



등록특허 10-2126610



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2020년06월25일
(11) 등록번호 10-2126610
(24) 등록일자 2020년06월18일

- (51) 국제특허분류(Int. Cl.)
H04N 19/27 (2014.01) *G06T 9/00* (2019.01)
H04N 19/23 (2014.01) *H04N 19/30* (2014.01)
- (52) CPC특허분류
H04N 19/27 (2015.01)
G06T 9/001 (2019.05)
- (21) 출원번호 10-2017-7004391
- (22) 출원일자(국제) 2015년09월04일
심사청구일자 2020년01월31일
- (85) 번역문제출일자 2017년02월16일
- (65) 공개번호 10-2017-0056516
- (43) 공개일자 2017년05월23일
- (86) 국제출원번호 PCT/US2015/048569
- (87) 국제공개번호 WO 2016/040161
국제공개일자 2016년03월17일
- (30) 우선권주장
62/048,041 2014년09월09일 미국(US)
14/845,076 2015년09월03일 미국(US)

(56) 선행기술조사문헌

JP03253190 A

H. Musmann, et al. Object-oriented analysis-synthesis coding of moving images. Signal Processing: Image Communication. Oct 1989, Vol.1, No.2, pp.117-138

C. Feller, et al. Model-based video compression for real world data. 2013 IEEE 3rd Int'l Conf. on Consumer Electronics. Sep. 2013, pp.210-214

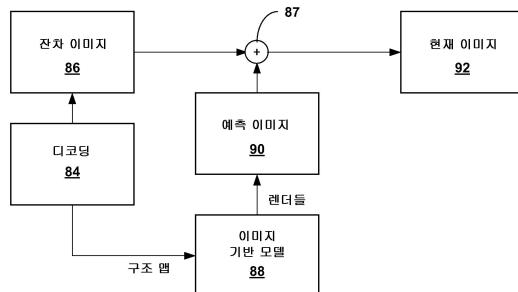
전체 청구항 수 : 총 23 항

심사관 : 이상래

(54) 발명의 명칭 비디오 코딩을 위한 동시 로컬리제이션 및 맵핑

(57) 요약

현재 이미지 또는 이전에 인코딩 또는 디코딩된 이미지의 3차원 구조의 근사화를 제공하는 프록시 지오메트리에 복합 이미지를 텍스처 맵핑하는 것으로부터 예측 이미지가 형성되는 비디오 인코딩 및 디코딩 기법들이 설명된다. 예측 이미지와 현재 이미지 사이의 잔차는 현재 이미지를 인코딩 또는 디코딩하는데 이용된다.

대 표 도

(52) CPC특허분류

H04N 19/23 (2015.01)

H04N 19/30 (2015.01)

명세서

청구범위

청구항 1

비디오 데이터를 디코딩하는 방법으로서,

복합 이미지와 장면 구조 맵에 기초하여 합성 이미지를 생성하는 단계로서, 상기 복합 이미지는 이전에 디코딩되었던 하나 이상의 이미지들로부터 구성되고, 상기 장면 구조 맵은 상기 비디오 데이터의 현재 이미지의 장면 구조 맵 또는 이전에 디코딩되었던 상기 비디오 데이터의 이미지의 장면 구조 맵을 포함하고, 상기 장면 구조 맵은 상기 현재 이미지 또는 이전에 디코딩되었던 상기 이미지 내에서, 포인트들의 포지션들 및 상대 깊이를 표시하는 3차원 포인트들에 대한 좌표 값을 포함하고, 상기 합성 이미지를 생성하는 단계는 상기 합성 이미지와 상기 현재 이미지에 대한 카메라 포지션 및 배향이 동일하도록 상기 합성 이미지를 렌더링하기 위해 상기 현재 이미지의 카메라 포지션 및 배향 정보를 사용하는 단계를 포함하고, 상기 합성 이미지를 생성하는 단계는:

프록시 지오메트리를 형성하기 위해 상기 장면 구조 맵의 포인트들을 상호접속하는 단계;

이미지-기반 모델을 형성하기 위해 상기 프록시 지오메트리에 상기 복합 이미지를 텍스처 맵핑하는 단계; 및

상기 합성 이미지를 생성하기 위해 상기 이미지-기반 모델을 렌더링하는 단계를 더 포함하는, 상기 합성 이미지를 생성하는 단계;

잔차 이미지를 결정하는 단계로서, 상기 잔차 이미지는 상기 현재 이미지와 상기 합성 이미지 사이의 차이를 나타내고, 상기 잔차 이미지를 결정하는 단계는 상기 현재 이미지의 하나 이상의 부분들이 배경의 정적 부분을 포함하는 것에 기초하여 상기 잔차 이미지를 결정하는 단계를 포함하는, 상기 잔차 이미지를 결정하는 단계; 및

상기 합성 이미지와 상기 잔차 이미지에 기초하여 상기 현재 이미지를 재구성하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 잔차 이미지를 결정하는 단계는, 상기 현재 이미지의 상기 잔차 이미지를 수신하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 3

제 1 항에 있어서,

상기 텍스처 맵핑하는 단계는, 하나 이상의 이전에 디코딩된 이미지들의 카메라 포지션 및 배향 정보에 기초하여 상기 프록시 지오메트리에 상기 복합 이미지를 텍스처 맵핑하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 4

제 1 항에 있어서,

상기 현재 이미지의 상기 장면 구조 맵의 정보, 또는 상기 현재 이미지의 상기 장면 구조 맵과 상기 이전에 디코딩되었던 이미지의 상기 장면 구조 맵 사이의 차이를 나타내는 정보 중 하나를 수신하는 단계; 및

수신된 상기 정보에 기초하여 상기 장면 구조 맵을 생성하는 단계를 더 포함하고,

상기 합성 이미지를 생성하는 단계는, 상기 복합 이미지와, 생성된 상기 장면 구조 맵에 기초하여 상기 합성 이미지를 생성하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 5

제 1 항에 있어서,

상기 복합 이미지를 구성하는데 이용되는 하나 이상의 이전에 디코딩된 이미지들과 상기 현재 이미지에 대한 카메라 포지션 및 카메라 배향 중 일방 또는 양방을 수신하는 단계를 더 포함하고,

상기 합성 이미지를 생성하는 단계는, 상기 하나 이상의 이전에 디코딩된 이미지들과 상기 현재 이미지의 상기 카메라 포지션 및 상기 카메라 배향 중 일방 또는 양방에 기초하여 상기 합성 이미지를 생성하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 6

비디오 데이터를 인코딩하는 방법으로서,

복합 이미지와 장면 구조 맵에 기초하여 합성 이미지를 생성하는 단계로서, 상기 복합 이미지는 이전에 인코딩되었던 하나 이상의 이미지들로부터 구성되고, 상기 장면 구조 맵은 상기 비디오 데이터의 현재 이미지의 장면 구조 맵 또는 이전에 인코딩되었던 상기 비디오 데이터의 이미지의 장면 구조 맵을 포함하고, 상기 장면 구조 맵은 상기 현재 이미지 또는 이전에 인코딩되었던 상기 이미지 내에서, 포인트들의 포지션들 및 상대 깊이를 표시하는 3차원 포인트들에 대한 좌표 값을 포함하고, 상기 합성 이미지를 생성하는 단계는 상기 합성 이미지와 상기 현재 이미지에 대한 카메라 포지션 및 배향이 동일하도록 상기 합성 이미지를 렌더링하기 위해 상기 현재 이미지의 카메라 포지션 및 배향 정보를 사용하는 단계를 포함하고, 상기 합성 이미지를 생성하는 단계는:

프록시 지오메트리를 형성하기 위해 상기 장면 구조 맵의 포인트들을 상호접속하는 단계;

이미지-기반 모델을 형성하기 위해 상기 프록시 지오메트리에 상기 복합 이미지를 텍스처 맵핑하는 단계; 및

상기 합성 이미지를 생성하기 위해 상기 이미지-기반 모델을 렌더링하는 단계를 더 포함하는, 상기 합성 이미지를 생성하는 단계;

상기 합성 이미지와 상기 현재 이미지에 기초하여 잔차 이미지를 결정하는 단계로서, 상기 잔차 이미지는 상기 현재 이미지와 상기 합성 이미지 사이의 차이를 나타내고, 상기 잔차 이미지를 결정하는 단계는 상기 현재 이미지의 하나 이상의 부분들이 배경의 정적 부분을 포함하는 것에 기초하여 상기 잔차 이미지를 결정하는 단계를 포함하는, 상기 잔차 이미지를 결정하는 단계; 및

상기 비디오 데이터의 상기 현재 이미지를 인코딩하기 위해 상기 잔차 이미지를 나타내는 정보를 출력하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 7

제 6 항에 있어서,

상기 복합 이미지를 구성하는데 이용되는 하나 이상의 이전에 인코딩된 이미지들에 대한 카메라 포지션 및 배향 정보와, 상기 현재 이미지에 대한 카메라 포지션 및 배향 정보를 결정하는 단계를 더 포함하고,

상기 텍스처 맵핑하는 단계는, 상기 하나 이상의 이전에 인코딩된 이미지들의 상기 카메라 포지션 및 배향 정보에 기초하여 상기 프록시 지오메트리에 상기 복합 이미지를 텍스처 맵핑하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 8

제 7 항에 있어서,

동시 로컬리제이션 및 맵핑 (simultaneous localization and mapping; SLAM) 기법들을 이용하여 상기 현재 이미지의 상기 카메라 포지션을 결정하는 단계를 더 포함하고,

상기 하나 이상의 이전에 인코딩된 이미지들에 대한 상기 카메라 포지션을 결정하는 단계가, 상기 SLAM 기법들을 이용하여 상기 하나 이상의 이전에 인코딩된 이미지들에 대한 상기 카메라 포지션을 결정하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 9

제 6 항에 있어서,

동시 로컬리제이션 및 맵핑 (SLAM) 기법들을 이용하여 상기 장면 구조 맵을 생성하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 10

제 6 항에 있어서,

상기 현재 이미지의 상기 장면 구조 맵의 정보, 또는 상기 현재 이미지의 상기 장면 구조 맵과 상기 이전에 인코딩되었던 이미지의 상기 장면 구조 맵 사이의 차이를 나타내는 정보 중 하나를 결정하는 단계; 및

결정된 상기 정보를 출력하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 11

제 6 항에 있어서,

상기 합성 이미지는 제 1 합성 이미지를 포함하고,

상기 방법은:

상기 복합 이미지를 구성하는데 이용되는 하나 이상의 이전에 인코딩된 이미지들에 대한 그리고 상기 현재 이미지에 대한 카메라 포지션 및 카메라 배향 중 일방 또는 양방을 출력하는 단계를 더 포함하고,

상기 카메라 포지션 및 상기 카메라 배향 중 일방 또는 양방은 디코더 프로세서에 의해, 상기 제 1 합성 이미지와 동일한 제 2 합성 이미지를 생성하는데 이용되는, 비디오 데이터를 인코딩하는 방법.

청구항 12

제 6 항에 있어서,

상기 현재 이미지의 전경의 비정적 부분 및 상기 현재 이미지의 상기 배경의 정적 부분을 결정하는 단계를 더 포함하고,

상기 잔차 이미지를 출력하는 단계가, 상기 현재 이미지의 상기 전경의 비정적 부분에 대한 잔차 데이터를 포함하는 제 2 계층과는 상이한 제 1 계층에서 상기 잔차 이미지를 출력하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 13

비디오 데이터를 코딩하는 디바이스로서,

복합 이미지를 구성하는데 이용되고 이전에 코딩되었던 하나 이상의 이미지들을 저장하도록 구성되는 비디오 메모리; 및

코더 프로세서를 포함하고,

상기 코더 프로세서는:

상기 복합 이미지와 장면 구조 맵에 기초하여 합성 이미지를 생성하는 것으로서, 상기 장면 구조 맵은 상기 비디오 데이터의 현재 이미지의 장면 구조 맵 또는 이전에 코딩되었던 상기 비디오 데이터의 이미지의 장면 구조 맵을 포함하고, 상기 장면 구조 맵은 상기 현재 이미지 또는 이전에 코딩되었던 상기 이미지 내에서, 포인트들의 포지션들 및 상대 깊이들을 표시하는 3차원 포인트들에 대한 좌표 값을 포함하는, 상기 합성 이미지를 생성하고;

상기 현재 이미지의 잔차 이미지에 기초하여 상기 현재 이미지를 코딩하는 것으로서, 상기 잔차 이미지는 상기 현재 이미지와 상기 합성 이미지 사이의 차이를 나타내고, 상기 잔차 이미지는 상기 현재 이미지의 하나 이상의 부분들이 배경의 정적 부분을 포함하는 것에 기초하는, 상기 현재 이미지를 코딩하도록 구성되고,

상기 코더 프로세서는 상기 합성 이미지를 생성하기 위해, 상기 합성 이미지와 상기 현재 이미지에 대한 카메라 포지션 및 배향이 동일하도록 상기 합성 이미지를 렌더링하기 위해 상기 현재 이미지의 카메라 포지션 및 배향 정보를 사용하도록 구성되고, 상기 코더 프로세서는 상기 합성 이미지를 생성하기 위해 또한:

프록시 지오메트리를 형성하기 위해 상기 장면 구조 맵의 포인트들을 상호접속하고;

이미지-기반 모델을 형성하기 위해 상기 프록시 지오메트리에 상기 복합 이미지를 텍스처 맵핑하고; 그리고

상기 합성 이미지를 생성하기 위해 상기 이미지-기반 모델을 렌더링하도록 구성되는, 비디오 데이터를 코딩하는 디바이스.

청구항 14

제 13 항에 있어서,

상기 코더 프로세서는 디코더 프로세서를 포함하고,

상기 디코더 프로세서는 상기 현재 이미지의 상기 잔차 이미지를 수신하도록 구성되고, 상기 현재 이미지를 코딩하기 위해, 상기 디코더 프로세서는 상기 합성 이미지와 상기 잔차 이미지에 기초하여 상기 현재 이미지를 재구성하는 것에 의해 상기 현재 이미지를 디코딩하도록 구성되는, 비디오 데이터를 코딩하는 디바이스.

청구항 15

제 14 항에 있어서,

상기 디코더 프로세서는:

상기 현재 이미지의 상기 장면 구조 맵의 정보, 또는 상기 현재 이미지의 상기 장면 구조 맵과 이전에 코딩된 이미지의 상기 장면 구조 맵 사이의 차이를 나타내는 정보 중 하나를 수신하고; 그리고

수신된 상기 정보에 기초하여 상기 장면 구조 맵을 생성하도록 구성되고,

상기 합성 이미지를 생성하기 위해, 상기 디코더 프로세서는 상기 복합 이미지와, 생성된 상기 장면 구조 맵에 기초하여 상기 합성 이미지를 생성하도록 구성되는, 비디오 데이터를 코딩하는 디바이스.

청구항 16

제 14 항에 있어서,

상기 디코더 프로세서는:

상기 복합 이미지를 구성하는데 이용되는 하나 이상의 이전에 코딩된 이미지들과 상기 현재 이미지에 대한 카메라 포지션 및 카메라 배향 중 일방 또는 양방을 수신하도록 구성되고,

상기 합성 이미지를 생성하기 위해, 상기 디코더 프로세서는 상기 하나 이상의 이전에 코딩된 이미지들과 상기 현재 이미지의 상기 카메라 포지션 및 상기 카메라 배향 중 일방 또는 양방에 기초하여 상기 합성 이미지를 생성하도록 구성되는, 비디오 데이터를 코딩하는 디바이스.

청구항 17

제 13 항에 있어서,

상기 코더 프로세서는 인코더 프로세서를 포함하고,

상기 현재 이미지를 코딩하기 위해, 상기 인코더 프로세서는:

상기 합성 이미지와 상기 현재 이미지에 기초하여 상기 잔차 이미지를 결정하고; 그리고

상기 비디오 데이터의 상기 현재 이미지를 인코딩하기 위해 상기 잔차 이미지를 나타내는 정보를 출력하도록 구성되는, 비디오 데이터를 코딩하는 디바이스.

청구항 18

제 17 항에 있어서,

상기 인코더 프로세서는 동시 로컬리제이션 및 맵핑 (SLAM) 기법들을 이용하여 상기 장면 구조 맵을 생성하도록 구성되는, 비디오 데이터를 코딩하는 디바이스.

청구항 19

제 17 항에 있어서,

상기 인코더 프로세서는:

상기 현재 이미지의 상기 장면 구조 맵의 정보, 또는 상기 현재 이미지의 상기 장면 구조 맵과 이전에 코딩된 이미지의 상기 장면 구조 맵 사이의 차이를 나타내는 정보 중 하나를 결정하고; 그리고

결정된 상기 정보를 출력하도록 구성되는, 비디오 데이터를 코딩하는 디바이스.

청구항 20

제 17 항에 있어서,

상기 합성 이미지는 제 1 합성 이미지를 포함하고,

상기 인코더 프로세서는:

동시 로컬리제이션 및 맵핑 (SLAM) 기법들을 이용하여 하나 이상의 이전에 코딩된 이미지들에 대한 카메라 포지션을 결정하고; 그리고

상기 복합 이미지를 구성하는데 이용되는 하나 이상의 이전에 코딩된 이미지들에 대한 그리고 상기 현재 이미지에 대한 카메라 포지션 및 카메라 배향 중 일방 또는 양방을 출력하도록 구성되고,

상기 카메라 포지션 및 상기 카메라 배향 중 일방 또는 양방은 디코더 프로세서에 의해, 상기 제 1 합성 이미지와 동일한 제 2 합성 이미지를 생성하는데 이용되는, 비디오 데이터를 코딩하는 디바이스.

청구항 21

제 13 항에 있어서,

상기 합성 이미지를 생성하기 위해, 상기 코더 프로세서는 그래픽 프로세싱 유닛 (graphics processing unit; GPU) 을 포함하고, 상기 GPU 는 상기 합성 이미지를 생성하도록 구성되는, 비디오 데이터를 코딩하는 디바이스.

청구항 22

제 13 항에 있어서,

상기 텍스처 맵핑하기 위해, 상기 코더 프로세서는 하나 이상의 이전에 코딩된 이미지들의 카메라 포지션 및 배향 정보에 기초하여 상기 프록시 지오메트리에 상기 복합 이미지를 텍스처 맵핑하도록 구성되는, 비디오 데이터를 코딩하는 디바이스.

청구항 23

명령들을 저장한 비일시적 컴퓨터 판독가능 저장 매체로서,

상기 명령들은 실행될 때, 비디오 데이터를 코딩하기 위한 디바이스에 대한 하나 이상의 프로세서들로 하여금:

복합 이미지와 장면 구조 맵에 기초하여 합성 이미지를 생성하게 하는 것으로서, 상기 복합 이미지는 이전에 코딩되었던 하나 이상의 이미지들로부터 구성되고, 상기 장면 구조 맵은 상기 비디오 데이터의 현재 이미지의 장면 구조 맵 또는 이전에 코딩되었던 상기 비디오 데이터의 이미지의 장면 구조 맵을 포함하고, 상기 장면 구조 맵은 상기 현재 이미지 또는 이전에 코딩되었던 상기 이미지 내에서 3차원 포인트들에 대한, 포인트들의 포지션들 및 상대 깊이를 표시하는 좌표 값들을 포함하는, 상기 합성 이미지를 생성하게 하고; 그리고

상기 현재 이미지의 잔차 이미지에 기초하여 상기 현재 이미지를 코딩하게 하는 것으로서, 상기 잔차 이미지는 상기 현재 이미지와 상기 합성 이미지 사이의 차이를 나타내고, 상기 잔차 이미지는 상기 현재 이미지의 하나 이상의 부분들이 배경의 정적 부분을 포함하는 것에 기초하는, 상기 현재 이미지를 코딩하게 하고,

상기 하나 이상의 프로세서들로 하여금 상기 합성 이미지를 생성하게 하는 명령들은, 상기 하나 이상의 프로세서들로 하여금 상기 합성 이미지와 상기 현재 이미지에 대한 카메라 포지션 및 배향이 동일하도록 상기 합성 이미지를 렌더링하기 위해 상기 현재 이미지의 카메라 포지션 및 배향 정보를 사용하게 하는 명령들을 포함하고,

상기 하나 이상의 프로세서들로 하여금 상기 합성 이미지를 생성하게 하는 명령들은 상기 하나 이상의 프로세서 들로 하여금:

프록시 지오메트리를 형성하기 위해 상기 장면 구조 맵의 포인트들을 상호접속하게 하고;

이미지-기반 모델을 형성하기 위해 상기 프록시 지오메트리에 상기 복합 이미지를 텍스처 맵핑하게 하고; 그리고

상기 합성 이미지를 생성하기 위해 상기 이미지-기반 모델을 렌더링하게 하는 명령들을 포함하는, 명령 들을 저장한 비일시적 컴퓨터 관독가능 저장 매체.

