

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 15/00 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200780015783.9

[43] 公开日 2009年5月13日

[11] 公开号 CN 101432711A

[22] 申请日 2007.3.7

[21] 申请号 200780015783.9

[30] 优先权

[32] 2006.5.18 [33] US [31] 11/437,853

[86] 国际申请 PCT/US2007/063460 2007.3.7

[87] 国际公布 WO2007/136908 英 2007.11.29

[85] 进入国家阶段日期 2008.10.31

[71] 申请人 伏尔甘·波塔尔斯公司

地址 美国华盛顿

[72] 发明人 D·巴内

[74] 专利代理机构 永新专利商标代理有限公司
代理人 王 英

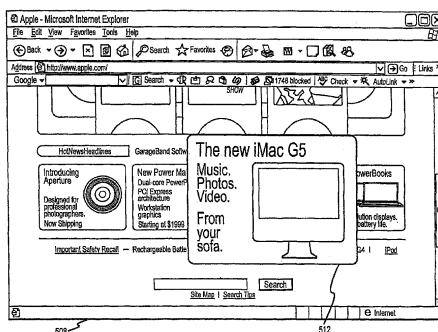
权利要求书5页 说明书12页 附图8页

[54] 发明名称

用于选择性显示部分显示屏的用户界面系统和方法

[57] 摘要

一种计算机系统或计算设备包括显示器，用于显示任意数量的软件应用程序的可视输出。在该计算机系统或计算设备上执行一种选择性显示显示屏的一部分的放大视图的计算机实现方法。该方法允许用户选择显示屏的一部分用于放大，然后显示显示屏的该部分的放大视图。所放大的视图保留下面的未放大的源内容的功能和交互特性。该方法还提供了用于控制放大视图中放大量的可配置手段。该方法允许使用放大视图来移动显示下面的未放大的源内容。



