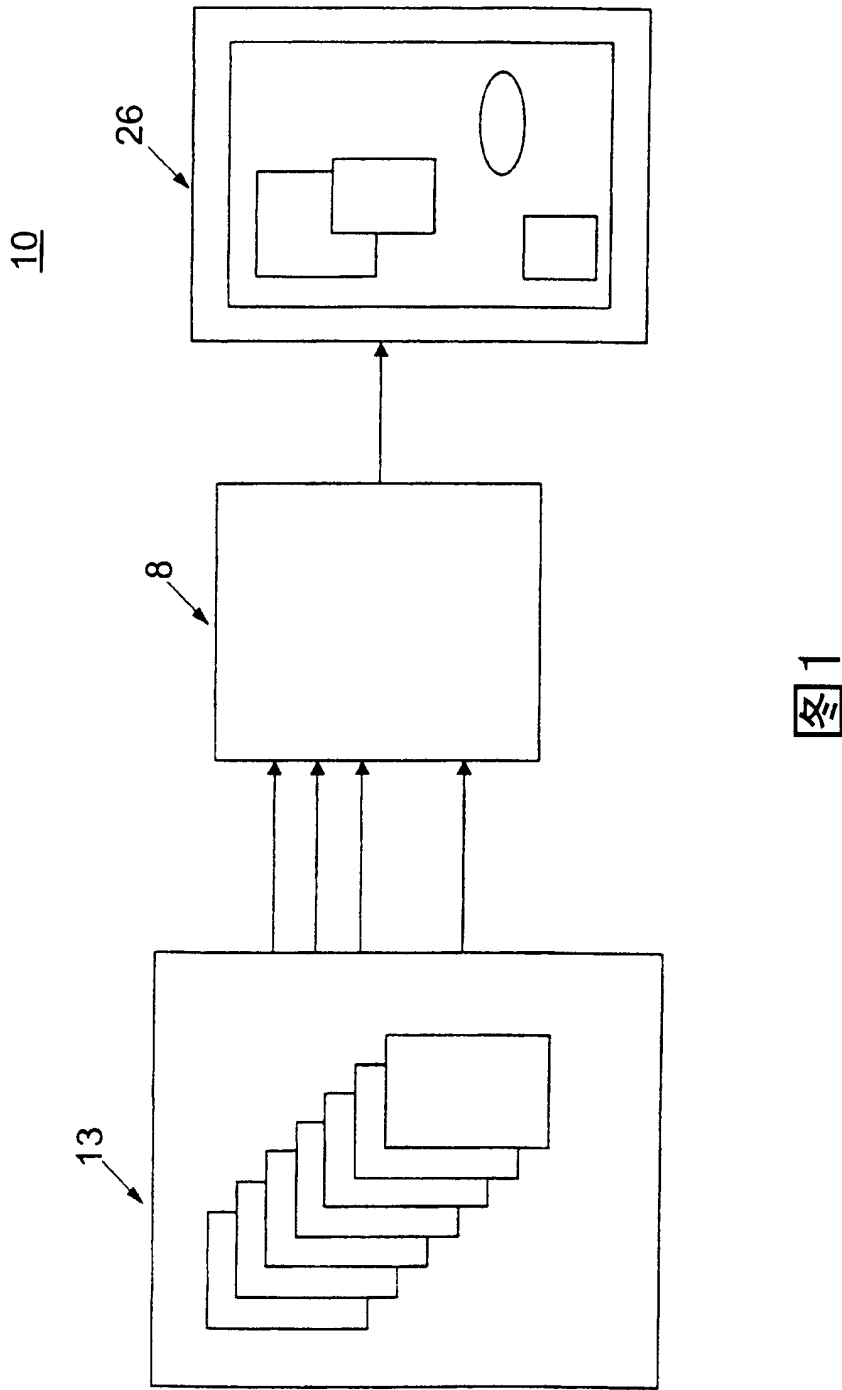
当合并到打印机中时，所述系统使该打印机能够接收来自主机数据处理系统的原始数据文件，并准确地再现所述原始数据文件的内容，而无需由所述主机系统提供特别的应用程序或者驱动程序。这避免或者减少了对配置计算机系统以驱动特定类型打印机的需要。本系统直接地产生源文件的点映射图像，适用于由打印机输出（无论本系统是否被合并到打印机本身还是合并到主机系统中都可以）。类似的设想适用于其他硬拷贝设备，例如绘图仪。

