

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06T 1/20 (2006.01)



[12] 发明专利说明书

专利号 ZL 01812700.2

[45] 授权公告日 2006 年 10 月 25 日

[11] 授权公告号 CN 1282127C

[22] 申请日 2001.7.19 [21] 申请号 01812700.2

[30] 优先权

[32] 2000. 7. 20 [33] US [31] 09/620,355

[86] 国际申请 PCT/GB2001/003226 2001. 7. 19

[87] 国际公布 WO2002/009083 英 2002. 1. 31

[85] 进入国家阶段日期 2003. 1. 13

[71] 专利权人 国际商业机器公司

地址 美国纽约

[72] 发明人 R·P·德施纳奥克斯 国 维

审查员 黄 捷

[74] 专利代理机构 北京市中咨律师事务所

代理人 李 峥 于 静

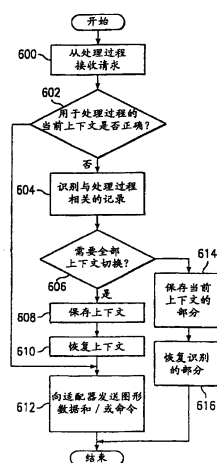
权利要求书 4 页 说明书 13 页 附图 3 页

[54] 发明名称

在数据处理系统中用于动态图形上下文切换的方法和装置

[57] 摘要

一种在数据处理系统中用于多个图形处理过程的上下文管理的方法和装置。接收来自该多个图形处理过程的第一图形处理过程的向图形适配器发送图形数据的请求。判断对该图形适配器处理来自该第一图形处理过程的图形数据而言是否需要当前上下文进行完全改变。如果仅需对当前上下文的一部分进行改变，则保存要被改变的一部分。然后使用存储在存储器中的一组上下文检索用于该图形处理过程的上下文信息，并用需要的当前上下文信息置入。



1. 一种在数据处理系统中用于处理来自多个图形处理过程的数据的方法:

响应于接收到来自该多个图形处理过程中的一个图形处理过程的向图形适配器发送图形数据的请求, 判断是否需要完全改变该图形适配器中的当前上下文, 以处理用于来自该多个图形处理过程的该图形处理过程的图形数据; 以及

在不需要当前上下文的完全改变的情况下, 使用来自存储在存储器中的一组上下文的在该图形适配器中相关的上下文改变当前上下文的一部分, 其中该相关上下文与该图形处理过程相关。

2. 根据权利要求1所述的方法, 其特征在于, 该上下文包括纹理映射和颜色查找。

3. 根据权利要求1所述的方法, 其特征在于, 该当前上下文是用于二维图形环境。

4. 根据权利要求1所述的方法, 其特征在于, 该当前上下文是用于三维图形环境。

5. 根据权利要求1所述的方法, 其特征在于, 该存储器为缓冲器。

6. 根据权利要求5所述的方法, 其特征在于, 该缓冲器存储一组上下文。

7. 根据前面任何一项权利要求所述的方法, 其特征在于, 该多个图形处理过程可访问该组上下文。

8. 根据权利要求1至6任意一项所述的方法, 其特征在于, 进一步包括:

在恢复相关上下文之前保存该当前上下文。

9. 根据权利要求1至6任意一项所述的方法, 其特征在于, 该当前上下文与该多个图形处理过程中的另外一个图形处理过程相关。

10. 一种在数据处理系统中用于多个图形处理过程的上下文管理的

方法，该方法包括：

接收来自多个图形处理过程的第一图形处理过程的向一个图形适配器发送图形数据的请求；

判断对于该图形适配器处理来自该第一图形处理过程的图形数据来说是否需要小于当前上下文的全部的改变，从而形成一个确定要改变的上下文的量；以及

如果需要少于当前上下文的全部被改变，使用存储在存储器中的一组上下文，将该确定要改变的上下文的量的当前上下文改变为另一上下文。

11. 根据权利要求 10 的方法，其特征在于，进一步包括：

如果判断对于该图形适配器来说小于当前上下文的全部的改变不足以处理来自该第一图形处理过程的图形数据，则改变当前上下文的全部。

12. 根据权利要求 11 的方法，其特征在于，进一步包括：

向该图形适配器发送图形数据。

13. 根据权利要求 11 或 12 所述的方法，其特征在于，该改变当前上下文的全部的步骤包括：

使用该组上下文由又一个上下文替换该当前上下文。

14. 根据权利要求 10-12 任何一项所述的方法，其特征在于，在该多个图形处理过程中的一个图形处理过程将上下文信息写入与该图形处理过程相关联的上下文。

15. 一种用于处理来自多个图形处理过程的数据的数据处理系统，包括：

判断装置，响应于接收到来自该多个图形处理过程中的一个图形处理过程的向图形适配器发送图形数据的请求，用于判断是否需要图形适配器中的当前上下文的完全改变以处理用于来自该多个图形处理过程的该图形处理过程的图形数据；以及

改变装置，用于在不需要该当前上下文的完全改变的情况下，使用来自存储在存储器中的一组上下文的在该图形适配器中相关的上下文改变该当前上下文的一部分，其中该相关上下文与该图形处理过程相关联。

