[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
G06F 9/44

[12] 发明专利说明书

[21] ZL 专利号 94190867.4

[45] 授权公告日 2003 年 10 月 29 日

[11] 授权公告号 CN 1126031C

[22] 申请日 1994.1.5 [21] 申请号 94190867.4
[30] 优先权
[32] 1993. 7.27 [33] US [31] 08/097,507
[86] 国际申请 PCT/US94/00125 1994.1.5
[87] 国际公布 WO95/04320 英 1995.2.9
[85] 进入国家阶段日期 1995.7.4
[71] 专利权人 塔里根特公司
地址 美国加里福尼亚州
[72] 发明人 杰夫·A·齐亚斯
审查员 杨 蕊

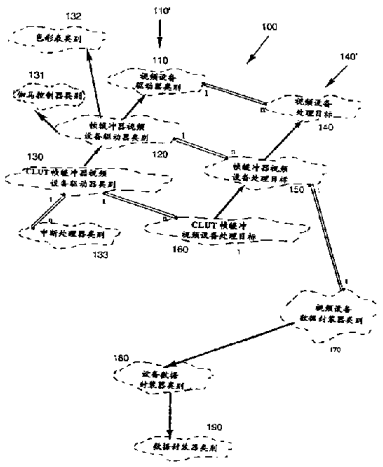
[74] 专利代理机构 北京康信知识产权代理有限责任公司
代理人 余 刚

权利要求书 4 页 说明书 22 页 附图 10 页

[54] 发明名称 面向目标的图形处理方法

[57] 摘要

用于数据处理系统的视频结构提供了并行层级的视频设备驱动器和视频设备处理，它们由视频设备驱动器目标生成，该视频结构还提供了由视频设备处理层级的子类别包括数据封装器层级，以便封装描述与数据处理系统相连的显示设备的功能性的配置信息。 应用程序通过视频设备处理之一在视频设备处理层的不同级访问视频设备驱动器。



ISSN1008-4274

1. 一种用于在具有面向目标的操作系统的计算机中处理图形信息的方法，其特征在于包括以下步骤：

(a) 在处理器的控制下由存储器中的配置存取管理器目标生成一个视频设备驱动器目标；

(b) 在处理器的控制下由存储器中的视频设备驱动器目标生成一个视频设备处理目标；

(c) 由存储器中的视频设备处理目标生成一个视频设备数据封装器目标；

(d) 在处理器的控制下将视频设备处理目标的参数送至存储器中的一个层服务器；以及

(e) 使所述视频设备处理目标与视频设备驱动器目标关联。

2. 根据权利要求 1 所述的方法，还包括将所述视频设备数据封装器目标的参数送至一个硬件配置服务器的步骤。

3. 根据权利要求 1 所述的方法，还包括向处理器标识视频设备处理目标的步骤，该目标从视频设备处理目标的类生成。

4. 根据权利要求 1 所述的方法，还包括把状态信息记录到存储器的步骤。

5. 根据权利要求 1 所述的方法，还包括向处理器报告状态改变的步骤。

6. 根据权利要求 1 所述的方法，还包括由处理器管理多个视频结构的步骤。

7. 根据权利要求 6 所述的方法，还包括激励视频结构的程序。

8. 根据权利要求 6 所述的方法，还包括对视频结构去激励的程序。

9. 一种用于在具有面向目标的操作系统的计算机的存储器中处理图形信息的方法，其特征在于包括以下步骤：

(a) 响应对处理器的至少一个配置变化进行的测试而生成一个视频设备数据封装器目标；

(b) 将配置变化的参数送至包含有视频设备数据封装器目标的视频设备处理目标；

(c) 响应视频设备处理目标生成的信号在至少一个显示设备上锁定绘制任务；

(d) 通过修改或生成一个视频设备处理而改变处理器的配置；以及

(e) 向视频设备处理发布命令以使视频设备处理重新生成对应于锁定的绘图任务的显示，并释放对绘图任务的锁定。

10. 一种使用视频结构生成视频设备驱动器层级的方法，该层级易于接近用户，该视频结构具有一个主视频设备类和一个主视频设备处理类，该方法包括以下步骤：

(a) 从主视频设备类衍生一个视频设备子类的层级，这样每个视频设备子类包括一个设备数据部分以及用于操纵预定类型的视频设备的设备控制程序；

(b) 从主视频设备处理类衍生一个的视频设备处理子类的层级，以包括与每个视频设备子类对应的视频设备处理子类，这样每个

视频设备处理子类包括一个数据部分以及用于与用户交互的程序；

其中，每个衍生的视频设备子类包括用于从一个相应的视频设备处理子类生成一个视频设备处理目标的目标程序，以及一个用于把至少一个设备控制程序拷贝到视频设备处理目标的目标程序。

11. 根据权利要求 10 所述的方法，其中所述视频结构还包括一个基本数据封装器类，所述方法还包括以下步骤：

(c) 衍生一个视频设备数据封装器子类的层级，这样每个包括一个数据部分以及用于存储和操纵视频数据的程序，

其中步骤(b)包括衍生至少一个视频设备处理子类，其包括用于从一个视频数据封装器子类中生成一个数据封装器目标的目标程序。

12. 根据权利要求 10 所述的方法，其中该视频结构在一个操作系统中运行，并且其中步骤(b)包括：

(b1) 衍生每个视频设备处理子类，以便包括用于把从每个视频设备处理子类生成的视频设备处理目标标识到操作系统的程序。

13. 根据权利要求 10 所述的方法，其中步骤(b)还包括：

(b2) 在视频设备处理子类的层级中衍生一个帧缓存视频设备处理子类，且其中步骤(a)包括以下步骤：

(a1)在视频设备子类的层级中衍生一个帧缓存视频设备驱动

器子类，这样帧缓存视频设备驱动器子类包括控制数据部分和用于操纵视频帧缓存的控制程序以及用于从帧缓存视频设备处理子类生成一个帧缓存视频设备处理目标的程序。

14. 根据权利要求 13 所述的方法，其中步骤(b)还包括：

(b3)在视频设备处理子类的层级中从帧缓存视频设备处理子类衍生一彩色查阅表帧缓存视频设备处理子类，

而且其中步骤(a)包括以下步骤：

(a2)在视频设备子类的层级中从帧缓存视频设备驱动器子类衍生一彩色查阅表帧缓存视频设备驱动器子类，该彩色查阅表帧缓存视频设备驱动器子类包括控制数据部分和操纵彩色查阅表的控制程序，以及从彩色查阅表帧缓存视频设备处理子类生成一个彩色查阅表帧缓存视频设备处理目标的程序。

15. 根据权利要求 14 所述的方法，其中还包括以下步骤：

(d)衍生一个中断处理器类，并且其中步骤(a2)还包括以下步骤：

(a2.1)衍生彩色查阅表帧缓存视频设备驱动器子类，以包括对至少一个从中断处理器类生成的中断处理器目标的引用。

16. 根据权利要求 10 所述的方法，其中步骤(b)还包括以下步骤：

(b4)衍生至少一个视频设备处理子类以包括用于生成一个提供一组最小图形函数的图形设备目标的程序。

面向目标的图形处理方法

本发明涉及具有输出设备的计算机系统，尤其涉及可连接到不同类型和数量的显示设备以获取兼容性并改进性能的一致性和用户方便性的计算机系统。

显示设备长期以来是计算机和各种交互数据处理系统以及与用户交互的数据处理系统的最佳输出设备，显示可快速改变并能向用户展示大量信息，这些信息可能与数据处理系统的操作和运行于其中的应用程序以及由用户输入和处理的数据操作有关。

为使用户易于理解这些信息，高分辨率的彩色显示器被广泛使用并由许多厂家生产。虽然工业界对此有一些标准，但性能的一致性或在数据处理器、运行于其中的应用程序和与其相连的输出设备之间的兼容性相对不同输出设备而言是不够的。例如，为获取打印机的兼容性和性能一致性，软件开发商和供应商要为不同的打印机提供驱动程序以便控制由应用程序处理的文件的内容。但是，可以理解在打印机之间获取一致性的程度比显示设备而言要直接的多，其原因在于许多打印机是单色的而且包括其自己的常驻标准字型。进而，打印机几乎没有在应用程序和分辨率或可被显示的单独图象点的图案之间的交互。反之，在显示中，应用程序通常必

须控制图象生成以便适当准确地与用户进行信息通讯。还有，很少有两个以上的计算机被用来生成一个文件且期望其有绝对一致的性能。

与之相比，在目前使用的工作站和其他数据处理器系统中普遍同时使用多个显示器。极为普遍的另一情况是随意用一种显示设备取代另一显示设备以更新系统从而运行某些程序。对监视器进行修理的要求可能导致用户用另一监视器将之取代。显示还包括比打印机要多得多的变化，例如，显示屏的高宽比可以变化，例如在不同电视信号格式中的数据，或所谓“风景摄影(长的水平轴)”或“肖象(长的垂直轴)”显示，后者通常最好是显示文件图象，高宽比也可以在这些显示类型中的后一情况下进行微妙变化，其取决于显示尺寸。在水平和垂直方向上的分辨率可以改变，就象在每个坐标方向上可寻址的象素一样，两者都是显示和应用程序或甚至操作系统一个事件。

进而，彩色再现很难准确的一致，既使在同一厂家生产的同型号显示器之间也如此。同时，现在的趋势是用大的色域，其中，应用程序可使用更多的色彩。因此，色彩，色调、强度和对比度的准确再现越来越重要，尤其是在用户与应用程序的交互过程中。