当合并到一显示设备（例如一投影仪）中时，本系统使该设备能够准确地显示原始数据文件的内容，而无需使用主机系统上的应用程序或者驱动程序，也无需主机系统和/或显示设备的特定的配置。当这些类型的外围设备装备有本系统时，可以通过任何类型的数据通信网络，接收与输出来自任何源的数据文件。

此外，此处所述的系统和方法可以被合并到汽车内系统中以为驾驶员提供信息、或者娱乐系统，有助于在车辆内传送信息或者是将信息分送给车辆以外进行通信的网络。进一步来讲，应该理解的是，此处所述的系统可以驱动具有多个输出源的设备，以便仅仅对控制参数修改就能维持一致的显示。例子包括但是不局限于合并有一个直观显示器和打印头的一个STB或者汽车内系统，从而允许文件的查看和打印而无需源应用程序和驱动程序。

根据以上所述，应该理解的是，本发明的系统可以是“硬布线的”；例如实现在ROM中和/或集成到专用集成电路ASIC或者其他单片系统里，或者可以被实现为固件（可编程只读存储器，例如可擦除可编程只读存储器ePROM），或者被实现为软件，并被本地或者远端存贮而在一特定设备要求时被取出和执行。

因此应该理解的是，本发明不限于此处公开的实施例，而是要根据权利要求书理解，依法解释为相应的范围。



