



## [12] 发明专利说明书

专利号 ZL 99805163.2

[45] 授权公告日 2008 年 6 月 4 日

[11] 授权公告号 CN 100392588C

[22] 申请日 1999.1.22 [21] 申请号 99805163.2

[30] 优先权

[32] 1998.2.20 [33] US [31] 09/026,862

[86] 国际申请 PCT/US1999/001398 1999.1.22

[87] 国际公布 WO1999/042924 英 1999.8.26

[85] 进入国家阶段日期 2000.10.18

[73] 专利权人 英特尔公司

地址 美国加利福尼亚州

[72] 发明人 M·R·菲克特纳

M·A·莫哈梅德 D·R·库斯

E·P·托马斯策夫斯基

[56] 参考文献

US 5477264A 1995.12.19

US 5226145A 1993.7.6

审查员 邓 茜

[74] 专利代理机构 中国专利代理(香港)有限公司

代理人 吴立明 陈景峻

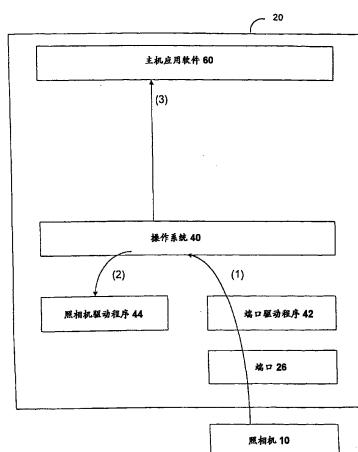
权利要求书 3 页 说明书 13 页 附图 10 页

[54] 发明名称

自动更新照相机固件的方法和系统

[57] 摘要

本发明公开了一种成像装置(10)和主机操作系统(20)之间的固件的更新方法。主机系统检测成像装置的固件与主机系统的配置不兼容性。如果检测到了不兼容，更新的固件图像就从主机系统传送到成像装置。



1. 一种成像装置的固件更新方法，该方法包括以下步骤：

当在成像装置和主机系统之间建立连接时，自动检测成像装置的固件版本是否与主机系统的固件版本匹配；

如果成像装置的固件版本不匹配，

则通过代之以从主机系统传送到成像装置的固件映像来自动更新成像装置的固件，并

执行被更新的固件；以及

如果成像装置的固件版本匹配，那么执行成像装置的固件。

2. 如权利要求 1 的方法，其中检测步骤还包括：

从成像装置中接收装置接口信息；以及

比较装置接口信息与主机系统中存储的接口信息。

3. 如权利要求 1 的方法，其中更新步骤还包括：

将固件映像从主机系统装载到成像装置的一个缓冲存储器；

检测被装载的固件映像中的错误；以及

如果被装载的固件映像没有错误，那么用被装载的固件映像刷新成像装置中的固件。

4. 如权利要求 3 的方法，其中如果被装载的固件映像不是没有错误，那么重复更新步骤。

5. 如权利要求 1 的方法，其中如果成像装置中的固件映像是一个比主机系统中的固件版本更早的一个固件版本，那么执行更新步骤。

6. 如权利要求 1 的方法，还包括步骤：

通过所述建立的连接从主机系统中装入至少一个配置表到成像装置中。

7. 如权利要求 6 的方法，其中至少一个配置表用于处理成像装置中的图像数据。

8. 如权利要求 1 的方法，其中执行更新固件的步骤是通过复位到引导装载器的开头而实现的。

9. 一种更新照相机固件的方法，该方法包括以下步骤：

连接照相机到主机系统；

检测照相机的固件版本是否与主机系统的固件版本匹配；

如果照相机的固件版本不匹配，那么通过代之以从主机系统传送到照相机的固件映像来自动更新照相机的固件，并执行被更新的固件；以及

如果照相机的固件版本匹配，那么执行照相机的固件。

10. 如权利要求 9 的方法，其中检测步骤包括：

从照相机中接收装置接口信息；以及

比较装置接口信息与主机系统中存储的接口信息。

11. 如权利要求 9 的方法，还包括步骤：

当照相机和主机系统已被连接时从主机系统中装入至少一个配置表到照相机。

12. 如权利要求 11 的方法，其中至少一个配置表用于处理成像装置中的图像数据。

13. 一种更新成像装置中的固件的系统，包括：

检测成像装置的固件版本是否与主机系统的固件映像的固件版本匹配的装置；

通过代之以从主机系统传送到成像装置的固件映像来自动更新成像装置的固件的装置，所述更新当成像装置的固件版本不匹配时进行，和

执行被更新的固件的装置；以及

当成像装置的固件版本匹配时执行成像装置的固件的装置。

14. 一种更新成像装置的固件的方法，该方法包括以下步骤：

根据来自主机系统的请求，向主机系统发送成像装置中的固件版本；

如果主机系统向成像装置传送固件映像，

通过代之以来自主机系统的固件映像来自动更新成像装置的固件，

执行被更新的固件；以及

如果主机系统不传送固件映像，则执行成像装置的固件。

15. 如权利要求 14 的方法，其中更新步骤包括：

将固件映像从主机系统装入缓冲存储器；检测被装载的固件映像中的错误；以及

---

如果被装载的固件映像没有错误，那么用被装载的固件映像刷新固件。

16. 如权利要求 15 的方法，其中如果被装载的固件映像不是没有错误，那么重复更新步骤。

17. 一种更新成像装置的固件的系统，包括：

根据来自主机系统的请求，向主机系统发送成像装置中的固件版本的装置；

通过代之以从主机系统传送来的固件映像来自动更新成像装置的固件的装置，所述更新当主机系统向成像装置传送固件时进行，和

当主机系统传送固件映像到成像装置时执行被更新的固件的装置；以及

当主机系统不传送固件映像时执行成像装置的固件的装置。

18. 一种更新成像装置的固件的系统，包括：

一个接口，用于检索成像装置中固件的固件版本的标识符；以及

在主机系统中的更新逻辑，其被配置成：如果成像装置的固件版本与主机系统中的固件映像的固件版本不匹配，那么通过代之以从主机系统传送到成像装置的固件映像来自动更新成像装置的固件。

19. 如权利要求 18 的系统，还包括：

成像装置中的缓冲存储器，用于接收来自主机系统的固件映像；以及

成像装置中的引导装载器，用于检验由缓冲存储器从主机系统接收的固件映像中的错误，其中，代之以传送到成像装置的固件映像还包括：如果引导装载器未在固件映像中检测到错误，则代之以固件映像。

## 自动更新照相机固件的方法和系统

### 技术领域

本发明涉及到成像领域，进一步说，本发明涉及到成像装置和主机系统之间的固件的更新。

### 背景技术

成像装置，如照相机，都是将静止的或活动的（视频）图象存储到胶片、录像带或其它的媒体上。数字相机以数字形式捕捉图像信息并存储到存储中，如快速存储器，或其它的数字存储媒体中。数字图像信息可以下载到主机系统，如个人计算机中。然后可以通过主机系统中的应用软件对数字图像进行旋转、剪切、或修改操作。

成像装置包括能使成像装置与主机软件通信的固件。这种固件包括不同操作功能的指令。例如，固件可以被用于决定图像的曝光程度、以某种特定的方式着色、压缩图像数据、保存能量、自检、和/或规定照相机存储媒体的存取和设置协议。

经常，需要用新发布的软件或固件来更新主机软件和照相机固件，或其中一。通常的软件更新的方法是通过使用补丁程序或维护软件包来完成。该方法是通过软盘、CD - ROM 或通过互联网来提供一套程序。维护软件包在运行时修改主机软件所需更新的部分。

固件的更新有很多问题。一般的更新方法是由用户通过手工来完成的。这可能设计到运行一个执行程序，然后重新设置成像装置。固件的手工更新是不方便的，可能会导致因固件和主机系统软件版本的不兼容而引起的错误。

### 发明内容

本发明公开了一种与主机系统相连接的成像装置的固件更新方法。主机系统可以检测成像装置的固件与主机系统的配置兼容性。当检测到不兼容时，就会从主机传输一个更新的固件图像到成像装置。在一个实施方案中，更新的固件图像是一个比被替换的固件图像更早的版本。

