



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 198 82 944 B4** 2007.06.14

(12)

Patentschrift

(21) Deutsches Aktenzeichen: **198 82 944.2**
(86) PCT-Aktenzeichen: **PCT/US98/26970**
(87) PCT-Veröffentlichungs-Nr.: **WO 1999/035826**
(86) PCT-Anmeldetag: **17.12.1998**
(87) PCT-Veröffentlichungstag: **15.07.1999**
(43) Veröffentlichungstag der PCT Anmeldung
in deutscher Übersetzung: **04.01.2001**
(45) Veröffentlichungstag
der Patenterteilung: **14.06.2007**

(51) Int Cl.⁸: **H04N 5/232 (2006.01)**
H04N 7/00 (2006.01)
H04N 7/14 (2006.01)

Innerhalb von drei Monaten nach Veröffentlichung der Patenterteilung kann nach § 59 Patentgesetz gegen das Patent Einspruch erhoben werden. Der Einspruch ist schriftlich zu erklären und zu begründen. Innerhalb der Einspruchsfrist ist eine Einspruchsgebühr in Höhe von 200 Euro zu entrichten (§ 6 Patentkostengesetz in Verbindung mit der Anlage zu § 2 Abs. 2 Patentkostengesetz).

(30) Unionspriorität:
09/003,732 07.01.1998 US

(73) Patentinhaber:
Intel Corporation, Santa Clara, Calif., US

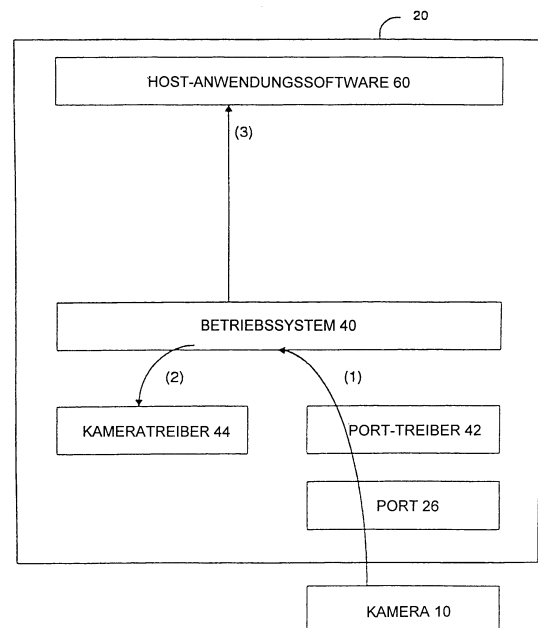
(74) Vertreter:
Zenz, Helber, Hosbach & Partner GbR, 45128 Essen

(72) Erfinder:
Fichtner, Mark R., Chandler, Ariz., US

(56) Für die Beurteilung der Patentfähigkeit in Betracht
gezogene Druckschriften:
US 56 75 358 A
US 53 86 567 A
US 52 20 654 A
WO 97/24 880 A1

(54) Bezeichnung: **Automatische Übertragung von Bildinformationen zwischen Bildgebungseinrichtung und Host-System**

(57) Hauptanspruch: Verfahren zum Übertragen von Bildinformationen zwischen einer Bildgebungseinrichtung (10) und einem Host-System (20), wobei von einer auf dem Host-System (20) ausführenden Anwendungssoftware (60) eine Bildgebungseinrichtungs-Anwendungsprogrammierschnittstelle (API) (62) erzeugt und initialisiert (102) wird; von der API (62) periodisch versucht wird (106, 108), einen auf dem Host-System (20) vorhandenen Bildgebungseinrichtungstreiber (44) zu öffnen, wobei der Bildgebungseinrichtungstreiber (44) geöffnet wird, sofern die Bildgebungseinrichtung (10) mit dem Host-System (20) verbunden ist; und nachdem die API (62) den Bildgebungseinrichtungstreiber (44) erfolgreich geöffnet hat, Bildinformationen aus der Bildgebungseinrichtung an das Host-System übertragen werden (118).



Beschreibung

[0001] Die vorliegende Erfindung bezieht sich auf ein Verfahren zum Übertragen von Bildinformationen zwischen einer Bildgebungseinrichtung und einem Host-System.

[0002] Bildgebungseinrichtungen, wie beispielsweise Kameras, speichern üblicherweise Informationen sich bewegender (Video-) oder Standbilder auf Film, Videoband oder anderen Medien. Digitale Kameras nehmen Bildinformationen in einem digitalen Format auf und speichern die Bildinformationen in einem Speicher, wie beispielsweise einem Flash-Speicher, oder auf anderen digitalen Speichermedien. Die digitalen Bildinformationen können zu einem Host-System, wie beispielsweise einem Personalcomputer, heruntergeladen werden. Die Bildinformationen können dann bearbeitet werden, indem mit Hilfe von sich auf dem Host-System aufhaltenden Software-Anwendungen das Bild rotiert, das Bild beschnitten oder auf andere Weise geändert wird.

[0003] Um ein Bild auf einem Host-System zu verarbeiten, befestigt ein Benutzer eine Bildgebungseinrichtung an dem Host-System, initiiert eine Anwendungssoftware zur Schnittstellenbildung zu der Bildgebungseinrichtung und überträgt Bildinformationen zwischen der Bildgebungseinrichtung und dem Host-System. Jede dieser Aufgaben kann mehrere Schritte beanspruchen und kann für eine Bilder aufnehmende Person, die auf dem Gebiet des Computers ein Anfänger ist, abschreckend sein.

[0004] Aus dem US-Patent 5,675,358 ist eine Einrichtung und ein Verfahren zum Ankoppeln einer Digitalkamera an einen Computer bekannt, bei dem ein Kameratreiber in dem Computer stets unabhängig davon geladen ist, ob die Kamera in den Computer eingesteckt ist oder nicht. Das US-Patent 5,386,567 beschreibt die Selbstkonfiguration von Computersystemen beim dynamischen Anstecken und Abtrennen von Geräten. Die Veröffentlichung WO 97/24880 A1 beschreibt u.a. eine Videokamera, die zur Bildübertragung über eine USB-Schnittstelle mit einem Computer verbunden ist. Das US-Patent 5,220,654 beschreibt ein dynamisches Konfigurieren von I/O-Komponenten eines Computers.

