



**(19) 대한민국특허청(KR)**  
**(12) 등록특허공보(B1)**

(45) 공고일자 2017년09월29일  
 (11) 등록번호 10-1783379  
 (24) 등록일자 2017년09월25일

(51) 국제특허분류(Int. Cl.)  
 HO4N 13/00 (2016.01) G06T 15/00 (2006.01)  
 HO4N 13/02 (2006.01) HO4N 13/04 (2006.01)

(21) 출원번호 10-2012-7020257  
 (22) 출원일자(국제) 2011년01월10일  
 심사청구일자 2015년12월08일

(85) 번역문제출일자 2012년08월01일  
 (65) 공개번호 10-2012-0138741  
 (43) 공개일자 2012년12월26일

(86) 국제출원번호 PCT/US2011/020690  
 (87) 국제공개번호 WO 2011/097049  
 국제공개일자 2011년08월11일

(30) 우선권주장  
 12/698,940 2010년02월02일 미국(US)

(56) 선행기술조사문헌  
 US20040218269 A1\*  
 US20070236510 A1\*  
 US20090315982 A1\*  
 \*는 심사관에 의하여 인용된 문헌

(73) 특허권자  
**마이크로소프트 테크놀로지 라이선싱, 엘엘씨**  
 미국 워싱턴주 (우편번호 : 98052) 레드몬드 원  
 마이크로소프트 웨이

(72) 발명자  
**카츠 사기**  
 미국 워싱턴주 98052-6399 레드몬드 원 마이크로  
 소프트 웨이 엘씨에이 - 인터내셔널 페이턴즈 마  
 이크로소프트 코포레이션

**애들러 아비샤이**  
 미국 워싱턴주 98052-6399 레드몬드 원 마이크로  
 소프트 웨이 엘씨에이 - 인터내셔널 페이턴즈 마  
 이크로소프트 코포레이션

**길보아 가이**  
 미국 워싱턴주 98052-6399 레드몬드 원 마이크로  
 소프트 웨이 엘씨에이 - 인터내셔널 페이턴즈 마  
 이크로소프트 코포레이션

(74) 대리인  
**김태홍**

전체 청구항 수 : 총 13 항

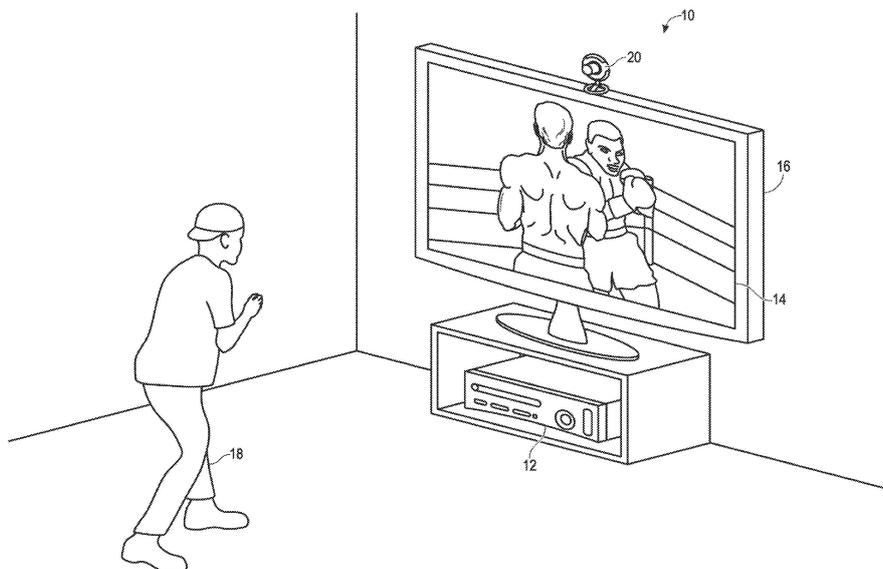
심사관 : 정성훈

(54) 발명의 명칭 **깊이 카메라 호환성**

**(57) 요약**

깊이 이미지 컨슈머와 호환되지 않는 비지원 유형을 갖고 있는 원래 깊이 이미지를 수신하여, 원래 깊이 이미지를 깊이 이미지 컨슈머와 호환되는 지원 가상 깊이 카메라 파라미터를 갖고 있는 가상 깊이 이미지로 변환함으로써, 깊이 이미지 컨슈머와 복수의 다른 깊이 이미지 프로듀서 간에 호환성을 제공한다. 이후에, 이 가상 깊이 이미지는 깊이 이미지 컨슈머에 출력된다.

**대표도**



## 명세서

### 청구범위

#### 청구항 1

깊이 카메라 상호 호환 시스템(30)에 있어서,

깊이 이미지 컨슈머(32)로부터, 가상 카메라의 해상도 및 뷰 프리스텝(46)을 명시하는, 지원되는 가상 깊이 카메라 파라미터들의 표시를 수신하는 초기화 모듈(36);

깊이 이미지 프로듀서(34, 178)로부터 원본(native) 깊이 이미지(54)를 수신하는 캡처 모듈(38);

상기 원본 깊이 이미지(54)를, 상기 깊이 이미지 컨슈머(32)와 호환가능한 지원되는 가상 깊이 카메라 파라미터들을 갖는 가상 깊이 이미지로 변환하는 가상화 모듈(40); 및

상기 깊이 이미지 컨슈머(32)에게 상기 가상 깊이 이미지를 출력하는 출력 모듈(42)

을 포함하는, 깊이 카메라 상호 호환 시스템(30).

#### 청구항 2

제1항에 있어서,

상기 지원되는 가상 깊이 카메라 파라미터들의 표시는, 가상 깊이 카메라가 기반하는 실제(actual) 깊이 카메라의 모델을 더 명시하는 것인, 깊이 카메라 상호 호환 시스템(30).

#### 청구항 3

제1항 또는 제2항에 있어서,

상기 초기화 모듈(36)은, 상기 깊이 이미지 컨슈머(32)의 상기 지원되는 가상 깊이 카메라 파라미터들과의 호환을 위해 깊이 카메라(20)를 조준하는 것을 가능케하는(facilitating) 조준 명령어(aiming instruction)들을 생성하도록 구성되는 것인, 깊이 카메라 상호 호환 시스템(30).

#### 청구항 4

제3항에 있어서,

상기 초기화 모듈(36)은 또한, 상기 깊이 카메라(20)에 상기 조준 명령어들을 전송하도록 구성되는 것인, 깊이 카메라 상호 호환 시스템(30).

#### 청구항 5

제1항 또는 제2항에 있어서,

상기 가상화 모듈(40)은, 상기 원본 깊이 이미지(54)의 뷰 프리스텝(46)을 잘라냄으로써(clipping) 상기 원본 깊이 이미지(54)를 상기 가상 깊이 이미지로 변환하도록 구성되는 것인, 깊이 카메라 상호 호환 시스템(30).

#### 청구항 6

제1항 또는 제2항에 있어서,

상기 가상화 모듈(40)은 상기 원본 깊이 이미지(54)의 해상도를 감소시킴으로써 상기 원본 깊이 이미지(54)를 상기 가상 깊이 이미지로 변환하도록 구성되는 것인, 깊이 카메라 상호 호환 시스템(30).

#### 청구항 7

제1항 또는 제2항에 있어서,

상기 초기화 모듈(36), 상기 캡처 모듈(38), 상기 가상화 모듈(40), 및 상기 출력 모듈(42)은 애플리케이션 프로그래밍 인터페이스의 일부로서 구현되는 것인, 깊이 카메라 상호 호환 시스템(30).

**청구항 8**

제1항 또는 제2항에 따른 깊이 카메라 상호 호환 시스템(30) 및 깊이 이미지 프로듀서(34, 178)를 포함하는 시스템(170)에 있어서,

상기 깊이 이미지 프로듀서(34, 178)는, 실제(raw) 깊이 카메라 데이터(60)를 원본 깊이 이미지(54)로 변환(translate)하는 온-카메라(on-camera) 프로세서(52)를 포함하고,

상기 캡처 모듈(38)은 상기 온-카메라 프로세서(52)를 통해 상기 원본 깊이 이미지(54)를 수신하도록 구성되는 것인, 시스템(170).

**청구항 9**

제1항 또는 제2항에 따른 깊이 카메라 상호 호환 시스템(30) 및 깊이 이미지 프로듀서(34, 178)를 포함하는 시스템(170)에 있어서,

상기 깊이 이미지 프로듀서(34, 178)는, 실제(raw) 깊이 카메라 데이터(60)를 원본 깊이 이미지(54)로 변환(translate)하는 오프-카메라(off-camera) 프로세서(56)를 포함하고,

상기 캡처 모듈(38)은 상기 오프-카메라 프로세서(56)를 통해 상기 원본 깊이 이미지(54)를 수신하도록 구성되는 것인, 시스템(170).

**청구항 10**

제1항 또는 제2항에 따른 깊이 카메라 상호 호환 시스템(30) 및 깊이 이미지 프로듀서(34, 178)를 포함하는 시스템(170)에 있어서,

상기 깊이 이미지 프로듀서(34, 178)는, 실제(raw) 깊이 카메라 데이터(60)를 원본 깊이 이미지(54)로 변환(translate)하도록 협력하는 온-카메라 프로세서(52) 및 오프-카메라 프로세서(56)를 포함하고,

상기 캡처 모듈(38)은 상기 온-카메라 프로세서(52) 및 오프-카메라 프로세서(56)를 통해 상기 원본 깊이 이미지(54)를 수신하도록 구성되는 것인, 시스템(170).

**청구항 11**

깊이 이미지 컨슈머(32)와 복수의 상이한 깊이 이미지 프로듀서들(34, 178) 간의 호환성을 제공하는 방법(100)에 있어서,

깊이 이미지 컨슈머(32)와 호환되지 않는, 지원되지 않는 깊이 카메라 파라미터들을 갖는 원본 깊이 이미지(54)를 수신하는 단계(104);

가상 깊이 카메라의 해상도 및 뷰 프러스텀(46)을 명시하는, 지원되는 가상 깊이 카메라 파라미터들의 표시를 수신하는 단계;

상기 원본 깊이 이미지(54)를, 상기 깊이 이미지 컨슈머(32)와 호환가능한 지원되는 가상 깊이 카메라 파라미터들을 갖는 가상 깊이 이미지로 변환하는 단계(106); 및

상기 깊이 이미지 컨슈머(32)에게 상기 가상 깊이 이미지를 출력하는 단계(108)

를 포함하는, 호환성 제공 방법(100).

**청구항 12**

제11항에 있어서,

가상 깊이 카메라가 기반하는 실제(actual) 깊이 카메라의 모델을 명시하는 상기 지원되는 가상 깊이 카메라 파라미터들의 표시를 수신하는 단계를 더 포함하는, 호환성 제공 방법(100).

**청구항 13**

하나 이상의 물리적, 비일시적인(non-transitory) 장치를 포함하는 데이터-홀딩 서브시스템(174)으로서, 상기 장치는 제11항 또는 제12항의 호환성 제공 방법(100)을 구현하도록 로직 서브시스템(172)에 의해 실행가능한 명

령어들 및 데이터를 보유하도록 구성된 것인, 데이터-홀딩 서브시스템(174).

**청구항 14**

삭제

**청구항 15**

삭제

**청구항 16**

삭제

**청구항 17**

삭제

**발명의 설명**

**배경 기술**

[0001] 카메라를 사용하여 장면의 정지 이미지(still images)를 캡처할 수 있다. 빠르게 연속적으로 찍힌 여러 개의 정지 이미지는, 각각의 프레임이 다른 정지 이미지에 해당하는 복수의 프레임을 포함하는 영화를 제작하는 데 사용될 수 있다. 이런 이미지들은 여러 다양한 애플리케이션에서 매우 유용하지만, 몇몇 목적에는 적합하지 않다. 특히, 종래의 정지 이미지와 영화는 장면에서 캡처된 다양한 표면의 상대 깊이를 정확하게 가능할 만큼 충분한 정보를 제공하지 않는다. 이러한 요구를 충족시키기 위해 여러 다른 유형의 깊이 카메라가 발전해왔다. 그러나, 다양한 다른 유형의 깊이 카메라는 여러 면에서 서로 다른 깊이 이미지를 생성할 수 있다. 따라서, 깊이 이미지를 사용하는 애플리케이션은 바라는 특성이 있는 깊이 이미지를 생성하는 특정 유형의 깊이 카메라와만 호환 가능하게 된다.

**발명의 내용**

**과제의 해결 수단**

[0002] 본 요약은 아래의 상세한 설명에서 추가적으로 설명되는 일련의 개념을 간략화된 형태로 소개하기 위한 것이다. 본 요약은 특허청구된 대상의 핵심적인 특징 또는 필수적인 특징을 밝히기 위한 것이 아니며, 특허청구된 대상의 범위를 결정하는 데 일조하기 위해 사용되는 것도 아니다. 또한 특허청구된 대상은 본 내용에서 언급된 임의의 또는 모든 단점들을 해결하는 구현에 한정되지 않는다.