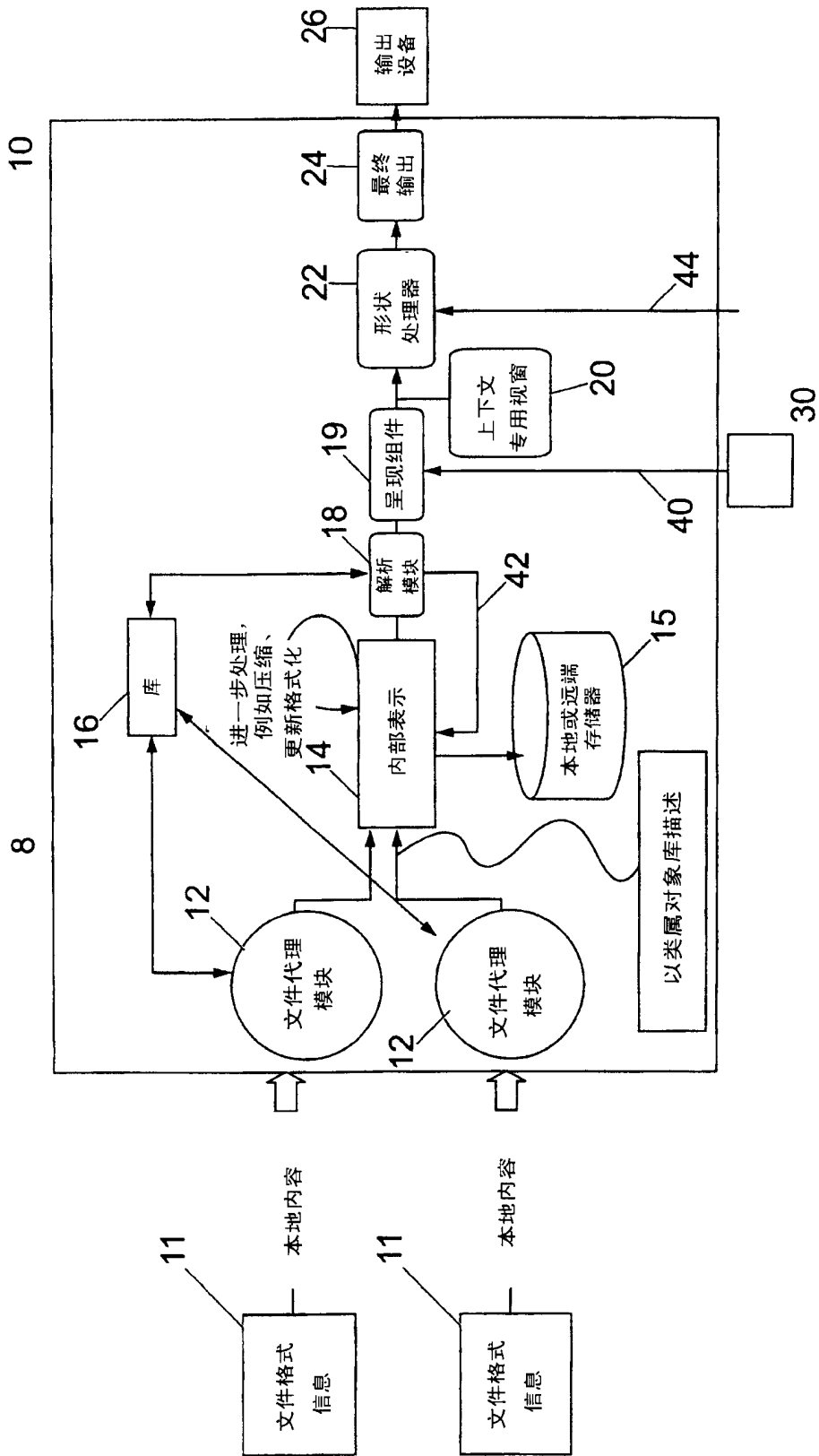


图2

50

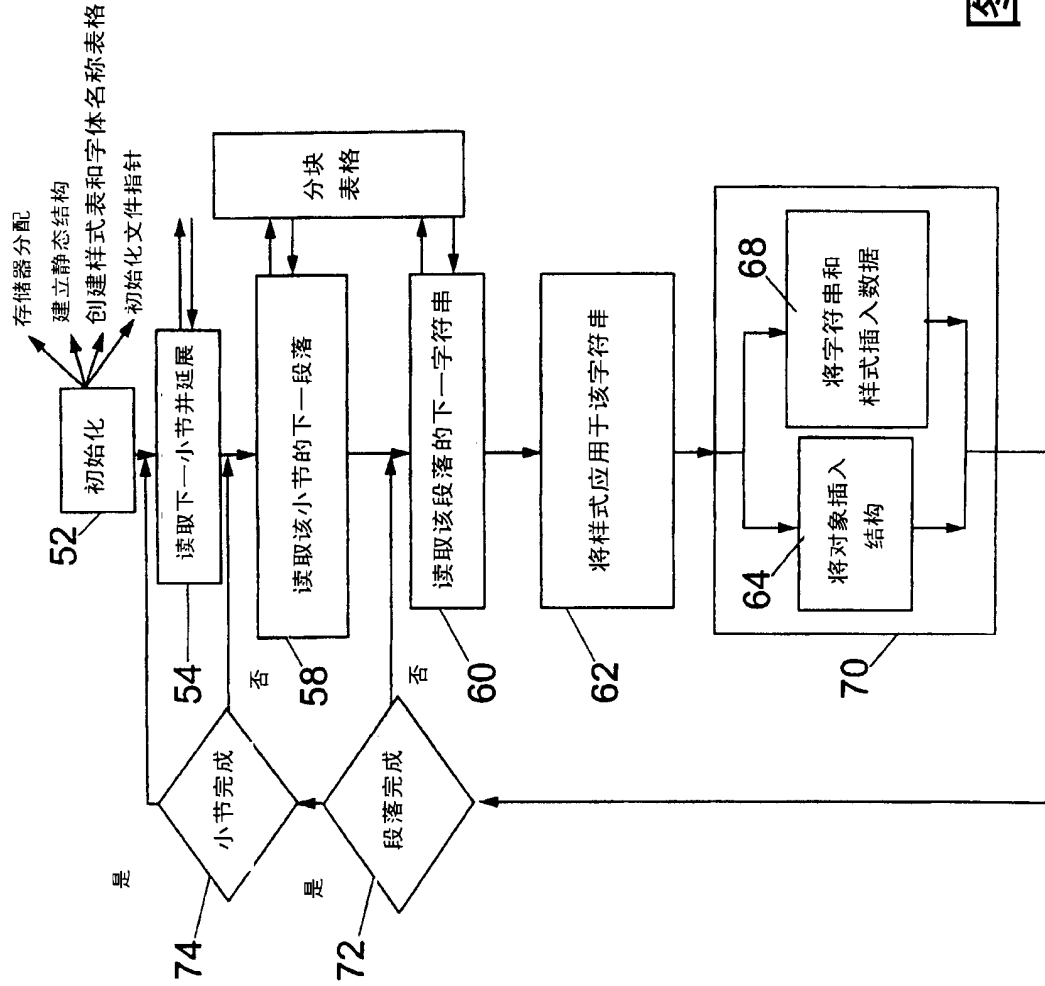


图3

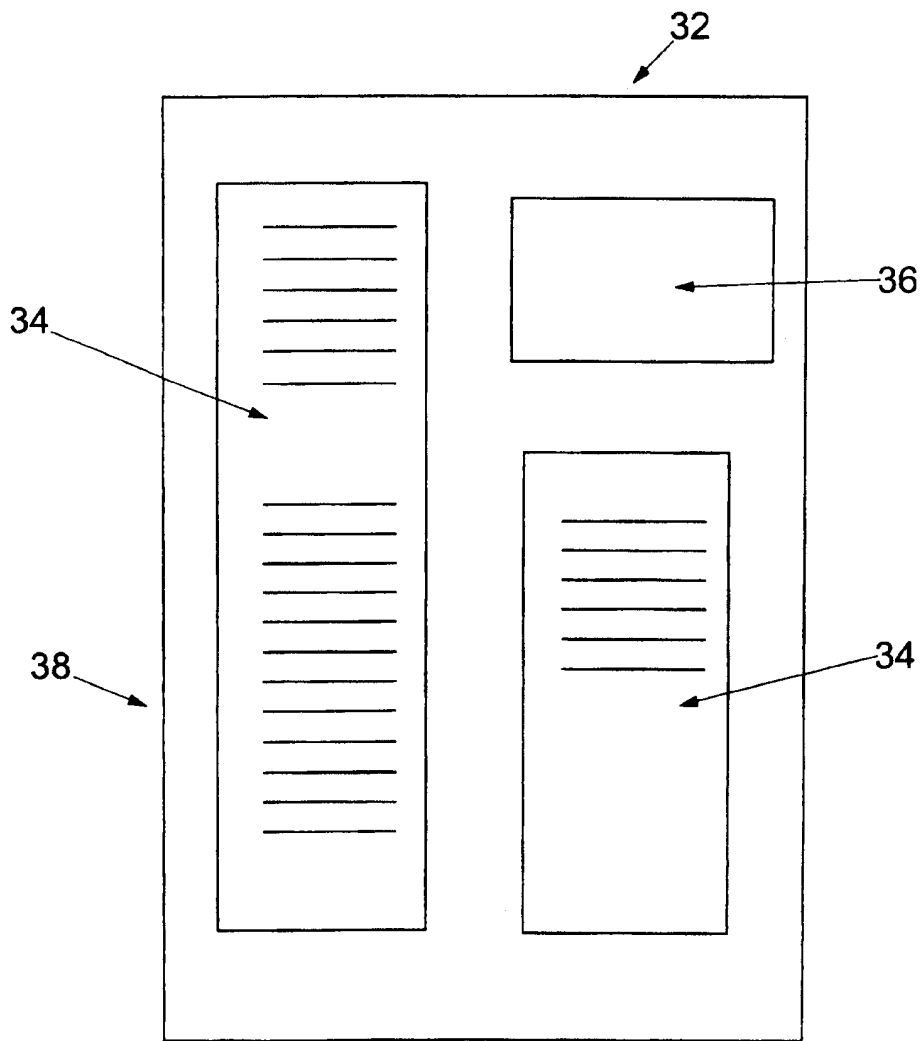


图4