16. 根据权利要求 15 所述的数据处理系统，其特征在于，进一步包括：

保存装置，用于在恢复该相关上下文之前保存该当前上下文。

17. 根据权利要求 15 的数据处理系统，其特征在于，该当前上下文与该多个图形处理过程中的另外一个图形处理过程相关。

18. 一种用于多个图形处理过程的上下文管理的数据处理系统，该数据处理系统包括：

接收装置，用于接收来自该多个图形处理过程的第一图形处理过程的向图形适配器发送图形数据的请求；

判断装置，用于判断对于图形适配器是否需要少于当前上下文的全部的改变以处理来自该第一图形处理过程的图形数据，从而形成一确定要改变的上下文的量；

改变装置，用于如果少于当前上下文的全部被改变，使用存储在存储器中的一组上下文将该确定要改变的上下文的量的当前上下文改变为另一上下文。

19. 根据权利要求 18 所述的数据处理系统，其特征在于，进一步包括：

如果判断为对于图形适配器来说小于当前上下文的全部的改变不足以处理来自该第一图形处理过程的图形数据，用于改变当前上下文的全部的另一改变装置。

20. 根据权利要求 19 所述的数据处理系统，其特征在于，进一步包括：

发送装置，用于向图形适配器发送图形数据。

21. 根据权利要求 19 或 20 所述的数据处理系统，其特征在于，该另一改变装置包括：

替换装置，用于使用该组上下文以又一个上下文替换该当前上下文。

22. 根据权利要求 18 所述的数据处理系统，其特征在于，该改变装置包括：

替换装置，用于以来自该组上下文的上下文信息替换该当前上下文的一部分从而形成该另一上下文。

在数据处理系统中用于动态图形上下文切换的方法和装置

技术领域

本发明一般地涉及一种改进的数据处理系统，具体地涉及用于处理图形数据的方法和装置。

背景技术

数据处理系统，例如个人计算机和 workstation，通常被用来运行计算机辅助设计（CAD）应用、计算机辅助制造（CAM）应用和计算机辅助软件工程（CASE）工具。工程师、科学家、技师以及其他人员每天使用这些应用。这些应用涉及复杂的计算，例如有限元分析，用来模型化结构中的应力。其他一些应用包括化学的或分子建造模型应用。CAD/CAM/CASE 应用就传递给用户的信息而言通常是图形密集的。数据处理系统用户还可能使用其他的图形密集型的应用，例如桌面出版应用。通常，这些应用的用户需要或要求该数据处理系统能够提供极快的图形信息。

用于在视频显示终端上提供图形显示的图象数据流的处理需要极快的图形系统以向显示器提供快速响应。在这些类型的图形系统中，通过接收基元（primitive）来进行处理和显示。基元是被用作创建图像的建造块的图形元素，例如点、线、弧、圆锥或球体。基元可以由包括一个或多个顶点的顶点组来定义。属性被用来定义如何显示基元。所述属性包括，例如，线型，颜色和表面纹理（texture）。顶点定义一个点，一条边的端点，或多边形两个边相交形成的角。包括例如位置坐标、颜色、法线（normals）及纹理坐标这样一些信息的数据也与顶点相关联。向该图形系统发送命令以定义如何对基元和其他数据进行处理以用于显示。

在处理图形数据中涉及大量的数据和计算，尤其是对三维应用而言，这些计算中的许多计算已被从中央处理单元转移到图形适配器中。在这些

图形系统中，使用位于该图形适配器中的图形流水线（pipeline）来处理图形数据。使用流水线技术，可将图形数据处理分割为若干级处理元素（element）的阶段，其中可由不同处理元素顺序执行对数据的处理。

在多任务图形环境中，常常是多个处理过程共享相同的图形适配器。为使每个图形处理过程向该图形适配器发送图形数据，每个处理过程需要一个绘制上下文（rendering context）。在从一个特定的图形处理过程将图形数据发送至图形适配器用于显示之前，该绘制上下文被一上下文切换处理器恢复在图形适配器上。该上下文切换处理器为图形处理过程执行上下文保存和恢复功能，以便每个图形处理过程都能够访问该图形适配器。用于不同图形处理过程的上下文可能有很大的不同。例如，三维图形环境与二维图形环境相比就相当大。当前，对所有的图形处理过程来说，该上下文被设定为相同的大小。对每个处理过程都要切换整个上下文的做法降低了用于显示图形数据的处理速度。而且，该机制既浪费适配器中的存储器，还降低了不同处理过程中上下文的切换速度。

发明内容

本发明提供了一种数据处理系统中用于多个图形处理过程的上下文管理的方法和装置。从该多个图形处理过程中的第一图形处理过程接收到将图形数据发送至一个图形适配器的请求。判断对于图形适配器是否需要全部改变当前上下文以处理来自该第一图形处理过程的图形数据。如果仅是当前上下文的一部分需要被改变，则保存要被改变的部分。然后，使用存储在存储器中的一组上下文，检索用于该图形处理过程的上下文信息，并以所需的当前上下文信息置入。

第一方面，本发明提供了一种在数据处理系统中用于处理来自多个图形处理过程的数据的方法：响应于接收到来自该多个图形处理过程中的一个图形处理过程的将图形数据发送至图形适配器的请求，判断是否需要完全改变该图形适配器中的当前上下文，以处理用于来自该多个图形处理过程的该图形处理过程的图形数据；以及在不需完全改变该当前上下文的

