

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
H04L 9/32 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200880018199.3

[43] 公开日 2010年3月24日

[11] 公开号 CN 101682512A

[22] 申请日 2008.5.30

[21] 申请号 200880018199.3

[30] 优先权

[32] 2007.5.31 [33] US [31] 11/756,688

[86] 国际申请 PCT/US2008/065337 2008.5.30

[87] 国际公布 WO2009/038828 英 2009.3.26

[85] 进入国家阶段日期 2009.11.30

[71] 申请人 真实网络公司

地址 美国华盛顿州

[72] 发明人 J·M·查森 J·D·莱特纳

[74] 专利代理机构 中国专利代理(香港)有限公司
代理人 柯广华 徐予红

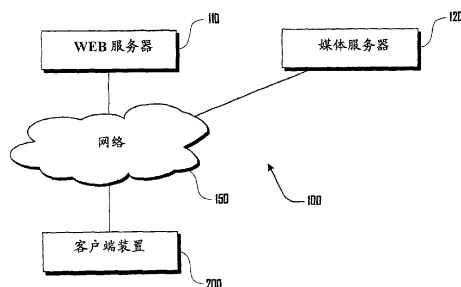
权利要求书2页 说明书13页 附图10页

[54] 发明名称

WEB 媒体资源标识系统和方法

[57] 摘要

本文提供在客户端装置上对服务器所供媒体资源进行动态标识的方法。



1. 一种对客户端装置可检索的服务器所供媒体资源进行动态标识的计算机实现方法，所述方法包括：

获得 WEB 页面，所述 WEB 页面具有至媒体资源的链接的表示；

观察与所述服务器的通信量，寻找表示媒体的通信量；

如果观察到表示媒体的通信量，则确定所述 WEB 页面中与由所述表示媒体的通信量所表明的所述媒体资源相关联的页面单元；以及

在所述客户端装置上呈现所述 WEB 页面，使得可从所述页面单元标识所述媒体资源。

2. 如权利要求 1 所述的方法，其中，通过可视指示使所述媒体资源可标识。

3. 如权利要求 1 所述的方法，其中，通过向与所述页面单元关联的菜单添加可选择项使所述媒体资源可标识。

4. 一种计算设备，包括处理器和存储器，所述存储器有用于执行权利要求 1 所述方法的可执行指令。

5. 一种计算机可读介质，包括用于执行权利要求 1 所述方法的可执行指令。

6. 一种对在线可检索媒体数据进行电子标记的呈现装置实现方法，所述方法包括：

获取超文本数据，所述超文本数据包括能够对至所述在线可检索媒体数据的链接进行标识的可呈现数据；

分析所述可呈现数据的呈现，以便对至所述在线可检索媒体数据的可能链接进行定位；

从所述可能链接标识至所述在线可检索媒体数据的链接；

确定所述可呈现数据中对定位所述媒体数据有帮助的部分；以及

用与所述在线可检索媒体数据关联的标记对所述部分进行电子标记。

7. 如权利要求 6 所述的方法, 其中, 所述标记包括用于所述媒体数据的 URI。

8. 如权利要求 7 所述的方法, 其中, 所述标记还包括所述 URI 的可用性的可视指示。

9. 如权利要求 7 所述的方法, 其中, 所述标记还包括用于链接到所述 URI 的可触发界面。

10. 如权利要求 9 所述的方法, 还包括触发所述标记。

11. 如权利要求 10 所述的方法, 其中, 触发所述标记包括访问经修补的输入表。

12. 如权利要求 6 所述的方法, 其中, 分析所述呈现包括分析所述呈现装置和远程装置之间的通信量。

13. 如权利要求 6 所述的方法, 其中, 分析所述呈现包括虚拟呈现所述可呈现数据。

14. 如权利要求 6 所述的方法, 其中, 混淆所述媒体数据。

15. 如权利要求 14 所述的方法, 其中, 所述混淆的媒体数据包括 JavaScript 数据、Flash 数据、ActiveX 数据和 VBScript 数据中的至少一种。

16. 如权利要求 6 所述的方法, 其中, 分析所述呈现包括遍历文档对象模型。

17. 如权利要求 6 所述的方法, 其中, 分析所述呈现包括定位符合至少一种预定 MIME 类型的数据对象。

18. 如权利要求 17 所述的方法, 其中, 所述预定 MIME 类型是视频 MIME 类型。

19. 一种计算设备, 包括处理器和存储器, 所述存储器具有用于执行权利要求 6 所述方法的可执行指令。

20. 一种计算机可读介质, 包括用于执行权利要求 6 所述方法的可执行指令。

WEB 媒体资源标识系统和方法

技术领域

[01] 本发明涉及在线内容，尤其涉及在所呈现（rendered）WEB 页面中标识和提供至媒体资源的下载链接的方法。

背景技术

[02] 因特网（Internet）是互联计算机网络的全世界可公开访问网络，互联计算机网络通过使用标准因特网协议（“IP”）进行分组交换来传输数据。该“网络的网络”由许多较小的国内、校园、商业和政府网络组成，这些网络一起使诸如电子邮件、在线聊天、文件传递以及互相链接的 WEB 页面与万维网（world wide web）的其它文档的多种服务成为可能。

[03] 在因特网上分发视频和音频已变得非常流行。特别是，因为宽带因特网访问已变得更常见，经常将媒体片段（clip）嵌入到或链接到 WEB 页面中。现在，随着向用户提供在线媒体的新 WEB 站点不断涌现，有大量在线可用的媒体片段。这些最流行的站点中之一是由加州 Mountain View 的 Google 公司提供的 YouTube，其特点是既有已创建媒体源制作的媒体又有小团体和业余爱好者制作的媒体。2006 年 3 月和 7 月之间，YouTube 每天的视频浏览量从 3 千万独自增长到 1 亿。

[04] 除了诸如 YouTube 的专业视频共享站点以外，许多现有的无线电和电视广播公司向因特网提供它们直播音频和视频流的“喂料”（feeds）（例如，英国广播服务）。广播公司还可允许用户对它们的观看和收听进行时间移位。

[05] 因为有这么多提供在线媒体的 WEB 站点，所以可使用诸如

计算机、游戏控制台、机顶盒、手持计算机、蜂窝电话或其它装置的因特网连接装置来按与先前可能仅通过电视和无线电接收机的相同方式访问在线媒体。

[06] 在因特网上分发媒体所用的最常见格式之一是 Flash Video 或 FLV 格式。其它常见格式包括 Windows Media Video、RealMedia、Quicktime 和 Divx。以这些格式中的多种格式、包括 Flash Video 所编码的在线媒体，可流 (stream) 到 WEB 浏览器或其它客户端以供在线观看，或者可将其下载到存储装置。许多用户因多种原因更喜欢下载媒体段而不是使其流出 (stream)：用户可能更喜欢在他或她离线的时候观看或收听媒体；用户可能更喜欢将拷贝存档使得他或她在将来即使现在提供 (host) 该媒体的 WEB 站点关闭也能够观看和收听该媒体；用户可能更喜欢与别人分享该文件本身，而不是分享至该文件的链接；或者用户可能因种种其它原因而更喜欢下载该媒体。

[07] 有很多从 WEB 站点下载媒体内容的方式。如果该站点在所呈现页面上提供至文件的直接链接，则下载媒体文件可如右击并选择 “Save Link As... (将链接存为...)” 或某些类似命令一样容易。也有用于从 WEB 页面的 HTML 源提取显式链接的众所周知方法，即使 WEB 站点没有呈现直接链接。但是，对于 WEB 站点来说，使用户难以下载媒体文件是常见惯例。一种使下载困难的技术是链接到 WEB 页面或将 “包装器” (wrapper) 嵌入到 WEB 页面，而不是直接链接到媒体片段或嵌入媒体片段。媒体包装器的常见例子包括 JavaScript 媒体播放器、Flash 媒体播放器、ActiveX 媒体播放器、VBScript 播放器等。

[08] JavaScript 是 (加州 Mountain View 的) Mozilla 基金会所实现的 ECMAScript 标准的名称，是一种以基于原型编程的概念为基础脚的语言。该语言因其在 WEB 站点中的使用 (作为客户端侧 JavaScript) 而人所共知，但也用于对至在其它应用中所嵌入的对象的访问编写脚本。

