

**(19) 대한민국특허청(KR)**  
**(12) 공개특허공보(A)**

**(51) Int. Cl.<sup>7</sup>**  
**H04N 5/232**

**(11) 공개번호** 특2001-0033898  
**(43) 공개일자** 2001년04월25일

(21) 출원번호	10-2000-7007472		
(22) 출원일자	2000년07월05일		
번역문제출일자	2000년07월05일		
(86) 국제출원번호	PCT/US1998/26970	(87) 국제공개번호	WO 1999/35826
(86) 국제출원출원일자	1998년12월17일	(87) 국제공개일자	1999년07월15일
(81) 지정국	AP ARIP0특허 : 가나 감비아 케냐 레소토 말라위 수단 스와질랜드 우간다 짐바브웨		
EA 유라시아특허 : 아르메니아 아제르바이잔 벨라루스 키르기즈 카자흐스탄 몰도바 러시아 타지키스탄 투르크메니스탄			
EP 유럽특허 : 오스트리아 벨기에 스위스 사이프러스 독일 덴마크 스페인 핀란드 프랑스 영국 그리스 아일랜드 이탈리아 룩셈부르크 모나코 네덜란드 포르투칼 스웨덴			
OA OAPI특허 : 부르키나파소 베냉 중앙아프리카 콩고 코트디브와르 카메룬 가봉 기네 기네비쓰 말리 모리타니 니제르 세네갈 차드 토고			
국내특허 : 알바니아 아르메니아 오스트리아 오스트레일리아 아제르바이잔 보스니아-헤르체고비나 바베이도스 불가리아 브라질 벨라루스 캐나다 스위스 중국 쿠바 체코 독일 덴마크 에스토니아 스페인 핀란드 영국 그레나다 그루지야 가나 감비아 크로아티아 헝가리 인도네시아 이스라엘 인도 아이슬란드 일본 캐냐 키르기즈 북한 대한민국 카자흐스탄 세인트루시아 스리랑카 라이베리아 레소토 리투아니아 룩셈부르크 라트비아 몰도바 마다가스카르 마케도니아 몽고 말라위 멕시코 노르웨이 뉴질랜드 폴란드 포르투칼 루마니아 러시아 수단 스웨덴 싱가포르 슬로베니아 슬로바키아 시에라리온 타지키스탄 투르크메니스탄 터키 트리니다드토바고 우크라이나 우간다 미국 우즈베키스탄 베트남 유고슬라비아 짐바브웨			

(30) 우선권주장	09/003,732 1998년01월07일 미국(US)
(71) 출원인	인텔 코오퍼레이션 피터 엔. 데트킨
미합중국 캘리포니아 산타클라라 미션 칼리지 블러바드 2200	
(72) 발명자	피치너마크알
미국아리조나85226챈들러노스리타코트844	
(74) 대리인	특허법인 신성 최종식, 특허법인 신성 박해천, 특허법인 신성 배순자, 특허법인 신성 정지원, 특허법인 신성 박정후, 특허법인 신성 원석희

**심사청구 : 있음****(54) 활상 장치와 호스트 시스템 사이에서의 영상 정보의 자동전송방법****요약**

활상 장치(10)와 호스트 시스템(20) 사이에서 영상 정보가 전송된다. 호스트 시스템(20)은 활상 장치(10)가 그 호스트 시스템(20)에 접속되는 것을 검출한다. 이러한 활상 장치(10)의 검출에 응답하여, 활상 장치(10)와 호스트 시스템(20) 사이에서 하나 또는 그 이상의 영상이 전송된다.

**대표도****도3****색인어**

영상, 전송, 디지털 카메라, 호스트, 활상장치, 컴퓨터, 운영체제

**명세서****기술분야**

본 발명은 영상 분야에 관한 것으로서, 특히 활상 장치(imaging device)와 호스트 시스템 사이에서 영상 정보를 전송하는 것에 관한 것이다.

## 배경기술

카메라와 같은 활상 장치는 통상적으로 정지 영상 또는 동영상(비디오 영상) 정보를 필름이나 비디오 테이프 또는 다른 매체에 저장한다. 디지털 카메라는 영상 정보를 디지털 포맷으로 포착하고(capture), 그 영상 정보를 플래시 메모리와 같은 메모리나 또는 다른 디지털 저장 매체에 저장한다. 디지털 영상 정보는 개인용 컴퓨터와 같은 호스트 시스템으로 다운로드될 수 있다. 다음에, 영상 정보는 호스트 시스템에 내재된 소프트웨어 애플리케이션을 이용하여 영상을 회전시키거나, 영상을 다듬거나 또는 그렇지 않으면 영상을 변경함으로써 조작될 수 있다.

