

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 9/44 (2006.01)



[12] 发明专利说明书

专利号 ZL 200510052883.1

[45] 授权公告日 2008 年 9 月 17 日

[11] 授权公告号 CN 100419676C

[22] 申请日 2005.2.25

[21] 申请号 200510052883.1

[30] 优先权

[32] 2004.2.27 [33] US [31] 10/789,603

[73] 专利权人 微软公司

地址 美国华盛顿州

[72] 发明人 C·H·丘 J·W·福勒

K·C·本特利 R·O·格利

Z·科奇

[56] 参考文献

US6670964B1 2003.12.30

US4841375A 1989.6.20

US6044408A 2000.3.28

审查员 姚天宇

[74] 专利代理机构 上海专利商标事务所有限公司
代理人 张政权

权利要求书 3 页 说明书 10 页 附图 5 页

[54] 发明名称

使应用程序与像素密度改进的显示装置兼容的方法和装置

[57] 摘要

一种使应用程序与具有增加的像素密度的显示装置兼容的方法设备。根据一种方法，从应用程序接收指向用于执行屏幕输入或输出函数的应用编程接口的调用，该调用包括一个或多个参数。响应接收调用，判断应用程序是构造成用于具有较低像素密度的显示装置还是用于具有较高像素密度的显示装置。响应判定应用程序是构造成用于具有较低像素密度的显示装置，缩放参数以适用于较高像素密度的显示装置，并用缩放后的参数调用应用编程接口。

1. 一种使得被设定成用于具有较低像素密度的显示装置的应用程序能使用具有较高像素密度的显示装置的方法，该方法包括：

从应用程序接收指向用于执行屏幕输入或输出功能的应用编程接口的调用，所述调用包括一个或多个参数；

响应于接收所述调用，判断应用程序是否被设定成用于具有较低像素密度的显示装置和具有较高像素密度的显示装置的其中之一，其中所述判断包括：

检查应用程序的属性以判断应用程序被编译时的软件开发工具包的版本号，其中所述版本号对应于所述应用程序的像素密度支持水平，

如果所述版本号大于或等于预定值，则判断所述应用程序被设定成用于具有较高像素密度的显示装置，

如果所述版本号小于预定值，则判断所述应用程序被设定成用于具有较低像素密度的显示装置；以及

响应于判断出所述应用程序被设定成用于具有较低像素密度的显示装置，缩放所述参数以适用于较高像素密度的显示装置，并用缩放后的参数调用应用编程接口。

2. 如权利要求 1 所述的方法，其特征在于，还包括响应于判断出应用程序被设定成用于具有较高像素密度的显示装置，采用未缩放的参数调用应用编程接口。

3. 如权利要求 2 所述的方法，其特征在于，还包括响应于判断出应用程序被设定成用于具有较低像素密度的显示装置，从应用编程接口接收一个或多个返回参数，缩放所述返回参数以适用于较低像素密度的显示装置，并将缩放后的返回参数返回到应用程序。

4. 如权利要求 3 所述的方法，其特征在于，较低像素密度的显示装置具有每英寸 96 点的像素密度。

5. 如权利要求4所述的方法,其特征在于,较高像素密度的显示装置具有每英寸192点的像素密度。

6. 一种计算机控制的设备,其使得被设定成用于具有较低像素密度的显示装置的应用程序能使用具有较高像素密度的显示装置,所述计算机控制的设备包括:

第一装置,用于从应用程序接收指向用于执行屏幕输入或输出功能的应用编程接口的调用,所述调用包括一个或多个参数;

第二装置,用于响应接收所述调用,判断应用程序是否被设定成用于具有较低像素密度的显示装置和具有较高像素密度的显示装置的其中之一,其中所述判断包括:

检查应用程序的属性以判断应用程序被编译时的软件开发工具包的版本号,其中所述版本号对应于所述应用程序的像素密度支持水平,

如果所述版本号大于或等于预定值,则判断所述应用程序被设定成用于具有较高像素密度的显示装置,

如果所述版本号小于预定值,则判断所述应用程序被设定成用于具有较低像素密度的显示装置;以及

第三装置,用于响应于判断出所述应用程序被设定成用于具有较低像素密度的显示装置,缩放所述参数以适用于较高像素密度的显示装置,并用缩放后的参数调用应用编程接口。

7. 如权利要求6所述的设备,其特征在于,还包括:

第四装置,用于响应于判断出应用程序被设定成用于具有较高像素密度的显示装置,不缩放参数就调用应用编程接口。

8. 一种计算机系统,它被配置为使被创建为用于具有较低像素密度的显示装置的应用程序能使用具有较高像素密度的显示装置,该计算机系统包括:

中央处理器,其中操作系统在所述中央处理器上运行,为用于具有较低像素密度的显示装置而创建的应用程序在所述操作系统上执行;

具有较高像素密度的显示装置；以及

用于对显示装置执行输入和输出操作的应用编程接口模块，以及

转换层模块，用于截获应用程序对应用编程接口模块的调用、将调用缩放至显示装置、并采用缩放后的参数来调用应用编程接口模块，其中所述转换层模块还可运作以判断应用程序是否被设定成用于具有较高像素密度的显示装置，所述判断包括：