[09] Adobe Flash 或只是 Flash 是指来自于加州 San Jose 的 Adobe System 公司的 Adobe Flash Player。该 Flash Player 是在大多数常见 WEB 浏览器中可用的客户端应用。其特征是支持矢量或光栅图形、称为 ActionScript 的脚本编写语言以及音频和视频的双向流 (streaming)。还有用于移动电话和诸如因特网输入板 (Internet Tablets)、个人数字助理、来自纽约 Rochester 的 Eastman Kodak 公司的 Kodak Easyshare One 照相机以及来自日本 Sony 公司的 Playstation Portable 等的其它非 PC 装置的 Flash Player 版本。

[10] ActiveX 是一个术语,用于表示基于华盛顿州 Redmond 的 Microsoft 公司的微软组件对象模型 (Micro Component Object Model: COM) 的可重用软件组件。ActiveX 控件向程序提供经封装的可重用功能性,并且它们本质上一致但不是一直可视。示例 ActiveX 控件包括:加州 San Jose 的 Adobe Systems 公司的 Adobe Reader 和 Adobe Flash Playe、加州 Cupertino 的 Apple 公司的 QuickTime Player、华盛顿州 Redmond 的 Microsoft 公司的 Microsoft Windows Media Player、华盛顿州 Seattle 的 RealNetworks 公司的 RealPlayer、以及加州 Santa Clara 的 Sun Microsystems 公司的 Java Virtual Machine。

[11] VBScript (Visual Basic Scripting Edition 的简称) 是经由华盛顿州 Redmond 的 Microsoft 公司的 Windows® Script Host 解释的活动脚本编写 (Active Scripting) 语言。当在 Microsoft Internet Explorer 中使用, VBScript 在功能上与 JavaScript 非常类似—其处理嵌入在 HTML 中的代码。也可使用 VBScript 来创建独立的 HTML 应用 (文件扩展名为.hta)。

[12] 诸如 YouTube、Vimeo 和 Grouper 的流行 WEB 站点现在使用包装器来混淆 (obfuscate) 媒体资源。本领域中存在对被混淆在包装器之后的媒体资源进行下载的已知方法,但是现有方法在许多方面都存在不足。

[13] 一种已知方法通过将给定 WEB 页面上的内容与详细描述某

些 WEB 站点混淆媒体资源所使用的技术方法的目录进行比较，来标识媒体资源。如果所涉及的 WEB 页面由已知 WEB 站点提供，那么该方法能够标识和允许用户下载该页面上被混淆的媒体资源。例如，该目录可允许用户推断出 WEB 页面 <http://hiddenvids.com/video/foo> 将包含位于 <http://secret.hiddenvids.com/video/foo.mpg> 的视频资源。该方法可在客户端上、例如作为浏览器插件（plug-in）实现，或者它可使用代理服务器来实现，代理服务器用作客户端和目标 WEB 站点之间的媒介。

[14] 这种方法的缺点涉及以下事实：它要求对每个单独 WEB 站点编目，并要求连续监控每个单独目录条目使其可在 WEB 站点操作者改变其混淆策略时被更新。例如，该方法将无法发现新 WEB 站点上的媒体资源，直到该站点的混淆方案经过分析并被输入到目录中为止。另外，如果 WEB 站点操作者以甚至诸如改变存储该媒体资源的主机的名称的简单方式来改变其混淆方案，该方法也将无法标识甚至在已知 WEB 站点上的媒体资源。

[15] 另一种已知方法是“流切开”（stream ripping）。使用这种方法，客户端截取与流媒体资源对应的数据流，允许用户将该数据重定向到存储介质上的文件。该方法的一个缺点是媒体资源可实时流出，这意味着将花高达 30 分钟去捕捉 30 分钟的媒体资源。

发明内容

附图说明

[16] 图 1 是在根据一个实施例的网络中的若干装置的系统图。

[17] 图 2 是提供用于一个实施例的示范操作环境的设备的框图；

[18] 图 3 是根据一个实施例示出多种装置之间的关系的流程图；

[19] 图 4 是根据一个实施例示出媒体资源标识例程的流程图；

[20] 图 5 是根据一个实施例示出媒体资源页面单元关联子例程的流程图;

[21] 图 6 是根据一个实施例的 WEB 页面的示图, 其中媒体资源通过可视指示标记;

[22] 图 7 是根据一个实施例的 WEB 页面的示图, 其中媒体资源通过可选择菜单标识;

[23] 图 8 是根据一个实施例的 WEB 页面的示图, 其中流媒体资源可通过可选择单元下载;

[24] 图 9 是根据一个实施例的 WEB 页面的示图, 其中流媒体资源被标识为不可下载;

[25] 图 10 是根据一个实施例的具有流媒体资源和同时下载的 WEB 页面的示图。

具体实施方式

[26] 主要用过程和常规计算机组件操作的符号表示来表示后面的详细描述, 计算机组件包括处理器、用于处理器的存储器存储装置、相连的显示装置和输入装置。此外, 这些过程和操作可采用不同类型的分布式计算环境中的常规计算机组件, 包括远程文件服务器、计算机服务器和存储器存储装置。这些常规分布式计算组件中的每一个可由处理器通过通信网络访问。

[27] 现在详细参考如附图所示的实施例的描述。虽然结合附图和有关说明来描述实施例, 但并无将范围限制到这里公开的实施例的意图。相反, 意图是覆盖所有替代、更改和等同。在备选实施例中, 可添加或组合附加装置或示出装置的组合, 而不将范围限制到这里公开的实施例。

[28] 用户希望能够即便从采取措施混淆媒体资源的 WEB 站点下载媒体内容。这里描述的实施例通过提供媒体资源标识例程来克服现有方法中的缺陷, 媒体资源标识例程可分析浏览器和 WEB 服务器之

间的通信以标识被混淆或未被混淆的媒体资源并将媒体资源与特定 WEB 页面单元相关联。

[29] 图 1 示出示例操作环境 100，其中客户端装置 200（在图 2 中示出并在下文描述）使用网络 150（例如因特网等）来访问 WEB 页面，所述 WEB 页面在 WEB 服务器 110 上提供并包含对在媒体服务器 120 上提供的媒体资源的引用，媒体服务器 120 在网络 150 上也是可到达的。备选实施例中，WEB 服务器 110 和媒体服务器 120 可为单个装置，或客户端装置 200 和 WEB 服务器 110 可为单个装置。另外，多个 WEB 服务器 110 都可访问单个媒体服务器 120；单个 WEB 服务器 110 可访问多个媒体服务器 120；多个 WEB 服务器 110 可访问多个媒体服务器 120；或它们的任何其它组合。多个客户端装置 200 也可存在。

[30] 图 2 示出示范客户端装置 200 的几个组件。在某些实施例中，客户端装置 200 可包括比图 2 中所示的那些组件多得多的组件。但是，不需要示出所有这些一般的常规组件来公开说明性实施例。如图 2 中所示，客户端装置 200 包括用于与网络 150 连接的网络接口 230。本领域技术人员将理解，网络接口 230 包括用于此类连接的必要电路，并构建成与合适协议配合使用。

[31] 客户端装置 200 还包括处理单元 210、存储器 250，并可包括可选的显示器 240，所有装置通过总线 220 与网络接口 230 互连在一起。存储器 250 通常包括随机存取存储器（“RAM”）、只读存储器（“ROM”）、以及诸如磁盘驱动器或闪存驱动器（flash drive）的永久大容量存储装置。存储器 250 存储用于 WEB 浏览器 260、媒体播放器 265 以及媒体资源标识例程 270 的程序代码。另外，存储器 250 还存储操作系统 255。要理解，可通过使用与诸如软盘、磁带、DVD/CD-ROM 驱动器、存储卡的计算机可读介质相关联的输入机构（未示出）、通过网络接口 230 等，将这些软件组件从计算机可读介质加载到客户端装置 200 的存储器 250 中。

[32] 虽然已经描述了通常符合常规通用计算装置的示范客户端装置 200，但本领域普通技术人员将理解，客户端装置 200 可为能够与网络 150 通信的大量装置中的任一种装置，例如个人计算机、游戏控制台、机顶盒、手持计算机、蜂窝电话、或能够访问在线媒体并能够运行媒体资源标识例程 400 且显示 WEB 页面的任何其它装置。