호스트 시스템에서 영상을 처리하기 위해, 사용자는 호스트 시스템에 활상 장치를 부착하고, 그 활상 장치와의 인터페이스를 위한 애플리케이션 소프트웨어를 개시하고, 활상 장치와 호스트 시스템 사이에서 영상 정보를 전송한다. 이러한 각각의 작업은 여러 단계로 이루어질 수 있으며, 컴퓨터 기술에 숙련되지 않은 촬영자에게는 위험적인 것이 될 수 있다.

## 발명의 상세한 설명

활상 장치와 호스트 시스템 사이에서 영상 정보를 전송하는 방법이 개시된다. 호스트 시스템은 활상 장치가 그 호스트 시스템에 접속되는 것을 검출한다. 이러한 활상 장치의 검출에 응답하여, 활상 장치와 호스트 시스템 사이에서 하나 또는 그 이상의 영상이 전송된다.

본 발명의 다른 특징 및 장점은 첨부도면 및 다음의 상세한 설명으로부터 명백해질 것이다.

## 도면의 간단한 설명

도1은 호스트 시스템에 부착 가능한 활상 장치를 도시한 도면.

도2는 활상 장치(10)가 호스트 시스템(20)에 처음으로 접속될 때 호스트 시스템의 구성요소 사이에서의 정보 흐름의 일실시예를 도시한 도면.

도3은 호스트 애플리케이션 소프트웨어(60)가 개시된 이후의 호스트 시스템의 구성요소들 사이의 정보 흐름의 일실시예를 도시한 도면.

도4는 폴링 초기화 프로세스의 일실시예를 도시한 도면.

도5는 카메라 API(62)와 호스트 애플리케이션 소프트웨어(60) 사이의 폴링 프로세스의 일실시예를 도시한 도면.

도6은 활상 장치와 호스트 시스템 사이에서 영상을 전송하는 프로세스의 일실시예의 흐름도.

도7은 카메라와 같은 활상 장치가 호스트 시스템에 부착된다는 것을 검출하는 프로세스의 일실시예의 흐름도.

## 실시예

활상 장치와 호스트 시스템 사이에서 영상 정보를 전송하는 방법이 개시된다. 일실시예에서, 영상 정보의 전송은 활상 장치가 호스트 시스템에 접속될 때 자동적으로 수행된다.

활상 장치는 카메라와 같은 영상 포착(capture) 장치가 될 수 있다. 대안으로, 여기에 개시되는 기술은 영상 정보를 저장할 수 있는 어떠한 장치와도 함께 사용될 수 있다. 호스트 시스템은 영상 정보를 조작할 수 있는 어떠한 시스템도 될 수 있다. 예를 들어, 호스트 시스템은 Intel Pentium 또는 Pentium 프로세서 상에서 실행되는 IBM-호환성 개인용 컴퓨터가 될 수 있다. 그러나, 대안으로 호스트 시스템은 프린터, 플로터, 팩스 머신, 디스플레이 장치 또는 저장 장치가 될 수도 있다.

도1은 호스트 시스템(20)에 부착될 수 있는 활상 장치(10)를 도시하고 있다. 일실시예에서, 활상 장치(10)는 케이블(26)을 통해 호스트 시스템(20)의 포트(26)에 부착된다. 활상 장치(10)는 높은 데이터 전송 속도를 지원하는 데이터 전송 프로토콜을 이용하여 호스트 시스템(20)에 연결되는 것이 바람직하다. 일실시예에서, 활상 장치(10)는 범용 직렬 버스(USB) 접속을 통해 호스트 시스템(20)에 연결된다. USB 접속은 최대 15Mb/s의 데이터 전송 속도를 제공한다. 대안으로, 1394 프로토콜과 같은 다른 접속 및 전송 프로토콜이 사용될 수 있다. (<http://www.usb.org/>로부터 USB에 관해 더욱 많은 정보를 얻을 수 있다. 1394 표준은 IEEE에 의해 유지관리되고 분배된다. 1394의 구현예인 Firewire가 IEEE 표준 1394-1995에 의해 정의되어 있다.)

도2 및 도3은 호스트 시스템(20)의 구성요소들과 활상 장치(카메라)(10) 사이의 관계 및 메시징을 보여주는 실시예를 도시하고 있다. 도2는 활상 장치(10)가 호스트 시스템(20)에 처음으로 접속될 때 호스트 시스템(20)의 구성요소들 사이의 정보 흐름의 일실시예를 도시하고 있다. 호스트 시스템(20)은 운영체제(O/S)(40)와 호스트 애플리케이션 소프트웨어(60)를 포함한다. 호스트 시스템(20)은 카메라(10)와 같은 활상 장치가 호스트 시스템(20)에 접속되는 때를 검출한다. 일실시예에서, 운영체제(40)는 포트(26)를 폴링(polling)함으로써 카메라(10)가 호스트 시스템에 접속되어 있는지 여부를 검출한다. 운영체제(40)와 포트(26) 사이의 인터페이스를 제공하기 위해 포트 드라이버(42)가 사용될 수 있다. 일실시예에서, 포트(26)는 USB 포트이고, 포트 드라이버는 USB 드라이버이다.