就象在打印机中一样，通常至少部分地以软件形式来作为显示驱动器以便易于控制和调整，硬件显示驱动器不大用于简单设备中。这样便遇到三种问题。

第一种即最关键的问题来自于先有技术采用显示来展

示错误消息和系统操作指令。在显示驱动卡与特定软件一起使用时，可以在该卡的软件被安装前安装该卡并进行系统引导。在引导过程中。处理器或总线管理程序将寻找视频卡并寻找相应的软件文档以生成显示驱动程序和视频卡目标。若这些文档(即库文档)找不到，则系统无法启动显示器以便既使提供所要求文档的快速安装，从而运行该显示卡。由于打印机无需被表示成为输出设备，所以所有可做之事便是导致生成某些声音，这些声音对简单对话指示具有足够的语法内容，对这一情况的唯一合理变化即为提供底级交替显示或该卡的防真，例如在 VGA 卡上所做的（其增加成本），提供包括视频驱动程序(也存在成本问题)的可引导硬盘，或通过文件迫使用户在安装板子之前就要装该软件。但是，只要当这一问题出现，即可对用户的自信心产生不利影响。

第二类问题相对系统操作而言不是很关键，但对由显示表示的用户界面却极为重要，尤其是在同时运行两个以上的程序或者多个显示设备与系统相连之时。在任意给定时刻只运行一个程序的情况下，显示表示在确保适当色彩和形成足够色彩或灰度级差以便于用户识别和区别方面是重要的。准确的色彩对于图形应用程序而言更为重要，但只要用户可以识别被处理的图形的色彩的情况下，有些色彩变化要是可以容忍的。进而，在这种情况下，操作员可以指定显示的假色以便区别分类似的色调，类似地，在寻址的象素数和图象的高宽比方面的某些变化要可容忍。

但是，当在不同视窗中由多个应用程序生成多个显示

时，更为重要的是一致均匀地表示彩色并在应用程序之间以应用程序与显示驱动程序交至的方式进行配色。因此，若不同应用程序提供其自己的界面给显示，则可提供某些机制以使它们协调。当多个显示被提供并生成用于可在显示之间任意移动或甚至在监视器之间划分(即在分别可寻址的多个设备上形成的相邻显示区域，例如通常在液晶显示或光纤屏CRT 中提供的)的视窗的规则时，问题会趋于复杂化，在这种情况下，每个应用程序都必须有能力访问每个显示的设备驱动程序并在其之间同步显示生成。

第三个问题可看作为第二个问题的特殊情况。如上所述，通常是在引导操作期间对系统进行配置并在同时识别所有可用的显示。但是在提供多个显示的情况下，系统的配置可改变，例如当一特定监视器被接通或断开的情况，或一个监视器由另一处理器或用户程序占用的情况。后一种情况可能发生于由一个以上用户使用具有多个显示设备的工作站而且某些显示设备可以共用或唯一地分配给一个特定用户的情况下。在任何这类情况下，必须制订不仅用于在操作过程中的系统重新配置而且可用于所有用户显示保存(例如在可用于这些显示的多个显示设备可能突然变化时分配给一个用户的视窗)的规则。

虽然以上问题的其中一些可由特殊的例行程序或在改变发生时调用的目标所处理，但没有可以同时解决这些问题的方法。

因此，本发明的一个目的是提供一种视频结构体系（Video framework architecture），它用于安装，配置和利用提供绘图，动画，和视频配置服务的视频设备。

本发明的另一目的是提供一种通用结构，它用于应用程序和显示驱动程序相接口以便获取正确和一致的显示表达控制，该显示表达与机器和显示设备无关。

本发明的另一目的是提供视频结构（Video framework），尤其是用于应用程序开发的视频结构，它以一致和系统化方式提供显示特性的易调节性和灵活性。

本发明的再一目的是提供一种视频结构，当为满足用户要求而增加系统灵活性时，对显示的要求的变化也随之存在，上述视频结构将能满足数据处理系统的动态再配置。

为达到本发明的目的，所提供的视频结构包括具有至少一个子类别的视频设备驱动程序类别，一个由视频设备驱动程序的类别的目标包括的视频设备处理类别，该视频设备处理类别具有由其衍生的子类别，它与视频设备驱动程序和从驱动设备驱动程序的类别衍生的每个子类别的各个目标包含的每个子类别相对应；该视频结构还包括一个数据封装程序的类别，该封装程序具有至少一个由其衍生并包括在视频设备处理的类别衍生出来的子类别的目标中的视频设备数据封装程序子类别。

根据本发明的另一方面，一种视频结构方法包括根据一个配置访问管理程序目标生成一个视频设备驱动程序目标的步骤；根据视频设备驱动程序目标生成视频设备处理目标

的步骤；根据视频设备处理目标生成视频设备数据封装程序目标的步骤；把视频设备处理目标送到层级服务器；把视频设备数据封装程序目标的参数送到硬件配置服务器的步骤。

根据本发明的另一方面， 用于操作一个至少连有一个显示设备的数据处理系统的方法包括以下步骤：响应至少一个配置变化的检测和数据处理系统配置变化的请求生成一个视频设备数据封装程序目标，把配置变化的一个参数送到包含视频设备数据封装程序目标的一个视频设备处理目标，响应由视频设备处理目标生成的信号在一个显示设备上锁定绘图任务，通过修改或生成一个视频设备处理来改变数据处理设备的配置，向视频设备处理发出一个命令以重新生成或请求重新生成与锁定的绘图任务对应的显示，并且释放锁定的绘图任务。

图 1A 是一个计算机系统的硬件图，本发明可以在该系统上运行；

图 1B 是一个 Booch 图，用于说明根据本发明的目标的类别构成；

图 2 是一个 Booch 图，示出了图 1B 中各个类别中的目标的合作和生成；

图 3 和图 4 是采用 NuBus 的数据处理器来实现本发明的示图；

图 5 是将本发明用于减少指令集(RISC)计算机的示图；

图 6 是将本发明用于小型计算机接口系统（SCSI）视频控制器的示图；

图 7 是根据本发明由若干目标的类别生成图形设备的

示图；

图 8 是根据本发明的视频结构的从属关系；

图 9 是根据本发明的视窗服务器的最佳行为。

参见附图，尤其图 1B，其示出了根据本发明的最佳实施例的基本构成的 Booch 图。正如在这类图中所规定的，“云彩”由虚线形成并用于指示目标的类别或子类别（按面向目标的编程术语）并用箭头展示其中的层级关系；每个箭头都指向该级的下一个更高层级，并且表示一个“是(is a)”的关系（可以理解虚线通常用于从单个目标中区分目标的类别并且没有含义说如此描述的类别是常规的。例如，实线框可由 Booch 图的云彩所取代以形成实质上具有同样内容的图形）。与云彩相连的双线表示在类别和子类别之间“有(has a)”或包含关系。与双线端是相邻的数字指示多个类别之间的对应性或它们的对应性的“重要性”。这一关系亦成为“使用”关系而且在双线端部的点指示使用发生的方式,双线端处的实点指出该类别是所指示的另一类别的内部实施，且无需用于该原因的界面。反之，一个开端点指出最好提供一个界面。Booch 图的其它原理级在其后遇到的符号处予以解释。但是，可参见“面向目标设计和应用（Object Oriented Design-With Applications）”一书，该书作者为 Grady Booch, 由 Benjamin/Cummings Publishing Company, Inc.在 1991 年出版。

最好要理解本发明的实施例是由根据从所谓面向目标编程发展而成的目标技术所实施的，面向目标的编程已变得

越来越重要，其原因在于数字处理技术已为许多不同任务的并行处理提供了更多的支持。在目标技术中，由编程的通用计算机仿真但可被看作或实际作为特定目的的数据处理器的目标被提供以执行所需的各种功能。所需要由每个目标执行的少量方法或过程的数据与目标本身密切相关，方法被封装或相对其他目标而言被隐藏起来，这些其他目标可调用由该目标执行的一个特定方法。

进而，可根据类似特性而将目标归类或分成子类别，而且可由一个子类别继承一个类别(数据或方法)的特性而无需再做其它指定。此外，继承的特性可由在一子类别中的目标所置换，从而导致一个称为多形性(有时亦称为运行时间配连，因为置换是由正在执行的方法调用的)的属性。

因此，在本发明最佳实施例的实施过程中，目标和其类别是整个系统最基本的功能元件。在这些元件之间的功能关系由责任(即包括但不限于生成其他目标的方法或操作)定义和与目标和目标的类别相关的层级相关性所定义。定义目标的层级和继承的行为称为“子分类”。因此，本发明的最佳实施例及根据一般技术人员所理解的类别和子类别的组成的责任关系和从属关系进行叙述。

实际上，任何数据处理系统将包括至少一个显示设备或一个显示设备驱动。当系统被引导时，内部码将访问在(或联接到)系统中的各种设备，包括显示设备或驱动器，并在每个设备和中央处理器之间通过内部系统总线提供通信。广义而言，根据本发明一个最佳实施例的视频结构在此时被启动而

且由至少一个配置访问管理程序响应总线的连续或周期遍历(或移动)以例示或删除显示设备和驱动器,并修改显示表达的控制以反映系统的当前状态。

重要的是对本发明最佳实施例的性质进行了解以理解"结构(framework)"的概念以及对于"目标"的结构和"面向目标的编程"的关系,由Kurt A·Schmucker所著且由Byte杂志于1986年8月期刊登的题为"MACAPP:An Application Framework"一文是较早叙述结构和目标概念的文章,该文被引用于此以为参照。目标的一个重要属性在于它们封装其导致的数据和方法的能力。即一个类属命令可以发布给一个目标而不须任何其他目标了解该目标如何执行该命令的细节。由同样道理,无需要命令,数据,文件名等有全局兼容性而且目标可以随意与其他目标相关联,目标亦可作为类别与层级中的子类别关联并继承从该层级衍生的目标的属性。在目标执行任务时可置换继承的属性,从而提供称之为"多形性(Polymorphism)"的特性。

实质上,结构是一个一般应用程序,它包括在必要时把目标的类别与其他可以关联的目标关联起来,以形成更为特殊的应用。作为与定义的目标的类别之间的功能间的相互关系相关联的目标类别的结构而提供了任意需要的通用和特定功能级,该结构要亦将提供可与该结构关联的其他目标的正确功能性。