[0003] 본 발명의 일 양태에 따르면, 깊이 이미지 컨슈머(depth image consumer)와 복수의 다른 깊이 이미지 프로듀서들(depth image producers) 간의 호환성은, 깊이 이미지 컨슈머와 호환되지 않는 비지원 유형을 갖고 있는 원래 깊이 이미지를 수신하여, 원래 깊이 이미지를 깊이 이미지 컨슈머와 호환되는 지원 가상 깊이 카메라 파라미터를 갖고 있는 가상 깊이 이미지로 변환함으로써 제공될 수 있다. 이후에, 이러한 가상 깊이 이미지는 깊이 이미지 컨슈머에 출력될 수 있다.

**도면의 간단한 설명**

- [0004] 도 1은 예시적인 깊이 카메라 사용 환경을 도시한다.
- 도 2는 본 발명의 일 실시예에 따른 깊이 카메라 상호 호환(interchangeability) 시스템을 도시한다.
- 도 3(a) 및 3(b)는 깊이 카메라 뷰-프러스텀(view frustums)을 개략적으로 도시한다.
- 도 4(a)-4(c)는 본 발명에 따른 예시적인 깊이 이미지 프로듀서를 도시한다.
- 도 5는 깊이 이미지 컨슈머와 복수의 다른 깊이 이미지 프로듀서들 간에 호환성을 제공하는 예시적인 방법을 도

시한다.

도 6은 깊이 카메라 호환성을 제공하는 예시적인 다른 방법을 도시한다.

도 7은 광원으로부터 떨어진 센서에서 생성되는 음영 효과(shadow effect)의 예를 개략적으로 도시한다.

도 8은 그림자 합성 알고리즘(shadow synthesis algorithm)을 통해 음영진 화소 찾기를 보여주는 예시적인 도표를 도시한다.

도 9는 깊이 추상화층(depth abstraction layer) 내의 예시적인 이미지 처리 블록을 개략적으로 도시한다.

도 10은 도 9의 예시적인 이미지 처리 블록을 보다 상세하게 개략적으로 도시한다.

도 11은 본 발명의 일 실시예에 따른 컴퓨팅 시스템을 개략적으로 도시한다.

**발명을 실시하기 위한 구체적인 내용**

- [0005] 본 발명은 깊이 카메라 호환성에 관한 것이다. 호환성 기술로 인해, 본원에 설명된 바와 같이, 다른 모델의 깊이 카메라들이 동일한 컴퓨팅 시스템에서 사용될 수 있다. 애플리케이션 개발자는, 애플리케이션 개발자가 애플리케이션을 제작할 당시에는 존재하지 않은 깊이 카메라를 포함하는 여러 가지 다양한 깊이 카메라들과 호환 가능한 애플리케이션을 용이하게 제작할 수 있다. 본 발명의 일 실시예에 따르면, 애플리케이션 개발자는 일반적인 가상 깊이 카메라와 함께 동작하도록 구성된 애플리케이션을 개발할 수 있고, 호환성 기술로 인해 다양한 다른 깊이 카메라들로부터의 원래 입력이 일반적인 가상 깊이 카메라와 호환되는 형태로 변환될 수 있다. 이처럼, 애플리케이션은 완전히 다른 깊이-파악 기술들(예컨대, 구조광(structured light), 비행 시간(time-of-flight), 스테레오 비전(stereo vision) 등)을 이용하는 깊이 카메라들을 포함하는 여러 가지 다양한 깊이 카메라들과 호환될 수 있다.
- [0006] 본원에 설명된 기술을 사용하여, 애플리케이션 개발자는 여러 가지 다른 깊이 카메라들을 지원하기 위한 지루하고 힘든 작업을 걱정할 필요가 없고, 대신 하나의 일반적인 가상 깊이 카메라에 대해 개발하면 된다. 그와 동시에, 호환성 기술은 향상된 성능, 절감된 제작 및/또는 운영 비용, 개선된 에너지 효율, 및/또는 기타 유용한 특성을 제공하는 새로운 깊이 카메라의 개발 및 구현을 용이하게 한다. 따라서, 새로운 카메라 기술이 발전함에 따라, 이러한 기술은 이전에 개발된 애플리케이션들과도 역으로 호환되는 새로운 카메라에서 구현될 수도 있다.
- [0007] 도 1은 앞서 소개한 카메라 호환성을 제공하는 깊이 카메라 상호 호환 시스템으로 설계될 수 있는 컴퓨팅 시스템(10)의 비제한적인 예를 도시한다. 특히, 도 1은 여러 가지 다양한 게임을 플레이하고, 하나 이상의 다른 미디어 유형을 플레이하고, 및/또는 비-게임 애플리케이션을 제어하거나 조작하는 데 사용될 수 있는 게임 콘솔(12)을 도시한다. 도 1은 컨슈머들, 예컨대, 컨슈머(18)에게 시각 정보를 프리젠테이션하는 데 사용되는 고화질 텔레비전 또는 HDTV(16) 형태의 디스플레이(14) 또한 도시한다. 또한, 도 1은 하나 이상의 컨슈머, 예컨대, 컨슈머(18)를 시각적으로 모니터링하는 데 사용되는 깊이 카메라(20) 형태의 캡처 장치를 도시한다. 게임 시스템과 관련하여 설명되었지만, 깊이 카메라 상호 호환 시스템은 깊이 카메라로부터의 깊이 이미지를 사용하도록 구성된 사실상 모든 컴퓨팅 시스템에서 구현될 수 있음을 이해할 것이다.
- [0008] 게임 콘솔(12)과 같은 컴퓨팅 시스템은 컨슈머(18)와 같은 하나 이상의 타겟을 인식, 분석 및/또는 트래킹하는 데 사용될 수 있다. 타겟 움직임은 운영 체제 및/또는 애플리케이션 컨트롤로서 해석될 수 있다. 운영 체제 및/또는 애플리케이션의 사실상 제어가능한 모든 양태가 컨슈머(18)와 같은 타겟의 움직임에 의해 제어될 수 있다.
- [0009] 이에 더해서 또는 이 대신에, 깊이 카메라(20)에 의해 얻어진 깊이 정보는 타겟의 트래킹 이외의 목적으로도 사용될 수 있다. 예를 들어, 깊이 카메라(20)를 사용하여 방 및/또는 하나 이상의 사물을 시각적으로 탐색할 수 있다. 깊이 카메라(20)로부터 얻어진 깊이 정보는 탐색 중이던 방 및/또는 사물(들)의 삼차원 컴퓨터 관독가능 모델을 생성하는 데 사용될 수 있다.
- [0010] 다른 예로, 깊이 카메라는 자동차나 로봇과 같은 움직이는 컴퓨팅 시스템 상의 근접 센서(proximity sensor)로서 사용될 수 있다.
- [0011] 깊이 카메라, 예컨대, 깊이 카메라(20)를 사용하여 이들 및 각종 다른 유용한 기능을 제공할 수 있다. 특히, 애플리케이션은 깊이 카메라의 깊이 정보를 이용하는 수많은 다른 기능을 제공하기 위해 개발되고 사용될 수 있다. 한편, 대부분의 애플리케이션은 보았던 장면의 현실적인 특성을 깊이 정보로부터 추론하기 위해 복잡한 알고리즘과 처리 전략을 포함한다. 예를 들어, 타겟을 트래킹하는 골격의 기계 형상으로 컨슈머(18)와 같은 타겟을 모델링하기 위해 골격 트래킹 알고리즘 및 처리 전략이 이용될 수 있다. 다른 예로, 장면의 바닥을 찾기

위해 바닥 찾기 알고리즘(floor finding algorithms)을 이용할 수 있다.

- [0012] 본원에서 설명된 카메라 호환성 기술을 이용하지 않는 애플리케이션은 특정 형식의 및/또는 특정 파라미터를 갖는 깊이 정보를 수신하도록 설계될 수 있다. 이러한 애플리케이션은 원하는 형식 및/또는 파라미터와 일치하지 않는 깊이 정보를 사용할 수 없다. 경우에 따라서는, 원하는 형식 및/또는 파라미터와 일치하지 않는 깊이 정보가 이 애플리케이션과 충돌할 수도 있다.
- [0013] 각각의 다른 유형의 깊이 카메라는 하나 이상의 형식 차이 및/또는 다른 파라미터를 갖는 깊이 정보를 생성할 수 있다. 따라서, 지금까지는 엄두해 둔 특정 깊이 카메라로 애플리케이션을 설계해왔다. 그러나, 애플리케이션이 단지 특정 깊이 카메라와만 사용되도록 설계되면, 그 애플리케이션은 생소한 형식의 및/또는 예상치 못한 파라미터를 갖는 깊이 정보를 출력하는 새로운 카메라들을 지원하지 못할 수 있다.
- [0014] 도 2는 깊이 이미지 컨슈머(32)와 복수의 다른 깊이 이미지 프로듀서들(34)(예컨대, 깊이 이미지 프로듀서(34a), 깊이 이미지 프로듀서(34b), 및 깊이 이미지 프로듀서(34c)) 간에 호환성을 제공하는 깊이 카메라 상호 호환 시스템(30)을 도시한다. 본원에서 사용된 바와 같이, 깊이 이미지 컨슈머는 깊이 카메라로부터의 깊이 정보를 이용하도록 구성된 임의의 운영 체제, 애플리케이션, 서비스, 프로세스, 모듈, 엔진 또는 기타 객체를 말한다.
- [0015] 깊이 카메라 상호 호환 시스템(30)은 깊이 이미지 컨슈머들이 어떤 모델의 깊이 카메라를 사용할 것인지에 대해 많은 유연성을 제공한다. 다른 파라미터들을 갖는 카메라, 다른 깊이 파악 기술에 기반하는 카메라 및 다른 유형의 깊이 이미지를 생성하는 카메라들이 호환되어 깊이 카메라 상호 호환 시스템(30)과 함께 사용될 수 있다.
- [0016] 예를 들어, 깊이 이미지 프로듀서(34a)는 깊이 이미지를 가늠하기 위해 구조광 기술을 사용하는 깊이 카메라를 포함할 수 있고, 깊이 이미지 프로듀서(34b)는 깊이 이미지를 가늠하기 위해 TOF(time of flight) 기술을 사용하는 깊이 카메라를 포함할 수 있다. 깊이 카메라 상호 호환 시스템(30)으로 인해, 이들 카메라는 깊이 이미지 컨슈머(32)와 호환될 수 있다. 깊이 이미지 컨슈머(32)가 어느 한 쪽 카메라와의 호환을 위해 특별히 설계되었을 필요는 없다. 깊이 이미지 컨슈머(32)는 이들 카메라가 출시되기 전에 개발되었을 수 있다. 즉, 깊이 카메라 상호 호환 시스템(30)이 깊이 이미지 컨슈머 및, 깊이 이미지 컨슈머가 개발된 후에 개발된 깊이 카메라들을 포함하는 각종 다른 깊이 카메라들과의 호환성을 용이하게 한다.
- [0017] 도 2에 도시된 바와 같이, 깊이 카메라 상호 호환 시스템(30)은 초기화 모듈(initialization module)(36), 캡처 모듈(38), 가상화 모듈(virtualization module)(40), 및 출력 모듈(42)을 포함한다. 적어도 일부 실시예에서는, 깊이 카메라 상호 호환 시스템(30)의 초기화 모듈, 캡처 모듈, 가상화 모듈, 출력 모듈 및/또는 그 밖의 컴포넌트가 API(application programming interface)의 일부로서 구현될 수 있다. API는 깊이 이미지 프로듀서와 깊이 이미지 컨슈머 간에 사용될 통신 협약을 구축하기 위해 운영 체제, 애플리케이션, 라이브러리 및/또는 기타 객체들에 의해 구현될 수 있다. API는 깊이 이미지 프로듀서와 깊이 이미지 컨슈머 간의 통신을 위해 사용되는 루틴, 데이터 구조, 객체 클래스 및 프로토콜에 대한 규격을 포함할 수 있다. 나아가, API는 깊이 이미지의 다양한 양태를 처리하도록 구성된 기능 블록을 포함하거나 기능 블록에 접속할 수 있으며, 이는 예컨대, 가상화 모듈(40) 및 에뮬레이션 모듈(44)과 관련하여 아래에서 설명될 것이다.
- [0018] 초기화 모듈(36)은 깊이 이미지 컨슈머, 예컨대, 깊이 이미지 컨슈머(32)에 대해 지원되는 가상 깊이 카메라 파라미터를 인식하도록(acknowledge) 구성될 수 있다. 이러한 인식을 가능하게 하기 위해, 특정 깊이 카메라(예컨대, 특정 제조사 및 모델의 구조광 깊이 카메라)뿐만 아니라 혹은 이를 대신하여, 가상 깊이 카메라와 호환할 수 있도록 골격 트래킹 게임 애플리케이션과 같은 깊이 이미지 컨슈머가 개발될 수 있다. 이와 같은 경우에, 깊이 이미지 컨슈머는 깊이 이미지 컨슈머가 예상하는 가상 깊이 카메라 파라미터들의 표시(예컨대, x-y-z 해상도, 뷰-프러스텀, 깊이 이미지 유형 등)를 제공하도록 구성될 수 있다. 나아가, 초기화 모듈은 지원되는 가상 깊이 카메라 파라미터들의 표시를 깊이 이미지 컨슈머로부터 수신하도록 구성될 수 있다. 앞서 논의한 바와 같이, 초기화 모듈은 깊이 이미지 컨슈머와 깊이 이미지 프로듀서가 서로 통신하기 위해 사용할 수 있는 API의 일부로서 선택적으로 구현될 수 있다.
- [0019] 지원되는 가상 깊이 카메라 파라미터들의 표시는 가상 깊이 카메라의 예상 뷰-프러스텀 및/또는 가상 깊이 카메라의 예상 해상도를 명시할 수 있다. 이런 예상 뷰-프러스텀 및/또는 예상 해상도는 특정 깊이 카메라의 원래 뷰-프러스텀 및 원래 해상도와 다를 수 있다. 지원되는 가상 깊이 카메라 파라미터들의 표시는 가상 깊이 카메라가 기초로 하는 실제 깊이 카메라의 모델을 명시할 수 있다. 다음의 설명은 초기화 모듈(36)에 의해 인식될 수 있는 상이한 파라미터들의 비제한적인 샘플링을 요약한다.

