



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2012-0071407
(43) 공개일자 2012년07월02일

- | | |
|--|---|
| <p>(51) 국제특허분류(Int. Cl.)
H04N 7/26 (2006.01) H04N 13/00 (2006.01)</p> <p>(21) 출원번호 10-2012-7013246</p> <p>(22) 출원일자(국제) 2010년10월22일
심사청구일자 2012년05월22일</p> <p>(85) 번역문제출일자 2012년05월22일</p> <p>(86) 국제출원번호 PCT/US2010/053823</p> <p>(87) 국제공개번호 WO 2011/050304
국제공개일자 2011년04월28일</p> <p>(30) 우선권주장
12/696,499 2010년01월29일 미국(US)
61/254,558 2009년10월23일 미국(US)</p> | <p>(71) 출원인
퀄컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775</p> <p>(72) 발명자
장 룡
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775</p> <p>천 잉
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775</p> <p>카르체비츠 마르타
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775</p> <p>(74) 대리인
특허법인코리아나</p> |
|--|---|

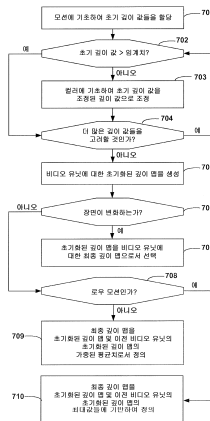
전체 청구항 수 : 총 33 항

(54) 발명의 명칭 **2D 비디오 데이터의 3D 비디오 데이터로의 컨버전을 위한 깊이 맵 생성 기술들**

(57) 요약

본 개시물은 비디오 프레임들 또는 비디오 프레임들의 슬라이스들과 같은 비디오 유닛들에 대한 깊이 맵들을 생성하기 위한 기술들을 설명한다. 이 기술들은 2 차원 (2D) 비디오를 3 차원 (3D) 비디오로 컨버팅하기 위하여 비디오 인코더에 의해 수행될 수도 있다. 이 기술들은 대안적으로, 수신된 2D 비디오를 3D 비디오로 컨버팅하기 위해 비디오 디코더에 의해 수행될 수도 있다. 이 기술들은 깊이 맵 생성 프로세스에서 모션과 컬러 고려 사항들의 조합을 이용할 수도 있다.

대표도 - 도7



특허청구의 범위

청구항 1

비디오 시퀀스의 이전 비디오 유닛의 대응하는 픽셀들에 대한 비디오 유닛의 픽셀들과 연관된 모션에 기초하여 초기 깊이 값들을 상기 비디오 유닛의 픽셀들에 할당하는 단계;

상기 초기 깊이 값들이 임계치를 만족하는지 여부를 확인하는 단계;

상기 초기 깊이 값들이 상기 임계치를 만족하는 상기 비디오 유닛의 하나 이상의 픽셀들에 조정된 깊이 값들을 할당하는 단계로서, 상기 조정된 깊이 값들은 상기 픽셀들과 연관된 컬러에 기초하는, 상기 조정된 깊이 값들을 할당하는 단계; 및

상기 비디오 유닛에 대한 깊이 맵을 생성하는 단계를 포함하고,

상기 깊이 맵은 상기 픽셀들의 제 1 서브세트에 대한 상기 초기 깊이 값들 및 상기 픽셀들의 제 2 서브세트에 대한 상기 조정된 깊이 값들을 포함하는, 방법.

청구항 2

제 1 항에 있어서,

상기 모션에 기초하여 초기 깊이 값들을 할당하는 단계는, 상기 이전 비디오 유닛의 공동-위치된 픽셀들의 루마 값들에 대한 상기 비디오 유닛의 픽셀들의 루마 값들에 기초하여 상기 초기 깊이 값들을 할당하는 단계를 포함하는, 방법.

청구항 3

제 1 항에 있어서,

상기 컬러에 기초하는 상기 조정된 깊이 값들을 할당하는 단계는, 상기 비디오 유닛의 픽셀들의 크로마 값들에 적어도 부분적으로 기초하여 상기 조정된 깊이 값들을 할당하는 단계를 포함하는, 방법.

청구항 4

제 1 항에 있어서,

상기 모션에 기초하여 초기 깊이 값들을 할당하는 단계는, 상기 이전 비디오 유닛의 공동-위치된 픽셀들의 루마 값들에 대한 상기 비디오 유닛의 픽셀들의 루마 값들에 기초하여 상기 초기 깊이 값들을 할당하는 단계를 포함하고,

상기 컬러에 기초하는 상기 조정된 깊이 값들을 할당하는 단계는, 상기 비디오 유닛의 픽셀들의 크로마 값들에 적어도 부분적으로 기초하여 상기 조정된 깊이 값들을 할당하는 단계를 포함하며, 그리고

상기 조정된 깊이 값들을 할당하는 단계는, 허용된 구간의 크로마 값들 내의 크로마 값들을 제로 (0) 와 상기 임계치 사이의 깊이 값들로 맵핑하는 단계를 포함하는, 방법.

청구항 5

제 1 항에 있어서,

상기 깊이 맵은 초기화된 깊이 맵을 포함하고,

상기 방법은,

상기 비디오 유닛이 상기 이전 비디오 유닛에 대한 장면 변화에 대응하는지 여부를 결정하는 단계;

상기 비디오 유닛이 상기 이전 비디오 유닛에 대한 장면 변화에 대응한다면, 상기 초기화된 깊이 맵을 상기 비디오 유닛에 대한 최종 깊이 맵으로서 선택하는 단계;

상기 비디오 유닛이 상기 이전 비디오 유닛에 대한 장면 변화에 대응하지 않는다면, 상기 비디오 유닛이 상기 이전 비디오 유닛에 대하여 낮은 레벨의 모션을 나타내는지 여부를 결정하는 단계;

상기 비디오 유닛이 상기 낮은 레벨의 모션을 나타내지 않는다면, 상기 비디오 유닛에 대한 상기 초기화된 깊이 맵의 깊이 값들과 상기 이전 비디오 유닛에 대한 이전 깊이 맵의 깊이 값들의 가중된 평균들에 기초하여 상기 최종 깊이 맵을 정의하는 단계; 및

상기 비디오 유닛이 상기 낮은 레벨의 모션을 나타낸다면, 상기 비디오 유닛에 대한 상기 초기화된 깊이 맵의 깊이 값들과 상기 이전 비디오 유닛에 대한 상기 이전 깊이 맵의 깊이 값들의 최대치들에 기초하여 상기 최종 깊이 맵을 정의하는 단계를 더 포함하는, 방법.

청구항 6

제 5 항에 있어서,

상기 방법은 비디오 디코더에 의해 수행되고,

상기 방법은, 상기 비디오 유닛에 대한 2차 뷰를 생성하기 위해 상기 최종 깊이 맵에 기초하여 뷰 합성을 수행하는 단계를 더 포함하는, 방법.

청구항 7

제 5 항에 있어서,

상기 방법은 비디오 인코더에 의해 수행되고,

상기 방법은, 상기 최종 깊이 맵을 상기 비디오 유닛과 통신하는 단계를 더 포함하는, 방법.

청구항 8

제 5 항에 있어서,

상기 방법은 비디오 인코더에 의해 수행되고,

상기 방법은,

상기 비디오 유닛에 대한 2차 뷰를 생성하기 위해 상기 최종 깊이 맵에 기초하여 뷰 합성을 수행하는 단계; 및

상기 2차 뷰를 상기 비디오 유닛과 통신하는 단계를 더 포함하는, 방법.

청구항 9

깊이 맵 생성 유닛을 포함하는 비디오 코더 장치로서,

상기 깊이 맵 생성 유닛은,

비디오 시퀀스의 이전 비디오 유닛의 대응하는 픽셀들에 대한 비디오 유닛의 픽셀들과 연관된 모션에 기초하여 초기 깊이 값들을 상기 비디오 유닛의 픽셀들에 할당하고;

상기 초기 깊이 값들이 임계치를 만족하는지 여부를 확인하고;

상기 초기 깊이 값들이 상기 임계치를 만족하는 상기 비디오 유닛의 하나 이상의 픽셀들에 조정된 깊이 값을 할당하는 것으로서, 상기 조정된 깊이 값들은 상기 픽셀들과 연관된 컬러에 기초하는, 상기 조정된 깊이 값들을 할당하며; 그리고

상기 비디오 유닛에 대한 깊이 맵을 생성하고,

상기 깊이 맵은 상기 픽셀들의 제 1 서브세트에 대한 상기 초기 깊이 값들 및 상기 픽셀들의 제 2 서브세트에 대한 상기 조정된 깊이 값들을 포함하는, 비디오 코더 장치.

청구항 10

제 9 항에 있어서,

상기 모션에 기초하여 초기 깊이 값들을 할당할 시, 상기 깊이 맵 생성 유닛은 상기 이전 비디오 유닛의 공동-위치된 픽셀들의 루마 값들에 대한 상기 비디오 유닛의 픽셀들의 루마 값들에 기초하여 상기 초기 깊이 값들을 할당하는, 비디오 코더 장치.

청구항 11

제 9 항에 있어서,

상기 컬러에 기초하는 상기 조정된 깊이 값들을 할당할 시, 상기 깊이 맵 생성 유닛은 상기 비디오 유닛의 픽셀들의 크로마 값들에 적어도 부분적으로 기초하여 상기 조정된 값들을 할당하는, 비디오 코더 장치.

청구항 12

제 9 항에 있어서,

상기 모션에 기초하여 초기 깊이 값들을 할당하는 것은, 상기 이전 비디오 유닛의 공동-위치된 픽셀들의 루마 값들에 대한 상기 비디오 유닛의 픽셀들의 루마 값들에 기초하여 상기 초기 깊이 값들을 할당하는 것을 포함하고,

상기 컬러에 기초하는 상기 조정된 깊이 값들을 할당하는 것은, 상기 비디오 유닛의 픽셀들의 크로마 값들에 적어도 부분적으로 기초하여 상기 조정된 깊이 값들을 할당하는 것을 포함하며, 그리고

상기 조정된 깊이 값들을 할당하는 것은, 허용된 구간의 크로마 값들 내의 크로마 값들을 제로 (0) 와 상기 임계치 사이의 깊이 값들로 맵핑하는 것을 포함하는, 비디오 코더 장치.

청구항 13

제 9 항에 있어서,

상기 깊이 맵은 초기화된 깊이 맵을 포함하고,

상기 깊이 맵 생성 유닛은,

상기 비디오 유닛이 상기 이전 비디오 유닛에 대한 장면 변화에 대응하는지 여부를 결정하고;

상기 비디오 유닛이 상기 이전 비디오 유닛에 대한 장면 변화에 대응한다면, 상기 초기화된 깊이 맵을 상기 비디오 유닛에 대한 최종 깊이 맵으로서 선택하고;

상기 비디오 유닛이 상기 이전 비디오 유닛에 대한 장면 변화에 대응하지 않는다면, 상기 비디오 유닛이 상기 이전 비디오 유닛에 대하여 낮은 레벨의 모션을 나타내는지 여부를 결정하고;

상기 비디오 유닛이 상기 낮은 레벨의 모션을 나타내지 않는다면, 상기 비디오 유닛에 대한 상기 초기화된 깊이 맵의 깊이 값들과 상기 이전 비디오 유닛에 대한 이전 깊이 맵의 깊이 값들의 가중된 평균들에 기초하여 상기 최종 깊이 맵을 정의하며; 그리고

상기 비디오 유닛이 상기 낮은 레벨의 모션을 나타낸다면, 상기 비디오 유닛에 대한 상기 초기화된 깊이 맵의 깊이 값들과 상기 이전 비디오 유닛에 대한 상기 이전 깊이 맵의 깊이 값들의 최대치들에 기초하여 상기 최종 깊이 맵을 정의하는, 비디오 코더 장치.

청구항 14

제 13 항에 있어서,

상기 비디오 코더 장치는 비디오 디코더를 포함하고,

상기 비디오 디코더는,

상기 비디오 유닛에 대한 2차 뷰를 생성하기 위해 상기 최종 깊이 맵에 기초하여 뷰 합성을 수행하는 2 차원 (2D) - 3 차원 컨버전 유닛을 더 포함하는, 비디오 코더 장치.

청구항 15

제 13 항에 있어서,

상기 비디오 코더 장치는 비디오 인코더를 포함하고,

비디오 코딩 장치는, 상기 최종 깊이 맵을 상기 비디오 유닛과 통신하는 송신기를 더 포함하는, 비디오 코더 장치.

청구항 16

제 13 항에 있어서,

상기 비디오 코더 장치는 비디오 인코더를 포함하고,

상기 비디오 인코더는, 상기 비디오 유닛에 대한 2차 뷰를 생성하기 위해 상기 최종 깊이 맵에 기초하여 뷰 합성을 수행하는 2 차원 (2D) - 3 차원 컨버전 유닛을 더 포함하고,

비디오 코딩 장치는, 상기 2차 뷰를 상기 비디오 유닛과 통신하는 송신기를 더 포함하는, 비디오 코더 장치.

청구항 17

제 9 항에 있어서,

상기 비디오 코더 장치는,

집적 회로;

마이크로프로세서;

비디오 인코더를 포함하는 무선 통신 디바이스; 및

비디오 디코더를 포함하는 무선 통신 디바이스

중 적어도 하나를 포함하는, 비디오 코더 장치.

청구항 18

비디오 시퀀스의 이전 비디오 유닛의 대응하는 픽셀들에 대한 비디오 유닛의 픽셀들과 연관된 모션에 기초하여 초기 깊이 값들을 상기 비디오 유닛의 픽셀들에 할당하는 수단;

상기 초기 깊이 값들이 임계치를 만족하는지 여부를 확인하는 수단;

상기 초기 깊이 값들이 상기 임계치를 만족하는 상기 비디오 유닛의 하나 이상의 픽셀들에 조정된 깊이 값들을 할당하는 수단으로서, 상기 조정된 깊이 값들은 상기 픽셀들과 연관된 컬러에 기초하는, 상기 조정된 깊이 값들을 할당하는 수단; 및

상기 비디오 유닛에 대한 깊이 맵을 생성하는 수단을 포함하고,

상기 깊이 맵은 상기 픽셀들의 제 1 서브세트에 대한 상기 초기 깊이 값들 및 상기 픽셀들의 제 2 서브세트에 대한 상기 조정된 깊이 값들을 포함하는, 디바이스.

청구항 19

제 18 항에 있어서,

상기 모션에 기초하여 초기 깊이 값들을 할당하는 수단은, 상기 이전 비디오 유닛의 공동-위치된 픽셀들의 루마 값들에 대한 상기 비디오 유닛의 픽셀들의 루마 값들에 기초하여 상기 초기 깊이 값들을 할당하는 수단을 포함하는, 디바이스.

청구항 20

제 18 항에 있어서,

상기 컬러에 기초하는 상기 조정된 깊이 값들을 할당하는 수단은, 상기 비디오 유닛의 픽셀들의 크로마 값들에 적어도 부분적으로 기초하여 상기 조정된 값들을 할당하는 수단을 포함하는, 디바이스.

청구항 21

제 18 항에 있어서,

상기 모션에 기초하여 초기 깊이 값들을 할당하는 수단은, 상기 이전 비디오 유닛의 공동-위치된 픽셀들의 루마 값들에 대한 상기 비디오 유닛의 픽셀들의 루마 값들에 기초하여 상기 초기 깊이 값들을 할당하는 수단을 포함하고,

상기 컬러에 기초하는 상기 조정된 깊이 값들을 할당하는 수단은, 상기 비디오 유닛의 픽셀들의 크로마 값들에 적어도 부분적으로 기초하여 상기 조정된 깊이 값들을 할당하는 수단을 포함하며, 그리고

상기 조정된 깊이 값들을 할당하는 수단은, 허용된 구간의 크로마 값들 내의 크로마 값들을 제로 (0) 와 상기 임계치 사이의 깊이 값들로 맵핑하는 수단을 포함하는, 디바이스.

청구항 22

제 18 항에 있어서,

상기 깊이 맵은 초기화된 깊이 맵을 포함하고,

상기 디바이스는,

상기 비디오 유닛이 상기 이전 비디오 유닛에 대한 장면 변화에 대응하는지 여부를 결정하는 수단;

상기 비디오 유닛이 상기 이전 비디오 유닛에 대한 장면 변화에 대응한다면, 상기 초기화된 깊이 맵을 상기 비디오 유닛에 대한 최종 깊이 맵으로서 선택하는 수단;

상기 비디오 유닛이 상기 이전 비디오 유닛에 대한 장면 변화에 대응하지 않는다면, 상기 비디오 유닛이 상기 이전 비디오 유닛에 대하여 낮은 레벨의 모션을 나타내는지 여부를 결정하는 수단;

상기 비디오 유닛이 상기 낮은 레벨의 모션을 나타내지 않는다면, 상기 비디오 유닛에 대한 상기 초기화된 깊이 맵의 깊이 값들과 상기 이전 비디오 유닛에 대한 이전 깊이 맵의 깊이 값들의 가중된 평균들에 기초하여 상기 최종 깊이 맵을 정의하는 수단; 및

상기 비디오 유닛이 상기 낮은 레벨의 모션을 나타낸다면, 상기 비디오 유닛에 대한 상기 초기화된 깊이 맵의 깊이 값들과 상기 이전 비디오 유닛에 대한 상기 이전 깊이 맵의 깊이 값들의 최대치들에 기초하여 상기 최종 깊이 맵을 정의하는 수단을 더 포함하는, 디바이스.

청구항 23

제 22 항에 있어서,

상기 디바이스는 비디오 디코더를 포함하고,

상기 디바이스는, 상기 비디오 유닛에 대한 2차 뷰를 생성하기 위해 상기 최종 깊이 맵에 기초하여 뷰 합성을 수행하는 수단을 더 포함하는, 디바이스.

청구항 24

제 22 항에 있어서,

상기 디바이스는 비디오 인코더를 포함하고,

상기 디바이스는, 상기 최종 깊이 맵을 상기 비디오 유닛과 통신하는 수단을 더 포함하는, 디바이스.

청구항 25

제 22 항에 있어서,

상기 디바이스는 비디오 인코더를 포함하고,

상기 디바이스는,

상기 비디오 유닛에 대한 2차 뷰를 생성하기 위해 상기 최종 깊이 맵에 기초하여 뷰 합성을 수행하는 수단; 및

상기 2차 뷰를 상기 비디오 유닛과 통신하는 수단을 더 포함하는, 디바이스.

청구항 26

명령들을 포함하는 컴퓨터 판독가능 저장 매체로서,

상기 명령들은, 프로세서에 의해 실행될 시, 상기 프로세서로 하여금,

비디오 시퀀스의 이전 비디오 유닛의 대응하는 픽셀들에 대한 비디오 유닛의 픽셀들과 연관된 모션에 기초하여 초기 깊이 값들을 상기 비디오 유닛의 픽셀들에 할당하게 하고;

상기 초기 깊이 값들이 임계치를 만족하는지 여부를 확인하게 하고;

상기 초기 깊이 값들이 상기 임계치를 만족하는 상기 비디오 유닛의 하나 이상의 픽셀들에 조정된 깊이 값들을 할당하게 하는 것으로서, 상기 조정된 깊이 값들은 상기 픽셀들과 연관된 컬러에 기초하는, 상기 조정된 깊이 값들을 할당하게 하며; 그리고

상기 비디오 유닛에 대한 깊이 맵을 생성하게 하고,

상기 깊이 맵은 상기 픽셀들의 제 1 서브세트에 대한 상기 초기 깊이 값들 및 상기 픽셀들의 제 2 서브세트에 대한 상기 조정된 깊이 값들을 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 27

제 26 항에 있어서,

상기 모션에 기초하여 초기 깊이 값들을 할당할 시, 상기 명령들은 상기 프로세서로 하여금, 상기 이전 비디오 유닛의 공동-위치된 픽셀들의 루마 값들에 대한 상기 비디오 유닛의 픽셀들의 루마 값들에 기초하여 상기 초기 깊이 값들을 할당하게 하는, 컴퓨터 판독가능 저장 매체.

청구항 28

제 26 항에 있어서,

상기 컬러에 기초하는 상기 조정된 깊이 값들을 할당할 시, 상기 명령들은 상기 프로세서로 하여금, 상기 비디오 유닛의 픽셀들의 크로마 값들에 적어도 부분적으로 기초하여 상기 조정된 값들을 할당하게 하는, 컴퓨터 판독가능 저장 매체.