一个结构因此可被称为是一个系统,它提供目标之间隐含的响应性网络。该系统还提供目标的类别之间的继承(目

标类别的更高层级的主类别的数据和方法）以及响应于事件对库的调用。构成结构的系统亦可由执行更具体功能且可置换由结构提供的功能的其他目标所定制。在结构的不同类别和子类别中针对机器和针对设备的目标允许结构本身是机器和设备无关的并具备通用的应用性。进而，一个特定结构的特征在于根据所获取的功能性，继承性和责任性划分而在目标和目标的类别之间建立的内部关系。一个结构的本身也用作样板以供开发特定的应用程序，其中，定制和功能置换作为特定目标而被提供。

具体参见图 1B，本发明最佳实施例的基本配置 100 提供了设备/驱动器的类别和子类别层级 110`，根据衍生顺序，最好包括视频设备类别/视频设备驱动器类别（它们被并行分类）110，帧缓冲器视频设备驱动器类别 120 和色彩查寻表（CLUT）帧缓冲器视频设备驱动器类别 130。还提供了处理类别的并行层级 140`，它最好包括（以衍生顺序）视频设备处理目标 140 的类别，帧缓冲器视频设备处理目标 150 的子类别以及 CLUT 帧缓冲视频设备处理目标 160 的子类别。在处理类别层级中的每个类别包括了多个目标类型类别，它们作为驱动器类别层级 110`的各个类别和子类别产生的相应驱动器目标的替代。处理类别层级的多个级允许客户（应用程序）以多个不同的继承级访问驱动器类别。对每个驱动器类别和子类别规定的多个处理类别和子类别提供了多个针对客户的目标，这些目标被提供用于对驱动器层级的任何级进行访问。

图 1B 所示类别表达了抽象的不同级别。根据本发明的

最佳实施例的关键抽象是视频设备处理类别，视频设备驱动器类别以及视频设备数据封装程序类别（该程序对多个层级进行进一步分类）。

为客户访问之缘故，视频设备处理类别 140 表示（或被替代）视频设备或用于操纵视频设备硬件的视频设备驱动器的部分。视频设备处理类别的特定设备相关子类别用于配置硬件设备并根据当前的配置报告视频定时，中断状态和视频信号的其他方面。该抽象因此明白一个硬件的视频设备的定时特性并可根据经验测量一个小精灵的位级传送矩形的绘制时间以回答随后的询问，从而调度绘图，正如其下以小精灵为例进行的讨论所述。

由视频设备驱动器类别 110 产生的视频设备驱动器目标包括直接操纵视频设备硬件并由相应的处理类别和子类别表达的目标。如上所述，视频设备目标与其相应的视频设备驱动器子类型并行地分类，由这些类别产生的视频设备目标不包括特定的硬件实施或由视频驱动器目标包括的接口。但是，特定的硬件实施或接口可以在系统中以硬件或软件提供，而且单独的标准视频显示设备功能的子类别不是必要的。如果不需要视频显示设备的非标准行为的话。由视频设备驱动器类别 110 和帧缓冲设备驱动器类别 120，以及 CLUT 帧缓冲视频设备驱动器类别 130 产生的目标中包含的目标最好在根据本发明最佳实施例的视频结构中提供所有标准的视频显示设备行为。

帧缓冲视频设备类别和 CLUT 帧缓冲视频设备类别是视频结构中的抽象的更为具体的子类别级别。CLUT 帧缓冲

视频设备驱动器的进一步抽象是伽马控制器类别 131 和色彩表类别 132（在本发明的上下文中，“伽马”指为了从任意图象参数的特定设备获得或多或少的线性可视响应的偏置的表，可能包括但不限于对比度，其取决于设备的色空间（即 RGB，CYMK 等）的视觉响应坐标系统或由显示再现的色域）。这使得色彩表和对比控制可用于整个结构。CLUT 结构缓冲类别也有（或包括）至少一个中断处理器类别 133 作为一个内部实施以便在必要或需要时加入特殊的非标准视频设备处理行为。

视频设备数据封装器类别 170（其中之一对应于帧缓冲视频设备处理类别 150 的每个类别），设备数据封装器类别 180 和数据封装器类别 190 对实施本发明的最佳实施例并不重要。由于目标本身就提供封装，因此只需提供目标的一个类别以便作为帧视频设备处理类别 150 中的内部实施以达到该目标。如此，任何表示一个显示设备的可能或潜在功能性的目标，最好是以配置的形式通过视频设备处理提供该功能的表示。

如此，通过一个应用程序或另一目标，操作员可调用一个特定显示设备的再配置（即例如由一个鼠标器选择一个显示的数据封装器图符）而且数据可以基于硬件的选择屏或图象（即由按键控制位深度或映射到 CLUT 的色彩）的作为控制对一特定显示设备的特定信号的响应的图形用户界面而返回。同理，由任何应用程序进行的显示图象的任意控制可作为导致在显示控制下的显示设备的潜在功能性的表达返

回而被感测。替换之，可提供目标来报告变化时在数据封装层级中的数据的任何变化，相对于每个在处理层级的适当级别中的显示设备的目标中变化的每个显示设备的潜在功能性，以及响应于其所解决的冲突。因此，一个显示设备的功能性中的变化（它可由一用户或系统中的其他行为实现，例如断开显示设备或由其他处理由应用程序或用户占用显示设备）可被立即感测而且适当的显示控制改变可响应于其而予实现，如下所详述。

参见图 2，它是图 1B 所示多个类别中的特定目标的 Booch 图(由实线“云彩”来表示目标)。当系统引导时，与系统连接的设备由配置访问管理程序 210 所发现，假设发现一个显示设备，则生成一个视频设备驱动器目标 220。为此目的，在通用形式中的至少一个系统预置显示驱动器类别（即通用于多个显示设备类型之一）最好包括在可由视频结构访问的系统库中。依次，视频设备驱动器目标 220 通过复制在视频设备处理目标的区域（由包括指示区域的字母“F”的方块指示的）中的方法调用来生成视频设备处理目标 230，而且其后可被客户程序作为设备处理目标 230 的内部实施 （由全黑填充块指示）而被访问和使用。一旦已经生成了视频设备处理目标 230，它便将其本身标识到在客户应用程序控制下对实际显示图象的组合进行控制的层服务器目标 240。视频设备处理目标也生成视频设备数据封装器目标 250，其方法是根据配置访问管理器 210 的初始报告并通过图 2 所示的其他目标向前进行而形成包含有与显示设备的功能性有关的信

息的一个文档，该功能性亦被报告给一个硬件配置服务器目标 260 以确认在设备的功能性和其当前配置之间的对应性。

再来看图 1B，一旦图 2 的目标已被生成，一个用户应用程序即可访问视频设备处理目标 230 并要求显示设备或系统的配置的变化。这一调用将修改存储于视频设备数据封装器目标 250 的文档中的功能性数据，封装器 250 依次把变化送到硬件配置服务器目标 260 以进行配置改变。反之，根据一个用户动作（例如关断一个显示设备）而产生的显示设备状态的变化由系统检测而且该视频设备处理目标 230 被因此更新。该状态变化然后通过视频设备处理目标 230 报告给层级服务器目标 240，它将为视频处理类别和/或子类别的其他成员提供一个系统预置或其他行为，以便在其他显示设备上重新绘制显示并在其间分配显示信息。一旦其被完成，把一个通知返回到特定的视频设备处理目标 230 以“释放“视频设备处理目标 230 以及相应的视频设备数据封装器目标 250 以及视频设备驱动器，或依次修改它们以反映系统配置中的被迫改变（即当显示设备由另一应用程序或处理器使用时）。

如此，总而言之，任何客户或应用程序可通过从目标的处理层级的级别之一中对目标进行选择而调用任意特定显示的控制。根据本发明的最佳实施例的视频结构所提供的系统是完全通用的。因为任何必要的特定设备目标都可以放在驱动器层级的适当级别中，而且由该层级中的继承确保了执行的效率。类似地，用于取得理想显示设备行为的任意特定系统或特定应用程序目标可放在处理层级的适当级别中。通过在处理类别或子类别中的目标对显示图象进行控制亦成

为一个事件，它可用来触发系统再配置和相应的显示控制，从而在以理想方式生成显示的设备之间分配显示。以这种方式，显示功能性的系统预置级总可把显示设备作为视频结构的一个部件而进行维护，而无需单独提供每个卡或显示设备。同样的结构体系也同时提供自动系统再配置并支持显示再定向以便在任何给定时间与存在的系统配置相对应。

根据以上对本发明最佳实施例的总体结构和基本体系的综述，以下将讨论几个实例。无论是在通用情况下还是在特定实施中，例如利用目标阵列（称为 NuBus，由苹果计算机公司实施的 32 位总线，其与 IBM 实施的 MicroChannel 和 EISA 总线相对照），都有一些对系统和设备配置信息的访问控制。因此，在一般情况下，足以实施本发明的最佳实施的以便响应于在应用程序或对系统和设备配置的访问进行控制的目标中的特定事件而在上述图 2 中讨论的目标之间提供连接（即责任和从属）生成。

参见图 3，在采用 Nubus 的系统上，最好提供配置访问管理程序类别 320 或子类别 330，目标的 NuBus 类别 310 是定义 NuBus 功能性的主类别，它包括 NuBus 结构，其内部细节由于被封装而对理解本发明的实施的并不重要，本领域的一般技术人员足以实施它。为理解本发明最佳实施例的视频结构，目标的 NuBus 类别至少包括查找与 NuBus 连接的卡并标识该卡的协议。实质上，目标的 NuBus 类别通过调用配置访问管理程序目标（即图 2 的 210）而提供周期性地或连续地行走总线（Walking the bus）,以监测配置的变化。从