- [0020] 깊이 카메라는 일반적으로 각각의 화소에서, 가장 가까운 표면으로부터 깊이 카메라 센서까지의 방사상 거리(radial distance)를 측정한다. 이 거리는 깊이 카메라로부터의 광선에 따른 거리를 현실 세계를 모델링하는 삼차원 공간에 투사시킴으로써 세계 좌표로 변환될 수 있다. 이러한 프로세스는 엄밀한 선형 경계가 존재하지 않는 삼차원 샘플들을 생성한다. 대신에, 포인트들이 깊이 카메라의 뷰-프러스텀에 의해 - 두 개의 동심원 공의 두 삼차원 호(arcs) 사이에 갇힌 공간(렌즈 왜곡은 없다고 가정함)에 의해 제한된다. 도 3(a)는 예시적인 깊이 카메라(48)에 대한 예시적인 뷰-프러스텀(46)을 도시한다.
- [0021] 동심원 공들의 위치는 깊이 카메라로 측정가능한 필드(field)의 깊이에 의해 결정된다. 작은 동심원 공은 그 깊이 카메라에 대한 최소 측정 거리와 같은 반경을 갖고, 큰 동심원 공은 깊이 카메라에 대한 최대 측정 거리와 같은 반경을 갖는다. 공간 상의 삼차원 호의 크기는 그 깊이 카메라에 대한 시야(즉, 시야각(view angles))에 의해 결정된다. 깊이 카메라의 뷰-프러스텀의 위치와 방향은 깊이 카메라의 물리적인 위치 및 물리적인 오리엔테이션(orientation)에 의해 결정된다. 얼마나 많은 다른 화소들이 뷰-프러스텀에 포함되었는지는 깊이 카메라의 해상도에 의해 결정된다. 깊이 카메라의 민감도(해상도)는 깊이 카메라의 출력으로서 수신되는 화소 당 비트수에 달려 있다. 한편, 실제 민감도는 더 낮을 수 있다.
- [0022] 파라미터(예컨대, 필드 깊이, 시야, 해상도, 위치, 방향, 렌즈 왜곡 등)는 카메라마다 다를 수 있다. 카메라들 간의 파라미터 차이는 깊이 이미지 컨슈머들이 이런 차이에 상당히 민감하기 때문에 문제가 될 수 있다. 따라서, 특정 깊이 카메라 파라미터를 갖고 있는 특정 깊이 카메라로부터 깊이 이미지를 수신하도록 개발되는 대신에, 가상 깊이 카메라 파라미터를 갖고 있는 가상 깊이 카메라와 호환되도록 깊이 이미지 컨슈머가 개발될 수 있다. 아래에 설명된 바와 같이, 깊이 카메라 상호 호환 시스템은 실제 카메라로부터 수신되는 깊이 정보를 가상 깊이 카메라의 가상 파라미터에 따라 효과적으로 변환하여, 실제 카메라로부터의 깊이 정보가 깊이 이미지 컨슈머에 의해 사용될 수 있게 한다.
- [0023] 몇몇 실시예에서, 초기화 모듈(36)은 깊이 이미지 컨슈머의 지원되는 가상 깊이 카메라 파라미터와의 호환을 위해 깊이 카메라를 용이하게 조준하도록 조준 명령어(aiming instructions)(37)를 생성하도록 구성될 수 있다. 예를 들어, 깊이 이미지 컨슈머는 컨슈머의 양쪽 발이 스캔될 수 있도록 깊이 카메라가 바닥을 향해 아래로 조준되길 바랄 수 있다. 따라서, 이러한 바램을 전달하기 위해 조준 명령어가 생성될 수 있다. 자동으로 자신의 위치를 바꾸기 위한 모터 또는 기타 수단을 포함하는 깊이 카메라(예컨대, 깊이 이미지 프로듀서(34c)의 깊이 카메라)에 대한 실시예에서, 초기화 모듈은 깊이 카메라가 조준 명령어에 따라 자신의 위치를 바꿀 수 있도록, 깊이 카메라에 조준 명령어를 전송하도록 구성될 수 있다. 몇몇 실시예에서, 컨슈머가 깊이 카메라의 위치를 수동으로 바꿀 수 있도록, 비주얼 및/또는 오디오 명령어를 통해 조준 명령어가 컨슈머에게 전달될 수도 있다.
- [0024] 몇몇 실시예에서, 깊이 이미지 컨슈머가 기대하고 있는 좌표 시스템을 갖고 있는 깊이 이미지를 수신할 수 있도록 좌표 시스템을 추상화할 수 있다. 경우에 따라서, 이는, 시선(line of sight)을 측정하는 하드웨어를 사용하거나 또는 카메라에 대한 바닥의 각도를 측정함으로써, (예컨대) 바닥과 나란하게 되도록 좌표 시스템을 회전하는 것을 포함할 수 있다.
- [0025] 캡처 모듈(38)은 깊이 이미지 프로듀서로부터 원래 깊이 이미지를 수신하도록 구성될 수 있다. 비제한적인 예로, 캡처 모듈은 TOF(time of flight) 카메라를 포함하는 깊이 이미지 프로듀서로부터 원래 깊이 이미지를 수신할 수 있다. 앞서 논의한 바와 같이, 캡처 모듈은 깊이 이미지 컨슈머와 깊이 이미지 프로듀서가 서로 통신하기 위해 사용 가능한 API의 일부로서 선택적으로 구현될 수 있다.
- [0026] 깊이 이미지 프로듀서는 단지 깊이 카메라만을 포함하거나, 또는 오프보드(off-board) 처리 엔진과 깊이 카메라를 포함할 수도 있다. 도 4(a)에 도시된 바와 같이, 깊이 이미지 프로듀서(34')는 실제(raw) 깊이 카메라 데이터(도시되지 않음)를 원래 깊이 이미지(54)로 변환하도록 구성된 온-카메라 프로세서(on-camera processor)(52)를 포함할 수 있다. 이 경우에, 캡처 모듈(38')은 온-카메라 프로세서(52)를 통해 원래 깊이 이미지(54)를 수신하도록 구성될 수 있다. 도 4(b)에 도시된 바와 같이, 깊이 이미지 프로듀서(34'')는 실제 깊이 카메라 데이터(60)를 원래 깊이 이미지(54'')로 변환하는 오프-카메라 프로세서(off-camera processor)(56)를 포함할 수 있다. 예를 들어, 오프-카메라 프로세서(56)는 깊이 이미지 컨슈머(32'')가 실행되고 있는 컴퓨팅 시스템(10'')의 일부일 수 있다. 이 경우에, 캡처 모듈(38'')은 오프-카메라 프로세서(56)를 통해 원래 깊이 이미지를 수신하도록 구성될 수 있다. 도 4(c)에 도시된 바와 같이, 깊이 이미지 프로듀서(34''')는 실제 깊이 카메라 데이터(도시되지 않음)를 원래 깊이 이미지(54''')로 변환하도록 협력하는 온-카메라 프로세서(52''')와 오프-카메라 프로세서(56''')를 포함할 수 있다. 이 경우에, 캡처 모듈(38''')은 온-카메라 프로세서(52''')와 오프-카메라 프로세서(56''')를 통해 원래 깊이 이미지(54''')를 수신하도록 구성될 수 있다. 특히, 온-카메라 프로세

서(52'')는 실제 깊이 카메라 데이터로부터 부분적으로 처리된 중간 데이터(62)를 오프-카메라 프로세서(56'')로 전달할 수 있다. 도 4(b)의 예와 같이, 오프-카메라 프로세서(56'')는 예를 들어, 깊이 이미지 컨슈머(32'')가 실행되고 있는 컴퓨팅 시스템(10'')의 일부일 수 있다.

[0027] 다시 도 2를 참조하면, 가상화 모듈(40)은 원래 깊이 이미지를 깊이 이미지 컨슈머와 호환되는 지원 가상 깊이 카메라 파라미터를 갖는 가상 깊이 이미지로 변환하도록 구성될 수 있다. 특히, 가상화 모듈은 깊이 이미지 컨슈머로부터 초기화 모듈을 통해 수신되는 가상 파라미터의 표시에 따라 원래 깊이 이미지를 가상 깊이 이미지로 변환하도록 구성될 수 있다. 가상화 모듈은 API의 일부로서 또는, 깊이 이미지 컨슈머와 깊이 이미지 프로듀서가 통신을 위해 사용하는 API와 호환가능한 애플리케이션이나 서비스로서 선택적으로 구현될 수 있다.

[0028] 변환의 일례로, 가상화 모듈(40)은 원래 깊이 이미지의 뷰-프러스텀을 잘라냄으로써(clipping) 원래 깊이 이미지를 가상 깊이 이미지로 변환하도록 구성될 수 있다. 이러한 예는 도 3(b)에서 어느 정도 개략적으로 도시된다. 가상 뷰-프러스텀(70)은 실선으로 도시된다. 가상 뷰-프러스텀(70)은 가상 깊이 카메라 파라미터들에 의해 정해질 수 있다. 동일한 도면에 실제 깊이 카메라(74)의 원래 뷰-프러스텀(72)이 겹쳐져 있다. 원래 뷰-프러스텀(72)은 점선으로 도시된다. 비교에서 볼 수 있듯이, 원래 뷰-프러스텀(72)은 가상 뷰-프러스텀(70)보다 더 넓은 시야와 더 깊은 필드 깊이를 갖고 있다. 따라서, 원래 뷰-프러스텀(72)은 가상 뷰-프러스텀(70)을 기대하고 있는 깊이 이미지 컨슈머와 호환되지 않을 수 있다.

[0029] 가상화 모듈은 원래 뷰-프러스텀을 가상 뷰-프러스텀으로 잘라낼 수 있다. 즉, 가상 뷰-프러스텀의 시야 및/또는 필드 깊이 밖에서 깊이 카메라(74)에 의해 관측되는 깊이 정보가 깊이 이미지에서 제거될 수 있다. 몇몇 실시예에서, 가상 뷰-프러스텀의 깊이 정보에는 변함이 없는 반면, 제거된 깊이 정보는 단순히 무시할 수 있다. 몇몇 실시예에서, 가상 뷰-프러스텀의 깊이 정보를 선택적으로 수정하기 위해, 제거된 깊이 정보가 사용될 수도 있다.

[0030] 변환의 다른 일례로, 가상화 모듈(40)은 원래 깊이 이미지의 해상도를 바꿈으로써(예컨대, 원래 깊이 이미지의 해상도를 낮춤으로써) 원래 깊이 이미지를 가상 깊이 이미지로 변환하도록 구성될 수 있다. 해상도를 바꾸기 위해 사실상 모든 리샘플링(resampling) 알고리즘을 사용할 수 있다. 비제한적인 예시로, 원하는 해상도를 갖는 샘플 그리드(grid)가 실제 깊이 카메라로부터의 깊이 이미지와 개념상 맞는 최근접 알고리즘(nearest-neighbor algorithm)이 사용될 수 있다. 샘플 그리드의 각각의 샘플 화소에 그 샘플 화소와 가장 가까운 실제 깊이 카메라로부터의 화소의 깊이 값이 부여될 수 있다. 비제한적인 다른 예시로, 리샘플링 알고리즘은 가장 가까운 화소들의 평균 또는 거리 가중 평균을 취할 수도 있다.

[0031] 뷰-프러스텀을 잘라내고 해상도를 바꾸는 상기에서 제공된 예들에 제한되지 않는다. 깊이 이미지 컨슈머는 다양한 가상 파라미터들을 갖고 있는 가상 카메라를 예상하도록 설계될 수 있고, 깊이 이미지 컨슈머의 가상 깊이 카메라와 호환을 이루기 위해 실제 깊이 이미지가 이러한 파라미터들에 따라 변환될 수 있음을 이해하여야 한다.