[33] 图 3 描述示例实施例中客户端装置 200、WEB 服务器 110 和媒体服务器 120 之间的关系以及它们执行的动作。

[34] 最初，用户可访问 (visit) 包含一个或多个媒体资源的 WEB 页面。在用户在浏览器 260 中导航到该页面之后，客户端装置 200 向合适的 WEB 服务器 110 请求 305 该页面。接收到该请求 305 时，WEB 服务器 110 查找所请求页面在其存储器中的位置 310，和/或另外处理该请求 305 使得其能响应 315 客户端装置的请求 305。WEB 服务器 110 通过至少传输客户端装置 200 用于在浏览器中呈现 320 该页面的初始数据组来响应 315 客户端装置的请求 305。然后，客户端装置 200 开始呈现 320 所请求的页面。

[35] 呈现 320 该页面通常可包括：客户端装置 200 向 WEB 服务器 110 请求 325 附加数据。例如，如果所请求的页面包括诸如图像的媒体资源或经混淆的媒体资源，则客户端可向 WEB 服务器 110 对允许浏览器 260 呈现该页面的附加数据进行附加请求 325。收到这些附加请求时，WEB 服务器 110 将处理 330 这些请求，并且如果所请求的数据被定位于媒体服务器 120 上，则 WEB 服务器 110 将向媒体服务器 120 请求 335 该数据。然后，WEB 服务器 110 将请求的信息传回 340 客户端装置 200。

[36] 在该呈现过程期间，媒体资源标识例程 400 (在图 4 中示出并在下文描述) 能够解释 345 所呈现的页面，并分析客户端装置 200 和 WEB 服务器 110 之间的通信以标识 350 媒体资源。如果发现了媒体资源，则客户端装置 200 将向所呈现的 WEB 页面添加 355 用于媒体资源的控制或标记。

[37] 刚才描述的在客户端装置 200 上进行的媒体资源标识过程在图 4 中详细示出，图 4 示出媒体资源标识例程 400 的示范流程图。例程 400 开始于框 405，在那里浏览器 260 请求可包含一个或多个媒体资源的 WEB 页面。框 410 中，接收到所请求的 WEB 页面，且在循环框 (looping block) 415 中，例程 400 开始分析可呈现的每个页面单元。页面单元可用 DOM (文档对象模型—用于表示 HTML 或 XML 及相关格式的平台无关和语言无关标准对象模型) 表示，在这种情况下，媒体资源标识例程 400 将通过遍历 (traverse) DOM 来操作。在备选方法中，页面单元可通过某种别的合适方式被迭代 (iterate)。

[38] 框 420 中，例程 400 分析客户端装置 200 和 WEB 服务器 110 之间的用于呈现所请求页面的通信量，寻找表示媒体 (media-indicative) 的通信量。在判定框 425 中，例程 400 确定在分析的通信量中是否发现媒体资源。可通过检查单元的多用途因特网邮件扩展 (“MIME”) 类型来确定该单元是否具有将由媒体播放器进行处理类型来标识媒体资源。MIME 是其中包括登记内容类型并将其与处理程序 (handler) 应用相关联的方法的因特网标准。在 RFC 2045 中详述了 MIME 的细节，将其整体合并于此。

[39] 若判定框 425 中未发现媒体资源，则处理循环 (cycle) 回循环框 415，在那里分析下一可呈现页面单元。如果在判定框 425 中定位了媒体资源，则于例程框 500 (在图 5 中示出并在下文描述) 执行附加处理以将可呈现页面单元与媒体资源关联。

[40] 接着，在框 430 中，例程 400 存储已标识媒体资源的位置及其已确定的关联页面单元。如果有要分析的附加可呈现页面单元，则循环框 435 将处理循环回循环框 415；如果没有，则循环框 435 将处理指引到框 440。

[41] 在循环框 440 中开始，处理所有存储于框 430 中的媒体资源。在框 450 中，例程 400 添加与关联于所标识媒体资源的已存储页面单元相关联的控制或“标记”。在某些情况下，这种标记可以是将与页

面单元一起呈现的可视指示，可视指示例如为图像或文本链接。该链接可包含对媒体资源进行标识或命名的统一资源标识符（“URI”）。其它情况下，这种标记可包括添加可选择项到与该页面单元关联的菜单（例如，“右击”菜单）。

[42] 添加命令到此类菜单可包括：“钩挂”（hooking）到浏览器的 API。应用编程接口（“API”）是计算机系统或程序库提供来通过计算机程序支持对要由其提供的服务的请求的源代码接口。“钩挂”是一种采用所谓的“钩子”（hooks）来使流程链成为处理程序（handler）的技术。因此，在经处理的事件发生后，控制流以特定顺序遵循该链。新的钩子将其自身的地址登记为用于该事件的处理程序，并预期在某点、通常在结尾调用原始处理程序。“输入表修补”（Import Table Patching）是一种可用于某些实施例的 API 钩挂的合适方法。在来自华盛顿州 Redmond 的 Microsoft 公司的 Windows® 操作系统中，大多数应用和模块具有所谓的“输入表”，其可包括应用和模块调用的 API 列表。其它操作环境可包括类似技术，以用于钩挂到适合于呈现页面单元的应用。

[43] 如果有附加的所存储页面单元，则循环框 455 将处理循环回循环框 440；如果没有，则循环框 455 将处理指引到框 460。框 460 在浏览器 260 中呈现该页面的描绘，且处理在框 499 结束。

[44] 媒体资源标识例程 400 也可适合按照预期标识在当前 WEB 页面提供至其链接的 WEB 页面上的可呈现媒体资源。例如，WEB 页面可包含至包含可呈现媒体资源的 WEB 页面的链接的列表。通过使媒体资源标识例程 400 适合从每个所链接页面检索（retrieve）数据并“虚拟地”（virtually）呈现该数据（处理该数据，好像将其呈现而不向用户提供结果）以便标识其中包含的媒体资源，可方便地向用户提供可允许用户从许多页面下载媒体资源而不必访问每个单独页面的控制、以及可降低在给用户导航到的页面上的媒体资源添加控制时的任意延迟的控制。

[45] 在混合方法中，另一个备选实施例可仅在已分析和呈现了当前 WEB 页面之后、即在浏览器的空闲时间期间才处理到其它 WEB 页面的链接。

[46] 图 5 示出子例程 500 的示范流程图，子例程 500 确定哪个可呈现页面单元与已标识媒体资源关联。当可呈现页面单元与媒体资源的关联不清楚时，可采用子例程 500。例如，Flash 媒体资源可由多个音频片段、视频片段、图像或其它媒体段组成。在一个示范情形中，Flash 媒体资源可包括可以是广告的第一视频片段以及作为用户想看的视频段的第二视频片段。在循环框 505 中，子例程 500 开始在与已标识媒体资源关联的单元树的每个页面单元上进行迭代。在框 510 中，检查当前的页面单元，使得在判定框 515 中可确定该页面单元是否是被呈现的那个。如果该页面单元不将被呈现（例如，因为该页面单元不可见或仅为隐藏链接），那么处理被指引到循环框 530，其将要检查的页面单元设置为下一个页面单元并将处理循环回循环框 505，使得可检查下一个页面单元。如果判定框 515 发现要被呈现的页面单元，则框 525 在循环框 530 将处理返回到循环框 505 之前存储该页面单元，使得可检查下一页面单元。

[47] 一旦已检查所有页面单元，则在框 535 中，子例程 500 确定是否已存储要被呈现的页面单元。倘若如此，则将处理指引到返回框 599，其向调用例程返回存储的页面单元。如果框 505-530 没有标识和存储要被呈现的页面单元，则将处理指引到框 540。在框 540 中，子例程 500 通过检查其特征来确定哪些页面单元将被呈现。例如，页面单元的大小可表明其可能是将被呈现的那个页面单元。框 545 存储被标识为可能会被呈现的页面单元。在返回框 599 中，该已存储页面单元被返回到媒体资源标识例程 400。

[48] 图 6 根据一个实施例示出在浏览器 260 中显示的示范 WEB 页面 605。已确定页面单元 610A-B 与媒体资源相关联。本实施例中，使用可视界面单元 615A-B 来标记已确定与已标识媒体资源关联的页