운영체제(40)는 다양한 상이한 운영체제 중 하나가 될 수 있다. 일실시예에서, 운영체제는 마이크로소프

트사에 의해 제조된 Windows 95 또는 Windows 98과 같은 윈도우 운영체제이다. Windows 98은 포트의 폴링을 허용하는 흑(hooks)을 포함한다. 다른 운영체제가 이러한 폴링을 제공하도록 변경될 수도 있다. 폴링은 사용자가 그것이 수행되고 있다는 것을 알 필요가 없도록 백그라운드로 수행되는 것이 바람직하다. 대안으로, 호스트 애플리케이션 소프트웨어(60)가 포트(26)의 폴링을 수행할 수 있다. 그러나, (호스트 애플리케이션 소프트웨어(60) 대신에) 운영체제(40)에 의한 폴링이 성능상의 장점을 갖고 있는데, 그 이유는 운영체제에는 이미 키보드 누름, 마우스 이동 등과 같은 다양한 동작을 폴링하도록 셋업되어 있기 때문이다. 예시 목적을 위해, 다음의 설명에서는 운영체제가 폴링하는 것으로 가정한다. 이 기술분야에 통상의 지식을 가진 자는 애플리케이션이 폴링을 할 수 있도록 허용하는 변형예를 만들 수 있을 것이다.

카메라(10)가 호스트 시스템(20)의 포트(26)에 접속되면, 포트 드라이버(42)는 카메라가 호스트 시스템(20)에 접속되었다는 것을 운영체제(40)에 알려준다. 이것은 도2에 화살표(1)로 도시되어 있다. 그러면, 운영체제(40)는 그 장치를 카메라로서 식별하고, 화살표(2)로 도시된 바와 같이 대응하는 소프트웨어 드라이버(44)를 메모리로 로드한다. 일실시예에서, 운영체제(40)는 식별자를 얻기 위해 카메라(10)에 질의한다. 다음에, 운영체제(40)는 그 식별자에 대응하는 소프트웨어 드라이버(44)를 로드한다. 본 예에서, 카메라 드라이버(44)는 운영체제(40)에 의해 로드된다.

다음에, 운영체제(40)는 카메라에 대응하는 하나 또는 그 이상의 소프트웨어 애플리케이션을 로드한다. 일실시예에서, 운영체제(40)는 소프트웨어 애플리케이션이 등록될 수 있도록 허용한다. (특정 식별자를 가진 카메라가 검출되는 바와 같이) 소정의 조건에 정합하면, 등록된 호스트 소프트웨어 애플리케이션이 로드된다. 이 경우에, 화살표(3)로 도시된 바와 같이, (카메라를 위한) 호스트 애플리케이션 소프트웨어(60)가 로드된다. 일실시예에서, 카메라 드라이버(44)는 호스트 애플리케이션 소프트웨어(60)를 개시하도록 운영체제(40)에 신호한다. 다음에, 호스트 애플리케이션 소프트웨어(60)는 활성 장치(카메라)(10)와 호스트 시스템(20) 사이에서 영상 정보의 전송을 개시한다. 또한, 호스트 애플리케이션 소프트웨어(60)는 영상을 처리할 수 있다. 예를 들어, 호스트 애플리케이션 소프트웨어(60)는 영상에 대한 압축해제 및/또는 칼라 교정을 수행할 수 있다. 또한, 호스트 애플리케이션 소프트웨어(60)는 회전, 다풀기 및 다른 영상 조작 기능을 수행할 수 있다.

Windows 98과 같은 일부 운영체제는 특정 이벤트에 의해 소프트웨어 애플리케이션이 개시되도록 허용한다. 예를 들어, 카메라 드라이버(44)는 "카메라와의 접속 검출"이나 또는 "카메라의 셔터 버튼이 눌려지는 것"과 같은 등록된 이벤트에 의해 셋업될 수 있다. 그러므로, 운영체제는 카메라(10)가 접속될 때 호스트 애플리케이션 소프트웨어(60)와 같은 애플리케이션을 자동적으로 개시하도록 셋업될 수 있다.