[0032] 도 2를 다시 참조하면, 몇몇 실시예에서, 가상화 모듈(40)은 원래 깊이 이미지를 깊이 이미지 컨슈머에 의해 지원되는 지원 유형을 갖고 있는 애플리케이션 깊이 이미지로 처리함으로써 원래 깊이 이미지를 가상 깊이 이미지로 변환하도록 구성되는 애플리케이션 모듈(44)을 포함할 수 있다. 즉, 깊이 이미지 컨슈머는 특정 모델의 깊이 카메라 및/또는 특정 깊이 습득 기술(예컨대, 구조광 또는 비행 시간)을 사용하여 생성되는 깊이 이미지를 예상하도록 설계될 수 있다. 다른 유형의 깊이 카메라들의 깊이 이미지가 궁극적으로는 각각의 화소에 깊이 값이 부여되는 깊이 이미지를 생성하는 한편, 다른 유형의 깊이 카메라들 간의 차이가 각각의 화소에 부여된 깊이 값들 간에 다양한 차이를 야기할 수 있다. 깊이 이미지 컨슈머는 특정 깊이 카메라로부터의 특정 유형의 깊이 이미지를 처리하도록 설계될 수 있다. 따라서, 애플리케이션 모듈(44)은 지원되지 않는 깊이 카메라로부터의 깊이 이미지와 마치 지원되는 깊이 카메라로부터 온 것처럼 보이게 바꾸도록 구성될 수 있다.

[0033] 예를 들어, 애플리케이션 모듈(44)은 TOF 깊이 카메라로부터의 원래 깊이 이미지를 구조광 깊이 카메라에 의해 생성된 깊이 이미지를 애플리케이션하는 애플리케이션 깊이 이미지로 변환시킬 수 있다. 애플리케이션 모듈(44)은 사실상 모든 카메라 기술로 얻는 원래 깊이 이미지를 사실상 다른 모든 유형의 카메라 기술로 생성된 깊이 이미지를 애플리케이션하는 애플리케이션 깊이 이미지로 변환하도록 구성될 수 있다.

[0034] 몇몇 실시예에서 애플리케이션 모듈(44)은 가상화 모듈(40)의 일부일 수 있지만, 본원에서 설명된 애플리케이션 기법은 뷰-프러스텀 잘라내기 및/또는 해상도 바꾸기와 같은 다른 모든 가상화 기법과는 독자적으로 실행될 수 있음을 이해하여야 한다. 어떤 경우든, 애플리케이션 모듈(44)은 API의 일부로서 또는, 깊이 이미지 컨슈머와 깊이

이미지 프로듀서가 통신을 위해 사용하는 API와 호환가능한 애플리케이션이나 서비스로서 선택적으로 구현될 수 있다.

- [0035] 애플리케이션 모듈이 포함되는 경우, 애플리케이션 모듈은 노이즈 억제 모듈(noise suppression module)(80), 에지 강조 모듈(edge enhancement module)(82), 무효화 모듈(invalidation module)(84), 깊이 양자화 모듈(depth quantization module)(86), 작은 객체 정정 모듈(small object correction module)(88), 음영 시뮬레이션 모듈(shadow simulation module)(90) 중 하나 이상을 포함할 수 있다.
- [0036] 노이즈 억제 모듈(80)은 원래 깊이 이미지에서 서로 다른 깊이 값을 갖고 있는 인접 화소 영역들 사이의 깊이 에지들을 보존하도록 구성될 수 있다. 노이즈 억제 모듈(80)의 기능은 도 6의 118과 관련하여 아래에서 보다 상세하게 설명된다.
- [0037] 에지 강조 모듈(82)은 원래 깊이 이미지에서 서로 다른 깊이 값을 갖고 있는 인접 화소 영역들 사이의 깊이 에지들을 강조하도록 구성될 수 있다. 에지 강조 모듈(82)의 기능은 도 6의 126과 관련하여 아래에서 보다 상세하게 설명된다.
- [0038] 무효화 모듈(84)은 사진 결정된 범위 밖의, 결합된 조도(illumination) 및 경사도(obliqueness)를 갖고 있는 화소를 무효화하도록 구성될 수 있다. 무효화 모듈(84)의 기능은 도 6의 120-124와 관련하여 아래에서 보다 상세하게 설명된다.
- [0039] 깊이 양자화 모듈(86)은 깊이 값을 양자화하도록 구성될 수 있다. 깊이 양자화 모듈(86)의 기능은 도 6의 128과 관련하여 아래에서 보다 상세하게 설명된다.
- [0040] 작은 객체 정정 모듈(88)은 임계 크기보다 작은 객체에 속해 있는 화소들에 더 깊은 깊이 값을 할당하도록 구성될 수 있다. 작은 객체 정정 모듈(88)의 기능은 도 6의 130과 관련하여 아래에서 보다 상세하게 설명된다.
- [0041] 음영 시뮬레이션 모듈(90)은 TOF 깊이 카메라로부터 가상적으로 떨어져 있는 가상 일루미네이터의 가상 밴티지 포인트(virtual vantage point)로부터 가상적으로 가려진 화소들에 음영 화소 값을 할당하도록 구성될 수 있다. 음영 시뮬레이션 모듈(90)의 기능은 도 6의 132와 관련하여 아래에서 보다 상세하게 설명된다.
- [0042] 가상화 모듈(40)은 실제 깊이 카메라로부터의 원래 깊이 이미지를 깊이 이미지 컨슈머와 호환되는 가상 깊이 카메라 파라미터들을 지원하는 가상 깊이 이미지로 변환한다. 출력 모듈(42)은 이런 가상 깊이 이미지를 깊이 이미지 컨슈머에게 출력하도록 구성된다. 이후에, 깊이 이미지 컨슈머는 분석 및/또는 추가 처리를 위해 가상 깊이 이미지를 수신할 수 있다. 출력 모듈은 깊이 이미지 컨슈머와 깊이 이미지 프로듀서가 서로 통신하기 위해 사용할 수 있는 API의 일부로서 선택적으로 구현될 수 있다. 상기에서 초기화 모듈(36), 캡처 모듈(38), 가상화 모듈(40) 및 출력 모듈(42)은 별개의 모듈인 것으로 설명되었지만, 이 모듈 중 둘 이상이 공통 API로 결합되어 동작할 수도 있음을 이해할 것이다.
- [0043] 도 5는 깊이 이미지 컨슈머와 복수의 다른 깊이 이미지 프로듀서들 사이에 호환성을 제공하는 방법(100)을 도시한다. 방법(100)은 예컨대, 도 2의 깊이 카메라 상호 호환 시스템을 사용하여 구현될 수 있다.
- [0044] 102에서, 방법(100)은 지원되는 가상 깊이 카메라 파라미터의 표시를 수신하는 단계를 포함한다. 전술한 바와 같이, 가상 깊이 카메라 파라미터는 가상 깊이 카메라의 뷰-프러스텀, 가상 깊이 카메라의 해상도, 가상 깊이 카메라가 기반하는 실제 깊이 카메라의 모델, 및/또는 가상 깊이 카메라의 기타 특성들을 명시할 수 있다.
- [0045] 104에서, 방법(100)은 깊이 이미지 컨슈머와 호환되지 않는, 비지원 깊이 카메라 파라미터를 갖고 있는 원래 깊이 이미지를 수신하는 단계를 포함한다. 106에서, 방법(100)은 상기 원래 깊이 이미지를 깊이 이미지 컨슈머와 호환되는, 지원 가상 깊이 카메라 파라미터를 갖고 있는 가상 깊이 이미지로 변환하는 단계를 포함한다. 전술한 바와 같이, 원래 깊이 이미지의 변환에는 원래 깊이 이미지의 뷰-프러스텀 잘라내기, 원래 깊이 이미지의 해상도 낮추기, 및/또는 원래 깊이 이미지를 깊이 이미지 컨슈머에 의해 지원되는 지원 유형을 갖는 애플리케이션 깊이 이미지로 처리하기 중 하나 이상이 포함될 수 있다.
- [0046] 108에서, 방법(100)은 깊이 이미지 컨슈머에게 가상 깊이 이미지를 출력하는 단계를 포함한다. 지원되는 가상 깊이 카메라 파라미터에 맞게 가상 깊이 이미지가 구체적으로 조정되기 때문에, 깊이 이미지 컨슈머가 사용할 수 있다. 한편, 방법(100)은 다른 깊이 카메라들의 다양한 다른 원래 깊이 이미지에 적용될 수 있기 때문에, 깊이 이미지 컨슈머는 하나의 특정 유형의 깊이 카메라로부터 깊이 이미지를 수신하는 것에 제한되지 않는다.
- [0047] 도 6은 깊이 카메라 호환성을 제공하는 다른 방법(110)을 도시한다. 112에서, 방법(110)은 깊이 이미지 컨슈머

로부터 그 깊이 이미지 컨슈머에 대해 지원되는 가상 깊이 카메라 파라미터들을 수신하는 단계를 포함한다. 이는 예를 들어, 깊이 이미지 컨슈머와 호환되는 모델 깊이 카메라(예컨대, 구조광 깊이 카메라)의 표시를 수신하는 단계를 포함한다.

[0048] 114에서, 방법(110)은 깊이 이미지 컨슈머에 의해 지원되지 않는 비지원 유형을 갖고 있는 원래 깊이 이미지를 수신하는 단계를 포함한다. 예를 들어, 원래 깊이 이미지는 모델 깊이 카메라가 아닌 소스로부터 올 수 있다. 따라서, 원래 깊이 이미지는 깊이 이미지 컨슈머와 호환되지 않을 수 있고, 따라서 비지원 유형을 가질 수 있다. 앞서 소개한 예를 계속 들면, 깊이 이미지 컨슈머는 구조광 깊이 카메라로부터의 깊이 이미지를 지원할 수 있다. 그러나, 원래 깊이 이미지는 TOF 깊이 카메라와 같이 다른 소스로부터 수신될 수 있고, 따라서 원래 깊이 이미지는 깊이 이미지 컨슈머에 대한 비지원 유형을 가지게 된다.

[0049] 116에서, 방법(110)은 원래 깊이 이미지를 깊이 이미지 컨슈머에 의해 지원되는 지원 유형을 갖는 에플레이션 깊이 이미지로 처리하는 단계를 포함한다. 즉, 에플레이션 깊이 이미지는 모델 깊이 카메라에 의해 생성되는 깊이 이미지와 같이, 깊이 이미지 컨슈머와 호환되는 깊이 이미지를 에플레이션하도록 수정될 수 있다. 앞서 소개한 예를 계속 들면, 예컨대, TOF 깊이 카메라로부터 수신된 원래 깊이 이미지는 구조광 깊이 카메라에 의해 생성된 깊이 이미지를 에플레이션하는 에플레이션 깊이 이미지로 처리될 수 있다.

[0050] 도 6에 도시된 바와 같이, 원래 깊이 이미지를 에플레이션 깊이 이미지로 처리하는 단계는 다양한 기법을 이용할 수 있으며, 이는 아래에서 보다 상세하게 설명된다. 이런 기법은 제한되지 않음을 이해할 수 있다. 또한, 이 대신에 또는 이에 더하여, 도 6에 도시되지 않은 추가적인 기법이 적용될 수도 있다.

[0051] 118에서, 방법(110)은 원래 깊이 이미지에 에지 보존 필터(edge preserving filter)를 적용하는 단계를 포함한다. TOF 깊이 카메라로부터 수신되는 원래 깊이 이미지를 처리하는 경우에, 이런 원래 깊이 이미지는 TOF 깊이 카메라의 일반적인 부산물인 랜덤 노이즈를 포함할 수 있다. 반면, 구조광 깊이 카메라는 원래 더 부드러운(smooth) 신호를 갖고 있고, 이런 신호는 추가적으로 소프트웨어로도 필터링될 수 있다. 따라서, 이런 유형의 부드러운 신호를 에플레이션하기 위해서, TOF 깊이 카메라로부터 수신되는 원래 깊이 이미지는 중요한 깊이 특징을 잃지 않고도 TOF 깊이 지도로부터의 노이즈를 억제하도록 처리될 수 있다. 이를 위해, 에지 보존 필터를 사용하여 원래 깊이 이미지의 노이즈를 억제할 수 있다. Perona-Malik (Scale-Space and Edge Detection Using Anisotropic Diffusion, IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 12 n. 7, p. 629-639, 1990) 및 Weickert et al. (J. Weickert, B. M. ter Haar Romeny, M. A. Viergever, Efficient and reliable schemes for nonlinear diffusion filtering, IEEE Trans. Image Proc., v. 7 n. 3, pp. 398-410, 1998)의 저작물에 기술된 내용에 기반하는 비선형 편미분 방정식을 이용하는 등 적절한 임의의 접근 방법을 사용할 수 있다. 임계 파라미터의 에지는 카메라의 깊이 정확도의 상한(예컨대,  $K=10[\text{cm}]$ )으로 설정될 수 있다. 원래 깊이 이미지에 에지 보존 필터를 적용함으로써, 원래 깊이 이미지에서 객체들 간의 단절(discontinuities)을 잘 유지하면서도, 원래 깊이 이미지의 노이즈 레벨을 현저하게 떨어뜨릴 수 있다.