[0005] Die der Erfindung zugrundeliegende Aufgabe besteht darin, die Übertragung von Bildinformationen zwischen einer Bildgebungseinrichtung (Kamera) und einem Host-System für einen Benutzer zu vereinfachen.

[0006] Diese Aufgabe wird erfindungsgemäß durch ein Verfahren mit den Merkmalen des Anspruchs 1 gelöst. Vorteilhafte und/oder bevorzugte Weiterbildungen der Erfindung sind in den Unteransprüchen gekennzeichnet.

[0007] Nachfolgend wird die Erfindung anhand von in der Zeichnung dargestellten bevorzugten Ausführungsbeispielen näher beschrieben.

[0008] Es wird ein Verfahren zum Übertragen von Bildinformationen zwischen einer Bildgebungseinrichtung und einem Host-System offenbart. Das Host-System erfaßt, daß eine Bildgebungseinrichtung mit dem Host-System verbunden ist. In Erwiderung des Erfassens der Bildgebungseinrichtung werden ein Bild oder mehrere Bilder zwischen der Bildgebungseinrichtung und dem Host-System übertragen.

[0009] Andere Merkmale und Vorteile der vorliegenden Erfindung werden aus den begleitenden Zeichnungen und aus der folgenden detaillierten Beschreibung ersichtlich.

[0010] [Fig. 1](#) zeigt eine Darstellung einer Bildgebungseinrichtung, die mit einem Host-System verbindbar ist.

[0011] [Fig. 2](#) zeigt ein Ausführungsbeispiel des Informationsflusses zwischen den Komponenten des Host-Systems, wenn die Bildgebungseinrichtung **10** erstmals mit dem Host-System **20** verbunden wird.

[0012] [Fig. 3](#) zeigt ein Ausführungsbeispiel des Informationsflusses zwischen den Komponenten des Host-Systems, nachdem die Host-Anwendungssoftware **60** initiiert worden ist.

[0013] [Fig. 4](#) zeigt ein Ausführungsbeispiel des abfragenden Initialisierungsprozesses.

[0014] [Fig. 5](#) zeigt ein Ausführungsbeispiel des Abfrageprozesses zwischen der Kamera-API **62** und der Host-Anwendungssoftware **60**.

[0015] [Fig. 6](#) zeigt ein Ablaufdiagramm eines Ausführungsbeispiels des Prozesses der Übertragung von Bildern zwischen der Bildgebungseinrichtung und dem Host-System.

[0016] [Fig. 7](#) zeigt ein Ablaufdiagramm eines Ausführungsbeispiels des Prozesses des Erfassens, das eine Bildgebungseinrichtung, wie beispielsweise eine Kamera, mit einem Host-System verbunden ist.

[0017] Ein Verfahren zum Übertragen von Bildinformationen zwischen einer Bildgebungseinrichtung und einem Host-System wird offenbart. Bei einem Ausführungsbeispiel wird die Übertragung von Bildinformationen automatisch bei Verbinden der Bildgebungseinrichtung mit dem Host-System durchgeführt.

[0018] Die Bildgebungseinrichtung kann eine Bildaufnahmeeinrichtung, wie beispielsweise eine Kame-

ra, sein. Alternativ können die offenbarten Techniken mit einer beliebigen Einrichtung verwendet werden, die in der Lage ist, Bildinformationen zu speichern. Das Host-System kann ein beliebiges System sein, welches in der Lage ist, Bildinformationen zu bearbeiten. Beispielsweise kann das Host-System ein Personalcomputer, wie beispielsweise ein IBM-kompatibler Personalcomputer sein, auf dem auf einem Intel-Pentium®- oder einem Pentium®II-Prozessor abgearbeitet wird. Jedoch könnte das Host-System alternativ ein Drucker, ein Plotter, ein Faxgerät, eine Anzeigeeinrichtung oder eine Speichereinrichtung sein.

[0019] [Fig. 1](#) zeigt eine Darstellung einer Bildgebungseinrichtung **10**, die mit einem Host-System **20** verbindbar ist. Bei einem Ausführungsbeispiel ist die Bildgebungseinrichtung **10** über ein Kabel **22** mit einem Port **26** des Host-Systems **20** verbunden. Die Bildgebungseinrichtung **10** wird mit dem Host-System **20** vorzugsweise über die Verwendung eines Datenübertragungsprotokolls gekoppelt, das eine hohe Datenübertragungsrate unterstützt. Bei einem Ausführungsbeispiel wird die Bildgebungseinrichtung **10** mit dem Host-System **20** über eine Verbindung eines universellen seriellen Busses (USB) gekoppelt. Die USB-Verbindung ermöglicht eine Datenübertragungsrate von bis zu 12 Mb/s. Andere Verbindungen und Datenübertragungsprotokolle können alternativ verwendet werden, wie beispielsweise das 1394-Protokoll (weitere Informationen über USB können aus dem Internet bei der URL <http://www.usb.org/> gewonnen werden. Der 1394-Standard wird von dem Institute of Electrical and Electronic Engineers gehalten und verteilt. Firewire, eine Implementierung von 1394, ist durch den IEEE-Standard 1394-1995 definiert.)