本发明的其它特点和优越性将在附图及随后的详细说明中陈

述。

#### 附图说明

图 1, 图示一台典型的可以与主机系统连接的成像装置。

图 2, 图示成像装置 10 首次连接到主机系统 20 时主机系统部件中的信息流动一个实施方案。

图 3, 图示主机应用软件 60 启动后主机系统部件中的信息流动一个实施方案。

图 4, 图示查询启动过程的一个实施方案。

图 5, 图示照相机 API62 和主机应用软件 60 之间的查询过程的一个实施方案。

图 6, 图示检测有否兼容的成像装置连接到主机系统的流程图。

图 7, 图示一个实施方案的固件更新流程图。

图 8, 图示一个实施方案的固件导入过程的状态转换框图。

图 9, 图示存储固件的典型的非易失性的存储器 400 的框图。

图 10, 图示启动主机应用软件建立与照相机之间通信过程的一个实施方案的流程图。

图 11, 图示检测有无成像装置, 如照相机, 连接到主机系统的一个实施方案的流程图。

#### 具体实施方式

本发明公开了一种成像装置和主机系统之间的固件的更新方法。固件包括了用于控制嵌入系统, 如成像装置, 的指令。在一个实施方案中, 固件的更新是通过将成像装置连接到主机系统来自动实现的。这一对用户而言简化了操作, 同时保证了成像装置和主机软件的兼容。固件更新可以提供“故障”定位、改进运算、改进颜色传感、改进数据压缩, 采用新的选取和设置存储方式的协议等等。正如后面将要介绍的, 当多个不同固件版本的成像装置连同拥有不同软件版本的主机系统一起使用时, 固件的自动更新特别地有用。

下面的描述通过一个系统来说明固件的更新, 该系统把成像装置连于主机系统来实现图象信息在成像装置和主机系统之间的自动传输。但是, 固件的更新并不局限于这样的系统。

成像装置可以是一种图像捕捉装置, 如照相机。另外, 这项公开的技术可以适用于任何能够存储图像信息的装置。主机系统可以

是任何可以处理图像信息的系统。例如，主机系统可以是一台个人计算机，如采用 Pentium 或 Pentium II 处理器的 IBM 兼容个人计算机。不仅如此，主机系统还可以是打印机、绘图机、传真机、显示装置或存储装置。

为清楚起见，在下面的说明中把图像捕获装置用照相机，主机系统用计算机来描述。但应知道，也可以采用其他的成像装置和主机系统。

图 1 图示的是一台可连接到主机系统 20 的成像装置 10。在一个实施方案中，成像装置 10 通过电缆 22 连接到主机系统 20 的端口 26。成像装置 10 最好用一个支持高数据传输率的数据传输协议来与主机系统 20 相连接。在一个实施方案中，成像装置 10 通过一条通用串行总线（USB）连接到主机系统 20。USB 连接能提供高达 12Mb/s 的数据传输率。也可采用其它的连接和数据传输协议，如 1394 协议。（更多的 USB 信息可以通过互联网从网址：<http://www.usb.org/> 上获取。1394 标准是由电气和电子工程师协会即 IEEE 来维护和发布的。Fireware，是由 IEEE 标准 1394 - 1995 定义的一种 1394 实施方式。）

图 2 和 3 所示的实施方案是主机系统 20 的部件和成像装置（照相机）10 之间的关系和消息传递。图 2 图示的是成像装置 10 首次和主机系统 20 连接时主机系统部件的信息流动的一个实施方案。主机系统 20 包括一个操作系统（O/S）40 和主机系统应用软件 60。主机系统 20 检测成像装置如照相机 10 何时连接到主机系统 20。在一个实施方案中，操作系统 40 通过查询端口 26 来检测照相机 10 是否连接到系统中。端口驱动程序 42 可以用来提供操作系统 40 和端口 26 之间的接口。在一个实施方案中，端口 26 是一个 USB 端口，并且端口驱动程序是一个 USB 驱动程序。

操作系统可以是各种不同的操作系统其中之一。在一个实施方案中，操作系统为 Windows\* 操作系统，如微软的 Windows\* 95 和 Windows\* 98 系统。Windows\*98 包含可以查询端口的异常分支。其它的操作系统可以在改进后进行类似的端口查询。端口查询最好在后台进行，这样用户无需察觉这一过程。也可采用主机应用软件 60 来进行端口 26 的查询。但通过操作系统 40（而不是用主机应用软件 60）进行端口查询具有一个操作上的优势，这是由于操作系统已经

设定好不同的查询功能，如键盘的按动、鼠标的移动等等。为说明原理起见，以下的描述假定由操作系统来完成查询。本领域的熟练技术人员可以通过改进，使应用软件实现查询功能。

当照相机 10 连接到主机系统 20 的端口 26 时，端口驱动程序 42 发出信号通知操作系统 40：照相机已连接到主机系统 20。图 1 中标号（1）的箭头图示了这一过程。操作系统 40 确认装置为照相机并将相应的驱动软件 44 调入内存，如图中箭头（2）所示。在一个实施方案中，操作系统 40 询问照相机 10 得到一个标识符。根据标识符调入驱动软件 44。该例中，照相机驱动程序是通过操作系统 40 调入的。

然后，操作系统 40 调入一个和多个与照相机相应地应用软件。在一个实施方案中，操作系统允许软件应用注册。在遇到预先确定的情况（如检测到带有特定标示的照相机），注册过的主机应用软件即被调入。这样，主机应用软件 60（针对照相机的）被装入，如图中箭头（3）所示。在一个实施方案中，照相机驱动程序 44 发出信号通知操作系统 40 启动主机应用软件 60。主机应用软件 60 开始开始在成像装置（照相机）10 和主机系统 20 之间传输图像信息。主机应用软件 60 也可以处理图像。例如，主机应用软件 60 可以进行图像解压缩和色彩校正，或进行两者之一。而且主机应用软件还可以执行旋转、剪切和其它的图像操作功能。

有些操作系统，如 Windows 98 允许由特定的事件引发应用软件的运行。例如，照相机驱动程序 44 可以用注册事件来设定，如“检测到与照相机相连”。或“照相机快门被按动。”这样，操作系统可以被设定为当照相机 10 连入时自动调入一个程序，如主机应用软件。

在一个实施方案中，如果当照相机 10 被连接到主机系统 20 时主机系统没有安装照相机驱动程序 44 或主机应用软件 60，就要求用户为连接到端口 26 的装置提供照相机驱动程序 44 和主机应用软件 66，或其中之一。一旦安装完成，系统将按前面描述的过程运行。

图 3 图示的是主机应用软件 60 启动后主机系统部件之间的信息流动的一种实施方案。在一种实施方案中，主机应用软件 60 装入后，生成并启动一个照相机应用编程界面（API）62，如图中箭头（4）所

示。照相机 API62 可以通过后台线程完成工作。在这种方式中，主机应用软件 60 不用等到照相机 API62 运行结束即可以执行其它的任务。在一种实施方案中，照相机 API62 是一个 COM 目标码，它可以装入一个动态连接库 (DLL) 64, 如图中箭头 (5) 所示。DLL 可以是基于 O/W 的。照相机 API62 通过 DLL64 与操作系统 40 进行通讯。(另一实施方案中，照相机 API62 是与 DLL64 合并在一起的。) 反过来，主机系统 40 通过照相机驱动程序 44 和端口驱动程序 42 与照相机 10 进行通信，如图中箭头 (6) 所示。

图 4 图示查询功能启动过程的一种实施方案。通过主机系统打开主机应用软件来启动查询功能。然后主机应用软件 60 产生并启动照相机 API62。在一种实施方案中，主机应用软件 60 将自身添加到照相机 API 的回调列表中，这样主机应用软件 60 将意识到什么时候照相机 API 查询成功。

在一种实施方案中，照相机 API 启动后，重新设置内部变量，装入 DLL，产生并开始后台线程。然后照相机 API 在后台线程的队列中插入信息，试图打开照相机驱动程序。（驱动程序通过建立照相机 API 和驱动程序间的连接来“打开”。）在一种实施方案中，照相机驱动程序只有在有照相机连接时才可以打开：如果照相机驱动程序不能打开，则表明照相机没有连接到主机系统中。如果照相机驱动程序能够打开，说明照相机是连接到主机系统上的。在一种实施方案中，照相机 API44 每隔半秒钟就进行一次打开照相机驱动程序的尝试。