1、一种用于根据放大信号选择性放大显示屏一部分的计算机实现方法，所述方法包括：

确定显示上下文；

确定所述显示上下文的放大图像在所述显示屏上的位置，其中所述位置与在所述显示屏上选择的定位相关；

确定所述放大图像的尺寸和放大系数；以及

利用所确定的位置、所确定的尺寸和所确定的放大系数来显示所述放大图像，其中所述放大图像具有所述显示上下文的功能特性。

2、如权利要求1所述的方法还包括：

在显示所述放大图像时，检测第一信号；

根据检测的所述第一信号，增加用于所述放大图像的放大系数并且利用所确定的尺寸、所确定的位置和所述增加的放大系数重新显示所述放大图像。

3、如权利要求2所述的方法还包括：

在显示所述放大图像时，检测第二信号；

根据检测的所述第二信号，减小用于所述放大图像的放大系数并且利用所确定的尺寸、所确定的位置和所述减小的放大系数重新显示所述放大图像。

4、如权利要求3所述的方法还包括：

在显示所述放大图像时，检测第三信号；

根据检测的所述第三信号，终止显示所述放大图像。

5、如权利要求4所述的方法还包括：

在显示所述放大图像时，检测显示状态的改变，以提供新的显示状态；根据检测的所述显示状态的改变，按照新的显示状态重新确定用于所

述放大图像的尺寸、位置和放大系数，并且重新显示所述放大图像。

6、如权利要求 1 所述的方法，其中所述确定显示上下文的步骤包括：
使用指示设备在示出所确定的显示上下文的所述显示屏的一部分上放置指针。

7、如权利要求 1 所述的方法，其中所述确定所述显示上下文的放大图像在所述显示屏上的位置的步骤包括：

确定在所述显示屏上所述显示上下文的定位；以及
使用所述显示上下文的位置作为在所述显示屏上所选择的定位。

8、一种用于根据放大信号选择性放大显示屏一部分的系统，所述系统包括：

处理部件；

显示屏，耦合至所述处理部件；

数据存储部件，耦合至所述处理部件并且存储由所述处理部件执行的放大程序，所述放大程序用于：

确定显示上下文；

确定所述显示上下文的放大图像在所述显示屏上的位置，其中所述位置与在显示屏上的指针定位相关；

确定用于所述放大图像的尺寸和放大系数；

利用所确定的位置、所确定的尺寸和所确定的放大系数显示所述放大图像，其中所述放大图像具有所述显示上下文的功能特性。

9、如权利要求 8 所述的系统，其中由所述处理部件执行的所述放大程序还用于：

在显示所述放大图像时，检测第一信号；

根据检测的所述第一信号，增加用于所述放大图像的放大系数并且利用所确定的尺寸、所确定的位置和所述增加的放大系数重新显示所述放大图像。

10、如权利要求 9 所述的系统，其中由所述处理部件执行的所述放大程序还用于：

在显示所述放大图像时，检测第二信号；

根据检测的所述第二信号，减小用于所述放大图像的放大系数并且利用所确定的尺寸、所确定的位置和所述减小的放大系数重新显示所述放大图像。

11、如权利要求 10 所述的系统，其中由所述处理部件执行的所述放大程序还用于：

在显示所述放大图像时，检测第三信号；

根据检测的所述第三信号，终止显示所述放大图像。

12、如权利要求 11 所述的系统，其中由所述处理部件执行的所述放大程序还用于：

在显示所述放大图像时，检测显示状态的改变，以提供新的显示状态；

根据检测的所述显示状态的改变，按照新的显示状态重新确定用于所述放大图像的尺寸、位置和放大系数，并且重新显示所述放大图像。

13、如权利要求 8 所述的系统，还包括指示设备，并且其中由所述处理部件执行的所述放大程序用于：

通过使用指示设备在示出所确定的显示上下文的所述显示屏的一部分上放置指针，来确定所述显示上下文。

14、如权利要求 8 所述的系统，其中由所述处理部件执行的所述放大程序用于通过下列步骤确定所述显示的放大图像在所述显示屏上的位置：

在所述显示屏上确定所述显示上下文的定位；以及

使用所述显示上下文的位置作为在所述显示屏上所选择的定位。

15、一种计算机可读介质，其内容致使计算机系统通过执行下列步骤

能够根据放大信号选择性放大显示屏的一部分：

确定显示上下文；

确定放大图像在所述显示屏上的位置；

确定用于所述放大图像的尺寸和放大系数；以及

利用所确定的位置、所确定的尺寸和所确定的放大系数显示所述放大图像，其中所述放大图像具有所述显示上下文的功能特性。

16、如权利要求 15 所述的计算机可读介质，其中所述计算机可读介质的内容还致使计算机系统：

在显示所述放大图像时，检测第一信号；以及

根据检测的所述第一信号，增加用于所述放大图像的放大系数并且利用所确定的尺寸、所确定的位置和所述增加的放大系数重新显示所述放大图像。

17、如权利要求 16 所述的计算机可读介质，其中所述计算机可读介质的内容还致使计算机系统：

在显示所述放大图像时，检测第二信号；以及

根据检测的所述第二信号，减小用于所述放大图像的放大系数并且利用所确定的尺寸、所确定的位置和所述减小的放大系数重新显示所述放大图像。

18、如权利要求 17 所述的计算机可读介质，其中所述计算机可读介质的内容还致使计算机系统：

在显示所述放大图像时，检测第三信号；

根据检测的所述第三信号，终止显示所述放大图像。

19、如权利要求 18 所述的计算机可读介质，其中所述计算机可读介质的内容还致使计算机系统：

在显示所述放大图像时，检测显示状态的改变，以提供新的显示状态；以及

根据检测的所述显示状态的改变，按照新的显示状态重新确定用于所述放大图像的尺寸、位置和放大系数，并且重新显示所述放大图像。

20、如权利要求 15 所述的计算机可读介质，其中所述计算机可读介质的内容还致使计算机系统使用指示设备在示出所确定的显示上下文的所述显示屏的一部分上放置指针。

21、如权利要求 15 所述的计算机可读介质，其中所述计算机可读介质的内容还致使计算机系统通过下列步骤确定所述显示上下文的放大图像在所述显示屏上的位置：

在所述显示屏上确定所述显示上下文的定位；以及
使用所述显示上下文的位置作为在所述显示屏上所选择的定位。

用于选择性显示部分显示屏的用户界面系统和方法

技术领域

本发明涉及用于在计算机或计算设备显示屏上显示内容的计算机软件。具体地，本发明涉及用于选择性地显示显示屏的一部分的放大视图的计算机软件。

背景技术

近几年来，计算机、计算设备以及相关外围设备经历了持续不断的改进过程。例如，事实上由于电子设备的小型化的发展，所有计算机和计算设备已经变得越来越小并且越来越轻。与这些设备相关的外围设备，包括显示屏，同样也变得越来越小。这些显示屏在体积上越来越小的同时，也在显示分辨率上有显著的增加。

这种像素（“pixel”）密度的增加能够产生一种情况，即人们较难有效地看出在显示屏上显示的内容。对于视力减弱的人而言，使用与这种显示屏关联的计算机或计算设备就变得不可能。甚至对于具有正常敏锐视力的人而言，这种查看困难会导致艰辛的和令人失望的经历，这会严重影响计算机或计算设备的生产力和使用感受。

现有技术的计算机系统及其相关的应用通常具有对显示屏一部分进行放大和缩放的能力。一些版本的 Microsoft Windows®操作系统，例如，包括称作“放大器”的程序。诸如放大器的现有技术方法还存在一些缺陷。例如，放大器永远专用屏幕的一部分来显示鼠标指针下任意内容的放大视图。这种屏幕空间的分配减少了可用于显示非放大内容的屏幕分辨率的数量。此外，由于放大器显示区域的尺寸是固定的，其不能同时放大显示器的所有相关部分。放大器也不允许用户直接与被放大内容进行交互。被放大的显示区域在其中不是“活动的”，例如，放大按钮不起作用了。能够点击该显示区域，但是点击后不会发生任何结果。最后，没有保持被放大图像的位置与在屏幕上所显示的图像之间的空间关系。例如，如果放大视图窗口

在显示屏的顶端，而选择屏幕底端图像的一部分用于放大，则被选择的处于屏幕底端的部分将以放大的形式出现在屏幕的顶端。虽然诸如放大器的现有技术方法提供了改变放大量的能力，但是其仅经由麻烦的基于菜单的系统来做到这些。为了改变放大设置，用户必须通过菜单操作并选择一些新的预设放大级别，这将会一直生效直至下次这种设置。放大器，以及相似的现有技术方法，仅仅是简单的计算机实现的具有固定放大量的放大镜。

其它现有应用程序通常包括对由这种软件所显示的内容进行放大或缩放的一些能力。例如，Adobe Reader 6.0 允许用户改变打开的 PDF 文件的显示放大率。在从 Reader 6.0 的菜单中选择缩放选项后，指针图标变成放大镜的形状。用户指向他们希望放大的文档区域，并点击按钮。则放大了 PDF 文档视图，并重新以应用程序窗口的视图区域为中心。除增加视图的放大率以外，最终被放大视图的尺寸和位置总是一样的。此外，这种在 Reader 6.0 内的放大器功能局限于仅用于 PDF 文档内容。也就是，除了应用程序窗口的视图区域，没有 Reader 6.0 应用程序窗口的其它部分或任意其它窗口中的内容可以利用这种功能来进行放大。这是一个严重的缺点，因为用户没有能力放大 Reader 6.0 的任何工具栏或菜单。此外，通过放大整个 PDF 文档，也放大了不需要放大的部分文档，从而导致文档的其它部分延伸到显示屏的外部。

另一类现有软件应用程序使用一种功能基本类似于图片剪切的放大方法。例如，Microsoft®Streets and Trips（‘Streets’）是一种用于查看地图的软件应用程序。在启动 Streets 时，该应用程序通常在应用程序窗口的视图区域显示西半球的地图。用户可以通过使用鼠标或其它指针在所选取的子区域周围拖拽一个方框，并且在所选子区域内点击鼠标按钮来在这个显示子区域上进行放大。随后将所选子区域放大以符合应用程序窗口的视图区域，而处于所选区域之外的部分基本上被从中剪切掉。与 Adobe Reader 6.0 一样，被放大视图的尺寸和位置是固定的。同样与 Adobe Reader 6.0、Streets 及其放大显示屏一部分的方法，对任何 Streets 工具栏、菜单或者甚至任何其它应用程序的任何部分都不起作用。

因此，需要一种改进的放大方法，用于智能地放大显示屏的任何或所有部分而保持与被放大内容进行交互的能力。这种方法还将优选地提供一

种改变放大率的直观、快捷的方法。理论上，这种方法也会自动地允许在放大视图内移动显示下面的内容。

发明内容

本发明涉及一种系统和计算机实现方法，当该系统和方法结合计算机系统和其它计算设备使用时可用于选择性地显示显示屏的所放大部分。典型计算机系统和计算设备通常安装有显示信息和各种类型图形的模块。某些计算机操作系统和软件应用程序仅提供显示显示屏的某些子部分被放大或被缩放的视图的有限的模块。这样的操作系统或应用程序，例如，不为用户提供任何直接与放大视图交互的能力。该计算机实现方法提供了一种选择并显示显示屏的某些部分的被放大视图的增强模块。基于屏幕指针的定位，该方法自动确定在屏幕指针下和屏幕指针周围的区域中的显示屏的内容和上下文。可选地，该方法放大当前的前台窗口，而不是屏幕指针下面和其周围的窗口。该方法使用关于内容的信息来显示该内容的放大视图。通过如此操作，该方法使得被放大图像保持下面的内容和上下文的功能方面。该方法还提供了一种可配置的交互式增加或减少被放大内容放大率的装置。该方法还允许在所放大视图内自动移动显示下面的内容。因此，所提供的方法能够大大增强计算机系统及其相关应用程序的可用性和亲和力。

附图说明

图 1 是计算机系统框图，在该计算机系统中实现了本发明实施例。

图 2 是本发明实施例的高级流程图。

图 3 是图 2 所示的高级流程图中准备放大窗口子例程的流程图。

图 4 是放大例程的一个实施例的流程图。

图 5a 是显示包含图形图像的典型网页的网页浏览器应用程序的屏幕截图。

图 5b 是显示典型网页并示出图 5a 所示的图形图像的被放大视图的网页浏览器应用程序的屏幕截图。

图 6a 是显示典型对话框窗口的字处理应用程序的屏幕截图。

图 6b 是显示图 6a 所示对话框的放大视图的字处理程序的屏幕截图。

具体实施方式

本发明的实施例提供一种用于选择性地显示显示屏幕的一部分的系统和方法。在下面的描述中以及在图 1 至图 6b 中提出了本发明某些实施例的许多具体细节，以提供对这些实施例的全面理解。尽管如此，本领域技术人员可以理解本发明还具有其它的实施例，或者在没有下面所描述的若干细节的情况下可以实施本发明。