情况下，使用来自存储在存储器中的一组上下文的在该图形适配器中相关的上下文改变该当前上下文的一部分，其中，该相关上下文与该图形处理过程相关联。

第二方面，本发明提供了一种在数据处理系统中用于多个图形处理过程的上下文管理的方法，该方法包括：接收来自该多个图形处理过程的第一图形处理过程的向一个图形适配器发送图形数据的请求；判断对于该图形适配器处理来自该第一图形处理过程的图形数据来说是否需要小于当前上下文的全部的改变，从而形成一个确定要改变的上下文的量；以及如果需要少于当前上下文的全部被改变，使用存储在存储器中的一组上下文，将该确定要改变的上下文的量的当前上下文改变为另一上下文。

第三方面，本发明提供了一种用于处理来自多个图形处理过程的数据的数据处理系统，包括：判断装置，响应于接收到来自该多个图形处理过程中的一个图形处理过程的向图形适配器发送图形数据的请求，用于判断是否需要完全改变该图形适配器中的当前上下文以处理用于来自该多个图形处理过程的该图形处理过程的图形数据；以及改变装置，用于在不需要完全改变该当前上下文的情况下，使用来自存储在存储器中的一组上下文的在图形适配器中相关的上下文改变该当前上下文的一部分，其中，该相关上下文与该图形处理过程相关联。

第四方面，本发明提供了一种用于多个图形处理过程的上下文管理的数据处理系统，该数据处理系统包括：接收装置，用于接收来自该多个图形处理过程的第一图形处理过程的向图形适配器发送图形数据的请求；判断装置，用于判断对于图形适配器是否需要少于当前上下文的全部的改变以处理来自该第一图形处理过程的图形数据，从而形成一确定要改变的上下文的量；以及改变装置，用于如果少于当前上下文的全部被改变，使用存储在存储器中的一组上下文将该确定要改变的上下文的量的当前上下文改变为另一上下文。

附图说明

在下文中将结合以下附图更详细地描述本发明的优选实施例。

图 1 示出根据本发明一优选实施例可在其中实施本发明的一种数据处理系统；

图 2 为在其中可实施本发明的数据处理系统的方块图；

图 3 为示出用于根据本发明一优选实施例的动态上下文切换中的组成部分的方块图；

图 4 图示了根据本发明一优选实施例的 `com_record`；

图 5 为根据本发明一优选实施例的建立上下文的流程图；

图 6 为根据本发明一优选实施例的用于上下文切换的处理过程的流程图；

图 7 为根据本发明一优选实施例的存储上下文信息的流程图；

图 8 为根据本发明一优选实施例的恢复一处理过程的上下文的流程图。

具体实施方式

现在参考附图，特别参考图 1，该图示出根据本发明一优选实施例的其中可实施本发明的数据处理系统。图中所示的计算机 100 包括系统单元 110，视频显示终端 102，键盘 104，存储装置 108，该存储装置可包括软驱和其他类型的永久和可拆卸存储介质，以及鼠标 106。个人计算机 100 还可包括一些附加的输入装置，例如游戏杆，触摸板，触摸屏，跟踪球，麦克风等等。计算机 100 可以使用任何适合的计算机实现，例如 IBM RS/6000 计算机或 IntelliStation 计算机，它们是 IBM 公司的产品，该公司位于纽约的 Armonk。尽管图中所示的为一计算机，在本发明的其他实施例中，也可以被实现在其他类型的数据处理系统中，例如网络计算机。计算机 100 优选地包括一图形用户界面，其可借助于驻留在计算机可读媒体中在计算机 100 内处于运行状态的系统软件来实现。

现在参考图 2，图中示出在其中可以实现本发明的数据处理系统的方块图。数据处理系统 200 以计算机，例如图 1 中的计算机 100 为例，在其中可存储实现本发明的处理过程的代码或指令。数据处理系统 200 采用外围组件扩展接口（PCI）本地总线结构。尽管图示的例子采用的是 PCI 总线，也可以使用其他的总线结构，例如加速图形接口（AGP）和工业标准结构（ISA）。处理器 202 和主存储器 204 通过 PCI 桥 208 连接至 PCI 本地总线 206。PCI 桥 208 还可包括集成的存储器控制器和用于处理器 202 的高速缓冲存储器。到 PCI 本地总线 206 的其他连接可通过直接组件互连或通过内插式附件（add-in）板来完成。在图示例子中，局域网（LAN）适配器 210、小型计算机系统接口 SCSI 主机总线适配器 212 以及扩展总线接口 214 均通过直接组件连接与 PCI 本地总线 206 连接。相反，音频适配器 216，和图形适配器 218 则通过插入扩展槽中的内插式附板连接至 PCI 本地总线 206。扩展总线接口 214 提供用于键盘和鼠标适配器 220、调制解调器 222 和附加存储器 224 的连接。SCSI 主机总线适配器 212 提供用于硬盘驱动器 226、磁带驱动器 228 和 CD-ROM 驱动器 230 的连接。典型的

PCI 本地总线实现可支持 3 或 4 个 PCT 扩展槽或内插式连接器。

操作系统在处理器 202 上运行，并被用于协调和提供如图 2 所示的数据处理系统 200 内的各个组成部分的控制。该操作系统可以是商业可用的操作系统，例如可从微软公司获得的 Windows2000。面向对象的编程系统，例如 Java，可结合该操作系统来运行，并且可提供从在数据处理系统 200 上执行的 Java 程序或应用对该操作系统的调用。“Java”是 Sun 微系统公司的商标。操作系统的指令、面向对象的操作系统、以及应用或程序均被置于存储装置，例如硬盘驱动器 226 中，并且可以被调入主存储器 204 中由处理器 202 来执行。具体地，本发明提供了一种机制，用于为向诸如图形适配器 218 的图形适配器发送图形数据的不同的处理过程动态地切换上下文。这一优点是通过分配存储器以存储用于共享图形适配器 218 的不同处理过程的上下文来获得的。