功能上说，这些目标的责任是分别提供对于配置访问管理程序目标 210 或目标 320 的类别（如图 2 和图 3 所示）的访问地址。NuBus 类别的例子还包括用于由特定 NuBus 卡目标 340 产生目标（例如 NuBus 视频卡类别 350）的方法，这可通过把一个板识别码送给 NuBus 卡的目标的方式而实现。

此外，如图 3 所示，目标 360 的一个 NuBus 配置只读存储器（ROM）类别具有一个与 NuBus 卡目标 340 的类别一对一的基数，该存储器类别包括与内部操作有关的信息以及可与系统相连以控制生成更多特定目标（例如 NuBus 视频卡类别 350）的任何硬件卡的功能性。该 NuBus 配置 ROM 目标因此表示了 NuBus 卡 ROM，它包括强制信息区，例如格式标题块以及插接板资源表。NuBus 配置 ROM 类别也可再分类以提供用于卡的唯一特性或功能性的选择程序，正如具有与 NuBus 视频卡类别 350 一对一的基数的视频 NuBus 配置 ROM 类别 370 所示的那样。

参见图 4（它可看作是图 3 右部的连接），NuBus 实施被看作是包括驱动器类别和子类别的层级，这些类别和子类别包括视频设备驱动器类别 330（对应图 1B 的类别 110），NuBus 视频设备驱动器类别 120`和 NuBusCLUT 视频设备驱动器子类别 130。最好能获得由视频设备处理类别表达的显示设备功能性信息，以便用于设备配置的图形可视显示以协助用户进行设备再配置。为此原因，视频设备驱动器类别 330 以一对一的基数直接包括（作为内部实施）视频设备数据封装器类别 410，和视频设备处理类别 140（由延伸通过视

频设备处理类别 140 的点划线表示）。

如此数据封装器的一个类别的一种选择允许视频设备处理目标得以表示一个设备的可能的功能性的适当表达以及用户控制选择，例如使用的 CLUT 对 32 位彩色的使用，伽马（即可见响应非线形改正或补偿）表选择，以及视频设备处理目标的类似用户可配置方面（据此，可由一个应用程序访问一个显示设备）。在该实施中，NuBus 视频驱动器类别 120 还包括中断处理器类别 133`。

用于减少的指令集计算机（RISC）的彩色液晶显示（LCD）将作为本发明最佳实施例的另一例子而参见图 5 予以叙述。在该实施中，一个关键抽象是一个与图 1B 所示的同样类别对应的视频设备处理 140 的类别。假设 LCD 显示，则“处理”层级的其他将包括一个彩色 LCD 视频设备处理类别 520 以及一个 RISC 彩色 LCD 视频设备处理类别 530（它顺序地被分类于视频设备处理类别 140 之下）。另一关键抽象是由 RISC 彩色 LCD 视频设备处理类别 530 所包括的 RISC 主机板目标 510 的类别。在这一情况中，RISC 主机板类别提供协议以查找可访问的显示设备以及控制配置访问管理程序以便“在总线上行走”，如先前所述。当一个显示设备被发现时，示例一个 RISC 彩色 LCD 视频设备处理目标。这一目标然后可以进一步利用 RISC 主机板目标的功能性来确认其本身各方面的位置，例如它使用的存储器映射位置和其他可能是针对主机板的细节，如此，可保持本发明最佳实施例实施的通用性。

参见图 6，现在讨论包括一个 SCSI （小计算机系统接口）视频控制器的本发明最佳实施例的实现。在该实施例中，驱动器层级包括一个视频设备驱动器类别 110 和位于其下被进一步分类的帧缓冲视频设备驱动器类别 120，如图 1B 所示，以及一个在帧缓冲视频设备驱动器类别 120 之下被再分类的 SCSI 帧缓冲设备驱动器类别 130。由 SCSI 帧缓冲视频设备驱动器类别 130 产生的一个目标中的目标包括用于生成 SCSI 视频设备目标（它然后被平化至送至层级服务器目标（图 2 的 240）的信息和过程该 SCSI 帧缓冲视频设备驱动器目标（如图 1B 的 133）最好也包括 SCSI 能力作为一个区或所谓“混合（mixin）”以使用 SCSI 总线协议以询问 SCSI 设备硬件并接收信息以正确地标识它。在面向目标的技术中，混合是一个类别，它指采用多个继承来强化一个衍生类别的属性和行为从而实施衍生类别从其基本主类别继承的特性的类别，混合是一个多继承性的并加入其他功能性的类别，而 mixin 则是一个标准的 OOP 术语。SCSI 帧缓冲视频设备驱动器目标最好是 SCSI 意识以便有能力分解 SCSI 操作并因此最好包括 SCSI 访问管理程序目标和用于视频设备本身的 SCSI 中断处理器目标(如图 1B 所示由类别 133 生成)。

应该了解，上述包括 SCSI 视频控制器的示例说明按照本发明最佳实施例的视频结构是如何支持帧缓冲和伽马控制行为的而且本发明的最佳实施例的视频结构是如何能够使用和控制传送与系统的其余部分无关的协议的。视频结构功能性与系统总线和界面体系是分开的。从以上示例可以了解本发明的任意或所有变形可通过把特定目标加至图 1B 所

示的视频结构体系的不同点的方式而予实现。例如，图 5 的实施可通过把彩色 LCD 视频设备处理和 RISC 彩色 LCD 视频设备处理（包括 RISC 主机板类别 510）类别 520 和 530 从视频设备处理类别 140 中进行分类而实现。该分类与 NuBus 实施和 SCSI 视频控制器实施无关而且因此可结合所有这些实施或其中之一而予实现。但是，可预见大部分加到本发明的视频结构的目标将包括目标的子类别（即提供其他不同的目标），这些子类别较之视频设备驱动器和视频设备处理类别 110 和 140 而言处于较底的层级，并可能包括最好在视频结构的层级中提供的目标的置换以便提供特定的设备功能性。从视频结构的层级中，实现了子分类。

可以根据本发明的最佳实施例继续加入具有或没有进一步分类目标的目标直到理想情况达到。例如，在引导过程和误差处理过程中，最好提供在某些显示设备的屏幕上显示某些信息的能力，即使缺乏大部分包括视频结构的目标，它们按需要以图 2 所述方式生成。当然，一个增强且独立的图形系统只好牺牲其灵活性，尤其是在更新和加入硬件显示设备、加速器、协同处理器等的时候，其原因在于这种增加会需要所有可能需要的目标继续可用（例如在 ROM 中），就现实而言，最好把某些目标作为图形类别提供，其任何应用程序或处理器本身（即通过一个引导导体目标）可以是一个客户，它有最小但仍然完全叙述的从属类别或服务器集。

因此，如图 7 所示，任意或所有多个视频设备处理目标(即 710)有一处理器，用于生成一个图形设备目标 720 作为

文档，它总体包括在共同存储器中的图形设备系统。每个图形设备目标720最好是图形屏幕图象的形式，它支持一个最小的图形功能集，例如显示文本，在屏幕上绘制四方形并复制图象。它最好还支持彩色或灰度级的某些多个象素体深度位，即使某些客户（例如引导导体目标730和应急错误处理器目标740）上可要求单色显示。这些图形设备目标720的生成意味着至少用于主处理器的配置访问管理程序从视频结构本身而言可用。

一旦生成之后，图形设备系统是一个小的无状态自满足系统，它无需系统中的其他服务，而且是可用的而无论是否丢失（即损坏或仍未开始）了象令牌服务器，层级服务器，号志服务器之类的目标。在生成图形设备目标720之后的设备配置或系统的每一变化都由视频设备处理目标710延迟到适当的图形设备，以致图形设备目标将具备屏幕状态的当前指示，例如象素位类型以便使用适当的绘图方法。

应该了解到在视频系统体系其他部分中的特定目标通常都用于根据多种可以运行的应用程序的屏幕显示。图形设备目标 720 的类别最好可通过把对视频结构的其他部分的依赖性减至最小的有效方式来包括这些目标的子集。以这种方式，限制了与用户进行交互时的干扰而且用户可以在需要的时候由与恰当的应用程序类似的图形设备立即采取补救行为。

图 1B 的视频结构 100 的主要系统从属性示于图 8 中。尤其是，视频结构 100 必须有效地与层级服务器交互以提供一个完整特性的绘图环境。该图形系统必须由适当的显示同步

820 支持随意分割的图形项的绘制。系统图形设备 830 包括图形设备目标 720，并在大部分情况下由视频结构的帧缓冲子类型目标的目标所支持。这些系统从属性将依次讨论以解释本发明的最佳实施例。

图 9 所示情况指出了层级服务器的最佳行为。图 9 的目标（即视频设备驱动器目标 920）可以但不必是上述的相同目标（即视频设备驱动器目标 220）而且新的标号因此被给出。如图 2 所简述的，一个引导过程包括一个系统起始顺序，它包括沿系统总线寻找系统各部分（包括显示设备）的配置访问管理程序目标 910。当发现一个视频显示设备时，起动或示例相应的视频设备驱动器目标 920。然后视频设备驱动器目标 920 生成一个视频设备数据封装目标 930 并把其相关的参数送到硬件配置服务器目标 940。视频设备驱动器目标 920 还生成一个视频设备处理目标（例如 960a 或 960b）并将其送至层级服务器目标 950。这些视频设备处理目标稍后将在配置变化时警告层级服务器目标。层级服务器目标依次警告配置变化时的视图系统目标 970，与之响应，视图系统目标锁定当前的绘图任务目标 980 直到层级服务器目标 950 可根据新的系统配置调用重新生成显示。一旦重新生成，新的显示通过视图系统的正常方式再次生成。当显示被任何方式切断或改变时，视频结构提供显示生成的自动改变。改变的显示生成将防止由于显示布局和配置中的变化导致的视窗信息或存储图象的丢失。