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

발명의 설명

기술 분야

[0001] 본 출원은 2014년 9월 9일 출원된 미국 특허 가출원 제62/048,041호의 이익을 우선권으로 주장하며, 그 전체 내용을 참조로서 포함한다.

기술 분야

[0003] 본 개시는 비디오 코딩에 관한 것이다.

배경 기술

[0004] 디지털 비디오 능력들은, 디지털 텔레비전, 디지털 다이렉트 브로드캐스트 시스템들, 무선 브로드캐스트 시스템 들, PDA들(personal digital assistants), 랩탑 또는 데스크탑 컴퓨터들, 태블릿 컴퓨터들, 이북 리더들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게임용 디바이스들, 비디오 게임 콘솔, 셀룰러 또는 위성 라디오 텔레폰들, 소위 "스마트폰들", 화상 원격회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함하는 광범위한 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Part 10, AVC (Advanced Video Coding), 현재 개발 하에 있는 HEVC (High Efficiency Video Coding)에 의해 정의된 표준들, 및 이러한 표준들의 확장들에서 설명된 바와 같은 비디오 코딩 기법들을 구현한다. 비디오 디바이스들은 이러한 비디오 코딩 기법들을 구현함으로써 좀더 효율적으로 디지털 비디오 정보를 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

발명의 내용

해결하려는 과제

과제의 해결 수단

[0005]

본 개시는 현재 이미지를 인코딩 또는 디코딩하기 위하여 현재 이미지의 합성 이미지 및 장면 구조 맵을 이용하여 비디오 코딩하기 위한 기법들을 설명한다. 장면 구조 맵은 프록시 지오메트리를 형성하기 위해 복수의 다각형들 (예를 들어, 삼각형들) 을 이용하여 상호접속되는 이미지의 포인트들의 밀도가 희소한 (sparsely populated) 세트를 포함할 수도 있다. 장면 구조 맵은 현재 이미지 또는 이전에 코딩된 (예를 들어, 인코딩되거나 또는 디코딩된) 이미지의 포인트들의 밀도가 희소한 세트를 포함할 수도 있다. 합성 이미지는 프록시 지오메트리에 매핑되는 하나 이상의 이전에 코딩된 이미지들 (예를 들어, 프록시 지오메트리에 오버레이된 이전에 코딩된 이미지들) 이다.

[0006]

인코더 프로세서는 합성 이미지와 현재 이미지 사이의 차이를 결정하고, 결과적인 잔차 이미지를 시그널링한다 (예를 들어, 잔차 이미지를 나타내는 정보를 출력한다). 또한, 인코더 프로세서는 장면 구조 맵의 정보를 때때로 시그널링하고 각각의 이미지에 대한 카메라 배향 정보 (카메라 포즈 정보) 와 카메라 포지션을 시그널링 할 수도 있다. 카메라 포즈 정보의 시그널링은 모든 경우에 필수적인 것은 아닐 수도 있다. 디코더 프로세서는 장면 구조 맵의 정보를 수신한다. 디코더 프로세서는 프록시 지오메트리와 하나 이상의 이전에 디코딩된 이미지들을 형성하기 위해 장면 구조 맵을 이용하여 합성 이미지를 재구성한다. 디코더 프로세서는 또한 (예를 들어, 인코더 프로세서로부터의 잔차 이미지를 나타내는 정보에 기초하여) 잔차 데이터, 잠재적으로 카메라 포즈 정보를 결정하고, 잔차 이미지, 잠재적으로 카메라 포즈 정보 및 합성 이미지에 기초하여 현재 이미지를 재구성한다. 일부 예들에서, 인코더 프로세서는 매 이미지에 대한 장면 구조 맵을 생성하여 시그널링하지 않을 수도 있다. 이들 예들에서, 인코더 프로세서 및 디코더 프로세서는 이전 장면 구조 맵을 재사용한다. 인코더 프로세서는 때때로 장면 구조 맵에 대한 업데이트들을 시그널링할 수도 있다.

[0007]

일 예에서, 본 개시는 비디오 데이터를 디코딩하는 방법을 설명하며, 본 방법은, 복합 이미지와 장면 구조 맵에 기초하여 합성 이미지를 생성하는 단계로서, 상기 복합 이미지는 이전에 디코딩되었던 하나 이상의 이미지들로부터 구성되고, 장면 구조 맵은 비디오 데이터의 현재 이미지의 장면 구조 맵 또는 이전에 디코딩되었던 비디오 데이터의 이미지의 장면 구조 맵을 포함하고, 장면 구조 맵은 현재 이미지 또는 이전에 디코딩되었던 이미지 내에서 3차원 포인트들에 대한 좌표 값들을 포함하는, 합성 이미지를 생성하는 단계, 비디오 데이터의 현재 이미지의 잔차 이미지를 결정하는 단계로서, 잔차 이미지는 현재 이미지와 합성 이미지 사이의 차이를 나타내는, 잔차 이미지를 결정하는 단계, 및 합성 이미지와 잔차 이미지에 기초하여 현재 이미지를 재구성하는 단계를 포함한다.

[0008]

일 예에서, 본 개시는 비디오 데이터를 인코딩하는 방법을 설명하며, 본 방법은, 복합 이미지와 장면 구조 맵에 기초하여 합성 이미지를 생성하는 단계로서, 상기 복합 이미지는 이전에 인코딩되었던 하나 이상의 이미지들로부터 구성되고, 장면 구조 맵은 비디오 데이터의 현재 이미지의 장면 구조 맵 또는 이전에 인코딩되었던 비디오 데이터의 이미지의 장면 구조 맵을 포함하고, 장면 구조 맵은 현재 이미지 또는 이전에 인코딩되었던 이미지 내에서 3차원 포인트들에 대한 좌표 값들을 포함하는, 합성 이미지를 생성하는 단계, 합성 이미지와 현재 이미지에 기초하여 잔차 이미지를 결정하는 단계로서, 잔차 이미지는 현재 이미지와 합성 이미지 사이의 차이를 나타내는, 잔차 이미지를 결정하는 단계, 및 비디오 데이터의 현재 이미지를 인코딩하기 위해 잔차 이미지를 나타내는 정보를 출력하는 단계를 포함한다.

[0009]

일 예에서, 본 개시는 비디오 데이터를 코딩하는 디바이스를 설명하며, 이 디바이스는 이전에 코딩되었고 복합 이미지를 구성하는데 이용되는 하나 이상의 이미지들을 저장하도록 구성되는 비디오 메모리, 및 코더 프로세서를 설명한다. 코더 프로세서는, 복합 이미지와 장면 구조 맵에 기초하여 합성 이미지를 생성하는 것으로서, 장면 구조 맵은 비디오 데이터의 현재 이미지의 장면 구조 맵 또는 이전에 코딩되었던 비디오 데이터의 이미지의 장면 구조 맵을 포함하고, 장면 구조 맵은 현재 이미지 또는 이전에 코딩되었던 이미지 내에서 3차원 포인트들에 대한 좌표 값들을 포함하는, 합성 이미지를 생성하고, 그리고 현재 이미지의 잔차 이미지에 기초하여 현재 이미지를 코딩하는 것으로서, 잔차 이미지는 현재 이미지와 합성 이미지 사이의 차이를 나타내는, 상기 현재 이미지를 코딩하도록 구성된다.

[0010]

일 예에서, 본 개시는 명령들을 저장한 컴퓨터 관독가능 저장 매체를 기술하며, 명령들은 실행될 때, 비디오 데이터를 코딩하기 위한 디바이스에 대한 하나 이상의 프로세서들로 하여금, 복합 이미지와 장면 구조 맵에 기초

하여 합성 이미지를 생성하게 하는 것으로서, 상기 복합 이미지는 이전에 코딩되었던 하나 이상의 이미지들로부터 구성되고, 장면 구조 맵은 비디오 데이터의 현재 이미지의 장면 구조 맵 또는 이전에 코딩되었던 비디오 데이터의 이미지의 장면 구조 맵을 포함하고, 장면 구조 맵은 현재 이미지 또는 이전에 코딩되었던 이미지 내에서 3차원 포인트들에 대한 좌표 값을 포함하는, 합성 이미지를 생성하게 하고, 그리고 현재 이미지의 잔차 이미지에 기초하여 현재 이미지를 코딩하게 하는 것으로서, 잔차 이미지는 현재 이미지와 합성 이미지 사이의 차이를 나타내는, 상기 현재 이미지를 코딩하게 한다.

[0011] 일 예에서, 본 개시는 비디오 데이터를 코딩하는 디바이스를 설명하며, 본 디바이스는 복합 이미지와 장면 구조 맵에 기초하여 합성 이미지를 생성하는 수단으로서, 상기 복합 이미지는 이전에 코딩되었던 하나 이상의 이미지들로부터 구성되고, 장면 구조 맵은 비디오 데이터의 현재 이미지의 장면 구조 맵 또는 이전에 코딩되었던 비디오 데이터의 이미지의 장면 구조 맵을 포함하고, 장면 구조 맵은 현재 이미지 또는 이전에 코딩되었던 이미지 내에서 3차원 포인트들에 대한 좌표 값을 포함하는, 합성 이미지를 생성하는 수단, 및 현재 이미지의 잔차 이미지에 기초하여 현재 이미지를 코딩하는 수단으로서, 잔차 이미지는 현재 이미지와 합성 이미지 사이의 차이를 나타내는, 상기 현재 이미지를 코딩하는 수단을 포함한다.

[0012] 하나 이상의 실시예들의 세부사항들은 첨부된 도면과 하기의 설명으로부터 설명된다. 다른 특징들, 목적들 및 이점들은 설명, 도면들, 및 청구항들로부터 명확해질 것이다.

도면의 간단한 설명

[0013] 도 1 은 본 개시에서 설명된 하나 이상의 예시적인 비디오 코딩 기법들을 구현하거나 또는 달리 이용하도록 구성되는 예시적인 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도이다.

도 2 는 본 개시에서 설명된 하나 이상의 예시적인 비디오 인코딩 기법들을 구현하거나 또는 달리 이용하도록 구성되는 인코더 프로세서의 일 예를 예시하는 블록도이다.

도 3a 및 도 3b 는 장면 구조 맵들의 예들을 예시하는 그래픽도들이다.

도 4 는 장면 구조 맵의 포인트들의 상호접속의 일 예를 예시하는 개념도이다.

도 5 는 본 개시에 따라 하나 이상의 예시적인 비디오 인코딩 기법들을 예시하는 데이터 흐름도이다.

도 6 은 본 개시에 따라 설명된 기법들에 따라 비디오 인코딩의 일 예를 예시하는 개념도이다.

도 7 은 본 개시에서 설명된 하나 이상의 예시적인 비디오 디코딩 기법들을 구현하거나 또는 달리 이용하도록 구성되는 디코더 프로세서의 일 예를 예시하는 블록도이다.

도 8 은 본 개시에 따라 하나 이상의 예시적인 비디오 디코딩 기법들을 예시하는 데이터 흐름도이다.

도 9 는 비디오 인코딩의 예시적인 방법을 예시하는 플로우차트이다.

도 10 은 비디오 디코딩의 예시적인 방법을 예시하는 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0014] 본 개시는 합성 이미지 및 장면 구조 맵에 기초하여 현재 이미지를 비디오 코딩 (예를 들어, 인코딩 또는 디코딩) 하기 위한 여러 기법들을 설명한다. 이러한 식으로, 적어도 현재 이미지의 부분에 대하여 비디오 코딩을 위하여 통상의 모션 보상이 불필요할 수도 있다. 본 개시에 어디에서나 보다 자세하게 설명된 바와 같이, 장면 구조 맵을 구성하는데 요구되는 정보의 양은 비교적 낮을 수도 있고, 합성 이미지는 표준 하드웨어를 이용하여 이전 이미지(들) 및 장면 구조 맵으로 구성될 수 있다. 따라서, 본 개시에 설명된 기법들은 통상의 모션 보상 기법들에 관련한 비디오 코딩에 요구되는 대역폭을 감소시킬 수도 있다. 또한, 본 기법들이 표준 하드웨어 상에서 구현될 수 있기 때문에, 어떠한 추가적인 컴포넌트들도 본 개시에서 설명된 기법들을 구현하기 위하여 요구되지 않을 수도 있다.

[0015] 장면 구조 맵은 이미지 내에서 수개의 포인트들 (예를 들어, 키포인트들)에 대한 좌표 값을 포함할 수도 있다. 예를 들어, 이미지는 3차원 (3D) 이미지일 수도 있다. 이 경우에, 3D 이미지는 스테레오스코픽 이미지가 아니라, 그 이미지 내에 오브젝트들의 상대 심도가 존재하는 이미지이다 (예를 들어, 이미지가 2D 공간을 수반하고 있음에도 불구하고 오브젝트가 다른 오브젝트에 비해 훨씬 뒤에 있는 것으로 보여진다).

[0016] 장면 구조 맵은 각각의 포인트에 대해 x, y, 및 z 좌표를 포함할 수도 있다. 장면 구조 맵의 포인트들은 접

속될 때 이미지의 구조를 정의하는 메시를 함께 형성하는 다각형들 (예를 들어, 삼각형들) 의 베틱스들을 정의 할 수도 있다. 이 장면에서, 장면 구조 맵의 상호접속은 이미지의 프록시 지오메트리로서 간주될 수도 있다. 본 개시에서, 프록시 지오메트리를 형성하는데 이용되는 장면 구조 맵은 아래 보다 자세하게 설명된 바와 같이, 현재 이미지의 장면 구조 맵 또는 이전에 인코딩 또는 디코딩되었던 이미지의 장면 구조 맵일 수도 있다.

[0017] 장면 구조 맵을 생성하는 여러 방식들이 있을 수도 있다. 장면 구조 맵을 생성하는 일 예시적인 방식은 동시 로컬리제이션 및 맵핑 (simultaneous localization and mapping; SLAM) 기법들을 이용하는 것이다. 그러나, 본 개시에 설명된 기법들은 장면 구조 맵을 생성하는 다른 방식들로 확장가능할 수도 있다. 예시의 목적들을 위하여, 장면 구조 맵을 생성하는 기법들은 SLAM 기법들에 대하여 설명된다. 예를 들어, 본 개시에 설명된 기법들은 장면 구조 맵을 생성하는 SLAM 프로세서 (예를 들어, SLAM 기법들을 구현하도록 구성되는 프로세서) 의 관점에서부터 설명된다.

[0018] 일부 예들에서, SLAM 프로세서는 이전에 인코딩 또는 디코딩된 이미지에 기초하여 또는 현재 이미지에 기초하여 현재 이미지를 인코딩 또는 디코딩하기 위하여 이용되는 장면 구조 맵을 생성할 수도 있다. 예를 들어, 비디오 인코딩을 수행하는 디바이스는 SLAM 프로세서를 포함할 수도 있다. 이 예에서, SLAM 프로세서는 현재 이미지에 기초하여 현재 이미지에 대한 장면 구조 맵을 생성할 수도 있다. 인코더 프로세서는 본 개시에 어디에서나 보다 자세하게 설명된 바와 같이, 생성된 장면 구조 맵에 기초하여 현재 이미지를 인코딩하고, 생성된 장면 구조 맵을 출력할 수도 있다. 비디오 디코딩을 수행하는 디바이스는 본 개시에 어디에서나 보다 자세하게 설명된 바와 같이, 장면 구조 맵을 수신할 수도 있고 장면 구조 맵에 기초하여 현재 이미지를 디코딩할 수도 있다.

[0019] 일부 예들에서, SLAM 프로세서는 현재 이미지에 기초하여 장면 구조 맵을 생성하는 것이 필요하지 않을 수도 있다. 이들 예들에서, 인코더 프로세서는 이전에 생성되었던 장면 구조 맵 (예를 들어, 인코더 프로세서가 이전에 생성하였던 장면 구조 맵)에 기초하여 현재 이미지를 인코딩할 수도 있다. 이와 유사하게, 디코더 프로세서는 이전에 수신되었던 장면 구조 맵 (예를 들어, 디코더 프로세서가 이전에 수신하였던 장면 구조 맵)에 기초하여 현재 이미지를 디코딩할 수도 있다.

[0020] 장면 구조 맵에 더하여, 인코더 프로세서 및 디코더 프로세서는 현재 이미지를 각각 인코딩 및 디코딩하기 위해 합성 이미지를 이용한다. 합성 이미지를 생성하도록 구성되는 비디오 디코딩을 수행하는 디바이스 또는 비디오 인코딩을 수행하는 디바이스 내에 여러 컴포넌트들이 존재할 수도 있다. 설명의 목적들을 위하여, 다음은 합성 이미지를 생성하는 그래픽 프로세싱 유닛 (graphics processing unit; GPU) 으로 설명한다.

[0021] 비디오 인코딩을 위한 디바이스 및 비디오 디코딩을 위한 디바이스 양쪽은 개개의 GPU들을 포함한다. GPU들은 각각은 합성 이미지를 생성하기 위해 실질적으로 동일한 기능들을 수행한다. 따라서, GPU 가 합성 이미지를 생성하는 방식의 다음의 설명은 비디오 인코딩을 위한 디바이스 및 비디오 디코딩을 위한 디바이스 양쪽에 적용 가능하다.

[0022] GPU 는 하나 이상의 이미지들 (예를 들어, 이전에 디코딩된 이미지들) 및 장면 구조 맵에 기초하여 합성 이미지를 생성할 수도 있다. 또한, 장면 구조 맵은 현재 이미지로부터 생성되는 장면 구조 맵이거나 또는 이전에 인코딩되거나 또는 디코딩된 이미지로부터 생성되는 장면 구조 맵일 수도 있다. GPU 가 복합 이미지를 생성하기 위해 하나보다 많은 이미지를 이용하는 예들에서, GPU 또는 일부 다른 컴포넌트는 합성 이미지를 형성하기 위해 이미지들 중 둘 이상의 이미지들 (예를 들어, 이전에 디코딩 또는 인코딩된 퍽처들) 을 블렌딩하는 블렌딩 동작을 수행할 수도 있다. GPU 가 오직 하나의 이미지만을 이용하는 예들에서, 이미지는 복합 이미지로서 고려될 수도 있다.

[0023] GPU 는 다각형들의 상호접속 상에 텍스처 맵을 오버레이하는 함수를 갖는 텍스처 엔진을 포함한다. 본 개시에 설명된 기법들에 따르면, GPU 는 예를 들어, 베틱스 쉐이더 또는 입력 어셈블러를 이용하여 장면 구조 맵의 밀도가 희소한 포인트들을 상호접속한다. 밀도가 희소한 포인트들의 예들은 아래에 제공된다. 예를 들어, 장면 구조 맵의 포인트들은 GPU 가 메시를 형성하기 위해 상호접속하는 삼각형들의 베틱스들로서 고려될 수도 있다. GPU 는 프록시 지오메트리를 형성하기 위해 장면 구조 맵의 포인트들을 상호접속할 수도 있다. 위에 설명된 바와 같이, 장면 구조 맵의 포인트들은 x, y, 및 z 좌표들에 의해 정의되기 때문에, 장면 구조 맵은 3D 공간에서 정의된다.

[0024] 복합 이미지 (예를 들어, 둘 이상의 이미지들의 블렌드 또는 하나의 이미지) 가 텍스처 맵으로서 고려될 수도

있다. 이 예에서, 복합 이미지의 컨텐츠는 3D 일 수도 있지만, 각각의 픽셀은 x 및 y 좌표에 의해 식별된다. 장면 구조 맵의 x, y 및 z 좌표들과의 혼란을 회피하기 위하여, 복합 이미지 (예를 들어, 텍스처 맵) 의 픽셀들의 x 및 y 좌표들은 각각 u 및 v 좌표들로 지칭된다.

[0025] GPU 의 텍스처 엔진은 장면 구조 맵 상에 복합 이미지를 오버레이하기 위해 3D 장면 구조 맵 상에 복합 이미지의 각각의 픽셀을 맵핑할 수도 있다. 즉, 텍스처 엔진은 (u, v) 좌표들에 의해 정의된 픽셀들 각각을 장면 구조 맵의 (x, y, z) 좌표들 상에 맵핑한다. 텍스처 엔진의 결과는 이미지-기반 모델 (image based model; IBM) 로서 고려될 수도 있다. 이 IBM 모델은 (x, y, z) 좌표들에 의해 정의되는 픽셀들을 갖는 그래픽 구성일 수도 있다.