检查应用程序属性以判断应用程序被编译时的软件开发工具包的版本号，其中所述版本号对应于所述应用程序的像素密度支持水平，

如果所述版本号大于或等于预定值，则判断所述应用程序被设定成用于具有较高像素密度的显示装置，

如果所述版本号小于预定值，则判断所述应用程序被设定成用于具有较低像素密度的显示装置。

9. 如权利要求 7 所述的计算机系统，其特征在于，所述转换层模块还可运作以判断应用程序是否构造成用于具有较高像素密度的显示装置，并响应于判断出应用程序被设定成用于具有较高像素密度的显示装置，不缩放参数就调用应用编程接口。

10. 如权利要求 9 所述的计算机系统，其特征在于，具有较高像素密度的显示装置具有 192dpi 的像素密度，为用于具有较低像素密度的显示装置而创建的应用程序被设定成用于具有 96dpi 像素密度的显示装置。

11. 如权利要求 10 所述的计算机系统，其特征在于，所述转换层模块还可运作以从应用编程接口接收一个或多个返回参数，缩放返回参数以适用于较低像素密度的显示装置，并将缩放后的返回参数返回到为用于具有较低像素密度的显示装置而创建的应用程序。

使应用程序与像素密度改进的显示装置兼容的方法和设备

技术领域

本发明总的涉及应用程序的向后兼容性领域，更具体的说，涉及使应用程序与像素密度改进的显示装置向后兼容的领域。

背景技术

随着手持式计算装置（这里也称为个人数字助理（PDA））的发展，这类装置的计算能力极大地提高。特别是，可以看到 PDA 的硬件组成之一的显示屏幕很大的发展了。随着时间的过去，PDA 显示屏幕的分辨率和像素密度都有了很大的增大。如本文所用的，术语“分辨率”指的是显示屏幕所能够达到的水平和垂直的像素数目。例如，一些 PDA 使用具有 240 水平方向像素和 320 垂直方向像素的显示屏幕，而其它 PDA 使用具有 480 水平方向像素和 640 垂直方向像素的显示屏幕。如本文所用的，术语“像素密度”指的是显示屏幕在特定度量单位下所能达到的像素数目。一般，像素密度用每英寸点（或者像素）（“dpi”）来表示。例如，一些 PDA 采用 96dpi 的显示屏幕，而其它 PDA 具有较高的 192dpi 的显示屏幕。

提高的现代 PDA 的分辨率和像素密度对使用者是一种福音。提高的分辨率和像素密度使得更多的信息能够被显示在屏幕上，故此文档、网页查看更轻松，也便于执行其它功能。提高的分辨率也提高了清晰度和可读性。然而，提高的分辨率和像素密度给编写 PDA 应用程序的程序员制造了挑战。一般，在假定显示屏幕具有特定的像素密度的情况下来编写很多 PDA 应用程序时，就导致了这些挑战。应用程序调用操作系统提供的编程接口（“API”）以实现屏幕相关的功能，这样的编程接口常把像素作为参数，这就引发了问题。那么，举例来说，如果用于 96dpi 显示屏幕的应用程序画出一个 1/2 英寸宽的屏幕上的用户界面按钮，它将调用 API 来创建具有宽度为 48 像素的按钮。然而，如果相同的应用程序在具有较高像素密度的 PDA 上执行，例如 192dpi，

相同的 48 像素宽度的按钮仅仅只有 1/4 英寸宽。由于应用程序在引入所述的具有较高像素密度的装置之前创建，不知道像素密度的增大，所以，所有这些程序画出的用户界面物体将比所要求的显示出来要小。将屏幕用作触敏装置的输入操作也会被这些旧的应用程序误解。

这个问题的一个解决方法是要求应用程序程序员重新编写他们的应用程序以支持新的 PDA 的较高的像素密度。然而，程序员必须对应用程序进行重编码，这可能会使程序员受挫且会许多花费时间。这也会使用户感到灰心，因为他们必须等到应用程序被重编码出来才能将应用程序用于新的提高了像素密度装置。在一些情况下，程序员可能简单的决定不为较高像素密度的显示器进行重编码。那么，程序的用户为使用较高像素密度的显示屏幕，将被迫使用其它程序。

这个问题的另一个解决方法是创建新的一套 API 以在较高分辨率屏幕上执行与屏幕相关的函数。然而，这个解决方法也不合需要，因为，它不允许旧的应用程序使用较高像素密度的显示器，且要求程序员使用全新的 API 来创建新的高密度感知的应用程序。因此，人们希望使得旧应用程序能够不重写应用程序就在改进的像素密度屏幕上正确地输入输出，使得新的高密度感知应用程序能使用与旧应用程序相同的一套屏幕操作 API，并且不重写执行屏幕输入输出的 API 就能够实现该功能。正是出于这些和其它的考虑作出了本发明的各种实施例。

发明内容

依照本发明，上面的和其它的问题可以通过一种使得应用程序与具有较增大的像素密度的显示装置相兼容的方法和设备来解决。使用本发明提供的功能，旧的应用程序可以不加更改地正确输入和输出到改进像素密度的屏幕上。而且，能感知高密度的应用程序可使用与不知道像素密度增大的旧的应用程序相同的一套屏幕操作的 API。不需要重写执行屏幕输出和输出的 API 就可以提供本功能。

依照本发明的一个方面，提供了一种方法，用来使得用于具有较低像素密度的显示装置的应用程序能完全地使用具有较高像素密度的显示装置。依照这

种方法，在 PDA 或其它类型的计算装置上执行的应用程序可以提出请求或者调用执行屏幕输入和输出功能的 API，诸如在屏幕上画图或者从触敏屏幕接收输入。该 API 调用可包括一个或多个定义所要执行的操作的参数，并可能用像素单位来表达的参数。例如，可能会调用用于画出一个宽度为 48 像素的屏幕上用户界面按钮的 API。