[0020] [Fig. 2](#) und [Fig. 3](#) sind Ausführungsbeispiele, die die Beziehungen und die Nachrichtenübertragungen zwischen den Komponenten des Host-Systems **20** und der Bildgebungseinrichtung (Kamera) **10** zeigen. [Fig. 2](#) zeigt ein Ausführungsbeispiel des Informationsflusses zwischen den Komponenten des Host-Systems, wenn die Bildgebungseinrichtung **10** erstmalig mit dem Host-System **20** verbunden wird. Das Host-System **20** enthält ein Betriebssystem (O/S) **40** und Host-Anwendungssoftware **60**. Das Host-System **20** erfaßt, wenn eine Bildgebungseinrichtung, wie beispielsweise eine Kamera **10**, mit dem Host-System **20** verbunden wird. Bei einem Ausführungsbeispiel erfaßt das Betriebssystem **40**, ob eine Kamera **10** mit dem System verbunden wird, indem sie den Port **26** abfragt. Ein Port-Treiber **42** kann verwendet werden, um eine Schnittstelle zwischen dem Betriebssystem **40** und dem Port **26** zur Verfügung zu stellen. Bei einem Ausführungsbeispiel ist der Port ein USB-Port und der Port-Treiber ist ein USB-Treiber.

[0021] Das Betriebssystem kann eines einer Vielzahl unterschiedlicher Betriebssysteme sein. Bei einem Ausführungsbeispiel ist das Betriebssystem ein Windows(Marken und Warenzeichen Dritter sind Eigentum der jeweiligen Inhaber.)-Betriebssystem, wie beispielsweise Windows(Marken und Warenzeichen Dritter sind Eigentum der jeweiligen Inhaber.) 95 oder Windows(Marken und Warenzeichen Dritter sind Eigentum der jeweiligen Inhaber.) 98 der Microsoft Corporation. Windows 98 enthält Hooks, welche die Abfrage von Ports ermöglichen. Andere Betriebssysteme können so modifiziert werden, daß sie ein derartiges Abfragen ermöglichen. Das Abfragen wird vorzugsweise im Hintergrund durchgeführt, so daß der Benutzer nicht wahrnimmt, daß es durchgeführt wird. Alternativ kann die Host-Anwendungssoftware **60** das Abfragen des Ports **26** durchführen. Jedoch hat das Abfragen durch das Betriebssystem **40** (anstelle der Host-Anwendungssoftware **60**) Leistungsvorteile, da das Betriebssystem bereits für das Abfragen verschiedener Aktivitäten, wie beispielsweise Tastaturbetätigungen, Mausbewegungen usw., eingerichtet ist. Für die Zwecke der Veranschaulichung nimmt die folgende Beschreibung an, daß das Betriebssystem das Abfragen ausführt. Ein Fachmann kann die Modifikationen durchführen, die erforderlich sind, um es einer Anwendung zu ermöglichen, das Abfragen auszuführen.

[0022] Wenn eine Kamera **10** mit dem Port **26** des Host-Systems **20** verbunden wird, signalisiert der Port-Treiber **42** dem Betriebssystem **40**, daß die Kamera mit dem Host-System **20** verbunden worden ist. Dies ist durch den mit (1) markierten Pfeil, der in [Fig. 1](#) gezeigt ist, veranschaulicht. Das Betriebssystem **40** identifiziert die Einrichtung als eine Kamera und lädt den entsprechenden Softwaretreiber **44** in den Speicher, wie es durch den Pfeil (2) veranschaulicht ist. Bei einem Ausführungsbeispiel fragt das Betriebssystem **40** die Kamera **10** ab, um einen Identifizierer zu erlangen. Das Betriebssystem **40** lädt den Softwaretreiber **44**, der dem Identifizierer entspricht. Bei diesem Beispiel wird ein Kameratreiber **44** durch das Betriebssystem **40** geladen.

[0023] Dann lädt das Betriebssystem **40** eine oder mehrere Softwareanwendungen, die der Kamera entsprechen. Bei einem Ausführungsbeispiel ermöglicht das Betriebssystem, daß die Softwareanwendungen registriert werden. Bei Erfüllung einer vorgegebenen Bedingung (beispielsweise dann, wenn eine Kamera mit einem bestimmten Identifizierer erfaßt worden ist), wird die registrierte Host-Softwareanwendung geladen. In diesem Fall wird die Host-Anwendungssoftware **60** (für die Kamera) geladen, wie es durch den Pfeil (3) gezeigt ist. Bei einem Ausführungsbeispiel signalisiert der Kameratreiber **44** dem Betriebssystem, daß dieses die Host-Anwendungssoftware **60** initiieren soll. Die Host-Anwendungssoftware **60** initiiert die Übertragung von Bildinformatio-

nen zwischen der Bildgebungseinrichtung (Kamera) **10** und dem Host-System **20**. Die Host-Anwendungssoftware **60** kann darüber hinaus Bilder verarbeiten. Beispielsweise kann die Host-Anwendungssoftware **60** eine Dekompression und/oder Farbkorrektur an den Bildern durchführen. Darüber hinaus kann die Host-Anwendungssoftware **60** eine Rotation, ein Beschneiden und andere Bildverarbeitungsfunktionen durchführen.

[0024] Einige Betriebssysteme, wie beispielsweise Windows 98, ermöglichen spezielle Ereignisse, um zu bewirken, daß Software-Anwendungen gestartet werden. Beispielsweise kann der Kameratreiber **44** mit registrierten Ereignissen eingerichtet werden, wie beispielsweise "Verbindung mit Kamera erfaßt" oder "Blendenknopf an der Kamera betätigt". So kann ein Betriebssystem so eingerichtet werden, daß es automatisch eine Anwendung, beispielsweise die Host-Anwendungssoftware **60**, startet, wenn die Kamera **10** befestigt wird.

[0025] Bei einem Ausführungsbeispiel wird dann, wenn der Kameratreiber **44** oder die Host-Anwendungssoftware **60** nicht auf dem Host-System **20** installiert ist, wenn die Kamera **10** mit dem Host-System **20** verbunden wird, der Benutzer aufgefordert, den Kameratreiber **44** und/oder die Host-Anwendungssoftware **60** für die Einrichtung, die mit dem Port **26** verbunden worden ist, bereitzustellen. Sobald die Installation abgeschlossen ist, fährt der Prozeß wie zuvor beschrieben fort.