청구항 29

제 26 항에 있어서,

상기 모션에 기초하여 초기 깊이 값들을 할당하는 것은, 상기 이전 비디오 유닛의 공동-위치된 픽셀들의 루마 값들에 대한 상기 비디오 유닛의 픽셀들의 루마 값들에 기초하여 상기 초기 깊이 값들을 할당하는 것을 포함하고,

상기 컬러에 기초하는 상기 조정된 깊이 값들을 할당하는 것은, 상기 비디오 유닛의 픽셀들의 크로마 값들에 적어도 부분적으로 기초하여 상기 조정된 깊이 값들을 할당하는 것을 포함하며, 그리고

상기 조정된 깊이 값들을 할당하는 것은, 허용된 구간의 크로마 값들 내의 크로마 값들을 제로 (0) 와 상기 임계치 사이의 깊이 값들로 맵핑하는 것을 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 30

제 26 항에 있어서,

상기 깊이 맵은 초기화된 깊이 맵을 포함하고,

상기 명령들은 상기 프로세서로 하여금,

상기 비디오 유닛이 상기 이전 비디오 유닛에 대한 장면 변화에 대응하는지 여부를 결정하게 하고;

상기 비디오 유닛이 상기 이전 비디오 유닛에 대한 장면 변화에 대응한다면, 상기 초기화된 깊이 맵을 상기 비디오 유닛에 대한 최종 깊이 맵으로서 선택하게 하고;

상기 비디오 유닛이 상기 이전 비디오 유닛에 대한 장면 변화에 대응하지 않는다면, 상기 비디오 유닛이 상기 이전 비디오 유닛에 대하여 낮은 레벨의 모션을 나타내는지 여부를 결정하게 하고;

상기 비디오 유닛이 상기 낮은 레벨의 모션을 나타내지 않는다면, 상기 비디오 유닛에 대한 상기 초기화된 깊이 맵의 깊이 값들과 상기 이전 비디오 유닛에 대한 이전 깊이 맵의 깊이 값들의 가중된 평균들에 기초하여 상기 최종 깊이 맵을 정의하게 하며; 그리고

상기 비디오 유닛이 상기 낮은 레벨의 모션을 나타낸다면, 상기 비디오 유닛에 대한 상기 초기화된 깊이 맵의 깊이 값들과 상기 이전 비디오 유닛에 대한 상기 이전 깊이 맵의 깊이 값들의 최대치들에 기초하여 상기 최종 깊이 맵을 정의하게 하는, 컴퓨터 판독가능 저장 매체.

청구항 31

제 30 항에 있어서,

상기 프로세서는 비디오 디코더를 포함하고,

상기 명령들은 상기 프로세서로 하여금,

상기 비디오 유닛에 대한 2차 뷰를 생성하기 위해 상기 최종 깊이 맵에 기초하여 뷰 합성을 수행하게 하는, 컴퓨터 판독가능 저장 매체.

청구항 32

제 30 항에 있어서,

상기 프로세서는 비디오 인코더를 포함하고,

상기 명령들은 상기 프로세서로 하여금, 상기 최종 깊이 맵을 상기 비디오 유닛과 통신하게 하는, 컴퓨터 판독가능 저장 매체.

청구항 33

제 30 항에 있어서,

상기 프로세서는 비디오 인코더를 포함하고,

상기 명령들은 상기 프로세서로 하여금,

상기 비디오 유닛에 대한 2차 뷰를 생성하기 위해 상기 최종 깊이 맵에 기초하여 뷰 합성을 수행하게 하고,

상기 2차 뷰를 상기 비디오 유닛과 통신하게 하는, 컴퓨터 판독가능 저장 매체.

명세서

기술분야

[0001] 본 출원은, 2009년 10월 23일에 출원된 미국 가출원 제61/254,558호의 이익을 주장하며, 그 전체 내용은 본 명세서에서 참조로서 통합된다.

[0002] **기술분야**

[0003] 본 개시물은 비디오 코딩, 및 2 차원 (2D) 비디오 데이터의 3 차원 (3D) 비디오 데이터로의 컨버전에 관한 것이다.

배경기술

[0004] 디지털 멀티미디어 능력들은 디지털 텔레비전들, 디지털 다이렉트 브로드캐스트 시스템들, 무선 통신 디바이스들, 무선 브로드캐스트 시스템들, 개인 휴대 정보 단말기들 (PDAs), 랩톱 또는 데스크톱 컴퓨터들, 디지털 카메라들, 디지털 레코딩 디바이스들, 비디오 게임 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 디지털 미디어 플레이어들, 등등을 포함하는 광범위한 디바이스들 내에 통합될 수 있다. 디지털 멀티미디어 디바이스들은 디지털 비디오 데이터를 더 효율적으로 송신 및 수신하거나 저장 및 검색하기 위해 MPEG-2, ITU-H.263, MPEG-4, 또는 ITU-H.264/MPEG-4 Part 10, AVC (Advanced Video Coding) 와 같은 비디오 코딩 기술들을 구현할 수도 있다. 비디오 인코딩 기술들은 비디오 시퀀스들 내에서 고유한 리던던시를 감소시키거나 제거하기 위해 공간적인 및 시간적인 예측을 통해 비디오 압축을 수행할 수도 있다.

[0005] 대부분의 종래의 비디오 시퀀스들은 2 차원 (2D) 뷰잉 포맷으로 인코딩 및 디코딩된다. 그러나, 3 차원 (3D) 시퀀스들 또한 가능하며, 그 비디오 시퀀스는 각각의 비디오 프레임과 연관된 2 이상의 뷰들을 갖는다.

이 경우에, 2 이상의 뷰들은 3D 디스플레이 상에서 결합되어 3D 비디오를 렌더링할 수도 있다.

[0006] 일부 경우들에서, 2 이상의 뷰들은 서로 다른 카메라들에 의해 캡처되어 복수의 뷰들을 포함하는 3D 시퀀스로 인코딩될 수도 있다. 대안적으로, 비디오 프레임의 하나 이상의 2차 뷰들이 원래의 2D 비디오 프레임에 기초하여 합성될 수도 있다. 2D - 3D 컨버전을 용이하게 하기 위해, 비디오 프레임들의 픽셀들에 깊이 값들을 할당하는데 깊이 맵들이 이용될 수도 있다. 소정의 비디오 프레임에 대한 깊이 맵은 비디오 프레임의 2차 뷰 또는 복수의 추가 뷰들을 생성하기 위해 뷰 합성 프로세스에서 그 비디오 프레임에 적용될 수도 있다.

발명의 내용

과제의 해결 수단

[0007] 본 개시물은 비디오 프레임들, 비디오 프레임들의 슬라이스들, 비디오 프레임들의 다른 부분들과 같은 비디오 유닛들에 대한 깊이 맵들을 생성하기 위한 기술들을 설명한다. 이 기술들은 2 차원 (2D) 비디오를 3 차원 (3D) 비디오로 컨버팅하기 위하여 비디오 인코더에 의해 수행될 수도 있다. 이 기술들은 대안적으로, 수신된 2D 비디오를 3D 비디오로 컨버팅하기 위해 비디오 디코더에 의해 수행될 수도 있다.

[0008] 본 개시물의 기술들은 깊이 맵 초기화 프로세스를 포함할 수도 있다. 깊이 맵 초기화를 위해, 초기 깊이 값들은 비디오 시퀀스의 이전 비디오 유닛의 대응하는 픽셀들과 관련된 픽셀들과 연관된 모션에 기초하여 비디오 유닛의 픽셀들에 할당될 수도 있다. 그 후에, 임계치를 만족하는 초기 깊이 값들은 조정될 수도 있고, 여기서 조정된 깊이 값들은 픽셀들과 연관된 컬러에 기초한다. 그 후에, 비디오 유닛에 대하여 초기화된 깊이 맵이 생성될 수 있고, 여기서 초기화된 깊이 맵은 픽셀들의 제 1 서브세트에 대한 초기 깊이 값들 및 픽셀들의 제 2 서브세트에 대한 조정된 깊이 값들을 포함한다. 일부 경우들에서, 초기화된 깊이 맵은 추가의 프로세싱 없이 최종 깊이 맵으로 이용될 수도 있고, 다른 경우들에서, 최종 깊이 맵을 정의하기 위해 초기 깊이 맵에 대하여 추가의 기술들이 적용될 수도 있다.

[0009] 예를 들면, 그 기술들은 비디오 유닛이 이전의 비디오 유닛에 대한 장면 변화에 대응하는지 여부를 결정할 수도 있다. 만약 비디오 유닛이 장면 변화에 대응한다면, 기술들은 초기화된 깊이 맵을 비디오 유닛에 대한 최종 깊이 맵으로서 선택할 수도 있다. 그러나, 비디오 유닛이 장면 변화에 대응하지 않는다면, 기술들은 비디오 유닛이 이전의 비디오 유닛에 대하여 낮은 레벨의 모션을 나타내는지 여부를 결정할 수도 있다. 만약, 비디오 유닛이 낮은 레벨의 모션을 나타내지 않는다면, 기술들은 비디오 유닛에 대한 초기화된 깊이 맵의 깊이 값들과 이전 비디오 유닛에 대한 이전 깊이 맵의 깊이 값들의 가중된 평균들에 기초하여 최종 깊이 맵을 정의할 수도 있다. 비디오 유닛이 낮은 레벨의 모션을 나타낸다면, 기술들은 비디오 유닛에 대한 초기화된 깊이 맵의 깊이 값들과 이전 비디오 유닛의 이전 깊이 맵의 깊이 값들의 최대치들에 기초하여 최종 깊이 맵을 정의할 수도 있다.

[0010] 일 실시예에서, 본 개시물은, 비디오 시퀀스의 이전 비디오 유닛의 대응하는 픽셀들에 대한 비디오 유닛의 픽셀들과 연관된 모션에 기초하여 초기 깊이 값들을 상기 비디오 유닛의 픽셀들에 할당하는 단계, 상기 초기 깊이 값들이 임계치를 만족하는지 여부를 확인하는 단계, 상기 초기 깊이 값들이 상기 임계치를 만족하는, 상기 비디오 유닛의 하나 이상의 픽셀들에 조정된 깊이 값들을 할당하는 단계로서, 상기 조정된 깊이 값들은 상기 픽셀들과 연관된 컬러에 기초하는, 상기 조정된 깊이 값들을 할당하는 단계, 및 상기 비디오 유닛에 대한 깊이 맵을 생성하는 단계로서, 상기 깊이 맵은 상기 픽셀들의 제 1 서브세트에 대한 상기 초기 깊이 값들 및 상기 픽셀들의 제 2 서브세트에 대한 상기 조정된 깊이 값들을 포함하는, 상기 깊이 맵을 생성하는 단계를 포함하는, 방법을 설명한다.

[0011] 다른 실시예에서, 비디오 코더 장치는, 비디오 시퀀스의 이전 비디오 유닛의 대응하는 픽셀들에 대한 비디오 유닛의 픽셀들과 연관된 모션에 기초하여 초기 깊이 값들을 상기 비디오 유닛의 픽셀들에 할당하고, 상기 초기 깊이 값들이 임계치를 만족하는지 여부를 확인하고, 상기 초기 깊이 값들이 상기 임계치를 만족하는, 상기 비디오 유닛의 하나 이상의 픽셀들에 조정된 깊이 값들을 할당하는 것으로서, 상기 조정된 깊이 값들은 상기 픽셀들과 연관된 컬러에 기초하는, 상기 조정된 깊이 값들을 할당하며, 그리고 상기 비디오 유닛에 대한 깊이 맵을 생성하는 것으로서, 상기 깊이 맵은 상기 픽셀들의 제 1 서브세트에 대한 상기 초기 깊이 값들 및 상기 픽셀들의 제 2 서브세트에 대한 상기 조정된 깊이 값들을 포함하는, 상기 깊이 맵을 생성하는, 깊이 맵 생성 유닛을 포함할 수도 있다.

[0012] 또 다른 실시예에서, 디바이스는, 비디오 시퀀스의 이전 비디오 유닛의 대응하는 픽셀들에 대한 비디오 유닛

의 픽셀들과 연관된 모션에 기초하여 초기 깊이 값들을 상기 비디오 유닛의 픽셀들에 할당하는 수단, 상기 초기 깊이 값들이 임계치를 만족하는지 여부를 확인하는 수단, 상기 초기 깊이 값들이 상기 임계치를 만족하는, 상기 비디오 유닛의 하나 이상의 픽셀들에 조정된 깊이 값들을 할당하는 수단으로서, 상기 조정된 깊이 값들은 상기 픽셀들과 연관된 컬러에 기초하는, 상기 조정된 깊이 값들을 할당하는 수단, 및 상기 비디오 유닛에 대한 깊이 맵을 생성하는 수단으로서, 상기 깊이 맵은 상기 픽셀들의 제 1 서브세트에 대한 상기 초기 깊이 값들 및 상기 픽셀들의 제 2 서브세트에 대한 상기 조정된 깊이 값들을 포함하는, 상기 깊이 맵을 생성하는 수단을 포함할 수도 있다.

[0013] 본 개시물에서 설명되는 기술들은 하드웨어, 소프트웨어, 펌웨어 또는 이들의 임의의 조합에서 구현될 수도 있다. 소프트웨어에서 구현되는 경우에, 소프트웨어는 마이크로프로세서, 주문형 집적 회로 (ASIC), 필드 프로그램가능한 게이트 어레이 (FPGA), 또는 디지털 신호 프로세서 (DSP) 와 같은 하나 이상의 프로세서들에서 실행될 수도 있다. 이러한 기술들을 실행하는 소프트웨어는 먼저 컴퓨터 판독가능한 매체에 저장되고, 프로세서에 로딩되고 실행될 수도 있다.

[0014] 따라서, 본 개시물은 또한, 명령들을 포함하는 컴퓨터 판독가능 저장 매체를 고려하며, 상기 명령들은, 프로세서에 의해 실행될 시, 상기 프로세서로 하여금, 비디오 시퀀스의 이전 비디오 유닛의 대응하는 픽셀들에 대한 비디오 유닛의 픽셀들과 연관된 모션에 기초하여 초기 깊이 값들을 상기 비디오 유닛의 픽셀들에 할당하게 하고, 상기 초기 깊이 값들이 임계치를 만족하는지 여부를 확인하게 하고, 상기 초기 깊이 값들이 상기 임계치를 만족하는, 상기 비디오 유닛의 하나 이상의 픽셀들에 조정된 깊이 값들을 할당하게 하는 것으로서, 상기 조정된 깊이 값들은 상기 픽셀들과 연관된 컬러에 기초하는, 상기 조정된 깊이 값들을 할당하게 하며, 및 상기 비디오 유닛에 대한 깊이 맵을 생성하게 하는 것으로서, 상기 깊이 맵은 상기 픽셀들의 제 1 서브세트에 대한 상기 초기 깊이 값들 및 상기 픽셀들의 제 2 서브세트에 대한 상기 조정된 깊이 값들을 포함하는, 상기 깊이 맵을 생성하게 한다.

[0015] 깊이 맵 초기화 이후에, 본 개시물에 부합하여, 최종 깊이 맵을 정의하기 위해 추가의 기술들이 적용될 수도 있다. 이러한 실시예들에서, 원래 생성된 깊이 맵은 초기화된 깊이 맵으로 지칭될 수도 있다. 이 경우에, 방법은, 상기 비디오 유닛이 상기 이전 비디오 유닛에 대한 장면 변화에 대응하는지 여부를 결정하는 단계, 상기 비디오 유닛이 상기 이전 비디오 유닛에 대한 장면 변화에 대응한다면, 상기 초기화된 깊이 맵을 상기 비디오 유닛에 대한 최종 깊이 맵으로서 선택하는 단계, 상기 비디오 유닛이 상기 이전 비디오 유닛에 대한 장면 변화에 대응하지 않는다면, 상기 비디오 유닛이 상기 이전 비디오 유닛에 대하여 낮은 레벨의 모션을 나타내는지 여부를 결정하는 단계, 상기 비디오 유닛이 상기 낮은 레벨의 모션을 나타내지 않는다면, 상기 비디오 유닛에 대한 상기 초기화된 깊이 맵의 깊이 값들과 상기 이전 비디오 유닛에 대한 이전 깊이 맵의 깊이 값들의 가중된 평균들에 기초하여 상기 최종 깊이 맵을 정의하는 단계, 및 상기 비디오 유닛이 상기 낮은 레벨의 모션을 나타낸다면, 상기 비디오 유닛에 대한 상기 초기화된 깊이 맵의 깊이 값들과 상기 이전 비디오 유닛에 대한 상기 이전 깊이 맵의 깊이 값들의 최대치들에 기초하여 상기 최종 깊이 맵을 정의하는 단계를 더 포함할 수도 있다.

[0016] 본 기술들의 하나 이상의 양태들의 세부사항을 첨부한 도면 및 하기의 설명에서 설명한다. 본 개시물에서 기술되는 기술들의 다른 특징들, 목적들 및 이점들은 상세한 설명 및 도면과 청구범위로부터 명백해질 것이다.

도면의 간단한 설명

[0017] 도 1 은 본 개시물의 기술들을 구현할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템을 예시한 블록도이다.

도 2 는 비디오 인코딩 프로세스의 일부로서 본 개시물의 기술들을 수행할 수도 있는 예시적인 비디오 인코더를 예시한 블록도이다.

도 3 은 비디오 디코딩 프로세스의 일부로서 본 개시물의 기술들을 수행할 수도 있는 예시적인 비디오 디코더를 예시한 블록도이다.

도 4 는 깊이 추정 및 뷰 합성의 프로세스를 예시한 흐름도이다.

도 5 는 깊이 맵을 초기화하는 프로세스를 예시한 흐름도이다.

도 6 은 최종 깊이 맵을 생성하는 프로세스를 예시한 흐름도이다.

도 7 은 깊이 맵 초기화 및 최종 깊이 맵의 생성을 예시한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

- [0018] 본 개시물은 비디오 프레임들 또는 비디오 프레임들의 슬라이스들과 같은 비디오 유닛들에 대한 깊이 맵들을 생성하기 위한 기술들을 설명한다. 그 기술들은 2 차원 (2D) 비디오를 3 차원 (3D) 비디오로 컨버팅하기 위해 비디오 인코더에 의해 수행될 수도 있다. 그 기술들은 대안적으로, 수신된 2D 비디오를 3D 비디오로 컨버팅하기 위해 비디오 디코더에 의해 수행될 수도 있다. 용어 "코딩" 은 여기서 비디오 인코딩 또는 비디오 디코딩을 지칭하도록 정의된다. 유사하게, 어구 "비디오 코더" 는 비디오 인코더 또는 비디오 디코더를 지칭한다. 일반적으로, 깊이 맵 초기화 및 생성은 본 개시물에 부합하는 인코더 또는 디코더에 의해 수행될 수도 있다.
- [0019] 본 개시물의 기술들은 깊이 맵 초기화 프로세스를 포함할 수도 있다. 깊이 맵 초기화를 위해, 초기 깊이 값들은 비디오 시퀀스의 이전 비디오 유닛의 대응하는 픽셀들과 관련된 픽셀들과 연관된 모션에 기초하여 비디오 유닛의 픽셀들에 할당될 수도 있다. 그 후에, 임계치를 만족하는 초기 깊이 값들은 조정될 수도 있고, 여기서 조정된 깊이 값들은 픽셀들과 연관된 컬러에 기초한다. 그 후에, 비디오 유닛에 대하여 (초기화된 깊이 맵으로 지칭될 수도 있는) 깊이 맵이 생성될 수 있고, 여기서 깊이 맵은 픽셀들의 제 1 서브세트에 대한 초기 깊이 값들 및 픽셀들의 제 2 서브세트에 대한 조정된 깊이 값들을 포함한다.
- [0020] (초기화된 깊이 맵이라 불릴 수도 있는) 초기화 동안 생성된 깊이 맵에 기초하여 최종 깊이 맵을 정의하기 위해 추가의 기술들이 적용될 수도 있다. 예를 들면, 그 기술들은 비디오 유닛이 이전의 비디오 유닛에 대한 장면 변화에 대응하는지 여부를 결정할 수도 있다. 만약 비디오 유닛이 장면 변화에 대응한다면, 기술들은 초기화된 깊이 맵을 비디오 유닛에 대한 최종 깊이 맵으로서 선택할 수도 있다. 그러나, 비디오 유닛이 장면 변화에 대응하지 않는다면, 기술들은 비디오 유닛이 이전의 비디오 유닛에 대하여 낮은 레벨의 모션을 나타내는지 여부를 결정할 수도 있다. 만약, 비디오 유닛이 낮은 레벨의 모션을 나타내지 않는다면, 기술들은 비디오 유닛에 대한 초기화된 깊이 맵의 깊이 값들과 이전 비디오 유닛에 대한 이전 깊이 맵의 깊이 값들의 가중된 평균들에 기초하여 최종 깊이 맵을 정의할 수도 있다. 비디오 유닛이 낮은 레벨의 모션을 나타낸다면, 기술들은 비디오 유닛에 대한 초기화된 깊이 맵의 깊이 값들과 이전 비디오 유닛의 이전 깊이 맵의 깊이 값들의 최대치들에 기초하여 최종 깊이 맵을 정의할 수도 있다.
- [0021] 도 1 은 본 개시물의 기술들을 구현할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템 (10) 을 예시한 블록도이다. 도 1 에 도시된 바와 같이, 시스템 (10) 은 인코딩된 비디오를 통신 채널 (15) 을 통해 목적지 디바이스 (16) 로 송신하는 소스 디바이스 (12) 를 포함한다. 소스 디바이스 (12) 및 목적지 디바이스 (16) 는 모바일 디바이스들 또는 일반적으로 고정된 디바이스들을 포함하는 광범위한 디바이스들 중 임의의 것을 포함할 수도 있다. 일부 경우에, 소스 디바이스 (12) 및 목적지 디바이스 (16) 는 무선 핸드셋들, 소위 셀룰러 또는 위성 무선 전화기들, 개인 휴대 정보 단말기들 (PDAs), 모바일 미디어 플레이어들, 또는 무선일 수도 있거나 무선이 아닐 수도 있는 통신 채널 (15) 을 통해 비디오 정보를 통신할 수 있는 임의의 디바이스와 같은 무선 통신 디바이스들을 포함한다. 그러나, 2D - 3D 비디오 컨버전을 위한 깊이 맵들의 생성 및 적용에 관한 본 개시물의 기술들은, 무선, 유선 또는 하이브리드 시스템들을 포함하는 다수의 서로 다른 시스템들 및 세팅들에서 사용될 수도 있다. 도 1 은 그러한 시스템의 하나의 실시예일 뿐이다.
- [0022] 도 1 의 실시예에서, 소스 디바이스 (12) 는 비디오 소스 (20), 비디오 인코더 (22), 변조기/복조기 (모뎀) (23) 및 송신기 (24) 를 포함할 수도 있다. 목적지 디바이스 (16) 는 수신기 (26), 모뎀 (27), 비디오 디코더 (28) 및 디스플레이 디바이스 (30) 를 포함할 수도 있다. 본 개시물에 따라, 소스 디바이스 (12) 의 비디오 인코더 (22) 또는 수신 디바이스의 비디오 인코더 (28) 는 2D - 3D 비디오 컨버전을 위한 깊이 맵들을 생성하도록 구성될 수도 있다.
- [0023] 소스 디바이스 (12) 는 비디오 정보를 인코딩하여 목적지 디바이스 (16) 로 전송할 수도 있다. 목적지 디바이스 (16) 는 소스 디바이스 (12) 로부터 수신된 무선 신호들을 수신하여 복조할 수도 있다. 소스 디바이스 (12) 와 목적지 디바이스 (16) 는 코딩 디바이스들의 실시예들이다. 예를 들어, 소스 디바이스 (12) 는 목적지 디바이스 (16) 로의 전송을 위해 코딩된 비디오 데이터를 생성하는 코딩 디바이스를 포함할 수도 있다. 일부 경우들에서, 디바이스들 (12, 16) 은 디바이스들 (12, 16) 각각이 비디오 인코딩 및 디코딩 컴포넌트들을 포함하도록 실질적으로 대칭적인 방식으로 동작할 수도 있다. 따라서, 시스템 (10) 은 예컨대 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅 또는 비디오 전화를 위해 비디오 디바이스들

(12, 16) 사이의 일 방향 또는 양방향 비디오 전송을 지원할 수도 있다.