当接收此类调用时，调用可在到达想要的 API 之前被截获。那时，会判断应用程序是构造成用于具有较低像素密度的显示装置还是用于具有较高像素密度的显示装置。可以通过检查应用程序属性以判断应用程序被编译时的软件开发工具包的版本号来作出这个判断。拥有大于或者等于预定编号的版本号的应用程序被假定为支持具有较高像素密度的显示器。拥有低于预定编号的版本号的应用程序被假定为仅支持具有较低像素密度的显示器。

如果应用程序构造成用于具有较低像素密度的显示装置，则将从 API 接收的参数缩放至较高像素密度的显示装置的像素密度。例如，如果应用程序是构造成用于 96dpi 像素密度，而显示设备具有 192dpi 像素密度的较高的像素密度，则参数会被翻倍。然后，采用缩放后的参数来调用所要的应用程序接口。如果应用程序是构造成用于具有较高像素密度的显示装置，则不缩放 API 函数调用的参数而直接将 API 调用传递给 API。如此，用于具有较高像素密度的显示屏幕的应用程序将按预定设计执行。

如果从 API 调用返回参数，诸如当 API 用于从触敏显示器接收指示笔输入时，则将这些参数缩放至较低的像素密度。通过截获对执行屏幕 I/O 操作的原 API 的函数调用，缩放参数，然后采用缩放后的参数调用原 API，不支持较高像素密度显示屏幕的旧应用程序就能够使用高分辨率设备的整个显示屏幕，而不需要重写应用程序或 API。

依照本发明的另一个方面，提供一种计算机系统，它被配置为使得创建出来用于具有较低像素密度的显示装置的应用程序能完全地使用具有较高像素密度的显示装置。该计算机系统包括中央处理器（“CPU”）、具有较高像素密度的显示装置以及可有效存储一个或多个可执行程序的存储器。特别是，存储器可有效地存储在 CPU 上运行的操作系统、构造成用于具有较低像素密度的显示装置的应用程序、用于在显示装置上执行 I/O 操作的 API 以及转换层。

转换层可运作以截获应用程序对 API 的调用。当接收此类调用时，转换层将任何从调用所接收的参数缩放至所安装的显示屏幕的较高像素密度。接着，转换层使用缩放后的参数调用所要的 API。如果从 API 调用返回参数，诸如当 API 是用于从触敏显示器接收输入时，则转换层将参数值缩放至应用程序所使用的较低像素密度。

转换层也能够运作以判断施行调用的应用程序是构造成用于具有较低像素密度的显示装置还是用于具有较高像素密度的显示装置。如果应用程序构造成用于较低像素密度的显示器，则转换层采用上述方法对参数进行缩放。然而，如果转换层判定应用程序已是创建成用于具有较高像素密度的显示器，则转换层不缩放参数就调用所要求的 API。关于施行调用的应用程序是支持较低还是较高像素密度的判断可以通过检查在应用程序编译时用于指示软件开发工具包的版本号的应用程序属性来实现。

本发明可实施为计算机进程、计算系统、或者实施为诸如计算机程序产品或者计算机可读媒介的制造物品。计算机程序产品可以是计算机系统的可读的计算机存储媒介，它编码执行计算机进程的指令的计算机程序。计算机程序产品也可是计算机系统可读的传播载波信号，它编码执行计算机进程的指令的计算机程序。

这些和其它一些表现本发明特色的特征和优点将通过后面的详细描述和附图而变得清晰。

附图说明

图 1 是计算机系统体系图，说明为本发明的实施例提供操作环境的手持式计算装置的体系；

图 2A-2B 是说明在各种本发明的实施例中可采用的两种不同的显示屏幕的分辨率和像素密度的简图；

图 3A-3B 是说明使用现有技术方法在两个具有不同像素密度的不同显示器上显示相同物体的效果的简图；

图 4A-4B 是本发明的实施例所采用的、保证应用程序与具有改进的像素密度的显示装置相兼容的软件体系；以及

图 5 是一个流程图，说明为根据本发明的一个实施例的、保证应用程序与具有改进的像素密度的显示装置相兼容的说明性方法所执行的逻辑操作。

具体实施方式

现在请参见附图，本文将描述本发明的不同方面，在附图中相同的编号代表相同的部件。图 1 和以下的讨论想要提供对可施行本发明的一个合适的计算环境的简洁的总体描述。应理解的是图 1 的这个计算环境仅仅是说明性的，而不是想要局限本发明。

图 1 描述一个诸如可存在于手持式计算装置（如 PDA）中的计算环境。不过，应理解的是本发明不局限于 PDA。本发明可以应用于其它的计算机系统构造中，包括多处理系统、基于微处理器或者可编程的消费者电子设备、网络 PC、小型机、大型机、无线电话、寻呼机等等。本发明也可以在分布式计算环境中实践，其中，任务由通过通信网络连接的远程处理设备来执行。在分布式计算环境中，程序模块可以位于本地和远程存储器存储设备中。

一般而言，程序模块包括例程、程序、对象、组件、数据结构，以及执行特定的任务或实现特定抽象数据类型的其它类型的结构。此外，熟悉本技术领域的人们会赞同，本发明可应用于其它的计算机系统构造，包括桌面计算系统、多处理系统、基于微处理器或者可编程的消费者电子设备、小型机、大型机等等。本发明也可应用于分布式计算环境，其中，任务由通过通信网络连接的远程处理设备来执行。在分布式计算环境中，程序模块可以位于本地和远程存储器存储设备中。