面单元 610A-B。在某些实施例中，可视标记可包括用于其各自的关联媒体资源的 URI。可视标记还可包括用户可用来触发诸如打开或链接到媒体资源的 URI 等的动作的链接或类似界面。在某些实施例中，可视指示标记 615A-B 可包括放置在已确定页面单元 610A-B 上的图像。不仅可视指示标记 615A-B 可通知用户已标识和确定媒体资源与特定页面单元关联，而且可视指示标记 615A-B 也可通知用户媒体资源当前是否可用。例如，如果已标识和确定媒体资源与特定 URI 关联，则该可视指示标记 615A-B 在该 URI 当前不可访问时可为红色图像、或者在该 URI 当前可访问时为绿色图像。在备选实施例中，可视指示标记可包括给与媒体资源关联的文本所添加的文本链接 620A-B，文本例如为媒体资源内容的文本描述、媒体资源的名称等。

[49] 图 7 根据另一个实施例示出在浏览器 260 中显示的示范 WEB 页面 605。已确定页面单元 610A-B 与媒体资源关联。在本实施例中，通过与页面单元 610A-B 关联的可选择菜单 710 来标识媒体资源。菜单 710 可包括一个或多个可选命令 715 的集合，集合中的每一个在被选择时可触发一个或多个任务，例如将某些或所有媒体资源保存到存储装置、对至某些或所有媒体资源的直接链接进行复制或发送等。在某些实施例中，用户可通过访问如上所述的已被修补成允许访问所述可选择菜单 710 的输入表来激活可选择菜单 710。

[50] 图 8 根据另一个实施例示出了浏览器 260 中显示的示范 WEB 页面 805。在一个示范实施例中，页面单元 815 是 WEB 页面 805 的显示部分 810 内的流媒体资源。本实施例中，将该流媒体资源 815 标识为具有用户界面组件 820 的页面单元，用户界面组件 820 有触发媒体资源下载（与该媒体资源的流分开，但与该媒体资源的流可能同时进行）的可选择动作单元 825。用户界面组件 820 可包括一个或多个可选择动作单元 825 的集合，集合中的每一个在被选择时可触发一个或多个任务，例如将某些或所有媒体资源保存到存储装置、对至某些或所有媒体资源的直接链接进行复制或发送等。在某些实施例中，

用户可通过访问如上所述的已被修补成允许访问所述可选择动作单元 825 的输入表来激活可选择动作单元 825。

[51] 图 9 根据另一个实施例示出在浏览器 260 中显示的示范 WEB 页面 905。在一个示范实施例中，页面单元 915 是 WEB 页面 905 的显示部分 910 内的流媒体资源 915。本实施例中，可将流媒体资源 915 标识为可下载的页面单元，但是本实施例中，已表明该流媒体资源 915 为不可下载。因此，在此类实施例中，显示用户界面组件 920 表明该媒体资源不可下载。多种备选实施例可采用各种不可下载性指示。例如，某些数字电视协议允许“广播视频标志”以指示内容不应被下载。也可采用其它系统、例如数字版权管理 (“DRM”) 系统或内容加密系统达到类似的效果。某些实施例中，WEB 页面 905 可包含其页面单元不应被下载的指示（例如通过具有保留权利的版权告示、通过页面中的元标记 (meta-tags) 等)。

[52] 图 10 根据另一个实施例示出在浏览器 260 中显示的示范 WEB 页面 805。如上所述，在一个示范实施例中，页面单元 815 是 WEB 页面 805 的显示部分 810 内的流媒体资源。在本实施例中，将该流媒体资源 815 标识为具有用户界面组件 820 的页面单元，用户界面组件 820 有触发媒体资源下载的可选择动作单元 825。另外，在图 10 中，下载状态组件 1010 在进程单元 1015A-C 上显示多个媒体资源下载的当前进程。在多个实施例中，下载可与流媒体资源 815 的回放同时进行，可导致流媒体资源回放被暂停或甚至被取消。在更进一步的实施例中，可限制多个下载使得它们排队等候。例如，如果客户端装置 200 没有足够的下载带宽，或被限制到预定的下载数目，则选择可选择动作单元 825 可让下载排队，以便在以后客户端（或服务器）不再受限时的时刻进行。

[53] 虽然这里已示出和描述特定的实施例，但本领域普通技术人员要理解，在不脱离本发明的范围的情况下，显示和描述的特定实施例可被多种备选和/或等同的实现替代。该申请旨在覆盖这里讨论的实