本领域一般技术人员将会理解图 2 中的硬件可根据实施的不同而改变。除图 2 所示的硬件外，还可使用其他的内部硬件或外围设备，例如闪速（flash）ROM（或等同的非易失性存储器）或光盘驱动器等等，或由这些硬件或设备取代图 2 所示的硬件。而且，本发明的处理过程还可应用到多处理器数据处理系统。

例如，如果可选地将数据处理系统 200 配置为网络计算机，则可不包括 SCSI 主机总线适配器 212、硬盘驱动器 226、磁带驱动器 228、和 CD-ROM 230，如图 2 中指示选择配置的点划线 232 所示。在这种情况下，计算机，确切地应称之为客户计算机，必须包括某些类型的网络通信接口，例如 LAN 适配器 210，调制解调器 222 等等。作为另外一个例子，数据处理系统 200 可以是被配置为不依赖于某种类型的网络接口也可启动的独立的系统，而不管数据处理系统 200 是否包括某种类型的网络通信接口。作为再另外一个例子，数据处理系统 200 可以是个人数字助理（PDA）设备，被配置有 ROM 和/或闪速 ROM，以便提供非易失性存储器用于存储操作系统文件和/或用户产生的数据。

图 2 所示的例子和上述的例子并非意味着结构的限制。例如，除了是

PDA 形式以外，数据处理系统 200 还可以是笔记本计算机或手提计算机。数据处理系统 200 还可以是便利站(kiosk)或 Web 设备(Web appliance)。

本发明的处理过程由处理器 202 使用计算机执行的指令来完成，这些指令可位于一存储器中，例如主存储器 204，存储器 224 中，或位于一个或多个外围设备 226-230 中。

本发明提供了一种多任务图形环境中的机制，在该环境中多个图形处理过程共享共同的图形适配器。在数据处理系统中的一部分存储器，例如图 2 中的存储器 204，被分配用于上下文切换。在这些例子中，存储器分配呈现固定(pinned)缓冲器的形式。固定缓冲器是不会被交换到磁盘的缓冲器，例如对虚拟存储器计算机系统而言，在该系统中磁盘空间被用作扩展存储器，用于存储最近没有被使用的信息。使用固定缓冲器或其他固定存储器使得设备驱动器能够快速访问该存储器中的信息，而无须从磁盘驱动器换入(swap in)该信息。

图形处理过程能够访问该固定缓冲器以指定是否需要某些预定的功能或动作。这些功能包括：例如纹理映射或颜色查找。其他功能包括：例如关于深度缓冲器测试的信息；硬件光源；凹凸映射(bump-mapping)，其为纹理映射的变形；和抗混淆(anti-aliasing)操作。

当发生上下文切换时，上下文切换机制读取固定缓冲器以决定是需要全部上下文切换还是需要进行部分上下文切换。上下文是在一时间点指定的信息或参数。在发出执行一操作的命令时可以不用重新声明或重新指定该信息，但该信息必然会影响该命令的执行。例如，一个程序可能被用来用绿色画一组线，其中，颜色为上下文。第二个程序可能被用来用红色画第二组线。当第一个程序运行时，颜色被设定为绿色。在程序 1 画完线条之前出现中断，程序 2 开始运行。程序 2 将颜色设定为红色并开始画线。第二次中断发生，程序 1 再继续执行。此时，颜色被设定为红色。程序 1 每次画线时不再重设颜色。结果，如果上下文，该情况下为线条的颜色，未被重设为程序 1 所使用的绿色，则线条会以错误的颜色绘出。在该例子中，使用设备驱动器来将该上下文恢复为绿色。这样，就不需要这些程序

来跟踪 (tract) 该上下文信息。

该信息还可被用来确定用于分配图形适配器中的存储器所需的图形上下文的大小。被当作上下文的其他信息包括, 例如是否使光源 (lighting) 或裁剪 (clipping) 可用; 是否绘制或填充基元; 当前线的点画模式; 以及当前纹理混合功能。

以下转到图 3, 其示出根据本发明一优选实施例的用于动态上下文切换的组件的方块图。在该例中, 应用 300, 302, 304 处于多任务图形环境中且共享单个图形适配器, 如图 2 所述的图形适配器 218。如图所示, 每个应用包括一图形处理过程, 其产生用于显示的图形数据。当然, 根据特定的应用, 一个应用可包括多个图形处理过程。这些应用向操作系统 308 中的设备驱动器 306 发送图形数据。设备驱动器是将外围设备, 例如图形适配器 218, 连接至操作系统的程序或例程。一个特定的设备驱动器接收请求和数据, 例如来自应用 300, 302, 304 的请求和数据, 并将这些请求转换成可由该外围设备识别的特定的命令语言和格式。