图 5 所示的是在照相机 API62 和主机应用软件 60 之间的查询过程一种实施方案。在这一实施方案中，照相机 API62 试图去打开照相机驱动程序 (CM-SIGNAL-STATUS)。当成功打开照相机驱动程序时，照相机 API 关闭照相机驱动程序并在应用软件的回调队列中发出通知。由于主机应用软件是在照相机 API 的回调队列中，因此会得知已经检测到照相机。

在这一实施方案中，主机应用软件 60 通过发出信号通知照相机 API62 打开照相机驱动程序 44 并检测兼容的照相机来重新打开照相机驱动程序 44 (CM-OPEN-DRIVER)。而后主机应用软件 60 可以通过照相机 API62 (以及操作系统 40 和驱动程序 44、42) 给照相机 10 发

出不同的指令。例如，主机应用软件 60 可以要求获得存储于照相机中的图像数量 (CM\_GET\_NO\_OF\_IMAGES)。主机应用软件 60 可以要求获得图像名称和大小的列表 (CM\_GET\_IMAGE\_LIST)，或要求获得某一幅特定的图像 (CM\_GET\_IMAGE\_BY\_NAME)。

在一种实施方案中，照相机 API62 检测有无兼容的成像装置连接到主机系统中，如果需要，则自动更新装置中的固件。根据生产厂家的出厂设定，固件的自动更新可以是禁止的或允许的，或者是可以由用户改动的。

图 6 所示的是检测是否有一个兼容的成像装置连接到主机系统的过程的一种实施方案的流程图。在一种实施方案中，这一过程是由照相机 API62 来执行的。如前所述，照相机 API62 是由主机应用软件 60 生成和启动的，如模块 100 中所示。照相机 API62 在模块 102 中装入 DLL，该 DLL 可以是从属于的操作系统的，主机应用软件 60 在模块 104 中给的照相机 API62 发出一个“打开”指令。作为响应，照相机 API 通过 DLL64 打开照相机驱动程序 62，如模块 106 中所示。

模块 108 中，照相机 API62 给照相机发出一个指令，取得照相机接口编号。在一种实施方案中，唯一的照相机接口编号值被赋给照相机所支持的一组指令。

在决策模块 110 中操作继续进行。将从照相机中的得到的接口编号与照相机 API62 的接口表进行比较。如果接口编号没有存储于照相机 API62 接口表中，那么照相机 API62 关闭照相机驱动程序并通知主机应用软件：照相机是不兼容的，如模块 112 和 114 中所示。照相机 API 不能与照相机通信，因为不知道照相机所支持的指令。

模块 110 中，如果从照相机中得到的接口编号是存储于照相机 API62 的接口列表中，则照相机 API62 给照相机发出一条指令获得硬件版本号，如模块 120 中所示。

模块 130 中，将从照相机中得到的硬件版本号与照相机 API62 中存储的硬件表进行比较。如果从照相机中得到的硬件版本号没有存储在照相机 API62 的硬件表中，那么照相机 API62 不能进行更新检测。照相机 API62 发出信号通知主机应用软件照相机是可以自由存取的，如模块 132 中所示。照相机 API62 能够与照相机通讯，因为它的接口是兼容的，但是不能更新固件，因为不能识别硬件配置。

模块 130 中，如果从照相机得到的硬件版本号是存储在照相机 API62 的硬件表中的，那么继续进行模块 140 的操作。在模块 140 中，照相机 API 给照相机发出一条指令返回固件的版本号。

接下来进行模块 150 的操作，如果固件版本号是照相机 API62 中存储的版本号，则表明照相机已经进行了更新。照相机 API62 发出信号通知主机应用软件，照相机是可以自由存取的。

如果固件版本号不是照相机 API62 中存储的版本号，则继续执行模块 160 的操作。如果制造厂商不允许固件更新，则照相机 API62 不能执行固件更新操作并通知主机应用软件照相机是可以自由存取的。例如，制造商可能在寄存器中设置一个禁止固件更新的位。尽管安装的固件可能不是最新的版本，但是只要接口和硬件是兼容的，所安装的固件仍能用主机软件来操作。

模块 170 中，检测是否在此之前曾尝试过更新并失败。在一种实施方案中，预设定更新尝试的次数。如果曾尝试过更新并失败，那么照相机 API62 通知主机应用软件更新失败，照相机是可以自由存取的，如模块 172 所示。

模块 170 中，如果还没有尝试过更新，则按图 7 所示的流程图进行固件的更新过程。

图 7 所示的是一种更新固件方案的流程图。在模块 200 中，照相机 API 传送一幅更新过的固件图像到照相机。在一种实施方案中，照相机 API 发出一条固件下载指令到照相机，以使固件准备好接收新的固件图像。然后更新的固件图像从主机系统传送到照相机。在一种实施方案中，更新的固件图像被存储到照相机的临时缓冲区，如易失存储器中。

模块 202 中，照相机 API62 关闭照相机驱动程序。这样即关闭了照相机和主机系统之间的通讯联系，以使固件的更新不被干扰。照相机 API 然后通知主机应用软件固件正在更新中，如模块 204 所示。照相机 API 开始查询固件重新建立与固件的联系。

固件由一个引导模块和一个代码模块组成。引导模块，也称为引导装入程序，在固件更新时是不被更换的。只有代码模块被更换。引导装入程序维护固件更新的例程。在模块 206 中，引导装入程序验证更新的固件图像没有错误。例如，可以通过产生一个校验和来

完成。然后引导装入程序用更新的固件图像替换旧的（已安装）的固件，如模块 208 所示。在一个实施方案中，这一过程是通过将易失存储器中的固件图像传送到非易失存储器来完成的。然后引导装入程序进行一次复位，或跳到引导装入程序的开头启动新近更新的固件，如模块 210 所示。这样使引导装入程序重启动，此时引导装入程序检测已更新固件有无错误，如模块 212 所示。如果校验和正确，则执行代码模块中的固件，如模块 214 和 218 中所示。

如果校验和不正确，引导装入程序重新建立与主机系统的连接并转移到等待状态，如模块 216 所示。在这个等待状态下，照相机 API 可以从引导装入程序得到状态情况以决定新的固件图像是被正确地装入到代码模块中，还是存在问题，引导装入程序处于等待状态。照相机 API 可以试图进行另一次固件更新。

模块 218 进程中，一旦固件能够根据其代码模块中正确的校验和来重新启动，那么固件就能建立与主机系统的连接。照相机 API 将通过查询检测到照相机连接到模块 220。照相机 API 可以重新打开照相机并执行照相机的存取。

图 8 所示的是固件引导过程的状态转换图表的一个实施方案。状态转换图表划分为图 8 中左侧的引导模块状态和右侧的代码模块状态。在状态 300 时，一个复位操作启动照相机的硬件。这可能包括复位寄存器和关闭照相机的某些部件或其中之一，以使照相机处于已知状态。从状态 300 转换到状态 302，对代码模块进行检测。在一个实施方案中，对代码模块进行一次校验和操作。如果校验和通过，则执行代码模块中的指令，如状态 304 所示。代码模块中的固件能够建立与主机系统的连接并处理从主机接收到的不同的指令。如果代码模块中的固件检测到主机系统发出下载命令，则转换到状态 306，在该状态下，照相机硬件被初始化为一个已知的状态。例如，可以关闭传感器和频闪观测器，打开 DRAM。在一个实施方案中，当不使用时为节能关闭照相机硬件。状态转换到模块 308，在该状态下，从主机系统接收更新的固件代码。在一个实施方案中，更新的固件代码临时存储于 DRAM 中。

如果状态 308 中有通信问题，那么转换到等待状态 310。在等待状态 310，引导模块等待从主机发来的指令。

如果校验和失败，状态 302 也会转换到等待状态 310。在等待状态 310，引导模块等待从主机系统发来的指令。例如，主机系统可能需要引导模块的状态信息。当主机系统意识到引导模块处于等待状态 310 时，能够发出另一个下载指令，使其转换回状态 308。