일실시예에서, 만일 카메라(10)가 호스트 시스템(20)에 부착될 때 카메라 드라이버(44) 또는 호스트 애플리케이션 소프트웨어(60)가 호스트 시스템(20) 상에 설치되지 않았으면, 사용자는 포트(26)에 부착된 장치를 위한 카메라 드라이버(44) 및/또는 호스트 애플리케이션 소프트웨어(60)를 제공하도록 요구받는다. 일단 설치가 완료되면, 프로세스는 전술한 바와 같이 진행된다.

도3은 호스트 애플리케이션 소프트웨어(60)가 설치된 이후에 호스트 시스템의 구성요소들 사이의 정보 흐름의 일실시예를 도시하고 있다. 이 실시예에서, 로드된 이후에, 호스트 애플리케이션 소프트웨어(60)는 화살표(4)로 표시된 바와 같이 카메라 애플리케이션 프로그래밍 인터페이스(API)(62)를 생성하고 초기화시킨다. 카메라 API(62)는 그 태스크를 백그라운드 스레드에서 수행할 수 있다. 이러한 방식으로, 호스트 애플리케이션 소프트웨어(60)는 카메라 API(62)가 다른 태스크를 수행하기 전에 완료되는 것을 대기할 필요가 없다. 일실시예에서, 카메라 API(62)는 화살표(5)로 도시된 바와 같이 0/S-의존 동적 링크 라이브러리(DLL)(64)를 로드하는 COM 객체이다. 카메라 API(62)는 DLL(64)을 통해 운영체제(40)와 통신한다. (다른 실시예에서는, 카메라 API(62)가 DLL(64)을 포함한다.) 다음에, 운영체제(40)는 화살표(6)로 도시된 바와 같이 카메라 드라이버(44) 및 포트 드라이버(42)를 통해 카메라(40)와 통신한다.

도4는 폴링 초기화 프로세스의 일실시예를 도시하고 있다. 이 폴링 초기화 프로세스는 운영체제가 호스트 애플리케이션 소프트웨어를 개방(open)하는 것으로 시작된다. 다음에, 호스트 애플리케이션 소프트웨어(60)는 카메라 API(62)를 생성하고 초기화시킨다. 일실시예에서, 호스트 애플리케이션 소프트웨어(60)는 그 자체를 카메라 API의 콜백 리스트(callback list)에 부가하고, 따라서 호스트 애플리케이션 소프트웨어(60)는 카메라 API가 폴링 프로세스에 성공할 때 통지 받는다.

일실시예에서, 카메라 API는 초기화시 그 내부 변수를 리셋하고, 0/S 의존 DLL을 리셋하고, 백그라운드 스레드를 생성하고 개시한다. 다음에, 카메라 API는 카메라 드라이버를 개방하려고 시도하는 백그라운드 스레드 큐 내로 메시지를 삽입한다. (드라이버는 그 드라이버와 접속을 설정함으로써 개방된다.) 일실시예에서, 카메라 드라이버는 카메라가 부착될 때에만 개방된다. 만일 카메라 드라이버가 개방될 수 없으면, 카메라가 호스트 시스템에 접속되지 않는다. 카메라 드라이버가 개방될 수 있는 경우에, 카메라가 접속된다. 일실시예에서, 카메라 API(62)는 매 1/2초마다 카메라 드라이버를 개방하려고 시도한다.

도5는 카메라 API(62)와 호스트 애플리케이션 소프트웨어(60) 사이의 폴링 프로세스의 일실시예를 도시하고 있다. 이 실시예에서, 카메라 API(62)는 카메라 드라이버를 개방하려고 시도한다. 카메라 드라이버를 개방하는데 성공하면, 카메라 API는 카메라 드라이버를 폐쇄(closes)하고, 그 콜백 큐(callback queue) 내의 애플리케이션에 통지한다. 호스트 애플리케이션 소프트웨어(60)는 카메라 API의 콜백 큐 내에 있기 때문에, 카메라가 검출되었다는 것을 통지받는다.

이 실시예에서, 호스트 애플리케이션 소프트웨어(60)는 카메라 드라이버(44)를 개방하고 호환성 카메라 인지 검사하기 위해 카메라 API(62)에 신호함으로써 카메라 드라이버를 재개방한다. 다음에, 호스트 애플리케이션 소프트웨어(60)는 카메라 API(44)(및 운영체제(40)와 드라이버(44,42))를 통해 다양한 명령어를 카메라(10)로 전송할 수 있다. 예를 들어, 호스트 애플리케이션 소프트웨어(60)는 카메라에 저장된 영상의 수를 요구할 수 있다. 그리고, 호스트 애플리케이션 소프트웨어(60)는 영상의 명칭 및 영상 크기의 리스트를 요구하거나 또는 특정 영상을 요구할 수 있다.