在该例子中, 应用 300, 302, 304 能够访问缓冲器 310 以写入或改变上下文信息。在该例子中, 该上下文信息采取 com_records 312 的形式。当这些应用中的一个向设备驱动器 306 发送图形数据用于在图形适配器上显示时, 设备驱动器 306 将查看缓冲器 310 中的 com_records 312 来看是否需要上下文切换。如果需要上下文切换, 则设备驱动器 306 还判断是需要部分上下文切换还是需要全部上下文切换。基于该判断, 设备驱动器 306 将有选择地切换该图形适配器中的上下文。作为部分切换该上下文, 设备驱动器 306 还可改变该图形上下文的大小以动态地改变图形适配器中用于该上下文的存储器的分配。这样, 当对于一特定上下文需要较少的存储空间时, 则在该图形适配器中可以使更多的存储空间以供它用。每个硬件都具有设备驱动器; 例如图形适配器。设备驱动器知道潜在的上下文信息, 且通常保存所有的上下文信息。一特定上下文所需的存储器的量可从用于该图形适配器的设备驱动器或从由该设备驱动器保存的信息获得。

在这些例子中设定一标志用来识别现用的上下文的片段或部分。设备

驱动器使用该标志来识别应被保存的上下文和不应保存的上下文。

现在参考图 4，该图图示了根据本发明一优选实施例的 `com_record`。`Com_record 400` 被用来识别用于一特定处理过程的上下文信息。通过使用 `Com_record 400`，本发明的机制能够在一个新的处理过程发送用于处理的图形数据时，判断是否需要全部的上下文切换。

该例子中的 `Com_record 400` 包括一标识，纹理和查找功能设定用于与 `Com_record400` 相关的处理过程。在这些例子中，每个处理过程与一个 `com_record` 相关联。`Com_record 400` 是包括几行上下文信息的例子。这些行是指示符或标志，用来识别由一特定处理过程当前使用的上下文。一个上下文通常包括成百上千的元素来识别该上下文。在创建上下文中，一处理过程在初始化时创建该上下文。一图形处理过程进行 API 调用以创建一个或多个图形上下文。当一特定上下文要被激活时，该图形处理过程进行 API 调用以使一特定上下文与一窗口相关联。这样，在该图形处理过程、上下文和窗口之间存在一种关系。通过由该设备驱动器提供的唯一的标识符来识别该上下文。该唯一的标识符通过被用来创建该上下文的 API 调用被提供给该应用。这些唯一的标识符被存储在 `com_record 400` 中。以这种方式，该设备驱动器便具有了每个上下文的唯一标识符和由 API 提供的关于当前激活使用的上下文部分的信息之间的唯一的配对。

该 API 在共享的存储器中创建 `com_record 400` 并且通过一次对设备驱动器的调用告诉该设备驱动器 `com_record400` 在共享存储器中的位置。之后，该设备驱动器查看共享存储器做出有关上下文切换的决定。如果该信息发生了改变，则该应用更新该共享存储器。例如，一个应用开始，并且一个应用初始化共享存储器并假定：`using_texture` 为 `FALSE`（假），`using_lookup` 为 `FALSE`（假）。接着，该应用调用该设备驱动器以告诉它该共享存储器的位置以及该共享存储器已准备好可以使用。然后，该设备驱动器进行上下文切换。当该设备驱动器保存该上下文时，它看到 `using_texture` 为 `FALSE`，于是该设备驱动器不存储纹理信息。该设备驱动器看到 `using_lookup` 为 `FALSE`，于是其不存储查找信息。当该设备驱动

器将该处理过程切换回活动模式时,该设备驱动器看到 `using_texture` 为假,因此该设备驱动器不恢复纹理信息。该设备驱动器看到 `using_lookup` 为假,则该设备驱动器不恢复查找表信息。该应用使用纹理查找开始。该应用将 `TRUE` (真) 写入称作 “`using_lookup`” 的共享存储器的片段中。设备驱动器切换掉 (switch off) 该上下文。该设备驱动器查看 `com_record` 中的共享存储器,看到 `using_texture` 为假,因此它不保存纹理信息。该设备驱动器看到 `using_lookup` 为真,因此设备驱动器保存查找信息。接着,设备驱动器切换回该上下文。设备驱动器查看共享存储器中的 `com_record` 并且看到 `using_texture` 为假,因此该设备驱动器不恢复查找信息。该设备驱动器看到 `using_lookup` 为真,因此,该设备驱动器恢复查找信息。

现在参考图 5,该图中示出了根据本发明一优选实施例的建立上下文的流程图。图 5 所示的处理过程以一应用,例如图 3 中的应用 300, 302, 304, 所使用的处理过程为例。

该处理过程以创建一个记录开始 (步骤 500)。在这些例子中,该记录体现为 `com_record` 的形式,例如图 4 中的 `com_record` 400。当然,该记录也可采取不同的形式,例如表中的条目或数据库中的记录。然后在该记录中设定预定的功能 (步骤 502)。在所示例子中,这些功能包括纹理和查找功能。在建立一基本上下文时,该设备驱动器接收一个以创建上下文的调用并使用缺省值或功能来建立该基本上下文。例如,在基本上下文中提供了缺省的线条颜色。当一上下文和一窗口之间的配对出现时,向该上下文加入附加的信息,例如宽度,高度和窗口类型。该信息被加至与一特定处理过程相关的上下文记录。

接着,由图形应用或处理过程进行图形操作 (步骤 504), 之后该处理过程终止。每次应用请求由设备驱动器创建一个上下文时,图形应用或处理过程创建一个唯一的记录。该信息存在直到该上下文被销毁或处理过程退出。