如果固件图像下载无错误，状态 308 转换到状态 320，该状态下检测下载的代码有无错误。在一种实施方案中，执行校验和操作。如果校验和失败，引导模块回到状态 310。如果校验和通过，则状态转移到模块 322，此状态中代码模块被清除。随后，代码模块在状态 324 下被重新编写。从状态 324 回到状态 300，重新启动照相机硬件，并且如前面所述，状态转换图表重新开始。

图 9 所示的是一个典型的非易失性存储器 400 的图表。在一个实施方案中，非易失性存储器是一个快速存储器。在一个实施方案中，引导模块 402 和代码模块 404 存储于快速存储器，引导模块被锁定，以免被重写。代码模块是可以重新编写的。

在一个实施方案中，非易失性存储器也存储一个或多个图像表 406，图像表可以含有与照相机配置相关的信息。

在一个实施方案中，快速存储器中存储下列图像表。

- 1) 完全像素表
- 2) 编码器表
- 3) 曝光表
- 4) 压缩扩展器表
- 5) 颜色校正表

图像表允许处理照相机中的一幅图像。例如，完全像素表可能包含不能正常工作的照相机中的详细的像素信息。例如，完全像素表可以在制造测试时确定。照相机也可以动态地确定完全像素表。完全像素表允许使用临近的像素来修补有缺陷的像素。

类似地，其它的图像表也允许图像数据的处理。例如，编码器表可以给一幅图像的编码和压缩指定特殊的值，曝光表可以包含曝光时间、放大率、频闪观测器的使用和亮度的信息，压缩扩展器表可以包含用于从使用字节第一数位的位图表示到使用字节较小数位的位图表示的映射数据的信息，颜色校正表可用来校正颜色。

在一个实施方案中，图像表包含与一个特定的发光体相关的数

值。例如，图像表可以包含一幅与在阳光下，或荧光灯照明下，或钨丝灯照明下所拍摄的图像相对应的不同的数值。

图像表在制造的过程中可以存储到非易失性存储器 400，但可以在以后通过固件进行更新。在一个实施方案中，固件可以删除非易失性存储器中的图像表并从主机系统传送新的图像表到照相机的非易失性存储器中。固件接收主机系统发来的指令，这些指令表明一个表是否被传送到照相机，或一个图像表是否被删除。

在一个实施方案中，由于一些非易失性存储器（即一些快速存储器）不能够同时从中既读取数据又向非易失性存储器写入数据，所以一个 DRAM 被用来临时存储非易失性存储器中的代码。固件将一部分代码复制到 DRAM，然后传送执行程序到 DRAM。然后一个执行 DRAM 的指令的微控制器能够在到非易失性存储器写入新的图像表，随后执行程序被传送到非易失性存储器的代码段。

表 1 和表 2 所示的例子说明不同照相机和个人计算机上的不同版本的接口、固件和照相机 API 是如何能够互相联系的。表 1 表示的是存储于第一个照相机 API (CAMAPI # 400) 中的配置。表 2 表示的是存储于第二个照相机 API (CAMAPI # 420) 中的配置。一些使用这些表格中的配置信息的不同的实施方案的例子有助于说明这一问题。

表 1

接口标识符	硬件版本	硬件版本/固件 图像
0x0101	1.00	1.00/文件名 3
	1.01	1.02/文件名 4
	1.02	1.07/文件名 5
0x0202	2.00	1.08/文件名 6

表 2

接口标识符	硬件版本	硬件版本/固件 图像
0x0303	3.00	1.09/文件名 8
	3.01	1.02/文件名 8
	3.02	1.07/文件名 12
0x0404	4.00	1.08/文件名 13

例 1：假设两个用户，A 和 B，有同样的主机应用软件版本和同样的固件版本，设软件版本为 1.0，产生具有表 1 中的接口标识符表的 CAMAPI # 400。接口标识符为 0x0101，而固件版本为 1.00。接下来，用户 A 接受一次更新，将其主机应用软件更新为 1.02 版，该版本具有同样的接口标识符（0x0101）。

主机应用软件更新后，用户 A 首次试图访问照相机，照相机 API62 将检测到一个不兼容的固件版本，这将触发如前面所述的自动固件更新。

接下来，假设用户 B（仍在他的照相机中使用 1.00 版的固件）将他的照相机接入 A 的个人计算机。A 的计算机上的照相机 API 将再次检测固件和软件版本。照相机 API 将把 B 的照相机固件的版本更新为 1.02。B 的照相机当前则有一个与 B 自己的计算机上的主机软件不一致的固件（1.02）。当 B 将他的照相机带回自己的计算机时，照相机 API62 将不能识别固件 1.02 版本的固件。照相机 API62 将传送 1.00 版的固件图像到 B 的照相机。这样，B 的照相机恢复到原来固件，以使照相机能与 B 的计算机上的应用软件进行通讯。

对 B，固件更新是透明的。B 既可以在 A 的更新了的系统中，也可以在有老版本主机应用软件的 B 的系统中使用他的照相机。

例 2：假设 A 拥有 1.00 版本的固件，接口标识为 0x0101，而 B 拥有 1.08 版本的固件，接口标识为 0x0202。除非 A 为了识别新的接口 0x0202 而更新了她的应用软件，B 将不能使用 A 的个人计算机。如果 B 的个人计算机使用的照相机 API 包含表 1 中所有修定版本，则 B 的个人计算机既可用于 A 的照相机，也可用于 B 的照相机。

例 3：如果用户 A 拥有 1.0 版本的固件并使用 CAMAPI # 400，如表 1 中所示，而用户 B 拥有 3.0 版本的固件并使有 CAMAPI # 420，如

表格 2 中所示，那么为了对两台照相机都能够存取，一台个人计算机必须有两个版本的应用软件，通过各自的接口与照相机相接。

在一个实施方案中，接口具有唯一的标识符。以一个 TWAIN 接口为例，来描述一个图像应用软件是如何从扫描仪或其它的数字成像装置中获取图像的。当前的 TWAIN 版本为 1.7。TWAIN 是通过几个工业合作者共同制定的一组标准来维护的。可以通<http://www.twain.org> 来获取更多的 TWAIN 信息。也可以使用有专利权的接口，例如，可以采用 DLL，DLL 支持特定的应用软件，使应用软件能够传送和删除从成像装置得到的图像。

硬件标识符指定硬件的变动，例如可以在成像装置上附加一台步进电动机。可设置固件来跟踪较小的和较大的改变。比如，较小的改变可能是缺陷修定或改进，较大的变动可以是添加新的硬件。

图 10 所示的是启动主机应用软件建立与成像装置的通信过程的一种实施方案流程图。在这个公开的实施方案中，成像装置是一台照相机，主机系统是一台个人计算机。流程图从模块 500 开始。继续运行到模块 502，在模块 502 中，主机应用软件生成一个照相机 API。照相机 API 在模块 502 中装入一个可以是依从于操作系统的 DLL。在模块 506 中，照相机 API 测定照相机是否有效。

在模块 508 中，如果照相机是无效的，则流程图返回到模块 506。但如果照相机是有效的，程序继续执行到模块 510。在模块 510 中，照相机 API 向主机应用软件发出消息表明照相机是有效的。在模块 512 中主机应用软件要求打开照相机驱动程序。照相机 API 做出响应打开照相机驱动程序，如模块 514 中所示。打开驱动程序的过程意味着建立了照相机 API 和照相机驱动程序之间的联系。如前面所述，此时开始固件的更新过程。

在模块 516 中，主机应用软件要求从照相机传送图像到主机系统。在模块 518 中，照相机 API 做出响应将图像信息从照相机传递到主机系统。图像信息可能包含图像像素数据及其它的信息，如调色板信息、压缩信息、图像定位信息等等，流程图在模块 520 结束。

图 11 所示的是检测是否有成像装置如照相机连接到主机系统的过程的一个实施方案流程图。流程图起始于模块 600。接下来进入模块 602，在该模块中操作系统测定照相机是否有效。这个操作可以借