[0024]

소스 디바이스 (12) 의 비디오 소스 (20) 는 비디오 카메라와 같은 비디오 캡처 디바이스, 이전에 캡처된 비디오를 저장하는 비디오 아카이브 (archive), 또는 비디오 콘텐츠 제공자로부터의 비디오 피드를 포함할 수도 있다. 추가의 대안으로서, 비디오 소스 (20) 는 소스 비디오로서 컴퓨터 그래픽-기반 데이터, 또는 라이브 비디오, 보관된 비디오, 및 컴퓨터-생성된 비디오의 조합을 생성할 수도 있다. 일부 경우에, 비디오 소스 (20) 가 비디오 카메라이면, 소스 디바이스 (12) 및 목적지 디바이스 (16) 는 소위 카메라폰 또는 비디오 오픈을 형성한다. 각각의 경우에, 캡처되거나, 미리 캡처되거나, 컴퓨터-생성된 비디오는 비디오 인코더 (22) 에 의해 인코딩될 수도 있다. 인코딩된 비디오 정보는 예컨대, 코드 분할 다중 액세스 (CDMA), 모바일 통신용 글로벌 시스템 (GSM), 주파수 분할 다중 접속 (FDMA), 시간 분할 다중 접속 (TDMA), "wifi", 블루투스, 임의의 광대역 통신 또는 임의의 다른 통신 기술, 표준 또는 이들의 조합들과 같은 통신 표준에 따라 모뎀 (23) 에 의해 변조될 수도 있다. 그 후에, 변조된 정보는 송신기 (24) 를 통해 목적지 디바이스 (16) 로 전송될 수도 있다. 모뎀 (23) 은 신호 변조를 위해 설계된 다양한 믹서들, 필터들, 증폭기들 또는 다른 컴포넌트들을 포함할 수도 있다. 송신기 (24) 는 증폭기들, 필터들, 및 하나 이상의 안테나들을 포함하는, 데이터를 전송하기 위해 설계된 회로들을 포함할 수도 있다.

[0025]

목적지 디바이스 (16) 의 수신기 (26) 는 채널 (15) 을 통해 정보를 수신하고, 모뎀 (27) 은 그 정보를 복조한다. 상이한 실시예들에서, 소스 디바이스 (12) 의 비디오 인코딩 프로세스 또는 목적지 디바이스 (16) 의 비디오 디코딩 프로세스는 깊이 맵들을 생성하기 위해 여기서 설명된 기술들 중 하나 이상을 구현할 수도 있다. 일부 경우에, 채널 (15) 을 통해 통신된 정보는 소스 디바이스 (12) 에 의해 생성된 깊이 맵들을 포함할 수도 있고, 다른 경우에, 깊이 맵들은 소스 디바이스 (12) 로부터 통신된 2D 비디오 프레임들에 기초하여 목적지 디바이스 (16) 에서 생성될 수도 있다. 디스플레이 디바이스 (30) 는 디코딩된 비디오 데이터를 사용자에게 디스플레이하고, 음극선관, 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 타입의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 것을 포함할 수도 있다. 디스플레이 디바이스 (30) 는 멀티-뷰 렌더링을 위한 3D 능력들을 가질 수도 있다.

[0026]

도 1 의 실시예에서, 통신 채널 (15) 은 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적인 전송 라인들과 같은 임의의 무선 또는 유선 통신 매체, 또는 무선 및 유선 매체의 임의의 조합을 포함할 수도 있다. 따라서, 모뎀 (23) 및 송신기 (24) 는 다수의 가능한 무선 프로토콜들, 유선 프로토콜들, 또는 유선 및 무선 프로토콜들을 지원할 수도 있다. 통신 채널 (15) 은 근거리 네트워크 (LAN) 와 같은 패킷-기반 네트워크, 광역 네트워크 (WAN), 또는 하나 이상의 네트워크들의 상호 접속을 포함하는, 인터넷과 같은 글로벌 네트워크의 일부를 형성할 수도 있다. 통신 채널 (15) 은 일반적으로 소스 디바이스 (12) 로부터 목적지 디바이스 (16) 로 비디오 데이터를 전송하기 위한, 임의의 적절한 통신 매체 또는 서로 다른 통신 매체의 집합을 나타낸다. 통신 채널 (15) 은 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (16) 로의 통신을 용이하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다. 본 개시물의 기술들은 반드시 하나의 디바이스로부터 다른 디바이스로 인코딩된 데이터의 통신을 요구하는 것이 아니며, 상호 디코딩 없이 인코딩 시나리오들에 적용할 수도 있다. 또한, 본 개시물의 양태들은 상호 인코딩 없이 디코딩 시나리오들에 적용할 수도 있다.

[0027]

비디오 인코더 (22) 및 비디오 디코더 (28) 는 대안적으로 MPEG-4, Part 10, AVC (Advanced Video Coding) 로서 설명되는 ITU-T H.264 표준과 같은 비디오 압축 표준과 실질적으로 일치하게 동작할 수도 있다. 그러나, 본 개시물의 기술들은 임의의 특정 코딩 표준 또는 그 확장에 제한되지 않는다. 도 1 에 도시되지는 않았지만, 일부 양태에서, 비디오 인코더 (22) 및 비디오 디코더 (28) 가 오디오 인코더 및 디코더와 각각 통합될 수도 있고, 적절한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함하여, 공통 데이터 스트림 또는 개별 데이터 스트림들에서 오디오 및 비디오 양쪽 모두의 인코딩을 핸들링할 수도 있다. 적용가능하다면, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 사용자 데이터그램 프로토콜 (UDP) 과 같은 다른 프로토콜들에 따를 수도 있다.

[0028]

ITU-T H.264/MPEG-4 (AVC) 표준은 JVT (Joint Video Team) 로 알려진 집단 파트너십의 제품으로서 ISO/IEC MPEG (Moving Picture Experts Group) 과 함께 ITU-T VCEG (Video Coding Experts Group) 에 의해 만들어졌다. H.264 표준은 2005 년 3 월에 ITU-T Study Group 에 의한, 일반적인 시청각 서비스들을 위한 ITU-T 권고 H.264, AVC (Advanced Video Coding) 에서 설명되며, 이는 본 명세서에서 H.264 표준 또는 H.264 규격, 또는 H.264/AVC 표준 또는 규격으로 지칭될 수도 있다. JVT (Joint Video Team) 은 H.264/MPEG-4

AVC 로의 확장을 위해 계속해서 노력하고 있다.

- [0029] H.264/MPEG-4 AVC 표준을 강화시키는 작업은 KTA (Key Technologies Area) 포럼과 같은 ITU-T 의 다양한 포럼들에서 발생한다. KTA 포럼은 H.264/AVC 표준에 의해 보여지는 것보다 더 높은 코딩 효율을 보이는 코딩 기술들을 개발하려 한다. 본 개시물에서 설명되는 기술들은 특히 3D 비디오를 위한 H.264/AVC 표준 및 2D - 3D 비디오 컨버전과 관련하여 코딩 개선들을 제공할 수도 있다.
- [0030] 비디오 인코더 (22) 및 비디오 디코더 (28) 각각은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서 (DSP) 들, 주문형 집적회로 (ASIC) 들, 필드 프로그램가능 게이트 어레이 (FPGA) 들, 개별 로직 회로, 마이크로프로세서 또는 다른 플랫폼에 실행하는 소프트웨어, 하드웨어, 펌웨어 또는 이들의 임의의 조합들로서 구현 될 수도 있다. 비디오 인코더 (22) 및 비디오 디코더 (28) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수 있고, 그 중 어느 하나는 각각의 모바일 디바이스, 가입자 디바이스, 브로드캐스트 디바이스, 서버 등등 내에 결합된 인코더/디코더 (CODEC) 의 일부로서 통합될 수 있다.
- [0031] 비디오 시퀀스는 통상적으로 일련의 비디오 프레임들을 포함한다. 비디오 인코더 (22) 및 비디오 디코더 (28) 는 비디오 데이터를 인코딩하고 디코딩하기 위해 개개의 비디오 프레임들 내의 픽셀들의 블록들 상에서 동작할 수도 있다. 비디오 블록들은 고정 또는 가변 사이즈들을 가질 수도 있고, 특정된 코딩 표준에 따라 사이즈가 달라질 수도 있다. 각각의 비디오 프레임은 일련의 슬라이스들 또는 다른 독립적으로 디코딩 가능한 유닛들을 포함할 수도 있다. 각각의 슬라이스는 일련의 매크로블록들을 포함할 수도 있고, 그 매크로블록들은 서브-블록들로 배열될 수도 있다. 일 실시예로서, ITU-T H.264 표준은, 루마 성분들에 대하여 16×16, 8×8, 4×4 및 크로마 성분들에 대하여 8×8 과 같은 다양한 블록 사이즈들에서의 인트라 예측 (intra prediction) 뿐만 아니라, 루마 성분들에 대하여 16×16, 16×8, 8×16, 8×8, 8×4, 4×8 및 4×4 및 크로마 성분들에 대하여 대응하는 스케일링된 사이즈들과 같은 다양한 블록 사이즈들에서의 인터 예측 (inter prediction) 을 지원한다. 비디오 블록들은 예컨대, 이산 코사인 변환과 같은 변환 프로세스 또는 개념적으로 유사한 변환 프로세스들 이후에 변환 계수들의 블록들 또는 나머지 픽셀 데이터의 블록들을 포함할 수도 있다.
- [0032] 비디오 블록들이 작을수록 더 양호한 분해능을 제공할 수 있고, 이는 높은 세부 레벨을 포함하는 비디오 프레임의 위치들을 위해 이용될 수도 있다. 일반적으로, 매크로블록들 및 다양한 서브-블록들 또는 파티션들은 모두 비디오 블록들인 것으로 고려될 수도 있다. 또한, 하나의 슬라이스는 매크로블록들 및/또는 서브-블록들 또는 파티션들과 같은 일련의 비디오 블록들인 것으로 고려될 수도 있다. 일반적으로, 매크로블록은 픽셀들의 16×16 영역을 정의하는 크로미넌스 및 루미넌스 값들의 세트를 지칭할 수도 있다. 루미넌스 블록은 값들의 16×16 세트를 포함할 수도 있지만, 8×8 블록들, 4×4 블록들, 8×4 블록들, 4×8 블록들 또는 다른 사이즈들과 같은 더 작은 비디오 블록들로 추가로 분할될 수도 있다. 2 개의 서로 다른 크로미넌스 블록들은 매크로 블록에 대한 컬러를 정의할 수도 있고, 각각 픽셀들의 16×16 영역과 연관된 컬러 값들의 8×8 서브-샘플링된 블록들을 포함한다. 매크로블록들은 매크로블록들에 적용된 코딩 모드들 및/또는 코딩 기술들을 정의하기 위한 신택스 (syntax) 정보를 포함할 수도 있다.
- [0033] 매크로블록들 또는 다른 비디오 블록들은 슬라이스들, 프레임들 또는 다른 독립적인 유닛들과 같은 디코딩 가능한 유닛들로 그룹화될 수도 있다. 각각의 슬라이스는 비디오 프레임의 독립적으로 디코딩 가능한 유닛일 수도 있다. 대안적으로, 프레임들은 자체적으로 디코딩 가능한 유닛들이 될 수도 있거나, 프레임의 다른 부분들이 디코딩 가능한 유닛들로 정의될 수도 있다. 본 개시물에서, 용어 "코딩된 유닛" 은 전체 프레임, 프레임의 슬라이스, 픽처 그룹 (GOPs), 또는 이용되는 코딩 기술들에 따라 정의되는 다른 독립적으로 디코딩 가능한 유닛과 같은 비디오 프레임의 임의의 독립적으로 디코딩 가능한 유닛을 지칭한다.
- [0034] 인트라-기반 또는 인터-기반 예측 인코딩 이후 및 (H.264/AVC 에서 이용되는 4×4 또는 8×8 정수 변환 또는 이산 코사인 변환 또는 DCT 와 같은) 임의의 변환들 이후에, 양자화가 수행될 수도 있다. 양자화는 일반적으로 계수들을 나타내기 위해 이용되는 데이터의 양을 감소시킬 수 있도록 계수들이 양자화되는 프로세스를 지칭한다. 양자화 프로세스는 계수들 중 일부 또는 전부와 연관된 비트 깊이를 감소시킬 수도 있다. 예를 들면, 16-비트 값은 양자화 동안 15-비트 값으로 절단 (round down) 될 수도 있다. 양자화 이후, 예컨대, 콘텐츠 적응형 가변 길이 코딩 (CAVLC), 컨텍스트 적응형 이진 산술 코딩 (CABAC), 또는 다른 엔트로피 코딩 방법에 따라 엔트로피 코딩이 수행될 수도 있다.
- [0035] 3D 비디오는 각각의 최초 인코딩된 프레임과 연관된 하나 이상의 추가의 비디오 프레임들 (예컨대, 추가의 뷰들) 을 요청할 수도 있다. 예를 들어, 2 개의 서로 다른 뷰들은 비디오 프레임의 입체적인 3D 렌더션

(rendition) 을 정의하기 위해 이용될 수도 있다. 2 개의 뷰들, 3 개의 뷰들, 3 개 초과인 뷰들을 포함할 수도 있는 다중 뷰들은 멀티-뷰 3D 렌더션들을 지원할 수도 있다. 3D 비디오의 서로 다른 뷰들은 2 이상의 뷰들과 연관된 비디오 프레임들 또는 슬라이스들이 비디오 시퀀스의 동일한 시간 인스턴스들에 대응하도록 유사한 타이밍 또는 동기화를 가질 수도 있다. 이러한 방식으로, 2 이상의 뷰들은 일반적으로 3D 시퀀스를 함께 형성하는 2 이상의 2D 시퀀스들을 정의할 수도 있고, 이러한 3D 시퀀스는 3D 비디오를 제공하도록 집합적으로 렌더링될 수도 있다.

[0036] 초기의 2D 비디오 시퀀스에 기초한 3D 비디오 렌더링을 지원하기 위해, 본 개시물은 깊이 맵들의 생성을 설명한다. 깊이 맵들은 프레임 또는 슬라이스와 같은 비디오 유닛의 상이한 픽셀들에 대하여 깊이 값들을 포함할 수도 있다. 깊이 맵들은 인코더에서 생성될 수도 있고, 이 경우에 깊이 맵들은 비트스트림의 일부로서 디코더로 통신될 수도 있거나, 비트스트림으로 통신될 수 있는 하나 이상의 추가 뷰들을 생성하기 위해 인코더에서의 뷰 합성 프로세스에 적용될 수도 있다. 대안적으로, 깊이 맵들은 디코더에서 생성될 수도 있고, 이 경우에 인코더 디바이스는 간단히 2D 스트림을 디코더 디바이스로 전송하여 깊이 맵들을 생성할 수도 있다. 깊이 맵 초기화 및 최종 깊이 맵 생성 기술들은, 본 개시물에서 설명되는 것과 같이, 인코더에서 전체적으로 및 디코더에서 전체적으로, 또는 인코더에서 부분적으로 및 디코더에서 부분적으로 수행될 수도 있다. 깊이 맵들이 생성되면, 2D 시퀀스에 대한 하나 이상의 2차 뷰들을 생성하기 위해 뷰 합성 프로세스에서 이용될 수도 있으며, 따라서 2D 시퀀스는 3D 로 렌더링될 수 있다.

[0037] 도 2 는 비디오 인코딩 프로세스의 일부로서 본 개시물과 부합하는 기술들을 수행할 수도 있는 비디오 인코더 (50) 의 일 실시예를 예시한 블록도이다. 도 2 의 실시예에서, 깊이 맵들이 생성되어 비디오 비트스트림의 일부로서 통신될 수도 있다. 그러나, 언급된 것과 같이, 본 개시물은 인코더에서의 깊이 맵들의 생성 및 적용을 고려하며, 이 경우에 깊이 맵들 또는 생성된 3D 시퀀스는 비디오 인코더 (50) 로부터 또 다른 디바이스로 통신될 수도 있다.

[0038] 비디오 인코더 (50) 는 소스 디바이스 (12) 의 비디오 인코더 (22) 또는 다른 디바이스의 비디오 인코더에 대응할 수도 있다. 비디오 인코더 (50) 는 비디오 프레임들 내의 블록들의 인트라-코딩 및 인터-코딩을 수행할 수도 있다. 인트라-코딩은 공간적 예측에 의존하여, 소정의 비디오 프레임 내에서 비디오의 공간적 리던던시를 감소 또는 제거한다. 인터-코딩은 시간적 예측에 의존하여, 비디오 시퀀스의 인접 프레임들 내에서 비디오의 시간적 리던던시를 감소 또는 제거한다. 인트라-모드 (I-모드) 는 공간 기반 압축 모드를 지칭할 수도 있고, 예측 모드 (P-모드) 또는 양방향 모드 (B-모드) 와 같은 인터-모드들은 시간 기반 압축 모드들을 지칭할 수도 있다.

[0039] 도 2 에 도시된 바와 같이, 비디오 인코더 (50) 는 인코딩될 비디오 프레임 또는 슬라이스 내의 현재 비디오 블록을 수신한다. 도 2 의 실시예에서, 비디오 인코더 (50) 는 예측 유닛 (35), 메모리 (34), 합산기 (48), 변환 유닛 (38), 양자화 유닛 (40), 및 엔트로피 코딩 유닛 (46) 을 포함한다. 비디오 블록 복원을 위해, 비디오 인코더 (50) 는 또한 역 양자화 유닛 (42), 역변환 유닛 (44) 및 합산기 (51) 를 포함한다. 또한, 본 개시물에 따라, 비디오 인코더 (50) 는 본 명세서에서 설명되는 깊이 맵들을 생성하는, 깊이 맵 생성 유닛 (36) 을 포함할 수도 있다. 비디오 인코더 (50) 는 또한 복원된 비디오로부터 블록화 아티팩트들 (blockiness artifacts) 을 제거하기 위해 블록 경계들을 필터링하는 디블로킹 (deblocking) 필터 (비도시) 와 같은 다른 컴포넌트들을 포함할 수도 있다. 요구되는 경우, 디블로킹 필터는 통상적으로 합산기 (51) 의 출력을 필터링할 것이다.