接下来转到图 6,该图是示出根据本发明的一优选实施例的用于上下文切换的流程图。示于图 6 中的处理过程可以在一设备驱动器,例如图 3

中的设备驱动器 306, 中实现。当然, 根据不同的实施也可在其他地方实现该处理过程。例如, 可在图形内核中实现该处理过程。

事件序列以接收到来自一个处理过程的将图形数据发送至图形适配器的请求开始(步骤 600)。判断当前上下文对提出请求的处理过程而言是否正确(步骤 602)。如果当前上下文不正确, 则识别与该处理过程相关的记录(步骤 604)。在这些例子中, 这些记录为位于固定缓冲器中的 `com_records`。

接下来, 判断是否需要进行全部的上下文切换(步骤 606)。如果需要进行全部的上下文切换, 则保存当前上下文(步骤 608), 并恢复该处理过程所需的上下文或将其置于适当的位置以供使用(步骤 610)。然后将该请求中的图形数据和/或命令发送至适配器(步骤 612), 之后该流程结束。

再来参考步骤 606, 如果不需要全部的上下文切换, 则保存当前上下文(步骤 614)。在步骤 614, 仅保存那些要被改变的当前上下文的部分。然后, 恢复被识别用于该处理过程的上下文的部分(步骤 616), 流程进行到步骤 612, 如上所述。换句话说, 检索需要被改变的上下文信息并将其置于当前上下文中以修改该当前上下文。回来参考步骤 602, 如果该图形适配器中的当前上下文对于发送该请求的处理过程而言是正确的, 如上所述, 则流程进行到步骤 610。

现在转到图 7, 其中示出根据本发明一优选实施例的用于保存上下文信息的处理过程的流程图。该处理过程在上下文切换期间被使用, 且在设备驱动器, 例如图 3 所示的设备驱动器 306, 中实现。

该处理过程开始先将基本上下文信息保存在与用于处理过程的上下文相关联的记录中(步骤 700)。判断在该上下文中是否已经设定了纹理(步骤 702)。如果已设定了纹理, 则将纹理信息存入该记录中(步骤 704)。接着, 判断在该上下文中是否设定了查找功能(步骤 706)。如果已设定了查找, 则保存该查找信息(步骤 708), 此后该处理过程结束。如果这两个功能均未设定, 则仅保存该基本上下文信息, 该处理过程终止。在这

个例子中，为说明目的，仅示出两类上下文功能。根据可能的不同类型的上下文，所有这些上下文都被保存到该记录中。

现在参考图 8，其中示出根据本发明一优选实施例的用于恢复一个处理过程的上下文的流程图。该处理过程以恢复基本上下文开始（步骤 800）。使用在与该处理过程相关的记录中的信息在该图形适配器中恢复该基本上下文。当一设备驱动器被调用来建立用于图形处理过程的上下文时，该基本上下文包括由该设备驱动器设定的缺省的设置。判断在该记录中是否设定了纹理功能（步骤 802）。如果设定了纹理功能，则使用该记录中的信息在该图形适配器中恢复用于该功能的纹理信息（步骤 804）。

接下来，判断在该记录中是否设定了查找功能（步骤 806）。如果设定了查找功能，则使用该记录在该图形适配器中恢复该查找信息（步骤 808），此后，该处理过程终止。如果在步骤 802，未设定纹理功能，如上所述，则该处理过程进行到步骤 806。如果在步骤 806 未设查找功能，则该处理过程终止。尽管该图仅示出除基本上下文信息外用于两个特定功能的上下文信息的恢复，但是本发明的机制可恢复用于在该记录中指定的任何功能的上下文。

图 7 和 8 示出的步骤既可应用于保存和恢复全部的上下文，也可应用于保存和恢复部分上下文。如果与一个选择的上下文相关的 `com_record` 指示并非使用该上下文中所有的元素，则发生部分保存。例如，如果对于图 4 的例子，`using_texture` 和 `using_lookup` 不是均被设定，则发生部分上下文保存。类似地，在恢复上下文信息时，如果该上下文的某些部分未被使用，则全部上下文恢复操作可以不是必须的。该机制体现在图 7 和 8 的步骤 702，706，802 和 806 中。

当上下文所需的空间的量改变时，则该上下文的大小也会被改变以分配新的空间。

由此，本发明提供了一种用于图形环境中使多个图形处理过程共享该相同的图形适配器的构制。该机制允许上下文切换，其中可基于用于图形处理过程的特定上下文来重新设定在一个图形适配器中使用的存储器的

量。此外，如果需要部分上下文切换，则仅在该图形适配器中恢复所需的信息。性能的提高是在其中发生部分上下文改变而不是全部上下文改变的情况下实现的。由于需要保存和恢复的信息较少，因此可节约时间。

应当注意，尽管是在全功能的数据处理系统的环境下对本发明进行了描述，本领域一般技术人员将会理解，可将本发明的处理过程以指令的计算机可读媒体的形式和多种形式分发，且不管实际用于进行分发的信号承载媒体的特定的类型如何，本发明均可同样适用。计算机可读媒体的例子包括可记录型媒体，例如软盘，硬盘驱动器，RAM，CD-ROM，DVD-ROM，以及传输型媒体，例如，数字和模拟通信链路，使用传输形式的有线或无线通信链路，例如射频和光波传输。该计算机可读媒体可采取编码形式，在一特定的数据处理系统中，其被解码用于实际的应用。