参见图 1，执行本发明的一个示例性系统包括手持式装置 100。计算装置 100 有处理器 102、存储器 104、显示器 106，以及可能包括用户界面按钮和/或键盘输入设备 108。存储器 104 一般包括易失性存储器（例如 RAM）和非易失性存储器（例如 ROM、PCMCIA 卡等等）。操作系统 110 常驻于存储器 104，并在处理器 102 上执行。手持式计算装置 100 包括操作系统 110，诸如来自 MICROSOFT CORPORATION 的“WINDOWS MOBILE 2003”操作系统、来自 PALMSOURCE 的“PALM OS”操作系统，或者适用于手持式装置或台式计算机的其它操作系统。

可将一个或多个应用程序 112 载入存储器 104，并在操作系统 110 上运行。

应用程序的例子包括电子邮件程序、日程安排程序、PIM（个人信息管理）程序、字处理程序、电子数据表程序等等。另一个应用程序 112 的例子是网络浏览器程序模块，诸如来自 MICROSOFT CORPORATION 的“POCKET INTERNET EXPLORER”网络浏览器应用程序或者其它网络浏览器。

手持式计算装置 100 有电源 114，它具体实现为一个或者多个电池。电源 114 可还包括一个外部电源，它取代内置电池或者给内置电池充电，诸如 AC 适配器或者有源船坞式支架。手持式计算装置 100 还可包括声频发生器 116，它用来产生声频输出，以在手持式装置 100 所包含的扬声器上播放。每个传统的手持式计算装置 100 的部件 102-116 可通过总线 118 互连，以传输数据和电能信号。

如上简要所述，手持式计算装置 100 包括显示器 106。显示器 106 可包括背光薄膜晶体管液晶显示器，或适于在手持式装置中工作的其它类型显示器。亦如上简要所述，显示器 106 可配备有触摸屏功能。通过使用指示笔或者其它指示设备，用户能通过触摸显示器 106 的所要部分对手持式装置 100 进行输入。然后，用户所触摸的屏幕坐标被返回到操作系统 110 以进行处理。因此，应理解的是，显示器可向手持式装置提供输入和输出功能。

现请参见图 2A 和图 2B，将详细描述其它有关用于本发明各种实施例的显示器 106 的更多的详细信息。如图 2A 和图 2B 所示，在手持式装置中使用的显示器 106 根据所用的显示板类型可构造成具有不同的像素密度。例如，如图 2A 所示，可将一板用于具有 240 水平方向像素和 320 垂直方向像素的像素分辨率的显示器 106。图 2A 中的显示器 106 的尺寸可为垂直方向 3.33 英寸，水平方向 2.5 英寸。在这些尺寸和像素分辨率下，图 2A 所示的显示器 106 的像素密度为 96dpi。

图 2B 所示的显示器 106 具有比图 2A 所示的显示器 106 高的像素密度。特别是，尽管图 2B 所示的显示器 106 和图 2A 所示的显示器 106 的尺寸是相同的，但显示器 106 具有 480 水平方向像素和 640 垂直方向像素的分辨率。因此，图 2B 所示的显示器 106 的像素密度为 192dpi。应理解的是，具有 96dpi 和 192dpi 像素密度的显示器仅仅是说明性的，本发明的实施例可用于具有任何像素密度的显示器。

现请参见图 3A 和图 3B，将提供本发明的实施例所用的显示器 106 的更多的详细信息。图 3A 示出具有 96dpi 像素密度的显示器的屏幕视图。如图 3A 所示，屏幕由许多像素 122A-122C 组成。每个像素可被单独的寻址，以向显示器屏幕写入数据。例如，如图 3A 所示，可通过激活合适的像素在显示器屏幕上显示方块 120。

图 3B 示出显示器屏幕 106，它具有 192dpi 或者说两倍于图 3A 所示的显示器屏幕的像素密度。图 3B 亦示出了图 3A 所示的相同方块 120 的图像。即使用于画出方块 120 的像素数量是相同的，但由于像素密度的增加，图 3B 所示的方块 120 是图 3A 所示的方块的大小的四分之一。例如，当一个专门为操作具有低像素密度屏幕（如图 3A 所示）所写的应用程序在具有高像素密度显示器屏幕（如图 3B 所示）的装置上运行时，这就会发生。本发明的实施例将使得方块 120 或者任何其它屏幕上的物体，以与图 3A 所示的方块 120 相同的方式显示在图 3B 所示的显示器 106 上。

现转到图 4A 和 4B，将描述几个软件组件，它们使得应用程序与具有增大像素密度的显示装置兼容。特别是，图 4A 示出了用于由一应用程序 112A 调用 API 以实现显示器屏幕 I/O 的软件组件，该应用程序编制为专门用在具有较高像素密度显示器的装置上。由于应用程序 112A 专门用于较高像素密度的显示器，调用 API 124 或者从 API 124 调用可以直接进行，并由操作系统 110 处理，而无需任何其它处理。

不过，如下面将更加详细地描述的，当接收 API 调用时，可以判断调用应用程序是否专门构造成用于具有较高像素密度的显示装置上。如果应用程序是用在较高像素密度的显示装置上的，无需其它处理来使得它和上述的较高像素密度显示器兼容。然而，如果应用程序不是专门给较高像素密度的显示器使用的，则需要额外的处理来保证输入输出兼容于较高像素密度显示器。如图 4B 所示。