[0026] [Fig. 3](#) zeigt ein Ausführungsbeispiel des Informationsflusses zwischen den Komponenten des Host-Systems, nachdem die Host-Anwendungssoftware **60** initiiert worden ist. Bei diesem Ausführungsbeispiel erzeugt die Host-Anwendungssoftware **60**, nachdem sie geladen worden ist, eine Kamera-Anwendungsprogrammierschnittstelle (API) **62** und initialisiert diese, wie es durch den Pfeil (4) angezeigt ist. Die Kamera-API **62** kann ihre Aufgabe in einem Hintergrund-Thread durchführen. Auf diese Weise braucht die Host-Anwendungssoftware **60** nicht darauf zu warten, daß die Kamera-API **62** abschließt, bevor sie andere Aufgaben durchführt. Bei einem Ausführungsbeispiel ist die Kamera-API **62** ein COM-Objekt, welches eine O/S-abhängige dynamische Verknüpfungsbibliothek (DLL) **64** lädt, wie es durch die Pfeile (5) angezeigt ist. Die Kamera-API **62** kommuniziert mit dem Betriebssystem **40** über die DLL **64**. (Bei einem anderen Ausführungsbeispiel enthält die Kamera-API **62** die DLL **64**.) Das Betriebssystem **40** wiederum kommuniziert mit der Kamera **10** über den Kameratreiber **44** und den Port-Treiber **42**, wie es durch die Pfeile (6) gezeigt ist.

[0027] [Fig. 4](#) zeigt ein Ausführungsbeispiel des Abfrageinitialisierungsprozesses. Der Abfrageinitialisierungsprozeß beginnt damit, daß das Betriebssystem

die Host-Anwendungssoftware öffnet. Dann erzeugt und initialisiert die Host-Anwendungssoftware **60** eine Kamera-API **62**. Bei einem Ausführungsbeispiel fügt die Host-Anwendungssoftware **60** sich selbst zu der Rückrufliste (callback list) der Kamera-API hinzu, so daß die Host-Anwendungssoftware **60** benachrichtigt wird, wenn die Kamera API bei dem Abfrageprozeß erfolgreich ist.

[0028] Bei einem Ausführungsbeispiel setzt die Kamera-API bei der Initialisierung ihre internen Variablen zurück, lädt eine O/S-abhängige DLL und erzeugt und startet einen Hintergrund-Thread. Dann fügt die Kamera-API eine Nachricht in die im Hintergrund eingefädelte (threaded) Warteschlange ein, die versucht, den Kameratreiber zu öffnen. (Ein Treiber wird "geöffnet", indem eine Verbindung zu dem Treiber hergestellt wird.) Bei einem Ausführungsbeispiel wird der Kameratreiber nur dann geöffnet, wenn eine Kamera angekoppelt ist: Wenn der Kameratreiber nicht geöffnet werden kann, dann ist keine Kamera mit dem Host-System gekoppelt. Wenn ein Kameratreiber geöffnet werden kann, dann ist eine Kamera angekoppelt. Bei einem Ausführungsbeispiel versucht die Kamera-API **44** den Kameratreiber jede halbe Sekunde zu öffnen.

[0029] [Fig. 5](#) zeigt ein Ausführungsbeispiel des Abfrageprozesses zwischen der Kamera-API **62** und der Host-Anwendungssoftware **60**. Bei diesem Ausführungsbeispiel versucht die Kamera-API **62**, den Kameratreiber zu öffnen. Wenn sie beim Öffnen des Kameratreibers erfolgreich ist, schließt die Kamera-API den Kameratreiber und benachrichtigt die Anwendungen in ihrer Rückrufwarteschlange (callback queue). Da die Host-Anwendungssoftware **60** sich in der Rückrufwarteschlange der Kamera-API befindet, wird sie benachrichtigt, daß eine Kamera erfaßt worden ist.

[0030] Bei diesem Ausführungsbeispiel öffnet die Host-Anwendungssoftware **60** den Kameratreiber **44** erneut, indem sie die Kamera-API **62** benachrichtigt, daß diese den Kameratreiber **44** öffnen und hinsichtlich einer kompatiblen Kamera überprüfen soll. Die Host-Anwendungssoftware **60** kann dann verschiedene Kommandos an die Kamera **10** über die Kamera-API **44** (und das Betriebssystem **40** und die Treiber **44** und **42**) senden. Beispielsweise kann die Host-Anwendungssoftware **60** die Anzahl der Kamera gespeicherten Bilder anfordern. Die Host-Anwendungssoftware **60** kann eine Liste der Namen der Bilder und der Bildgrößen anfordern oder sie kann ein bestimmtes Bild anfordern.

[0031] [Fig. 6](#) zeigt ein Ablaufdiagramm eines Ausführungsbeispiels des Prozesses der Übertragung von Bildern zwischen der Bildgebungseinrichtung und dem Host-System. In diesem Fall ist die Bildgebungseinrichtung eine Kamera und das Host-System

ist ein Personalcomputer. Das Ablaufdiagramm beginnt beim Block **100**. Der Prozeß fährt zum Block **102** fort, bei welchem die Host-Anwendungssoftware eine Kamera-API erzeugt. Die Kamera-API lädt beim Block **104** eine DLL, die betriebssystemabhängig ist. Beim Block **106** bestimmt die Kamera-API, ob eine Kamera verfügbar ist.

[0032] Beim Block **108** kehrt der Ablauf dann, wenn keine Kamera verfügbar ist, zum Block **106** zurück. Sofern jedoch eine Kamera verfügbar ist, wird der Prozeß beim Block **110** fortgesetzt. Beim Block **110** sendet die Kamera-API eine Nachricht an die Host-Anwendungssoftware, die anzeigt, daß eine Kamera verfügbar ist. Die Host-Anwendungssoftware fordert beim Block **112** an, daß der Kameratreiber geöffnet wird. Die Kamera-API antwortet durch Öffnen des Kameratreibers, wie es beim Block **114** gezeigt ist. Der Prozeß des Öffnens eines Treibers bedeutet das Einrichten einer Verbindung mit dem Kameratreiber. Beim Block **116** fordert die Host-Anwendungssoftware an, daß Bilder von der Kamera zu dem Host-System übertragen werden. Die Kamera-API antwortet, indem sie Bildinformationen aus der Kamera an die Host-Anwendungssoftware am Block **118** überträgt. Bildinformationen können Bildpixeldaten ebenso wie andere Informationen, wie beispielsweise Farbpaletteninformationen, Kompressionsinformationen, Orientierungen des Bildes und dergleichen umfassen. Das Ablaufdiagramm endet beim Block **120**.