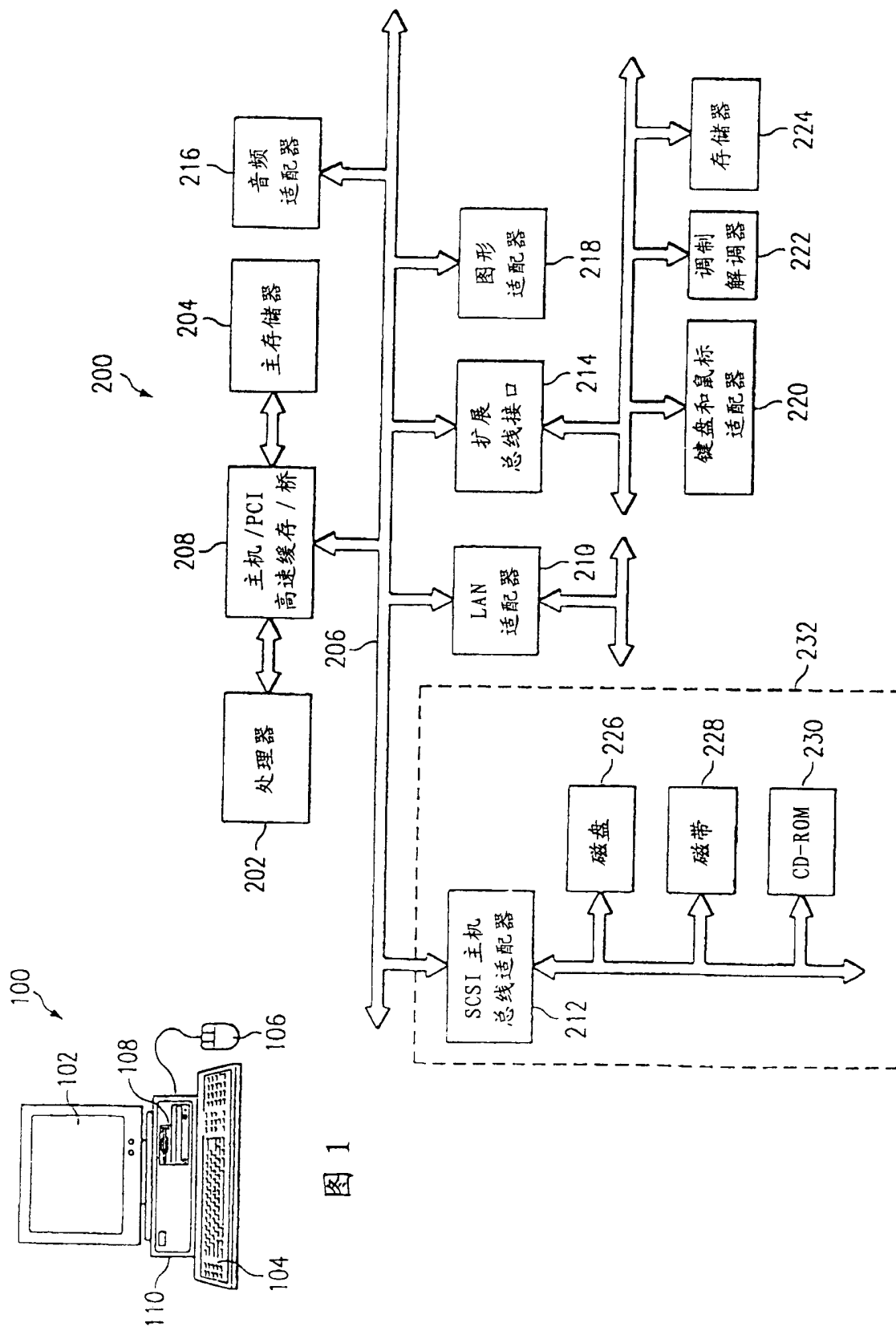


图 1

图 2

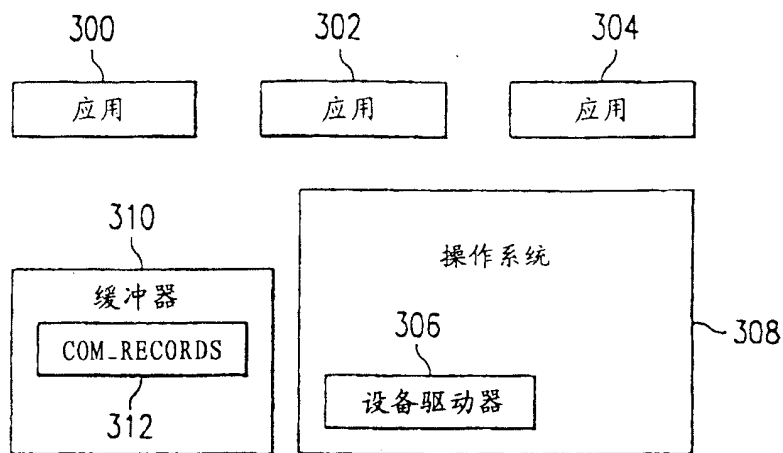


图 3

400

```

typedef struct comRecord{
    int using-texture;
    int using-lookup
} ComRecord
  