도6은 활상 장치와 호스트 시스템 사이에서 영상을 전송하는 프로세스의 일실시예의 흐름도이다. 이 경우에, 활상 장치는 카메라이고, 호스트 시스템은 개인용 컴퓨터이다. 흐름도는 블록(100)에서 시작된다. 다음에, 프로세스는 블록(102)에서 계속되며, 여기서 호스트 애플리케이션 소프트웨어는 카메라 API를 생성한다. 카메라 API는 블록(104)에서 운영체제에 의존적인 DLL을 로드한다. 다음에 블록(106)에서, 카메라 API는 카메라가 이용가능한지 판단한다.

블록(108)에서 만일 카메라가 이용가능하지 않으면, 흐름도는 블록(106)으로 복귀한다. 그러나, 만일 카메라가 이용가능하면, 프로세스는 블록(110)에서 계속된다. 블록(110)에서, 카메라 API는 카메라가 이용가능하다는 것을 나타내는 메시지를 호스트 애플리케이션 소프트웨어로 전송한다. 호스트 애플리케이션 소프트웨어는 블록(112)에서 카메라 드라이버를 개방할 것을 요구한다. 카메라 API는 블록(114)에 도시된 바와 같이 카메라 드라이버를 개방함으로써 응답한다. 드라이버를 개방하는 프로세스는 카메라 드라이버와 접속을 설정하는 것을 의미한다. 블록(116)에서, 호스트 애플리케이션 소프트웨어는 영상을 카메라로부터 호스트 시스템으로 전송할 것을 요구한다. 그러면, 카메라 API는 블록(118)에서 카메라로부터의 영상 정보를 호스트 애플리케이션 소프트웨어로 전송함으로써 응답한다. 영상 정보는 영상 필셀 데이터뿐만 아니라 영상의 칼라 팔레트(palette) 정보, 압축 정보, 지향성 등과 같은 다른 정보도 포함할 수 있다. 다음에, 이 흐름도는 블록(120)에서 종료된다.

도7은 카메라와 같은 활상 장치가 호스트 시스템에 부착되는 것을 검출하는 프로세스의 일실시예의 흐름도이다. 이 흐름도는 블록(200)에서 시작된다. 다음에 블록(202)으로 진행되며, 여기서 운영체제는 카메라가 이용가능한지 판단한다. 이것은 USB 드라이버와 같은 포트 드라이버의 도움으로 수행될 수 있다. 만일 카메라가 이용가능하지 않으면, 프로세스는 블록(206)으로 복귀한다. 만일 블록(204)에서 카메라가 이용가능하면, 프로세스는 블록(206)으로 계속되며, 운영체제는 카메라 드라이버를 로드한다.

일실시예에서는, Windows 98과 같은 운영체제가 사용된다. 윈도우 98은 블록(208)에 도시된 바와 같이, 카메라가 호스트 시스템에 접속되었다는 것을(접속 이벤트) 드라이버가 운영체제가 신호할 수 있도록 허용한다. 다음에, 운영체제는 접속 이벤트에 의해 등록된 애플리케이션을 개방한다. 이 경우에, 블록(210)에 도시된 바와 같이 카메라를 위한 호스트 애플리케이션 소프트웨어가 개시된다. 다음에, 흐름도는 도6의 흐름도와 함께 계속된다.

만일 운영체제가 접속 이벤트에 근거하여 애플리케이션을 개방하는 방식을 제공하지 못하면, 대안의 실시예는 블록(208,210)으로 도시된 단계 대신에 "서비스"를 이용할 수 있다. 이러한 서비스는 사용자에 의해 설치되며, 호스트 시스템이 부팅될 때 호스트 시스템 상에서 자동적으로 개시된다. 이 서비스는 카메라가 검출될 때 호스트 애플리케이션 소프트웨어를 개방시킨다. 일실시예에서, 이 서비스는 카메라가 이용가능한지 판단하기 위해 카메라 API를 이용한다. 그러므로, 이 서비스는 도6에 도시된 바와 유사한 방식으로 멀티-호스트 애플리케이션으로서 작용한다. 그러나, 이 서비스는 카메라 드라이버로의 접속이 설정될 때 호스트 애플리케이션 소프트웨어를 개시시킨다. 다음에, 호스트 애플리케이션은 카메라로부터 영상을 전송하기 위해 카메라 드라이버로 그 자신의 접속을 설정한다.

일실시예에서, 호스트 애플리케이션 소프트웨어(60)와 카메라 드라이버(44)는 카메라(10)에 의해 적재된다. 이 호스트 애플리케이션 소프트웨어(60)와 카메라 드라이버(44)가 플로피디스크 또는 CD-ROM을 통해 적재될 수도 있다. 대안으로,

호스트 애플리케이션 소프트웨어(60)와 카메라 드라이버(44)가 월드 와이드 웹을 통해 다운로드될 수 있다. 이 호스트 애플리케이션 소프트웨어(60)와 카메라 드라이버(44)는 하드 디스크, DRAM, SRAM 또는 플래시 메모리와 같은 호스트 시스템 상의 저장 매체로 설치된다.