[0026] GPU 는 합성 이미지를 생성하기 위해 IBM 을 렌더링할 수도 있다. 예를 들어, GPU 는 합성 이미지를 생성하기 위해 IBM 을 프로세싱하는 표준 그래픽 프로세싱 작업들을 수행한다. 합성 이미지는 (예를 들어, 디스플레이 상에 (x, y) 픽셀 좌표들에 의해 정의되는 픽셀들을 갖는) 최종의 뷰잉가능 이미지로서 고려될 수 있다.

그러나, 뷰잉을 위한 합성 이미지를 출력하기 보다는, 인코더 프로세서 및 디코더 프로세서는 예측 이미지로서 합성 이미지를 이용할 수도 있다.

[0027] 인코더 프로세서는 현재 이미지와 합성 이미지 사이의 잔차 이미지로서 지칭되는 차이를 결정하고 잔차 이미지를 나타내는 정보를 출력한다. 디코더 프로세서는 (예를 들어, 잔차 이미지를 나타내는 수신된 정보에 기초하여) 잔차 이미지를 결정하고, 그리고 현재 이미지를 재구성하기 위해 국부적으로 생성된 합성 이미지와 잔차 이미지를 합산한다.

[0028] SLAM 프로세서가 현재 이미지에 기초하여 장면 구조 맵을 생성하였던 예들에서, 인코더 프로세서는 장면 구조 맵의 정보 (예를 들어, 장면 구조 맵의 포인트들의 좌표들) 를 추가적으로 출력할 수도 있다. 이들 예들에서, 디코더 프로세서는 장면 구조 맵의 정보를 수신하고 디코더 프로세서 상의 GPU 는 수신된 장면 구조 맵에 기초하여 합성 이미지를 생성한다. SLAM 프로세서가 현재 이미지에 기초하여 장면 구조 맵을 생성하지 않았고 그 대신에 GPU 가 합성 이미지를 생성하기 위해 이전 장면 구조 맵을 재사용하였던 예들에서, 인코더 프로세서는 장면 구조 맵의 정보를 출력하지 않을 수도 있다. 디코더 프로세서에서의 GPU 는 합성 이미지를 생성하기 위한 동일한 이전 장면 구조 맵을 재사용할 수도 있다.

[0029] 인코더 프로세서 측의 GPU 가 합성 이미지를 생성하고 SLAM 프로세서가 현재 이미지에 기초하여 장면 구조 맵을 생성하였던 일부 예들에서, 인코더 프로세서는 장면 구조 맵의 포인트들의 좌표들을 반드시 출력할 필요가 있는 것은 아니다. 오히려, 인코더 프로세서는 이전 장면 구조 맵에 관련하여 장면 구조 맵의 포인트들에 대한 충분적 업데이트들을 출력할 수도 있다. 디코더 프로세서는 장면 구조 맵의 업데이트 값들을 수신하고 GPU 는 이전 장면 구조 맵을 업데이트하여 GPU 가 현재 이미지를 디코딩하기 위한 합성 이미지를 생성하는데 이용하는 장면 구조 맵을 생성한다. 본 개시에서, 장면 구조 맵의 정보는 장면 구조 맵의 포인트들의 좌표들, 또는 이전 장면 구조 맵에 관련하여 포인트들의 좌표들에 대한 업데이트 값들을 포함한다.

[0030] 본 개시에 설명된 기법들은 비디오 인코딩 및 디코딩에 일반적으로 적용가능할 수도 있다. 일 예로서, 기법들은 장면이 이동 카메라에 의해 관측되는 예들에서 그리고 실시간 비디오 인코딩 및 디코딩이 필요한 예들에서 이용될 수도 있다. 예를 들어, 비디오 전화 또는 화상 회의에서, 비디오 인코딩 및 디코딩은 실시간으로 필요할 수도 있고 본 개시에 설명된 기법들은 이러한 실시간 비디오 인코딩 및 디코딩을 제공한다. 예를 들어, GPU 는 실시간 비디오 인코딩 및 디코딩을 비교적 신속하게 용이하게 하는 합성 이미지를 생성할 수 있고 장면 구조 맵을 출력하고 수신하는데 필요한 대역폭의 양이 비교적 낮을 수도 있다.

[0031] 위에서는 실시간 비디오 인코딩 및 디코딩에 대하여 그리고 이동 카메라에 의한 예들에 대하여 설명되어 있지만, 기법들은 이러한 것으로 제한되지 않음을 이해하여야 한다. 일부 예들에서, 비디오 인코딩 및 비디오 디코딩 기법들은 오프라인 비디오 인코딩 및 디코딩 기법 (예를 들어, 비실시간) 에도 적용가능하며, 카메라가 스틸 카메라인 경우의 비디오 인코딩 및 디코딩 기법들에 대해서도 적용가능하다.

[0032] 또한, 카메라가 이동 카메라인 일부 예들에서, 인코더 프로세서는 카메라 포즈로 지칭되는 카메라의 포지션 및 배향을 나타내는 정보를 디코더 프로세서에 출력할 수도 있다. 디코더 프로세서는 현재 이미지를 디코딩 (예를 들어, 재구성) 하는 목적을 위하여 카메라 포즈를 이용한다. 예를 들어, 카메라 포즈 정보는 이미지가 "어디에서부터" 촬영되었는지를 표시한다. 카메라가 이동 카메라인 예들에서, 현재 이미지에 대한 카메라 포즈는 이전에 코딩된 이미지들에 대한 카메라 포즈와는 상이할 수도 있다. 따라서, 합성 이미지를 렌더링하기 위하여, 개별적인 GPU들은 합성 이미지와 현재 이미지에 대한 카메라 포즈가 동일하게 되도록, 현재 이

미지의 카메라 포즈 정보를 이용할 수도 있다. 또한, 개별적인 GPU들은 장면 구조 맵에 복합 이미지를 맵핑 할 때 복합 이미지를 구성하는데 이용되는 하나 이상의 이전에 코딩된 이미지들에 대한 카메라 포즈 정보를 이용할 수도 있다.

[0033]

위에 예에서, 개별적인 GPU들은 장면 구조 맵에 복합 이미지를 맵핑하기 위하여 (GPU 가 인코더 측에 있는지 또는 디코더 측에 있는지에 기초하여) 합성 이미지를 렌더링하기 위하여 현재 이미지에 대한 카메라 포즈 정보와, 이전에 인코딩 또는 디코딩된 이미지들 중 하나 이상의 이미지들의 카메라 포즈 정보를 이용한다. 따라서, 일부 예들에서, 인코더 프로세서는 현재 이미지를 인코딩하기 위하여 카메라 포즈 정보를 이용할 수도 있고 각각의 이미지에 대한 카메라 포즈 정보를 시그널링할 수도 있다. 디코더 프로세서는 각각의 이미지에 대한 카메라 포즈 정보를 수신할 수도 있고 현재 이미지를 디코딩하기 위하여 카메라 포즈 정보를 이용할 수도 있다.

카메라가 이동 카메라가 아닌 예들에서, 카메라 포즈 정보는 암묵적으로 도출될 수도 있거나 또는 상수 값으로 설정될 수도 있다. 이러한 예들에서, 인코더 프로세서는 각각의 이미지에 대하여 카메라 포즈 정보를 반드시 시그널링할 필요가 있는 것은 아니고 디코더 프로세서는 각각의 이미지에 대하여 카메라 포즈 정보를 반드시 수신할 필요가 있는 것은 아니다.

[0034]

위의 예들에서, 디코더 프로세서는 인코더 프로세서의 프로세스와 역의 프로세스로서 구현하도록 구성될 수도 있다. 이 개시에서, 용어 "코더 프로세서" 는 인코더 프로세서 또는 디코더 프로세서를 일반적으로 지칭하는데 이용될 수도 있다. 예를 들어, 코더 프로세서는 복합 이미지와 장면 구조 맵에 기초하여 합성 이미지를 생성할 수도 있다. 장면 구조 맵은 현재 이미지의 장면 구조 맵, 또는 이전에 코딩 (예를 들어, 적용 가능한 경우에 따라 인코딩 또는 디코딩) 되었던 이미지의 장면 구조 맵을 포함하며, 장면 구조 맵은 현재 이미지 또는 이전에 코딩된 이미지 내에서 3차원 포인트들에 대한 좌표값들을 포함한다. 이 경우에, 디코더 프로세서 및 인코더 프로세서는 실질적으로 동일한 방식으로 합성 이미지를 생성하도록 구성될 수도 있으며, 따라서, 이 예는 코더 프로세서에 의해 수행되는 것으로서 기술된다.

[0035]

코더 프로세서는 현재 이미지의 잔차 이미지에 기초하여 현재 이미지를 코딩 (예를 들어, 적용 가능한 바에 따라 인코딩 또는 디코딩) 할 수도 있다. 잔차 이미지는 현재 이미지와 합성 이미지 사이의 차이를 나타낸다.

[0036]

예를 들어, 코더 프로세서가 디코더 프로세서로 지칭되는 경우에 대하여, 디코더 프로세서는 잔차 이미지를 수신하는 것에 기초하여 잔차 이미지를 결정할 수도 있고, 현재 이미지를 디코딩하기 위하여, 비디오 디코더는 합성 이미지와 잔차 이미지에 기초하여 현재 이미지를 재구성할 수도 있다. 코더 프로세서가 인코더 프로세서를 지칭하는 경우에 대하여, 인코더 프로세서는 합성 이미지와 현재 이미지에 기초하여 잔차 이미지를 결정하고 (현재 이미지를 디코딩하기 위한 디코더 프로세서에 의해 이용되는) 잔차 이미지를 나타내는 정보를 출력하여 현재 이미지를 인코딩할 수도 있다.

[0037]

이 개시에서, 이전에 코딩 (예를 들어, 인코딩 또는 디코딩) 된 이미지들은 이전에 디스플레이되었던 이미지를 지칭할 수도 있지만 이 기법들은 이에 제한되지 않는다. 예를 들어, 이전에 코딩된 이미지들은 디스플레이 순서에서 시간적으로 앞에 있는 픽처들일 수도 있다. 그러나, 일부 예들에서, 이전에 코딩된 이미지들은 현재 이미지와 동시에 또는 바로 즉시 디스플레이되는 이미지들일 수도 있다.

[0038]

일 예로서, 스테레オス코픽 뷰에 대하여, 둘 이상의 이미지들이 다른 것과 동시에 또는 바로 뒤에 이어서 디스플레이된다. 스테레オス코픽 뷰에 대해, 이미지를 중 하나의 이미지는 스테레オス코픽 뷰의 다른 이미지를 코딩 (예를 들어, 인코딩 또는 디코딩) 하는데 이용될 수도 있다. 이러한 예들에서, 코딩에 이용되는 이미지들은 이전에 코딩된 이미지들의 예들인데, 그 이유는 이들 이미지들은, 잔차를 결정하기 위하여 이들 코딩된 이미지들을 이용하는 현재 이미지에 앞서 인코딩 또는 디코딩되기 때문이다.

[0039]

본 개시에서, 용어 이전 이미지는 종종 이용된다. 이전 이미지는 (이 경우가 한번만 가능하더라도) 이전 이미지가 현재 이미지보다 더 이르게 디스플레이되는 경우로만 제한되는 것은 아님을 이해해야 한다. 오히려, 이전 이미지는 현재 이미지에 대하여 이전에 인코딩 또는 디코딩되었던 이미지를 지칭한다. 이전에 인코딩 또는 디코딩된 이미지는 현재 이미지에 앞서 디스플레이될 수도 있거나, (예를 들어, 스테레オス코픽 또는 3D 뷰에 대하여) 현재 이미지와 가상적으로 동시에 디스플레이될 수도 있거나 또는 가능하게는 현재 이미지 이후에 디스플레이될 수도 있다. 또한, 본 개시에서, 용어 "이미지" 및 "픽처" 는 상호교환적으로 이용될 수도 있다.

[0040]

도 1 은 본 개시에서 설명된 하나 이상의 예시적인 비디오 코딩 기법들을 구현하거나 또는 달리 이용하도록 구성되는 예시적인 비디오 인코딩 및 디코딩 시스템 (10) 을 예시하는 블록도이다. 도 1 의 예에서, 소스 디

바이스 (12) 는 비디오 소스 (18), 인코더 프로세서 (20) 및 출력 인터페이스 (22) 를 포함한다. 목적지 디바이스 (14) 는 입력 인터페이스 (28), 디코더 프로세서 (30), 및 디스플레이 디바이스 (32) 를 포함한다. 본 개시에 따르면, 소스 디바이스 (12) 의 인코더 프로세서 (20) 및 목적지 디바이스 (14) 의 디코더 프로세서 (30) 는 본 개시에 설명된 예시적인 기법들을 구현하도록 구성될 수도 있다. 일부 예들에서, 소스 디바이스 와 목적지 디바이스는 다른 컴포넌트들 또는 장치들을 포함할 수도 있다. 예를 들면, 소스 디바이스 (12) 는 외부 비디오 소스 (18), 이를 테면 외부 카메라로부터 비디오 데이터를 수신할 수도 있다. 마찬가지로, 목적지 디바이스 (14) 는 통합형 디스플레이 디바이스를 포함하기 보다는, 외부 디스플레이 디바이스와 인터페이스 할 수도 있다.

[0041] 도 1 에 도시된 바와 같이, 시스템 (10) 은 목적지 디바이스 (14) 에 의해 디코딩된 인코딩된 비디오 데이터를 제공하는 소스 디바이스 (12) 를 포함한다. 특히, 소스 디바이스 (12) 는 링크 (16) 를 통하여 목적지 디바이스 (14) 에 비디오 데이터를 제공한다. 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 데스크탑 컴퓨터들, 노트북 (즉, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋탑 박스들, 소위 "스마트" 폰들과 같은 전화 핸드셋들, 소위 "스마트" 패드들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게임용 콘솔들, 비디오 스트리밍 디바이스 등을 포함하는 다양한 범위의 디바이스들 중 어느 것을 포함할 수도 있다. 일부 경우들에서, 소스 디바이스 (12) 및 목적지 디바이스 (14) 에는 무선 통신이 설치될 수도 있다.

[0042] 목적지 디바이스 (14) 는 링크 (16) 를 통하여, 디코딩된 인코딩된 비디오 데이터를 수신할 수도 있다. 링크 (16) 는 인코딩된 비디오 데이터를 소스 디바이스 (12) 에서 목적지 디바이스 (14) 로 이동 가능한 임의의 유형의 매체 또는 디바이스를 포함할 수도 있다. 일 예에서, 링크 (16) 는 소스 디바이스 (12) 로 하여금 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 실시간으로 직접 송신하게 하는 통신 매체 (유선 또는 무선 매체) 를 포함할 수도 있다. 인코딩된 비디오 데이터는 무선 통신 프로토콜과 같은 통신 표준에 따라 변조되고, 목적지 디바이스 (14) 에 송신될 수도 있다. 통신 매체는 임의의 무선 또는 유선 통신 매체, 예컨대 무선 주파수(RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들을 포함할 수도 있다. 통신 매체는 패킷 기반의 네트워크, 예컨대 근거리 통신망 (local area network), 광역 통신망 (wide-area network), 또는 인터넷과 같은 글로벌 네트워크의 일부를 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 에서 목적지 디바이스 (14) 로의 통신을 용이하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0043] 일부 예들에서, 인코딩된 데이터는 출력 인터페이스 (22) 로부터 저장 디바이스, 이를 테면, 저장 디바이스 (31) 로 출력될 수도 있다. 이와 유사하게, 인코딩된 데이터는 입력 인터페이스 (28) 에 의해 저장 디바이스 (31) 로부터 액세스될 수도 있다. 저장 디바이스 (31) 는 하드 드라이브, 블루레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 휘발성 또는 비휘발성 메모리와 같은 임의의 다양한 분산된 또는 로컬하게 액세스된 데이터 저장 매체, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적절한 디지털 저장 매체를 포함할 수도 있다. 다른 예에서, 저장 디바이스 (31) 는 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오를 저장할 수도 있는 다른 중간 저장 디바이스 또는 파일 서버에 대응할 수도 있다. 목적지 디바이스 (14) 는 스트리밍 또는 다운로드를 통하여 저장 디바이스로부터의 저장된 비디오 데이터에 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장할 수 있고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수 있는 임의의 유형의 서버일 수도 있다. 예시적인 파일 서버들은 (예를 들면 웹사이트 용의) 웹 서버, FTP 서버, NAS (network attached storage) 디바이스들, 또는 로컬 디스크 드라이브를 포함한다. 목적지 디바이스 (14) 는 인터넷 접속을 포함하는 임의의 표준 데이터 접속을 통하여 인코딩된 비디오 데이터에 액세스할 수도 있다. 이것은 파일 서버에 저장된 인코딩된 비디오 데이터를 액세스하는데 적합한 무선 채널 (예를 들어, 와이파이 접속), 유선 접속 (예를 들어, DSL, 케이블 모뎀 등), 또는 이들의 조합을 포함할 수도 있다. 저장 디바이스로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 이들의 조합일 수도 있다.

[0044] 본 개시의 기술들은 무선 애플리케이션들 또는 설정들에 반드시 제한되는 것은 아니다. 기법들은 다양한 멀티미디어 애플리케이션들, 예컨대 지상파 (over-the-air) 텔레비전 방송들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 인터넷 스트리밍 비디오 송신들, 이를 테면 DASH (dynamic adaptive streaming over HTTP), 데이터 저장 매체 상에서 인코딩된 디지털 비디오, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩 또는 다른 애플리케이션들의 지원에 있어서 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 시스템 (10) 은, 비디오 스트리밍, 비디오 플레이백, 비디오 방송, 및/또는 비디오 전화와 같은 애플리케이션들을 지원하기 위해 일방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0045]

도 1 의 예시된 시스템 (10) 은 단지 일 예에 불과하며, 본 개시에 설명된 기법들은 임의의 디지털 비디오 인코딩 및/또는 디코딩 디바이스에 의해 수행될 수도 있다. 일반적으로, 본 개시의 기법들이 비디오 인코딩/디코딩 디바이스에 의해 수행되지만, 그 기법들은 통상적으로 "CODEC"으로서 지정되는 인코더 프로세서/디코더 프로세서에 의해 또한 수행될 수도 있다. 또한, 본 개시의 기법들은 비디오 프리프로세서에 의해 또한 수행될 수도 있다. 소스 디바이스 (12) 와 목적지 디바이스 (14) 는, 소스 디바이스 (12) 가 목적지 디바이스 (14) 로의 전송을 위한 코딩된 비디오 데이터를 생성하는 이러한 코딩 디바이스들의 단지 예들에 불과하다. 몇몇 실시형태들에서, 디바이스들 (12, 14) 은, 디바이스들 (12, 14) 의 각각이 비디오 인코딩 및 디코딩 컴포넌트들을 포함하는 것과 실제 대칭적인 방식으로 동작할 수도 있다. 따라서, 시스템 (10) 은, 예를 들면, 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 또는 비디오 전화를 위해, 비디오 디바이스들 (12, 14) 사이의 일방향 또는 양방향 비디오 송신을 지원할 수도 있다.

[0046]

소스 디바이스 (12) 의 비디오 소스 (18) 는 비디오 캡처 디바이스, 이를 테면 비디오 카메라, 이전에 캡처된 비디오를 포함하는 비디오 아카이브, 및/또는 비디오 컨텐츠 공급자로부터 비디오를 수신하는 비디오 피드 인터페이스를 포함할 수도 있다. 다른 대안으로, 비디오 소스 (18) 는 소스 비디오로서 컴퓨터 그래픽 기반 데이터, 또는 라이브 비디오, 아카이브된 비디오, 및 컴퓨터 생성 비디오의 조합을 생성할 수도 있다. 그러나, 위에서 언급된 바와 같이, 본 개시에서 설명된 기법들은 일반적으로 비디오 코딩에 적용될 수 있으며, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다. 각각의 경우에서, 캡처된, 프리캡처된, 또는 컴퓨터에 의해 생성된 비디오는 인코더 프로세서 (20) 에 의해 인코딩될 수도 있다. 출력 인터페이스 (22) 는 그 후, 링크 (16) 상에 또는 저장 디바이스 (31) 로 인코딩된 비디오 정보를 출력할 수도 있다.