```

图 4

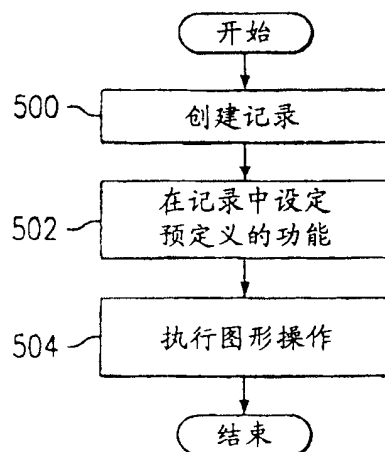


图 5

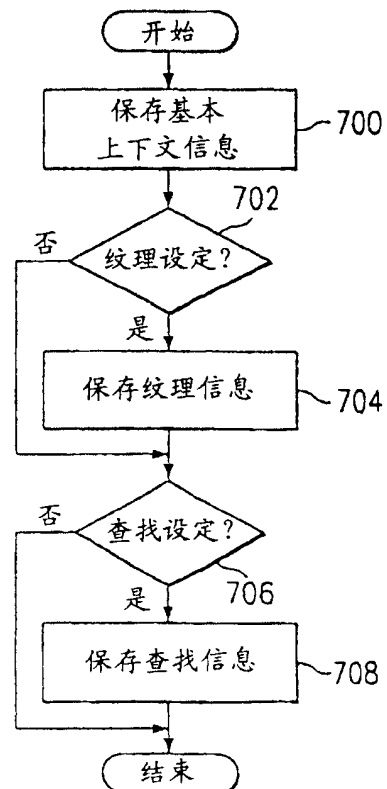


图 7

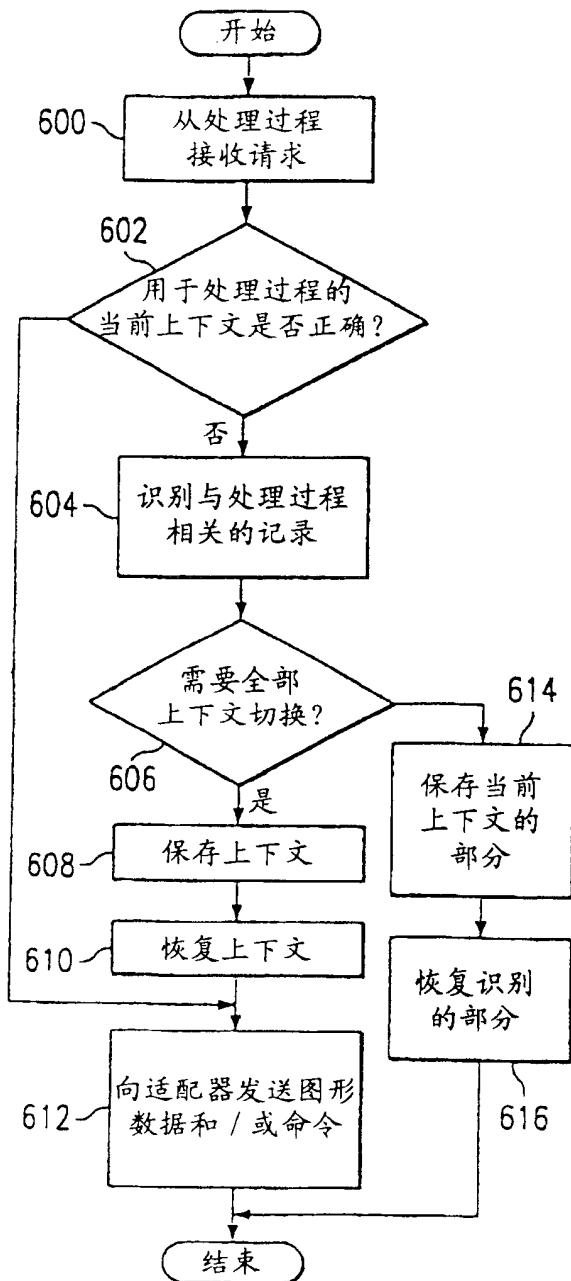


图 6

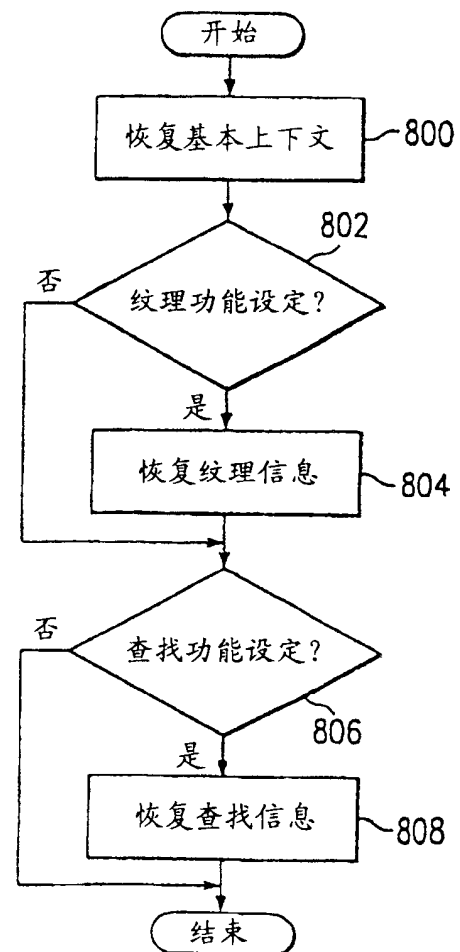


图 8