图 1 示出一种示例性计算机系统 100，在其中实现了本发明实施例。虽然以在操作系统中实现的实施例的形式或按照在计算机上运行的独立的应用程序来进行描述，但是本领域技术人员将会认识到本发明也可以在与其它程序模块的组合中实施。一般说来，程序模块包括例程、程序、组件，DLL，插件 (plug-in)，JAVA 程序，数据结构等，其执行特定的任务或实现特定的抽象数据类型。本领域技术人员还应当注意，可以使用其它计算机系统结构实现本发明，这些计算机系统结构包括例如手持设备，比如蜂窝电话，个人数字助理 (PDA) 或其它基于微处理的或可编程的消费电子产品，多处理器系统，微型计算机，大型计算机等。

计算机系统 100 包括处理器 104，其通过存储器/总线接口 112 耦合至主存储器 108。将存储器/总线接口 112 耦合至扩展总线 116，例如工业标准结构 (ISA) 总线或周边部件互连 (PCI) 总线。计算机系统 100 还包括一个或多个输入设备 120，例如键盘，鼠标或手写笔，所述输入设备通过扩展总线 116 和存储器/总线接口 112 耦合至处理器 104。输入设备 120 允许操作者或电子设备向计算机系统 100 输入数据。将一个或多个输出设备 120 耦合至处理器 104 以提供由处理器 104 生成的输出数据。输出设备 124 通过扩展总线 116 和存储器/总线接口 112 耦合至处理器 104。输出设备 124 的实例包括打印机和声卡驱动的音频扬声器。将一个或多个数据存储设备 128 通过存储器/总线接口 112 和扩展总线 116 耦合至处理器 104，以在存储介质 (未标示) 中存储数据或从其中取回数据。存储设备 128 和存储介质的实例包括固定磁盘驱动器、软盘驱动器、盒式磁带和压缩盘只读存储驱动器。

计算机系统 100 还包括图形处理系统 132, 其通过扩展总线 116 和存储器/总线接口 112 耦合至处理器 104。可选地, 图形处理系统 132 可以通过其它类型的架构耦合至处理器 104 和主存储器 108。例如, 图形处理器 132 可以通过存储器/总线接口 112 以及诸如加速图形端口 (AGP) 的高速总线 136 来耦合, 以利用直接存储器存取 (DMA) 向主存储器 108 提供图形处理系统 132。即, 高速总线 136 和存储器总线接口 112 允许图形处理器 132 在没有处理器 104 干预的情况下读取和写入主存储器 108。因此, 可以远大于通过扩展总线 116 传输的传输速率向主存储器 108 来回传送数据。将显示器 140 耦合至图形处理系统 132, 以显示图形图像。显示器 140 可以是任意类型的显示器, 例如阴极射线管 (CRT), 场发射显示器 (FED), 液晶显示器 (LED) 及其触摸屏变体等, 这些显示器通常用于桌上型电脑, 手持电脑和 workstation 或服务器应用。

图 2 示出本发明的一个实施例的高级流程图 200, 其可用于对在图 1 的计算机系统 100 中使用的处理器 104 进行编程。在对显示器的一部分进行放大之前, 用户指向其希望放大的显示器部分。这通常通过使用鼠标或其它输入设备将鼠标指针放到用户希望放大的区域上来实现。然后当用户按下某种组合按键或发起与放大显示屏相关的一些其它动作时, 则在步骤 204 处放大处理开始。本领域技术人员应当理解, 另一个实施例可以改变事件的顺序。例如, 在支持触摸屏的 PDA 上实现的本发明的实施例, 例如 PalmPilot®, 将要求用户首先初始化程序执行, 然后使用手写笔在恰当的定位敲击屏幕。在本发明的另一个实施例中, 将设置存储在系统寄存器中或其它适当的定位, 并且使得利用当前前台窗口而不是鼠标指针下的显示区域来开始放大。

尽管是按照按下一个或多个按键的方式进行描述的, 但是当用户按下新式鼠标上某一个可配置的按钮时, 放大过程也会开始。在触摸屏的情况下, 放大过程同样可以通过预定义的屏幕点击设置或其它输入机制初始化。尽管如此, 为了实现图 2, 可以理解, 用户指向其希望放大的显示器的部分, 然后按下预定义的一个或多个放大按键。

在步骤 204 处按下放大按键后, 本发明的实施例随后确定显示上下文。显示上下文是用户希望放大的显示器的一个矩形区域, 并包括关于指针下

面区域的图形内容以及关于显示区域内指针定位的信息。然而，具有不同形状的显示器部分，例如圆形，也可以用作显示上下文。例如，指针下面的区域可以包括纯图形内容，例如图形图像，或者该区域也可以包括对话框或用户需要用于与应用程序进行交互的其它窗口。在下面图 4 的描述中将更具体地解释显示上下文的确定。

通过在步骤 208 中首先测试指针是否位于对话框之上来确定显示上下文。对话框是特殊类型的窗口，其不允许用户重新调整大小。在某些情况下，不允许这种重新调整大小是很重要的，因为如果不恰当地调整对话框的大小，对话框内的控制或重要信息可能被隐藏。当指针位于对话框之上时，将显示上下文确定为指针及其定位之下的对话框。然后程序流程继续至步骤 220，以准备放大该对话框窗口。如果指针不在对话框之上，则在步骤 212 处进行进一步检查以确定指针是否在图形项的上面。步骤 208 和 212 处的检测顺序确保放大例程永远不会放大位于对话框内的图形项。图形项可以是任意类型的图形图像，例如 JPEG, GIF, TIFF, BMP, PCX, PCD, PICT, PNG, TGA, ICO 等等，这些是常用数字图像类型并且可以由计算机或计算设备来显示。图形项也可以是一个动态项，例如电子数据表中的图表。缩放功能使用微软提供的 API 来确定项目在用户界面中的作用。如果在步骤 212 中确定指针位于图形项的上面，则将显示上下文确定为由指针选择的图形项本身及其定位，并且程序执行继续至步骤 216。如果在步骤 212 中确定指针没有位于图形项的上面，则将显示上下文确定为指针下的应用程序窗口，及其由指针选择的定位。在这种情况下，程序执行继续至步骤 220。

在步骤 216 处准备放大图形项或在步骤 220 处准备放大对话框包括许多相同的任务。在两种情况下，确定并复制屏幕的相关子矩形的像素。在步骤 216 处，屏幕的这个矩形部分根据图形项的性质来确定，并且通常对应于该项的边框。在对话框的情况下，步骤 220 处的准备过程将子矩形设置为其自身窗口的边界区域。一旦确定了被放大的子矩形，则创建硬件视频覆盖，以便显示被放大的图像。正如本领域技术人员将会理解的，使用硬件视频覆盖是优选的，因为其利用图形处理系统 132（如图 1 所示）的特性来计算和维护图像，从而减轻了 CPU 的负担。另外，由于硬件视频覆盖

独立于视窗/操作系统进行操作，因此不存在与放大相关的副作用以降低与被放大窗口关联的软件应用程序的性能。具有不能支持硬件视频覆盖的不复杂图形处理系统的系统仍然可以用于实施本发明的实施例，尽管性能低下。