[0052] 도 6을 계속 참조하면, 120에서, 방법(110)은 (예컨대, TOF 분석에서 장면을 조명하기 위해 사용되는 적외선광 으로부터 측정되는 것과 같이) 조명 이미지(illumination image)에 중간값 필터(median filter)를 통과시킴으로써 신뢰도 지도(confidence map)를 구축하는 단계를 포함할 수 있다. 구조광 깊이 카메라의 깊이 이미지를 에플레이션하기 위해 TOF 깊이 카메라로부터 수신되는 원래 깊이 이미지를 처리하는 경우에, 구조광 깊이 카메라에서 발생하는 화소 무효화 현상(pixel invalidation phenomenon)을 에플레이션하기 위해 이 신뢰도 지도가 이용될 수 있다. 구조광 기술에서, 패턴 일치를 찾음으로써 깊이가 계산되고, 즉, 일치가 발견되면, 상대적으로 정확하게 그 깊이를 계산할 수 있는 반면, 일치가 발견되지 않으면, 깊이를 계산할 수 없으며, 그 화소에서의 측정도 유효하지 않게 된다. TOF 기술에서, 통상적으로 어디에서든지 깊이가 측정될 수 있지만, (예컨대, 조도 레벨에 따라) 정확도는 달라진다. 따라서, TOF 깊이 이미지의 조도 레벨로 깊이 측정이 어디에서 부정확한지(예컨대, 어두운 영역에서 신호에 노이즈가 많음)를 용이하게 예측하고, 따라서 구조광 깊이 이미지를 에플레이션할 수 있다. 이를 위해, 조명 이미지를 입력으로서 사용하여 신뢰도 지도를 구축할 수 있다. 신뢰도 지도는 먼저 조명 이미지에 중간값 필터를 통과시킴으로써 구축되어, 가외치(outliers)를 제거하고 노이즈를 억제할 수 있다. 이후에, 각각의 화소에 대해, 다음과 같은 소프트 임계 함수(soft threshold function)가 사용될 수 있다,

[0053] 
$$F_k(s) = \frac{1}{1 + (s/k)^2}$$

[0054] 여기서, 파라미터  $k$ 는 카메라 버전마다 바뀔 수 있다. 예를 들어, 몇몇 실시예에서  $k$ 는 60으로 설정될 수 있다.

[0055] 신뢰도 지도를 구축하는 것 외에도, 이미지 안에 있는 다른 영역들을 식별하는 것이 바람직하다. 따라서, 122에서, 방법(110)은 원래 깊이 이미지로부터 경사면 지도(oblique surface map)를 구축하는 단계를 포함할 수 있다. 구조광 깊이 카메라의 깊이 이미지를 예물레이션하기 위해 TOF 깊이 카메라로부터 수신되는 원래 깊이 이미지를 처리하는 경우에, 구조광 깊이 카메라에서 발생하는 화소 무효화 현상 역시도 패턴 일치 문제에 해당될 가능성이 있는 영역들을 식별함으로써 예물레이션될 수 있다. 구조광 조명 장치에 의해 투사되는 패턴이 경사진 객체(즉, 조명 광선에 대해 경사진 각도의 표면)에 생겨(smearred), 종종 그곳에서 패턴 일치가 실패하여 유효하지 않은 측정을 낳는다. 따라서, TOF 깊이 카메라로부터 수신되는 원래 깊이 이미지는 경사면 지도를 구축하여 경사면을 식별함으로써 처리될 수 있다. 이는 예컨대, 시야 및 해상도와 같은 카메라 파라미터를 사용하여 각각의 화소에 대한 세계 표면 각도(world surface angle)를 계산하고, 이를 가우시안 필터링(Gaussian filtering)으로 스무딩(smoothing)하는 것을 포함할 수 있다. 또한, 앞서 정의한 바와 같이,  $k=45$ 인  $F_k$ 와 같은 소프트 임계 함수를 사용할 수도 있다.

[0056] 124에서, 방법(110)은 신뢰도 지도 및 경사면 지도를 무효화 테스트 지도(invalidation testing map)로 통합하는 단계를 포함할 수 있다. 0과 1 사이의 값을 갖는 신뢰도 지도 및 경사면 지도의 경우에, 예컨대, 두 지도를 곱하고 임계값 0.5을 적용함으로써 두 지도들을 통합할 수 있다. 이 결과를 정규화하기(regularize) 위해 중간값 필터가 사용될 수도 있다. 이런 프로세스는 사전 결정된 범위 밖에 있는 화소들을 무효화하는 데 사용될 수 있다.

[0057] 도 6을 계속 참조하면, 126에서, 방법(110)은 원래 깊이 이미지에서 서로 다른 깊이 값을 갖고 있는 인접 화소 영역들 사이의 에지들을 강조하는 단계를 포함할 수 있다. TOF 깊이 카메라로부터 수신되는 원래 깊이 이미지를 처리하는 경우에, 이 기술로 계산되는 깊이 값은 화소 시야에서의 깊이의 평균이기 때문에, 이 TOF 깊이 카메라는 에지를 모호하게 하는 경향이 있다. 반면, 구조광 깊이 카메라의 에지에서의 깊이는 통상적으로 측정되지 않고 데이터가 종합되어, 객체들 간에 뚜렷한 전환이 일어나게 된다. 따라서, 구조광 깊이 카메라로부터의 깊이 이미지를 예물레이션하기 위해, TOF 깊이 카메라로부터의 원래 깊이 이미지는 서로 다른 깊이 값을 갖고 있는 인접 화소 영역들 간의 에지들을 강조하여 그 에지들을 선명하게 하도록 처리될 수 있다. 이런 에지 강조를 위해 임의의 적절한 접근 방법이 사용될 수 있으면, 이런 적절한 접근 방법의 하나가 다음과 같이 상세하게 설명된다.

[0058] 각 화소에 대해  $x$  방향의 전향 차분(forward difference)  $D+$  및 후향 차분(backward difference)  $D-$ 가 계산될 수 있다. 그 다음에,  $(\min(D+, D-) > 4cm)$ 이면  $Mask_x=1$ , 그 밖에는 0으로, 스텝 에지를 피하기 위해 마스크가 결정될 수 있다.  $Mask_y$ 를 계산하기 위해 유사한 계산이  $y$  방향에서 행해질 수 있다. 그 다음에,  $Mask_x$  또는  $Mask_y == 1$ 인 각각의 화소에 대해 3x3 이웃 화소 내에서 최대값이 취해진다.

[0059] 도 6을 계속 참조하면, 128에서, 방법(110)은 깊이 값을 양자화하는 단계를 포함할 수 있다. TOF 깊이 카메라로부터 수신되는 원래 깊이 이미지를 처리하는 경우에 있어서, 구조광 깊이 카메라로부터 수신되는 깊이 이미지를 예물레이션하기 위해 깊이 값이 양자화될 수 있다. 구조광 기술은 삼각 측량(triangulation)에 기반하여 깊이를 계산한다. 깊이는, 원래 센서 해상도가 유한하므로, 양자화되는 패턴 변위의 함수이다. TOF 기술에서, 깊이 측정과 원래 해상도는 관계가 없다. 따라서, 양자화 효과를 변환된 깊이 지도에 포함시키는 것이 바람직할 수 있다. 이는 임의의 적절한 접근 방법에 의해 실행될 수 있다. 이런 적절한 접근 방법의 하나로 구조광 깊이 카메라의 파라미터(예컨대, 시야, 원래 센서 해상도, 초점 거리, 센서와 조명 중심 간의 거리) 및 깊이 지도를 가지고, 삼각 측량 기반 카메라에서 생기는 것과 유사한 비선형 양자화 식을 구성하는 것을 들 수 있다. 예를 들어, 깊이  $D$ 의 함수로서 화소에서의 변환(translation)  $T$ 는 다음과 같이 정의될 수 있다,

[0060] 
$$T(D) = INT(focal\_length * (camera\_illum/D + TAN(angle))/pixel\_size + 0.5)$$

[0061] 여기서, focal\_length는 구조광 깊이 카메라의 초점 거리이며, pixel\_size는 센서 화소 크기이며, camera\_illum는 카메라 센서와 조명 중심 간의 거리이며, angle은 카메라 센서 중심에 수직인 선에 대한 객체의 각도이다. 그 다음에, 양자화  $Q$ 는 변환  $T(D)$ 의 함수로서 다음과 같이 기술될 수 있다,

[0062] 
$$Q(T) = camera\_illum / ((T) * pixel\_size / focal\_length - TAN(angle)) - camera\_illum / ((T + 1) * pixel\_size / focal\_length - TAN(angle))$$

[0063] 따라서, 깊이 지도는 구조광 기술의 삼각 측량 계산에 의해 생성되는 방식과 유사한 방식으로 깊이를 양자화하였다.

[0064] 도 6을 계속 참조하면, 방법(110)은 임계 크기보다 작은 객체에 속해 있는 화소들에 더 깊은 깊이 값을 할당하는 단계를 포함한다. TOF 깊이 카메라로부터 수신되는 원래 깊이 이미지를 처리하는 경우에, 구조광 깊이 카메라로부터 수신되는 깊이 이미지를 에플레이션하기 위해 작은 객체의 화소들에 더 깊은 깊이 값이 할당될 수 있다. 구조광 깊이 카메라는 너무 작은 객체들에는 잘 투사될 수 없는 패턴에 기반한다. 따라서, 이런 작은 객체들에는 보통 배경 깊이 값이 할당된다. 더 깊은 깊이 값은 임의의 적절한 방식으로 작은 객체들의 화소에 할당될 수 있다. 이런 접근 방식의 하나로 모폴로지컬 클로징(morphological closing)의 실행을 포함한다. 따라서, 구조화 요소(structuring element)는 해상도에 달려 있다. 예를 들어, 대략 3x3 크기의 공 요소(ball element)가 사용될 수 있다. 따라서, 깊이 이미지에서 관찰될 수 있는 가장 작은 객체 크기가 정정될 수 있고, 그 효과는 구조광 깊이 카메라에서 행해진 후처리와 유사하다.

[0065] 도 6을 계속 참조하면, 132에서, 방법(110)은 TOF 깊이 카메라로부터 가상적으로 떨어져 있는 가상 일루미네이터의 가상 벡터지 포인트로부터 가상적으로 가려진 화소들에 음영 화소 값을 할당하는 단계를 포함한다. TOF 깊이 카메라로부터 수신되는 원래 깊이 이미지를 처리하는 경우에, 구조광 깊이 카메라로부터 수신되는 깊이 이미지를 에플레이션하기 위해 음영 화소 값이 할당될 수 있다. 전술한 바와 같이, 구조광 깊이 카메라는 삼각 측량 원리에 의해 동작한다. 이와 같은 사실로 인해, 광원은 센서와 떨어져 있게 되고, “음영” 효과가 카메라의 센서에서 생성된다. 따라서, 도 7에 도시된 바와 같이, “음영진 화소”는 센서에서는 보이지만 광원의 위치에서는 직접적으로 보이지 않는 화소이다. 여기서, 제 1 객체(140)는 제 2 객체(142)가 발광기(light emitter)(144)로부터 직사 광선을 수신하는 것을 막는다. 따라서, 센서(150)는 제 1 객체(140)의 이미지(146) 및 제 2 객체(142)의 이미지(148)를 수신하는 것뿐만 아니라, 제 2 객체(142)의 음영진 화소들(152)도 수신한다.

[0066] 따라서, TOF 깊이 카메라로부터 수신되는 원래 깊이 이미지에 이런 “음영” 아티팩트를 에플레이션하는 것이 바람직할 수 있다. 센서와 주위 발광기 사이의 짧은 거리로 인해, TOF 깊이 카메라에서는 음영 효과가 이미 발생한다는 것을 알 수 있지만, 이는 카메라에서 정규화되어(normalized), 제작된 깊이 비디오에서는 보이지 않을 수 있다.

[0067] 음영을 종합하기 위한 예시적인 해결책으로 가벼운 알고리즘을 사용한다. 알고리즘은 모델링된 깊이 카메라에서 발광기가 있을 장소와 동일한 곳에 가상 카메라를 생성한다. 알고리즘은 원래 센서로부터의 깊이 샘플을 이 가상 센서에 맞게 변환한다. 이런 변환은 다음의 수식을 사용하여 실행될 수 있다,

$$f_{l_x} = \frac{imgWidthPix}{2 \cdot TAN(FOV_x)}$$

$$X_{world} = \frac{Z_{world}}{f_{l_x}} (X_{sensorImage} - principalPt_x)$$

$$X_{emitterImage} = \frac{f_{l_x}(X_{world} + distanceToEmitter)}{Z_{world}} + \frac{1}{2} imgWidthPix$$

[0068] 이 가상 센서에서 가려진 샘플들은 음영지게 된다. “발광기 x 이미지”의 각 행(row)이 스캔될 수 있고, 화소들이 이미 스캔된 값 중에 최대  $X_{emitterImage}$ 를 갖고 있지 않으면 음영지게 될 수 있다. 도 8은 음영진 화소들(152)이  $X_{world}$ 값의 하락으로 인해 음영진 것으로 간주되는 개념을 도시한다.