[0033] [Fig. 7](#) zeigt ein Ablaufdiagramm eines Ausführungsbeispiels des Prozesses des Erfassens, daß eine Bildgebungseinrichtung, wie beispielsweise eine Kamera, mit einem Host-System gekoppelt ist. Das Ablaufdiagramm beginnt im Block **200**. Es fährt am Block **202** fort, bei welchem das Betriebssystem feststellt, ob eine Kamera verfügbar ist. Dies kann mit Hilfe eines Port-Treibers, wie beispielsweise eines USB-Treibers, ausgeführt werden. Sofern die Kamera nicht verfügbar ist, kehrt der Prozeß zum Block **202** zurück. Sofern am Block **204** eine Kamera verfügbar ist, fährt der Prozeß zum Block **206** fort, bei welchem das Betriebssystem den Kameratreiber lädt.

[0034] Bei einem Ausführungsbeispiel wird ein Betriebssystem, wie beispielsweise Windows 98 verwendet. Windows 98 ermöglicht es dem Treiber, das Betriebssystem darüber zu benachrichtigen, daß die Kamera mit dem Host-System verbunden wird (das Verbindungsereignis), wie es durch Block **208** gezeigt ist. Dann öffnet das Betriebssystem die Anwendungen, die bei dem Verbindungsereignis registriert sind. In diesem Fall wird die Host-Anwendungssoftware für die Kamera initiiert, wie es beim Block **210** gezeigt ist. Das Ablaufdiagramm fährt dann zum Ablaufdiagramm gemäß [Fig. 6](#) fort.

[0035] Sofern das Betriebssystem keine Möglichkeit des Öffnens einer Anwendung auf der Grundlage eines Verbindungsereignisses zur Verfügung stellt, kann ein alternatives Ausführungsbeispiel einen "Dienst" anstelle der durch die Blöcke **208** und **210** gezeigten Schritte verwenden. Der Dienst wird durch den Benutzer installiert und automatisch dann auf dem Host-System initiiert, wenn das Host-System startet (boots up).

[0036] Der Dienst öffnet die Host-Anwendungssoftware, wenn eine Kamera erfaßt wird. Bei einem Ausführungsbeispiel verwendet der Dienst die Kamera-API um festzustellen, ob die Kamera verfügbar ist. Somit dient der Dienst als Mini-Host-Anwendung in einer Weise, die ähnlich der in [Fig. 6](#) gezeigten ist. Jedoch initiiert der Dienst die Host-Anwendungssoftware, wenn eine Verbindung zu dem Kameratreiber eingerichtet wird. Die Host-Anwendung richtet dann ihre eigene Verbindung zu dem Kameratreiber ein, um Bilder aus der Kamera zu übertragen.

[0037] Bei einem Ausführungsbeispiel werden die Host-Anwendungssoftware **60** und der Kameratreiber **44** zusammen mit der Kamera vertrieben. Die Host-Anwendungssoftware **60** und der Kameratreiber **44** können mit Hilfe einer Diskette oder CD-ROM versandt werden. Alternativ können die Host-Anwendungssoftware **60** und der Kameratreiber **44** über das Internet heruntergeladen werden. Die Host-Anwendungssoftware **60** und der Kameratreiber **44** werden auf ein Speichermedium auf dem Host-System installiert, wie beispielsweise auf einer Festplatte, einem dynamischen Speicher mit wahlfreiem Zugriff (DRAM), einem statischen Speicher mit wahlfreiem Zugriff (SRAM) oder einem Flash-Speicher.

[0038] In der obigen Beschreibung wurde die Erfindung unter Bezugnahme auf ihre speziellen Ausführungsbeispiele beschrieben. Für jemanden, der Nutzen aus dieser Offenbarung zieht, ist es jedoch klar, daß verschiedene Modifikationen und Änderungen daran vorgenommen werden können, ohne vom breiteren Geist und Umfang der Erfindung abzuweichen, wie er in den anhängigen Ansprüchen angegeben ist. Die Beschreibung und die Zeichnungen sind demzufolge nur in einem veranschaulichenden und nicht einem einschränkenden Sinne zu interpretieren.

Patentansprüche

1. Verfahren zum Übertragen von Bildinformationen zwischen einer Bildgebungseinrichtung (**10**) und einem Host-System (**20**), wobei von einer auf dem Host-System (**20**) ausführenden Anwendungssoftware (**60**) eine Bildgebungseinrichtungs-Anwendungsprogrammierschnittstelle (API) (**62**) erzeugt und initialisiert (**102**) wird; von der API (**62**) periodisch versucht wird (**106**, **108**), einen auf dem Host-System (**20**) vorhandenen Bild-

gebungseinrichtungstreiber (**44**) zu öffnen, wobei der Bildgebungseinrichtungstreiber (**44**) geöffnet wird, sofern die Bildgebungseinrichtung (**10**) mit dem Host-System (**20**) verbunden ist; und nachdem die API (**62**) den Bildgebungseinrichtungstreiber (**44**) erfolgreich geöffnet hat, Bildinformationen aus der Bildgebungseinrichtung an das Host-System übertragen werden (**118**).

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß die API, nachdem sie erfolgreich den Bildgebungseinrichtungstreiber geöffnet hat, die Anwendungssoftware (**60**) darüber benachrichtigt (**110**), daß die Bildgebungseinrichtung verfügbar ist.