[0040] 인코딩 프로세스 동안, 비디오 인코더 (50) 는 코딩될 비디오 블록을 수신하고, 예측 유닛 (35) 은 인트라-예측 또는 인터-예측 코딩을 수행한다. 예를 들어, 인코더 (50) 의 예측 유닛 (35) 은 코딩된 유닛 (예컨대, 프레임 또는 슬라이스) 의 각각의 비디오 블록 또는 비디오 블록 파티션에 대하여 모션 추정 및 모션 보상을 수행할 수 있다. 예측 유닛 (35) 은 특정 블록을 인코딩하는 것과 연관된 각각의 적용가능한 모드에 대한 레이트-왜곡 비용 (rdcost) 을 계산할 수도 있고, 최저 비용을 산출하는 코딩 모드를 선택할 수도 있다. rdcost 는 이용되는 비트들의 수 및 원래의 비디오 데이터에 대하여 코딩된 데이터에서의 왜곡의 레벨과 관련하여 비용을 정량화할 수도 있다.

[0041] 레이트-왜곡 (RD) 분석은 비디오 코딩에서 상당히 통상적이며, 일반적으로 코딩 비용을 나타내는 비용 메트릭의 계산을 수반한다. 비용 메트릭은 코딩을 위해 요구되는 비트들의 수 (레이트) 와 코딩과 연관된 품질 레벨 (왜곡) 사이에서 균형을 이룰 수도 있다. 통상적인 레이트-왜곡 비용 계산은 일반적으로 하기의 포맷에 대응할 수도 있다:

$$J(\lambda) = \lambda R + D,$$

[0042]

[0043] 상기 $J(\lambda)$ 는 비용이고, R 은 비트 레이트이고, D 는 왜곡이며, λ 는 라그랑제 (Lagrange) 승수이다. 예측 유닛 (35) 은 비디오 블록 인코딩을 수행하기 위해 이용될 수 있는 다양한 인트라-코딩 및 인터-코딩 모드들 (및 적용가능한 파티션 사이즈들) 을 비교하기 위해 상기와 같은 타입의 비용 함수를 적용할 수도 있다.

[0044]

요구되는 예측 데이터가 예측 유닛 (35) 에 의해 식별되면, 비디오 인코더 (50) 는 잔여 블록을 생성하기 위해 코딩되는 원래의 비디오 블록으로부터 예측 데이터를 감산함으로써 잔여 비디오 블록을 형성한다. 합산기 (48) 는 이러한 감산 연산들을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 변환 유닛 (38) 은 이산 코사인 변환 (DCT) 또는 개념적으로 유사한 변환과 같은 변환을 블록 내의 잔여 값들에 적용하여 잔여 변환 블록 계수들을 포함하는 비디오 블록을 생성한다. 변환 유닛 (38) 은 DCT 와 개념적으로 유사한, H.264 표준에 의해 정의되는 것과 같은 변환들을 수행할 수도 있다. 웨이블릿 (wavelet) 변환들, 정수 변환들, 서브-대역 변환들 또는 다른 타입의 변환들이 또한 이용될 수 있다. 임의의 경우에, 변환 유닛 (38) 은 변환을 잔여 블록에 적용하여 잔여 변환 계수들의 블록을 생성한다. 변환은 잔여 정보를 픽셀 영역으로부터 주파수 영역으로 컨버팅할 수도 있다.

[0045]

양자화 유닛 (40) 은 잔여 변환 계수들을 양자화하여, 비트 레이트를 더 감소시킨다. 양자화 프로세스는 이러한 계수들의 전부 또는 일부와 연관된 비트 깊이를 감소시킬 수도 있다. 예를 들면, m-비트 값은 양자화 동안 m-n-비트 값으로 절단될 수도 있으며, 여기서 m 과 n 은 비-제로이고, m 은 n 보다 크다. 추가로, 양자화 유닛 (40) 은 오프셋이 이용되는 경우들을 위해 서로 다른 오프셋들을 양자화할 수도 있다.

[0046]

양자화 이후에, 엔트로피 코딩 유닛 (46) 은 양자화된 변환 계수들을 엔트로피 코딩한다. 예를 들면, 엔트로피 코딩 유닛 (46) 은 콘텐츠 적응형 가변 길이 코딩 (CAVLC), 콘텍스트 적응형 이진 산술 코딩 (CABAC), 또는 다른 엔트로피 코딩 방법을 수행할 수도 있다. 엔트로피 코딩 유닛 (46) 에 의한 엔트로피 코딩 이후에, 인코딩된 비디오는 다른 디바이스로 전송되거나, 이후 전송 또는 검색을 위해 보관될 수도 있다. 코딩된 비트스트림은 엔트로피 코딩된 잔여 블록들, 이러한 블록들에 대한 모션 벡터들 및 2D - 3D 비디오 컨버전을 지원하기 위해 본 명세서에 설명된 깊이 맵들과 같은 다른 신택스를 포함할 수도 있다.

[0047]

역 양자화 유닛 (42) 및 역변환 유닛 (44) 은 각각 역 양자화 및 역 변환을 적용하여, 픽셀 영역 내의 잔여 블록을, 예컨대 전송된 방식에서 참조 데이터로서의 향후 이용을 위해 복원한다. 합산기 (51) 는 복원된 잔여 블록을 모션 보상 유닛 (35) 에 의해 생성된 1차 및/또는 2차 예측 블록들에 합산하여 메모리 (34) 내의 저장을 위한 복원된 비디오 블록을 생성한다. 복원된 비디오 블록 및 잔여 데이터는 모션 보상 유닛 (35) 에 의해 후속 비디오 프레임 또는 다른 코딩된 유닛 내의 블록을 인터-인코딩하기 위한 참조 블록으로서 이용될 수도 있다.

[0048]

3D 비디오를 지원하기 위해, 비디오 인코더 (50) 는 메모리 (34) 내에 저장된 복원된 2D 비디오 시퀀스와 관련하여 동작하는 깊이 맵 생성 유닛 (36) 을 추가로 포함할 수도 있다. 깊이 맵 생성 유닛 (36) 은 비디오 시퀀스의 이전 비디오 유닛의 대응하는 픽셀들과 관련된 픽셀들과 연관된 모션에 기초하여 초기 깊이 값들을 비디오 유닛의 픽셀들에 할당할 수도 있다. 깊이 맵 생성 유닛 (36) 은 초기 깊이 값들이 임계치를 만족하는지 여부, 예컨대 임의의 초기 깊이 값들이 정의된 임계치 이상인지, 또는 대안적으로, 정의된 임계치 미만인지 여부를 확인할 수도 있다. 깊이 맵 생성 유닛 (36) 은 초기 깊이 값들이 임계치를 만족하는 비디오 유닛의 하나 이상의 픽셀들에 조정된 깊이 값들을 할당할 수도 있고, 여기서 조정된 깊이 값들은 픽셀들과 연관된 컬러에 기초한다. 그 후에, 깊이 맵 생성 유닛 (36) 은 비디오 유닛에 대한 초기화된 깊이 맵을 생성할 수도 있고, 여기서 초기화된 깊이 맵은 임계값을 만족하지 않는 픽셀들의 제 1 서브세트에 대한 초기 깊이 값들 및 임계값을 만족하는 픽셀들의 제 2 서브 세트에 대한 조정된 깊이 값들을 포함한다. 다시 말해서, 대안적인 실시예들에서, 임계치의 만족은 임계값보다 크거나, 다른 실시예들에서 임계값 미만인 깊이 값들과 관련하여 고려될 수도 있다.

[0049]

깊이 맵 생성 유닛 (36) 은 이전 비디오 유닛의 공동-위치된 픽셀들의 루마 값들에 대한 비디오 유닛의 픽셀들의 루마 값들에 기초하여 초기 깊이 값들을 할당함으로써 모션에 기초하여 초기 깊이 값들을 할당할 수도 있다. 깊이 맵 생성 유닛 (36) 은 비디오 유닛의 픽셀들의 크로마 값들에 적어도 부분적으로 기초하여 조정된 깊이 값들을 할당함으로써 컬러에 기초하여 조정된 깊이 값들을 할당할 수도 있다. 예를 들어, 하기에 더 상세히 설명되는 것과 같이, 모션에 기초하여 초기 깊이 값들을 할당하는 것은 하기의 식에 따라 초기 깊이 값들을 할당하는 것을 포함할 수도 있다:

$$m_{n,i} = |L_{n,i} - L_{n-1,i}|$$

[0050]

[0051] 상기 $m_{n,i}$ 는 초기 깊이 값을 나타내고, $L_{n,i}$ 는 비디오 유닛 n 에서 픽셀 i 에 대한 루마 값을 나타내며, $L_{n-1,i}$ 는 이전의 비디오 유닛 n-1 에서 픽셀 i 에 대한 대응하는 루마 값을 나타낸다. 컬러에 기초하여 조정된 깊이 값들을 할당하는 것은 하기의 식에 따라 조정된 깊이 값들을 할당하는 것을 포함할 수도 있다:

$$d_{n,i}^g = g(c_{n,i}) = \max\left(0, \min\left(T_m, \frac{c_{n,i} - c_{\min}}{c_{\max} - c_{\min}} \times T_m\right)\right)$$

[0052]

[0053] 상기 $d_{n,i}^g$ 는 조정된 깊이 값을 나타내고, $g(c_{n,i})$ 는 크로마 값들과 관련된 함수 g 를 나타내고, T_m 는 임계치이고, $c_{n,i}$ 는 비디오 유닛 n 의 소정의 픽셀에 대한 크로마 값을 나타내며, c_{\max} 는 허용되는 최대 크로마 값을 나타내고, c_{\min} 는 허용되는 최소 크로마 값을 나타낸다.

[0054]

깊이 맵 생성 유닛 (36) 은 또한 초기화된 깊이 맵에 적어도 부분적으로 기초하여 최종 깊이 맵을 생성하기 위해 추가의 기술들을 수행할 수도 있다. 예를 들면, 깊이 맵 생성 유닛 (36) 은 비디오 유닛이 이전의 비디오 유닛에 대한 장면 변화에 대응하는지 여부를 결정하고, 비디오 유닛이 이전 비디오 유닛에 대한 장면 변화와 대응한다면 초기화된 깊이 맵을 비디오 유닛에 대한 최종 깊이 맵으로서 선택할 수도 있다. 비디오 유닛이 이전 비디오 유닛에 대한 장면 변화에 대응하지 않는다면, 깊이 맵 생성 유닛 (36) 은 비디오 유닛이 이전 비디오 유닛에 대하여 낮은 레벨의 모션을 나타내는지 여부를 결정할 수도 있다. 이 경우에, 모션은 현재 비디오 유닛과 이전 비디오 유닛 사이의 루마 차이들의 크기에 의해 정량화될 수도 있다. 만약, 비디오 유닛이 낮은 레벨의 모션을 나타내지 않는다면, 깊이 맵 생성 유닛 (36) 은 비디오 유닛에 대한 초기화된 깊이 맵의 깊이 값들과 이전 비디오 유닛에 대한 이전 깊이 맵의 깊이 값들의 가중된 평균들에 기초하여 최종 깊이 맵을 정의할 수도 있다. 비디오 유닛이 낮은 레벨의 모션을 나타낸다면, 깊이 맵 생성 유닛 (36) 은 비디오 유닛에 대한 초기화된 깊이 맵의 깊이 값들과 이전 비디오 유닛의 이전 깊이 맵의 깊이 값들의 최대치들에 기초하여 최종 깊이 맵을 정의할 수도 있다.

[0055]

최종 깊이 맵이 생성되면, 깊이 맵은 다른 디바이스에서의 뷰 합성에 적용하여 원래의 뷰와 최종 깊이 맵에 기초하여 하나 이상의 2차 뷰들을 생성하기 위해 비디오 인코더 (50) 로부터 다른 디바이스로 통신될 수도 있다. 대안적으로, 비디오 인코더 (50) 는 비디오 유닛에 대한 2차 뷰를 생성하기 위해 최종 깊이 맵에 기초하여 뷰 합성을 수행할 수 있고, 이 경우에 (깊이 맵 보다) 2차 뷰는 비디오 유닛의 일부로서 비디오 인코더 (50) 로부터 다른 디바이스로 통신될 수도 있다.

[0056]

도 3 은 본 개시물의 기술들을 수행하도록 구성될 수도 있는 예시적인 비디오 디코더 (70) 을 예시한 블록도이다. 이 경우에, 비디오 디코더 (70) 는 2D 비디오 시퀀스를 수신할 수도 있고, 2D 시퀀스의 비디오 프레임들에 대한 깊이 맵들을 생성할 수도 있다. 깊이 맵들은 비디오 프레임들에 대한 2차 뷰들을 합성하기 위해 비디오 디코더 (70) 에 의해 이용될 수도 있으며, 따라서 2D 시퀀스는 3D 시퀀스로 컨버팅될 수 있다.

[0057]

비디오 디코더 (70) 는 엔트로피 디코딩 유닛 (72), 예측 유닛 (75), 역 양자화 유닛 (76), 역변환 유닛 (78), 메모리 (74), 및 합산기 (79) 를 포함할 수도 있다. 예측 유닛 (75) 은 모션 보상 유닛뿐만 아니라 공간 예측 컴포넌트들을 포함할 수도 있다.

[0058]

비디오 디코더 (70) 는 인코딩된 2D 비디오 시퀀스를 포함하는 인코딩된 비디오 비트스트림을 수신할 수도 있다. 엔트로피 디코딩 유닛 (72) 은 2D 시퀀스의 잔여 비디오 블록들에 대하여 양자화된 계수들을 생성하기 위해 비트스트림의 엔트로피 디코딩을 수행한다. 엔트로피 디코딩 유닛 (72) 은 비트스트림으로부터 신택스 엘리먼트들을 분석하여 이러한 신택스 엘리먼트를 비디오 디코더 (70) 의 다양한 유닛들로 포워딩할 수도 있다. 모션 정보 (예컨대, 모션 벡터들) 및 다른 신택스 정보는 예컨대, 예측 유닛 (75) 으로 포워딩될 수도 있다.

[0059]

역 양자화 유닛 (76) 은 엔트로피 디코딩 유닛 (72) 으로부터 데이터를 역 양자화하고, 역변환 유닛은 역변환을 수행하여 픽셀 영역 내에 잔여 블록을 생성한다. 그 후에, 합산기 (79) 는 예측 유닛 (75) 에 의해 생성된 예측 블록과 잔여 블록을 결합하여 원래의 비디오 블록의 복원을 생성하며, 이는 메모리 (74) 내에 저장되고 및/또는 디코딩된 2D 비디오 데이터로서 디스플레이로 출력될 수도 있다. 다수의 비디오 블록들은 비디오 프레임들을 복원하고, 궁극적으로는 메모리 (74) 내의 비디오 프레임들의 전체 2D 시퀀스를 복원하기

위해 상기와 같은 방식으로 디코딩될 수도 있다. 이러한 방식으로, 비디오 디코더 (70) 는 전송된 비디오 인코더 (50) 에 의해 수행된 인코딩에 대하여 상호 디코딩을 수행한다.

- [0060] 본 개시물에 따라, 비디오 디코더 (70) 는 추가로 깊이 맵 생성 유닛 (77) 및 2D - 3D 컨버전 유닛 (79) 을 포함할 수도 있다. 깊이 맵 생성 유닛 (77) 은 본 개시물에 부합하는 깊이 맵 초기화 및 깊이 맵 생성 기술들을 수행할 수도 있다. 깊이 맵들이 생성되면, 2D - 3D 컨버전 유닛 (79) 은 비디오 프레임들에 대하여 하나 이상의 2차 뷰들을 생성하기 위해 뷰 합성 프로세스에서 깊이 맵들에 적용될 수도 있다.
- [0061] 일반적으로, 입체적인 디스플레이 기술의 개발에 따라, 입체적인 또는 3D 비디오는 엔터테인먼트 산업에서 유행하는 트렌드가 되고 있다. 종래의 2D 디스플레이와 비교하여 몰입형인 3D 경험으로 인해, 영화, 텔레비전 및 게임의 3D 콘텐츠에 대한 요구들이 계속해서 급격하게 증가하고 있다.
- [0062] 컴퓨터 그래픽들에서, 3D 렌더링은 3D 모델을 3D 포토리얼리스틱 효과들을 갖는 2D 이미지들로 자동으로 컨버팅하는 프로세스일 수도 있다. 컴퓨터 그래픽에서 때때로 몰입감 (immersiveness) 이라 지칭되는 포토리얼리스틱 효과들은 통상적으로 상이한 입체적인 뷰들의 뷰 각도 또는 위치를 서로에 대하여 변화시킴으로써 획득된다. 3D 비디오 애플리케이션들에서, 이러한 시나리오는 때때로 프리-뷰포인트 (free-viewpoint) 비디오라 불린다.
- [0063] 비디오 프로세싱에서, 3D 렌더링은 또한 뷰어들이 3D 안경을 이용하거나 이용하지 않고 스테레오 또는 오토 스테레오 뷰잉 경험을 가질 수 있도록 하는 몰입형 렌더링의 애플리케이션들을 포함할 수도 있다. 이 경우에, 디스플레이는 2 이상의 뷰들을 동시에 렌더링하는 능력을 가질 수도 있다. 가장 최근의 3D 디스플레이 기술들은 두 눈으로 보는 시력, 즉 좌측 및 우측 눈에 의해 감지되는 약간 상이한 뷰들을 이용하며, 그 결과 인간의 뇌에 의해 깊이에 대한 정신적인 느낌 (mental impression) 을 발생한다. 예를 들어, 일부 3D 시스템들은 스테레오 또는 오토 스테레오 디스플레이에서 2 이상의 뷰들을 허용한다.
- [0064] 다양한 3D 디스플레이 시스템들은 3D 비디오를 디스플레이할 수도 있지만, 종종 3D 렌더링에 필요한 추가의 데이터로 인해 비디오 통신에 대하여 증가된 대역폭들을 요구할 수도 있다. 추가로, 몇몇 뷰들을 동시에 획득하는 것은 종종 비디오 캡처 시스템들에서 사용될 수 없다. 또한, 상업적인 마켓 및 개인 레코딩들 양자에서 엄청난 양의 2D 콘텐츠가 존재하며, 이들을 위해 3D 비디오로 컨버팅하는 것은 바람직할 수도 있다. 따라서, 2D 로부터 3D 로의 데이터의 컨버전은 매우 바람직하다.
- [0065] 2D - 3D 비디오 컨버전 문제는 이미 알려진 원래의 2D 뷰로부터 하나의 가상 뷰 (즉, 2차 뷰) 또는 몇몇 가상 뷰들을 생성하는 문제로서 간주될 수 있다. 2차 뷰의 생성은 오브젝트 깊이 값들을 추정하고, 그 후에 그 깊이 값들을 적용하여 하나 이상의 2차 뷰들을 합성함으로써 달성될 수도 있다. 깊이 추정은 모노스코프 (monoscopic) 비디오 또는 몇몇 뷰들로부터의 카메라 면과 오브젝트들 사이의 절대적인 또는 상대적인 거리들을 추정하는 프로세스이다. 통상적으로 그레이-레벨 이미지로 표시되는 추정된 깊이 정보는 깊이 이미지 기반 렌더링 (DIBR) 기술들에 기초하여 임의의 적절한 각도의 가상 뷰들을 생성하는데 이용될 수 있다.
- [0066] 도 4 는 모노스코프 좌측-뷰로부터 우측-뷰를 생성하는 예시적인 뷰 합성 프로세스를 예시한 흐름도이다. 이 경우에, 좌측 뷰에 대한 2D 비디오 (401) 는 이 상기 프로세스로의 입력을 포함할 수도 있다. 각각의 비디오 프레임에 대하여, 깊이 맵 생성 유닛 (77) 은 좌측 뷰에 적용될 깊이 맵을 생성하기 위해 깊이 추정 (402) 을 수행할 수도 있다. 특히, 깊이 맵 생성 유닛 (77) 은 본 개시물에 설명되는 것과 같은 깊이 맵 초기화 및 깊이 맵 생성 기술들을 수행할 수도 있다. 깊이 맵들은 뷰 합성에서 사용하기 위한 추정된 깊이들 (403) 을 포함할 수도 있다. 깊이 맵들이 생성되면, 2D - 3D 컨버전 유닛 (79) 은 비디오 프레임들에 대하여 하나 이상이 2차 뷰들을 생성하기 위해 뷰 합성 프로세스 (404) 에 깊이 맵들을 적용할 수도 있다. 이 경우에, 뷰 합성 프로세스 (404) 의 출력은 합성된 우측 뷰 (405) 를 포함할 수도 있다.
- [0067] 깊이 추정 (402) 에서, 다양한 깊이 큐 (cue) 들이 이용될 수도 있다. 이러한 큐들은 오브젝트 모션 및 컬러와 같은 다른 시간 큐 또는 2D 공간 큐와 관련하여 정의된 원근법 기하 (perspective geometry) 를 포함할 수도 있다. 2 이상의 미리 캡처된 뷰들이 존재하는 경우에, 깊이 맵들은 공지된 카메라 내부 및 외부 파라미터들의 도움으로 에피폴라 (epipolar) 기하에 의해 획득될 수 있다. 깊이 추정 기술은, 예컨대 2 개의 상이한 스테레오 뷰들에서 동일한 오브젝트의 대응성을 발견함으로써 차이 정보 (예컨대, 일반적으로 오브젝트 깊이와 반비례하는 정보) 를 추정할 수도 있다. 유감스럽게도, 스테레오 매칭에 기초한 2D - 3D 컨버전은 매우 복잡할 수 있고, 집약적인 계산들을 필요로 할 수도 있다.
- [0068] 반면에, 일부 공통적으로 이용되는 비디오 캡처 디바이스들, 예컨대 카메라들, 캠코더들 및 모바일 전화기들

에 대하여 통상적으로는 단일의 뷰 만이 사용가능하며, 이는 예컨대 2 개의 상이한 카메라들에 의해 캡처되는 실세계 (real-world) 장면들로부터 충분한 정보의 부족으로 인해 2D - 3D 컨버전을 도전할만한 문제로 만든다. 그러나, 일부 깊이 큐들은 모노스코프 소스들에 여전히 사용가능하다. 비디오 프레임은 몇몇 전경 오브젝트들과 배경의 합성으로서 고려될 수도 있다. 포커싱된 카메라의 뷰 포인트로부터, 카메라는 포커싱되지 않은 영역들의 컬러 세기들이 포커싱된 영역들과 비교하여 더 많은 블러 (blur) 를 포함할 것임을 가정할 수도 있다. 더 가까운 오브젝트들은 통상적으로 멀리 떨어진 오브젝트들 보다 더 적은 블러 현상 (blurriness) 을 가지는 것으로 예상된다. 따라서, 블러 현상은 깊이 할당을 위해 중요한 큐 (cue) 로서 이용될 수도 있다.