[0047]

목적지 디바이스 (14) 의 입력 인터페이스 (28) 는 링크 (16) 및/또는 저장 디바이스 (31) 로부터 정보를 수신한다. 수신된 정보는 인코더 프로세서 (20) 에 의해 생성되어 디코더 프로세서 (30) 에 의해 또한 이용되는 신택스 정보를 포함하며, 이 신택스 정보는 이미지들의 특징들 및/또는 프로세싱을 기술하는 신택스 엘리먼트들을 포함한다. 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 사용자에게 디스플레이하며, 음극선관 (cathode ray tube; CRT), 액정 디스플레이 (liquid crystal display; LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (organic light emitting diode; OLED) 디스플레이, 또는 다른 유형의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 어느 것을 포함할 수도 있다.

[0048]

도 1 에 도시되지 않았으나, 일부 양태들에서 인코더 프로세서 (20) 및 디코더 프로세서 (30) 는 각각의 오디오 인코더 및 디코더와 통합될 수도 있고, 공통 데이터 스트림 또는 개별의 데이터 스트림들에서 오디오 및 비디오 양쪽의 인코딩을 처리하기 위해 적절한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함할 수도 있다. 적용 가능한 경우, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 다른 프로토콜들, 이를 테면, 사용자 데이터그램 프로토콜 (user datagram protocol; UDP) 에 따를 수도 있다.

[0049]

인코더 프로세서 (20) 및 디코더 프로세서 (30) 각각은 임의의 다양한 적절한 회로부, 이를 테면 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSPs), 주문형 반도체들 (application specific integrated circuits; ASICs), 필드 프로그램 가능 게이트 어레이들 (FPGAs), 이산 로직, 소프트웨어, 하드웨어, 펌웨어 또는 이들의 임의의 조합들로서 구현될 수도 있다. 인코더 프로세서 (20) 및/또는 디코더 프로세서 (30) 를 포함하는 디바이스는 접속 회로, 마이크로프로세서, 및/또는 무선 통신 디바이스, 이를 테면 셀룰러 전화기를 포함할 수도 있다. 기법들이 부분적으로 소프트웨어로 구현되는 경우, 디바이스는 그 소프트웨어에 대한 명령들을 적절한 컴퓨터 판독가능 저장 매체에 저장할 수도 있고, 본 개시물의 기법들을 수행하기 위해 하나 이상의 프로세서들을 이용하여 하드웨어에서 그 명령들을 실행할 수도 있다.

[0050]

본 개시에 설명된 기법들에 따르면, 인코더 프로세서 (20) 는 비디오 소스 (18) 로부터 비디오 데이터를 수신한다. 비디오 데이터는 현재 이미지에 대한 비디오 데이터를 포함한, 비디오를 함께 형성하는 복수의 이미지들에 대한 비디오 데이터를 포함한다. 현재 이미지에 대한 비디오 데이터는 현재 이미지의 픽셀들에 대한 픽셀 값들 (예를 들어, 컬러 값들) 및 좌표들을 포함한다. 현재 이미지를 인코딩하기 위하여, 인코더 프로세서 (20) 는 장면 구조 맵을 이용한다. 일부 예들에서, 인코더 프로세서 (20) 는 현재 이미지의 비디오 데이터에 기초하여 장면 구조 맵을 생성한다. 일부 예들에서, 인코더 프로세서 (20) 는 이전에 생성된 장면 구조 맵 (예를 들어, 이전에 인코딩된 이미지에 대하여 생성된 장면 구조 맵) 을 재사용한다.

[0051]

장면 구조 맵은 이전 이미지 (또는 이전에 인코딩된 이미지) 또는 현재 이미지의 최소 개수의 포인트들에 대한 3 차원 (3D) 좌표들을 포함한다. 예를 들어, 자연 구조 맵에 정의된 포인트들의 수는 현재 이미지에서의 픽셀들의 총 수 미만이다. 본 개시에 설명된 기법들에서, 장면 구조 맵에서 포인트들의 3D 좌표들을 송신하는

데 필요한 바이트들의 양은 현재 이미지를 송신하는데 필요한 바이트들의 양보다 실질적으로 작을 수도 있다.

일부 경우들에서, 현재 이미지에 비교하여, 1% 미만 (예를 들어, 0.26%) 의 바이트들이 장면 구조 맵을 송신하는데 필요하다. 또한, 일부 예들에서, 장면 구조 맵의 모든 포인트들이 출력되는 것은 아니며, 장면 구조 맵에 대한 업데이트들만이 출력되어, 그 결과, 이미지에 비하여 장면 구조 맵에 대하여 정보를 송신하는데 보다 적은 바이트들만이 필요하게 된다.

[0052] 본 개시에 설명된 본 기법에서, 현재 이미지는 2D 공간을 수반하고 있음을 이해해야 한다. 그러나, 현재 이미지 내에서 오브젝트들은 전경, 배경, 또는 중간에 어디에서나 있는 것으로 보여진다. 즉, 현재 이미지에서의 픽셀들이 2D 좌표들로 정의되지만, 현재 이미지에서의 오브젝트들의 상태 심도가 있는 것으로 보여진다.

인코더 프로세서 (20) 는 장면 구조 맵의 3D 좌표들을 형성하기 위해 현재 이미지에서의 오브젝트들의 이 상태 심도를 이용한다. 예를 들어, 인코더 프로세서 (20) 는 장면 구조 맵을 형성하기 위해 현재 이미지의 회 박 포인트들의 3D 좌표들을 형성하기 위한 동시 토컬리제이션 및 맵핑 (simultaneous localization and mapping; SLAM) 기법들을 이용할 수도 있다. 일부 예들에서, 인코더 프로세서 (20) 는 현재 이미지에 대한 장면 구조 맵을 반드시 형성할 필요가 있는 것은 아니며, 현재 이미지를 인코딩하기 위하여 이전에 구성된 장면 구조 맵을 재사용할 수도 있다. 이를 예들에서, 인코더 프로세서 (20) 는 유사한 기법들 (예를 들어, SLAM 기법들) 을 이용하여 이전에 인코딩된 이미지로부터 장면 구조 맵을 구성할 수도 있다.

[0053] SLAM 기법들은 하기와 보다 자세하게 설명된다. 그러나, SLAM 기법들은 장면 구조 맵을 구성하는 일 방식의 단지 일 예로서 설명된 것에 불과하다. 장면 구조 맵을 구성하는 다른 방식들이 가능할 수도 있으며, 본 기법들은 장면 구조 맵을 구성하기 위한 SLAM 기법들을 이용하는 것으로 제한되지 않는다.

[0054] 장면 구조 맵은 이미지 (예를 들어, 현재 이미지 또는 이전에 인코딩된 이미지) 의 프록시 지오메트리로서 고려될 수 있다. 예를 들어, 장면 구조 맵에 의해 정의되는 포인트들은 다각형들 (예를 들어, 삼각형들) 의 베 텍스들로서 고려될 수 있다. 포인트들이 상호접속되면 (예를 들어, 인코더 프로세서 (20) 또는 디코더 프로 세서 (30) 가 포인트들을 상호접속하면), 그 결과는 이미지의 3차원 구조이며, 따라서, 용어 "장면 구조 맵" 이 된다.

[0055] 장면 구조 맵은 멀티뷰 비디오 코딩에서 이용되는 심도 맵들과 혼란되지 않아야 된다. 멀티뷰 비디오 코딩에서, 다수의 이미지들은 실질적으로 유사한 시간에 제시되며, 뷰어의 오른쪽 눈은 이미지를 중 한 이미지를 뷰 임하고 뷰어의 왼쪽 눈은 이미지를 중 다른 이미지를 뷰 임한다. 각각의 눈이 상이한 이미지들을 뷰 임하는 결과는 뷰어가 3차원 뷰 임 볼륨을 수반하는 이미지를 경험하게 한다.

[0056] 멀티뷰 비디오 코딩에서, 현재 이미지의 심도 맵은 여러 비디오 코딩 기법들에 대하여 이용된다. 심도 맵은 자체적으로 이미지로서 처리되고, 여기에서, 심도 맵의 픽셀들의 픽셀 값들은 텍스처 맵 (즉, 실제 컨텐츠를 포함하는 이미지) 에서의 대응하는 픽셀들의 상태 심도를 나타낸다.

[0057] 본 개시에 설명된 장면 구조 맵은 심도 맵과 달리 이미지가 아니다. 장면 구조 맵은 이미지에서의 희소 개수의 포인트들에 대한 좌표들을 포함한다. 한편, 심도 맵은 이미지에서의 희소 개수의 포인트들보다 훨씬 많은 포인트들에 대해 상태 심도 값들을 제공한다. 일부 경우들에서, 이미지에서의 포인트들의 수는, 통상의 비디오 코딩 기법들을 이용하여 압축되는 심도 맵과 달리, 장면 구조 맵을 출력할 때 압축이 필요하지 않을 정도로 매우 적을 수도 있다. 또한, 보다 자세하게 설명된 바와 같이, 인코더 프로세서 (20) 는 현재 이미지를 인코딩하기 위하여 예측 이미지로서 이용되는 합성 이미지를 형성하도록 장면 구조 맵을 이용한다. 멀티뷰 비디오 코딩에서의 심도 맵은 이러한 목적에 이용되지 않는다. 즉, 심도 맵의 형태는 장면 구조 맵과 달리, 심도 맵을 본 개시에 설명된 기법들에 이용 가능하지 않게 할 수도 있고, 심도 맵의 데이터 양은 장면 구조 맵에 비해 추가적인 대역폭을 필요로 할 수도 있다.

[0058] 인코더 프로세서 (20) 는 합성 이미지를 형성하기 위해 장면 구조 맵을 이용한다. 예를 들어, 인코더 프로 세서 (20) 는 이전 이미지 또는 복수의 이전 이미지들 (예를 들어, 이전에 인코딩된 픽처들) 을 취출하여, (예를 들어, 이전에 인코딩된 이미지들을 블렌딩하는 것에 의해, 또는 복수의 이미지들이 이용되지 않으면 복합 이미지는 단일의 이미지일 수 있음) 복합 이미지를 형성한다. 인코더 프로세서 (20) 는 그 후, 메시를 형성하기 위해 장면 구조 맵의 포인트들을 상호접속하여, 메시 상에 복합 이미지를 오버레이한다. 예를 들어, 복합 이미지는 텍스처 맵으로서 고려될 수 있으며, 인코더 프로세서 (20) 는 그래픽 프로세싱 기법들을 이용하여 장면 구조 맵의 메시에 텍스처 맵을 오버레이한다. 그 결과는 장면 구조 맵의 메시 상에서의 복합 이미지의 3 차원 그래픽 구성인 이미지-기반 모델 (IBM) 이다. 인코더 프로세서 (20) 는 2 차원 이미지를 형성하기 위해 3 차원 그래픽 구성을 렌더링한다. 이 2차원 이미지는 합성 이미지이다.

- [0059] 인코더 프로세서 (20)는 예측 이미지로서 합성 이미지를 이용하며, 잔차 이미지로서 지칭되는, 합성 이미지와 현재 이미지 사이의 차이를 결정한다. 인코더 프로세서 (20)는 잔차 이미지에 대한 추가적인 (옵션적인) 인코딩을 수행할 수도 있고, 잔차 이미지를 나타내는 정보 (예를 들어, 잔차 이미지가 결정되게 할 수 있는 정보)를 출력할 수도 있다. 일부 예들에서, 인코더 프로세서 (20)는 또한 장면 구조 맵을 출력할 수도 있거나, 또는 이전 장면 구조 맵에 관련하여 장면 구조 맵의 충분적 변화들을 출력할 수도 있다. 장면 구조 맵에서의 정보의 양 또는 충분적 변화들은 비교적 작을 수도 있으며, 이러한 정보의 추가적인 압축은 이러한 정보를 추가로 압축하는 것을 가능하게 하더라도 필요하지 않을 수도 있다. 장면 구조 맵 또는 충분적 변화들을 출력하는 것은 모든 샘플에 대하여 필요한 것은 아닐 수도 있다. 예를 들어, 인코더 프로세서 (20)는 이전 장면 구조 맵을 재사용하여 합성 이미지를 구성할 수도 있고, 이 경우에, 인코더 프로세서 (20)는 장면 구조 맵, 또는 장면 구조 맵에 대한 충분적 변화들을 출력하지 않을 수도 있다.
- [0060] 일부 예들에서, 인코더 프로세서 (20)는 카메라 포즈 정보로 지칭되는 카메라 포지션 및/또는 카메라 배향 정보를 비디오 소스 (18)로부터 수신할 수도 있다. 일부 예들에서, 인코더 프로세서 (20)는 카메라 포지션을 결정하기 위해 SLAM 기법들을 이용할 수도 있다. 그러나, 카메라 포지션 및/또는 카메라 포즈를 결정하는 다른 기법들, 이를 테면, 외부 트랙킹 시스템들이 이용될 수도 있다. 인코더 프로세서 (20)는 현재 이미지를 인코딩하기 위하여 카메라 포즈 정보를 이용할 수도 있다. 예를 들어, 이미지에 대한 카메라 포즈 정보는 이미지가 촬영되었던 배향 및 포지션으로서 고려될 수도 있다.
- [0061] 인코더 프로세서 (20)는 장면 구조 맵 상에 복합 이미지를 오버레이하는 방법을 결정하기 위하여 복합 이미지를 구성하는데 이용되는 하나 이상의 이전에 인코딩된 이미지들에 대한 카메라 포즈 정보를 이용할 수도 있다. 예를 들어, 하나 이상의 이전에 인코딩된 이미지들의 카메라 포즈 정보에 의해 표시되는 관점에 기초하여, 인코더 프로세서 (20)는 이미지-기반 모델 (IBM)을 형성하기 위해 장면 구조 맵 상에 특정 픽셀이 어디에 배치되어야 하는지를 결정할 수도 있다. 예를 들어, 카메라 포즈 정보가, 하나 이상의 이전에 인코딩된 이미지들에 대한 카메라의 포지션 및 배향이 똑바른 (straight) 것임을 표시하면, 인코더 프로세서 (20)가 픽셀을 장면 구조 맵에 맵핑하는 로케이션은, 카메라의 포지션 및 배향이 비스듬함을 카메라 포즈 정보가 표시하는 것과는 상이하다. 이는 픽셀을 포함하는 오브젝트가 뷰잉 각도에 기초하여 상이한 상대적 로케이션들에서 보여지기 때문이다.
- [0062] 인코더 프로세서 (20)는 또한, 현재 이미지에 대한 카메라 포즈 정보를 이용할 수도 있다. 예를 들어, 위에 설명된 바와 같이, 합성 이미지는 예측 이미지를 형성한다. 잔차 데이터를 감소시키기 위해 (즉, 합성 이미지와 현재 이미지 사이의 차이를 최소화하기 위해), 인코더 프로세서 (20)는 합성 이미지의 카메라 포즈와 현재 이미지의 카메라 포즈가 동일하게 되도록 현재 이미지의 카메라 포즈 정보에 기초하여 이미지-기반 모델 (IBM)을 렌더링할 수도 있다.
- [0063] 일부 예들에서, 인코더 프로세서 (20)는 각각의 이미지에 대한 카메라 포즈 정보를 출력할 수도 있어, 디코더 프로세서 (30)가 현재 이미지를 디코딩하기 위한 것이지만 인코더 프로세서 (20)와 유사한 방식으로 카메라 포즈 정보를 이용할 수도 있다. 일부 예들에서, 인코더 프로세서 (20)는 선택적으로 출력될 필요가 있는 정보의 양을 감소시키기 위해 출력하기 전에 카메라 포즈 정보를 선택적으로 압축한다.
- [0064] 카메라 포즈 정보는 비디오 전화 또는 화상 회의에서와 같이, 카메라가 이동 카메라 (즉, 비디오 시퀀스에서의 하나 이상의 이미지들에 대한 카메라 포즈 정보가 상이하다)인 예들에서 이용가능할 수도 있다. 그러나, 일부 경우들에서, 카메라 포즈는 일정할 수도 있다. 이러한 예들에서, 인코더 프로세서 (20)는 각각의 이미지에 대한 카메라 포즈 정보를 반드시 출력하는 것이 필요한 것은 아니고 카메라 포즈 정보를 한번 출력할 수도 있다. 일부 예들에서, 인코더 프로세서 (20) 및 디코더 프로세서 (30)는 각각 특정 카메라 포즈 정보로 사전 구성될 수도 있고, 이 경우, 인코더 프로세서 (20)는 카메라 포즈 정보를 출력하지 않을 수도 있다. 카메라 포즈 정보가 변화하지 않는 예들에서도, 인코더 프로세서 (20)는 이미지를 중 하나 이상의 이미지들에 대한 카메라 포즈 정보를 여전히 송신할 수도 있다.
- [0065] 비디오 데이터, 이를 테면, 잔차 데이터, 장면 구조 맵 정보 및/또는 카메라 포즈 정보를 추가로 압축하기 위하여, 인코더 프로세서 (20)는 하나 이상의 예시적인 기법들을 이용할 수도 있다. 일 예로서, 잔차 데이터는 이미지로 고려될 수도 있다. 인코더 프로세서 (20)는 블록 기반 인코딩과 같은 비디오 압축 기법들을 이용하여 잔차 데이터를 압축할 수도 있다. 다른 예로서, 인코더 프로세서 (20)는 장면 구조 맵 정보 및/또는 카메라 포즈 정보에 대한 블록 기반 인코딩의 부분으로서 엔트로피 인코딩 기법들, 이를 테면, CAVLC (context-adaptive variable length coding), CABAC (context-adaptive binary arithmetic coding), SBAC (syntax-

based context-adaptive binary arithmetic coding), PIPE (Probability Interval Partitioning Entropy) 코딩 또는 다른 엔트로피 인코딩 방법론을 이용할 수도 있다. 엔트로피 인코딩 기법들은 단지 일 예로서만 설명된다. 다른 인코딩 기법들이 가능하다.

[0066] 일부 예들에서, 인코더 프로세서 (20)는 현재 이미지의 비디오 데이터를 세그먼트할 수도 있다. 예를 들어, 인코더 프로세서 (20)는 현재 이미지의 어느 부분들이 이전 이미지들 (예를 들어, 이전에 인코딩된 이미지들)에 대해 정적인지를 결정할 수도 있고, 현재 이미지의 어느 부분들이 변화하는지 (예를 들어, 비정적인지)를 결정할 수도 있다. 인코더 프로세서 (20)는 정적 부분으로부터 현재 이미지의 비정적 부분을 분리할 수도 있다 (예를 들어, 비정적 부분은 정적 부분을 포함하는 계층과는 상이한 계층이다). 인코더 프로세서 (20)는 비정적 부분에 대한 통상의 비디오 코딩 기법들을 적용하는 한편 정적 부분에 대하여 본 개시에 설명된 기법들을 이용할 수도 있다. 그러나, 인코더 프로세서 (20)에 대해, 정적 부분 및 비정적 부분 양쪽에 대하여 본 개시에 설명된 기법들을 이용하는 것도 가능할 수도 있다.

[0067] 현재 이미지의 어느 부분들이 정적인지 또는 어느 부분들이 비정적인지를 결정하는 여러 방법들이 있을 수도 있다. 일 예로서, 비디오 소스 (18)는 장면을 먼저 캡처할 수도 있고 인코더 프로세서 (20)는 배경의 이미지-기반 모델 (IBM)을 생성할 수도 있다. 비디오 소스 (18)는 그 후, 이전에 기록된 룸을 통과하는 사람과 같은 실제 장면을 캡처할 수도 있다. 인코더 프로세서 (20)는 2 개의 비디오 시퀀스들 사이에서의 (예를 들어, 배경 비디오 시퀀스와 이동하는 오브젝트들 (이 예에서는 사람)을 갖는 시퀀스 사이에서의) 차이를 결정할 수도 있다. 그 후, 인코더 프로세서 (20)는 비정적 부분들과 정적 부분들을 결정할 수도 있다. 이 예에서, 위와 유사하게, 인코더 프로세서 (20)는 개별적인 IBM들로부터 합성 이미지들을 렌더링하기 위하여 비정적 및 정적 이미지들의 기록 프로세스 (비디오 캡처 프로세스)에서 비디오 소스 (18)로부터 카메라 포즈 정보를 수신할 수도 있다.