本发明的最佳实施例最好也支持动画。动画包括但不限于实况视频，它由生成一个特定小精灵的视频信息的同步位

级传送(BLT 或 Blit)所实现,BLT 和该生成与显示设备同步以避免产生的精灵的图象分裂。因此,该视频结构必须明白显示硬件的特性而该信息最好由响应此后回答询问的视频设备处理进行的精灵的 BLT 矩形的绘制时间经验测定所得到。

由于本发明最佳实施例的动画小精灵的绘制的良好特性,一个动画中断处理器最好被用于不使用显示定位计数器(如光束位置计数器)的硬件显示设备。若一特定显示设备使用该计数器,则可从该计算器直接获得同步。但是,在使用诸如垂直消隐(VBL)中断之类的中断的设备中,可根据本发明在视频结构中放入一个单独的动画中断处理器。在这种情况下,该动画中断处理器会接收来自视频设备的中断并标记该中断的实际时钟时间。该精灵目标然后可以根据最后一个标志时间计算在提供下一个动画帧的 BLT 后绘制该小精灵的下一时间。

根据上述,可清楚知道本发明的视频结构以设备无关和机器无关的方式提供了数据处理系统的动态识别。该方式在系统配置改变或不能访问用于设备相关的显示驱动器的操作的库文档时可将显示的干扰减小最小。

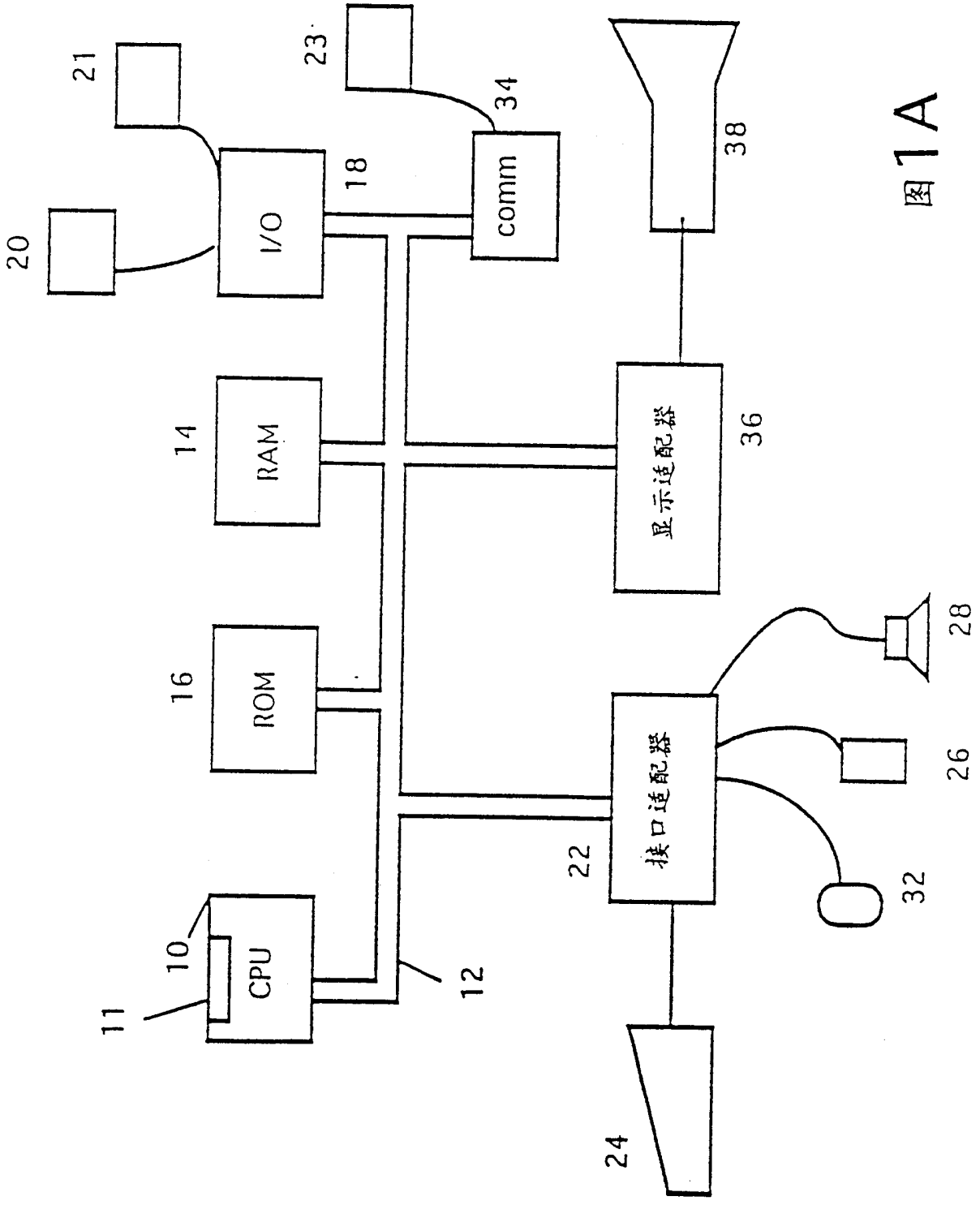

21
23
34
38
20
I/O
18
comm
14
RAM
显示适配器
36
16
ROM
28
22
接口适配器
26
32
10
CPU
11
12
24

图1A

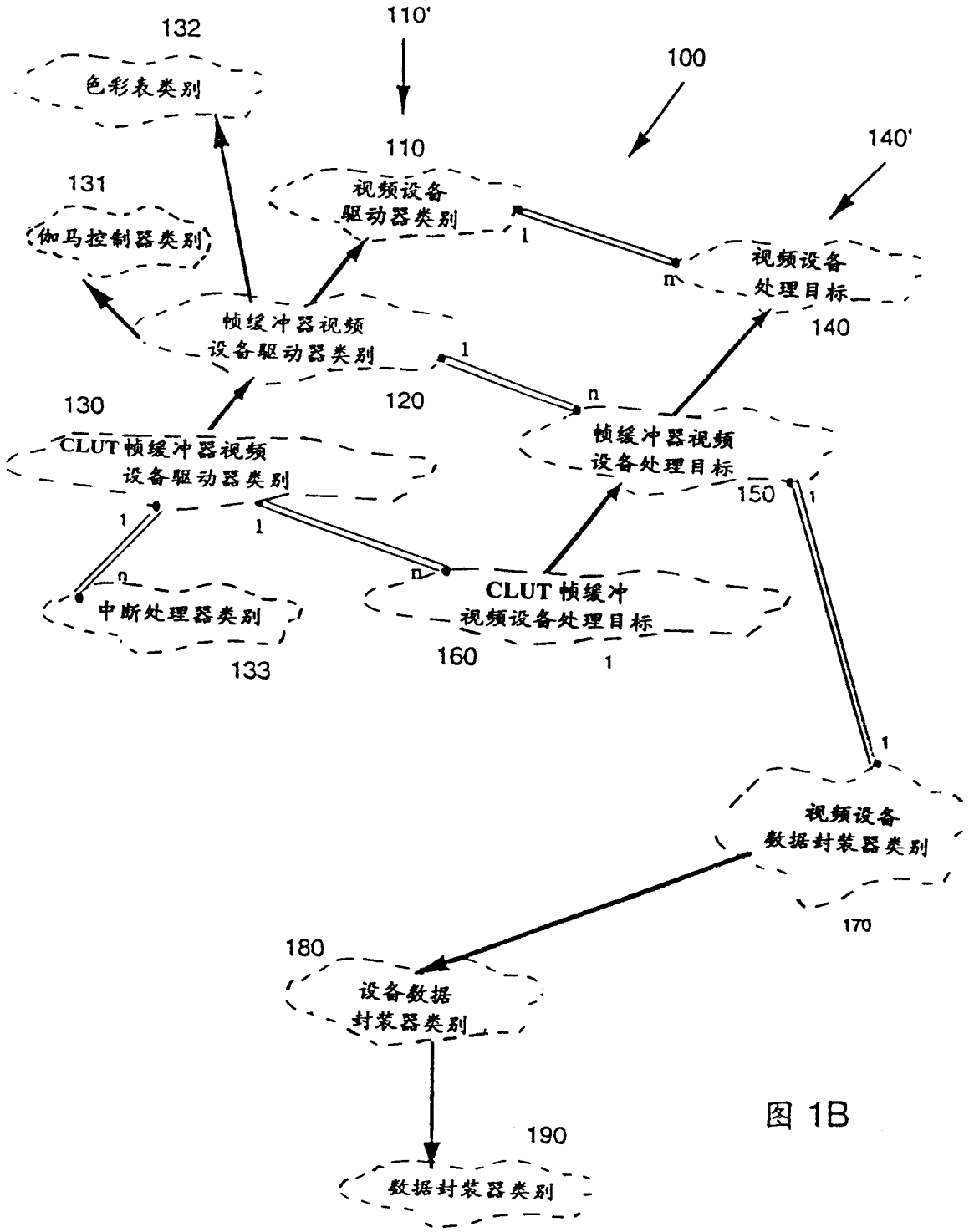

132
色彩表类别
110'
100
110
视频设备
驱动器类别
140'
131
伽马控制器类别
1
m
视频设备
处理目标
140
帧缓冲器视频
设备驱动器类别
1
n
120
130
CLUT帧缓冲器视频
设备驱动器类别
帧缓冲器视频
设备处理目标
150
1
1
1
n
n
中断处理器类别
CLUT 帧缓冲
视频设备处理目标
133
160
1
1
视频设备
数据封装器类别
170
180
设备数据
封装器类别
190
数据封装器类别