放大的图形项仅仅是基本图形项的放大视图，并且仅对于该放大的图形项的可视内容显示该图形项。除简单的放大该项以用于改善观看效果外，不需要操作或与该图形项进行交互。如果显示上下文是对话框或其它窗口，那么本发明的实施例将放大整个对话框或窗口，而同时允许操作和与放大的对话框或窗口进行交互。在步骤 220 处准备放大窗口还包括为放大该窗口而同时保持该窗口的放大视图是“活动的”所需要的那些操作。准备放大窗口或图形项还需要初始确定被放大图像可能具有的尺寸和位置。由许多因素来确定放大图像的位置。更典型地，放大图像的位置在被放大显示屏的矩形部分的中心。然后有时这样的位置将是不可能的。例如，在图形项处于十分接近显示屏边缘的情况下，以该位置为中心显示放大图像将导致放大图像的一部分移出显示屏的一个或多个边界。在那种情况下，确定放大图像具有尽可能接近指针定位的定位以避免任何由于边界的影响而丢失图像。对话框或窗口位置的确定也采用相似的形式。

当准备放大图形项或窗口时，还确定放大图像的尺寸。在图形项的情况中，该项放大视图的尺寸通常是以相对较小的系数来放大以保持背景的一些可视能力。例如，放大非常小的图形项使其占据整个屏幕将会不允许在该图形上下文中查看该图形项，并且将会使整个图像视图严重扭曲。通常优选地，初始以相对较小的放大系数来显示图形项。尽管本实施例是由程序确定的初始放大系数的形式进行描述的，但是另一实施例可以允许用户设置默认的放大系数。在以窗口替代显示上下文的情况下，该例程通常将试图最大化该窗口放大视图的区域。不管显示上下文是图形项还是窗口，图像的放大系数可以在任何时候由用户来改变，如下面更加全面讨论的。当显示上下文是对话框或其它窗口时，应当理解该对话框的放大显示在其放大状态中保持全部功能。即，尽管对话框现在以放大形式显示，但是所有按钮、菜单、选项、文本区域等都保持其全部功能。用户可以与放大对话框直接交互，就像其未被放大一样。

图3更具体地示出了准备放大窗口例程220。准备过程在步骤300处开始于检测所选择或活动窗口是否完全在屏幕上。如果不是,则在步骤304处重新放置窗口使其位于屏幕上。在确保窗口完全位于屏幕上之后,执行在步骤308处继续检测所选择窗口是否是前台窗口。同样,如果该窗口不是前台窗口,则在步骤312处例程将该窗口变为前台窗口并且流程继续至步骤316。该例程在步骤316处确定所选择窗口是否是对话框。当在步骤316处该例程确定该窗口不是对话框时,该例程计算放大图像的初始尺寸,并且随后在步骤320处确定该图像是否太宽以至于不适合于显示屏。如果图像太大,则重新计算放大图像的初始尺寸以便在步骤324中被放大后适于显示,程序流程继续至步骤330。如果所计算的初始图像的尺寸是可接受的,则执行同样继续至步骤330。这种调整大小的行为不包括对话框,因为这些对话框相对于显示屏的尺寸通常较大,并且调整该对话框的大小以适合于显示屏仅能提供较小的初始放大率或不能提供初始放大率。

准备放大继续在步骤330处确定指针是否当前在将被放大的窗口内。如果不是,则在步骤334处将指针移动至窗口中心。最后,在完成这些准备之后,建立并定位硬件覆盖,并且从而以预先计算的位置和放大系数来显示初始放大图像。然后程序控制转至用于管理放大图像的放大例程224。

尽管没有示出,但是准备放大图形项实质上与准备放大窗口220是相同的。在图形项的实例中,尽管不需要在步骤316处确定窗口是否是对话框。同样的,与例程320和324相关的逻辑也不是必须的。

图4更具体地示出了放大例程224。在本发明的一个实施例中,放大例程224包含一个循环。该循环在步骤404以检测该窗口是否已关闭进入。首次进入该循环时,虽然不是必须的,但是步骤404至420的各个检测通常是“假”并且程序流程将进入步骤428。通过该循环的连续迭代,在步骤404至420处的检测将导致不同的行为,如在下面更加全面描述的。在初始遍历该循环时,没有在放大图像上执行的更新,程序流程继续至步骤432。

放大循环在步骤432处继续检测用户是否按下放大键持续1秒钟的时间。在这种事件中,在步骤456处停止放大图像的显示。如果没有按下放大键,则在步骤436处通过检测期望的放大系数的变化来继续该例程。

如上所述,在放大图像的显示期间,用户可以增大或减小放大图像的

放大系数。典型地，设置两个热键以便控制放大系数的增大或减小。例如，可以将热键设置为持续按下键盘上的 Ctrl 键并按下 Up 箭头键将增大放大系数，同样的，持续按下 Ctrl 键并按下 Down 箭头键将减小放大系数。本领域技术人员可以理解，用于指示这些放大系数的其它方式也是可能的。例如鼠标滚轮，滚动滚轮、轻推滚轮、轻拨等都可以用于在放大图像显示过程中改变放大系数。本发明的这个实施例在步骤 436 处检测由用户发起的对放大系数的改变。当检测到这种改变时，将关于改变量的信息（例如，滚轮点击多少次和/或按下按键的数目）传回至步骤 424，以用于重新确定放大图像的尺寸和位置，然后流程如以前一样继续。当在步骤 424 处重新确定尺寸和位置之后，程序流程继续至步骤 428 用于放大图像的更新。在步骤 428 处使用新的尺寸和位置并根据改变的放大系数来重新显示放大图像。在一些实例中，需要对放大图像的其他更新，这些更新也在步骤 428 处完成。

本发明的一个实施例提供了对放大图像下面的内容的自动移动显示。如在图 3 的描述中所讨论的，在准备放大窗口结束时，但是在显示放大图像之前，将指针移动到窗口的中心。因此，在放大内容初始显示之上，指针在放大视图内是可视的。如果用户试图将鼠标指针移动到放大图像的边界之外，放大例程 224 将在步骤 428 处自动更新放大图像。这种更新通过从下面的窗口中选择新的将被放大的子矩形来完成。这个新矩形的尺寸与现有子矩形相同，但是在鼠标移动的方向上移位。例如，假设用户将指针移动至放大图像的最顶端边界。放大例程在步骤 428 处将检测指针到边界的接近度，并向上移动下面内容的所选子矩形。该行为对于放大图像的四个边界分别是相同的，并且该放大例程将允许移动显示下面的内容，向上到该内容的边界。一旦确定了这个新的子矩形，则如前所述拷贝其像素并更新硬件覆盖，从而重新显示移动显示的图像。

在放大图像在步骤 428 处更新之后，由于放大系数的改变或因为移动显示，流程通常继续至步骤 432 和 436，并返回至步骤 404，其中执行一系列的检测步骤 404 至步骤 420 以便确定显示是否改变了状态。这种改变能以若干方式中的任意方式发生。在显示状态的最后检查之后，与放大图像相关的窗口可能已被关闭、隐藏、最小化，失去焦点或被移动和/或调整大

小。如以上简要讨论的，在循环的第一次遍历期间，通常这些事件都不会发生。对于下面的对在步骤 404、408、412、416 和 420 处所示的检测的讨论，应当理解尽管以检测窗口状态的改变进行讨论，但是当显示上下文是图形项时以及当它是对话框或其它窗口时都执行这些检测。当显示上下文是图形项时，则在该图形项位于的窗口上执行这些检测。当显示上下文是对话框或其它窗口时，则在对话框或其自身的窗口上执行这些检测。