助于端口驱动程序如 USB 驱动程序来完成。如果照相机是无效的，程序返回到模块 602。如果在模块 604 中照相机是有效的，程序继续进行到模块 606，在该模块中操作系统装入照相机驱动程序。

在一个实施方案中，主机系统采用一个操作系统如 Windows 98。Windows 98 允许驱动程序发出信号通知操作系统照相机是连接到主机系统的（连接事件），如模块 608 所示。然后操作系统打开通过连接事件注册的应用软件。这时，启动用于照相机的主机应用软件，如模块 610 中所示。然后流程图继续到图 10 的流程图。

如果操作系统没有提供基于连接事件的应用软件的打开途径，一个替代的实施方案可以用一个“服务程序”代替模块 608 和 610 所示的步骤。服务程序由用户安装，并且在主机系统开机时自动地在主机系统上启动起来。当检测到照相机时，服务程序打开主机应用软件。在一个实施方案中，服务程序应用照相机 API 来测定照相机是否有效。这样，服务程序如同一个小型的主机应用软件，按照与图 10 中所示的相类似的方式运行。但当与照相机驱动程序建立起联系时，服务程序会启动主机应用软件。而后，主机应用软件建立自身与照相机驱动程序的联系以便从照相机中传送图像。

在一个实施方案中，主机应用软件 60 和照相机驱动程序 44 连同照相机 10 一同发货。主机驱动程序 60 和照相机驱动程序 44 可以通过软盘或 CD-ROM 发送。另外，主机应用软件 60 和照相机驱动程序 44 还可以通过互联网下载。主机应用软件 60 和照相机驱动程序 44 被安装到主机系统的存储媒体中，如硬盘、动态随机存储器（DRAM）、静态随机存储器（SRAM）或快速存储器中。

在上面的详细说明中，参考特定的典型实施方案，对发明进行了描述。然而，很显然有些人可以在不背离本公开发明实质和超出随后权利要求所声明范围的前提下，对本发明作出改进和变动，而从本发明获利。因此，说明和图示都只能看作是解释性的而绝非限制。