[0068] 디코더 프로세서 (30)는 인코더 프로세서 (20)의 상호 프로세스를 수행한다. 그러나, 디코더 프로세서 (30)는 장면 구조 맵을 구성하는 것을 필요로 하지 않는다. 다만, 디코더 프로세서 (30)는 현재 이미지에 대한 장면 구조 맵을 수신하거나 또는 이전 장면 구조 맵에 관련하여 현재 이미지에 대한 장면 구조 맵에 대한 충분적 변화들을 수신한다. 일부 예들에서, 디코더 프로세서 (30)는 장면 구조 맵의 정보를 수신하지 않고, 이전 장면 구조 맵 (예를 들어, 이전에 디코딩된 이미지에 대하여 생성된 장면 구조 맵)을 재사용한다.

[0069] 일 예로서, 디코더 프로세서 (30)는 잔차 데이터를 디코딩할 수도 있다. 예를 들어, 잔차 데이터가 블록 기반 인코딩되면, 디코더 프로세서 (30)는 잔차 데이터를 블록 기반 디코딩할 수도 있다. 잔차 데이터는 또한 다른 방식들로 인코딩될 수도 있다. 일부 예들에서, 잔차 데이터는 인코딩되지 않을 수도 있고, 이 경우에 디코더 프로세서 (30)는 잔차 데이터를 디코딩하지 않는다. 어느 경우에도, 디코더 프로세서 (30)는 잔차 데이터 (예를 들어, 인코딩된 잔차 데이터 또는 인코딩되지 않은 잔차 데이터)를 수신한다.

[0070] 추가로, 디코더 프로세서 (30)는 현재 이미지에 대한 카메라 포즈 정보를 수신하고, 이러한 정보가 인코딩되면, 카메라 포즈 정보를 디코딩할 수도 있다. 위에 설명된 바와 같이, 디코더 프로세서 (30)는 장면 구조 맵 또는 이전 장면 구조 맵에 관련하여 현재 이미지에 대한 장면 구조 맵에 대한 충분적 변화들을 또한 수신할 수도 있다. 일부 예들에서, 장면 구조 맵 또는 충분적 변화들의 정보는 인코딩되거나 달리 압축되지 않을 수도 있고, 디코더 프로세서 (30)는 장면 구조 맵 또는 충분적 변화들에 대한 정보를 디코딩하는 것이 필요하지 않을 수도 있다.

[0071] 인코더 프로세서 (20)와 유사하게, 디코더 프로세서 (30)는 합성 이미지가 예측 이미지인 경우 장면 구조 맵과 복합 이미지를 이용하여 합성 이미지를 생성한다. 디코더 프로세서 (30)가 카메라 포즈 정보를 수신하는 예들에서, 디코더 프로세서 (30)는 합성 이미지를 생성하기 위하여 카메라 포즈 정보를 이용할 수도 있다. 예를 들어, 인코더 프로세서 (20)와 마찬가지로, 디코더 프로세서 (30)는 복합 이미지를 형성하기 위해 하나 이상의 이전 이미지들 (예를 들어, 이전에 디코딩된 이미지들)을 블렌딩할 수도 있거나 또는 단일의 이전에 디코딩된 이미지가 복합 이미지일 수도 있다. 디코더 프로세서 (30)는 IBM을 형성하기 위해 장면 구조 맵에 (그리고 선택적으로 카메라 포즈 정보를 이용하여) 복합 이미지를 맵핑할 수도 있고, 합성 이미지를 생성하기 위해 IBM을 렌더링할 수도 있다. 디코더 프로세서 (30)는 현재 이미지를 재구성하기 위해 합성 이미지에 잔차 데이터를 추가할 수도 있다.

[0072] 인코더 프로세서 (20) 및 디코더 프로세서 (30)는 코더 프로세서로서 일반적으로 지칭될 수도 있다. 예를 들어, 인코더 프로세서 (20) 및 디코더 프로세서 (30)는 현재 이미지에 대한 예측 이미지를 형성하는 합성 이미지를 생성하도록 구성될 수도 있다. 합성 이미지를 생성하기 위해, 인코더 프로세서 (20) 및 디코더 프로

세서 (30)는 유사한 기능들을 수행하도록 구성될 수도 있고, 용이성을 위해 이들 기능들은 코더 프로세서에 의해 수행되는 것으로 설명된다 (이 코더 프로세서의 예들은 인코더 프로세서 (20) 및 디코더 프로세서 (30)를 포함한다).

[0073] 예를 들어, 코더 프로세서는 메모리에 저장된 이전 복합 이미지 및 장면 구조 맵에 기초하여 합성 이미지를 생성하도록 구성될 수도 있다. 장면 구조 맵은 현재 이미지의 장면 구조 맵 또는 이전에 코딩된 이미지의 장면 구조 맵을 포함하고, 장면 구조 맵은 현재 이미지 또는 이전에 코딩된 이미지 내에서 3 차원 포인트들에 대한 좌표 값들을 포함한다.

[0074] 코더 프로세서는 현재 이미지의 잔차 이미지에 기초하여 현재 이미지를 코딩하도록 구성될 수도 있고, 여기에서 잔차 이미지는 현재 이미지와 합성 이미지 사이의 차이를 나타낸다. 예를 들어, 코더 프로세서가 디코더 프로세서 (30) 인 경우에, 디코더 프로세서 (30)는 수신된 잔차 이미지와 합성 이미지에 기초하여 현재 이미지를 재구성하는 것에 의해 현재 이미지를 디코딩할 수도 있다. 코더 프로세서가 인코더 프로세서 (20) 인 경우, 인코더 프로세서 (20)는 잔차 이미지를 결정하고 잔차 이미지의 정보를 출력하는 것에 의해 현재 이미지를 인코딩할 수도 있다.

[0075] 일부 다른 기법들이 비디오 코딩 목적들을 위한 SLAM 기법들을 이용하여 제안되어 있다. 예를 들어, 이들 다른 기법들에서, 코더 프로세서 (인코더 프로세서 (20) 및 디코더 프로세서 (30) 과 달리)는 밀도가 희소한 포인트들에 기초하기 보다는 매우 세부적인 장면 구조 맵을 결정한다. 이들 다른 기법들의 세부적인 장면 구조 맵은, 세부적인 구조 맵 상의 이전 이미지가 현재 이미지와 동일한 결과를 가져오도록 세부적으로 이루어질 수도 있다.

[0076] 이들 다른 기법들에서, 세부적인 장면 구조 맵에 대한 포인트 좌표들이 출력되어 수신되지만 어떠한 잔차 데이터도 전송되거나 또는 수신되지 않는다. 이는 세부적인 장면 구조 맵 상에 이전 이미지를 오버레이하는 것 이 기본적으로 현재 이미지와 동일한 이미지를 가져오기 때문이다. 따라서, 이들 다른 기법들은 어떠한 잔차 데이터에도 의존하지 않는다. 이들 다른 기법들에서, 합성 이미지는 예측 이미지가 아닌, 현재 이미지의 복사본으로 고려된다.

[0077] 이들 다른 기법들에는 특정 결합들이 존재할 수도 있다. 예를 들어, 세부적인 장면 구조 맵에 대한 좌표들을 출력하고 수신하는 것은 대역폭을 희생시킬 수도 있다. 또한, (예를 들어, 이전 이미지가 세부적인 장면 구조 맵 상에 오버레이된 경우) 결과적인 이미지는 현재 이미지에 대한 매칭이 양호하지 않을 수도 있다. 따라서, 이미지 품질은 다른 비디오 코딩 기법들보다 안 좋을 수도 있다.

[0078] 본 개시에 설명되는 기법들에서, 코더 프로세서는 현재 이미지 또는 이전에 코딩된 이미지에서의 소수의 포인트들에 기초하여 합성 이미지를 생성하고 이 합성 이미지는 예측 이미지를 형성한다. 따라서, 소수의 포인트들이 출력되고 수신되는데 필요할 수도 있고, 제한된 데이터가 출력되고 수신되기에 필요한 정도로, 합성 이미지와 현재 이미지 사이의 차이가 충분히 작을 수도 있다 (예를 들어, 잔차 이미지가 비교적 작은 양의 데이터를 갖는다). 이러한 식으로, 송신 및 수신하는데 필요한 데이터의 양에서의 감소가 있을 수도 있어, 대역폭 효율성을 촉진한다. 또한, 합성 이미지가 예측 이미지이고 그리고 잔차 이미지가 송신되고 수신되기 때문에, 재구성된 현재 이미지는 어떠한 잔차 이미지도 전송되지 않은 예보다 더 우수한 품질로 이루어질 수도 있다.

[0079] 도 2 는 본 개시에서 설명된 하나 이상의 예시적인 비디오 인코딩 기법들을 구현하거나 또는 달리 이용하도록 구성되는 인코더 프로세서 (20)의 일 예를 예시하는 블록도이다. 예시된 바와 같이, 인코더 프로세서 (20)는 동시 로컬리제이션 및 맵핑 (simultaneous localization and mapping; SLAM) 프로세서 (34), 그래픽 프로세싱 유닛 (graphics processing unit; GPU)(36), 비디오 메모리 (38) 및 인코더 (48)를 포함한다. 일부 예들에서, SLAM 프로세서 (34), GPU (36), 및 인코더 (48)는 함께 단일의 칩으로 형성되어 시스템 온 칩 (system on chip; SoC) 을 형성할 수도 있다. 일부 예들에서, SLAM 프로세서 (34), GPU (36), 및 인코더 (48)는 별도의 칩들 상에 형성될 수도 있다.

[0080] 인코더 프로세서 (20)는 임의의 다양한 적절한 인코더 회로부, 이를 테면, 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSPs), 주문형 반도체들 (ASICs), 필드 프로그램가능 게이트 어레이들 (FPGAs), 이산 로직, 소프트웨어, 하드웨어, 폼웨어 또는 이들의 임의의 조합들로서 구현될 수도 있다. 예를 들어, 본 개시에 설명된 기법들은 장치 또는 디바이스의 관점으로부터 설명될 수도 있다. 일 예로서, 장치 또는 디바이스는 인코더 프로세서 (20)(예를 들어, 무선 통신 디바이스의 부분으로서의 소스 디바이스 (12))를 포함할 수도 있고, 인코더 프로세서 (20)는 본 개시에 설명된 기법들을 구현하도록 구성되는 하나 이상의 프로세서들을

포함할 수도 있다. 다른 예로서, 장치 또는 디바이스는 인코더 프로세서 (20)를 포함하는 집적 회로 (IC) 또는 마이크로프로세서를 포함할 수도 있고, 마이크로-프로세서 또는 IC는 소스 디바이스 (12) 또는 다른 유형의 디바이스의 부분일 수도 있다.

[0081] 인코더 프로세서 (20)의 컴포넌트들은 소스 디바이스 (12)의 기존의 예들의 부분일 수도 있다. 예를 들어, 하드웨어의 부분으로서, 소스 디바이스들 (예를 들어, 소스 디바이스 (12))의 대부분의 예들은 GPU, 이를 테면, GPU (36), 인코더, 이를 테면, 인코더 (48), 및 비디오 메모리, 이를 테면 비디오 메모리 (38)를 포함한다. 디바이스가 SLAM 프로세서 (34)를 포함하지 않은 경우에도, 디바이스의 중앙 프로세싱 유닛 (central processing unit; CPU) 가 SLAM 프로세서 (34)로서 기능하도록 구성될 수도 있다. 예를 들어, 디바이스의 CPU 상의 소프트웨어 또는 펌웨어 또는 호스트 프로세서를 실행하는 것에 의해, CPU 또는 호스트 프로세서는 SLAM 프로세서 (34)의 기능을 충족하도록 구성될 수도 있다.

[0082] 추가로, GPU (36)는 본 개시에 설명된 예시적인 기법을 구현하도록 구성되는 일 예시적인 컴포넌트로서 예시된다. GPU (36)의 고속 병렬 프로세싱 능력들은 GPU (36)가 본 개시에 설명된 기법들을 구현하기 위한 적절한 옵션을 행하게 한다. 그러나, 본 기법들은 이것으로 제한되지 않는다. CPU를 포함하는 예시적인 기법들을 구현하기 위해 GPU (36) 외의 다른 컴포넌트들을 이용하는 것도 가능할 수도 있다. 또한, 일부 GPU들은 범용 GPU들 (general purpose GPUs; GPGPUs)로서 기능하는 프로세싱 능력을 포함한다. 일부 예들에서, GPU (36)는 GPGPU 일 수도 있고, SLAM 프로세서 (34)의 기능들을 수행하도록 구성될 수도 있다 (즉, SLAM 프로세서 (34) 및 GPU (36)는 동일한 GPGPU의 부분일 수도 있다).

[0083] 비디오 메모리 (38)는 인코더 프로세서 (20)를 형성하는 칩의 부분일 수도 있다. 일부 예들에서, 비디오 메모리 (38)는 인코더 프로세서 (20)에 대해 외부에 있을 수도 있고 소스 디바이스 (12)의 시스템 메모리의 부분일 수도 있다. 일부 예들에서, 비디오 메모리 (38)는 시스템 메모리와 인코더 프로세서 (20)의 내에 있는 메모리의 조합일 수도 있다.

[0084] 비디오 메모리 (38)는 현재 이미지를 인코딩하는데 이용되는 데이터를 저장할 수도 있다. 예를 들어, 예시된 바와 같이, 비디오 메모리는 현재 이미지에 대하여 SLAM 프로세서 (34)에 의해 생성되면, 장면 구조 맵 (40)을 저장하고, 이전 장면 구조 맵 (42) (예를 들어, 이전에 인코딩된 이미지에 대하여 SLAM 프로세서 (34)가 생성한 장면 구조 맵), 하나 이상의 이미지들 (44) (예를 들어, 이전에 인코딩된 이미지들) 및 GPU (36)가 생성한 합성 이미지 (46)를 저장한다. 비디오 메모리 (38)는 다양한 메모리 디바이스들, 이를 테면 동기식 DRAM (synchronous DRAM; SDRAM)을 포함하여 동적 랜덤 액세스 메모리 (dynamic random access memory; DRAM), 자기 저항성 RAM (magnetoresistive RAM; MRAM), 저항성 RAM (resistive RAM; RRAM), 또는 다른 유형의 메모리 디바이스들 중 임의의 것에 의해 형성될 수도 있다.

[0085] 도 2에 예시된 바와 같이, SLAM 프로세서 (34)는 현재 이미지를 수신하고 장면 구조 맵 (40)을 생성한다. SLAM 프로세서 (34)는 현재 이미지로부터 밀도가 희소한 3D 포인트 클라우드 재구성을 추출하도록 구성되고, 일부 경우에 스크린 장면에서 재구성된 포인트들의 밸런싱된 분포를 보장할 수도 있다.

[0086] SLAM 기술들은 지오메트릭 서라운딩 맵을 생성하기 위해 센서들의 피드백에 의존하는 지오메트릭 서라운딩 맵을 생성하기 위하여 자율 차량들 및 로봇들에 통상 이용된다. SLAM 프로세서 (34)는 지오메트릭 서라운딩 맵을 생성하기 위해 비주얼 SLAM (visual SLAM; VSLAM) 센싱 기법들을 이용하도록 구성될 수도 있다. 예를 들어, SLAM 프로세서 (34)는 비디오 소스 (18)로부터 캡처된 이미지들을 수신하고, 지오메트릭 서라운딩 맵을 생성하도록 SLAM 기법들을 수행할 수도 있다.

[0087] 지오메트릭 맵을 생성함에 있어서, SLAM 프로세서 (34)는 비디오 소스 (18)의 3D 카메라 포즈를 트랙킹 할 수도 있다. 예를 들어, 사용자가 모바일 디바이스를 갖고 걷는 비디오 전화와 같은 예들에서 또는 카메라 포지션 및 배향이 제어되고 있는 비디오 회의와 같은 예들에서, 카메라는 이동하는 경향이 있지만 장면은 대부분 정적이다. 카메라 로케이션이 변화하기 때문에, 이미지가 캡처되는 관점도 또한 변화한다. 카메라의 3D 포지션 및 배향 (포즈)을 트랙킹하는 것에 의해, SLAM 프로세서 (34)는 지오메트릭 서라운딩 맵을 보다 우수하게 구성 가능할 수도 있다. 일부 예들에서, SLAM 프로세서 (34)는 카메라 포지션을 결정하기 위해 SLAM 기법들을 이용할 수도 있다.

[0088] 일부 경우들에서, SLAM 기법들은 증강 현실 (augmented reality; AR)에 이용된다. 예를 들어, 그래픽 비디오 또는 이미지들은 배경 영역 상에 오버레이된다. SLAM 프로세서 (34)는 SLAM 기법을 이용하여 장면 구조를 맵핑하고 AR 비디오는 (예를 들어, AR 비디오가 오버레이된) 변경된 비디오 스트림을 보여준다. AR을

위해 이용되고 있는 SLAM 의 전형적인 예는 영화 "Star Wars: Episode IV-A New Hope" 로부터의 장면이며, 여기에서 로봇 R2D2 가 Princess Leia 의 영화 녹화를 프로젝트한다.

[0089] 지오메트릭 서라운딩 맵을 생성하기 위해, SLAM 프로세서 (34) 는 장면 구조 맵을 구성한다. 위에 설명된 바와 같이, 장면 구조 맵은 현재 이미지에서의 회소 개수의 포인트들에 대한 (x, y, z) 좌표들을 포함한다.

현재 이미지들에서의 포인트들에 대한 x, y, z 좌표들은 카메라 포즈에 기초할 수도 있고, 주어진 카메라 포즈에 대한 포인트들 각각의 상대 로케이션들을 표시한다. 예를 들어, z-좌표의 값은 뷰어가 그 값을 어느 정도 멀리 인식하는지를 나타내는 특정 포인트의 상대 심도를 표시할 수도 있다. 또한, 현재 이미지는 2D 영역을 수반하지만, 이미지의 컨텐츠는 z-좌표에 의해 표시되는 포인트들의 상대 심도들을 포함한다.

[0090] 도 3a 및 도 3b 는 장면 구조 맵들의 예들을 예시하는 그래픽도들이다. 예를 들어, 도 3a 및 도 3b 는 상이한 이미지들을 예시한다. 도 3a 및 도 3b 에 예시된 포인트들은 이들 이미지들의 장면 구조 맵을 형성한다.

예를 들어, 도 3a 및 도 3b 는 y-좌표의 특정 값에 대하여 x-방향 및 z-방향으로 확장하는 그리드를 예시한다. 이 그리드 상에 존재하는 포인트들은 자신들의 x, y, 및 z 좌표들에 의해 정의된다.

[0091] SLAM 프로세서 (34) 는 비디오 메모리 (38) 에 생성된 장면 구조 맵을 저장한다 (예를 들어, 비디오 메모리 (38) 에 장면 구조 맵 (40) 을 저장한다). 그러나, SLAM 프로세서 (34) 는 각각의 이미지에 대하여 장면 구조 맵을 반드시 생성할 필요가 있는 것은 아니다. 예시된 바와 같이, 비디오 메모리 (38) 는 또한 이전 장면 구조 맵 (42) 을 저장할 수도 있다. SLAM 프로세서 (34) 는 이전 이미지 (예를 들어, 현재 이미지에 대하여 이전에 인코딩된 이미지) 에 대한 장면 구조 맵을 생성할 수도 있었다. 이전 이미지가 생성되었을 때의 시간에, SLAM 프로세서 (34) 는 비디오 메모리 (38) 에 결과적인 장면 구조 맵을 저장할 수도 있었다. 그 후, 현재 이미지에 대하여, 이전에 생성된 장면 구조 맵은 이전 장면 구조 맵 (42) 으로 된다.

[0092] 본 개시에 설명된 기법들에 따르면, GPU (36) 는 장면 구조 맵 (40)(이용가능한 경우) 또는 이전 장면 구조 맵 (42)(장면 구조 맵 (40) 이 이용가능하지 않은 경우) 중 하나를 취출할 수도 있다. GPU (36) 는 프록시 지오메트리를 형성하기 위해 장면 구조 맵의 포인트들을 상호접속할 수도 있다. 예를 들어, 장면 구조 맵에 의해 정의되는 포인트들은 다각형들 (예를 들어, 삼각형들) 의 베텍스들로서 고려될 수도 있다. GPU (36) 는 장면 구조 맵의 포인트들을 상호접속하기 위해 하드웨어된 입력 어셈블러를 이용할 수도 있거나 또는 베텍스 쉐이더를 실행할 수도 있다. 일 예로서, GPU (36) 는 포인트들을 상호접속하기 위해 충분적 Delauney 삼각측량 기법들을 이용할 수도 있다.