在步骤 404 处执行第一次检测，以确定窗口是否已经关闭。在窗口已经关闭的情况下，控制转至步骤 440，步骤 440 将在下面更加全面的讨论。如果窗口未被关闭，程序执行在步骤 408 处继续检测窗口是否变为隐藏。窗口可以由于多种原因而变为隐藏。最典型地，新窗口在放大窗口的上面打开，从而从视图中遮蔽了放大窗口。如果窗口被隐藏，则控制转至步骤 440，否则控制向下转至步骤 412。程序流程在步骤 412 处继续执行，在步骤 412 处执行测试以检测窗口是否被最小化。在几乎所有的情况中，窗口最小化是由于用户的人为操作。如果窗口被最小化，放大例程在步骤 440 处继续执行。否则，流程在步骤 416 处继续检测窗口是否失去焦点。当新窗口打开并且将用户输入指向该窗口时，窗口失去焦点。这可能由于多种原因而发生。例如，用户可以在软件应用程序中选择一个或多个菜单选项从而导致显示对话框。当窗口没有失去焦点时，则在步骤 420 处执行最后剩余的检测，即检测该窗口是否被移动或改变尺寸。

如果在步骤 416 处确定窗口已经失去焦点，则流程转移至步骤 448，在步骤 448 中所执行的检测是用于确定新窗口是否为相同的应用程序所有。通过实例方式可以更好地理解该检测。假设用户正在运行一个软件应用程序并且放大了该应用程序的对话框。如上所述，对话框在其放大形式中是完全起作用的，用户可以与对话框进行交互就像其未放大一样。此外，假设用户发起对话框中的一些动作，其导致打开新的对话框或窗口。当新窗口打开时而先前的对话框仍就可见时，则新窗口具有焦点而旧窗口失去焦点。由于旧窗口未被关闭、隐藏或最小化，放大循环将在步骤 416 检测到窗口失去焦点，并且将控制传送至步骤 448 处的测试。在步骤 448 处的测试确定当前处于焦点中的窗口是否由与曾经处于焦点中的窗口相同的应用程序占有或者在该相同的应用程序中运行。在这些实例中，新的对话框事

实上作为该相同的应用程序的一部分来运行。因此，在步骤 452 处保存显示的先前放大状态，并且取而代之放大当前处于焦点中的新窗口。当保存显示的先前放大状态时，保存关于先前处于焦点中的对话框的放大信息。该信息通常包括关于放大图像的定位及其尺寸和/或放大系数的信息。本领域技术人员应当理解，用于存储和取回该信息的最普通的抽象数据类型是堆栈。虽然该实施例是根据堆栈描述的，但是可以理解，本发明的其它实施例可以使用其它抽象数据类型。如下面更加全面讨论的，所保存的放大状态随后在步骤 440 和 444 处使用。在保存了显示的先前放大状态之后，在步骤 452 处通过确定新打开的对话框的位置和尺寸来放大该对话框，并且然后在步骤 428 处更新所显示的图像。然后如上所述，程序流程在步骤 432 进行执行。在新窗口不属于与先前窗口相同的应用程序的情况下，放大例程在步骤 456 处退出，放大图像从显示中移除。

当用户完成与当前放大窗口的交互，以及例如关闭该窗口时，被保存的放大状态随后将可以使用。假设已经保存了该放大状态，如果与当前显示的放大图像相关的当前窗口被关闭、隐藏或最小化，则放大例程将分别在步骤 404、408、412 检测到，并且将控制传送至步骤 440。在步骤 440 处，放大例程确定在堆栈中是否有所保存的放大状态。在该状态存在的情况下，则在步骤 444 中将其从堆栈中弹出并且利用所保存的尺寸合位置来重新显示与该状态相关的放大图像。这时，程序控制返回至在步骤 404 处的放大例程的主循环。如果在步骤 440 处确定在堆栈中没有存储放大状态，那么放大例程在步骤 456 处退出，并且将放大图像从显示中移除。

图 5a 示出了显示包含图形图像 504 的典型网页的网页浏览器应用程序的屏幕截图 500。如上文具体讨论的，图形图像 504 实际上可以是任何类型的图形图像。

图 5b 示出了显示典型网页并示出图 5a 所示的图形图像 504 的放大视图 512 的网页浏览器应用程序的屏幕截图 508。如上所述，该图形项以相对较小的初始放大系数来进行放大。这允许更容易地在整个网页环境中查看该图形项。

图 6a 是显示典型对话框窗口 604 的字处理应用程序的屏幕截图 600。

图 6b 是显示对话框 604 的放大视图 608 的字处理应用程序屏幕截图

602。如上所述，放大图像是完全“活动的”，用户可以直接与放大对话框进行交互。例如，可以选择“区域阴影（Field shading）”下拉框 612，并且其选项改变。同样，可以设置或清除“制表符（Tab characters）”复选框 616。如上所述，对话框以相对较大的系数进行放大，使得对话框几乎使用了整个显示区域。该初始放大系数是优选的，因为对话框通常较复杂并且包含大量的输入项。在这种情况下，放大对话框的可视性和可用性是极为重要的。同时，对话框的操作上下文通常没有该图形项重要。

尽管参照所公开的实施例对本发明进行了描述，但是本领域技术人员应当理解在不偏离本发明精神和范围的情况下，可以在形式和具体细节上进行修改。这种修改在本领域普通技术人员理解范围内。因此，除所附权利要求外，不对本发明进行限制。