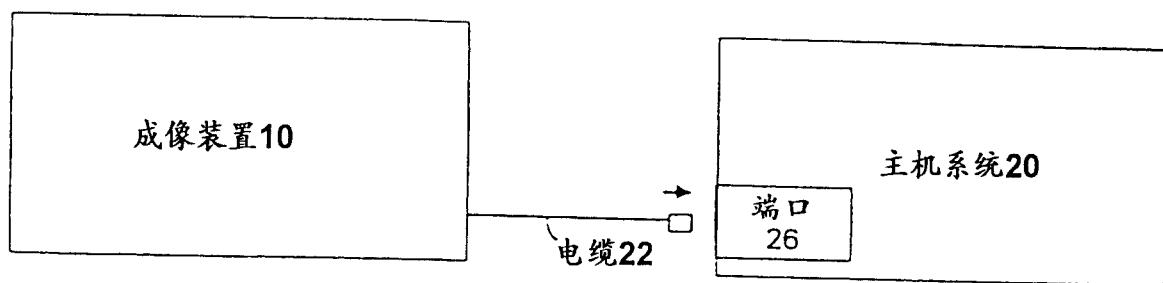


图 1

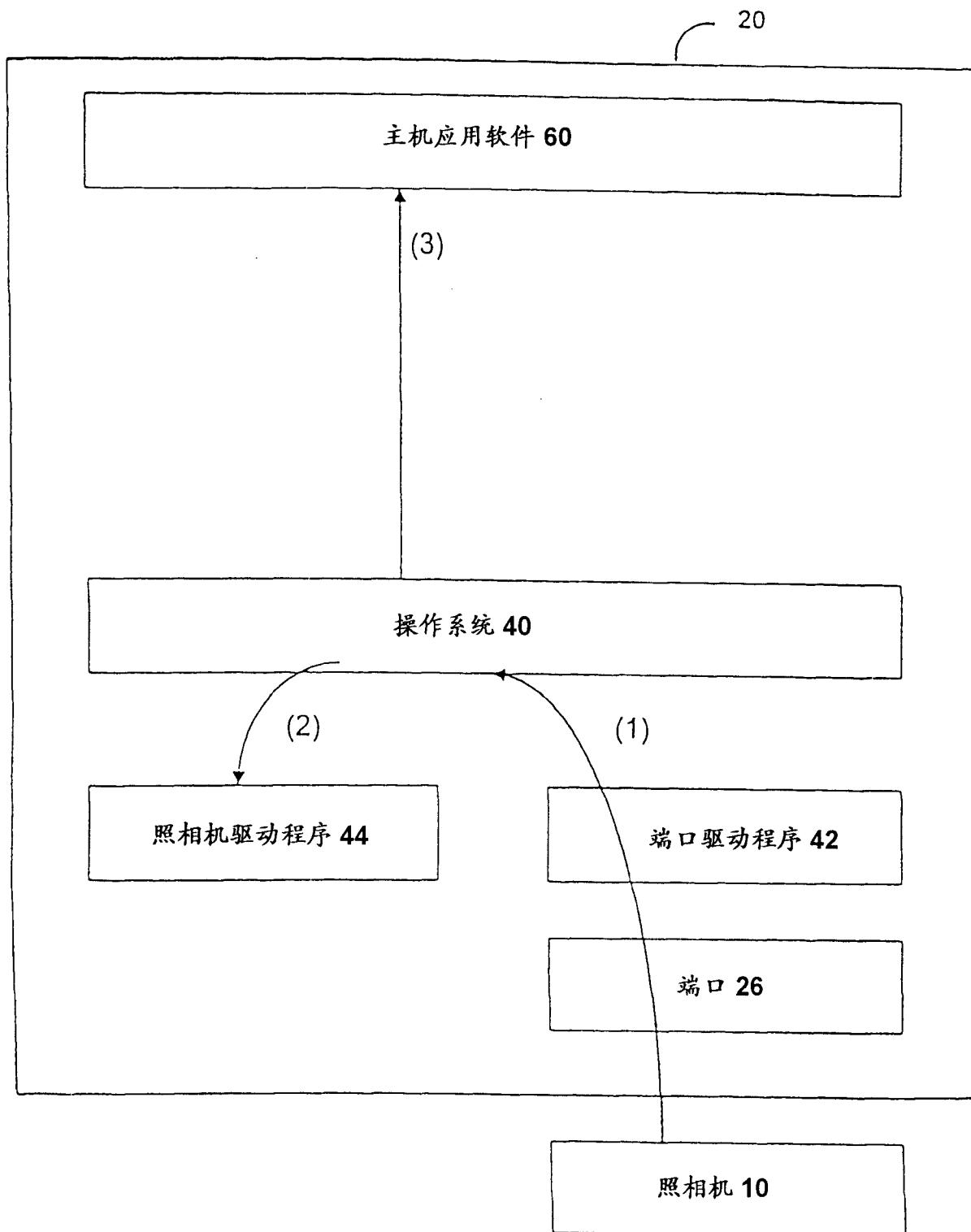


图 2

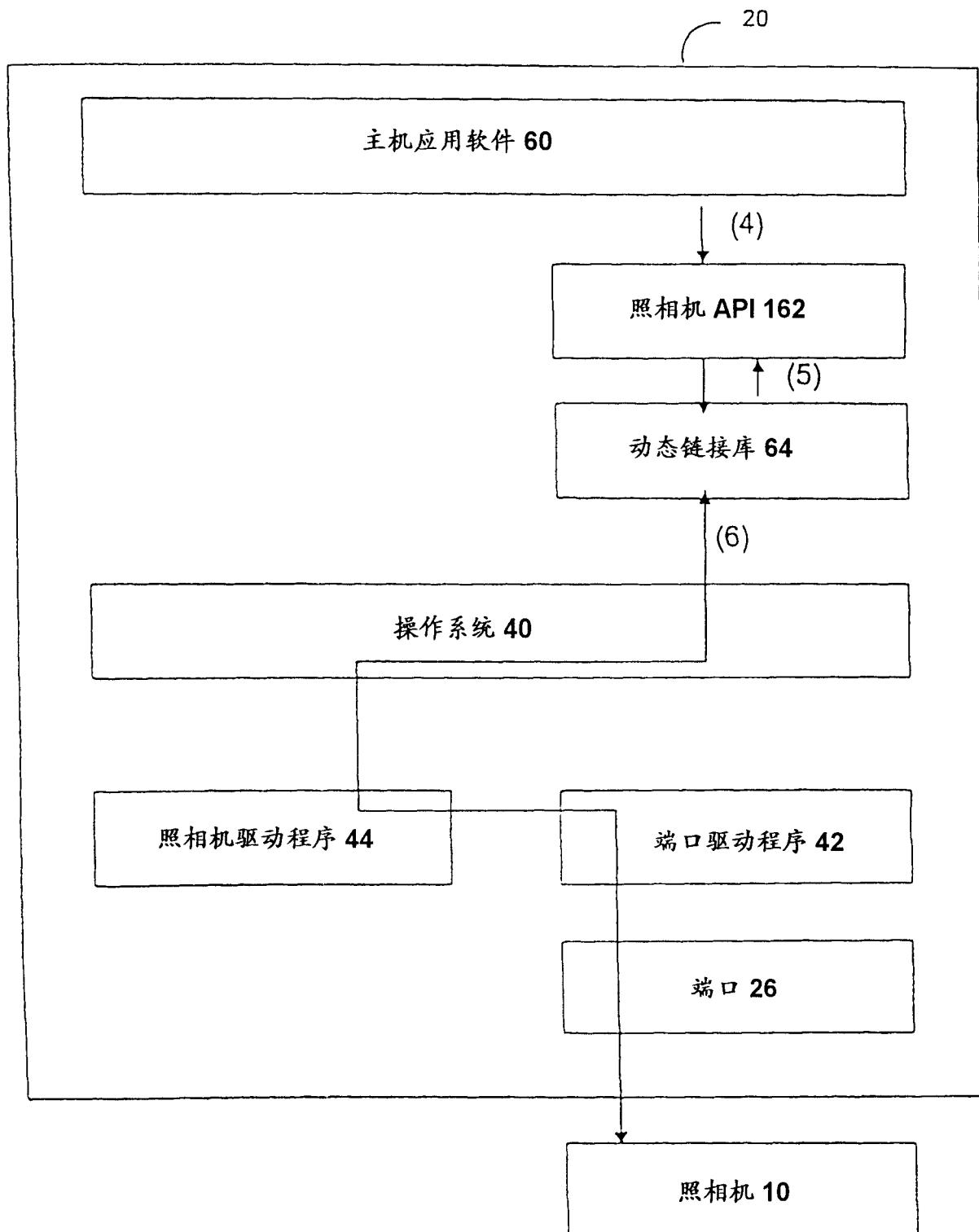


图 3

操作系统	主机应用软件	照相机 API
打开主机应用软件 →	创建和初始照相机 API → → → → 添加主窗口到照相机 API 的回调列表中	复位内部变量 装入 DLLs 创建和开始后台线程 插入 CM_SIGNAL_STATUS 信息到后台线程队列中

图 4

照相机 API	主机应用软件
CM_SIGNAL_STATUS 如果照相机没有开 如果 OpenDriver() 成功 CloseDriver() 消息未被发送 向所有回调窗口发送	→ WM_CAMERA_API_STATUS 如果信息是 CM_IS_CONNECTED OpenDriver()
CM_OPEN_DRIVER 打开照相机驱动程序 检测照相机兼容性	← 获取图像数目
CM_GET_NO_OF_IMAGES 返回照相机上图像的数量	← 如果图像数目 > 0 获取 图像列表
CM_GET_IMAGE_LIST 返回图像名和大小的列表	← 依次下载图像
CM_GET_IMAGE_BY_NAME 返回图像给定名	