如图 4B 所示，一个不是用于较高像素密度显示装置的旧应用程序 112B 可调用 API 124 来实现屏幕的输入输出操作。双向转换层 126 截获 API 调用，并判断应用程序是否构造成用于较高像素密度显示装置。既然旧应用程序 112B 不是用于较高像素密度显示装置，转换层将 API 调用的像素参数缩放至较高像素密度显示器。例如，如果旧应用程序 112B 是用于 96dpi 的显示器的，而手持式装置 100

实际装有 192dpi 的显示器,那么转换层 126 将把从 API 调用接收到的像素坐标翻倍。接着,转换层 126 使用缩放后的参数调用 API 124,以在较高的像素密度下执行所要求的操作。类似的,如果参数是从 API 124 返回的,诸如执行输入操作时,返回参数也可被正确的缩放。应理解的是,图 4A 和图 4B 所示的 API 124 是相同的,没有对 API 124 作任何的改变。图 5 及其相关的讨论给出有关通过转换层 126 所执行的处理更多详细信息,该转换层用于截获 API 调用并缩放所接收的参数。

现请参见图 5,将描述说明性例程 500,它示出使得应用程序和具有增大像素密度的显示器兼容的处理过程。当阅读此处给出例程的讨论时,应理解的是,本发明的不同实施例的逻辑操作是(1)作为在计算系统上的一系列计算机可执行行为或者程序模块来实现的,和/或(2)作为在计算系统内相互连接的机器逻辑电路或电路模块来实现的。该实现根据实现本发明的计算系统的性能需求来进行选择。因此,图 5 所示的并构成这里所描述的本发明实施例的逻辑操作被不同的称为操作、结构装置、行为和模块。熟悉本技术领域的人们会公认,这些操作、结构装置、行为和模块可用软件、用固件、专用数字逻辑电路、或者它们的任何组合中来实现,而不超出所附权利要求书限定的本发明精神和保护范围。

例程 500 从操作 502 开始,转换层 126 在操作 502 处判断是否已从应用程序 112 接收到指向 API 124 以执行屏幕的输入或输出的调用。如果或没有接收到这样的调用,例程 500 转回操作 502,在那里做另一这样的决定。如果已从应用程序 112 对 API 124 进行调用,那么例程 500 从操作 502 继续执行到操作 504。

在操作 504 处,转换层 126 截获来自应用程序 112 的 API 调用。例如,可以通过用转换层 126 的地址来取代地址表中的每个屏幕 API 124 调用来实现。用这种方式,对屏幕 API 124 函数的调用被替代地指向转换层 126。接着,例程从操作 504 继续执行到操作 506。

在操作 506 处,转换层 126 判断施行调用的应用程序是否构造成用于增大像素密度的显示器。转换层可执行这个过程,例如,通过检查应用程序的一个属性来判断应用程序 112 被编译时的软件开发工具包(“SDK”)的版本号。这个信息可在编译时被包含在应用程序 112 中。基于 SDK 的版本号,转换层 126 能判断应用程序 112 是构造成用于低像素密度的显示装置还是用于高像素密度的显示装

置。拥有大于或者等于预定编号的版本号的应用程序被假定为支持较高像素密度的显示器。拥有低于预定编号的版本号的应用程序被假定为仅支持低像素密度的显示器。也可使用其它方法来判断应用程序 112 是否构造成用于较高像素密度的显示装置。

从操作 506 开始, 例程 500 继续执行到判别操作 508, 在那里根据应用程序 112 是构造成用于高像素密度的显示装置还是用于低像素密度的显示装置而进行分支。如果应用程序 112 构造成用于高像素密度的显示器, 则例程 500 继续执行操作 510, 在那里不对被传递参数作任何改变地将 API 调用传递到所要的 API。由于应用程序 112 已专门地构造成用于较高像素密度的显示器, 对 API 124 的调用和来自 API 的调用能直接进行并被操作系统 110 处理, 而不经任何额外的处理。从操作 510 开始, 例程 500 返回到操作 502, 在那里可处理其它的 API 调用。

在判别操作 508 处, 如果判定施行调用的应用程序 112 不是构造成用于高像素密度的显示器的, 则例程 500 由判别操作 508 转向操作 512。在操作 512 处, 双向转换层 126 截获 API 调用。在操作 514 处, 双向转换层 126 将从 API 调用接收的像素参数缩放至较高像素密度的显示装置。接着, 在操作 514 处, 转换层 126 用缩放后的参数来调用所要的 API。接着, API 124 用缩放后的参数在较高像素密度的显示装置上执行所要的函数。

例程 500 从操作 514 继续执行到操作 516, 在那里转换层 126 判断 API 124 是否有像素值要返回给施行调用的应用程序 112, 例如在执行输入操作时就有这种情况。如果没有像素值要被返回给应用程序 112, 则例程 500 返回到操作 502 处, 在那里可截获和处理另一个 API 调用。如果要将值返回, 则例程 500 继续执行到操作 518, 在那里转换层将返回的像素值缩放至较低像素密度显示器的像素密度。然后, 例程 500 继续执行到 520, 在那里转换层将缩放后的返回值返回给应用程序 112。接着, 例程 500 返回到操作 502, 在那里可截获和处理另一个 API 调用。