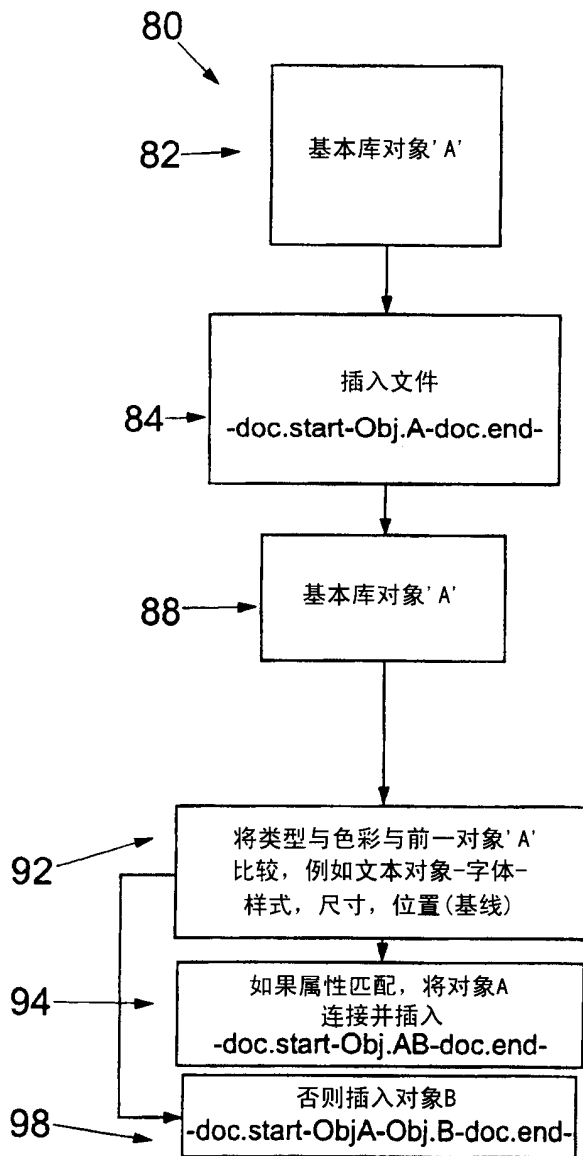


图5

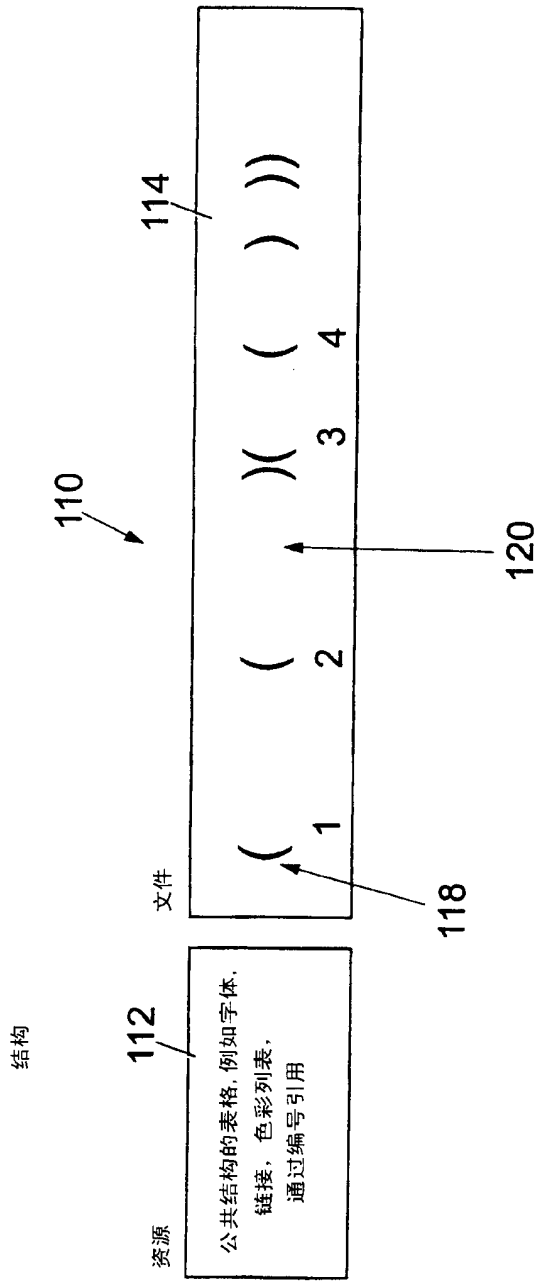


图6

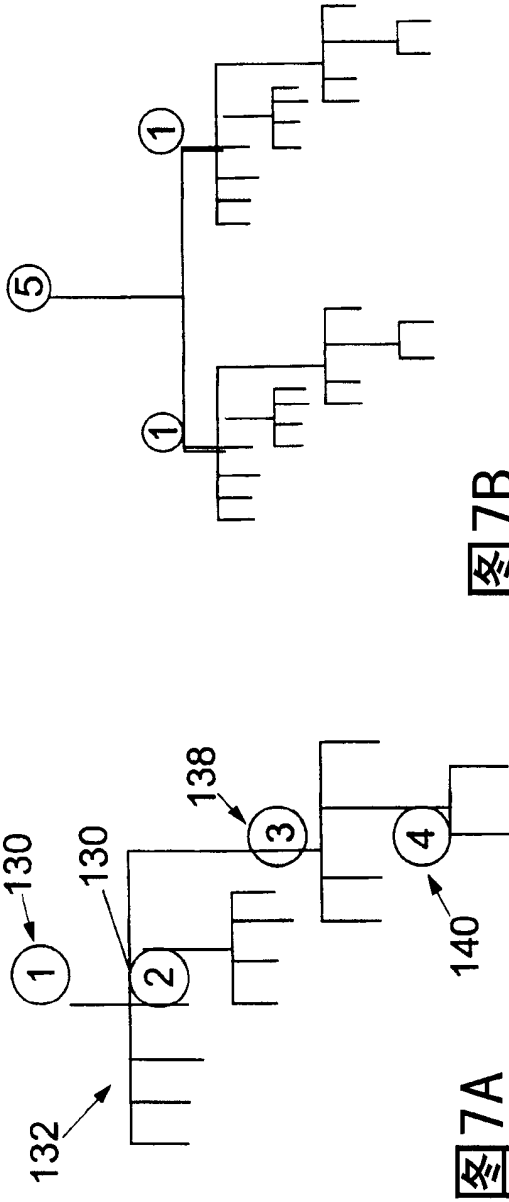


图7A

图7B

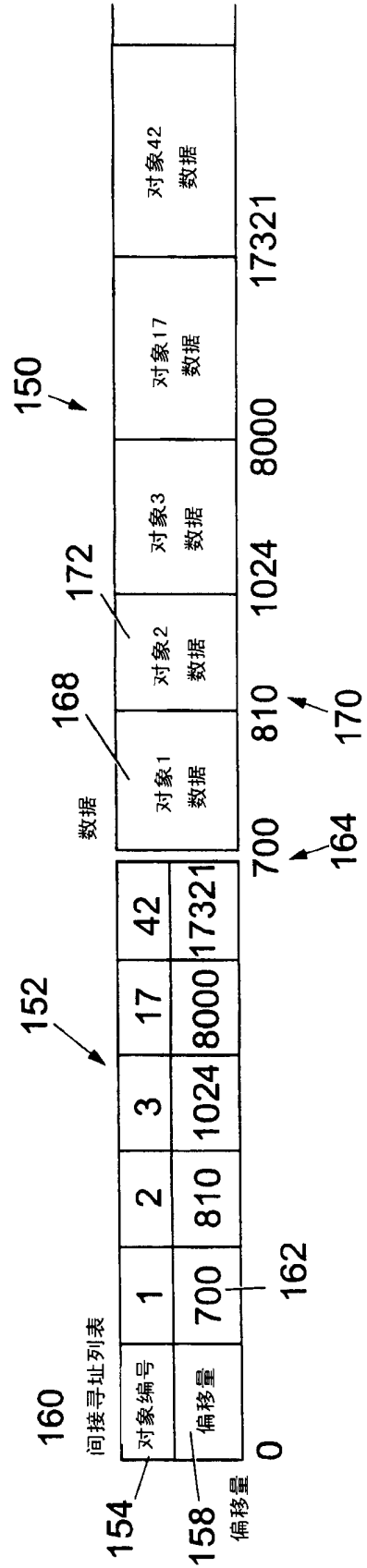


图8