3. Verfahren nach Anspruch 2, dadurch gekennzeichnet, daß die API, nachdem sie erfolgreich den Bildgebungseinrichtungstreiber geöffnet hat und bevor sie die Anwendungssoftware benachrichtigt, den Bildgebungseinrichtungstreiber schließt, wobei die Anwendungssoftware den Bildgebungseinrichtungstreiber nach Empfang der Benachrichtigung erneut öffnet.

4. Verfahren nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, daß eine Bildgebungseinrichtung erfaßt wird, sofern die Bildgebungseinrichtung mit einem Port (**26**) des Host-Systems verbunden ist.

5. Verfahren nach Anspruch 4, dadurch gekennzeichnet, daß ein Port-Treiber (**42**) über die Ankopplung der Bildgebungseinrichtung benachrichtigt.

Es folgen 6 Blatt Zeichnungen

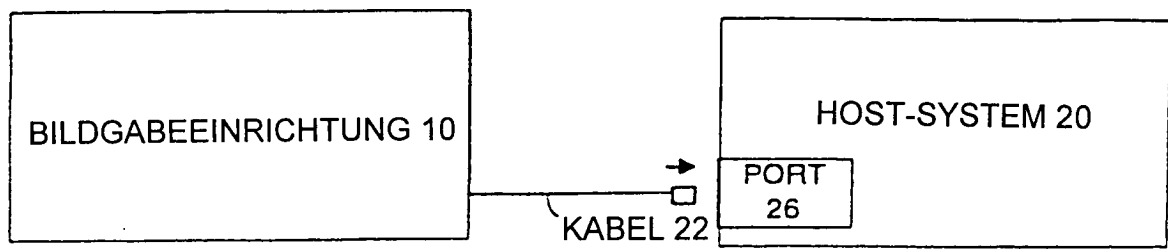


FIG. 1

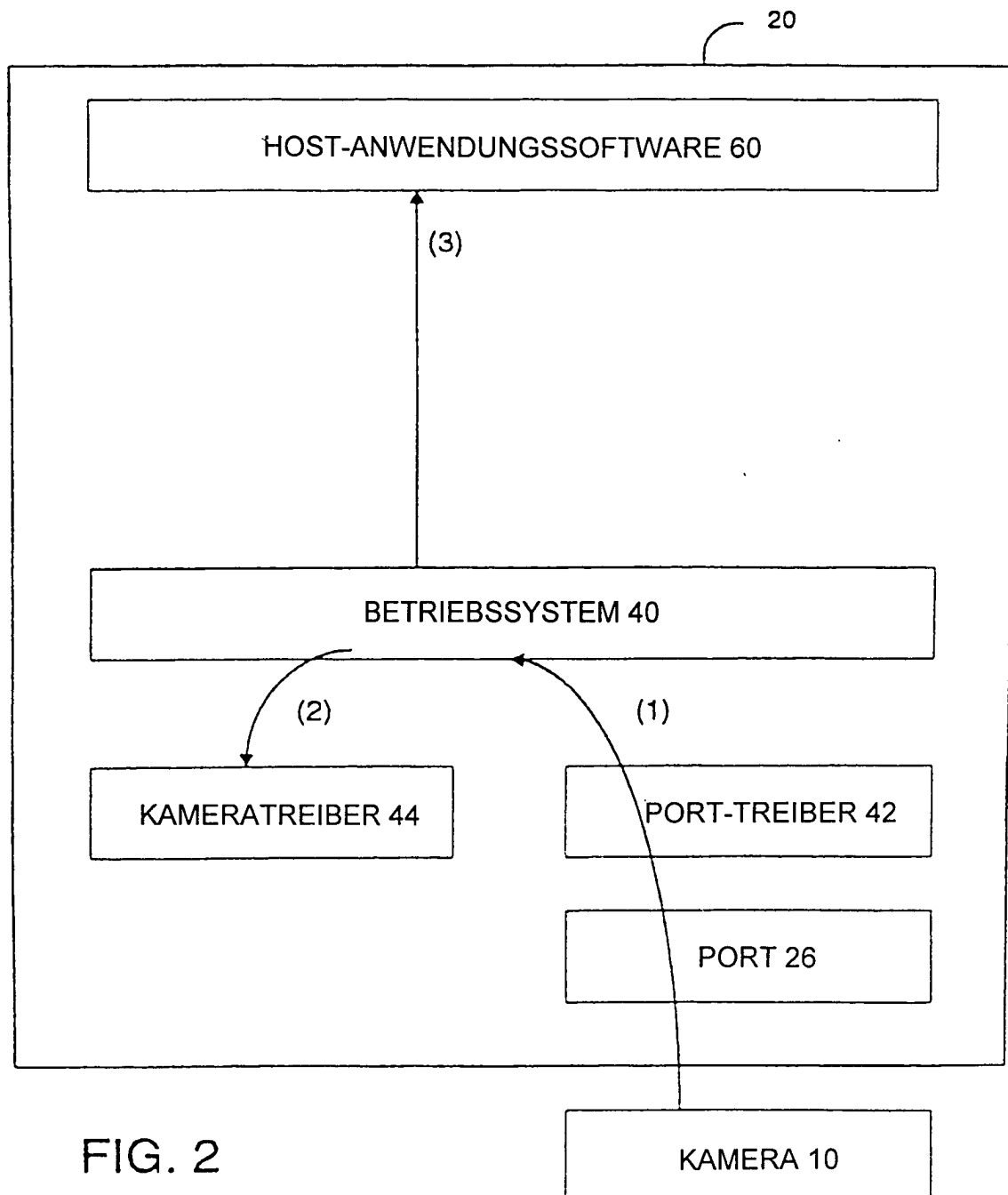
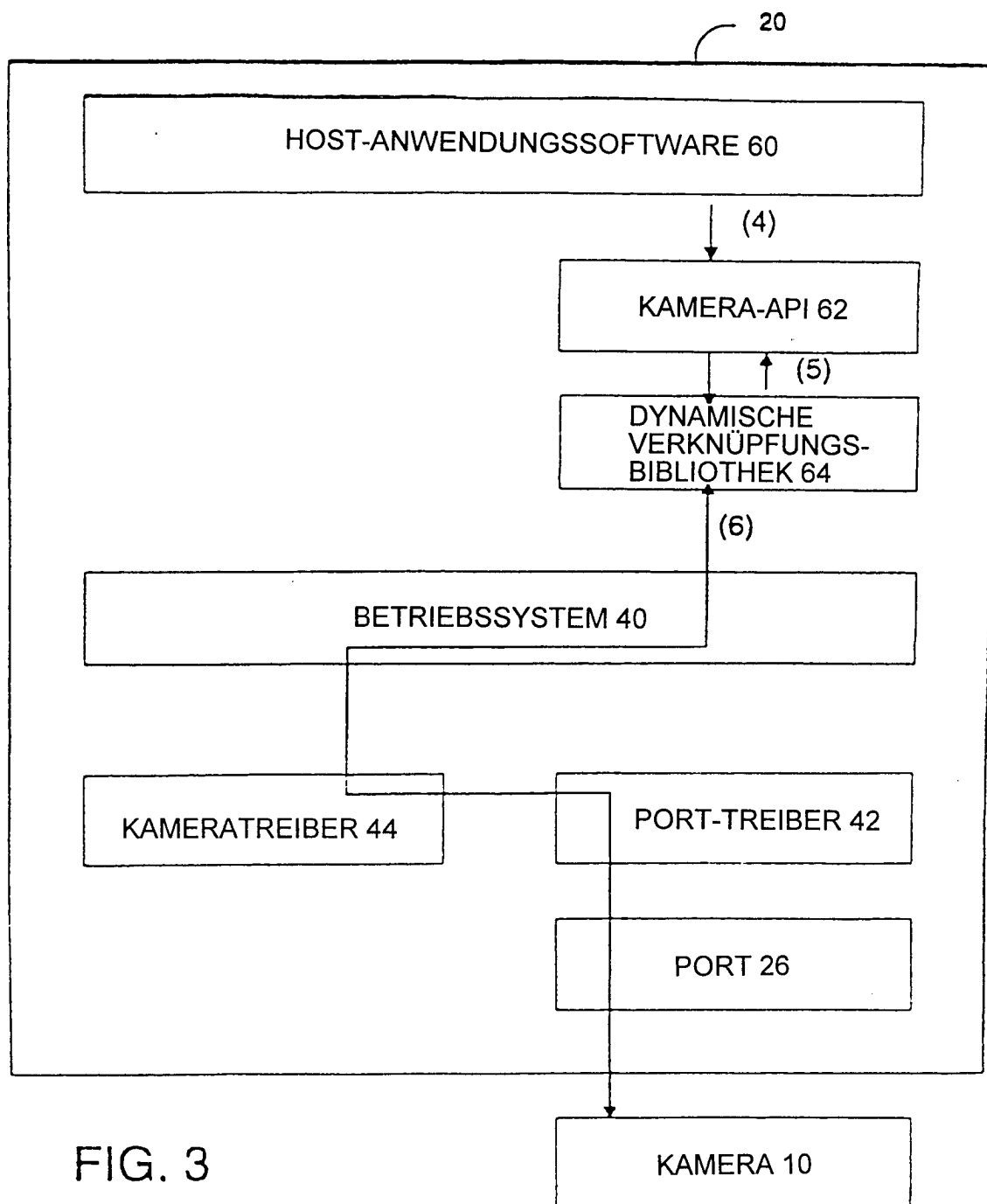