[0069] 2D - 3D 컨버전을 위해 이용된 블러 현상은 경사도 기반의 측정들 또는 주파수 영역 분석을 포함할 수도 있다. 큰 경사도 값들을 가지는 이미지 영역들은 더 작은 양의 블러를 갖는 영역들의 일부인 경향이 있는 반면, 작은 경사도 값들을 가지는 영역들은 더 큰 양의 블러를 갖는 영역들의 일부인 경향이 있다. 저밀도의 (sparse) 깊이 맵들은 비대칭의 깊이 맵 평활화 이전에 오브젝트 에지들과 경계들에 대한 경사도 값들로부터 추정될 수도 있다. 정확하지는 않지만, 이러한 타입의 깊이 맵들은 몇몇 레벨의 "의사" 3D 경험을 제공할 수도 있다.

[0070] 그러나, 일부 비디오 프레임들에 대하여, 전술된 가정들은 참 (true) 이 아닐 수도 있다. 먼저, 카메라는 카메라와 인접한 오브젝트들 대신에, 실제로 배경에 속하는 멀리 떨어진 오브젝트들을 포커싱할 수도 있다. 추가로, 전경의 텍스처 없는 (textureless) 영역들의 블러 분석은 전경의 동종의 영역들이 종종 고주파수 성분들을 포함하지 않기 때문에 실패할 수도 있다. 따라서, 저 블러 현상이 항상 더 적은 깊이 (카메라로의 거리) 를 나타내는 것은 아니며, 고 블러 현상이 항상 더 큰 깊이에 대응하는 것은 아니다.

[0071] 모션은 모노스코프 2D 비디오에 대한 깊이 추정에서 사용될 수 있는 다른 중요한 큐이다. 레코딩 동안 카메라 위치가 고정된다면, 더 가까운 오브젝트들이 원근법 기하로 인해 멀리 떨어진 오브젝트들과 비교하여 더 크게 보이고 더 하이 모션을 가지는 것으로 예상된다. 모션 추정은 인접한 비디오 프레임들을 통한 오브젝트 움직임 추정을 하는 프로세스이다. 모션 벡터들은 통상적으로 이전 비디오 프레임의 유사한 "예측 데이터" 와 관련된 오브젝트 수평 및 수직 병진 변위 (translational displacement) 들을 설명하는 벡터로서 정의된다. 특정 장면 세팅들에 대하여, 고정된 배경이 사용가능한 경우에, 모션은 등록된 배경 장면으로부터 현재의 프레임을 제거함으로써 (subtracting) 간단히 획득될 수 있다. 그러나, 고정된 배경의 제약들로 인해, 이러한 방법들은 바람직하지 않다. 그러므로, 인접 프레임들 사이의 차이들을 발견함으로써 다른 간단하고 신속한 모션 측정이 달성될 수도 있다. 이러한 타입의 모션 측정은 이웃하는 프레임들의 공동-위치된 픽셀들의 루마 세기들의 픽셀-기반 또는 윈도우-기반 감산들에 의해 획득될 수도 있다.

[0072] 모션 추정을 위해 이용되는 다른 기술들은 블록-기반의 매칭 및 시각적 흐름을 포함한다. 블록-기반 매칭 접근 방식들은 비디오 프레임들이 먼저 몇몇 비디오 블록들로 분할되는 비디오 압축에서 이용될 수도 있다. 현재 프레임 n 에서 각각의 비디오 블록은 동일한 사이즈를 가지지만 현재 블록에 대하여 기준 프레임에서 변위를 가지는 블록들과 비교될 수도 있다. 최소 매칭 비용과 연관된 변위, 예컨대 매칭 에러의 절대 값들의 최저 합은 그 블록에서의 픽셀들 모두에 대하여 추정된 모션 값을 정의할 수도 있다. 그러나, 블록-기반 매칭과는 달리, 시각적 흐름 기술들은 경사도-기반의 세기 변화들을 통해 픽셀 변위들을 오브젝트 속도들과 관련시키며, 여기서 픽셀-레벨 모션들이 획득될 수 있다. 그러나, 시각적 흐름 기술들은 통상적으로 잡음에 민감하며, 평활한 영역들에 대하여 매우 정확하지 않다.

[0073] 블러 현상 분석과 유사하게, 모션 큐들은 강인성 문제들을 경험할 수 있다. 다시 말해서, 일부 경우에, 모션-깊이 관계의 가정은 실패할 수 있다. 예를 들어, 카메라로의 거리가 동일한 오브젝트들은 종종 서로 독립적으로 이동하고, 상이한 속도들로 이동할 수 있다. 이 경우, 추정된 모션과 깊이 값 사이에 반비례 관계가 어긋날 수 있다. 추가로, 이동하는 카메라, 회전하는 카메라, 줌-인들, 줌-아웃들 등등을 통해 장면들이 캡처될 수도 있기 때문에, 글로벌 모션을 갖는 장면들이 존재한다. 이 경우에, 예컨대 카메라가 이동하고 있거나 장면에 대하여 변화하고 있는 경우에, 먼 거리의 오브젝트들은 더 가까운 오브젝트들과 비교할 때 더 높게 추정된 모션들을 가질 수도 있다. 또한, 전경 오브젝트들은 카메라 모션에도 고정될 수도 있다. 전체 이미지가 짧은 시간 간격에 걸쳐 어떤 모션 없이 고정되어 있는 경우에, 깊이 추정을 위해 모션 큐들을 이용하는 것은 어렵게 된다.

[0074] 깊이 값 맵핑과 더 관련되거나 덜 관련된 블러 현상 및 모션 큐들과는 달리, 컬러 세기로부터 측정된 비디오 프레임 평활화는 또한 공간 및 시간적으로 일관된 깊이 맵을 생성하는데 이용될 수 있다. 깊이 맵 일관성은 합성된 가상 뷰들에서 플리커링 (flickering) 및 로컬 변형과 같은 아티팩트들을 감소시키는 데 중요한 역

할을 한다. 일반적으로, 동일하거나 유사한 컬러들을 갖는 픽셀들은 통상적으로 동일한 오브젝트에 속하지만, 급격한 세기 변화들은 통상적으로 오브젝트 경계들을 나타낸다. 깊이 필드는 부분적으로 평활하고, 깊이의 불연속성은 이미지 세기의 불연속성을 반영하는 것으로 가정될 수도 있다.

[0075] 깊이 추정을 위한 또 다른 기술은 이미지 세분화와 관련된다. 비디오 프레임들은 먼저 몇몇 오브젝트들로 세분될 수도 있다. 그 후에, 이러한 세그먼트들에는 모션 및 블러 현상과 같은 큐들에 기초하여 상이한 깊이 값들이 할당될 수 있다. 이미지 세그먼트들의 깊이 추정은 통상적으로 픽셀-기반의 추정들과 비교하여 더 일관된 깊이 맵들을 달성하지만, 계산 복잡도는 증가할 수도 있다. 일부 경우에, 이미지 세분화는 자체적으로 문제들을 발생할 수도 있다. 예를 들어, 고 레벨의 텍스처를 갖는 영역들을 포함하는 몇몇 장면들은 세분하기가 매우 어려울 수도 있다. 이미지 세분화는 또한 컬러 변화, 예컨대 동일한 오브젝트들의 루미넌스 변화들을 경험할 수도 있다. 이 경우에, 하나의 오브젝트에 대하여 몇몇 서로 다른 세그먼트들이 획득될 수도 있고, 하나의 오브젝트의 부분들은 서로다른 세그먼트들 및 서로 다른 깊이들로 분류될 수도 있다. 이 경우에, 세분화는 열악한 깊이 추정을 발생할 수 있다.

[0076] 컬러 세기 값들에 따라 하나의 이미지를 몇몇 동종의 영역들로 분할하는 대신에, 컬러 세기 값들 (예컨대, 크로마 값들) 은 또한 깊이 추정에서 직접 이용될 수도 있다. 특정 컬러 공간들에서 그 성분들로부터의 깊이 추정에 있어 한가지 가능한 장점은 간단함이다. 크로미넌스 (크로마) 성분들은 통상적으로 동일한 오브젝트들에 속하는 픽셀들에 대하여 매우 평활하다. 세분화-기반의 깊이 추정과 비교할 때, 컬러 성분들로부터 직접 생성된 깊이 맵들은 통상적으로 오브젝트 형상을 보존하며, 따라서 더 양호한 공간적 일관성을 제공할 수도 있다. 추정된 깊이 값들이 실제 깊이에 비해 부정확한 경우에도, 합성된 스테레오 쌍들은 몇몇 레벨의 원하는 3D 효과를 제공할 수 있다.

[0077] 예컨대, 2D - 3D 컨버전 유닛 (79) 에 의해 수행되는 뷰 합성 프로세스는 임의의 뷰 각도로 뷰를 생성하기 위해 고밀도로 (densely) 샘플링된 뷰들을 필요로 하는 샘플링 문제로서 간주될 수도 있다. 그러나, 실제 애플리케이션들에서, 고밀도로 샘플링된 뷰들에 의해 요구되는 저장 또는 전송 대역폭은 실현되기에 너무 클 수도 있다. 저밀도로 샘플링된 뷰들에 대한 뷰 합성 알고리즘들은 3D 랩핑 (wrapping) 에 기초할 수도 있다. 3D 랩핑에서, 깊이와 카메라 모델이 제공될 때, 기준 뷰의 픽셀 u_1 은 2D 카메라 좌표로부터 세계 좌표의 한 점으로 다시 투영될 수도 있다. 그 후에 그 포인트는 투영되는 축을 따라 목적지 뷰 (생성될 가상 뷰) 로 투영될 수도 있다. 이 경우에, 재투영된 픽셀 위치는 u_2 로 정의될 수도 있고, 2 개의 픽셀들 (u_1 및 u_2) 은 동일한 오브젝트의 상이한 투영들에 대응할 수도 있다. 이 경우에, u_1 및 u_2 는 동일한 컬러 세기들을 가지는 것으로 예상된다.

[0078] 카메라 모델은 뷰 합성에서 이용될 수도 있다. 일 예는 이른바 "핀홀" 카메라 모델이다. 핀홀 카메라 모델은 내부 및 외부 메트릭들에 의해 파라미터화될 수도 있고, 세계 좌표들로부터 카메라 이미지 평면 좌표들로의 변환을 설명하는데 이용될 수 있다. 간략함을 위해, 언급되는 좌표 시스템들 모두는 3D 세계 좌표들, 3D 카메라 좌표들 및 2D 이미지 평면 좌표들을 포함하며, 직교하는 것으로 가정될 수도 있다.

[0079] 4×4 외부 행렬 $A = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$ 은 세계 좌표들에서 카메라 센터의 위치 및 카메라 방위를 정의할 수도 있고, 여기서 R 은 3×3 회전 행렬이고, t 는 3×1 평행 이동 벡터 (translation vector) 이다. 세계 좌표들 (x_w, y_w, z_w)^T 로부터 카메라 좌표들 (x, y, z)^T 로의 변환은 하기와 같이 기재될 수 있고:

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = A \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix},$$

[0080] 여기서 위 첨자 T 는 벡터 전치 연산을 나타낸다.
 [0081]

[0082] 카메라 좌표 시스템에서, z 축은 주축으로 불릴 수도 있다. 카메라 평면은 주축과 수직하는 평면이고, 카메라 좌표들에서 $(0,0,f)$ ^T 를 통과하며, 여기서 f 는 카메라 초점거리이다. 주점은 주축이 이미지 평면과

만나는 점으로 정의된다.

[0083] 내부 파라미터는 카메라 좌표들 $(x, y, z)^T$ 로부터 이미지 평면 좌표들 $(u, v)^T$ 로의 변환을 명시한다. 핀

$$\frac{u}{x} = \frac{v}{y} = \frac{f}{z}$$

홀 카메라 모델로부터, 주점이 이미지 평면 원점 (origin) 과 일치한다면, x, y, z 은 참이며, 상기 식은 하기와 같은 동등한 표현으로 기재될 수 있다:

[0084]

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} fx \\ fy \\ z \end{pmatrix} = \mathbf{Q} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}.$$

[0085]

상기 식들로부터, 하기와 같이 세계 좌표 시스템으로부터 이미지 평면으로의 변환을 유도할 수 있다:

[0086]

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = [\mathbf{Q} | \mathbf{0}] \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = [\mathbf{Q} | \mathbf{0}] \mathbf{A} \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} = \mathbf{Q} [\mathbf{R} | \mathbf{t}] \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix}.$$

[0087]

주된 (principal) 오프셋이 존재하는 경우에, 주점은 더 이상 이미지 평면 좌표 원점이 아니지만, 이미지 평면 좌표들에서 $(p_x, p_y)^T$ 가 될 수도 있다. 내부 카메라 행렬 \mathbf{Q} 은 더 복잡할 수 있다. 주된 오프셋, x 및 y 축들을 따라 f_x 및 f_y 에 의해 정의되는 상이한 초점거리, 및 카메라 스큐 (skew) 인자를 고려할 때, 행렬 \mathbf{Q} 은 하기와 같이 표현될 수도 있다:

[0088]

$$\mathbf{Q} = \begin{pmatrix} f_x & Skew & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{pmatrix}$$

[0089]

간략함을 위해, 스테레오 비디오에서, 2 개의 카메라들이 세계 좌표들에서 동일한 방위들을 가지고, 이 카메라들 중 하나는 세계 좌표 원점에 위치하며, 즉 $\mathbf{R}_1 = \mathbf{R}_2 = \mathbf{I}$ 이고, $\mathbf{t}_1 = \mathbf{0}$ 이라고 가정할 수도 있다. x 축을 따라 2 개의 카메라들 사이에 오직 한 번의 평행 이동 (translation) 이 발생하며, 따라서 $\mathbf{t}_2 = (t_x, 0, 0)^T$ 라고 가정하자.

[0090]

만약 2 개의 카메라들이 $f_x = f_y = f$, $Skew = 0$ 및 $p_x = p_y = p$ 인 동일한 내부 파라미터들 $\mathbf{Q}_1 = \mathbf{Q}_2$ 을 갖는다면, 2 개의 픽셀들은 이미지 평면 좌표들의 u 축을 따라 서로 달라질 수 있다. 전술된 식들에 기초하여,

[0091]

$$u_1 = fx_w / z_w,$$

[0092]

$$u_2 = f(x_w + t_w) / z_w$$

[0093]

따라서, $u_2 - u_1 = ft_x / z_w$ 이며, 여기서 $u_2 - u_1$ 은 깊이 z_w 와 반비례하는, 이른바 디스패리티 (disparity) 이다. 하나의 픽셀 위치 u_1 와 깊이 z_w 가 주어질 때, 3D 랩핑은 간단히 $u_2 = u_1 + ft_x / z_w$ 의 계산식으로서 정의될 수 있고, 여기서 f 및 t_x 는 공지된 카메라 파라미터들이다. 목적지 가상 뷰 상에서 투영 위치 u_2 가 결정된 후에, 픽셀 u_2 의 컬러 세기는 기준 뷰로부터의 픽셀 u_1 과 동일한 값으로 세팅될 수 있다.

[0094]

뷰 합성에서, 2 이상의 뷰가 참조들로 고려될 수 있다. 그러나, 단 하나의 기준 뷰가 존재하는 경우에도, 2 이상의 픽셀이 동일한 이미지 픽셀 u_2 로 투영될 수 있다. 이 경우에, 참조 픽셀들 또는 최소 깊이 값

을 가지는 픽셀의 가중된 평균은 픽셀 u_2 을 합성하기 위해 이용될 수 있다. 추가로, 교합 (occlusion) 현상으로 인해 기준 뷰 또는 뷰들로부터 어떤 픽셀도 u_2 로 투영되지 않는 경우가 존재한다. 이 경우에, 홀이 존재하며 홀 필링 기술들은 이웃하는 픽셀들로부터 u_2 에 대한 세기 값들을 구성하는데 이용될 수도 있다.

[0095] 깊이 추정에서 깊이 값들의 한가지 현실적인 문제는 깊이 값들이 8 개 비트들이 이용되는 0 부터 255 까지의 깊이 값 v 범위와 같은 동적 범위 내에서 통상적으로 표현된다는 것이다. 이 경우에, 0 은 최대 깊이를 나타낼 수 있고, 255 는 가장 가까운 깊이를 나타낼 수도 있다. 본 명세서에서 사용되는 어구 "깊이 값" 은 오브젝트 깊이 z 의 이러한 8 비트 표현을 지칭할 수도 있지만, 다른 비트 사이즈들이 이용될 수도 있다. $v = 255$ 의 최대 깊이 값은 최소 깊이 z_{near} , 즉 카메라로의 최근접 거리인 깊이에 대응할 수도 있지만, 최소 깊이 값 $v = 0$ 은 최대 깊이 z_{far} , 즉 카메라로의 최대 거리인 깊이를 나타낸다. 일부 통상적인 깊이 값 맵핑들은 하기와 같다:

[0096]
$$\left(1 - \frac{v}{255}\right)(z_{far} - z_{near}) = z - z_{near}$$

[0097] 또는

[0098]
$$\frac{1}{z} - \frac{1}{z_{far}} = \frac{v}{255} \left(\frac{1}{z_{near}} - \frac{1}{z_{far}} \right).$$

[0099] 3D 비디오 포맷을 지원하기 위해, JVT (Joint Video Team) 및 MPEG (Motion Picture Experts Group) 는 다수의 표준들 및 특징들을 도입하였다. 일부 경우에, 본 개시물의 기술들은 이러한 JVT 및 MPEG 표준들 또는 특징들과 호환가능할 수도 있거나, 다른 타입의 시스템들에서 사용될 수도 있다. MPEG-2 에서, 좌측 뷰와 같은 하나의 뷰는 감소된 프레임 레이트 (예컨대, 초당 15 프레임 (fps)) 로 코딩될 수도 있고, 다른 뷰는 더 높은 프레임 레이트 (예컨대, 초당 30 프레임 (fps)) 를 요구하는 시간 확장 계층으로서 코딩될 수도 있다.

[0100] H.264/AVC 에서, 스테레오 비디오 보충 강화 정보 (SEI) 메세지는 하나의 비트스트림 내에 2 개의 뷰들이 배열되는 방식을 나타내기 위해 사용될 수 있다. 2 개의 뷰들은 교번하는 프레임들 또는 상보적 필드 쌍들 일 수도 있다. 2 개의 뷰들이 교번하는 프레임들일 때, 그들은 시간 인터리빙 모드로 정렬된다. 2 개의 뷰들이 상보적 필드 쌍들인 경우에, 2 개의 뷰들로부터의 이미지 쌍은 실제로 하나의 픽처에서 로우 (row) 인터리빙된다. 본 개시물의 기술들이 인코더에 의해 깊이 맵들을 생성하고 2차 뷰들을 합성하기 위해 수행되는 경우, 그러한 2차 뷰는 SEI 메세지들의 이용을 통해 H.264/AVC 에 따라 인코딩되고 통신될 수 있다. 다른 경우에, 깊이 맵들은 비트 스트림에서 통신될 수 있다. 다른 경우에, 인코더 보다는 디코더가 본 개시물에서 설명되는 깊이 맵 생성 기술들을 수행할 수 있다.