图 5

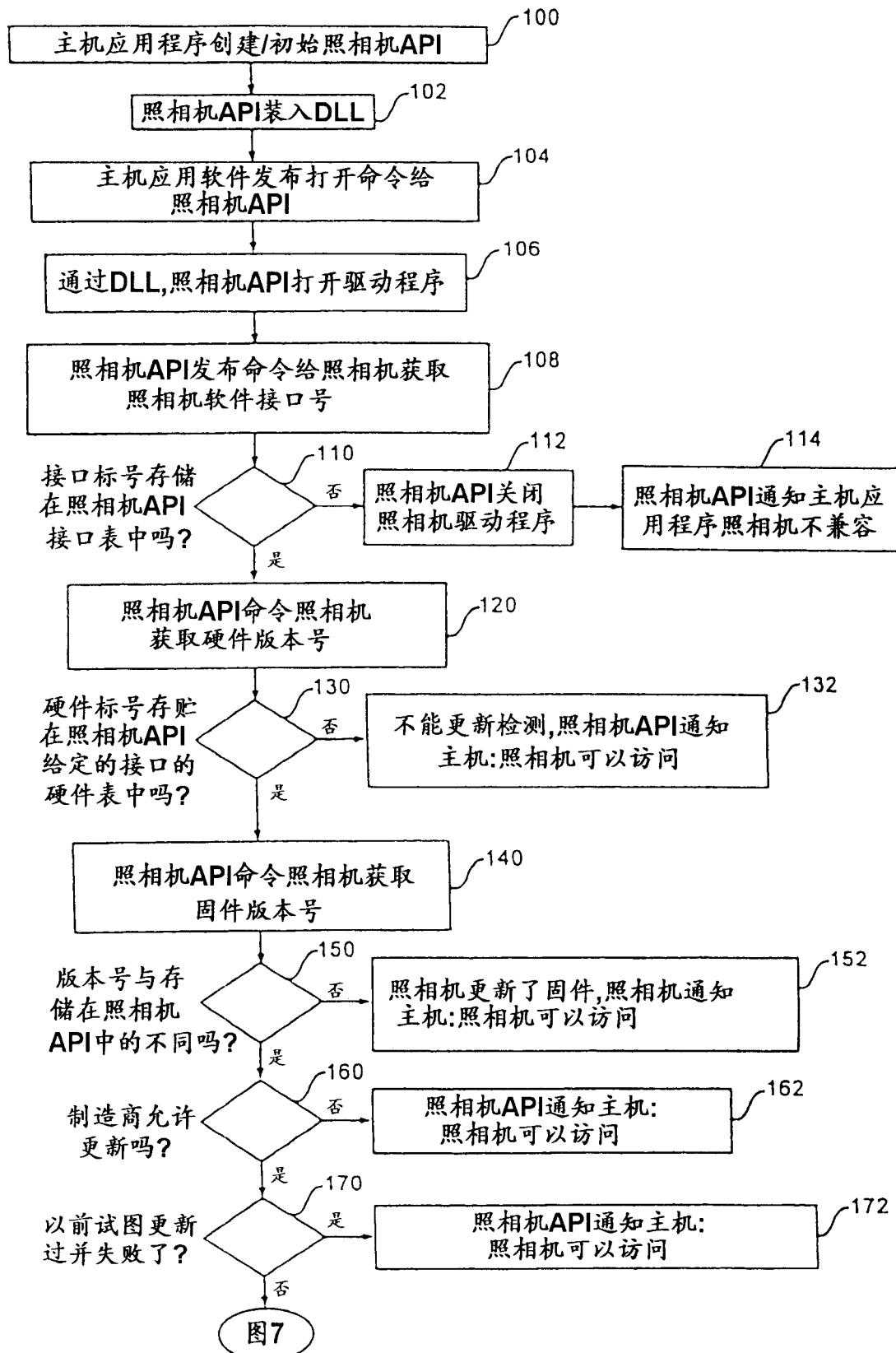


图 6

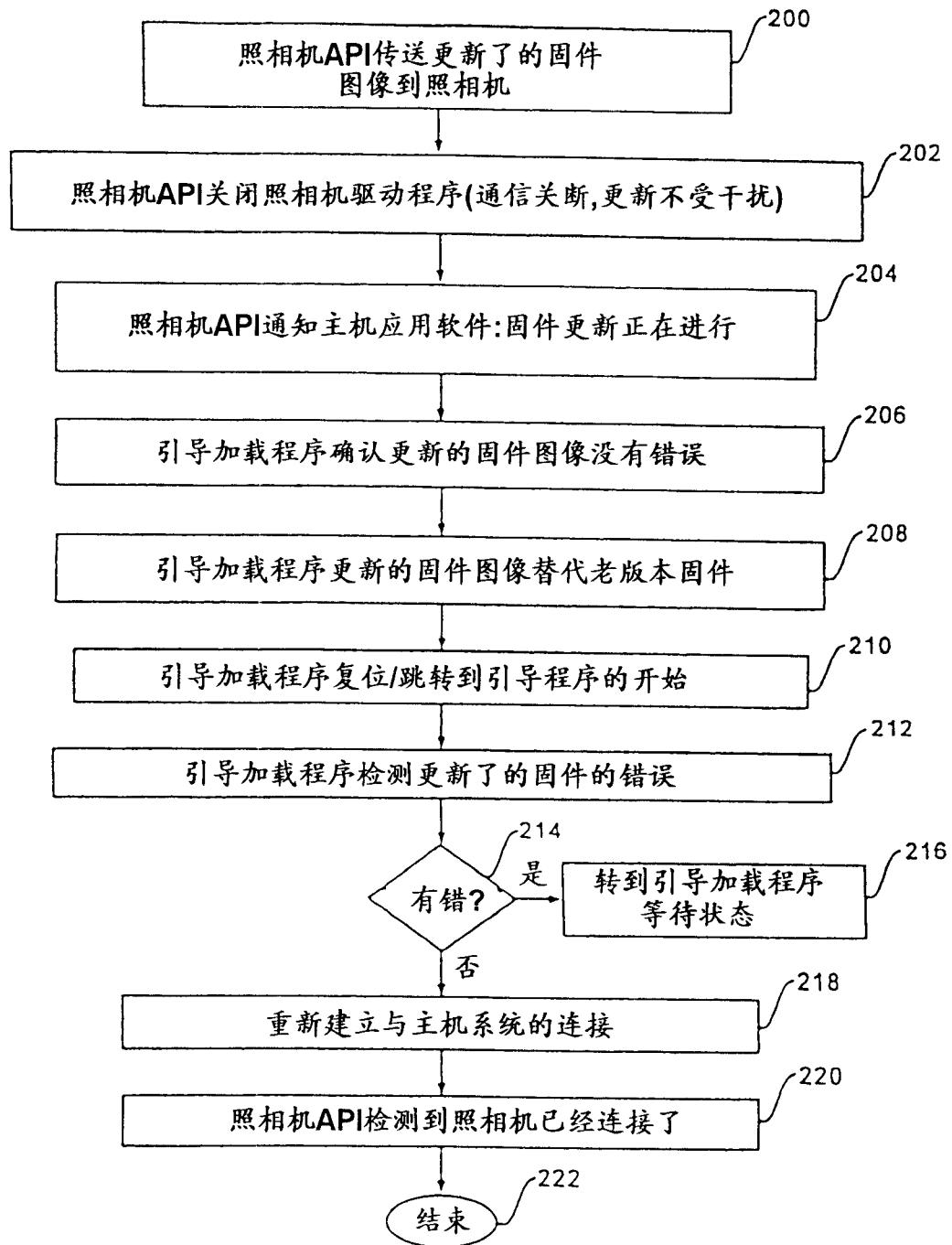


图 7

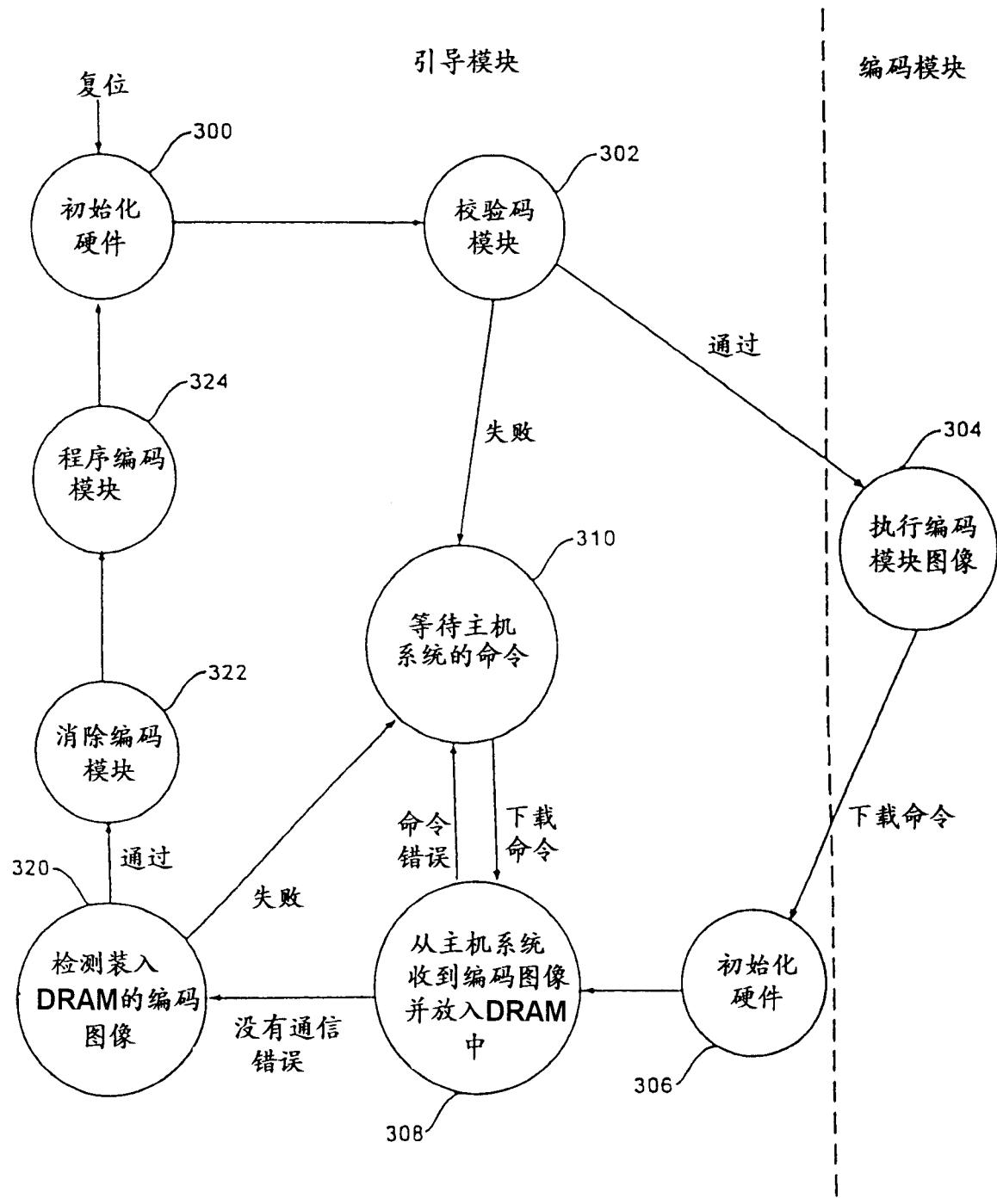