图 1B

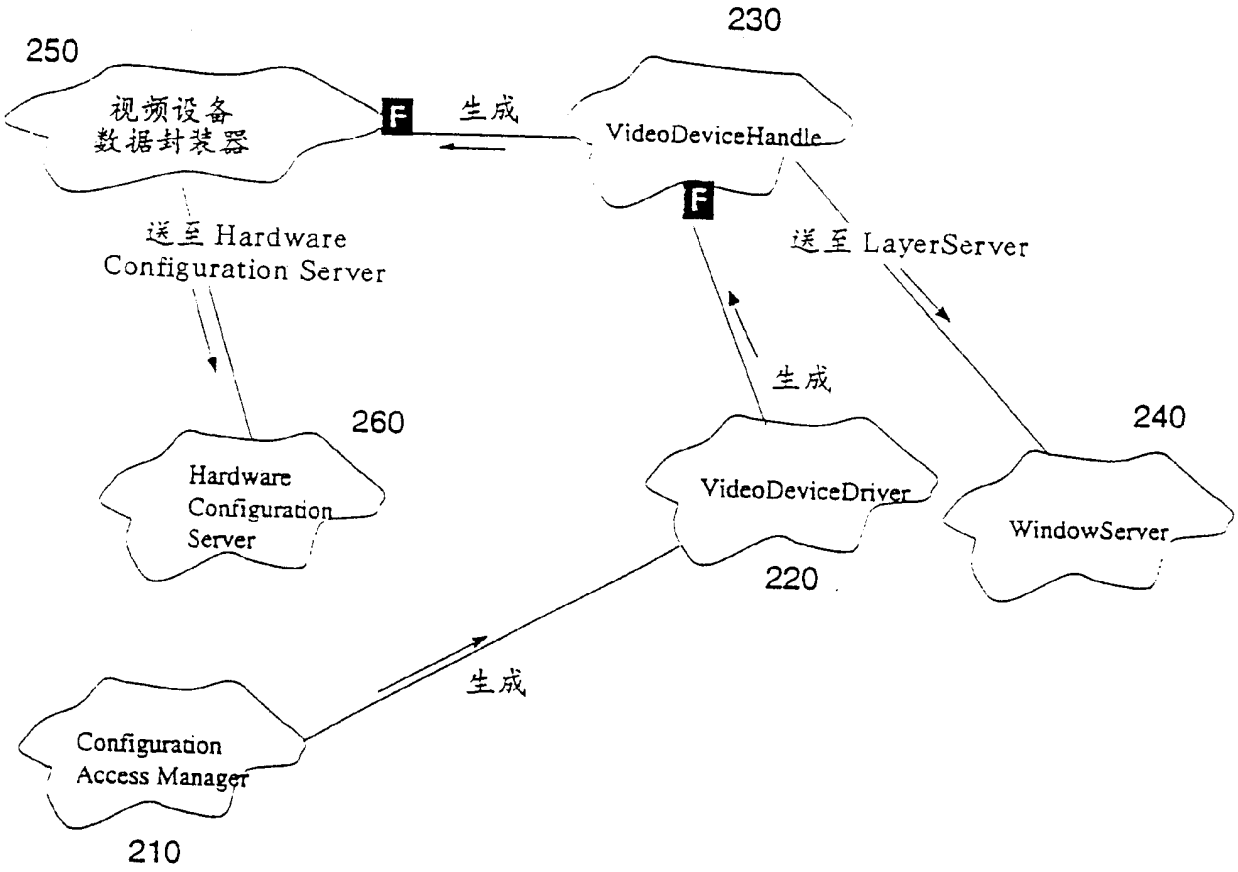

250
视频设备
数据封装器
F
生成
230
VideoDeviceHandle
F
送至 Hardware
Configuration Server
送至 LayerServer
生成
260
Hardware
Configuration
Server
VideoDeviceDriver
240
WindowServer
220
生成
Configuration
Access Manager
210

图 2

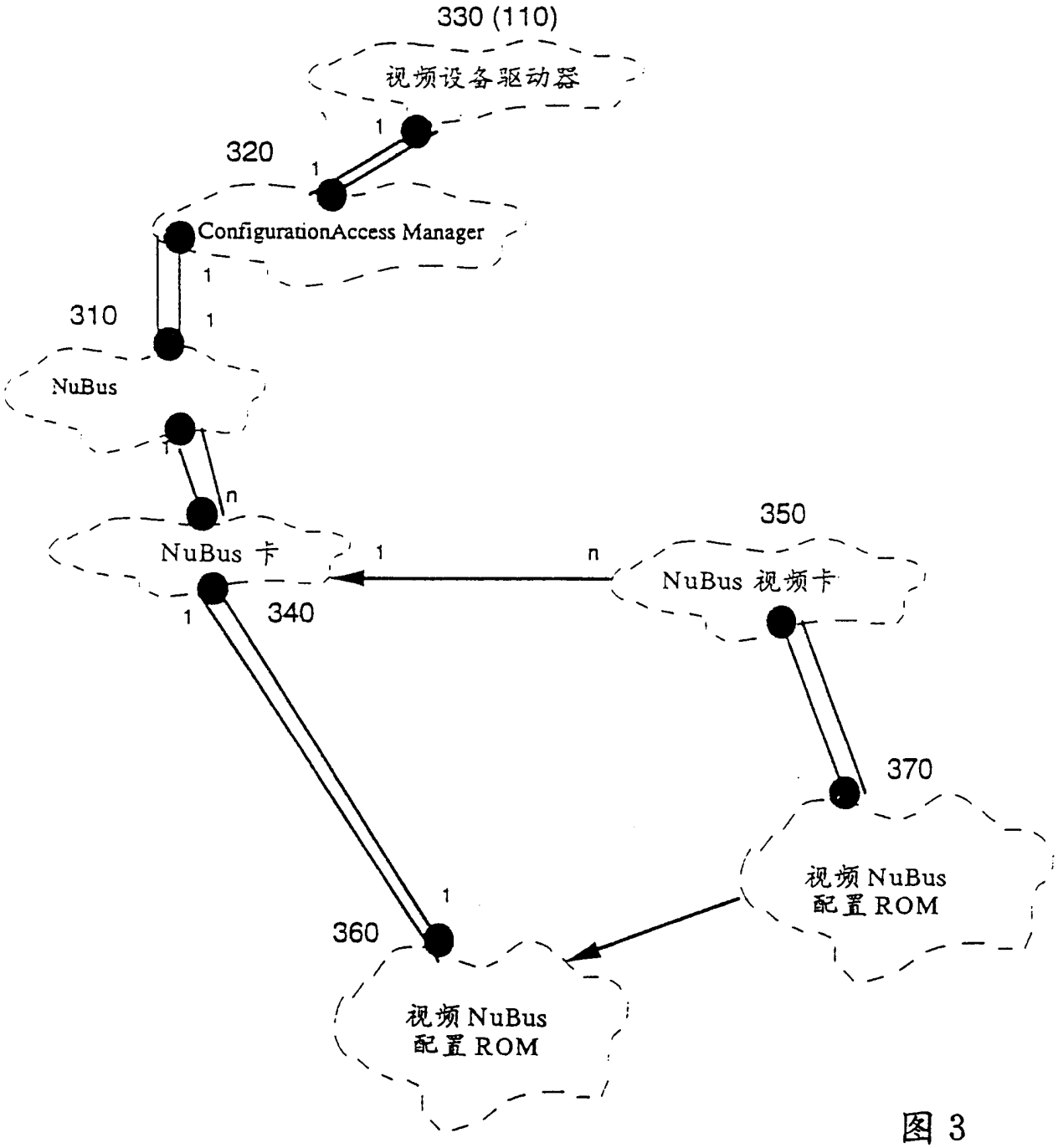

330 (110)
视频设备驱动器
1
320
1
ConfigurationAccess Manager
1
310
1
NuBus
1
n
NuBus 卡
1
n
350
NuBus 视频卡
1
340
370
视频 NuBus
配置 ROM
1
360
视频 NuBus
配置 ROM