전술한 명세서에서, 본 발명은 특정 실시예를 참조하여 설명되었다. 그러나, 이 기술분야에 통상의 지식을 가진 자에게는, 첨부된 청구범위에 기재된 본 발명의 사상 및 범위를 벗어나지 않고 다양한 변형 및 수정이 이루어질 수 있다는 것이 명백할 것이다. 따라서, 명세서 및 도면은 제한적 의미가 아니라 예시적인 의미로 간주되어야 한다.

### (57) 청구의 범위

#### 청구항 1

활상 장치와 호스트 시스템 사이에서 영상 정보를 전송하는 방법에 있어서,

- (a) 상기 활상 장치가 상기 호스트 시스템에 접속되는 것을 검출하는 단계; 및
- (b) 상기 단계(a)에 응답하여, 상기 활상 장치와 상기 호스트 시스템 사이에서 하나 또는 그 이상의 영상을 전송하는 단계

를 포함하는 방법.

#### 청구항 2

제 1 항에 있어서,

상기 단계(b)는 상기 활상 장치가 상기 호스트 시스템의 포트에 접속되는 것을 감지하는 단계를 더 포함하는

방법.

#### 청구항 3

제 1 항에 있어서,

상기 단계(b)는 상기 활상 장치가 상기 호스트 시스템의 USB 포트에 접속되는 것을 감지하는 단계를 더 포함하는

방법.

#### **청구항 4**

제 1 항에 있어서,

상기 단계(b)는 상기 활상 장치가 상기 호스트 시스템의 1394 포트에 접속되는 것을 감지하는 단계를 더 포함하는

방법.

#### **청구항 5**

제 1 항에 있어서,

상기 단계(a) 및 (b)는,

(a) 카메라가 상기 호스트 시스템에 접속되는 것을 검출하는 단계; 및

(b) 상기 단계(a)에 응답하여 상기 카메라와 상기 호스트 시스템 사이에서 하나 또는 그 이상의 영상을 전송하는 단계를 더 포함하는

방법.

#### **청구항 6**

제 5 항에 있어서,

상기 단계(b)는,

(b) 상기 단계(a)에 응답하여 상기 카메라와 호스트 개인용 컴퓨터 시스템 사이에서 하나 또는 그 이상의 영상을 전송하는 단계를 더 포함하는

방법.

#### **청구항 7**

제 1 항에 있어서,

상기 단계(b)는,

(b) 상기 단계(a)에 응답하여 상기 활상 장치와 호스트 개인용 컴퓨터 시스템 사이에서 하나 또는 그 이상의 영상을 전송하는 단계를 더 포함하는

방법.

#### **청구항 8**

제 1 항에 있어서,

상기 단계(b)는,

(b) 상기 단계(a)에 응답하여 상기 호스트 시스템 상에서 실행되는 애플리케이션 프로그램을 개시시키는 단계를 더 포함하는

방법.

#### **청구항 9**

시스템에 있어서,

프로세서; 및

명령어를 자장하기 위한 저장 매체

를 포함하고,

상기 명령어는 상기 프로세서에 의해 실행될 때 상기 프로세서로 하여금,

(a) 활상 장치가 상기 시스템에 접속되는 것을 검출하는 단계; 및

(b) 상기 단계(a)에 응답하여 상기 활상 장치와 상기 시스템 사이에서 하나 또는 그 이상의 영상을 전송하는 단계

를 수행하도록 하는

시스템.

#### **청구항 10**

제 9 항에 있어서,

상기 저장 매체는,

상기 프로세서에 의해 실행될 때 상기 프로세서로 하여금,

(c) 상기 단계(a)에 응답하여 호스트 시스템 상에서 실행되는 애플리케이션 프로그램을 개시시키는 단계를 수행하도록 하는 명령어를 저장하는 시스템.

#### 청구항 11

다수의 명령어를 저장하고 있는 컴퓨터-판독가능 매체에 있어서,

상기 명령어는 프로세서에 의해 실행될 때, 프로세서로 하여금, 다음의 단계, 즉

(a) 활상 장치가 호스트 시스템에 접속되는 것을 검출하는 단계; 및

(b) 상기 단계(a)에 응답하여 상기 활상 장치와 상기 호스트 시스템 사이에서 하나 또는 그 이상의 영상을 자동적으로 전송하는 단계

를 수행하도록 하는

컴퓨터-판독가능 매체.