[0093] 포인트들의 상호접속의 결과는 장면 구조 맵을 생성하는데 이용되는 이미지의 구조를 표현하는 프록시 지오메트리이다. 개념적으로, 프록시 지오메트리에서의 포인트들의 상호접속은 장면 구조 맵을 생성하는데 이용되는 이미지의 상대 3 차원 구조의 대략적인 추정값을 형성하는 삼각형 메시를 형성한다. 예를 들어, 포인트들의 상호접속은 심도의 근사화를 제공하며, 실제 심도 맵보다 훨씬 더 밀도가 회소하다. 그러나, 삼각형 메시를 형성하기 위한 포인트들의 상호접속은 아래 보다 자세하게 설명된 바와 같이, 예측 이미지를 생성하기 위해 복합 이미지를 오버레이하기에 충분한 세부성을 제공할 수도 있다.

[0094] 도 4 는 장면 구조 맵의 포인트들의 상호접속의 일 예를 예시하는 개념도이다. 예를 들어, 도 4 는 예시된 바와 같이, 프록시 지오메트리를 형성하는 복수의 삼각형들을 포함하고, 삼각형들의 베텍스들은 장면 구조 맵에 의해 정의된다. 베텍스들 각각은 x, y, 및 z 좌표에 의해 정의될 수도 있다. 예를 들어, 예시된 프록시 지오메트리는 소수의 3D 포인트들 및 3D 삼각형들을 포함한다. 일부 예들에서, GPU (36) 는 (예를 들어, 베텍스 쉐이더 또는 입력 어셈블러에서 통해) 프록시 지오메트리를 생성하도록 구성될 수도 있다. GPU (36) 는 프록시 지오메트리를 충분적으로 구축할 수도 있고 (아래 설명된 바와 같이 오직 합성 이미지만을 생성하기 위하여) 프록시 지오메트리를 임시적으로 저장할 수도 있다.

[0095] 이전 장면 구조 맵 (42) 및/또는 장면 구조 맵 (40) 을 저장하는 것에 더하여, 비디오 메모리 (38) 는 또한, 하나 이상의 이미지들 (44)(예를 들어, 이전에 인코딩된 이미지들), 및 카메라가 이동하는 예들에서와 같은 이들의 대응하는 카메라 포즈 정보를 저장할 수도 있다. 하나 이상의 이미지들 (44) 은 인코더 프로세서 (20) 가 이전에 인코딩하였던 이미지들이다. 비디오 메모리 (38) 는 인코딩 프로세서 (20) 가 이전에 인코딩한 오리지널 이미지들을 저장하거나 또는 인코더 (48) 는 인코딩된 이미지를 재구성하기 위해 피드백 경로를 포함하고 비디오 메모리 (38) 는 하나 이상의 이미지들 (44) 로서 이를 재구성된 이미지들을 저장한다. 일부 예들에서, 하나 이상의 이미지들 (44) 은 현재 이미지보다 더 조기에 디스플레이될 수도 있지만, 하나 이상의 이미지들 (44) 이 현재 이미지보다 더 조기에 디스플레이되는 것은 필수요건이 아니며, 하나 이상의 이미지들 (44) 은 현재 이미지 뒤에 또는 일부 경우들에서 (스테레오스코픽 뷰에 대해) 현재 이미지와 동시에 가상으로

디스플레이될 수도 있다.

[0096] 일부 예들에서, 하나 이상의 이미지를 (44)(예를 들어, 이전에 인코딩된 이미지들)은 I-프레임들로서 또한 지칭되는 키프레임들일 수도 있다. 예를 들어, 일부 예들에서, 인코딩 어려들을 전파하는 것을 회피하기 위하여, 인코더 프로세서 (20)는 다른 예측 이미지에 대한 참조 없이 이미지를 때때로 인코딩할 수도 있다. 예를 들어, 인코더 (48)는 임의의 다른 이미지로부터의 값들로부터가 아닌 이미지 내에서의 픽셀들의 샘플 값들을 이용하여 이미지를 인코딩할 수도 있다. 이미지 내의 샘플들만을 참조로 인코딩된 이미지들은 I-프레임들 또는 키프레임들로서 지칭된다. 다른 이미지들로부터 예측된 이미지는 P-프레임들로 지칭된다.

[0097] 하나 이상의 이미지를 (44)이 모든 예에서 반드시 키프레임들일 필요가 있는 것은 아니다. 그러나, 예 및 예시의 목적을 위하여, 본 기법들은 하나 이상의 이미지를 (44)이 키프레임들인 것에 대하여 설명하였다.

[0098] 추가로, 본 개시에 설명된 기법들은 복수의 키프레임들 (예를 들어, 복수의 이미지를 (44))을 이용할 필요는 없고 단일의 키프레임 (예를 들어, 단일의 이미지 (44))을 이용할 수도 있다. 복수의 이미지를 (44)이 이용되는 예들에서, GPU (36)는 복합 이미지를 형성하기 위해 둘 이상의 이미지를 (44)에 블렌딩 동작을 수행할 수도 있다. 예를 들어, GPU (36)는 픽셀 값들을 블렌딩하도록 특수하게 구성되는 자신의 그래픽 파이프라인에 블렌딩 유닛을 포함할 수도 있다. 복수의 이미지를 (44)은 블렌딩 유닛에 대한 입력일 수도 있고, 블렌딩 유닛의 출력은 블렌딩된 이미지일 수도 있다. 임의의 추가적인 그래픽 프로세싱 후에, 필요에 따라 GPU (36)의 출력은 복합 이미지이다. 예시되지 않았지만, GPU (36)는 비디오 메모리 (38)에 복합 이미지를 저장한다. 오직 하나의 이미지 (44)만이 이용되는 예들에서, GPU (36)는 블렌딩하지 않을 수도 있고 단일의 이미지 (44)가 복합 이미지이다.

[0099] 본 개시에 설명된 기법들에 따르면, 복합 이미지는 텍스처 맵이고, 프록시 지오메트리 (예를 들어, 장면 구조 맵의 포인트들을 상호접속하는 것에 의해 형성되는 메시)는 텍스처 맵이 맵핑되는 오브젝트이다. 예를 들어, 텍스처 맵핑에서, 텍스처 맵은 2 차원 좌표들 (u, v)에서 정의되고, 텍스처 맵이 맵핑되는 오브젝트는 3 차원들로 정의된다. GPU (36)의 기능들 중 한 기능은 그래픽 렌더링을 위하여 텍스처 맵핑을 수행하는 것일 수도 있고, 기법들은 GPU (36)의 능력을 활용하여 비디오 인코딩 및 디코딩을 위하여 예측 이미지로서 이용되는 합성 이미지를 생성하기 위해 고속의 그리고 상이한 방식으로 텍스처 맵핑을 수행한다.

[0100] 예를 들어, GPU (36)는 텍스처 기능들을 수행하기 위해 하드웨어로 사전 구성될 수도 있거나 또는 텍스처 엔진을 실행할 수도 있다. 어느 예에서도, GPU (36)는 그래픽 오브젝트 상에 텍스처 맵으로부터의 2 차원 포인트를 맵핑하여 본질적으로 텍스처 맵을 오브젝트 상에 오버레이한다. 텍스처링의 결과는 순수 그래픽 구성보다는 훨씬 실제감있게 보여지는 매우 세부적인 그래픽 오브젝트이다. 일 예로서, 오브젝트는 구이고, 텍스처 맵은 2D 이미지의 세계이다. 텍스처 맵핑의 부분으로서, 2D 이미지의 세계를 3 차원 구 상에 랙핑하는 것에 의해, GPU (36)는 지구를 그래픽적으로 구성하는 것에 비하여, 훨씬 더 세부적이고 시각적으로 양호한 지구를 렌더링 할 수도 있다.

[0101] 도 2에 예시된 예에서, GPU (36)는 복합 이미지로부터의 픽셀을 프록시 지오메트리 상의 3 차원 로케이션으로 맵핑한다. 위에 설명된 바와 같이, 일부 예들에서, GPU (36)는 복합 이미지를 구성하는데 이용되는, 하나 이상의 이미지를 (44)의 카메라 포즈 정보를 이용하여, 복합 이미지로부터의 픽셀을 프록시 지오메트리 상의 3 차원 로케이션에 맵핑한다 (예를 들어, 카메라 포즈 정보는 픽셀이 맵핑할 3 차원 로케이션에 영향을 준다). 복합 이미지의 컨텐츠가 상대 심도를 나타내고 있지만, 복합 이미지들의 픽셀들 모두는 2 차원 (u, v) 좌표들에 의해 정의된다. 프록시 지오메트리가 3차원이기 때문에, 텍스처 맵핑의 결과는 이미지-기반 모델 (IBM)로서 지칭되는 3 차원 오브젝트이다.

[0102] GPU (36)는 IBM에 대해 추가적인 그래픽 프로세스를 수행한다. 예를 들어, GPU (36)는 IBM을 렌더링 할 수도 있다. 렌더링 프로세서의 부분으로서, GPU (36)는 IBM의 3차원 좌표들을 합성 이미지 (46)를 형성하는 2 차원 스크린 좌표들로 변환한다. 일부 예들에서, GPU (36)는 현재 카메라 포지션에 기초하여 IBM을 렌더링 할 수도 있다. 예를 들어, 합성 이미지 (46)가 예측 이미지를 형성하기 때문에, GPU (36)는 현재 카메라 포즈 정보에 기초하여 IBM을 렌더링 할 수 있어, 합성 이미지 (46)와 현재 이미지 사이의 비교가 유효하게 되고 잔차 데이터가 최소화되게 된다.

[0103] GPU (36)는 비디오 메모리 (48)에 합성 이미지 (46)로서 렌더링하는 결과를 저장한다. 합성 이미지 (46)는 각각의 좌표에 대한 픽셀 값들을 갖는 2차원 이미지일 수도 있다. 본 개시에 설명된 기술들에서, 합성 이미지 (46)는 예측 이미지를 형성한다.

- [0104] 예를 들어, 인코더 (48) 가 합성 이미지 (46) 와 현재 이미지를 수신하고, 현재 이미지와 합성 이미지 사이의 잔차 이미지 (예를 들어, 잔차 데이터) 를 결정한다. 인코더 (48) 는 인코딩된 비디오 데이터로서 비트스트림으로 결과적인 잔차 이미지를 출력할 수도 있다. 일부 예들에서, 잔차 이미지를 출력하기 전에, 인코더 (48) 는 잔차 이미지의 엔트로피 인코딩 또는 블록 기반 인코딩과 같은 일부 추가적인 인코딩을 수행할 수도 있다. 이 방식으로, 인코더 (48) 가 잔차 이미지의 추가적인 인코딩을 수행하든 수행하지 않던 간에, 잔차 이미지를 나타내는 정보 (잔차 이미지가 결정되게 할 수 있는 정보) 를 출력할 수도 있다.
- [0105] 일부 예들에서, 인코더 (48) 는 본 개시에 설명된 기법들을 구현하도록 구성될 수도 있고 또한 통상의 비디오 인코딩 프로세스들을 구현하도록 구성될 수도 있다. 예를 들어, 인코더 (48) 는 이미지 내에서의 샘플들만을 기초하여 이미지를 인코딩 (예를 들어, I-프레임이도록 이미지를 인코딩) 하도록 구성될 수도 있다. 인코더 (48) 는 통상의 비디오 인코딩을 이용하여 이러한 비디오 인코딩 기법들을 수행할 수도 있다.
- [0106] 다른 예로서, 일부 예들에서, 현재 이미지에서의 비정적 전경 오브젝트들이 다른 인코딩 기법들을 이용하여 인코딩되고 정적 배경 오브젝트들이 본 개시에 설명된 기법들을 이용하여 인코딩되면 인코딩 이득들이 실현될 수도 있다. 예를 들어, 인코더 (48) 는 본 개시에 설명된 기법들을 이용하여 이미지의 부분들을 인코딩하는 것과, 다른 기법들을 이용하여 이미지의 다른 부분들을 인코딩하는 것 사이를 선택하는 모드 선택 유닛을 포함할 수도 있다. 본 개시에 설명되는 기법들을 이용하여 인코딩되는 부분들은 하나의 계층을 형성할 수도 있고 다른 기법을 이용하여 인코딩되는 부분들은 다른 계층을 형성할 수도 있다. 본 개시에 설명된 기법들을 이용하여 전체의 현재 이미지를 인코딩하는 것이 가능함을 이해해야 한다.
- [0107] 전경의 비정적 부분이 하나의 계층을 형성하고 배경의 정적 부분이 다른 계층을 형성하는 예들에서, SLAM 프로세서 (34) 는 전경에 대한 SLAM 기법들을 수행한다. 이러한 예들에서, 인코더 (48) 는 전경 계층과 배경 계층을 개별적으로 인코딩할 수도 있다.
- [0108] (도 2 에서 점선으로 예시된) 일부 예들에서, SLAM 프로세서 (34) 는 인코더 (48) 에 카메라 포즈 정보를 출력한다. 인코더 (48) 는 (예를 들어, 인코딩된 비디오 데이터의 부분으로서) 비트스트림으로 출력하기 위한 카메라 포즈 정보를 인코딩한다. 또한, 일부 예들에서, SLAM 프로세서 (34) 는 인코더 (48) 에 장면 구조 맵 (40) 의 정보를 출력할 수도 있고 인코더 (48) 는 장면 구조 맵 (40) 의 정보를 인코딩한다. 일부 예들에서, 인코더 프로세서 (20) 가 장면 구조 맵 (40) 을 이용할 때, 인코더 (48) 는 장면 구조 맵 (40) 과 이전 장면 구조 맵 (42) 사이의 차이를 결정할 수도 있고, 장면 구조 맵 (40) 의 실제 값들 보다는 (예를 들어, 장면 구조 맵에서의 충분적 변화들을 표현하는) 장면 구조 맵 (40) 과 이전 장면 구조 맵 (42) 사이의 차이를 결정할 수도 있다. 위에 나타낸 바와 같이, 인코더 프로세서 (20) 는 모든 예에서 장면 구조 맵 (40) 을 이용하는 것은 아닐 수도 있다. 이들 예들에서, 인코더 (48) 는 장면 구조 맵 (40) 과 이전 장면 구조 맵 (42) 사이의 차이 또는 장면 구조 맵 (40) 의 정보를 출력하지 않을 수도 있다. 이들 예들 중 일부에서, 인코더 (48) 는 디코더 프로세서 (30) 가 국부적으로 저장된 이전 장면 구조 맵을 이용해야 함을 표시하는 신택스 엘리먼트들을 출력할 수도 있다.
- [0109] 도 5 는 본 개시에 따라 하나 이상의 예시적인 비디오 인코딩 기법들을 예시하는 데이터 흐름도이다. 도 5 의 예에서 예시된 바와 같이, 인코더 프로세서 (20) 는 현재 이미지 (50) 를 수신한다. SLAM 프로세서 (34) 는 SLAM (52) 으로서 지칭되는 장면 구조 맵을 생성하기 위해 SLAM 기법들을 구현한다. GPU (36) 는 이미지-기반 모델 (54) 을 형성하기 위해 복합 이미지와 장면 구조 맵을 이용한다. GPU (36) 는 예측 이미지 (56) 인 합성 이미지 (46) 를 생성하기 위해 이미지-기반 모델 (54) 을 렌더링한다.
- [0110] 감산 유닛 (53) 은 현재 이미지 (50) 로부터 예측 이미지 (56) 를 감산한다. 감산의 결과는 잔차 이미지 (58) 이다. 위에 설명된 바와 같이, 인코더 (48) 는 도 5 에서의 인코딩 (60) 으로서 예시된 추가적인 인코딩을 수행할 수도 있다. 일부 예들에서, 현재 이미지 (50) 로부터 예측 이미지 (56) 를 감산하기 위해, 감산 유닛 (53) 은 예측 이미지 (56) 의 값들에 -1 을 곱하고 그 결과에 현재 이미지 (50) 를加산한다. 간략화를 위하여, 본 개시는 현재 이미지 (50) 로부터 예측 이미지 (56) 를 빼는 것으로서 감산 유닛 (53) 을 설명하며, 직접 감산 (straight subtraction) 에 의해, 이미지의 네가티브를 결정하고 (예를 들어, 값들에 -1 을 곱하고) 이를 다른 것에 加산하는 것에 의해 또는 일부 다른 기법에 의해 이러한 연산을 수행할 수도 있다.
- [0111] 인코더 (48) 는 인코딩된 비디오 데이터 비트스트림 (62) 의 부분으로서 결과적인 인코딩된 이미지를 출력한다. 추가적으로, 비트스트림 (62) 은 현재 이미지 (50) 에 대한 카메라 포즈 정보를 포함할 수도 있다. 비트스트림 (62) 은 또한 (예를 들어, 이전 장면 구조 맵에 관련하여 장면 구조 맵에서의 충분적 변화를 나타내는) 장면 구조 맵과 이전 장면 구조 맵 사이의 차이를 나타내는 정보 또는 장면 구조 맵의 정보를 또한 포함할 수도 있다.

있다. 그러나, 비트스트림 (62) 은 모든 예에서, 장면 구조 맵, 또는 장면 구조 맵과 이전 장면 구조 맵 사이의 차이를 반드시 포함할 필요가 있는 것은 아니다.

[0112] 도 6 은 본 개시에 따라 설명된 기법들에 따라 비디오 인코딩의 일 예를 예시하는 개념도이다. 도 6 에서, 이미지 (64) 는 현재 이미지이고 이미지 (66) 는 합성 이미지이다. 이미지 (68) 는 잔차 이미지 (예를 들어, 이미지 (64) 에서 이미지 (66) 를 뺀 것) 를 예시한다.

[0113] 도 7 은 본 개시에서 설명된 하나 이상의 예시적인 비디오 디코딩 기법들을 구현하거나 또는 달리 이용하도록 구성되는 디코더 프로세서 (30) 의 일 예를 예시하는 블록도이다. 예시된 바와 같이, 디코더 프로세서 (30) 는 GPU (70), 디코더 (80), 및 비디오 메모리 (82) 를 포함한다. 일부 예들에서, GPU (70), 및 디코더 (80) 는 함께 단일의 칩으로 형성되어 시스템 온 칩 (SoC) 을 형성할 수도 있다. 일부 예들에서, GPU (70) 및 디코더 (80) 는 별도의 칩들 상에 형성될 수도 있다.

[0114] 디코더 프로세서 (30) 는 임의의 다양한 적절한 디코더 회로부, 이를 테면, 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSPs), 주문형 반도체들 (ASICs), 필드 프로그램가능 게이트 어레이들 (FPGAs), 이산 로직, 소프트웨어, 하드웨어, 펌웨어 또는 이들의 임의의 조합들로서 구현될 수도 있다. 예를 들어, 본 개시에 설명된 기법들은 장치 또는 디바이스의 관점으로부터 설명될 수도 있다. 일 예로서, 장치 또는 디바이스는 디코더 프로세서 (30)(예를 들어, 무선 통신 디바이스의 부분으로서의 목적지 디바이스 (14)) 를 포함할 수도 있고, 디코더 프로세서 (30) 는 본 개시에 설명된 기법들을 구현하도록 구성되는 하나 이상의 프로세서들을 포함할 수도 있다. 다른 예로서, 장치 또는 디바이스는 디코더 프로세서 (30) 를 포함하는 집적 회로 (IC) 또는 마이크로프로세서를 포함할 수도 있고, 마이크로-프로세서 또는 IC 는 목적지 디바이스 (14) 또는 다른 유형의 디바이스의 부분일 수도 있다.

[0115] 디코더 프로세서 (30) 의 컴포넌트들은 목적지 디바이스 (14) 의 기존의 예들의 부분일 수도 있다. 예를 들어, 표준 하드웨어의 부분으로서, 목적지 디바이스 (14) 의 대부분의 예들은 GPU, 이를 테면 GPU (70), 디코더, 이를 테면, 디코더 (80) 및 비디오 메모리, 이를 테면, 비디오 메모리 (82) 를 포함한다. GPU (70) 는 본 개시에 설명된 예시적인 기법을 구현하도록 구성되는 일 예시적인 컴포넌트로서 예시된다. GPU (70) 의 고속 병렬 프로세싱 능력들은 GPU (70) 가 본 개시에 설명된 기법들을 구현하기 위한 적절한 옵션을 행하게 한다. 그러나, 본 기법들은 이것으로 제한되지 않는다. 목적지 디바이스 (14) 의 CPU 를 포함하여, 예시적인 기법들을 구현하기 위해 GPU (70) 외의 컴포넌트들을 이용하는 것도 가능할 수도 있다. 또한, 일부 GPU들은 범용 GPU들 (GPGPUs) 로서 기능하는 프로세싱 능력을 포함한다. 일부 예들에서, GPU (70) 는 GPGPU 일 수도 있고, 디코더 (80) 의 기능들을 수행하도록 구성될 수도 있다 (즉, 디코더 (80) 및 GPU (70) 는 동일한 GPGPU 의 부분일 수도 있다).