应予理解的是, 尽管此处描述本发明的实施例是在手持式计算装置的背景中提供的, 但本发明的各方面可用于台式机、膝上型电脑、或者其它类型的计算机系统, 以保证与较高像素密度显示器的向后兼容性。而且, 本发明的实施例描述为从低像素密度到高像素密度的参数缩放, 本发明也可用于从高像素密度向低像素密度缩放。以这种方式, 仅构造成用于高像素密度显示器的应用程序可不加改变

地用于低像素密度的显示屏幕上。

基于上述内容，应理解的是，本发明的不同实施例包括方法、系统、设备或者计算机可读媒体，用于使得应用程序与具有增大像素密度的显示装置相兼容。上面的说明书、例子和数据提供了对本发明成分的制造和使用的完整说明。由于可以不超出本发明精神和保护范围地作出许多本发明的实施例，则由其后的权利要求书来限定本发明。

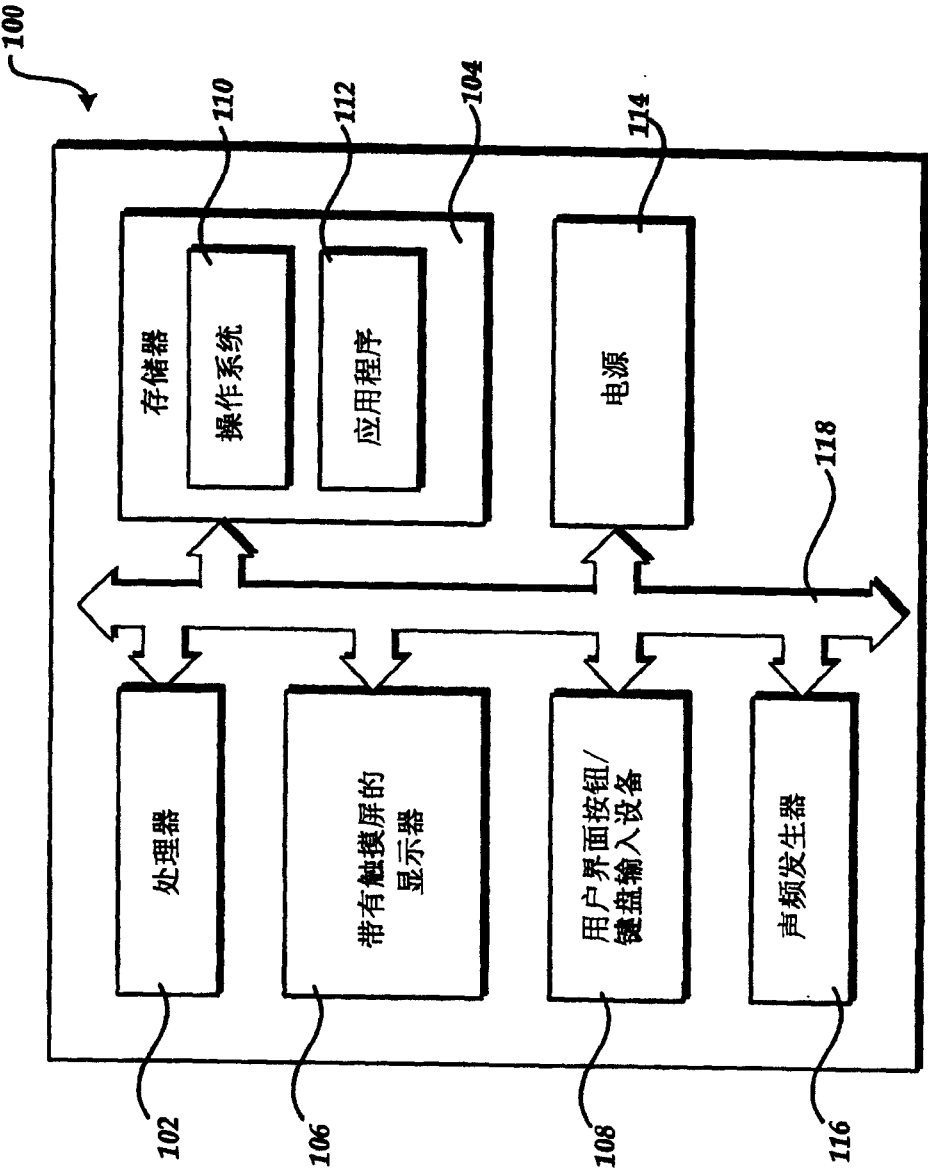


图 1

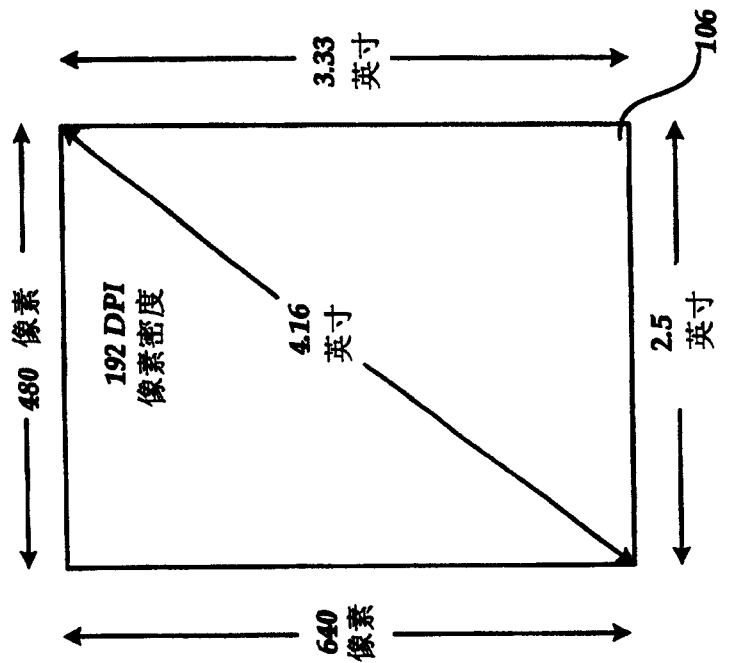


图 2A

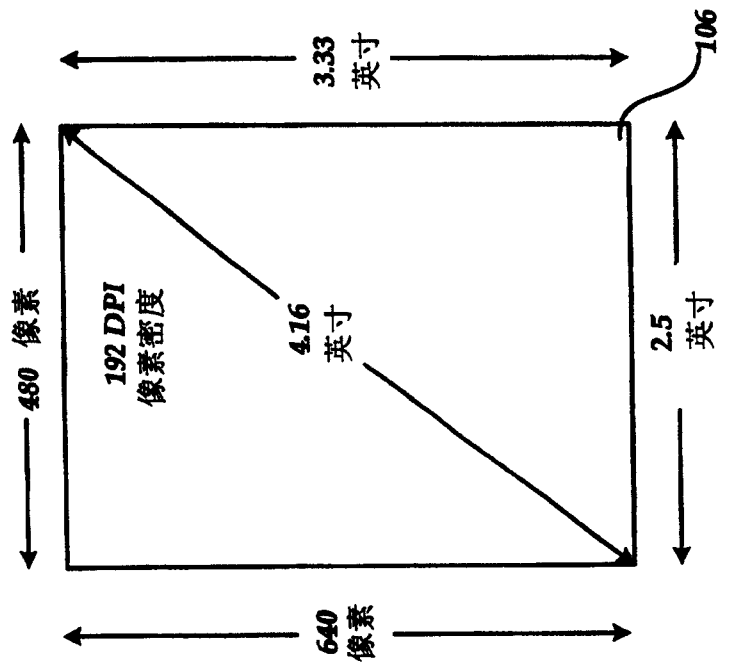