[0070] 깊이 측정의 부정확성으로 인한 일부 작은 변동은 극복하기 위해, 모폴로지컬 “오픈” 필터링을 “음영진 화소들”의 지도에 적용할 수 있다. 이런 단계는 작은 음영들을 제거하고, 예지들을 좀 더 사각형처럼 보이게 하여, 구조광 깊이 카메라로부터의 예지를 에플레이션한다.

[0071] 도 6을 계속 참조하면, 134에서, 방법(110)은 지원 유형을 갖는 에플레이션 깊이 이미지를 출력하는 단계를 포함한다. 이런 에플레이션 깊이 이미지는 깊이 카메라의 원래 깊이 이미지를 처리할 수 없는 깊이 이미지 컨슈머에 의해 사용될 수 있다. 상기의 예를 사용하면, 구조광 깊이 카메라로부터의 구조광 깊이 이미지를 처리하도록 설계된 애플리케이션이 TOF 깊이 카메라를 사용하여 측정된 원래 깊이 이미지에 기반하는 에플레이션 깊이 이미지를 수신하고 처리할 수 있다.

[0072] 도 9는 깊이 추상화층 내의 예시적인 이미지 처리 블록(160)을 도시하는 상위 레벨 구현 다이어그램이다. 도 9은 도 9의 이미지 처리 블록(160)을 보다 상세하게 도시한다. 도 9 및 10에 도시된 바와 같이, 이미지 처리 블록(160)은 입력으로서 실제 깊이 지도, 적외선 조명 이미지에 관한 정보, 및 깊이 카메라 파라미터를 수신할 수 있다. 이미지 처리 블록은 전술한 바와 같이, 깊이 이미지 컨슈머에 의해 사용될 수 있는 추상화된 에플레이션

깊이 이미지(162)를 출력할 수 있다.

- [0073] 도 1을 참조하며 아래에서 설명되는 바와 같이, 본 발명의 사상에서 벗어나지 않고 다양한 다른 컴퓨팅 시스템이 사용될 수 있다. 도 1을 참조하여 설명된 운영 환경이 예로서 제공되지만, 어떤 식으로든 제한하려는 의도는 없다. 반대로, 도시된 운영 환경은 본 발명의 범위를 벗어나지 않는 다양한 다른 운영 환경에 적용될 수 있는 일반적인 개념을 보여주기 위한 것이다. 마찬가지로, 도 2에 도시된 깊이 카메라 상호 호환 시스템의 개략적인 도시는 깊이 이미지 가상화 및 깊이 이미지 에플레이션을 설명하기 위한 간략화된 프레임워크를 제공하지만, 애플리케이션을 도면에 도시된 구성들만으로 제한하려고 의도한 것은 아니다. 오히려, 본원에서 설명된 방법 및 프로세스들은 여러 다른 유형의 컴퓨팅 시스템에 연관되어 있을 수 있다.
- [0074] 도 1은 게임 콘솔(12) 및 깊이 카메라(20)가 있는 깊이 카메라 상호 호환 시스템을 구현하도록 구성되는 비제한적인 컴퓨팅 시스템의 일례를 도시한다. 보다 일반적인 다른 예로, 도 11은 본원에서 설명된 바와 같이, 깊이 이미지 추상화 및/또는 깊이 이미지 에플레이션을 실행하는 컴퓨팅 시스템(170)을 개략적으로 도시한다. 컴퓨팅 시스템(170)은 특히 게임 콘솔, 개인용 컴퓨팅 시스템, 군사 트래킹 및/또는 타겟팅 시스템, 및 그린-스크린(green-screen)이나 모션-캡처 기능을 제공하는 캐릭터 획득 시스템을 포함하는, 그러나 이에 제한되지는 않는, 여러 다른 형태를 취할 수 있다.
- [0075] 컴퓨팅 시스템(170)은 로직 서브시스템(logic subsystem)(172), 로직 서브시스템에 접속되어 동작하는 데이터-홀딩 서브시스템(174), 디스플레이 서브시스템(176) 및/또는 깊이 이미지 프로듀서(178)를 포함할 수 있다. 컴퓨팅 시스템은 도 11에 도시되지 않은 컴포넌트를 선택적으로 포함할 수도 있으며, 및/또는 도 11에 도시된 일부 컴포넌트는 컴퓨팅 시스템에 포함되지 않은 주변 컴포넌트일 수 있다.
- [0076] 로직 서브시스템(172)은 하나 이상의 명령어를 실행하도록 구성된 하나 이상의 물리적 장치를 포함할 수 있다. 예를 들어, 로직 서브시스템은 하나 이상의 프로그램, 루틴, 객체, 컴포넌트, 데이터 구조, 애플리케이션 프로그래밍 인터페이스 또는 기타 논리적 구성의 일부인 하나 이상의 명령어를 실행하도록 구성될 수 있다. 이런 명령어는 작업을 수행하거나, 데이터 유형을 구현하거나, 하나 이상의 장치의 상태를 변환하거나, 서로 다른 컴퓨팅 객체들에 및/또는 그 객체들로부터 정보를 전달하거나, 또는 원하는 결과를 얻도록 구현될 수 있다. 로직 서브시스템은 소프트웨어 명령어를 실행하도록 구성되는 하나 이상의 프로세서를 포함할 수 있다. 이에 더하여 또는 이를 대신하여, 로직 서브시스템은 하드웨어 또는 펌웨어 명령어를 실행하도록 구성되는 하나 이상의 하드웨어 또는 펌웨어 로직 머신을 포함할 수도 있다. 로직 서브시스템은 몇몇 실시예에서 멀리 떨어져 있을 수 있는, 둘 이상의 장치에 분산되어 있는 개별 컴포넌트들을 선택적으로 포함할 수 있다.
- [0077] 데이터-홀딩 서브시스템(174)은 본원에서 설명된 방법 및 프로세스를 구현하기 위해 로직 서브시스템에 의해 실행될 수 있는 데이터 및/또는 명령어를 보유하도록 구성되는 하나 이상의 물리적, 비일시적인(non-transitory) 장치를 포함할 수 있다. 이들 방법 및 프로세스가 구현될 때, 데이터-홀딩 서브시스템(174)의 상태가 (예컨대, 다른 데이터를 보유하도록) 변환될 수 있다. 데이터-홀딩 서브시스템(174)은 이동식 매체 및/또는 빌트인 장치를 포함할 수 있다. 데이터-홀딩 서브시스템(174)은 특히 광 메모리 장치, 반도체 메모리 장치(예컨대, RAM, EEPROM, 플래시 등), 및/또는 자기 메모리 장치를 포함할 수 있다. 데이터-홀딩 서브시스템(174)은 휘발성, 비휘발성, 동적, 정적, 읽기/쓰기, 읽기-전용, 랜덤 액세스, 순차 액세스, 주소 지정 가능한 위치, 주소 지정 가능한 파일 및 주소 지정 가능한 컨텐츠 중 하나 이상의 특성이 있는 장치를 포함할 수 있다. 몇몇 실시예에서, 로직 서브시스템(172) 및 데이터-홀딩 서브시스템(174)은 ASIC(application specific integrated circuit) 또는 SOC(system on a chip)와 같이 하나 이상의 공통 장치에 통합될 수 있다.
- [0078] 용어 “모듈” 및 “엔진”은 하나 이상의 특정 기능을 실행하도록 구현되는 컴퓨팅 시스템(170)의 양태를 설명하기 위해 사용될 수 있다. 경우에 따라서, 이런 모듈이나 엔진은 데이터-홀딩 서브시스템(174)이 보유하는 명령어를 실행하는 로직 서브시스템(172)을 통해 인스턴스화될 수 있다(instantiated). 동일한 애플리케이션, 코드 블록, 객체, 루틴 및/또는 함수로부터 상이한 모듈 및/또는 엔진들이 인스턴스화될 수 있음을 이해하여야 한다. 마찬가지로, 경우에 따라서는 상이한 애플리케이션, 코드 블록, 객체, 루틴 및/또는 함수로부터 동일한 모듈 및/또는 엔진들이 인스턴스화될 수 있다. 예를 들어, 도 2를 참조하여 설명된 하나 이상의 모듈이 API로써 구현될 수 있다.
- [0079] 디스플레이 서브시스템(176)은 데이터-홀딩 서브시스템(174)이 보유하는 데이터의 비주얼 형상(visual representation)을 프리젠테이션하기 위해 사용될 수 있다. 본원에서 설명된 방법 및 프로세스가 데이터-홀딩 서브시스템이 보유한 데이터를 변경하여 데이터-홀딩 서브시스템의 상태를 바꾸는 것에 따라, 마찬가지로 디스플레이 서브시스템(176)의 상태도 기반 데이터(underlying data)의 변경을 시각적으로 나타내도록 바뀔 수

있다. 디스플레이 서브시스템(176)은 사실상 모든 유형의 기술을 이용하는 하나 이상의 디스플레이 장치를 포함할 수 있다. 이런 디스플레이 장치는 공유된 인클로저(enclosure)에서 로직 서브시스템(172) 및/또는 데이터-홀딩 서브시스템(174)과 결합될 수 있거나, 또는 이런 디스플레이 장치는 도 1에 도시된 바와 같이 주변 디스플레이 장치일 수 있다.

[0080] 컴퓨팅 시스템(170)은 또한 하나 이상의 타겟 및/또는 장면의 깊이 이미지를 얻도록 구성되는 깊이 이미지 프로듀서(178)를 포함한다. 깊이 이미지 프로듀서(178)는 임의의 적절한 기법(예컨대, 비행 시간, 구조광, 스테레오 이미지 등)을 통해 깊이 정보가 있는 비디오를 캡처하도록 구성될 수 있다. 따라서, 깊이 이미지 프로듀서(178)는 깊이 카메라, 비디오 카메라, 스테레오 카메라 및/또는 기타 적절한 캡처 장치를 포함할 수 있다. 상기의 도 4(a)-4(c)를 참조하여 설명한 바와 같이, 깊이 이미지 프로듀서는 실제 카메라 데이터를 깊이 이미지로 변환하기 위해 하나 이상의 온-카메라 프로세서 및/또는 오프-카메라 프로세서를 포함할 수 있다. 즉, 깊이 카메라는 하나 이상의 깊이 분석 기능을 실행하도록 구성되는 하나 이상의 온보드 처리 장치를 선택적으로 포함할 수 있다. 깊이 카메라는 이런 온보드 처리 로직의 업데이트를 용이하게 하는 펌웨어를 포함할 수도 있다.

[0081] 예를 들어, TOF 분석에서, 깊이 이미지 프로듀서(178)는 장면에 적외선을 방출한 후 센서를 사용하여 장면의 표면으로부터 후방 산란되는 빛을 탐지하도록 구성되는 TOF 카메라를 포함할 수 있다. 경우에 따라서는, 펄스 적외선(pulsed infrared light)을 사용하여, 나가는 광 펄스와 이에 대응하여 들어오는 광 펄스 간의 시간을 측정하고, 이를 사용하여 캡처 장치로부터 장면의 특정 위치까지의 물리적인 거리를 결정할 수 있다. 경우에 따라서는, 나가는 빛의 위상과 들어오는 빛의 위상을 비교하여 위상 변위(phase shift)를 결정할 수 있고, 이 위상 변위를 사용하여 캡처 장치로부터 장면의 특정 위치까지의 물리적인 거리를 결정할 수 있다.

[0082] 다른 예를 들면, TOF 분석법을 사용하여 셔터 광 펄스 이미징(shuttered light pulse imaging)과 같은 기법을 통해 시간에 따른 반사광의 강도를 분석함으로써, 캡처 장치로부터 장면의 특정 위치까지의 물리적인 거리를 간접적으로 결정할 수 있다.

[0083] 다른 예를 들면, 깊이 이미지 프로듀서(178)는 구조광을 사용하여 깊이 정보를 캡처할 수 있다. 이러한 분석법에서는, 패턴화된 광(즉, 그리드 패턴이나 스트라이프 패턴과 같이 공지된 패턴으로 디스플레이되는 광)이 장면에서 투사될 수 있다. 그 장면의 표면에서, 패턴이 그에 따라 변형될 수 있고(deformed), 이러한 패턴의 변형을 연구하여 캡처 장치로부터 장면의 특정 위치까지의 물리적인 거리를 결정할 수 있다.