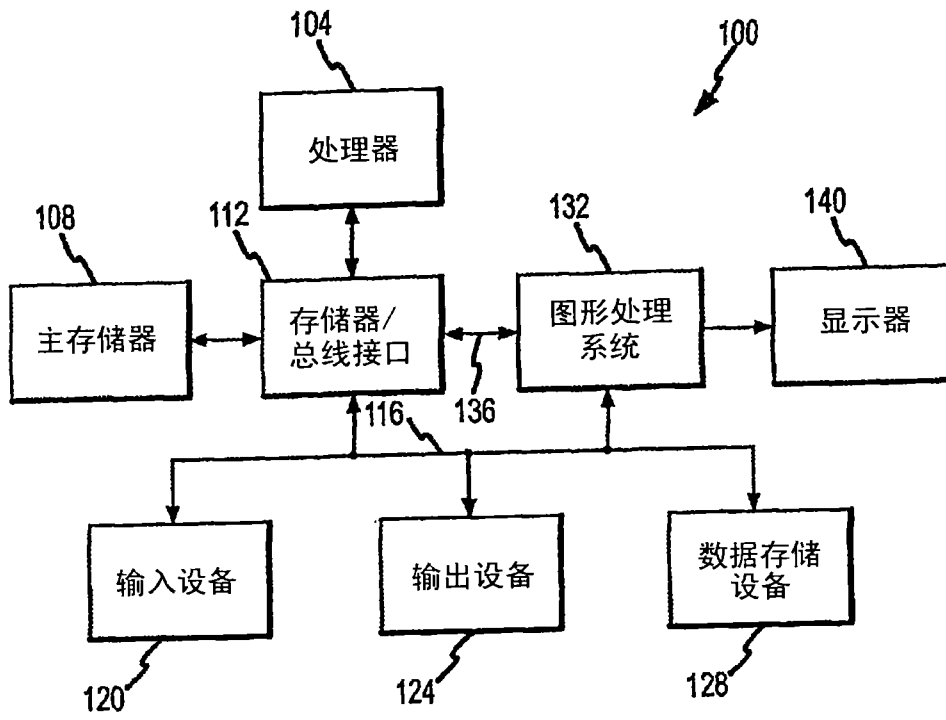


图 1

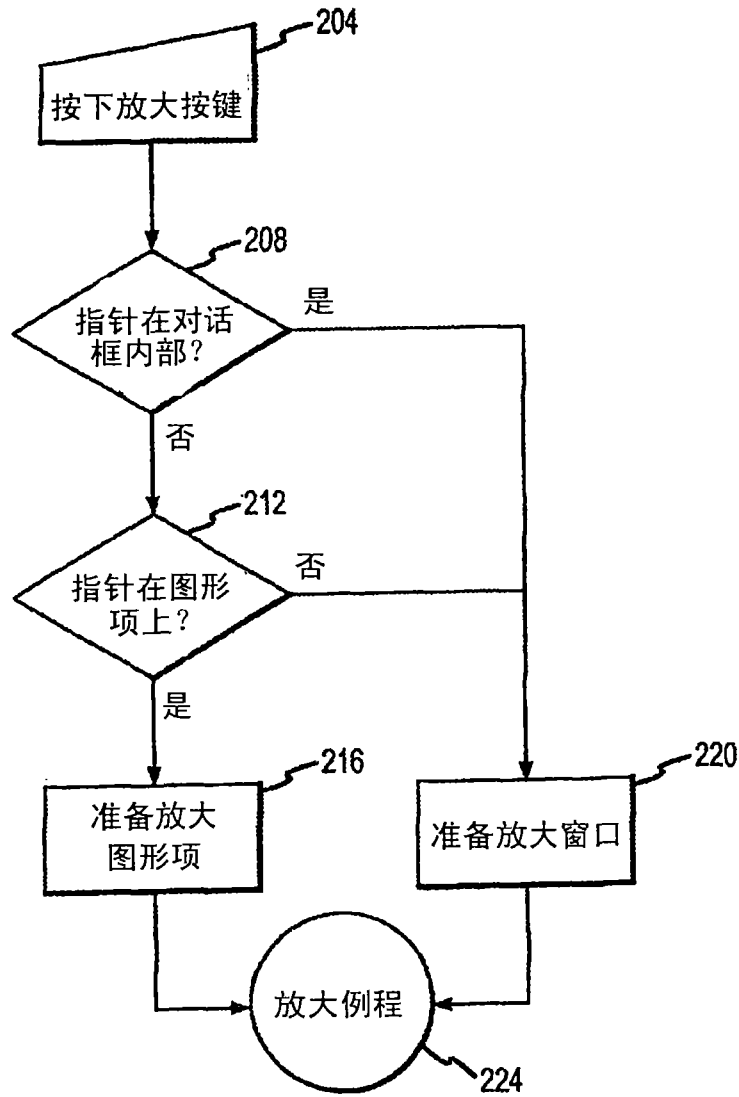


图2