#### 청구항 12

제 11 항에 있어서,

상기 단계(b)는,

상기 단계(a)에 응답하여 상기 호스트 시스템 상에서 실행되는 애플리케이션 프로그램을 개시시키는 단계를 더 포함하는

컴퓨터-판독가능 매체.

#### 청구항 13

카메라로부터 개인용 컴퓨터로 영상 정보를 전송하는 방법에 있어서,

(a) 상기 카메라가 상기 개인용 컴퓨터에 접속되는 것을 검출하는 단계;

(b) 상기 단계(a)에 응답하여 카메라 드라이버를 로드하는 단계;

(c) 상기 카메라가 상기 개인용 컴퓨터에 접속되었다는 것을 운영체제에 신호하는 단계; 및

(d) 상기 카메라로부터 상기 개인용 컴퓨터로 상기 영상 정보를 전송하는 단계

를 포함하는 방법.

#### 청구항 14

제 13 항에 있어서,

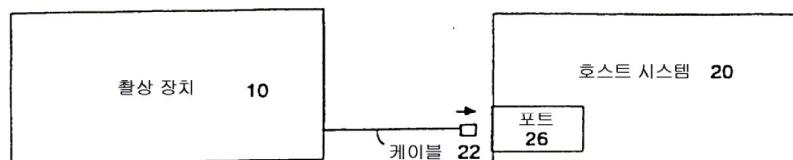
영상 정보를 전송하는 상기 단계(d)는,

(d) 상기 카메라로부터 상기 개인용 컴퓨터로 상기 영상 정보를 전송하기 위한 애플리케이션 프로그램을 개시시키는 단계를 더 포함하는

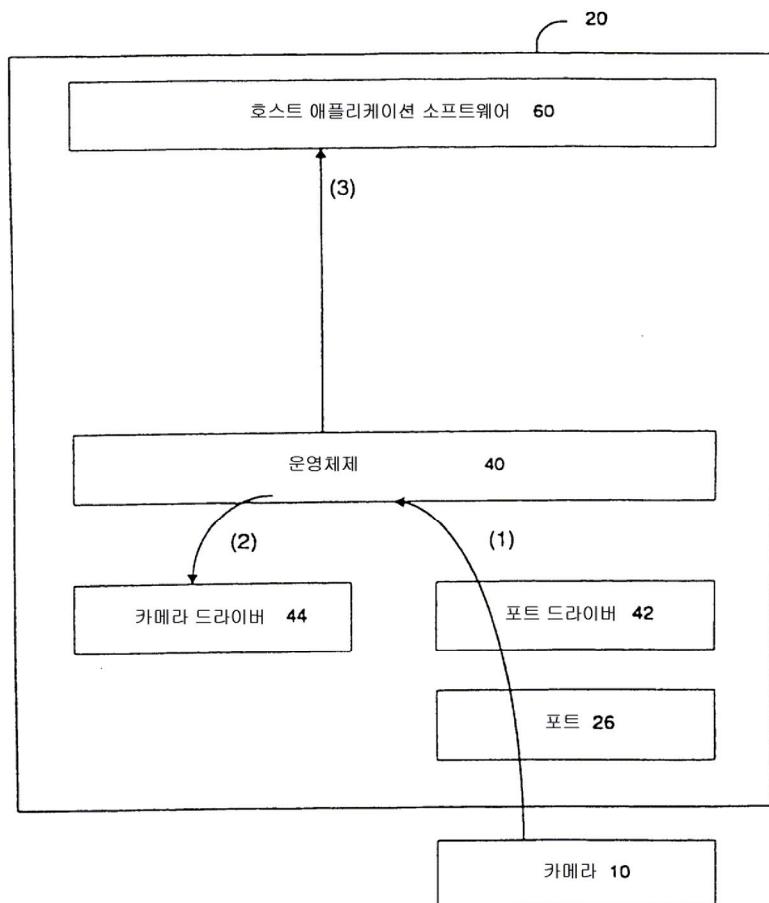
방법.