[0084] 다른 예를 들면, 캡처 장치는 비주얼 스테레오 데이터를 얻기 위해 각기 다른 각도에서 장면을 보는 물리적으로 분리된 둘 이상의 카메라들을 포함할 수 있다. 이 경우에, 비주얼 스테레오 데이터를 분해하여 깊이 이미지를 생성할 수 있다. 다른 실시예에서, 깊이 이미지 프로듀서(178)는 다른 기술을 이용하여 깊이 값을 측정 및/또는 계산할 수 있다.

[0085] 몇몇 실시예에서, 둘 이상의 카메라가 깊이 이미지 프로듀서의 일부로서 통합될 수 있다. 예를 들어, 깊이 카메라 및 비디오 카메라(예컨대, RGB 비디오 카메라)가 깊이 이미지 프로듀서로 통합될 수 있다. 비디오 카메라가 사용되면, 이는 타겟 트래킹 데이터, 장면 분석의 오류 정정을 위한 확인 데이터, 이미지 캡처, 안면 인식, 고정밀 손가락(또는 기타 작은 특징) 트래킹, 빛 감지, 및/또는 그 밖의 기능을 제공할 수 있다.

[0086] 또한, 상기의 설명은 하나의 깊이 카메라의 사용/에플리케이션에 중점을 두고 있지만, 전술한 호환성 기술들을 사용하여 둘 이상의 깊이 카메라를 동시에 사용/에플리케이션할 수 있음을 이해할 것이다. 예를 들어, 두 개의 카메라를 사용하여 인접 장면을 볼 수 있고, API는 양쪽 카메라로부터의 정보를 효과적으로 결합하여 더 넓은 시야를 가진 하나의 카메라를 에플리케이션할 수 있다. 다른 예로, 하나의 와이드 앵글 카메라를 사용할 수 있고, API는 다른 방향에서 보는 좁은 시야를 가진 두 개의 카메라에 의해 생성되는 것처럼 두 개의 분리된 깊이 이미지를 생성할 수 있다.

[0087] 몇몇 실시예에서, 둘 이상의 깊이 카메라를 사용하여 각기 다른 팬티지 포인트에서 동일한 장면을 볼 수 있다. 이 같은 경우에, API는 양쪽 카메라로부터의 정보를 효과적으로 결합하여 깊이 이미지 컨슈머에게 더 많고/나은 3D 데이터를 명료한 방식으로 제공할 수 있다.

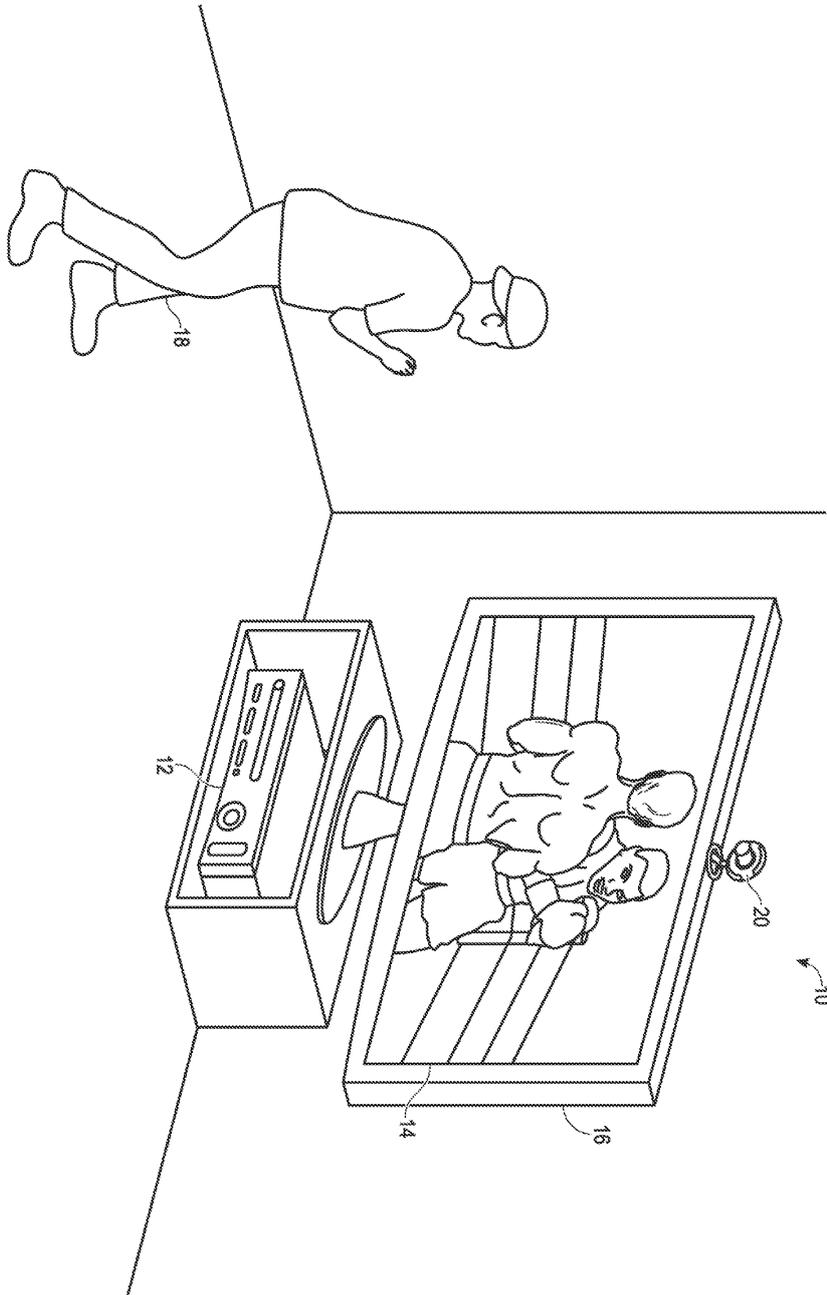
[0088] 본원에 기술된 구성 및/또는 접근 방법은 예시적인 것으로, 다양한 변형이 가능하기 때문에, 이러한 특정 실시예들이나 예시들이 제한적인 의미로 여겨지는 것은 아님을 이해할 것이다. 본원에 기술된 특정 루틴 또는 방법들은 하나 이상의 많은 처리 전략을 나타낸다. 따라서, 도시된 다양한 행위들은 도시된 순서대로, 다른 순서로, 동시에, 또는 경우에 따라서는 생략되어 실행될 수 있다. 마찬가지로, 전술한 프로세스의 순서 또한 변경될 수 있다.

[0089]

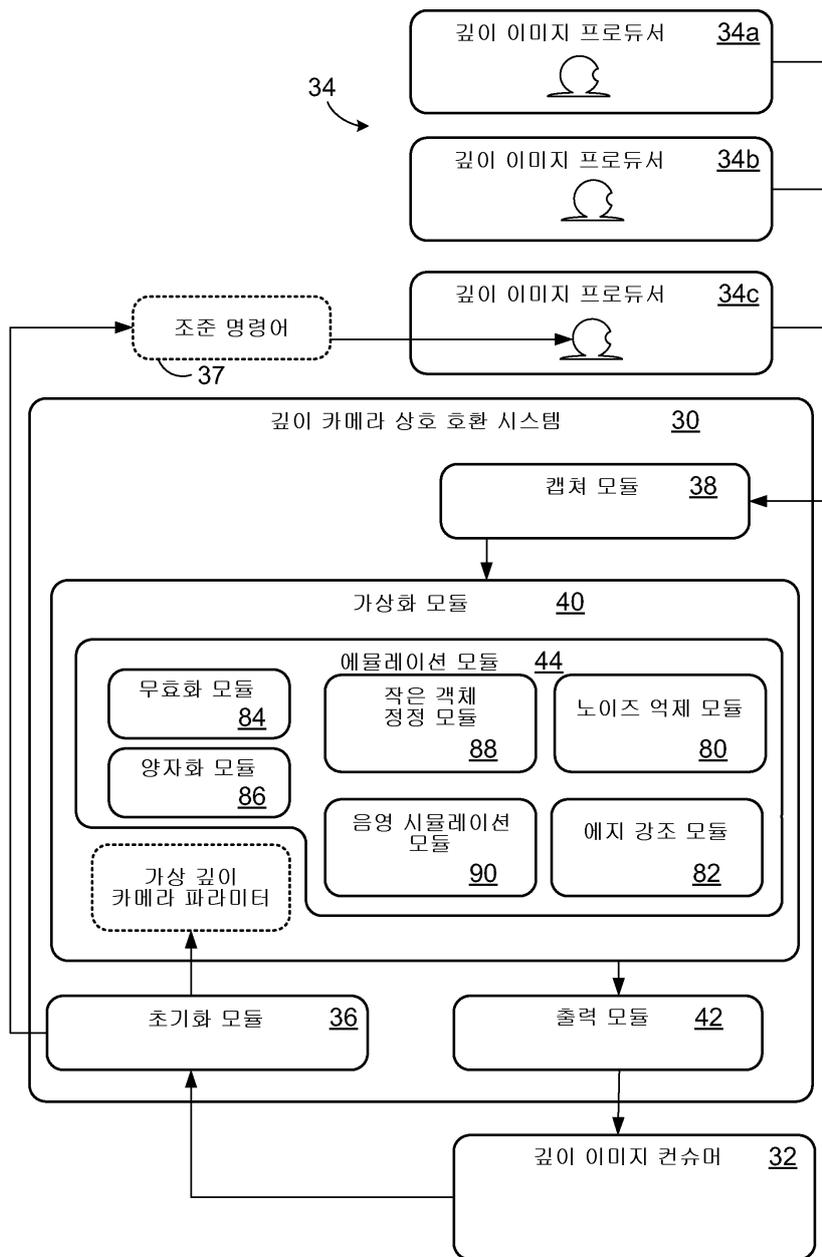
본 개시의 대상은, 본원에 기술된 다양한 프로세스, 시스템 및 구성의 신규하고 자명하지 않은 모든 콤비네이션 및 서브 콤비네이션, 및 기타 특징, 기능, 행위 및/또는 특성들뿐만 아니라 그들의 임의의 모든 등가물을 포함한다.

도면

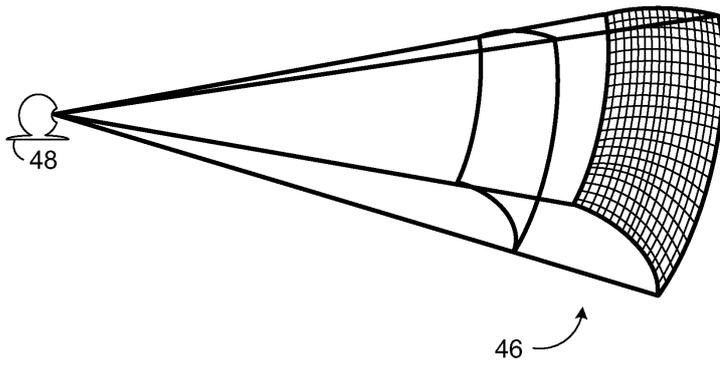
도면1



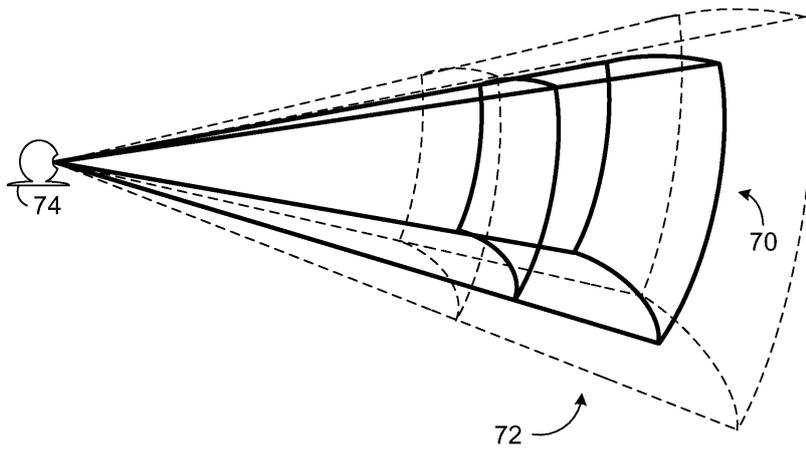
도면2



도면3

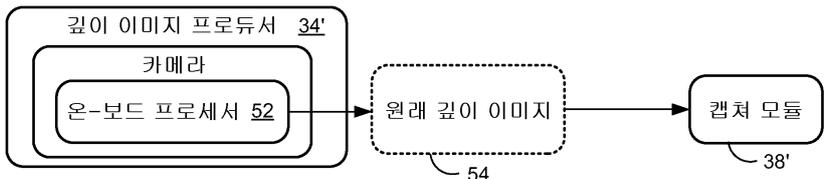


(a)