[0116] 비디오 메모리 (82) 는 디코더 프로세서 (30) 를 형성하는 칩의 부분일 수도 있다. 일부 예들에서, 비디오 메모리 (82) 는 디코더 프로세서 (30) 에 대해 외부에 있을 수도 있고 목적지 디바이스 (14) 의 시스템 메모리의 부분일 수도 있다. 일부 예들에서, 비디오 메모리 (82) 는 시스템 메모리와 디코더 프로세서 (30) 의 내에 있는 메모리의 조합일 수도 있다.

[0117] 비디오 메모리 (82) 는 현재 이미지를 디코딩하는데 이용되는 데이터를 저장할 수도 있다. 예를 들어, 예시된 바와 같이, 비디오 메모리 (82) 는, 장면 구조 맵 (72) 이 인코딩된 비디오 데이터 비트스트림의 부분으로서 수신되면, 장면 구조 맵 (72) 을 저장하고, 이전 장면 구조 맵 (74), 하나 이상의 이미지들 (76) 및 GPU (70) 가 생성한 합성 이미지 (78) 를 저장한다. 비디오 메모리 (82) 는 다양한 메모리 디바이스들, 이를 테면 동기식 DRAM (SDRAM) 을 포함하여 동적 랜덤 액세스 메모리 (DRAM), 자기 저항성 RAM (MRAM), 저항성 RAM (RRAM), 또는 다른 유형의 메모리 디바이스들 중 임의의 것에 의해 형성될 수도 있다.

[0118] 디코더 (80) 는 현재 이미지에 대한 잔차 이미지의 정보를 포함하는 비디오 데이터의 비트스트림을 수신한다 (예를 들어, 잔차 이미지는 현재 이미지로부터 합성 이미지의 감산으로부터 생성된다). 디코더 (80) 는 또한 카메라 포즈 정보를 수신한다. 일부 예들에서, 디코더 (80) 는, 디코더 (80) 가 비디오 메모리 (82) 에 장면 구조 맵 (72) 으로서 저장한, 현재 이미지의 장면 구조 맵의 정보를 수신한다. 일부 예들에서, 디코더 (80) 는 현재 이미지에 대한 장면 구조 맵과 이전 장면 구조 맵 (예를 들어, 이전 장면 구조 맵 (74)) 사이의 차이를 표시하는 정보를 수신한다. 디코더 (80) 는 이전 장면 구조 맵 (74) 의 값들과 차이 값들을 합산하고 장면 구조 맵 (72) 으로서 결과적인 값들을 저장한다.

[0119] GPU (70) 는 장면 구조 맵 (72) 또는 이전 장면 구조 맵 (74) 중 하나를 수신하고 예를 들어, 버텍스 쉐이더 또

는 입력 어셈블러를 이용하여 프록시 지오메트리의 형태로 포인트들을 상호접속한다. 추가로, GPU (70) 는 이미지들 (76) 중 둘 이상의 이미지를 블렌딩하여 합성 이미지를 형성하거나, 또는 이미지들 (76) 중 단일의 하나를 합성 이미지로서 이용한다.

[0120] 디코더 프로세서 (30) 의 GPU (70) 와 인코더 프로세서 (20) 의 GPU (36) 는 이미지들의 동일한 세트 (예를 들어, 이전에 디코딩된 픽처들) 를 이용하도록 구성될 수도 있다. 예를 들어, GPU (36) 와 마찬가지로, GPU (70) 는 하나 이상의 키프레임들 (예를 들어, I-프레임들) 을 이용할 수도 있고 하나 이상의 이미지들 (76) 은 모두 키프레임들일 수도 있다. 이러한 식으로, GPU (70) 가 형성한 복합 이미지는 실질적으로 GPU (36) 가 형성한 복합 이미지와 실질적으로 동일하다.

[0121] GPU (70) 는 텍스처 맵으로서 복합 이미지를 그리고 텍스처 맵핑이 맵핑된 오브젝트로서 프록시 지오메트리를 이용하여 텍스처 맵핑을 수행할 수도 있다. GPU (70) 는 그 후, 합성 이미지 (78) 를 생성하기 위해 결과를 렌더링할 수도 있다. GPU (70) 는 GPU (70) 가 합성 이미지 (46) 를 생성하도록 구현하였던 합성 이미지 (78) 를 생성하기 위해 실질적으로 동일한 프로세스를 구현할 수도 있다. 예를 들어, GPU (70) 가 GPU (36) 과 동일한 장면 구조 맵을 이용하여 프록시 지오메트리를 형성하고 GPU (36) 와 동일한 이미지들을 이용하여 복합 이미지들을 형성하기 때문에, 합성 이미지 (78) 및 합성 이미지 (46) 는 실질적으로 유사할 수도 있다. 추가로, GPU (70) 는 IBM 을 생성하도록 텍스처 맵핑을 위하여, 복합 이미지를 구성하는데 이용된 하나 이상의 이미지들 (76) 의 카메라 포즈 정보를 유사하게 이용할 수도 있고 합성 이미지를 렌더링하기 위하여 현재 이미지의 카메라 포즈 정보를 이용할 수도 있다.

[0122] 디코더 (80) 는 합성 이미지 (78) 에 잔차 이미지를 가산할 수도 있다. 가산의 결과는 재구성된 현재 이미지이다.

[0123] 또한, 인코더 (48) 와 유사하게, 디코더 (80) 는 본 개시에 설명된 기법들을 구현하도록 구성될 수도 있고 통상의 비디오 디코딩 프로세스를 또한 구현하도록 구성될 수도 있다. 예를 들어, 디코더 (80) 는 이미지 내에서 샘플들만을 기초하여 이미지를 디코딩 (예를 들어, I-프레임이도록 이미지를 디코딩) 하도록 구성될 수도 있다. 디코더 (80) 는 통상의 비디오 디코딩을 이용하여 이러한 비디오 디코딩 기법들을 수행할 수도 있다.

[0124] 다른 예로서, 일부 예들에서, 현재 이미지에서의 비정적 전경 오브젝트들이 다른 디코딩 기법들을 이용하여 디코딩되고 정적 배경 오브젝트들이 본 개시에 설명된 기법들을 이용하여 디코딩되면 디코딩 이득들이 실현될 수도 있다. 예를 들어, 디코더 (80) 는 본 개시에 설명된 기법들을 이용하여 이미지의 부분들을 디코딩하는 것과, 다른 기법들을 이용하여 이미지의 다른 부분들을 디코딩하는 것 사이를 선택하는 모드 선택 유닛을 포함할 수도 있다. 예를 들어, 디코더 (80) 는 본 개시에 설명된 기법들을 이용하여 인코딩된 현재 이미지의 부분들을 갖는 한 계층, 및 다른 기법들을 이용하여 인코딩된 현재 이미지의 부분들을 갖는 다른 계층을 수신할 수도 있다.

[0125] 전경의 비정적 부분이 하나의 계층을 형성하고 배경의 정적 부분이 다른 계층을 형성하는 예들에서, 디코더 (80) 는 전경 계층과 배경 계층을 개별적으로 디코딩할 수도 있다. 디코더 프로세서 (30) 는 전경의 비정적 계층과 배경의 정적 계층을 개별적으로 플레이백할 수도 있다.

[0126] 디코더 (80) 는 또한 디코딩 정보를 제공하는 신택스 엘리먼트들을 디코딩하도록 구성될 수도 있다. 예를 들어, GPU (70) 가 이전 장면 구조 맵 (74) 을 이용하여 프록시 지오메트리를 생성하기 위한 경우의 예들에서, 디코더 (80) 는 GPU (70) 가 이전 장면 구조 맵 (74) 을 이용해야 함을 표시하는 신택스 엘리먼트들을 디코딩할 수도 있다.

[0127] 도 8 은 본 개시에 따라 하나 이상의 예시적인 비디오 디코딩 기법들을 예시하는 데이터 흐름도이다. 예를 들어, 디코더 (80) 는 디코딩 (84) 으로서 예시되는 비트스트림 (예를 들어, 비트스트림 (62)) 의 디코딩을 수행한다. 디코더 (80) 는 인코더 프로세서 (20) 가 출력하는 잔차 이미지를 표시하는 정보에 기초하여 잔차 이미지 (86) 를 결정한다. 일부 예들에서, 디코더 (80) 는 장면 구조 맵 (예를 들어, 장면 구조 맵과 이전 장면 구조 맵 사이의 차이들을 표시하는 장면을 이전 장면 구조 맵의 값들에 합산하는 것에 의해 또는 직접) 을 출력한다.

[0128] GPU (70) 는 수신된 장면 구조 맵으로부터 프록시 지오메트리를 형성한다. 일부 예들에서, GPU (70) 는 이전 장면 구조 맵을 이용할 수도 있다. 예를 들어, 디코딩 (84) 은 장면 구조 맵, 또는 장면 구조 맵과 이전 장면 구조 맵 사이의 차이를 디코딩하는 것을 수반하지 않을 수도 있다. 이러한 예들에서, GPU (70) 는 이전 장면 구조 맵을 재사용할 수도 있다.

- [0129] 본 개시에 설명된 기법들에서, GPU (70)는 장면 구조 맵과 복합 이미지에 기초하여 프록시 지오메트리를 생성하고, 텍스처 맵핑을 수행하며, 여기에서 복합 이미지는 텍스처 맵이고 프록시 지오메트리는 텍스처 맵이 맵핑되는 오브젝트이다. 텍스처 맵핑의 결과는 이미지-기반 모델 (88)이다. GPU (70)는 예측 이미지 (90)에 의해 표현되는 합성 이미지를 생성하도록 이미지-기반 모델 (88)을 렌더링한다.
- [0130] 합산 유닛 (87)은 예측 이미지 (90)에 잔차 이미지 (86)를 가산한다. 가산의 결과는 현재 이미지 (92)이다.
- [0131] 이러한 식으로, 본 개시에 설명된 예시적인 기법들은 비디오 인코딩 및 디코딩을 위한 예시적인 기법들을 제공한다. 일부 예들에서, 통상의 모션 보상은 필요하지 않을 수도 있다. 예를 들어, 통상의 모션 벡터 보상에 이용된 모션 벡터들, 참조 픽처 리스트들 및 다른 이러한 정보는 필요하지 않을 수도 있어, 요구되는 대역폭의 양에서의 감소를 허용한다. 모션 보상 정보가 필요하지 않지만, 카메라 포즈 및 장면 구조 맵 (또는 장면 구조 맵에 대한 변화들) 정보는 출력되는 것이 필요할 수도 있다. 그러나, 카메라 메트릭스 및 장면 구조 맵의 3D 포인트들에 대해 몇몇 바이트들만이 이미지당 요구되어, 무시할 수 있는 대역폭 요건들을 가져온다.
- [0132] 또한, 본 개시에 설명된 기법들은 OTF (on-the-fly) 인코딩 및 디코딩에 매우 적합할 수도 있다. 예를 들어, 인코더 프로세서 (20)는 개별적인 GPU들과 OTF (on-the-fly)로 개별적인 예측 이미지들 (예를 들어, 합성 이미지)을 생성 가능하다. 이는 디코딩 동안에 GPU에 텍스처 맵핑 및 심도 버퍼 하드웨어에 의해 행해진 모션 보상과 유사한 것으로 고려될 수 있다.
- [0133] 또한, 기법들은 스테레오스코픽 디스플레이에 대해서도 또한 이용될 수도 있다. 예를 들어, 합성 이미지는 스테레오스코픽 뷰에 대한 이미지들을 생성하는데 이용될 수 있는 뷰포인트와 같은, 현재 이미지의 카메라 포즈 대신에 임의적인 뷰포인트로부터 이론적으로 렌더링될 수도 있다. 이 경우에, 결과적인 합성 이미지는 비-스테레오스코픽 뷰에서의 예측 목적들에 대해서는 이상적이지 않을 수도 있지만, 합성 이미지가 충분할 수도 있는 작은 변화에 대해 이상적일 수도 있다. 이러한 기법들은 제한된 프리-뷰포인트로서 지칭될 수도 있다.
- [0134] 도 9는 비디오 데이터를 인코딩하는 예시적인 방법을 도시하는 플로차트이다. 도 9의 예에 예시된 바와 같이, 인코더 프로세서 (20)의 GPU (36)는 복합 이미지와, 현재 이미지의 장면 구조 맵 또는 이전에 인코딩된 이미지의 장면 구조 맵에 기초하여 합성 이미지를 생성할 수도 있다 (100). 복합 이미지는 이전에 인코딩되었던 하나 이상의 이미지들로부터 구성된다. 예를 들어, 복합 이미지는 하나 이상의 이전에 인코딩된 키프레임들의 블렌드 또는 단일의 이전에 인코딩된 키프레임일 수도 있다. 장면 구조 맵은 비디오 데이터의 현재 이미지 또는 이전에 인코딩되었던 비디오 데이터의 이미지 내에서 3차원 포인트들에 대한 좌표값들을 포함한다. 예를 들어, 인코더 프로세서 (20)의 SLAM 프로세서 (34)는 동시 로컬리제이션 및 맵핑 (SLAM) 기법들을 이용하여 장면 구조 맵을 생성할 수도 있다.
- [0135] 일부 예들에서, 합성 이미지를 생성하기 위해, GPU (36)는 프록시 지오메트리를 형성하기 위해 장면 구조 맵의 포인트들을 상호접속할 수도 있다. GPU (36)는 이미지-기반 모델을 형성하기 위해 프록시 지오메트리에 복합 이미지를 텍스처 맵핑한다. 일부 예들에서, GPU (36)는 이미지-기반 모델을 형성하기 위해 복합 이미지를 구성하는데 이용되는 하나 이상의 이미지들의 카메라 포즈 정보에 기초하여 프록시 지오메트리에 복합 이미지를 텍스처 맵핑할 수도 있다. GPU (36)는 합성 이미지를 생성하기 위해 이미지-기반 모델을 렌더링할 수도 있다. 일부 예들에서, GPU (36)는 합성 이미지를 생성하기 위해 현재 이미지의 카메라 포즈 정보에 기초하여 이미지-기반 모델을 렌더링할 수도 있다.
- [0136] 인코더 프로세서 (20)의 인코더 (48)는 합성 이미지와 현재 이미지에 기초하여 잔차 이미지를 결정할 수도 있다 (102). 잔차 이미지는 현재 이미지와 합성 이미지 사이의 차이를 나타낸다. 인코더 (48)는 비디오 데이터의 현재 이미지를 인코딩하기 위해 잔차 이미지를 출력 (예를 들어, 잔차 이미지를 표시하는 정보를 출력) 할 수도 있다 (104). 일부 예들에서, 인코더 (48)는 현재 이미지의 장면 구조 맵의 정보, 또는 현재 이미지의 장면 구조 맵과 이전에 인코딩된 이미지의 장면 구조 맵 사이의 차이를 표시하는 정보 중 하나를 또한 출력할 수도 있고, 그리고 카메라 포지션 및 카메라 배향의 일방 또는 양방을 출력할 수도 있다.
- [0137] 일부 예들에서, 인코더 프로세서 (20) 또는 소스 디바이스 (12)의 일부 다른 유닛은 현재 이미지의 전경의 비정적 부분 및 현재 이미지의 배경의 정적 부분을 결정할 수도 있다. 이들 예들에서, 잔차 이미지를 결정하기 위해, 인코더 (48)는 합성 이미지 및 현재 이미지의 배경의 정적 부분에 기초하여 잔차 이미지를 결정하도록 구성될 수도 있다. 또한, 이들 예들에서, 잔차 데이터를 출력하기 위해 인코더 (48)는 현재 이미지의

전경의 비정적 부분에 대한 잔차 데이터를 포함하는 제 2 계층과는 상이한 제 1 계층에서 잔차 데이터를 출력하도록 구성될 수도 있다.

[0138] 도 10 은 비디오 데이터를 디코딩하는 예시적인 방법을 도시하는 플로차트이다. 디코더 프로세서 (30) 의 디코더 (80) 는 비디오 데이터의 현재 이미지의 잔차 이미지를 결정할 수도 있다 (200). 잔차 이미지는 현재 이미지와 합성 이미지 사이의 차이를 나타낸다. 예를 들어, 디코더 (80) 는 인코더 프로세서 (20) 가 출력하였던 잔차 이미지를 표시하는 정보를 수신할 수도 있다. 잔차 이미지를 표시하는 이 정보는 인코딩된 정보 또는 인코딩되지 않은 정보일 수도 있다. 어느 예에서도, 디코더 (80) 는 인코더 프로세서 (20) 가 출력하였던 잔차 이미지를 표시하는 정보에 기초하여 잔차 이미지를 결정 (예를 들어, 잔차 이미지를 재구성) 할 수도 있다.

[0139] 디코더 프로세서 (30) 의 GPU (70) 는 복합 이미지와, 현재 이미지의 장면 구조 맵 또는 이전에 디코딩된 이미지의 장면 구조 맵에 기초하여 합성 이미지를 생성할 수도 있다 (202). 복합 이미지는 이전에 인코딩되었던 하나 이상의 이미지들로부터 구성된다. 장면 구조 맵은 비디오 데이터의 현재 이미지의 장면 구조 맵, 또는 이전에 디코딩되었던 비디오 데이터의 이미지의 장면 구조 맵을 포함하며, 장면 구조 맵은 현재 이미지 또는 이전에 디코딩되었던 이미지 내에서 3 차원 포인트들에 대한 좌표 값들을 포함한다. 디코더 (80) 는 합성 이미지와 잔차 이미지에 기초하여 현재 이미지를 재구성할 수도 있다 (204).

[0140] 예를 들어, GPU (36) 와 마찬가지로, GPU (70) 는 프록시 지오메트리를 형성하기 위해 장면 구조 맵의 포인트들을 상호접속할 수도 있다. GPU (70) 는 그리고 이미지-기반 모델을 형성하기 위해 그리고 이전에 디코딩되었던 하나 이상의 이미지들의 카메라 포즈 정보에 기초하여 가능성있게 프록시 지오메트리에 복합 이미지를 텍스처 맵핑할 수도 있다. GPU (70) 는 합성 이미지를 생성하기 위해, 그리고 현재 이미지의 카메라 포즈 정보에 기초하여 가능성있게 이미지-기반 모델을 렌더링할 수도 있다.

[0141] 일부 예들에서, 복합 이미지는 둘 이상의 이전에 디코딩된 키프레임들의 블렌드 또는 단일의 이전에 디코딩된 키프레임일 수도 있다. 장면 구조 맵은 현재 이미지 또는 이전에 디코딩된 이미지 내에서 3 차원 포인트들에 대한 좌표값들을 포함한다.

[0142] 디코더 (80) 는 현재 이미지의 장면 구조 맵의 정보, 또는 현재 이미지의 장면 구조 맵과 이전에 디코딩된 이미지의 장면 구조 맵 사이의 차이를 표시하는 정보 중 하나를 수신할 수도 있다. 일부 예들에서, 디코더 (80) 는 카메라 포지션 및 카메라 배향 중 일방 또는 양방을 또한 수신할 수도 있다.

[0143] 일부 예들에서, 디코더 (80) 는 현재 이미지의 배경의 정적 부분에 대한 잔차 이미지를 수신할 수도 있다. 이들 예들에서, 디코더 (80) 는 합성 이미지와, 현재 이미지의 배경의 정적 부분에 대한 잔차 데이터에 기초하여 현재 이미지를 재구성할 수도 있다.

[0144] 예에 의존하여, 본원에 설명된 기법들의 소정의 작용들 또는 이벤트들은 상이한 시퀀스로 수행될 수 있으며, 부가, 병합, 또는 모두 배제될 수도 있음 (예를 들어, 반드시 모든 설명된 작용들 또는 이벤트들이 기법들의 실시를 위해 필요한 것은 아니다) 을 알아야 한다. 또한, 소정의 예들에서, 작용들 및 이벤트들은, 순차적으로 수행되는 대신에, 예를 들어, 멀티 스레드 프로세싱, 인터럽트 프로세싱, 또는 멀티 프로세서들을 통해 동시에 수행될 수도 있다.