施例的任何适配或变型。

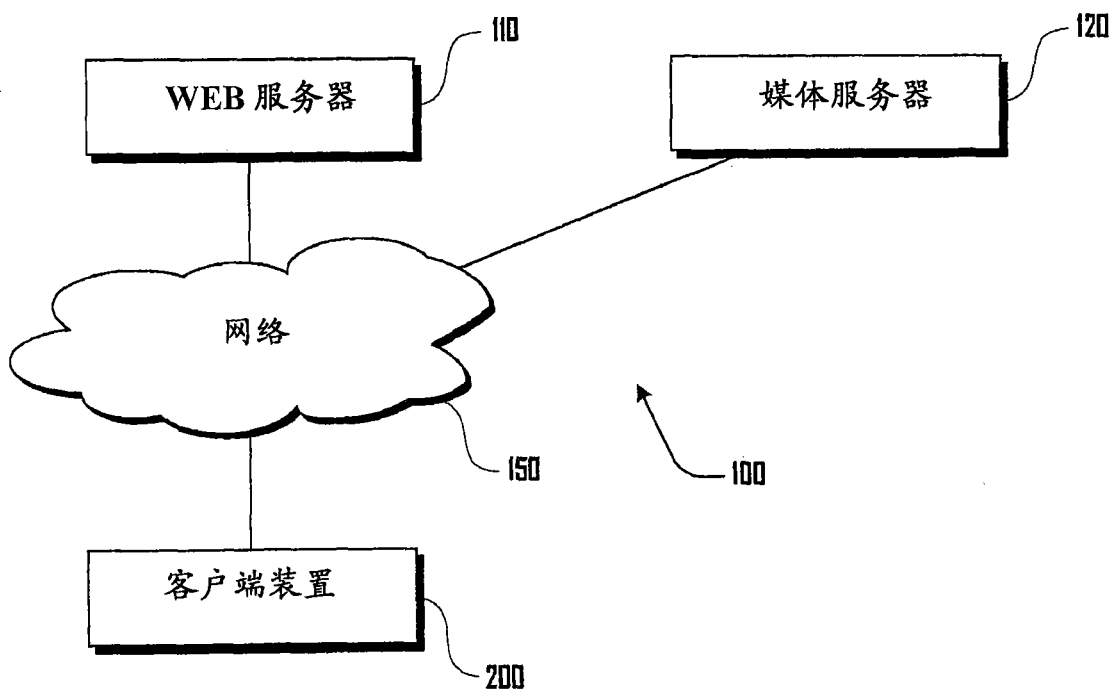


图 1

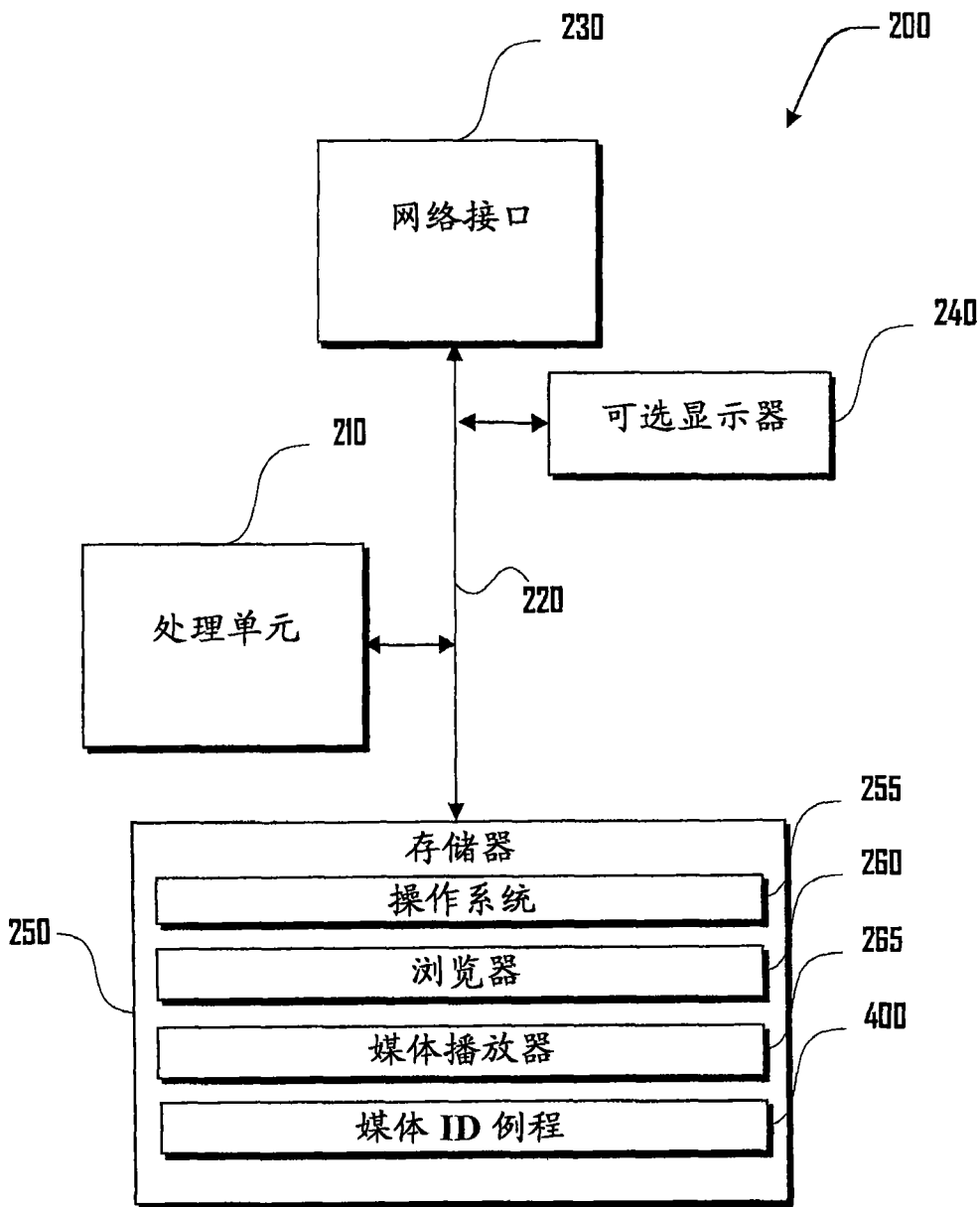


图 2