[0101] JVT 는 또한 시간 인터리빙 및 로우 (row) 인터리빙으로부터의 2 개의 뷰들의 지원을 더 유동적인 공간 인터리빙 모드들로 확장시킬 수 있는 공간 인터리빙 픽처 SEI 메세지를 채용하고 있다. 로우 인터리빙에 부가하여, SEI 메세지는 또한 이미지 쌍의 배열을 나란히, 위 아래로, 컬럼 인터리빙으로, 또는 체커보드 패턴으로 지원할 수도 있다. H.264/AVC 에서 상기 SEI 메세지들은 3D 입력이 2D 비디오 시퀀스인 것처럼 허용될 수 있는 경우를 지원할 수도 있다. SEI 메세지는 2 개의 뷰들이 인터리빙되어 디스플레이가 디코딩된 비디오 시퀀스를 2 개의 뷰들로 구분할 수 있는 방식을 디코더 또는 렌더러에 제시하는데 이용될 수도 있다. 이러한 경우로의 입력은 2 개의 뷰들에 의해 표현되는 3D 입력을 포함할 수도 있다.

[0102] H.264/AVC 의 확장으로서, 멀티뷰 비디오 코딩 (MVC) 은 3 이상의 뷰들의 코딩을 지원하며, 인코더가 깊이 맵들을 생성하고 뷰 합성을 수행하는 예들과 일치할 수도 있다. MVC 예측은 각각의 뷰 내의 인터-픽처 예측과 인터-뷰 예측 양자를 포함할 수도 있다. MVC 는 H.264/AVC 디코더들에 의해 디코딩될 수 있는, 이른바 기본 뷰를 포함할 수도 있으며, 2 개의 뷰들은 또한 MVC 에 의해 지원될 수 있다. MVC 의 장점은 MVC 가 3D 비디오 입력과 같은 3 이상의 뷰들이 이용되는 케이스를 지원할 수 있고, 다수의 뷰들에 의해 표시되는 이 3D 비디오를 디코딩하는 것이다. SEI 메세지들을 이용하는 H.264/AVC 에서와 같이, MVC 는 통상적으로 깊이 맵들을 이용하지 않는다.

- [0103] MPEG 는 또한 MPEG-C, 파트 3 에서 정규 비디오 스트림에 대하여 깊이 맵을 부가하기 위한 포맷을 명시한다. 이 규격은 하기에서 설명된다:
 “Text of ISO/IEC FDIS 23002-3 Representation of Auxiliary Video and Supplemental Information,” *ISO/IEC JTC 1/SC 29/WG 11, MPEG Doc, N8768, Marrakech, Morocco, January 2007.*
- [0104] MPEG-C 파트 3 에서, 이른바 보조 비디오는 깊이 맵 또는 시차 (parallax) 맵 일 수 있다. 깊이 맵을 표시할 때, MPEG-C 파트 3 는 깊이 맵의 각각의 깊이 값 및 해상도를 표시하기 위해 이용되는 비트 수와 관련하여 융통성을 제공한다. 그러나 MPEG-C 파트 3 는 깊이 맵을 생성하는 방식을 명시하지 않는다.
- [0106] MPEG 비디오 서브그룹에서, MPEG-C 파트 3와 유사한 애플리케이션 시나리오를 연구하는 3DV EE (Exploration Experiments in 3D Video Coding) 이 존재한다. 각각의 뷰에 대한 깊이 맵들을 포함하기 위해 이러한 애플리케이션들이 유용할 수도 있다. 깊이 추정 및 뷰 합성은 3DV EE 의 애플리케이션에 매우 유용할 수도 있다. 일부 경우에, 비디오 콘텐츠는 멀티-카메라 시스템에 의해 캡처될 수 있고, 깊이 맵들이 생성되어 뷰들 중 하나의 인코딩된 버전과 함께 전송될 수 있다. 이 경우에, 뷰 합성은 전송되지 않는 더 많은 뷰들을 생성하기 위해 디코딩된 디바이스에서 이용될 수도 있다. 통상의 3D 비디오 통신 시스템은 인코딩 디바이스에서 로우 데이터를 캡처하고 인코딩하며, 2D 시퀀스와 함께 전송할 깊이 맵들을 생성하며, 따라서 3D 시퀀스는 디코더 디바이스에서 생성될 수 있다.
- [0107] 포착 시스템들은 하나의 카메라 캡처 시스템들 또는 카메라 어레이들을 포함할 수도 있거나, 심지어 깊이 맵들을 생성할 수 있는 디바이스들과 연관될 수도 있다. 일반적으로, 비디오 캡처는 하기의 사항들 중 하나로 카테고리화될 수도 있다:
- [0108] 정상적으로 3D 콘텐츠를 제공하지 않는 2D 비디오 캡처.
- [0109] 스테레오스코프 (stereoscopic) 비디오를 제공하는 2 개의 카메라 시스템들.
- [0110] 멀티 뷰들을 캡처하는 카메라 어레이 시스템들.
- [0111] 하나의 뷰를 캡처하고, 깊이를 캡처하거나 합성하는 시스템들.
- [0112] 다른 시스템들은 깊이 정보를 직접 캡처하거나 3D 모델을 생성할 수도 있다.
- [0113] 3D 렌더링을 위해 사전 프로세싱이 또한 수행될 수도 있다. 3D 사전 프로세싱은 인코딩에 유익할 수 있는 잡음 추정 및 장면 검출과 같은 통상의 프로세싱 기술들을 지칭하지 않는다. 대신에, 3D 사전 프로세싱은, 3D 비디오 콘텐츠의 일부로서 코딩될 수도 있는 깊이 맵들의 생성을 지칭할 수도 있다. 이러한 3D 사전 프로세싱은 하나의 캡처된 뷰에 대한 하나의 깊이 맵 또는 몇몇 뷰들에 대한 몇몇 깊이 맵들을 생성할 수도 있다. 깊이 맵은 일 시퀀스의 각각의 비디오 프레임과 연관될 수도 있다. 인코더는 하나 이상의 뷰들 및 깊이 맵들을 인코딩할 수도 있고, 깊이 및 텍스처를 공동으로 코딩할 수 있는 H.264/AVC, MVC 또는 심지어 SVC 와 같은 비디오 코딩 표준들로 이들을 코딩할 수도 있다.
- [0114] 비디오 콘텐츠가 렌더링 디바이스에서 수신되면, 임의의 경우에, 대응하는 깊이 맵들과 함께 하나 이상의 뷰들로 디코딩되고 복원될 수 있다. 깊이 맵들의 디코딩 이후에, 뷰 합성 알고리즘은 전송되지 않는 다른 뷰들에 대한 픽셀 값들을 생성하기 위해 렌더링 디바이스에서 실행될 수도 있다. 정규 3D 디스플레이들은 2 개의 뷰들 또는 다수의 뷰들을 렌더링한다. 높은 프레임 레이트의 비디오들을 디스플레이할 수 있는 일부 2D 디스플레이들은 또한 입체 안경의 도움으로 3D 디스플레이로서 이용될 수 있다. 편광 (Polarization) 은 2 개의 뷰들을 출력으로서 제공할 수 있는 3D 디스플레이 기술의 일 예이다. 일부 디스플레이들 또는 3DTV들이 입력의 일부로서 깊이를 처리하도록 구성될 수도 있지만, 출력으로서 2 이상의 뷰들을 생성하는 "뷰 합성" 모듈이 여전히 존재할 수도 있다.
- [0115] 깊이 맵-기반의 2D - 3D 비디오 컨버전에 대하여, 문제들은 깊이 맵 정확성 및 계산 복잡도를 포함할 수도 있다. 깊이 추정의 정확성은 가상 뷰 합성에 있어 중요하다. 정확하지 않은 깊이 맵들은 합성된 가상 뷰들에서 곤란한 아티팩트들, 로컬 변형들 및 폴리커링을 발생시킬 수도 있다. 이러한 아티팩트들은 스테레오스코프 경험을 극적으로 저하시킬 수 있다. 다른 한편으로, 정확한 깊이 맵들은 통상적으로 높은 계산 복잡도로 획득된다. 유감스럽게, 일부 디바이스들에 대하여, 계산적으로 집약적인 연산들이 유용하지 않을 수도 있다. 성공적인 깊이 추정 알고리즘을 위해, 깊이 정확성과 깊이 추정 계산 복잡도 사이의 균

형이 신중하게 고려되어야 한다.

[0116] 본 개시물의 기술들은 깊이 정확성과 계산 복잡도 사이에 양호한 균형을 제공할 수도 있다. 이러한 기술들은 DIBR 기반의 뷰 합성 2D - 3D 컨버전 시스템에서 유용할 수도 있다. 본 개시물은 각 프레임에서 픽셀들의 2D 비디오 시퀀스 및 컬러 성분들의 공동으로 이용하는 모션 정보에 기초하는, 상대적으로 낮은 복잡도의 깊이 추정 알고리즘을 제공할 수도 있다.

[0117] 더욱 상세하게, 깊이 값들은 먼저 오브젝트 모션들로부터의 선형 맵핑에 의해 초기화될 수도 있다. 하이 모션 픽셀들에는 더 큰 모션 값들이 할당될 수도 있다. 로우 모션 픽셀들에 대하여, 깊이 값들은 픽셀 컬러 세기들로부터 직접 맵핑될 수 있다. 픽셀 컬러 정보로부터의 맵핑은 로우 모션 픽셀들에 대한 깊이 값들이 항상 더 하이 모션 픽셀들에 대한 깊이 값들보다 작도록 선택될 수 있다. 결과적으로, 깊이 사후 프로세싱 및 시간 방향에 따른 전과는 더 평활하고 일정한 깊이 맵을 보장하는 것을 돕도록 적용될 수 있다. 일반적으로, 본 명세서에서 설명되는 깊이 추정 기술들은 상대적으로 낮은 복잡도로 이루어질 수도 있고, 스테레오스코프 디바이스들에 대한 몰입형의 3D 경험 능력을 제공한다.

[0118] 상대적으로 낮은 복잡도의 깊이 추정 기술들은 실시간 2D - 3D 비디오 컨버전에 적용될 수도 있다. 다시 말해서, 이 기술들은 깊이 초기화를 위한 모션 및 컬러 큐들을 이용할 수도 있다. 그 후에, 연속하는 비디오 프레임들의 깊이 맵들은 시간적인 일관성을 위해 사후 프로세싱될 수 있다.

[0119] 도 5 는 본 개시물과 부합하는 깊이 맵 초기화를 위한 예시적인 프로세스를 예시한 흐름도이다. 도 5 의 프로세스는, 예컨대 도 2 및 도 3 을 참조하여 설명되는 것과 같이 인코더 또는 디코더에 의해 수행될 수 있다. 예를 들면, 프로세스는 비디오 인코더 (50) 의 깊이 맵 생성 유닛 (36) 또는 비디오 디코더 (70) 의 깊이 맵 생성 유닛 (77) 에 의해 수행될 수도 있다. 간략함을 위해, 도 5 의 프로세스는 비디오 디코더 (70) 의 깊이 맵 생성 유닛 (77) 의 관점에서 설명될 것이다.

[0120] 도 5 에 도시된 것과 같이, 깊이 맵 생성 유닛 (77) 은 이전 프레임 n-1 (502) 의 데이터에 기초하여 프레임 n (501) 에 대한 모션을 계산 (503) 함으로써 시작한다. 그 후에, 추정된 모션 값들은 프레임 n 내의 각각의 픽셀에 초기 깊이 값들을 할당하는데 이용될 수도 있다 (505). 한편, 컬러 성분은 프레임 n 의 픽셀 세기들로부터 추출될 수도 있다 (504). 픽셀 모션들로부터의 깊이 값들에 대체 기준 (replacement criterion) 이 적용될 수 있다. 특히, 모션에 기초하는 할당된 깊이 값이 임계치 이상이면 (506 에서 "예"), 할당된 깊이 값은 그러한 하이 모션 픽셀들에 대한 깊이 맵에서 이용될 수도 있다 (507). 그러나, 모션에 기초하는 할당된 깊이 값이 임계치 미만이면 (506 에서 "아니오"), 할당된 깊이 값은 컬러에 기초하는 깊이 값으로 대체될 수도 있고 (508), 대체된 깊이 값은 로우 모션 픽셀들에 대한 깊이 맵에서 이용될 수도 있다 (509). 깊이 맵 결합 (510) 은 일 프레임의 하이 모션 픽셀들에 대한 초기화된 깊이 값들과 일 프레임의 로우 모션 픽셀들에 대한 대체된 깊이 값들을 결합하는, 초기화된 깊이 맵을 형성하는 프로세스이다. 따라서, 최종 깊이 맵은 모션에 기초하여 정의된 일부 깊이 값들과 컬러에 기초하여 정의된 다른 깊이 값들을 포함할 수도 있다.

[0121] 다시 말해서, 임의의 로우 모션 (고정형) 픽셀들에 대하여, 모션에 기초하여 생성된 깊이 값들은 픽셀 컬러 성분들로부터 결정된 깊이 값들로 대체될 수도 있다 (508). 그 후에, 깊이 결합 (510) 은 2 가지 타입의 픽셀들의 깊이 값들을 전체 프레임에 대한 하나의 깊이 맵으로 통합한다. 결과적으로, 원하는 경우에, 결합된 깊이 맵은 더 공간적으로 일관된 깊이를 생성하기 위해 사후 프로세싱될 수도 있다 (511). 예컨대, 사후 프로세싱 (511) 은 깊이 값들이 공간적으로 일관되도록 보장하기 위해 필터링 또는 다른 타입의 프로세싱을 포함할 수도 있다. 도 5 의 출력 (512) 은 하기에서 더 상세히 논의되는 것과 같이 완성된 깊이 맵을 생성할 때 깊이 맵 생성 유닛 (77) 에 의해 이용될 수 있는, 초기화된 깊이 맵을 포함할 수도 있다.

[0122] 픽셀-기반의 연속하는 프레임 차이들은 간단하기 때문에 모션을 계산 (503) 하는데 이용될 수도 있다.

$i = (u, v)^T$ 는 픽셀 위치를 표시하고, 여기서 u 및 v 는 각각 이미지 평면 수평 및 수직 축들을 나타낸다고 하자. 픽셀 i 에서 프레임 n 의 컬러 세기 값 (예컨대, 루마 값) 이 $L_{n,i}$ 이면, 그 대응하는 모션 값은 픽셀 i 에서 프레임 n 과 프레임 n-1 사이에 컬러 세기 (루마 "L") 차이의 절대값인,

[0123]
$$m_{n,i} = |L_{n,i} - L_{n-1,i}|$$

[0124] 로서 정의될 수도 있다.

[0125] 컬러 세기 L 는 임의의 컬러 공간의 세기 값을 나타낸다. 세기 값 L 은 대안적인 컬러 세기 값들이 이용

될 수 있음에도 YUV 컬러 공간의 "Y" 또는 루미넌스에 대응할 수도 있다. YUV 컬러 공간에서, Y 는 루미넌스를 나타내고, U 및 V 는 크로미넌스의 차이 값들을 나타낸다. 변수들 "L", "cr" 및 "cb" 은 또한 루미넌스 "L" 와 2 개의 상이한 크로미넌스 값들 "cr" 및 "cb" 을 나타내는데 이용될 수도 있다. 크로미넌스 값 차이들 또는 가중 평균 컬러 성분 차이들은 또한 모션을 정의하는데 이용될 수 있다. 상기 식은 루마 차이들의 스칼라 값들에 기초하여 모션을 정의함으로써 간단함에 있어 장점들을 가질 수도 있다. 이 경우에, 모션은 실제 수평 및 수직 오브젝트 변위들을 나타내는 것이 아니라 적은 양의 오브젝트 변위를 반영할 수도 있다. 이는 더 큰 컬러 차이들이 통상적으로 더 큰 변위를 나타내는 반면, 매우 작은 컬러 차이들은 상대적으로 고정된 픽셀들에서 더 자주 발생하기 때문이다.

[0126] 대안적으로, 예컨대, 디코딩 프로세스 동안 압축된 비트스트림들로부터 직접 획득된 블록-레벨 모션 벡터들은 입력 소스가 압축된 비디오인 경우에 실시간 2D - 3D 비디오 컨버전 프레임워크에서 이용될 수 있다. 이 경우에, 픽셀 i 에 대하여, 디코더로부터의 모션 벡터는 $(m_{n,u}, m_{n,v})^T$ 라고 추정하자. 이 경우에, 깊이 할당을 위해 이용된 모션 값은 2-놈 (norm) 의 모션 벡터, 즉,

$$m_{n,i} = \left\| (m_{n,u}, m_{n,v})^T \right\|_2 = \sqrt{m_{n,u}^2 + m_{n,v}^2}$$

[0127] 로서 정의될 수 있다.

[0128] 블록-레벨 모션 맵들은 뷰 합성을 위한 깊이 값들로 맵핑되기 전에 픽셀-레벨 모션들로 보간될 (interpolated) 수도 있다.

[0129] 모션 값들을 획득한 후에, 모션 값 $m_{n,i}$ 으로부터 깊이 값 $d_{n,i}$ 으로의 맵핑 $f(\cdot)$ 이 정의될 수도 있다. 통상적으로 더 큰 모션이 더 인접한 오브젝트들과 관련된다는 가정에 따라, 맵핑은 구간 $[m_{\min}, m_{\max}]$ 으로부터 $[d_{\min}^f, d_{\max}^f]$ 으로의 맵핑으로 정의될 수도 있다. 따라서,

$$d_{n,i}^f = f(m_{n,i}) = \max\left(0, \min\left(1, \frac{m_{n,i} - m_{\min}}{m_{\max} - m_{\min}}\right)\right) \times (d_{\max}^f - d_{\min}^f) + d_{\min}^f$$

[0130] 이고, 상기 m_{\max} 및 m_{\min} 은 비디오 시퀀스에 대한 최대 및 최소 모션 값들을 나타내는 상수들이고, d_{\max}^f 및 d_{\min}^f 은 맵핑 이후에 최대 및 최소 깊이 값들에 대응하는 상수 값들이다.

[0131] m_{\max} , m_{\min} , d_{\max}^f 및 d_{\min}^f 의 상수 값들은 시간적으로 일관된 깊이 맵들을 획득하기 위해 하나의 비디오 시퀀스 내의 모든 프레임들에 대하여 이용될 수도 있다. 이러한 상수들의 통상의 값들은 $m_{\max} = 255$, $m_{\min} = 0$, $d_{\max}^f = 255$ 및 $d_{\min}^f = 64$ 이다. d_{\min}^f 의 값은 이 예에서 0 으로 세팅되지 않는다.

[0132] 도 5 의 단계 (510) 에 나타나는 것과 같은 사전 프로세싱은 평활화 및 중간값 필터링을 포함할 수도 있다. 중간값 필터링에서, 각각의 픽셀 i 에 대하여, 모션 값들 세트 $\{m_{n,j}\}$ 의 중간값은 최종 모션 값들로 세팅될 수도 있고, 여기서 $j \in \mathbb{W}$ 는 픽셀 i 의 이웃 픽셀들에 대응한다. 이웃 픽셀들 \mathbb{W} 은 픽셀 i 주위의 임의의 직사각형과 같은 형태들로 선택될 수도 있다. 중간값 필터링은 모션 값들의 원하지 않는 잡음을 제거하는데 이용될 수도 있다. 이러한 방식으로, 필터링은 전술된 방식으로 획득된 깊이 값들이 이웃 픽셀들의 깊이 값들보다 더 평활하고 더 일관되게 하기 위해 이용될 수도 있다.

[0133] 모션 값들을 깊이들로 맵핑할 때, 최대 모션 값은 카메라로의 가장 가까운 거리 (예컨대, 깊이 값 d_{\max}^f) 로 맵핑될 수도 있다. 제로 (0) 의 모션을 갖는 픽셀들은 카메라로부터 가장 먼 거리 (깊이 값 d_{\min}^f) 를 가지는 것으로 간주될 수도 있다. 이 경우에, 장면 내의 고정된 오브젝트들에 대하여, 오브젝트들이 캡처된 장면에서 카메라로부터 실제로 얼마나 멀어지는지는 중요하지 않다. 대신에, 고정된 오브젝트들은 최대 깊이를 가지는 것으로 간주된다. 이러한 단점을 극복하기 위해, 본 개시물의 기술들은 모션 값들로부터의 깊이 맵핑 이후에 깊이 값 대체들을 이용할 수도 있다.

[0136] 도 5 에 도시된 것과 같이, 모션 값 $d_{n,i}^f$ 이 임의의 임계치 T_m 보다 작거나 동일할 때 (506 에서 "아니오") 대체가 발생한다. T_m 은 양호한 깊이 맵 생성을 발생하는 값을 가지도록 미리 정의될 수도 있고, 다양한 비디오 코딩 시뮬레이션들을 고려한 후에 선택될 수도 있다. 조건이 만족되면 (예컨대, $d_{n,i}^f$ 가 T_m 보다 크면), $d_{n,i}^f$ 는 $d_{n,i}^g$ 로 대체될 수도 있고 (508), 여기서,

$$d_{n,i}^g = g(c_{n,i}) = \max\left(0, \min\left(T_m, \frac{c_{n,i} - c_{\min}}{c_{\max} - c_{\min}}\right) \times T_m\right).$$

[0137] 이 경우에, $d_{n,i}^g$ 는 컬러 성분들에 기초하는 깊이 값을 나타낸다. $c_{n,i}$ 는 소정의 위치에 대한 크로마 값을 나타내고, c_{\max} 및 c_{\min} 는 비디오 시퀀스에서 최대 및 최소 컬러 크로마 값들을 나타내는 상수들을 포함할 수도 있다.