图 3

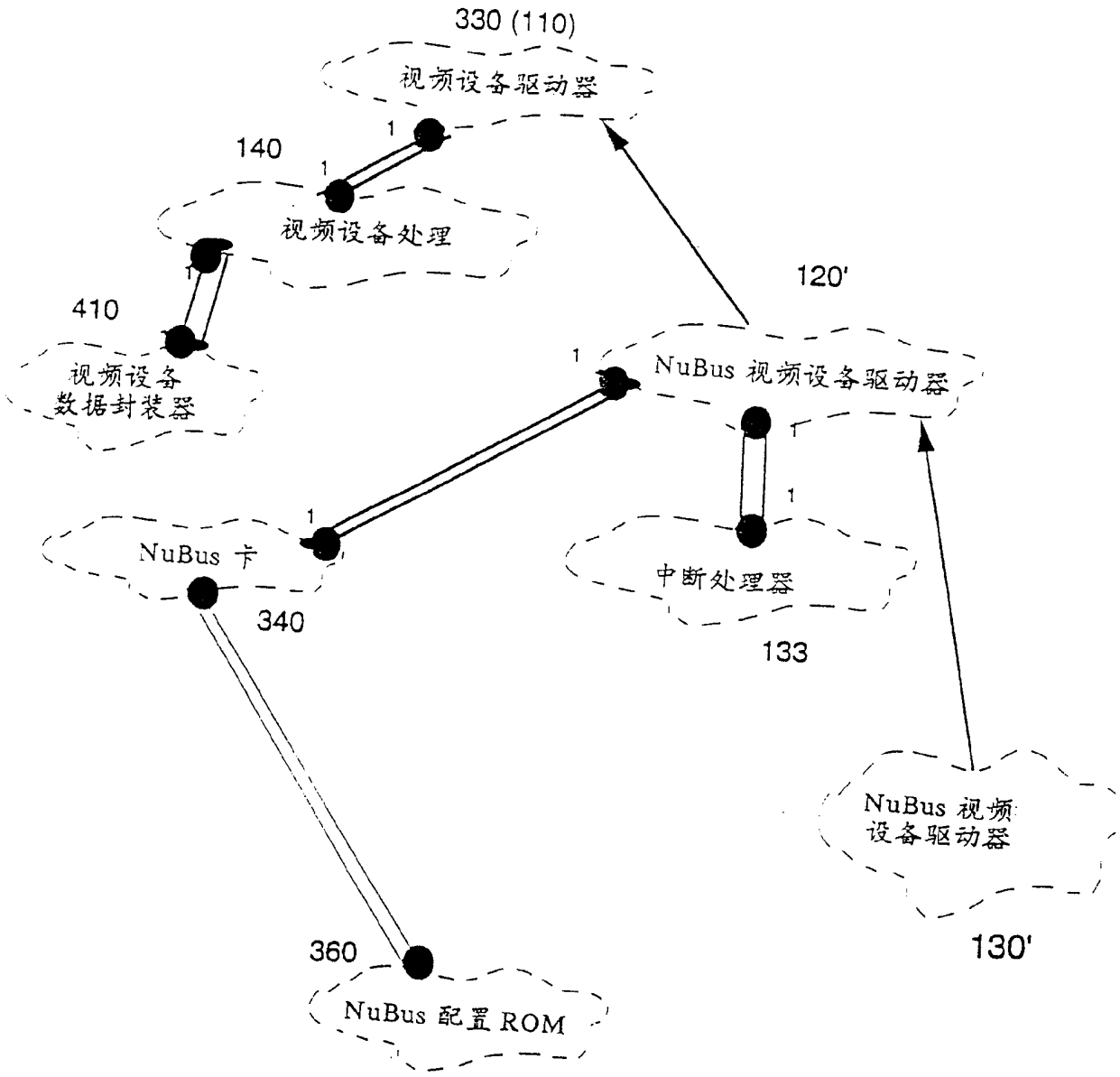

330 (110)
视频设备驱动器
140
视频设备处理
410
视频设备
数据封装器
120'
NuBus 视频设备驱动器
NuBus 卡
340
中断处理器
133
NuBus 视频
设备驱动器
130'
360
NuBus 配置 ROM

图 4

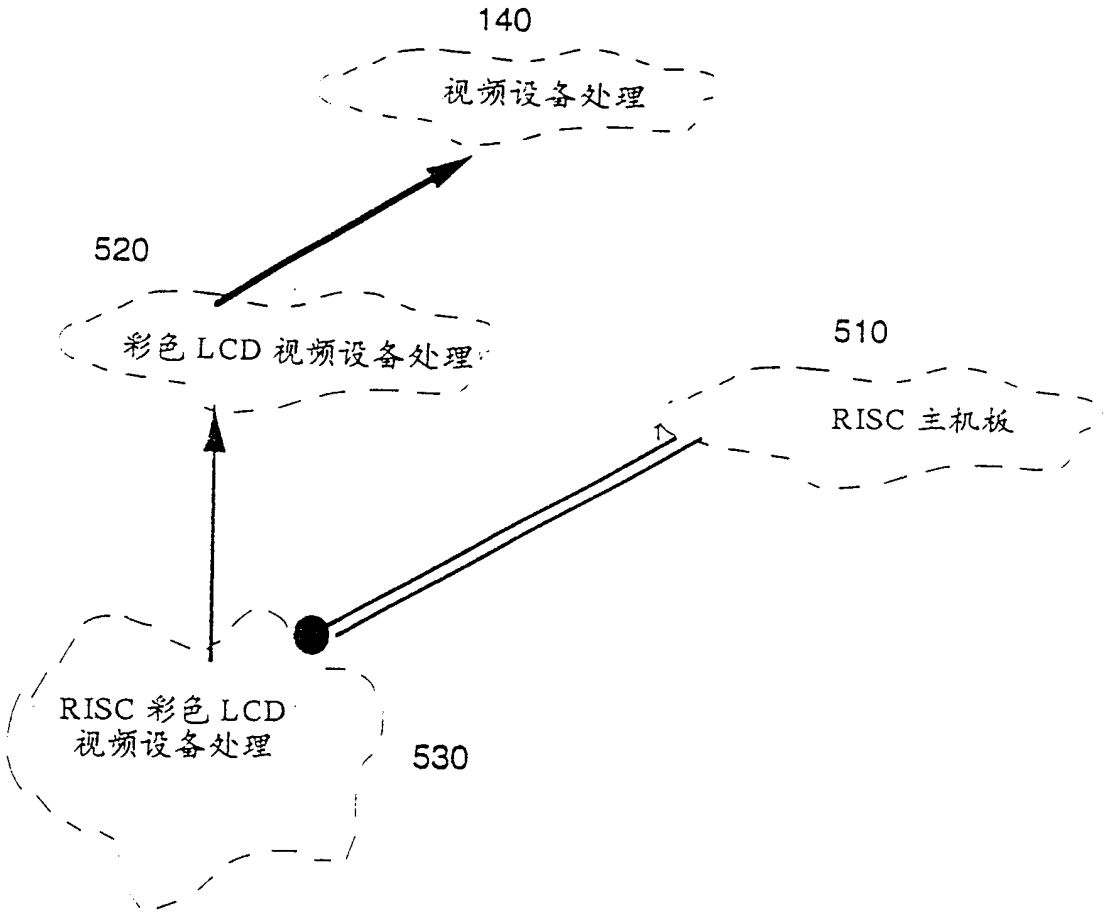

140
视频设备处理
520
彩色 LCD 视频设备处理
510
RISC 主机板
RISC 彩色 LCD
视频设备处理
530

图 5

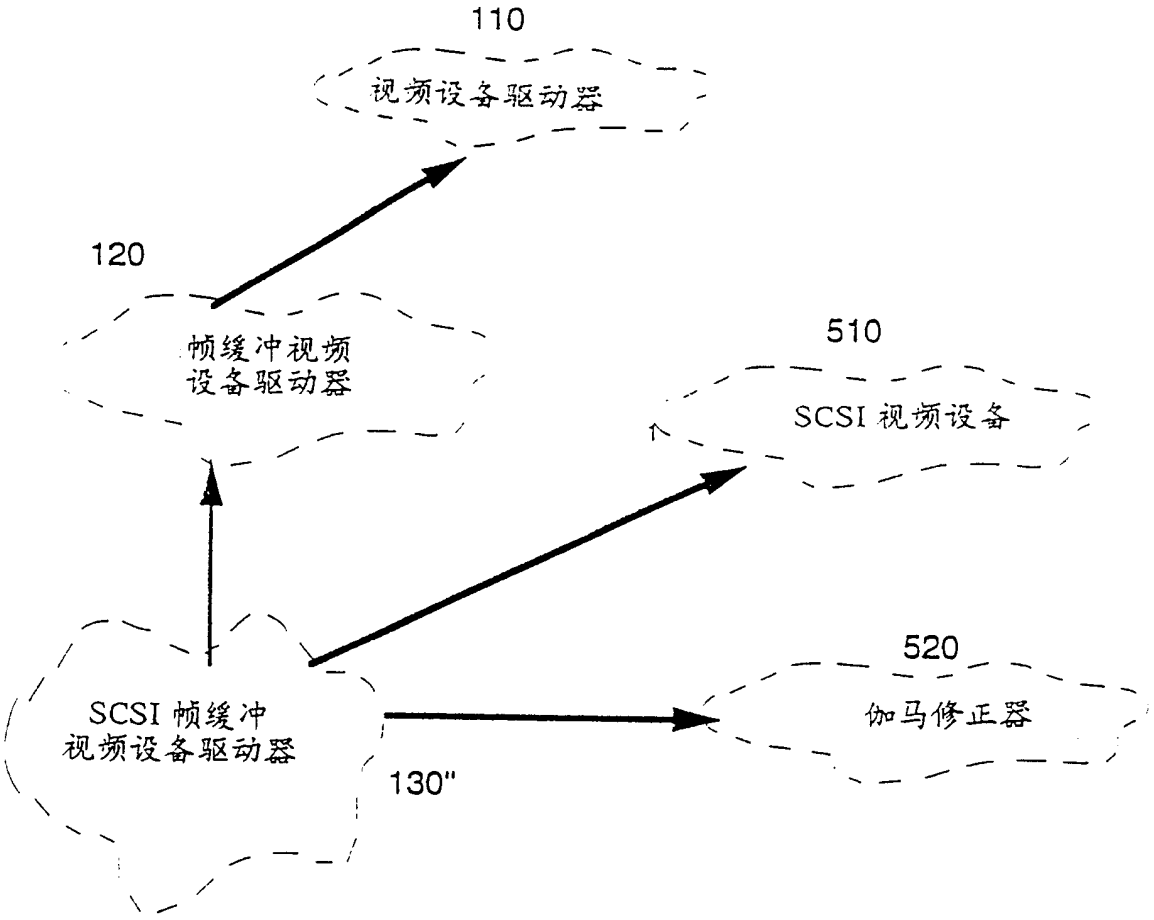

110
视频设备驱动器
120
帧缓冲视频
设备驱动器
510
SCSI 视频设备
520
伽马修正器
SCSI 帧缓冲
视频设备驱动器
130"