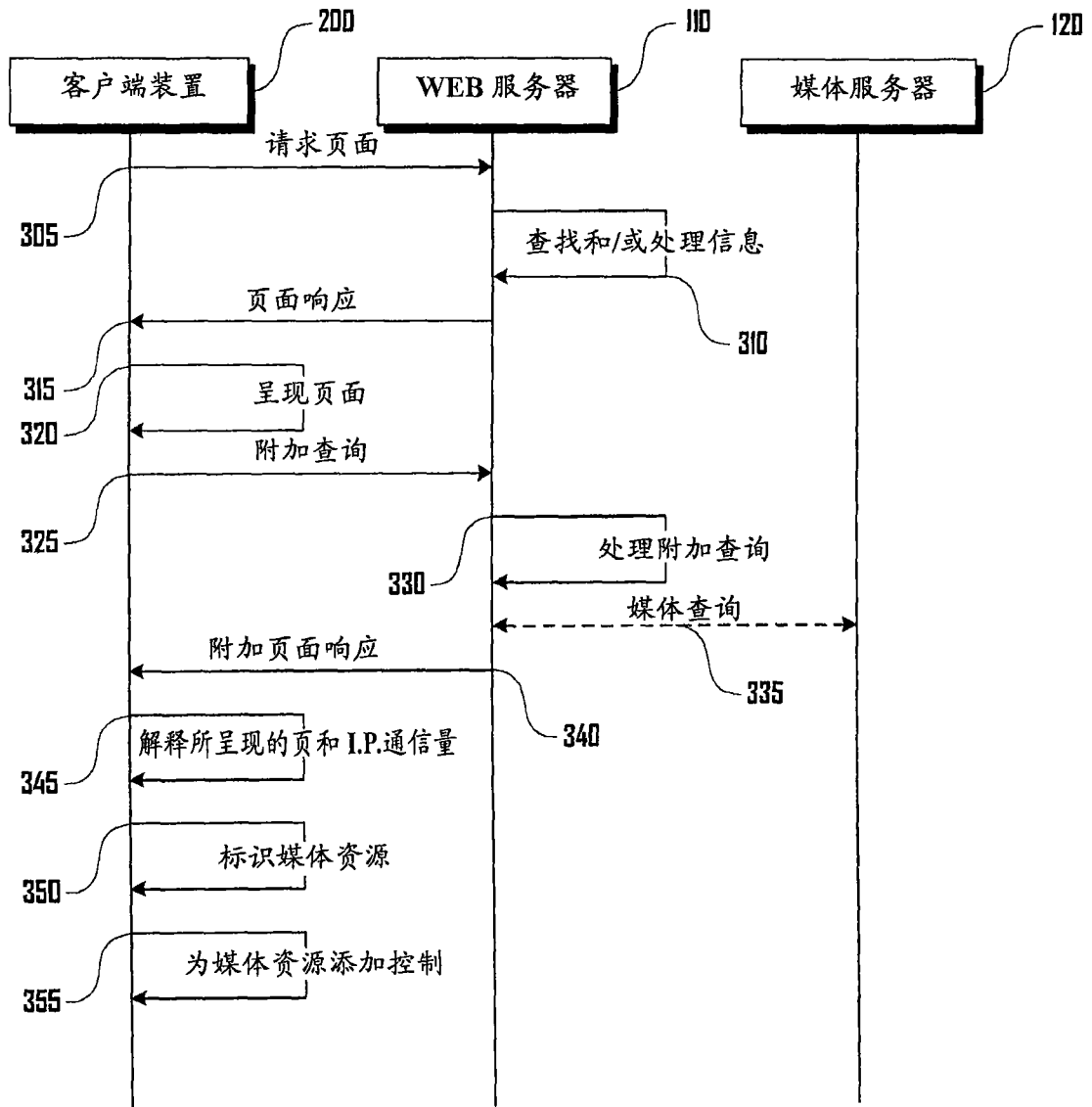


图 3

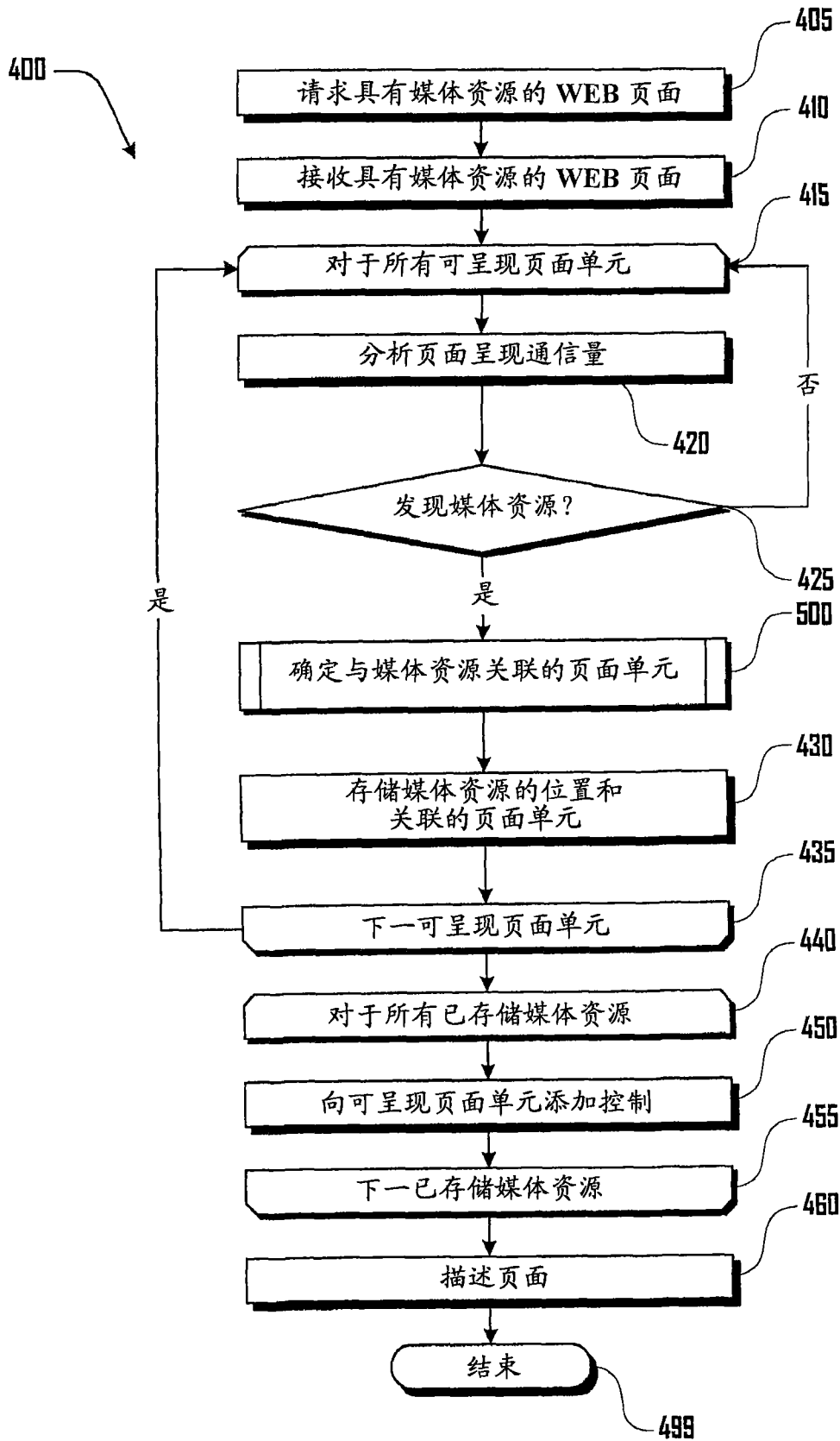


图 4

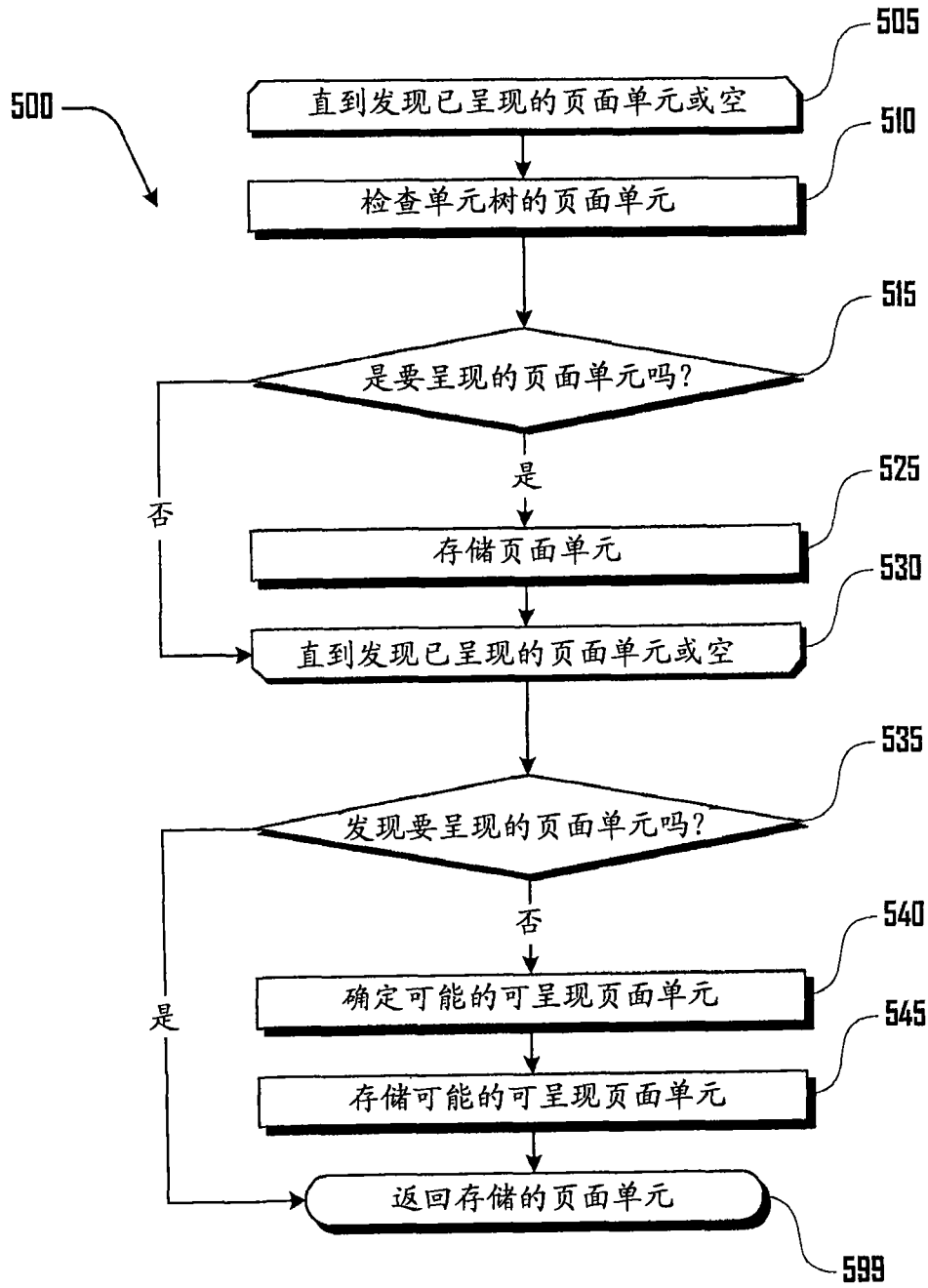