[0138] 함수 $g(?)$ 는 구간 $[0, T_m]$ 에서 컬러 값들을 깊이 값으로 맵핑하는데 이용될 수도 있다. 고정된 오브젝트들에 대응하는 픽셀들에 대하여, 모션 값 $m_{n,i}$ 은 0 일 수도 있고, 깊이 값은 $d_{n,i}^f = d_{\min}^f$ 일 수도 있다. 임계치 T_m 는 일부 예들에서 d_{\min}^f 으로 세팅될 수도 있다. 그러나, 약간의 조명 변화들로 인해, 고정된 픽셀의 컬러 세기는 프레임 $n-1$ 으로부터 프레임 n 으로 변화할 수도 있다. 이러한 이유로, 일부 예들은 임계치 T_m 에 대하여 약간 더 큰 값을 구현할 수도 있다.

[0139] 본 개시물에 따라, 모션 $d_{n,i}^f \in [d_{\min}^f, T_m]$ 에 기초한 깊이 값들을 만족하는 모든 픽셀들은 범위 $[0, T_m]$ 에서 컬러 $d_{n,i}^g$ 에 기초한 깊이 값들로 대체될 수도 있다. 따라서, 대체는 일부 상대적으로 고정된 픽셀들에 대하여 상이한 깊이 값들을 할당할 수도 있다. 일부 컬러 정보는 깊이 맵의 초기화로서 이용될 수도 있다. 본 개시물은 컬러에 기초한 깊이 값들에 대한 YUV 컬러 공간의 크로미넌스 값들 Cr 의 이용을 고려하지만, 대안적인 솔루션들은 RGB 컬러 공간의 R 성분, YUV 공간의 Cb 성분 및 L*a*b 컬러 공간의 a 또는 b 성분들을 포함할 수도 있다.

[0140] 본 개시물에 따라, 모션에 기초한 초기 깊이 값들을 할당하는 것은 이전 비디오 유닛의 공동-위치된 픽셀들의 루마 값들에 대하여 비디오 유닛의 픽셀들의 루마 값들에 기초하여 초기 깊이 값들을 할당하는 것을 포함할 수도 있고, 컬러에 기초하여 조정된 깊이 값들을 할당하는 것은 비디오 유닛의 픽셀들의 크로마 값들에 적어도 부분적으로 기초하여 조정된 깊이 값들을 할당하는 것을 포함한다. 앞서 식들과 부합하여, 조정된 깊이 값들을 할당하는 것은 허용되는 구간의 크로마 값들의 내의 크로마 값들을 제로와 임계치 사이의 깊이 값들로 맵핑하는 것을 포함할 수도 있다.

[0141] 본 개시물의 기술들은 컬러 세기들의 평활화를 이용할 수도 있고, 고정형 픽셀들에 대하여 깊이 값들을 할당하기 위해 이를 이용할 수도 있다. 본 명세서에 정의된 선형 맵핑들이 고정형 픽셀들에 대한 실제 깊이 정보를 나타내는 것은 아닐수도 있지만, 맵핑들은 원하는 3D 효과를 달성하는 일부 "지각과 관련된(perceptual)" 깊이를 제공할 수도 있다. 컬러 세기들의 깊이 값 범위 $[0, T_m]$ 로의 선형 맵핑들은 그 픽셀들의 깊이 값들이 항상, 상당한 모션과 연관된 픽셀들의 깊이 값들보다 작을 것을 보장한다.

[0142] 컬러 성분 추출에 관한 다른 실질적인 구현 문제는 컬러 성분 추출이 예측 유닛 (35) 또는 예측 유닛 (75) 의 보간 절차와 통합될 수도 있다는 점이다. 비디오 애플리케이션들에서, YUV 4:2:0 의 순차적인 포맷이 통상적으로 이용된다. YUV 4:2:0 의 순차적인 포맷에 대하여, 비디오 데이터는 $n_1 \times n_2$ 의 사이즈를 갖는 Y

성분을 포함할 수도 있고, $\frac{n_1}{2} \times \frac{n_2}{2}$ 의 사이즈를 갖는 Cb 및 Cr 성분들, 즉 크로미넌스 성분들은 수평 및 수직 양자의 방향들을 따라 2 의 인자로 서브 샘플링된다. 본 개시물의 기술들이 컬러 추출에서 크로미넌스 Cr 를 이용할 수도 있기 때문에, $\frac{n_1}{2} \times \frac{n_2}{2}$ Cr 성분은 깊이 추정을 위해 각각의 프레임에 대하여 $n_1 \times n_2$ 의 분

해능으로 업샘플링되어야 할 수도 있다.

[0144] 깊이 맵 초기화는 모션 정보와 컬러 정보 양자를 추정된 후에 획득될 수도 있다. 프레임 n 에서 픽셀 i 에 대한 깊이 값은,

$$d_{n,i} = \begin{cases} \text{만약 } d_{n,i}^f > T_m \text{ 이면, } d_{n,i}^f \\ \text{그렇지 않으면, } d_{n,i}^g \end{cases}$$

[0145] 로서 표현될 수도 있다.

[0147] 더 공간적으로 평활하고 일관된 깊이 맵들에 대하여, 깊이 값 $d_{n,i}$ 의 사후 프로세싱 (511) 이 수행될 수도 있다. 전술된 것과 같은 중간값 필터링 및 블러링 (blurring) 필터링은 사후 프로세싱에 적용될 수도 있다. 가우시안 평활화는 특정 윈도우 W 내에 적용될 수도 있다. 깊이 값 $d_{n,i}$ 의 사후 프로세싱 동안 가우시안 필터링을 위해 이용될 수도 있는 1D 가우시안 커널 (Gaussian kernel) 의 일 예는:

$$g(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{x^2}{2\sigma^2}\right\}$$

[0149] 에 의해 제공될 수도 있고, 여기서 x 는 입력 값이고, g(x) 는 가우시안 필터 함수 값이고, σ^2 는 깊이 값들의 분산이다. 이러한 가우시안 필터링은 각각 u 및 v 방향들에서 서로 다른 분산들 σ_u^2 및 σ_v^2 과는 별개로 u 및 v 방향들에서 이미지 평면에 적용될 수도 있다.

[0150] 도 6 은 본 개시물의 하나 이상의 기술들과 부합하는 완성된 깊이 맵을 생성하는 프로세스를 예시한 흐름도이다. 도 6 의 프로세스는 인코더 또는 디코더에 의해 수행될 수 있다. 예를 들면, 상기 프로세스는 비디오 인코더 (50) 의 깊이 맵 생성 유닛 (36) 또는 비디오 인코더 (70) 의 깊이 맵 생성 유닛 (77) 에 의해 수행될 수도 있다. 간략함을 위해, 도 6 의 프로세스는 비디오 디코더 (70) 의 깊이 맵 생성 유닛 (77) 의 관점에서 설명될 것이다.

[0151] 도 6 에 도시된 것과 같이, 깊이 맵 생성 유닛 (77) 은 2D 좌측 뷰 (601) 에서 시작한다. 도 6 의 기술은 결국 대응하는 우측 뷰를 생성하기 위해 이용될 수 있는 깊이 맵을 정의할 수도 있다. 물론, 초기 뷰는 대안적으로 우측 뷰일수 있고, 이 때, 깊이 맵은 좌측 뷰를 생성하는데 이용된다.

[0152] 깊이 초기화 (602) 는 궁극적으로 초기화된 깊이 맵 (603) 을 생성하는 도 5 의 프로세스와 대응할 수도 있다. 그러나, 도 6 에 따라, 깊이 맵 생성 유닛 (77) 은 궁극적으로 완성된 깊이 맵을 생성하기 위해 초기화된 깊이 맵에 대하여 추가의 기술들을 수행할 수도 있다. 도 6 에 도시된 것과 같이, 깊이 맵 생성 유닛 (77) 은 현재 비디오 프레임이 새로운 장면에 대응하는지 여부를 결정하기 위해 (605), 장면 검출 (604) 을 수행할 수도 있다. 예컨대 현재의 프레임이 장면 변화에 대응하는지 여부를 추정하기 위해 현재 프레임과 이전 프레임 사이에 픽셀 차이들을 정량화함으로써, 이러한 결정들을 수행하기 위해 다양한 기술들이 적용될 수도 있다.

[0153] 현재 프레임이 장면 변화에 대응한다면 (605 에서 "예"), 가중치 값 w 은 제로 (0) 로 세팅될 수도 있다 (607). 이는 깊이 가중 평균 (610) 에서의 임의의 가중치가 이전 프레임에 제공되지 않을 것을 보장한다. 이 경우에 (새로운 장면이 검출되면), 깊이 맵은 가중된 것으로 정의될 수도 있지만 (610), w=0 이기 때문에, 깊이 맵은 도 5 의 프로세스에서 생성된 초기화된 깊이 맵에 대응할 것이다. 사후 프로세싱이 수행될 수도 있고 (611), 깊이 맵 생성 유닛 (77) 은 2차 뷰 (예컨대, 도 6 의 프로세스로의 입력인, 입력 좌측 뷰에 대응하는 우측 뷰) 를 생성하기 위한 뷰 합성 프로세스에서 2D - 3D 컨버전 유닛 (79) 에 의해 이용될 수 있는 완성된 깊이 맵의 형태로 추정된 깊이 (612) 를 출력할 수도 있다.

[0154] 현재 프레임이 장면 변화에 대응하지 않으면 (605 에서 "아니오"), 깊이 맵 생성 유닛 (77) 은 프레임이 로우 모션 프레임인지 레귤러 모션 프레임인지 여부를 결정할 수도 있다 (608). 레귤러 프레임은 또한 로우 모션 프레임에 대하여 하이 모션이기 때문에 하이 모션 프레임으로 불릴 수도 있다. 따라서, 로우 모션 및 하이 모션은 전체 프레임의 루마 값들과 이전 프레임의 루마 값들의 상대적인 차이에 기초하여 정의될 수 있는 상대적인 용어들일 수도 있다. 그러나, 프레임이 로우 모션 프레임인지 레귤러 (하이 모션) 프레임인

지 여부의 결정은 다른 방식들로 수행될 수 있다.

[0155] 현재 프레임이 로우 모션 프레임이 아니면 (608 에서 "아니오"), 가중된 깊이 맵 평균이 이용되어, 완성된 깊이 맵은 현재 프레임의 초기화된 깊이 맵과 하나 이상의 선행하는 프레임들의 초기화된 깊이 맵의 가중된 평균일 수도 있다. 가중 인자 w 는 가중된 평균을 증진시키도록 정의될 수도 있다. 가중 인자 w 는 0.5 일 수도 있거나, 현재 프레임의 초기화된 깊이 맵을 이전 프레임에 비하여 더 심하게 가중하기 위해 더 낮은 값이 할당될 수도 있다.

[0156] 현재 프레임이 로우 모션 프레임이면 (608 에서 "예"), 깊이 최대치가 이용될 수도 있고 (609), 깊이 맵의 각각의 깊이 값은 현재 프레임의 초기화된 깊이 맵과 이전 프레임의 초기화된 깊이 맵 (또는 이전 프레임의 완성된 깊이 맵) 의 대응하는 깊이 값들의 최대치로서 정의된다. 깊이 맵 버퍼 (613) 는 이전 깊이 맵들을 저장하기 위해 깊이 맵 생성 유닛 (77) 에 포함될 수도 있다.

[0157] 다시 말해서, 초기화된 깊이 값들은 비디오 모션 및 컬러 정보로부터 추정될 수도 있다. 그러나, 전체 시퀀스의 초기화된 깊이 맵들은 시간 방향을 따라 매우 일정하지 않을 수도 있다. 더 양호한 합성된 뷰 품질을 위해, 깊이 맵 일관성은 중요한 역할을 담당할 수도 있다. 도 6 의 기술들은 이러한 깊이 맵 일관성을 보장하는 것을 도울 수도 있다.

[0158] 도 6 에서, 깊이 맵은 2D의 원래의 비디오 (예컨대, 좌측 뷰 (601)) 의 프레임 n 에 대하여 초기화된다 (602). 한편, 장면 검출 (604) 은 현재 프레임 n 이 새로운 장면에서 시작하는지 여부를 검사하도록 적용될 수 있다. 새로운 장면이 검출되면 (605 에서 "예"), 초기화된 깊이 맵은 사후 프로세싱 (611) 및 뷰 합성 (612) 을 위해 전송된다. 프레임 n 이 새로운 장면을 시작하지 않으면, 프레임 분류 단계가 적용된다. 프레임 분류 절차는 프레임을 로우 모션 프레임 또는 레귤러 프레임으로 분류한다.

[0159] 프레임이 로우 모션 프레임이면 (608 에서 "예"), 프레임 n 에 대한 깊이 맵 d_n 은 프레임 n 및 프레임 $n-1$ 의 각각의 개별 픽셀 i 에 대하여 $d_{n,i}$ 및 $d_{n-1,i}$ 의 최대값으로 세팅될 수도 있다. 다시 말해서,

[0160]
$$d_{n,i} = \max(d_{n-1,i}, d_{n,i})$$

[0161] 이다.

[0162] 그러나, 현재 프레임이 로우 모션 프레임이 아니면 (608 에서 "아니오"), 프레임 n 에 대한 깊이 맵 d_n 은 임의의 미리 정의된 가중 인자 " w " 에 의해 가중된, 이전 프레임의 깊이 맵 d_n 과 깊이 맵 d_{n-1} 의 가중된 평균일 수도 있다. 다시 말해서, 현재 프레임이 로우 모션 프레임이 아니면,

[0163]
$$d_{n,i} = w*d_{n-1,i}+(1-w)* d_{n,i}$$

[0164] 이다.

[0165] 사후 프로세싱 (611) 이후에 최종 깊이 맵들은 2D - 3D 컨버전 유닛에 의한 뷰 합성에서 사용하기 위한 깊이 맵 생성 유닛 (77) 으로부터 출력될 수도 있다. 추가로, 프레임 n 에 대한 깊이 맵 d_n 의 카피는 이후의 프레임들의 깊이 추정을 위해 깊이 맵 버퍼 (613) 에 저장될 수도 있다. 도 6 의 흐름도에 도시된 깊이 맵 버퍼 (613) 는 깊이 맵 생성 유닛 (77) 내에 포함되거나 깊이 맵 생성 유닛 (77) 에 의해 액세스될 수 있는 구조적인 하드웨어 메모리 엘리먼트일 수도 있다.

[0166] 장면 검출 기술들은 종종 비디오 트리밍, 비디오 트래킹, 및 장면 분류와 같은 비디오 분석 애플리케이션들에서 이용된다. 인접하는 프레임들의 유사성들이 추정되어 장면 변화 검출 (604) 을 위한 기준들로서 이용될 수 있다. 본 개시물의 기술들은 이러한 접근 방식의 간단함으로 인해 장면 변화들을 검출하기 위해 연속하는 프레임들의 평균 컬러 세기 차이를 이용할 수도 있다. 이 경우에, 2 개의 프레임들 사이의 차이는:

[0167]
$$diff_n = \frac{1}{n_1 \times n_2} \sum_i |L_{n,i} - L_{n-1,i}|$$

[0168] 로서 정의될 수도 있고, 상기 n_1 및 n_2 은 각각 이미지 수평축 및 이미지 수직축을 따른 픽셀들의 수이고, $L_{n,i}$

및 $L_{n-1,i}$ 는 2 개의 연속하는 프레임에 대한 픽셀 위치 i 에서의 루마 값들이다. 만약 \overline{diff}_n 이 임계치 T_s 보다 크면, 프레임 n 은 새로운 장면의 시작으로 간주될 수도 있다.

[0169] 본 개시물에 따른 장면 검출의 목표는, 시간적으로 일관된 깊이 맵들의 생성을 돕는 것을 수도 있다. 동일한 장면에 속하는 연속하는 깊이 맵들은 매우 유사한 깊이 값들을 가지는 경향이 있다. 그러나, 각각의 프레임에 대하여 독립적으로 획득되는 깊이 맵들은 시간 방향을 따라 플리커링 (flickering) 문제가 있을 수도 있다. 이전 프레임 깊이 맵 d_{n-1} 은 시간적인 일관성을 개선하기 위해 다음 프레임 깊이 맵 d_n 으로 전파될 수도 있다.

[0170] 연속하는 깊이 맵들 d_n 및 d_{n-1} 의 가중된 평균에 의해 하나의 솔루션이 획득될 수도 있다. 그러나, 매우 느린 모션들을 갖는 장면들에 대하여, 매우 적은 픽셀들에 깊이 값들 d_n^f 이 할당될 수도 있다. 이 경우에, 프레임 n 에 대하여 더 양호한 최종 깊이 맵을 생성하기 위해 깊이 최대치가 이용될 수도 있다. 추가로, 장면 검출과 유사한 방식으로, 본 개시물의 기술들은 로우 모션 프레임 분류를 이용할 수도 있다. 로우 모션 프레임에 대한 기준은 평균 프레임 차이 \overline{diff}_n 가 임계치 T_L 미만인지 여부일 수도 있고, 여기서 T_L 은 프레임이 새로운 장면이 아님을 나타내도록 T_s 보다 작지만, 정규 모션 또는 상대적으로 빠른 모션을 나타내는 충분한 차이들을 갖는다.

[0171] 이전 프레임 d_{n-1} 의 깊이 맵 일부 또는 전부를 현재 프레임 d_n 의 깊이 맵으로 전파한 후에, 더 평활한 깊이 맵들을 생성하기 위해 사후 프로세싱 (611) 이 이용될 수도 있다. 사후 프로세싱 이후, 추정된 깊이 맵은 그 후에 2D - 3D 컨버전 유닛 (79) 에 의해 수행될 수도 있는 뷰 합성에서 이용하기 위해 대기할 수도 있다. 동시에, 깊이 맵의 카피는 깊이 맵 버퍼 (613) 에 저장될 수도 있고, 깊이 맵 버퍼 (613) 는 이후 프레임들에 대한 깊이 맵들을 추정하는데 이용하기 위해 깊이 맵 생성 유닛 (77) 내에 있거나 깊이 맵 생성 유닛 (77) 에 의해 액세스될 수 있다. 버퍼 사이즈는 구현에 의존하여 하나의 이전 깊이 맵 또는 몇몇 이전 깊이 맵들을 수용하도록 설계될 수도 있다.

[0172] 도 7 은 깊이 맵 초기화 및 완성된 깊이 맵의 생성을 예시하는 흐름도이다. 도 7 의 프로세스는 인코더 또는 디코더에 의해 수행될 수 있다. 예를 들어, 프로세스는 비디오 인코더 (50) 의 깊이 맵 생성 유닛 (36) 에 의해 또는 비디오 디코더 (70) 의 깊이 맵 생성 유닛 (77) 에 의해 수행될 수 있다. 간단함을 위해, 도 7 의 프로세스는 비디오 디코더 (70) 의 깊이 맵 생성 유닛 (77) 의 관점에서 설명될 것이다.

[0173] 도 7 에 도시된 것과 같이, 깊이 맵 생성 유닛 (77) 은 비디오 시퀀스의 이전 비디오 유닛의 대응하는 픽셀들과 관련된 픽셀들과 연관된 모션에 기초하여 초기 깊이 값들을 비디오 유닛 (예컨대, 비디오 프레임 또는 슬라이스) 의 픽셀들에 할당하고 (701), 초기 깊이 값들이 임계치를 만족하는지 여부를 확인한다 (702). 깊이 맵 생성 유닛 (77) 은 초기 깊이 값들이 임계치를 만족하는 비디오 유닛의 하나 이상의 픽셀들에 조정된 깊이 값들을 할당하고 (703), 여기서 조정된 깊이 값들은 픽셀들과 연관된 컬러에 기초한다. 이 프로세스는 할당된 모든 초기 깊이 값에 대하여 반복된다 (704).

[0174] 모든 깊이 값들을 고려하고 임계치를 만족하지 않는 깊이 값들을 조정한 후에, 깊이 맵 생성 유닛 (77) 은 비디오 유닛에 대한 초기화된 깊이 맵을 생성하고 (705), 여기서 초기화된 깊이 맵은 픽셀들의 제 1 서브세트에 대한 초기 깊이 값들 및 픽셀들의 제 2 서브세트에 대한 조정된 깊이 값들을 포함한다. 전술된 것과 같이, 모션에 기초하여 초기 깊이 값들을 할당하는 것은 이전 비디오 유닛의 공동-위치된 픽셀들의 루마 값들에 대한 비디오 유닛의 픽셀들의 루마 값들에 기초하여 초기 깊이 값들을 할당하는 것을 포함할 수도 있다. 또한, 컬러에 기초하여 조정된 깊이 값들을 할당하는 것은 비디오 유닛의 픽셀들의 크로마 값들에 적어도 부분적으로 기초하여 조정된 깊이 값들을 할당하는 것을 포함할 수도 있다.