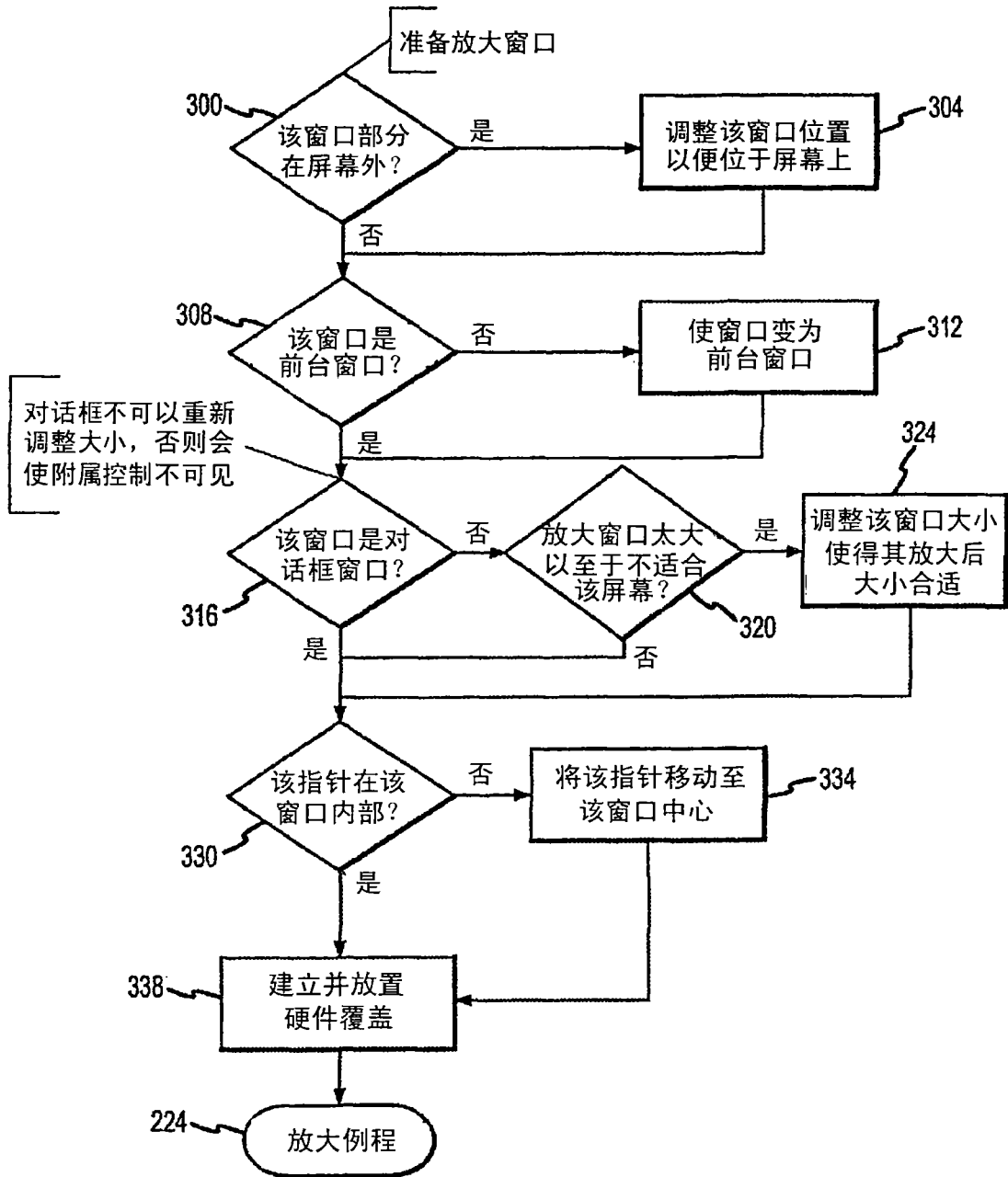


图3

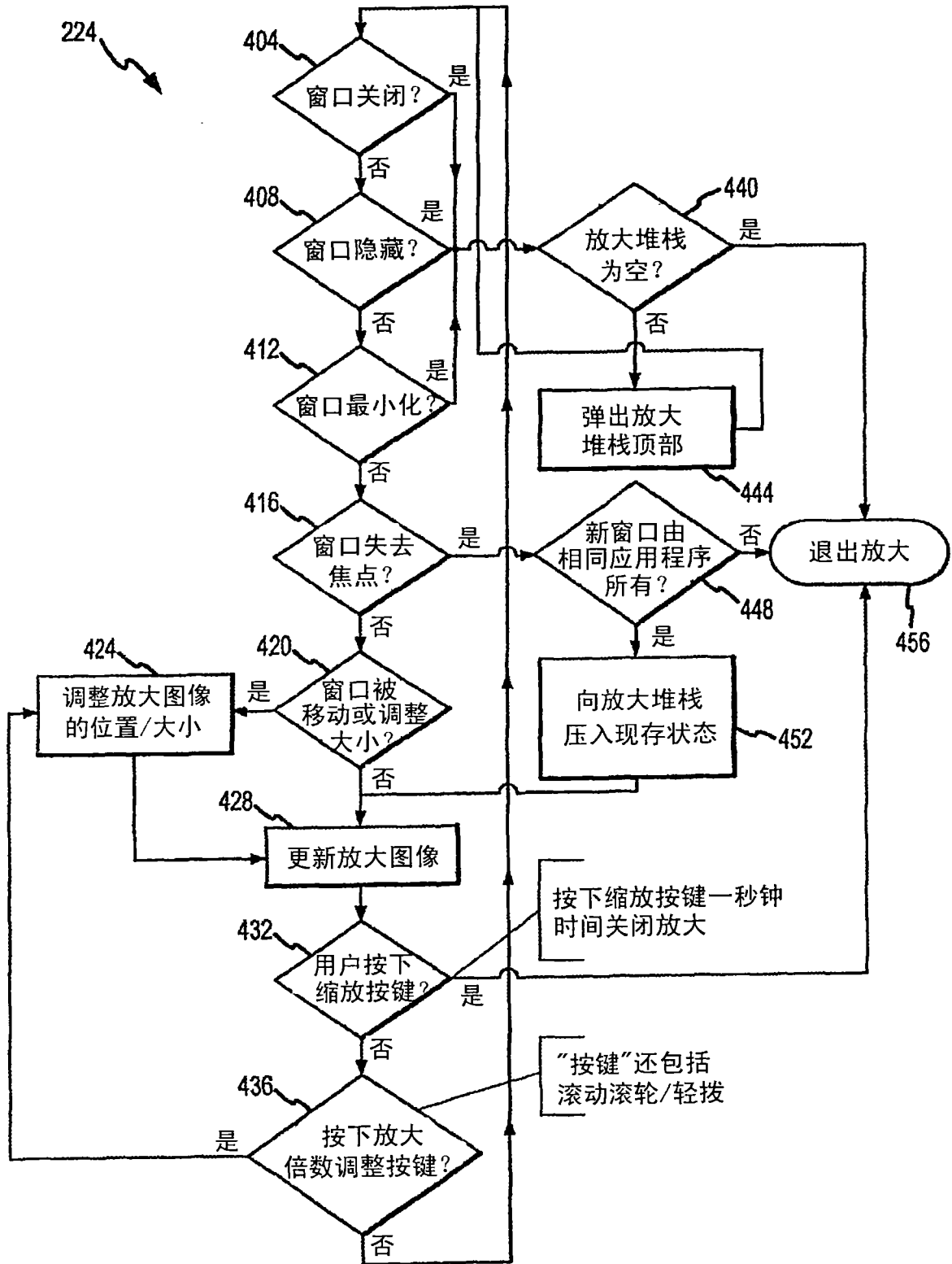
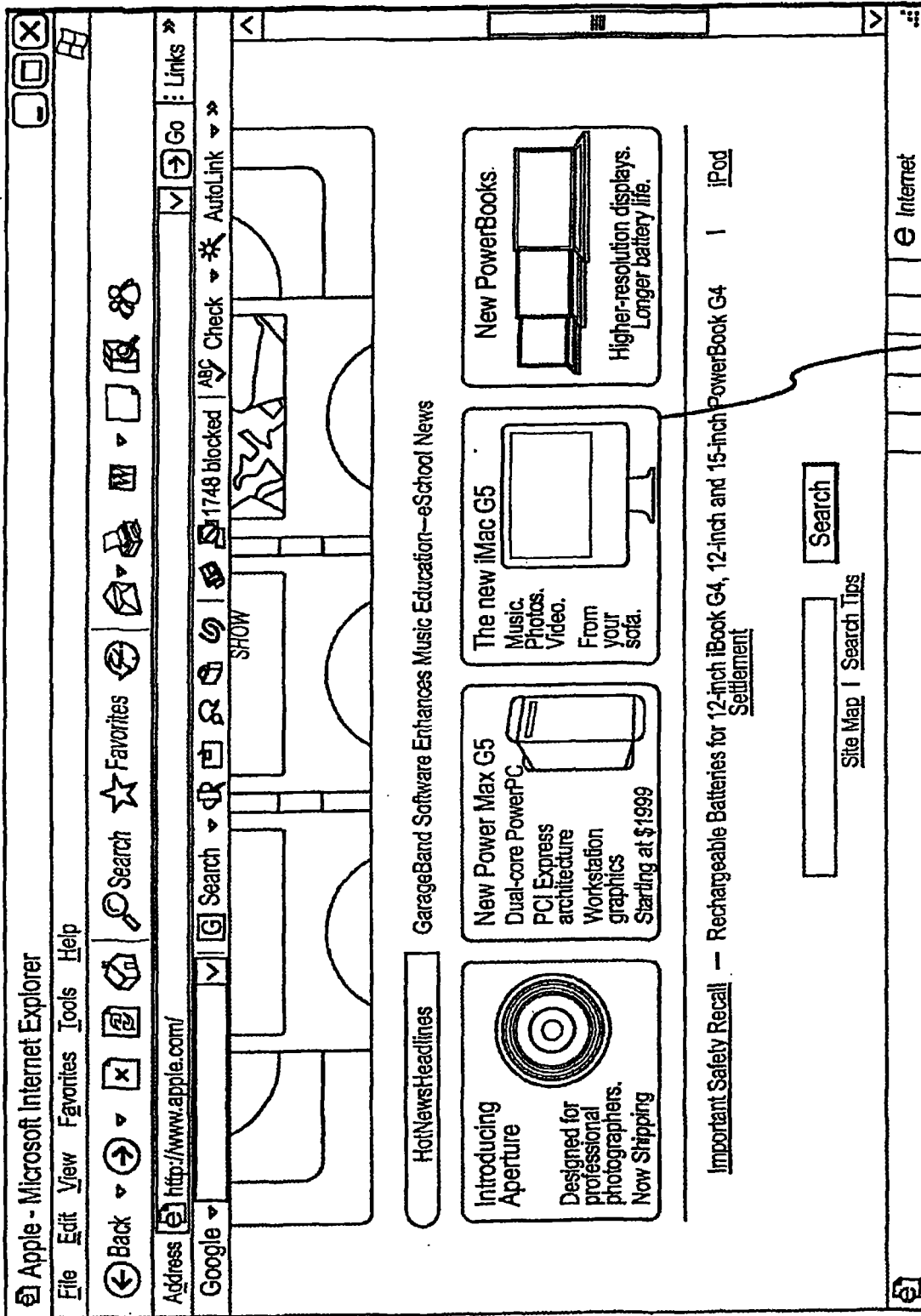