图 5

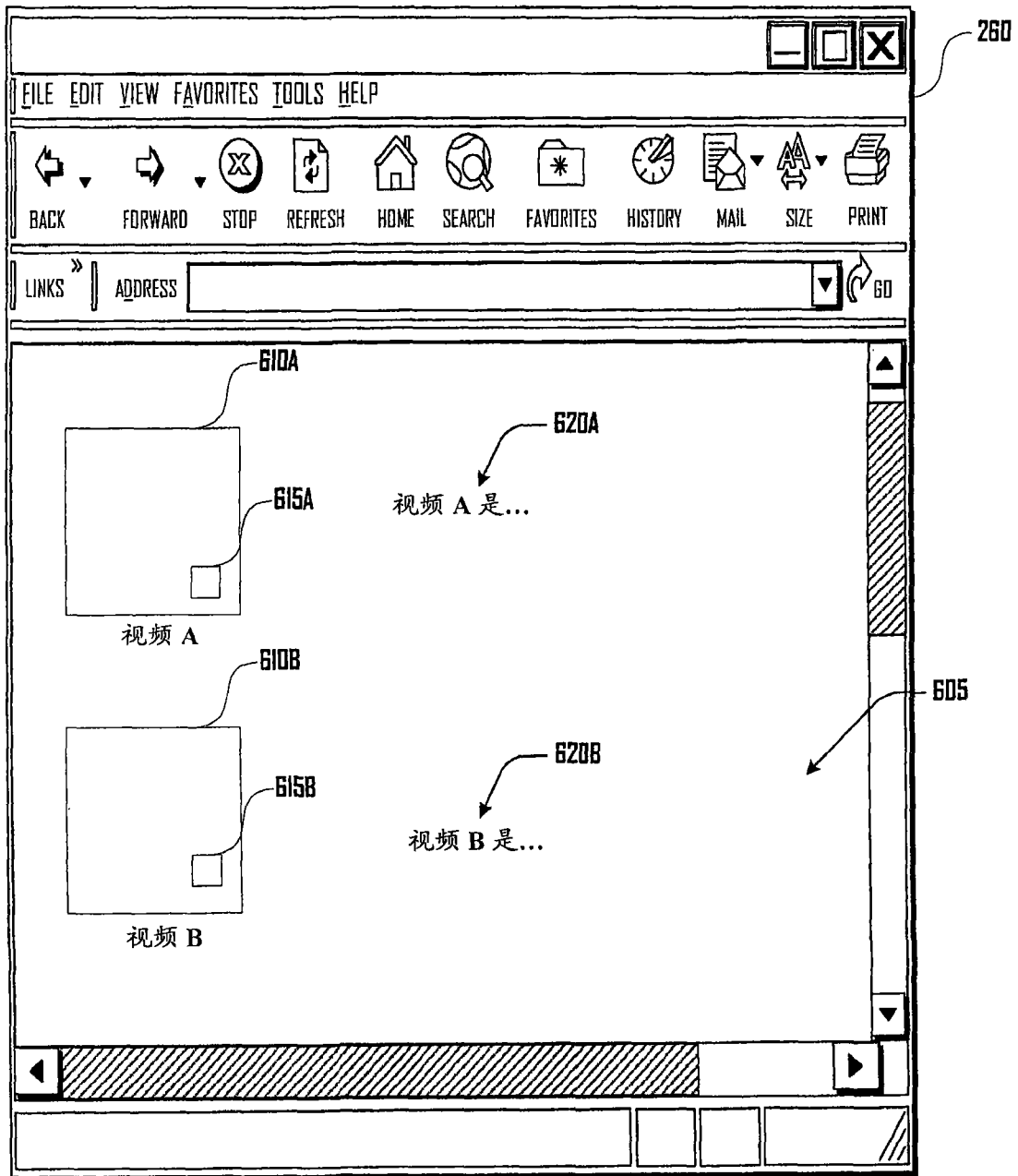


图 6

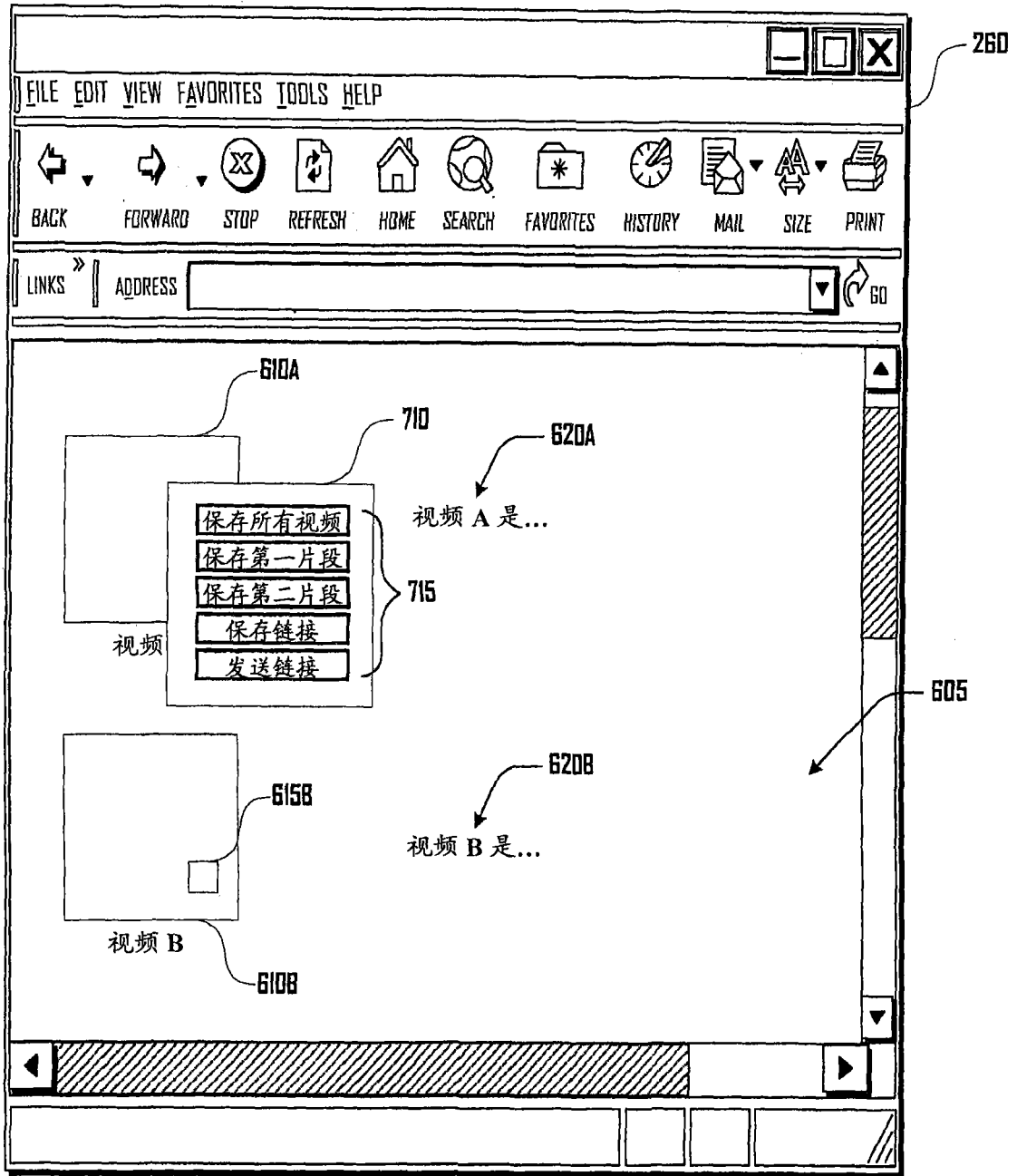


图 7