图 6

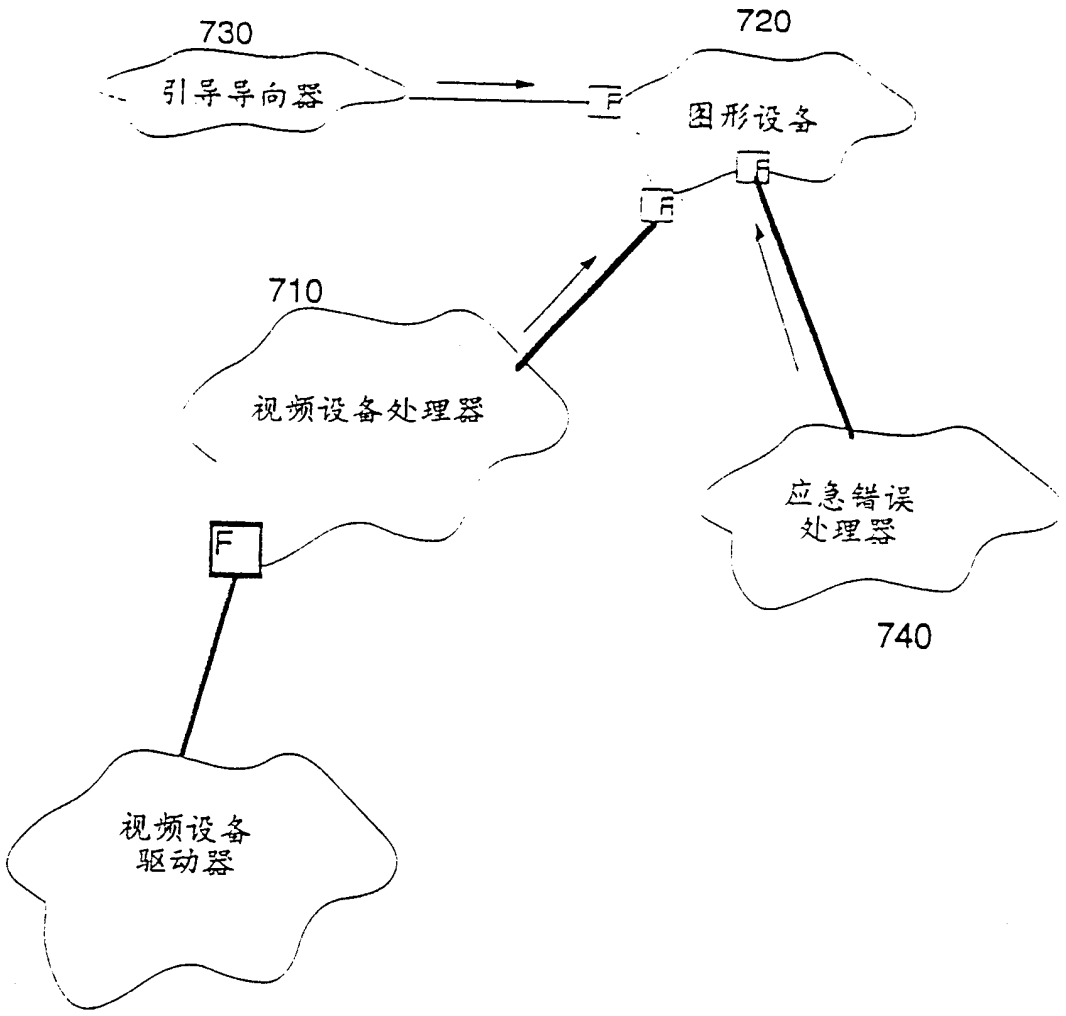


图 7

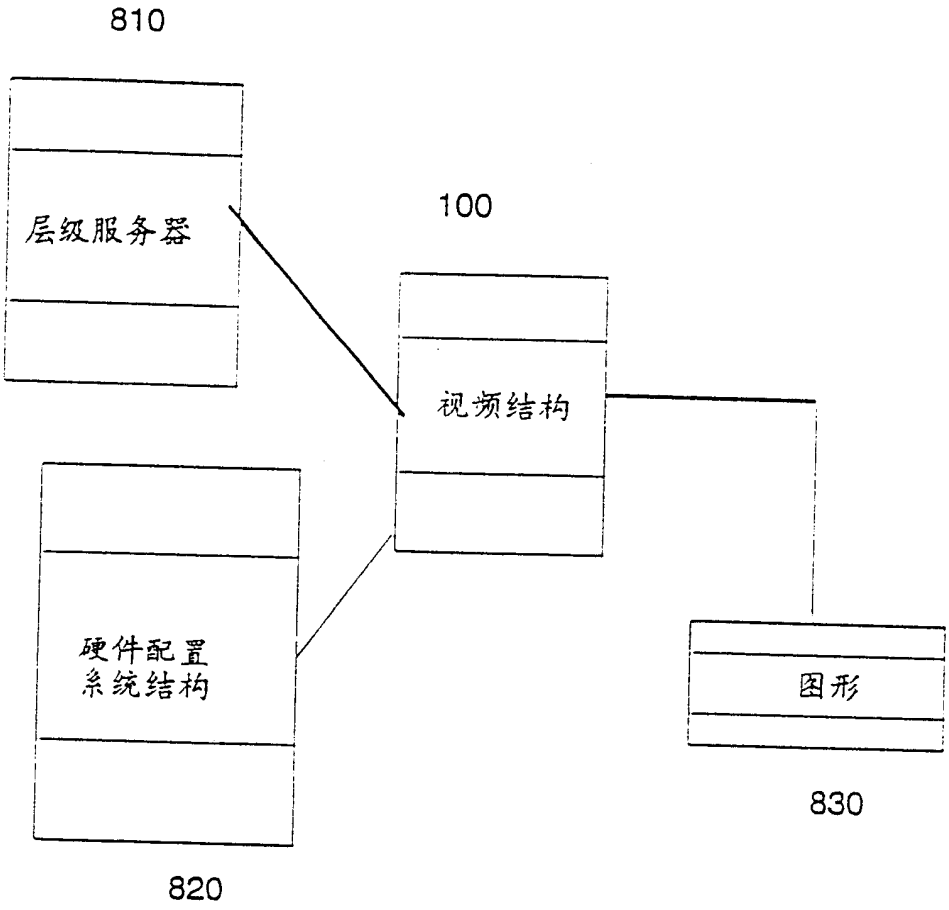


图 8

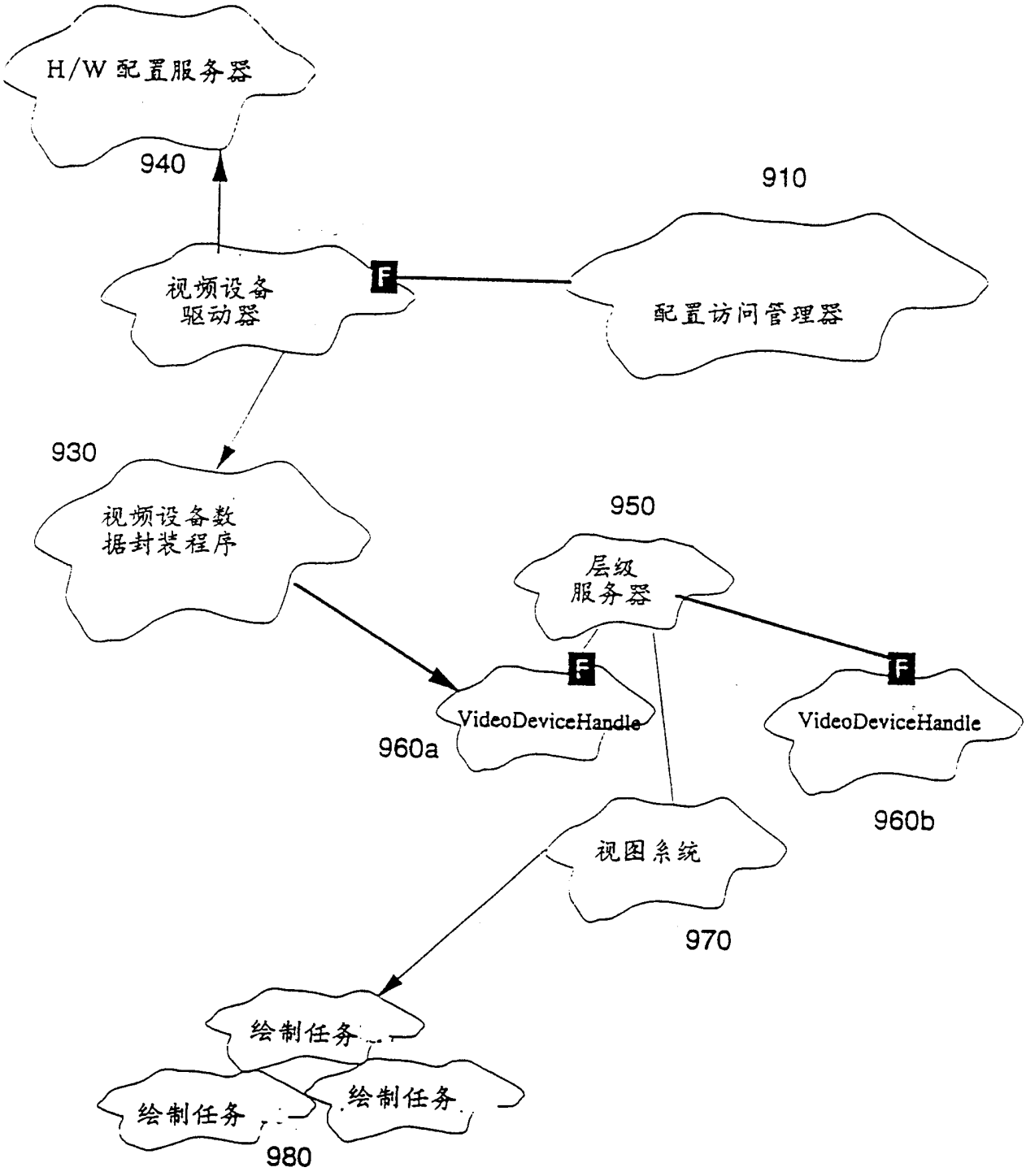

H/W 配置服务器
940
910
视频设备
驱动器
F
配置访问管理器
930
视频设备数
据封装程序
950
层级
服务器
F
F
VideoDeviceHandle
960a
VideoDeviceHandle
960b
视图系统
970
绘制任务
绘制任务
绘制任务
980

图 9