### 도면

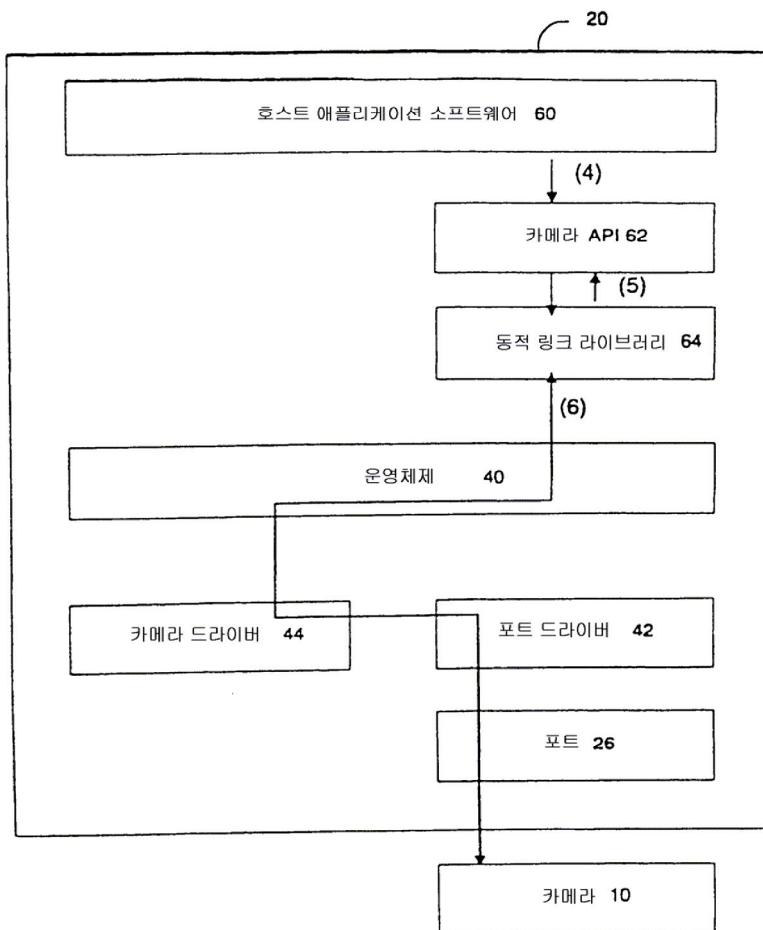
#### 도면1



## 도면2



## 도면3



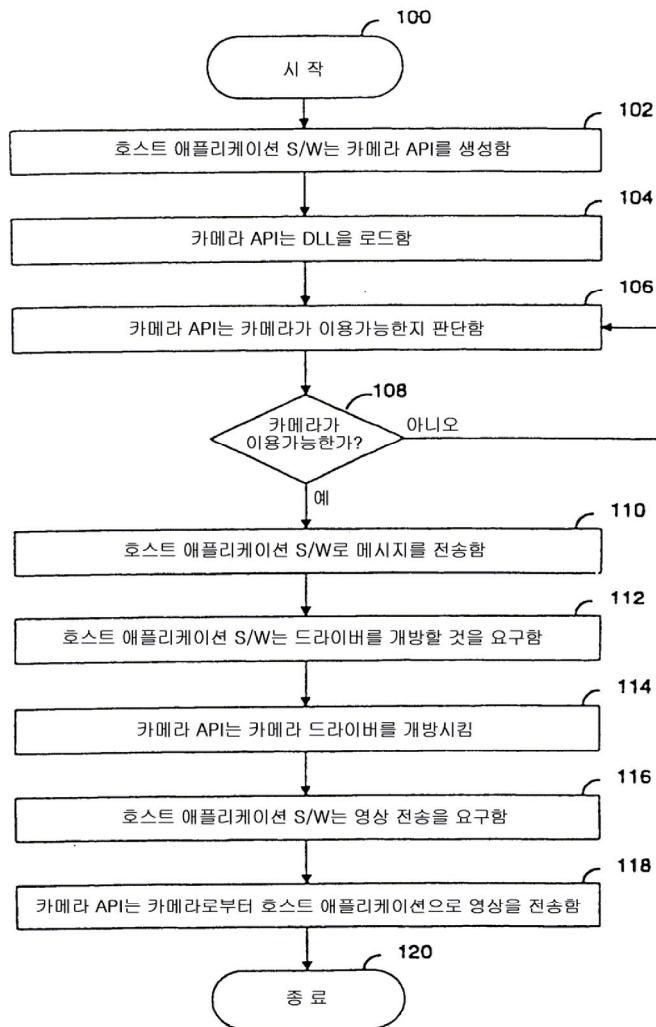
## 도면4

운영체제	호스트 애플리케이션 S/W	카메라 API
Open Host Application S/W →	Create and Initialize CameraAPI → Add main window to CameraAPI's callback list	Reset internal variables → Load O/S dependent DLLs → Create & start backgrd thread Insert a CM_SIGNAL_STATUS message into backgrd-threaded queue

## 도면5

카메라 API	호스트 애플리케이션 소프트웨어
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_CAMERAAPISIGNAL If Message is CM_IS_CONNECTED OpenDriver()
CM_OPEN_DRIVER Open camera driver Check for compatible camera	← Get number of images
CM_GET_NO_OF_IMAGES Returns number of images on camera	← If number of images > 0 Get Image List
CM_GET_IMAGE_LIST Returns list of image names and sizes	← Download images, one-by-one
CM_GET_IMAGE_BY_NAME Returns image with given name	

## 도면6



## 도면7