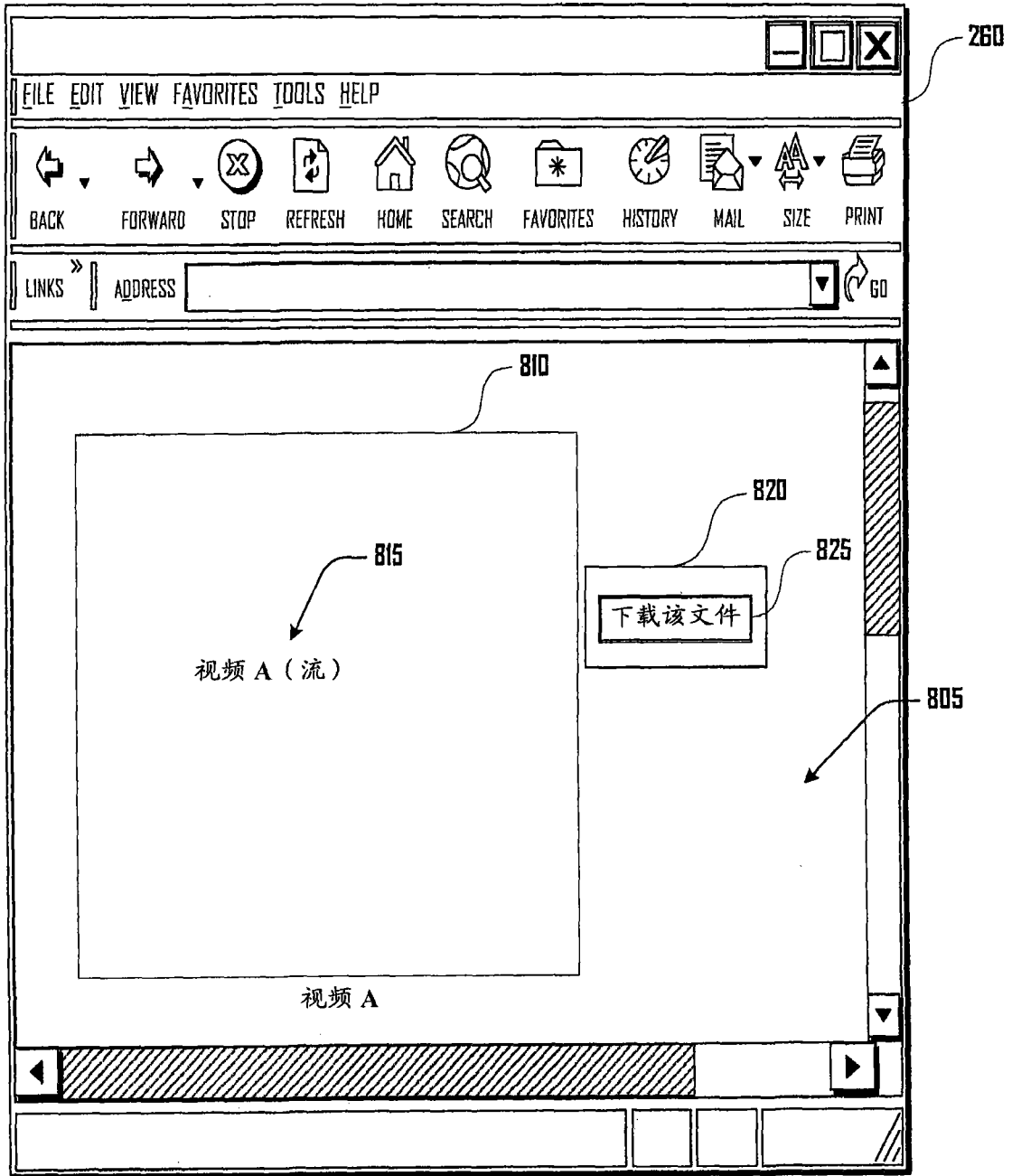


图 8

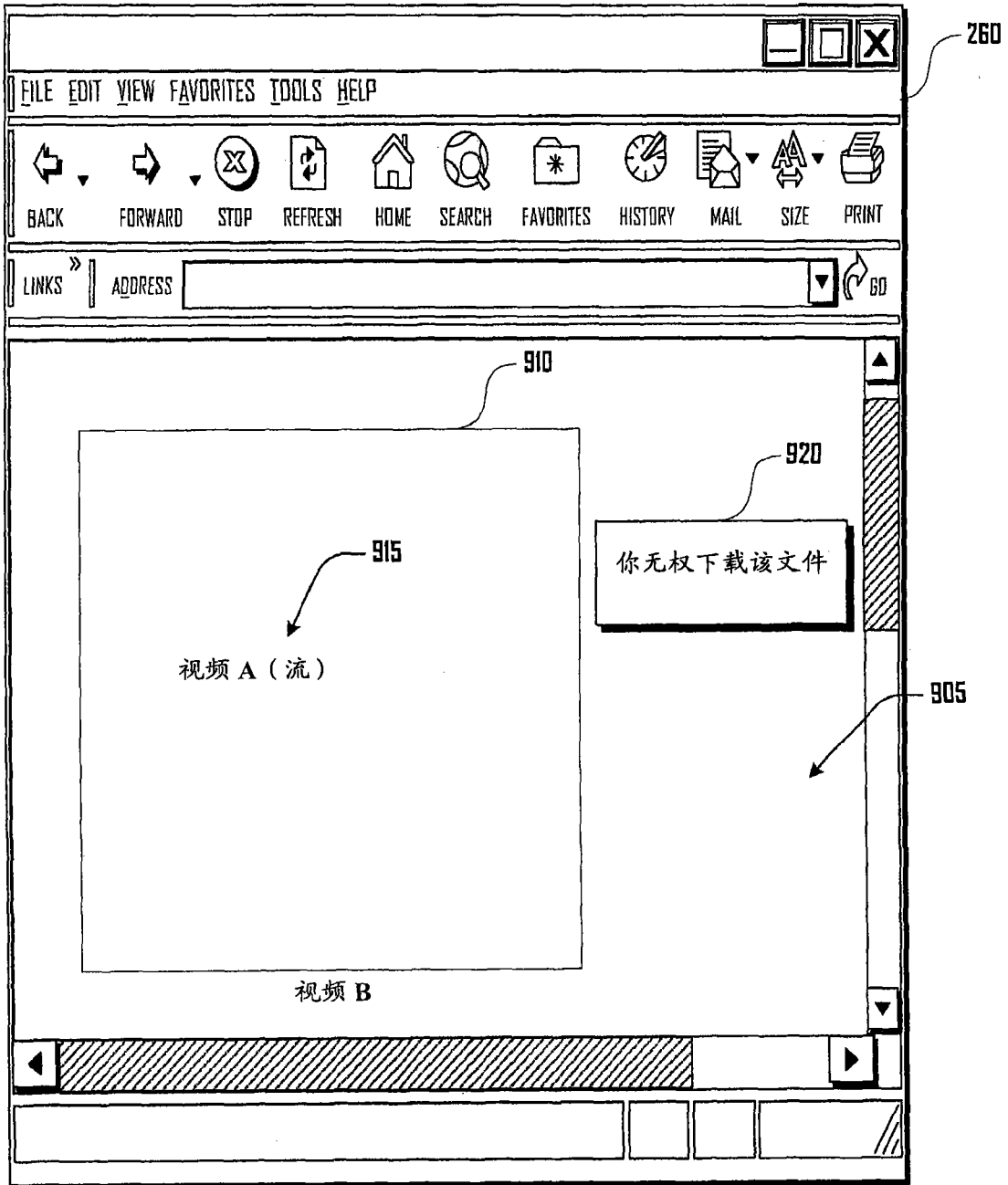


图 9

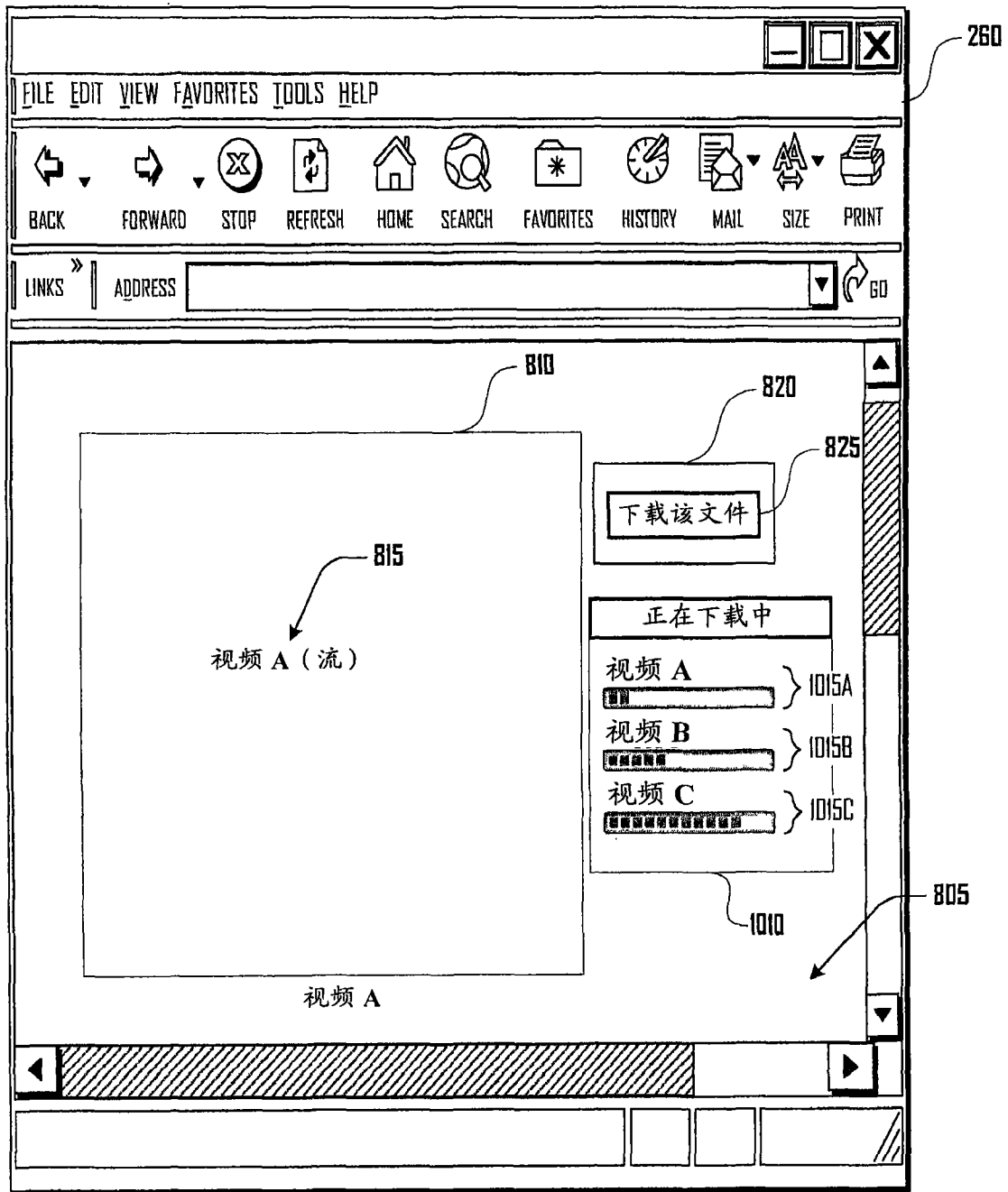


图 10