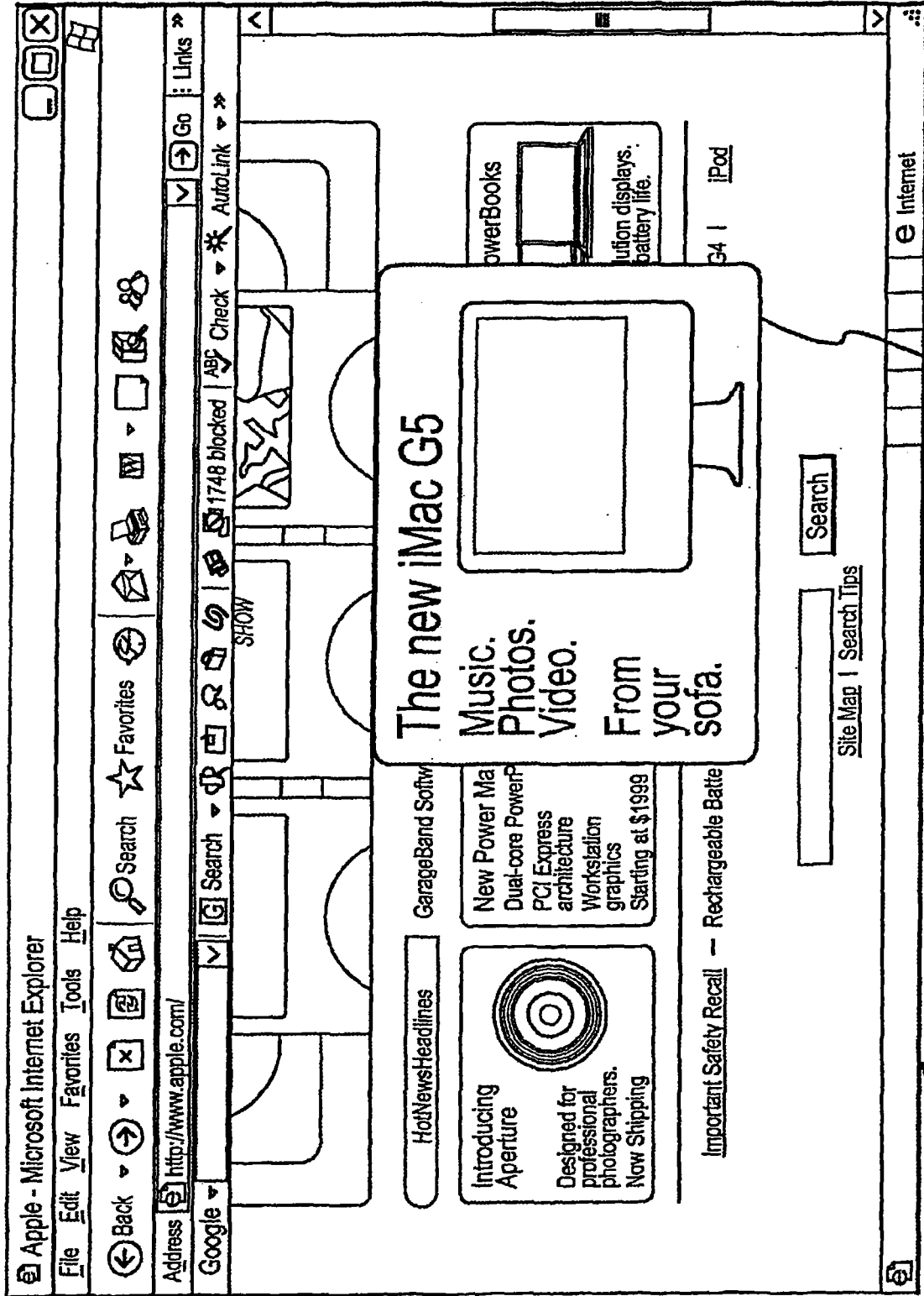
图4



504

500

图 5a



512

508

图 5b

602

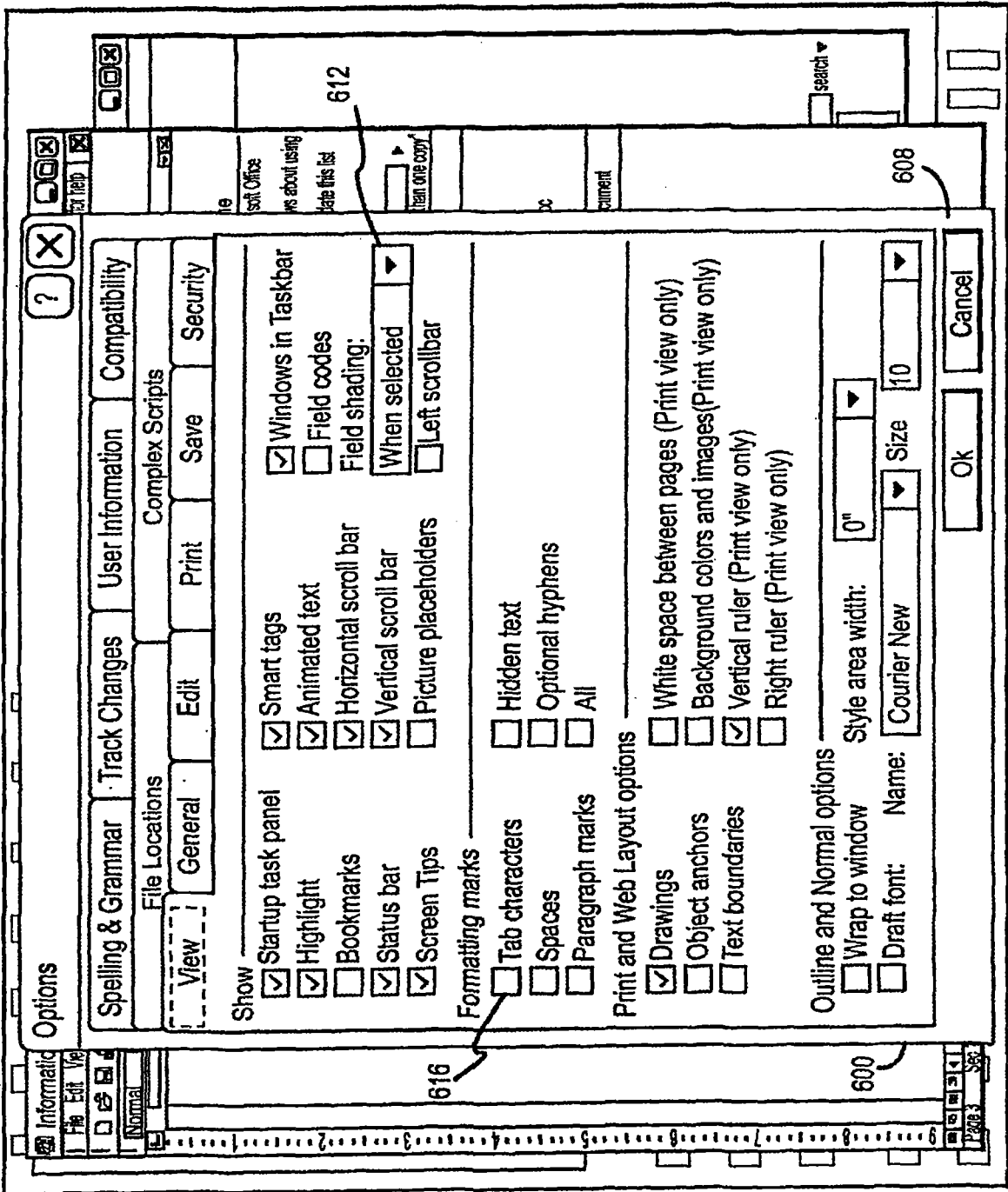


图 6b