图 8

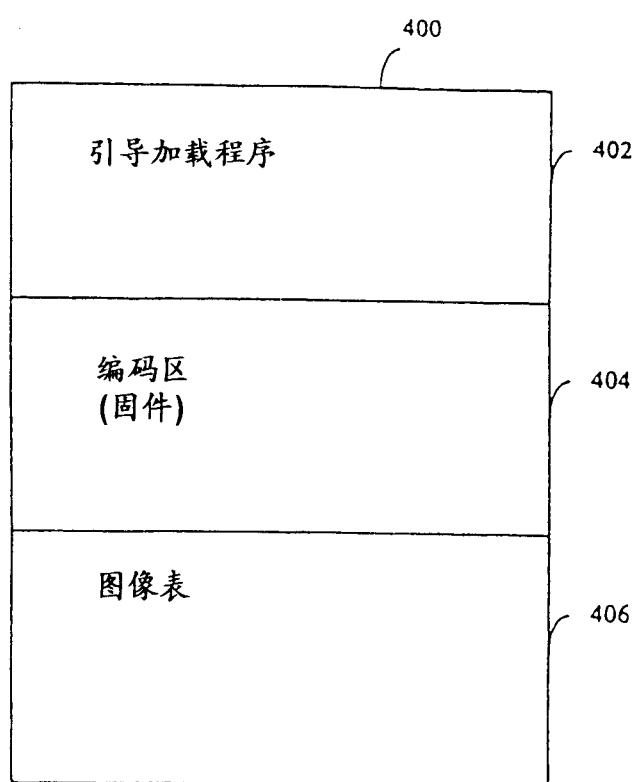


图 9

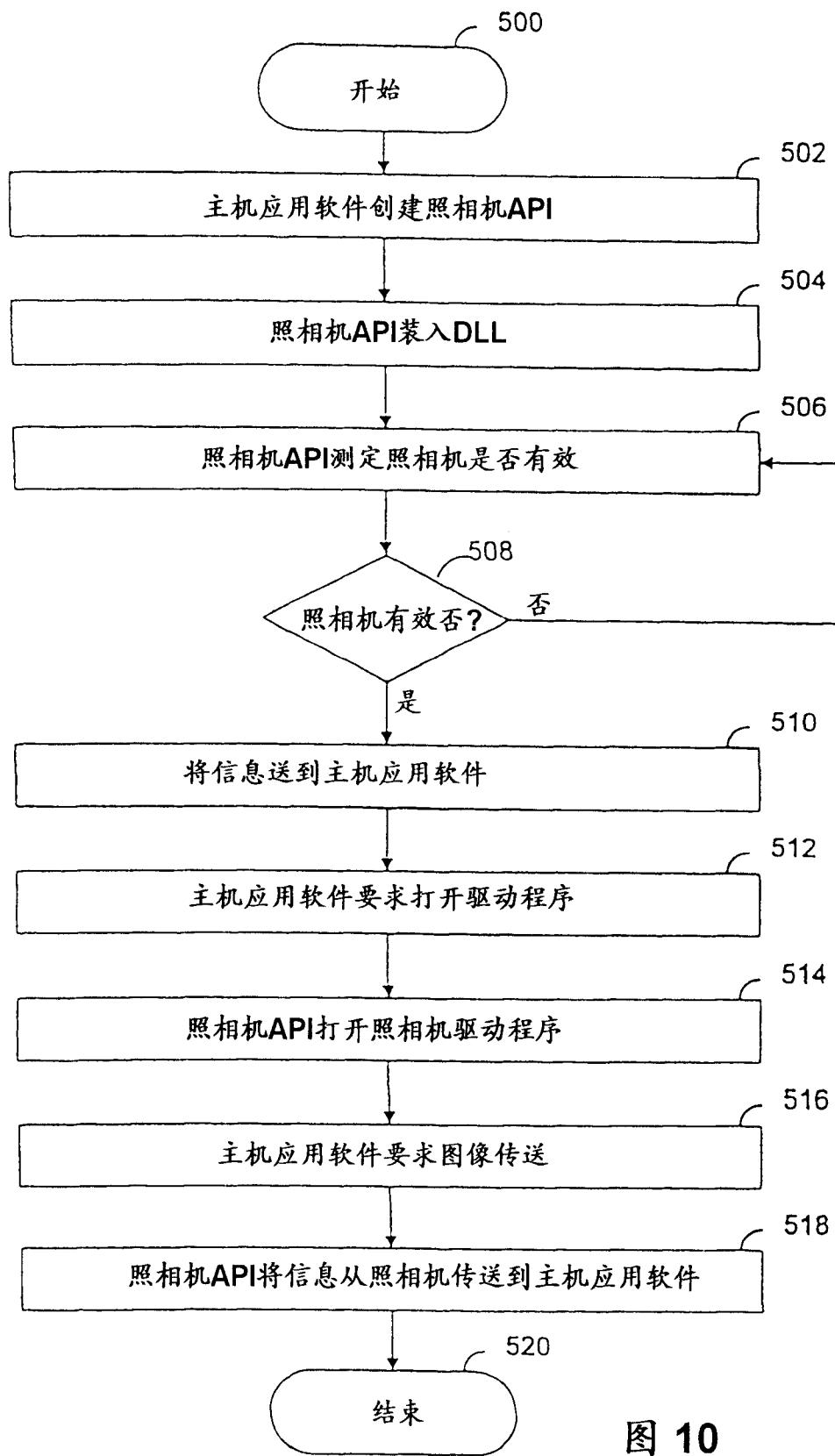


图 10

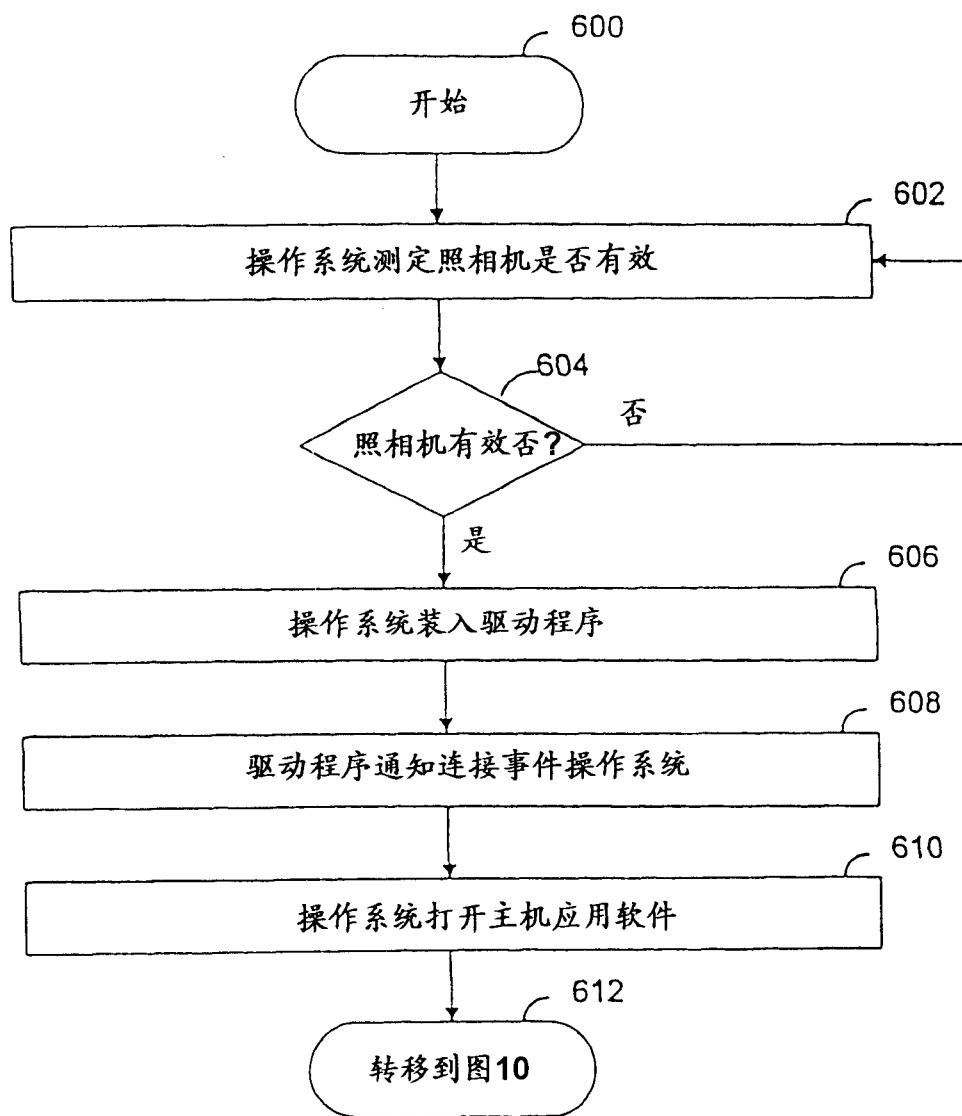


图 11