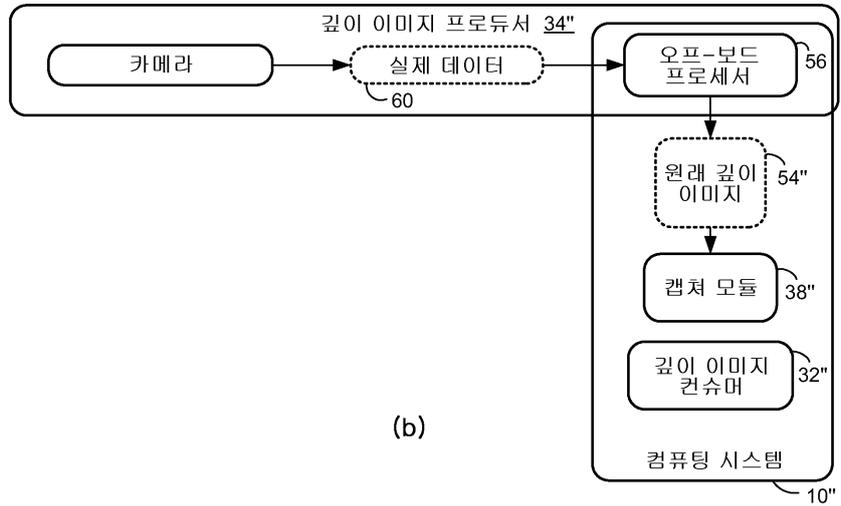


(b)

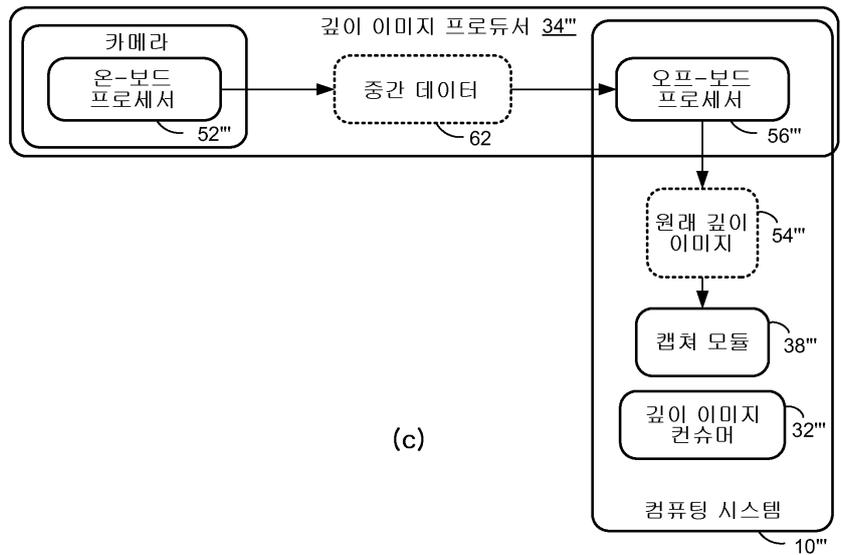
도면4



(a)

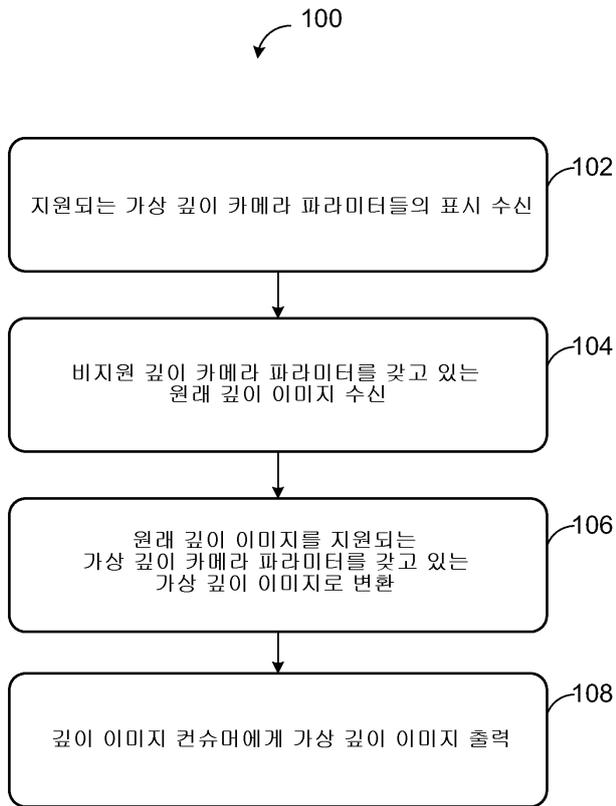


(b)

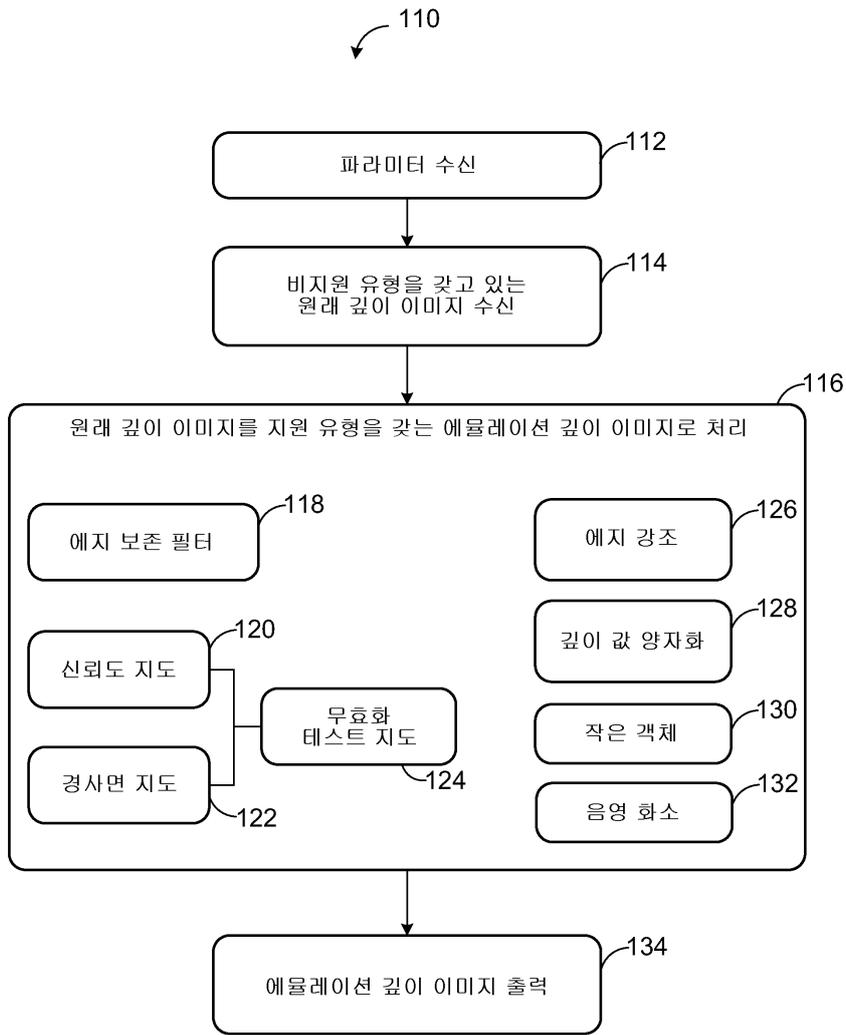


(c)

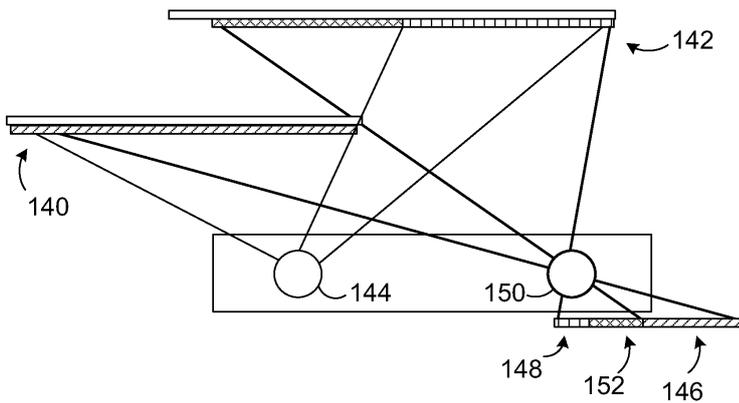
도면5



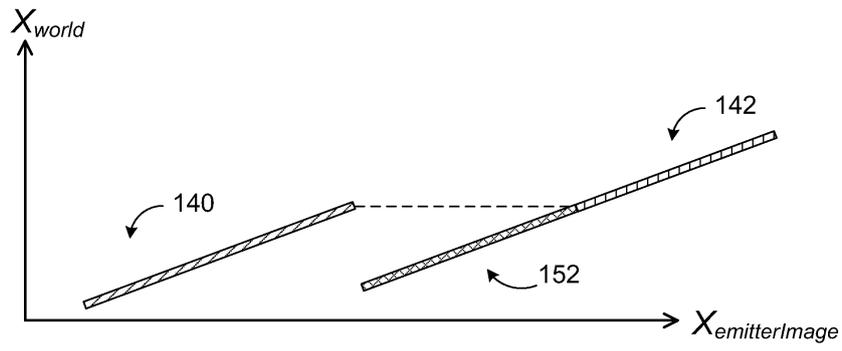
도면6



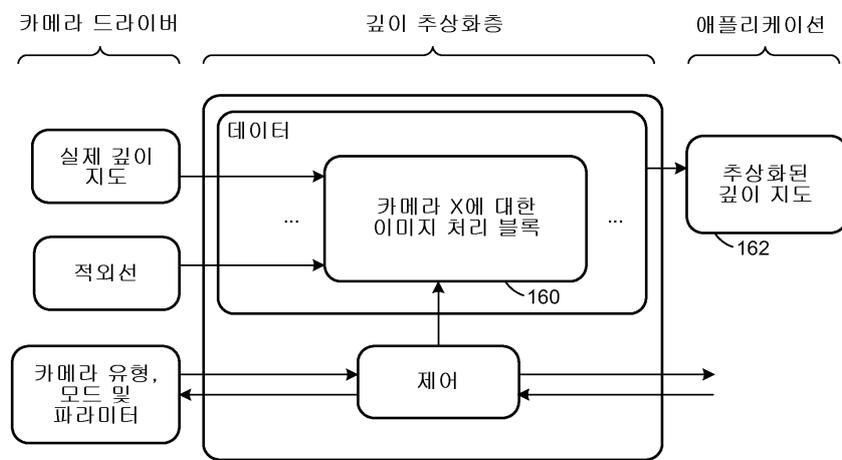
도면7



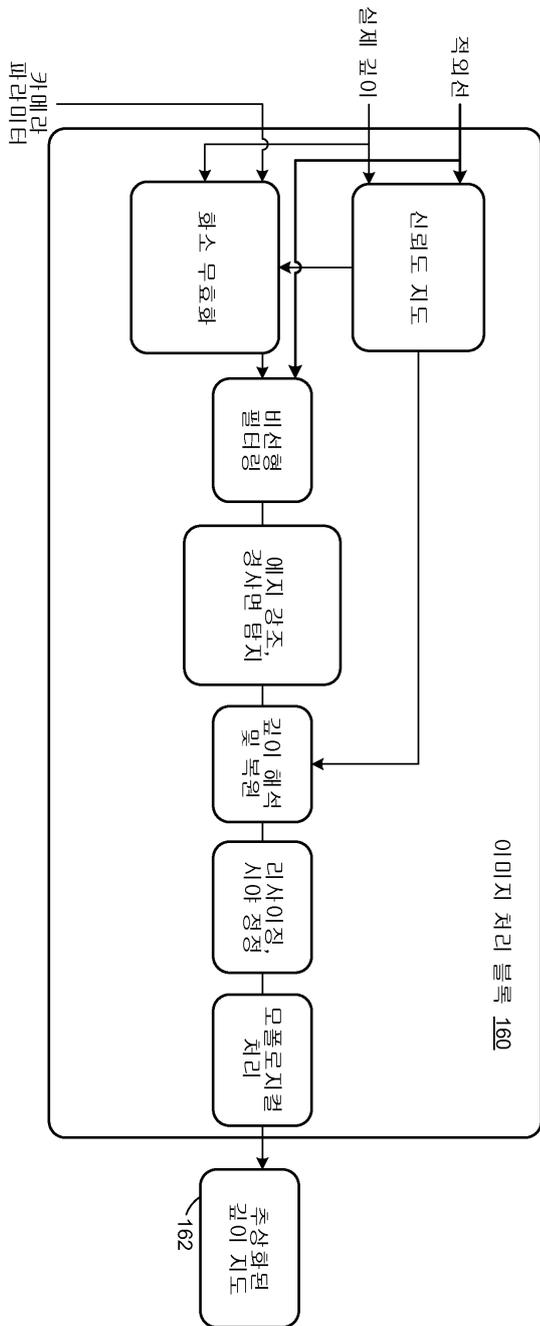
도면8



도면9



도면10



도면11