图 2B

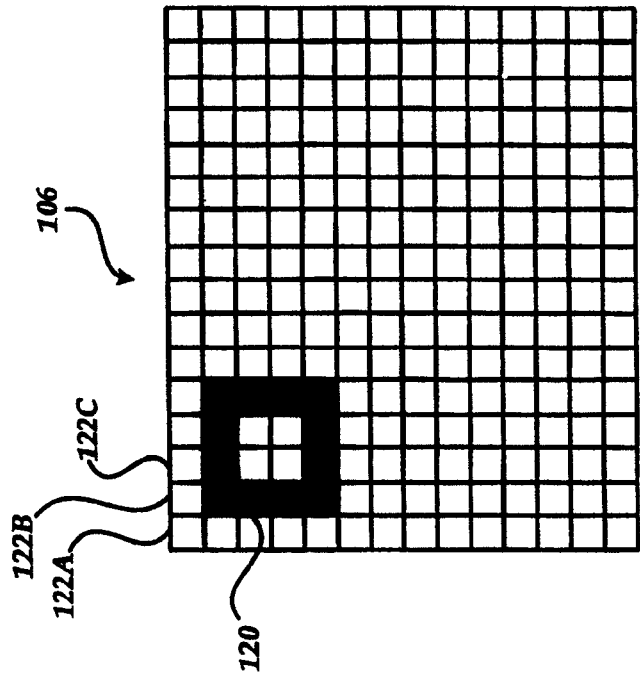


图 3B

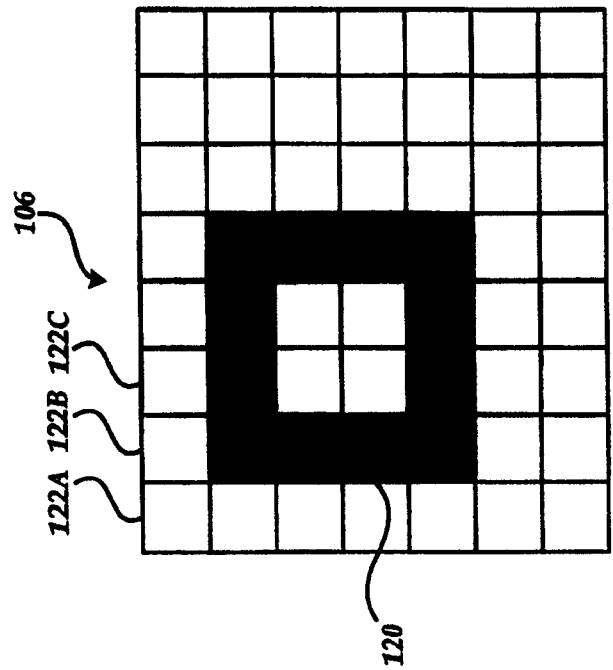


图 3A

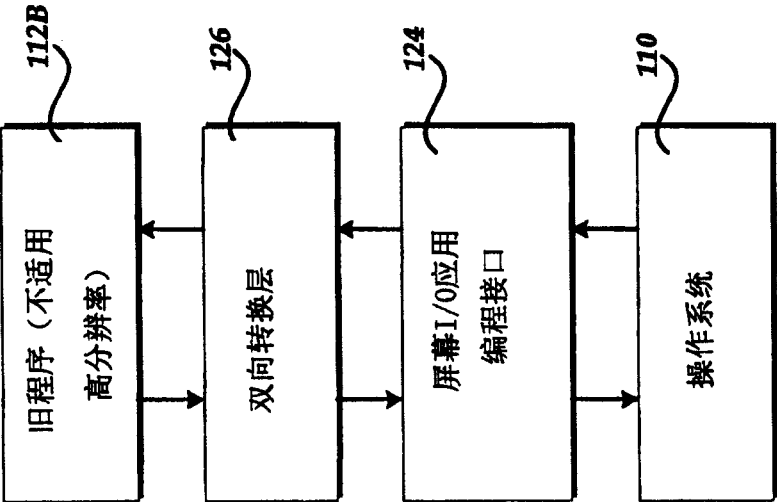


图 4B

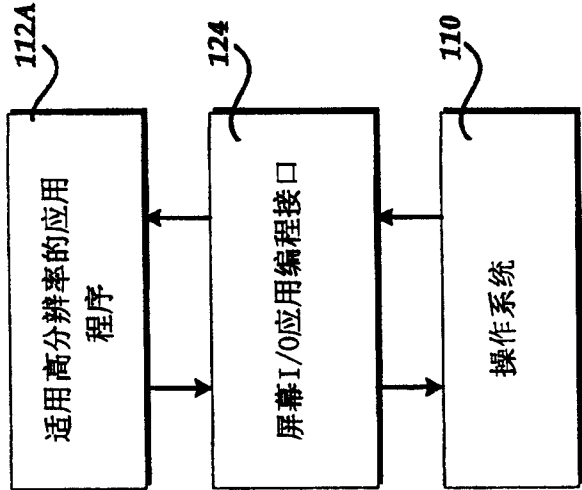


图 4A

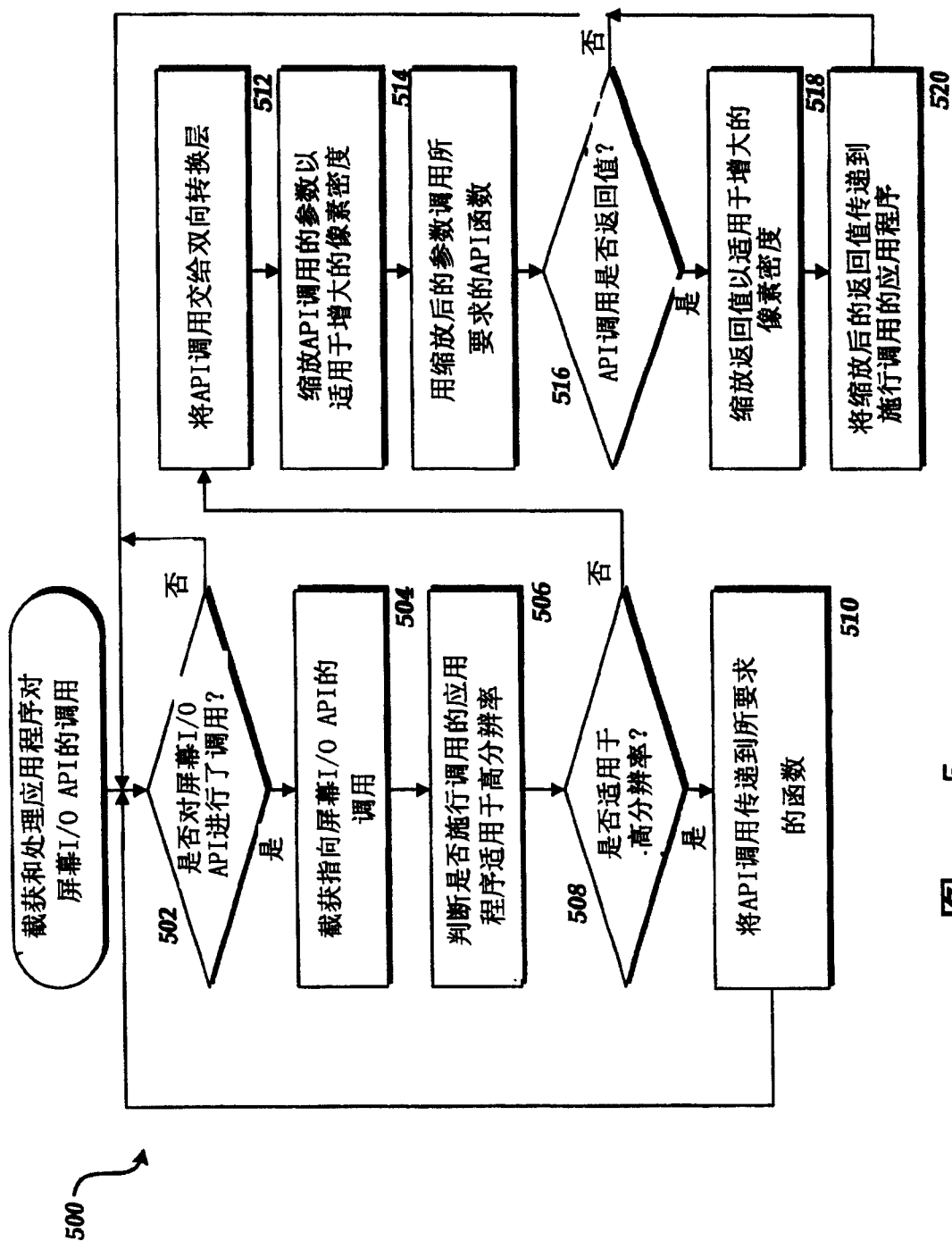


图 5