FIG. 2



BETRIEBSSYSTEM	HOST-ANWENDUNGSSOFTWARE	KAMERA-API
Öffne Host-Anwendungssoftware →	Erzeuge und Initialisiere Kamera-API → → → → Füge Hauptfenster zur Rückrufliste der Kamera-API hinzu	Setze interne Variablen zurück Lade betriebssystemabhängige DLLs Erzeuge und starte Hintergrund-Thread Füge eine CM_SIGNAL_STATUS-Nachricht in Hintergrund-threaded-Warteschlange ein

FIG. 4

KAMERA-API	HOST-ANWENDUNGSSOFTWARE
CM_SIGNAL_STATUS If Camera is not Open If OpenDriver() succeeds Close Driver() If message has not been sent before Signal all callback windows	→ WM_CAMERAAPI_STATUS If Message is CM_IS_CONNECTED
CM_OPEN_DRIVER Open camera driver Check for compatible camera	← OpenDriver()
CM_GET_NO_OF_IMAGES Returns number of images on camera	← Get number of images
CM_GET_IMAGE_LIST Returns list of image names and sizes	← If number of images > 0 Get Image List
CM_GET_IMAGE_BY_NAME Returns image with given name	← Download images, one-by-one

FIG. 5

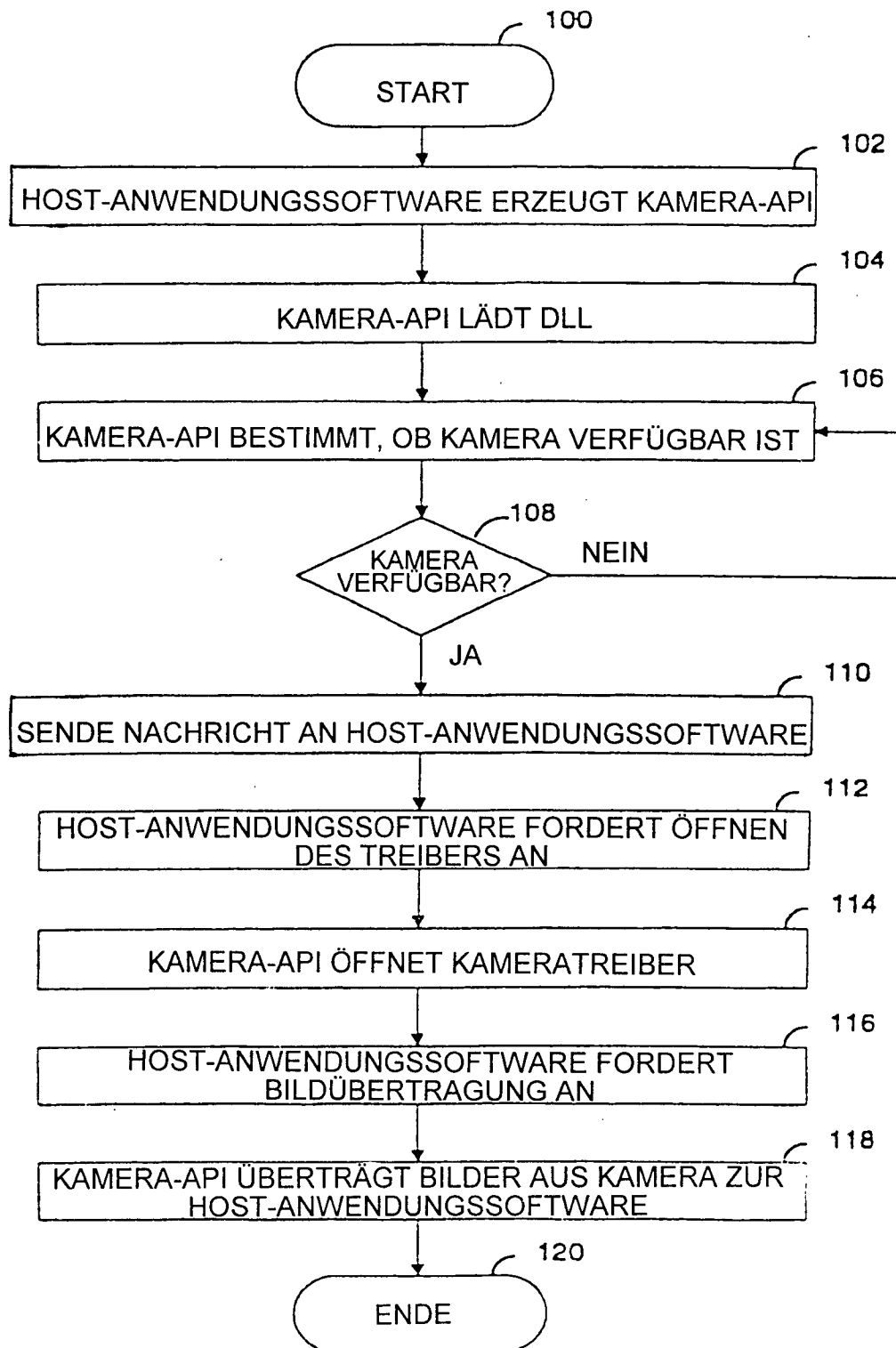


FIG. 6

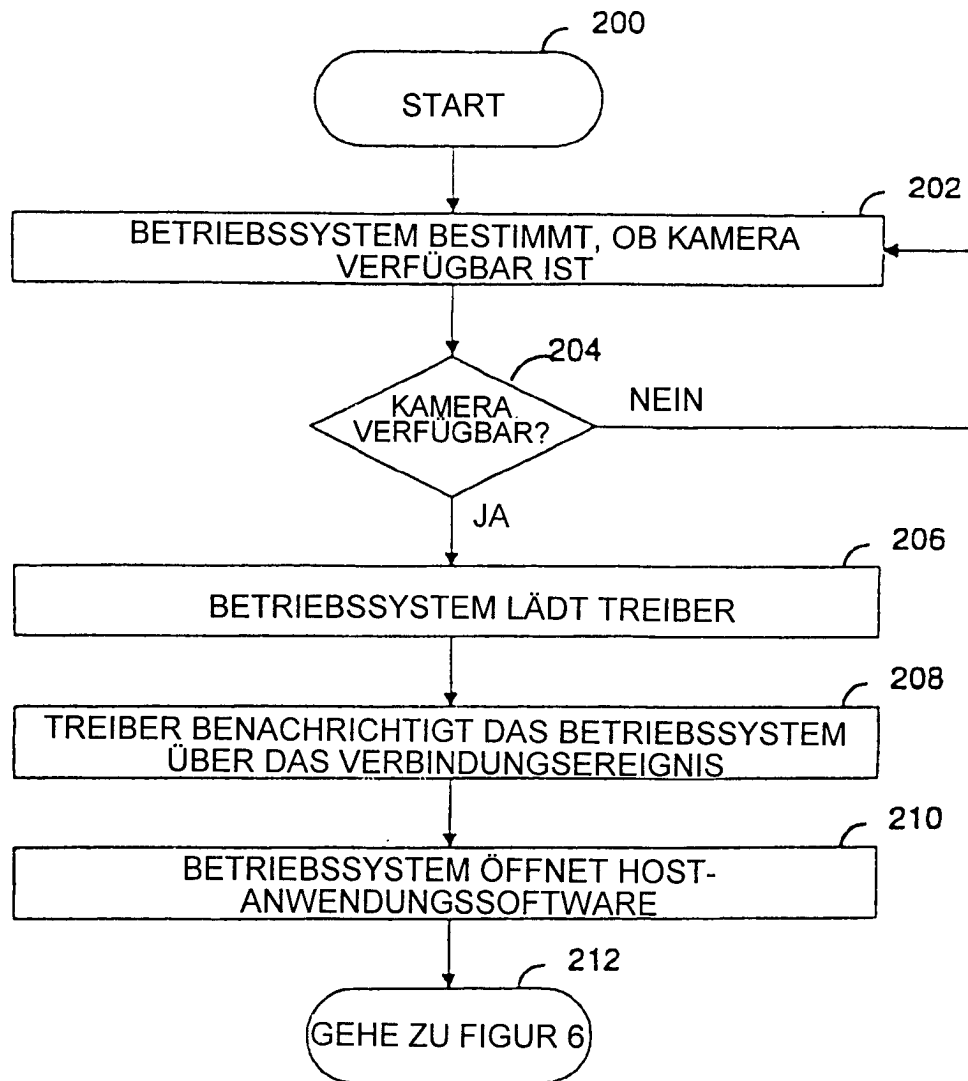


FIG. 7