[0145] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되는 경우, 기능들은 하드웨어 기반 프로세싱 유닛에 의해 실행되는 하나 이상의 명령들 또는 코드로서 컴퓨터 판독 가능 매체 상에 저장되거나 송신될 수도 있다. 컴퓨터 판독 가능 매체들은 데이터 저장 매체들과 같은 유형의 매체에 대응하는 컴퓨터 판독 가능 저장 매체들, 또는 예를 들어, 통신 프로토콜에 따라, 한 곳에서 다른 곳으로 컴퓨터 프로그램의 전송을 가능하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다. 이러한 방식으로, 컴퓨터 판독 가능한 매체들은 일반적으로 (1) 비일시적 인 유형의 컴퓨터 판독 가능 저장 매체들 또는 (2) 신호 또는 반송파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체는 이 개시물에 설명된 기법들의 구현을 위한 명령들, 코드, 및/또는 데이터 구조들을 검색하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용 가능한 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독 가능 매체를 포함할 수도 있다.

[0146] 비제한적인 예로서, 이러한 컴퓨터 판독 가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 스토리지, 자기 디스크 스토리지 또는 다른 자기 스토리지 디바이스들, 플래시 메모리, 또는 소망의 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 저장하기 위해 사용될 수 있으며 컴퓨터에 의해 액세스될 수 있는 임

의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터 판독 가능한 매체를 적절히 지칭한다. 예를 들면, 소프트웨어가 동축 케이블, 광섬유 케이블, 연선, 디지털 가입자 회선 (DSL), 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들을 이용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 명령들이 송신되면, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들은 매체의 정의 내에 포함된다. 그러나, 컴퓨터 판독 가능한 저장 매체들 및 데이터 저장 매체들은 접속부들, 반송파들, 신호들, 또는 다른 일시적 매체들을 포함하지 않고, 대신에 비일시적, 유형의 저장 매체들에 대한 것임이 이해되어야 한다. 본원에서 이용된 디스크 (disk) 와 디스크 (disc) 는, 컴팩트 디스크(CD), 레이저 디스크, 광학 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크, 및 블루레이 디스크를 포함하며, 여기서 디스크 (disk) 들은 통상 자기적으로 데이터를 재생하는 반면, 디스크(disc) 들은 레이저들을 이용하여 광학적으로 데이터를 재생한다. 상기의 조합들도 컴퓨터 판독 가능한 매체의 범위 내에 또한 포함되어야만 한다.

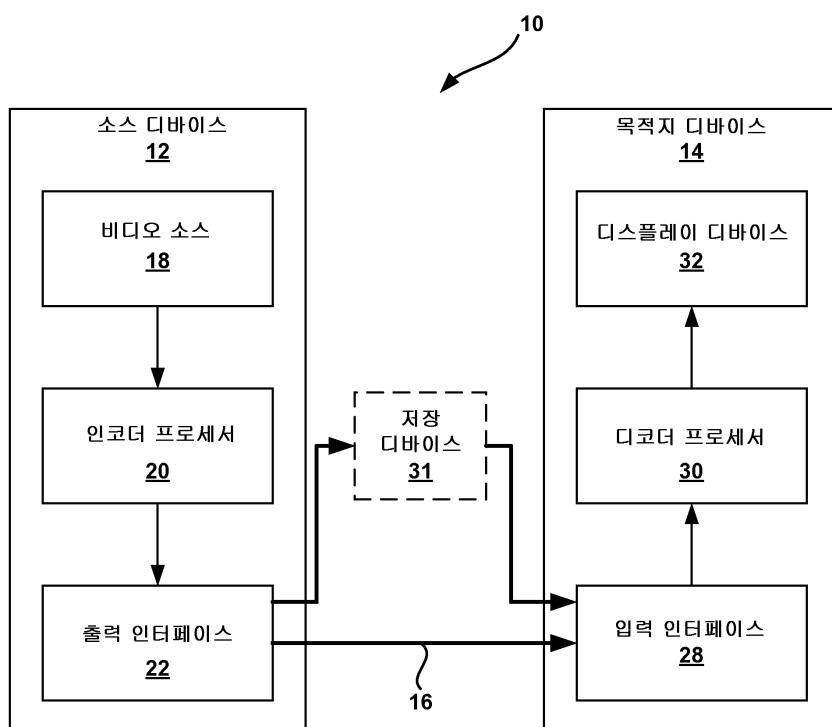
[0147] 명령들은, 하나 이상의 디지털 신호 프로세서들(DSPs), 범용 마이크로프로세서들, 주문형 집적 회로들(ASICs), 필드 프로그래머블 로직 어레이(FPGAs), 또는 다른 등가의 집적 또는 이산 로직 회로와 같은, 하나 이상의 프로세서들에 의해 실행될 수도 있다. 따라서, 본원에서 사용된 용어 "프로세서"는 임의의 앞서 설명된 구조 또는 본원에서 설명된 기술들의 구현에 적합한 임의의 다른 구조를 지칭할 수도 있다. 또한, 몇몇 양태들에서, 본원에서 설명된 기능성은 인코딩 및 디코딩을 위해 구성된 전용 하드웨어 및/또는 소프트웨어 모듈 내에 제공되거나, 또는 통합 코덱에 통합될 수도 있다. 또한, 상기 기술들은 하나 이상의 회로들 또는 로직 소자들로 완전히 구현될 수 있다.

[0148] 본 개시물의 기술들은, 무선 헤드셋, 집적 회로 (IC) 또는 IC들의 세트 (예를 들어, 칩 세트)를 포함하는 다양한 디바이스들 또는 장치들에서 구현될 수도 있다. 개시된 기술들을 수행하도록 구성된 디바이스들의 기능적 양태를 강조하기 위해 다양한 소자들, 모듈들, 또는 유닛들이 본 개시에서 설명되었지만, 반드시 상이한 하드웨어 유닛들에 의해 실현될 필요는 없다. 오히려, 상술된 바와 같이, 적합한 소프트웨어 및/또는 펌웨어와 결합하여, 상술된 하나 이상의 프로세서들을 포함해, 다양한 유닛들이 코덱 하드웨어 유닛으로 결합되거나 상호동작가능한 하드웨어 유닛들의 집합으로 제공될 수도 있다.

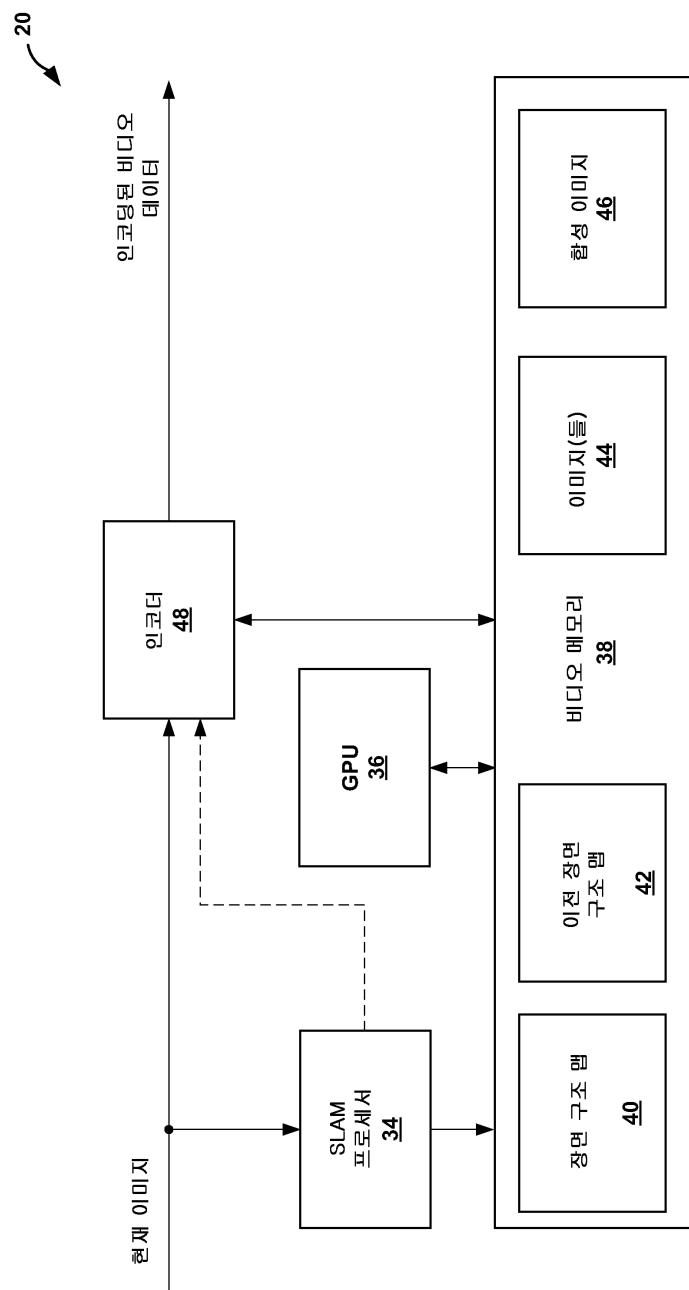
[0149] 다양한 예들이 설명되어 있다. 이러한 예들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

도면1



도면2



도면3a



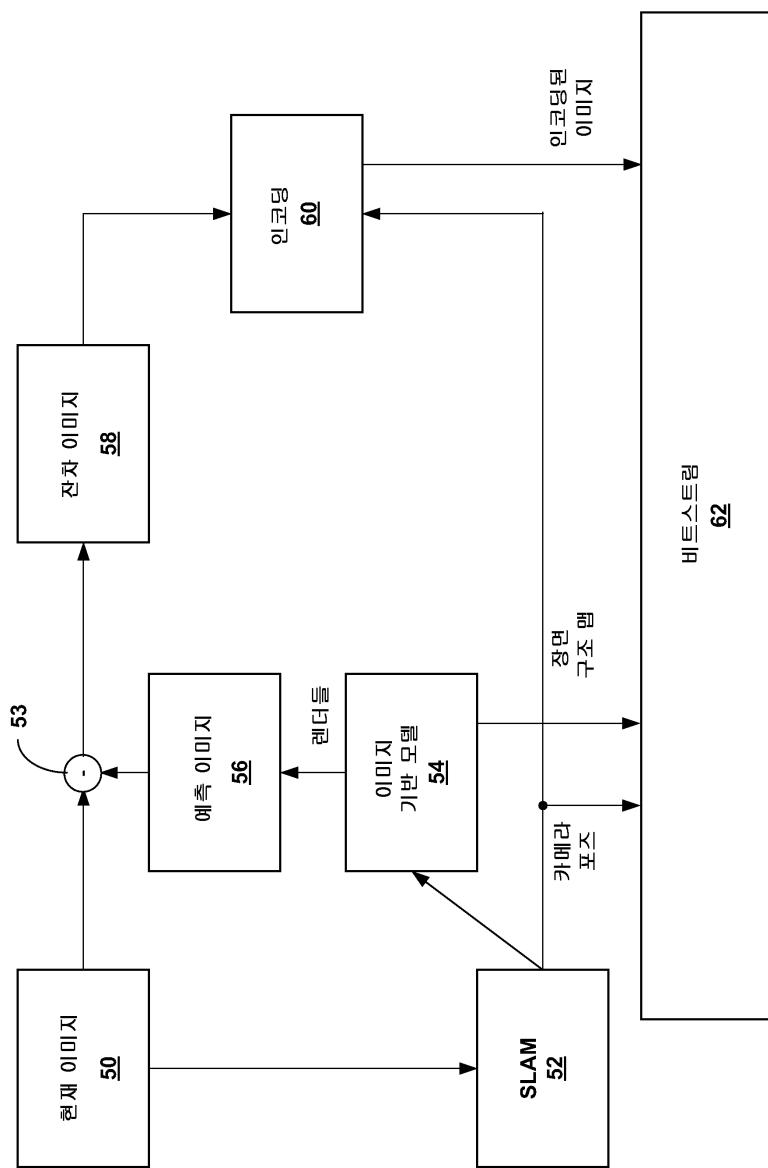
도면3b

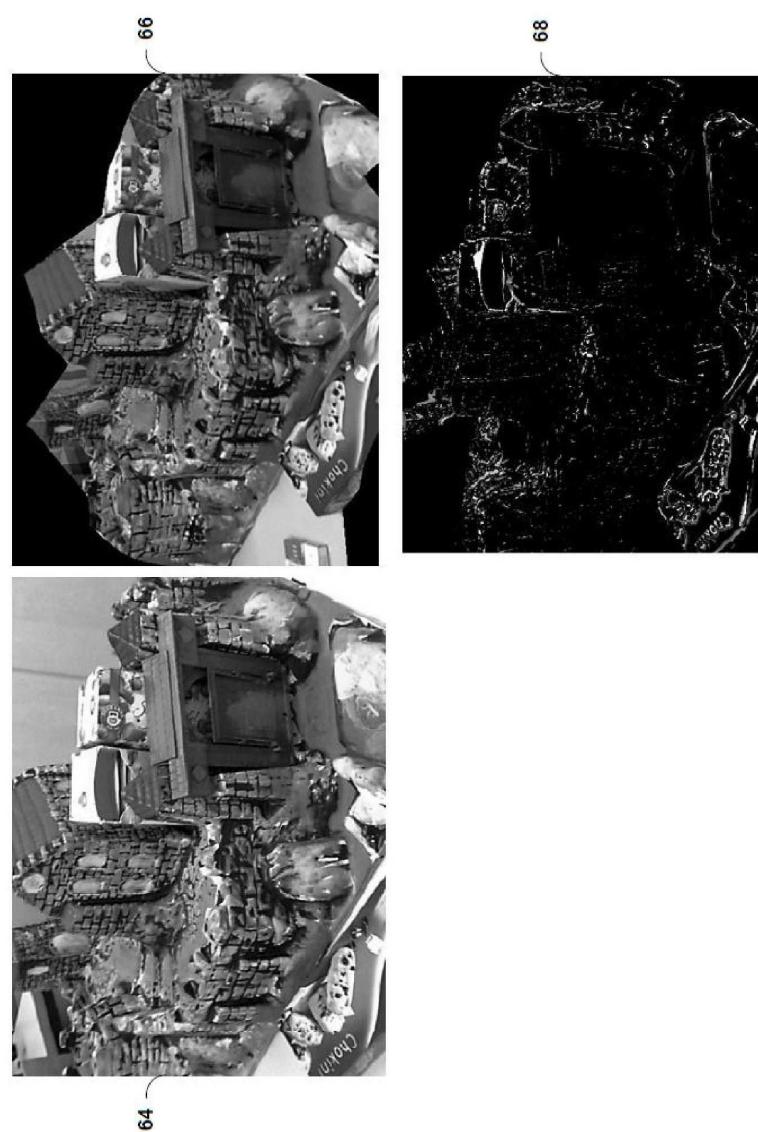


도면4

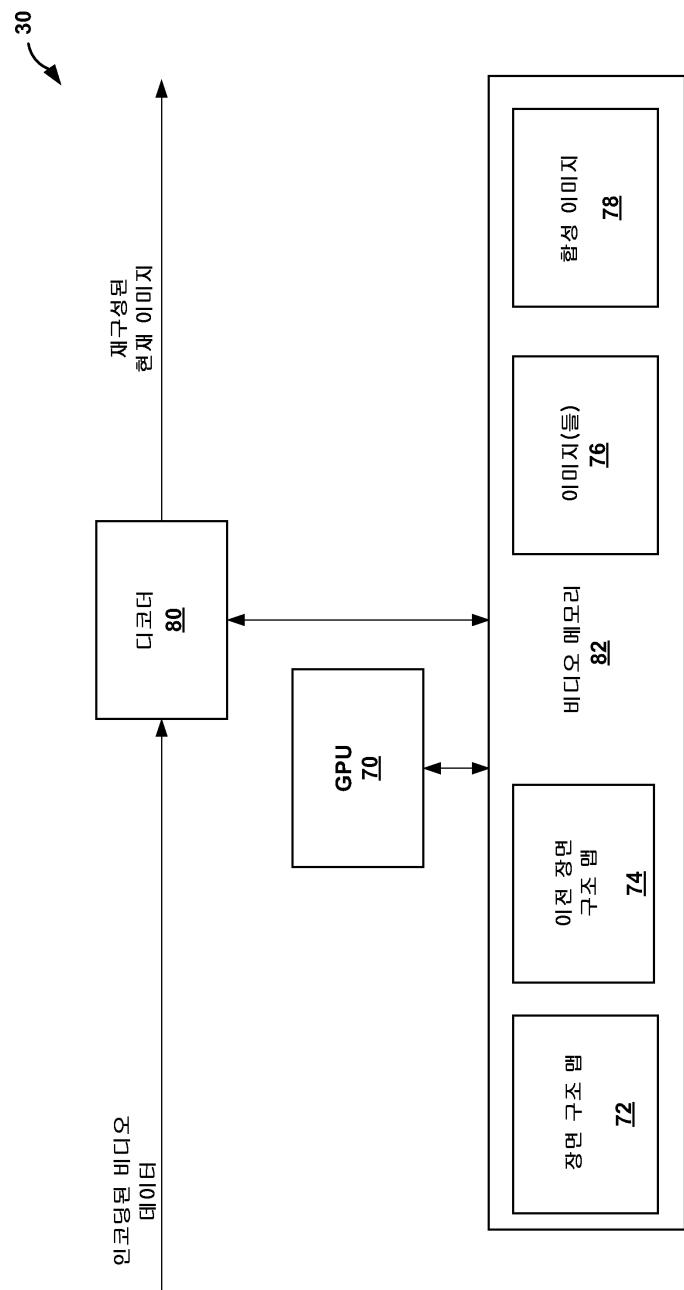


도면5

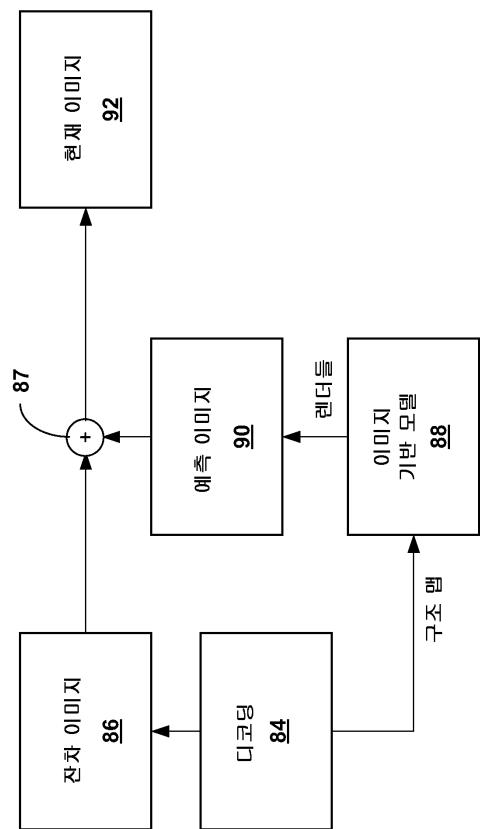




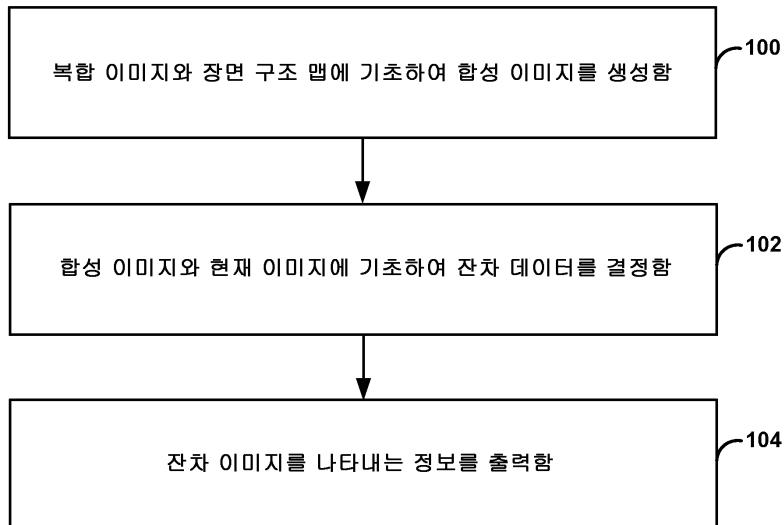
도면7



도면8



도면9



도면10