[0175] 예를 들어, 모션에 기초하여 초기 깊이 값들을 할당하는 것은 하기의 식에 따라 초기 깊이 값들을 할당하는 것을 포함하며:

[0176]
$$m_{n,i} = |L_{n,i} - L_{n-1,i}|$$

[0177] 상기 $m_{n,i}$ 는 초기 깊이 값을 나타내고, $L_{n,i}$ 는 비디오 유닛 n 에서 픽셀 i 에 대한 루마 값을 나타내며, $L_{n-1,i}$ 는 이전 비디오 유닛 $n-1$ 에서 픽셀 i 에 대한 대응하는 루마 값을 나타낸다. 추가로, 컬러에 기초하여

조정된 깊이 값들을 할당하는 것은 하기의 식에 따라 조정된 깊이 값들을 할당하는 것을 포함할 수도 있고:

$$d_{n,i}^g = g(c_{n,i}) = \max\left(0, \min\left(T_m, \frac{c_{n,i} - c_{\min}}{c_{\max} - c_{\min}} \times T_m\right)\right)$$

[0178]

상기 $d_{n,i}^g$ 는 조정된 깊이 값을 나타내고, $g(c_{n,i})$ 는 크로마 값들에 대한 함수 g 를 나타내고, T_m 는 임계치 이고, $C_{n,i}$ 는 비디오 유닛 n 에서 소정의 픽셀에 대한 크로마 값을 나타내며, C_{\max} 는 허용되는 최대 크로마 값을 나타내고, C_{\min} 는 허용되는 최소 크로마 값을 나타낸다.

[0179]

다음에, 깊이 맵 생성 유닛 (77) 은 비디오 유닛이 이전 비디오 유닛에 대한 장면 변화에 대응하는지 여부를 결정한다 (706). 만약 그렇다면 (706에서 "예"), 깊이 맵 생성 유닛 (77) 은 초기화된 깊이 맵을 비디오 유닛에 대한 최종 깊이 맵으로서 선택한다. 그러나, 비디오 유닛이 이전 비디오 유닛에 대한 장면 변화에 대응하지 않으면, 깊이 맵 생성 유닛 (77) 은 비디오 유닛이 이전 비디오 유닛에 낮은 레벨의 모션을 나타내는지 여부를 결정한다 (708). 비디오 유닛이 낮은 레벨의 모션을 나타내지 않는다면 (708 에서 "아니오"), 깊이 맵 생성 유닛 (77) 은 그 비디오 유닛에 대한 초기화된 깊이 맵의 깊이 값들과 이전 비디오 유닛에 대한 이전의 초기화된 깊이 맵의 깊이 값들의 가중된 평균들에 기반하여 최종 깊이 맵을 정의한다 (709).

[0180]

이 경우에, 비디오 유닛은 하이 모션 또는 정규 모션을 나타낼 수도 있다. 대안적으로, 비디오 유닛이로우 레벨의 모션을 나타내면 (708 에서 "예"), 깊이 맵 생성 유닛 (77) 은 그 비디오 유닛에 대한 초기화된 깊이 맵의 깊이 값들과 이전 비디오 유닛에 대한 이전의 초기화된 깊이 맵의 깊이 값들의 최대치들에 기반하여 최종 깊이 맵을 정의한다 (710).

[0181]

전술된 것과 같이, 본 개시물의 기술들은 디코더 또는 인코더에 의해 수행될 수도 있다. 기술들이 비디오 디코더에 의해 수행되면, 기술들은 추가로 비디오 유닛에 대한 2차 뷰를 생성하기 위해 최종 깊이 맵에 기초하여 뷰 합성을 수행하는 것을 포함할 수도 있다. 뷰 합성은, 예컨대 비디오 디코더 (70) 의 2D - 3D 컨버전 유닛 (79) 에 의해 수행될 수도 있다. 뷰 합성 프로세스는 원래의 뷰 및 원래의 뷰의 깊이 맵에 기초한 2차 뷰를 생성하는 프로세스를 포함할 수도 있다.

[0182]

본 개시물의 기술들이 비디오 인코더에 의해 수행된다면, 기술들은 추가로 최종 깊이 맵을 비디오 유니과 함께 통신하는 것을 포함할 수도 있다. 그러나, 일부 경우에, 뷰 합성은 비디오 유닛에 대한 2차 뷰를 생성하기 위해 최종 깊이 맵에 기초하여 인코더에 의해 수행될 수도 있고, 이 경우에 비디오 인코더와 연관된 디바이스는 비디오 유닛의 일부로서 2차 뷰를 통신할 수도 있다.

[0183]

이 기술들은 최종 깊이 맵을 생성하기 위한 추가의 프로세싱 이전에 깊이 맵 초기화를 포함하는 것으로 설명되고 있다. 그러나, 깊이 맵 초기화 기술들은 추가의 프로세싱 없이 이용될 수도 있고, 초기화된 깊이 맵은 일부 예들에서 임의의 추가의 프로세싱 없이 최종 깊이 맵으로서 정의될 수 있다.

[0184]

본 개시물의 기술들은 모바일 폰과 같은 무선 통신 디바이스 핸드셋, 집적 회로 (IC) 또는 IC들의 세트 (즉, 칩셋) 을 포함하는 폭넓게 다양한 디바이스들 또는 장치들로 구현될 수도 있다. 임의의 컴포넌트들, 모듈들 또는 유닛들은 기능적 양태들을 강조하기 위해 제공된 것으로 설명되었고, 상이한 하드웨어 유닛들에 의한 구현을 반드시 필요로 하지는 않는다. 본 명세서에서 설명된 기술들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합에서 구현될 수도 있다. 모듈들, 유닛들, 컴포넌트들로서 설명되는 임의의 특징들은 통합된 로직 디바이스에서 함께, 또는 별개이지만 공동사용가능한 로직 디바이스들로서 별도로 구현될 수도 있다. 일부 경우에, 다양한 특징들은 집적 회로 칩 또는 칩셋과 같은 집적 회로 디바이스로서 구현될 수도 있다.

[0185]

소프트웨어로 구현되는 경우, 그 기술들은 프로세서에서 실행될 때에 상술된 방법들 중 하나 이상을 수행하는 명령들을 포함하는 컴퓨터 판독가능 매체에 의해 적어도 부분적으로 실현될 수도 있다. 컴퓨터 판독가능 매체는 물리적 구조물인 컴퓨터 판독가능 저장 매체를 포함할 수도 있고, 패키징 재료들을 포함할 수도 있는 컴퓨터 프로그램 제품의 일부를 형성할 수도 있다. 컴퓨터 판독가능 저장 매체는 동기식 동적 랜덤 액세스 메모리 (SDRAM : synchronous dynamic random access memory) 와 같은 랜덤 액세스 메모리 (RAM : random access memory), 판독 전용 메모리 (ROM : read-only memory), 비휘발성 랜덤 액세스 메모리 (NVRAM : non-volatile random access memory), 전기적 소거가능 프로그램가능 판독전용 메모리 (EEPROM : electrically erasable programmable read-only memory), 플래시 메모리 (FLASH memory), 자기 또는 광학 데이터 저장 매체들을 포함할 수도 있다. 기술들은 추가로 또는 대안적으로, 명령들 또는 데이터 구조들의 형식으로 코드들

운반하거나 통신하고, 컴퓨터에 의해 액세스되고, 판독되고, 및/또는 실행될 수도 있는 컴퓨터 판독가능 통신 매체에 의해 적어도 부분적으로 실현될 수도 있다.

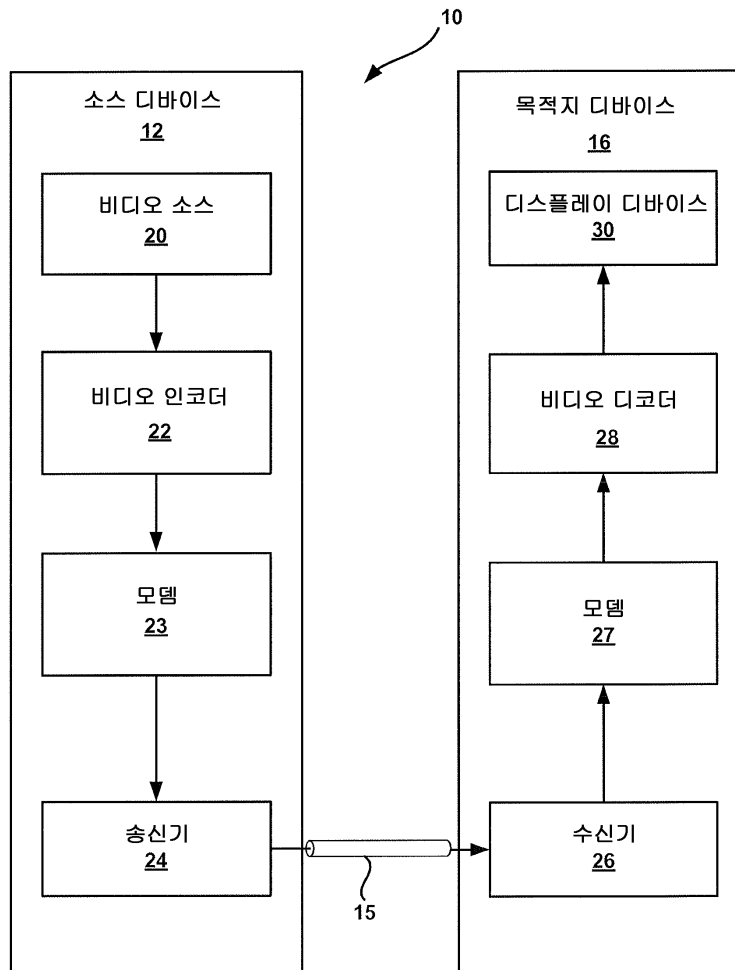
[0186] 명령들은 하나 이상의 디지털 신호 프로세서 (DSP) 들, 범용 마이크로프로세서들, 주문형 집적 회로 (ASIC), 필드 프로그램가능 게이트 어레이 (FPGA) 들, 또는 다른 동등한 통합된 또는 개별적 로직 회로와 같은 하나 이상의 프로세서들에 의해 실행될 수도 있다. 따라서, 여기에 사용되는 "프로세서" 라는 용어는 여기에 기술된 기술들의 구현에 적합한 임의의 상기 구조 또는 임의의 다른 구조를 지칭할 수도 있다. 또한, 일부 양태들에서는, 여기에 기술된 기능성이 인코딩 및 디코딩을 위해 구성된 전용 소프트웨어 모듈들 또는 하드웨어 모듈들 내에 제공될 수도 있거나, 조합된 비디오 코덱 내에 포함될 수도 있다. 또한, 그 기술들은 하나 이상의 회로들 또는 로직 엘리먼트들에서 완전히 구현될 수 있다.

[0187] 또한, 본 개시물은, 본 개시물에 기술된 기술들 중 하나 이상을 구현하기 위한 회로를 포함하는 다양한 집적 회로 디바이스들 중 임의의 집적 회로 디바이스를 고려한다. 이러한 회로가 소위 칩셋 내의 단일의 집적 회로 칩 또는 다수의 상호동작가능한 집적 회로 칩들로 제공될 수도 있다. 이러한 집적 회로 디바이스들은 그 일부가 포함될 수도 있는 다양한 애플리케이션들에서 이용될 수도 있고, 모바일 전화기 핸드셋들과 같은 무선 통신 디바이스들에서 이용할 수도 있다.

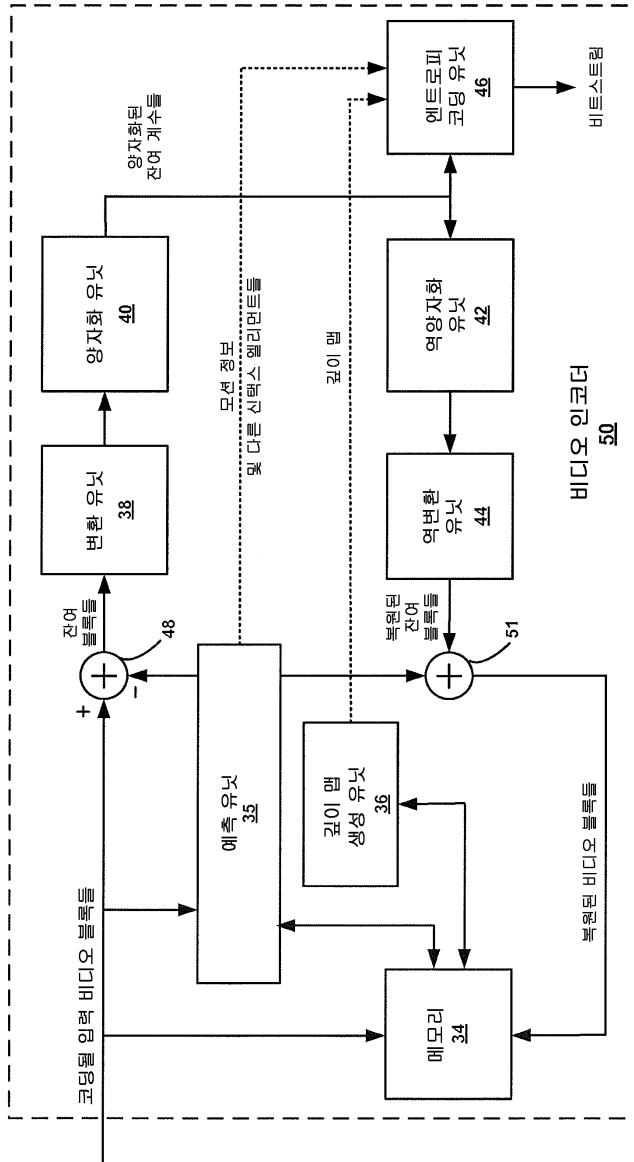
[0188] 기술들의 다양한 양태들이 기술되었다. 이들 양태들 및 다른 양태들은 하기의 청구항들의 범위 내에 있다.

도면

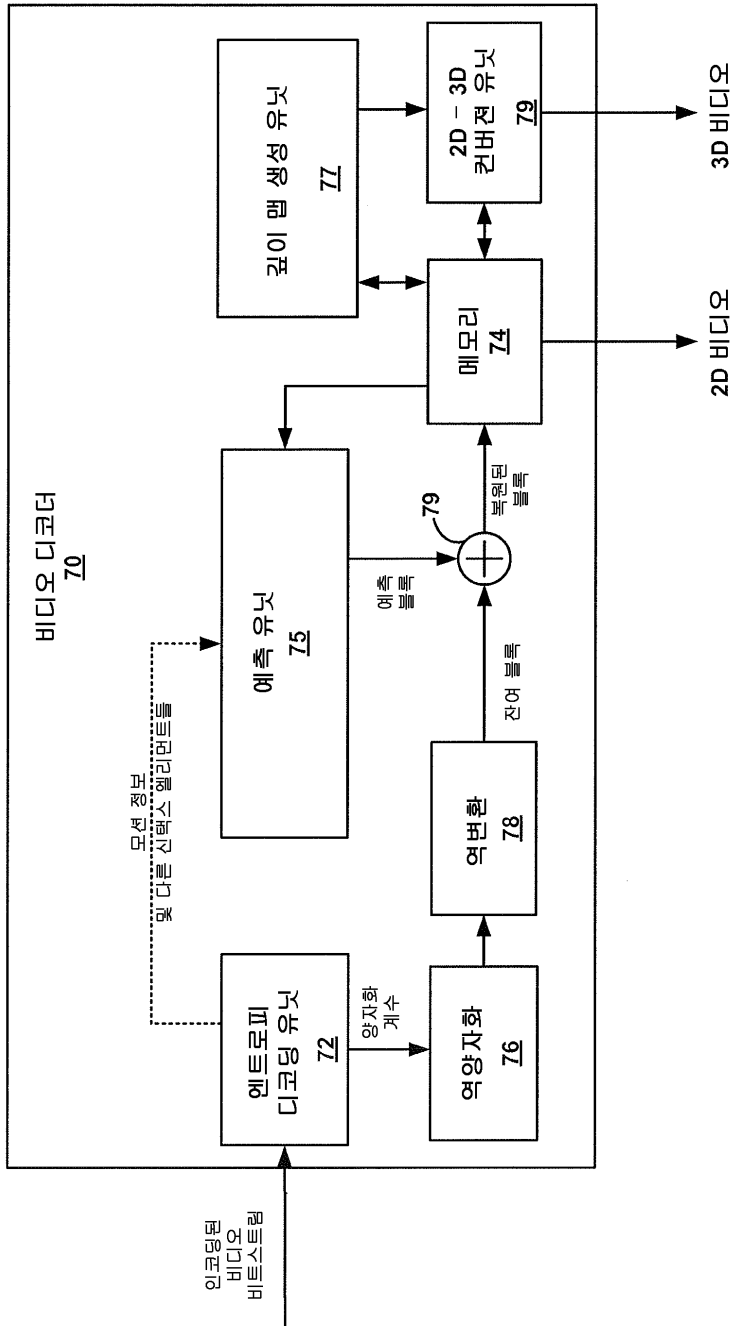
도면1



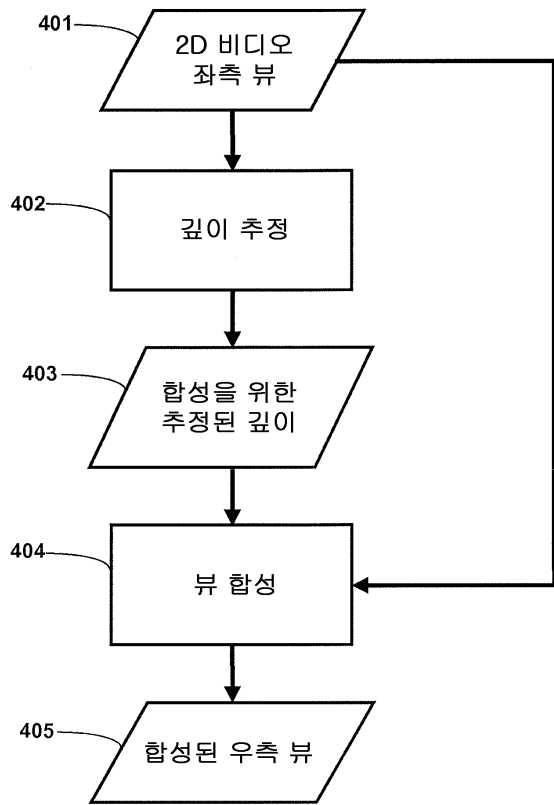
도면2



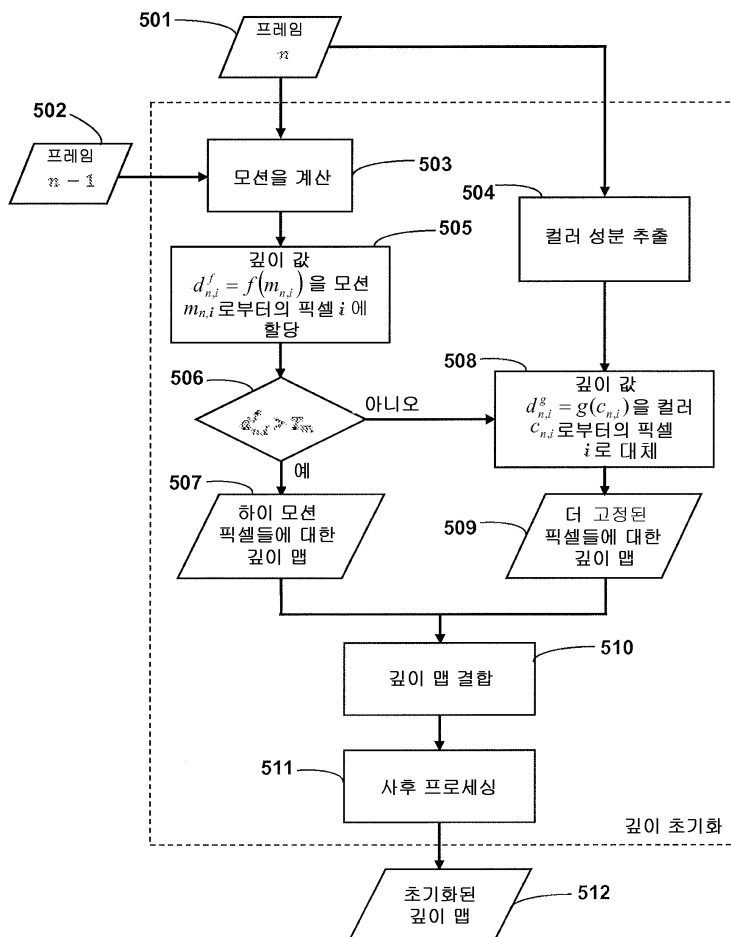
도면3



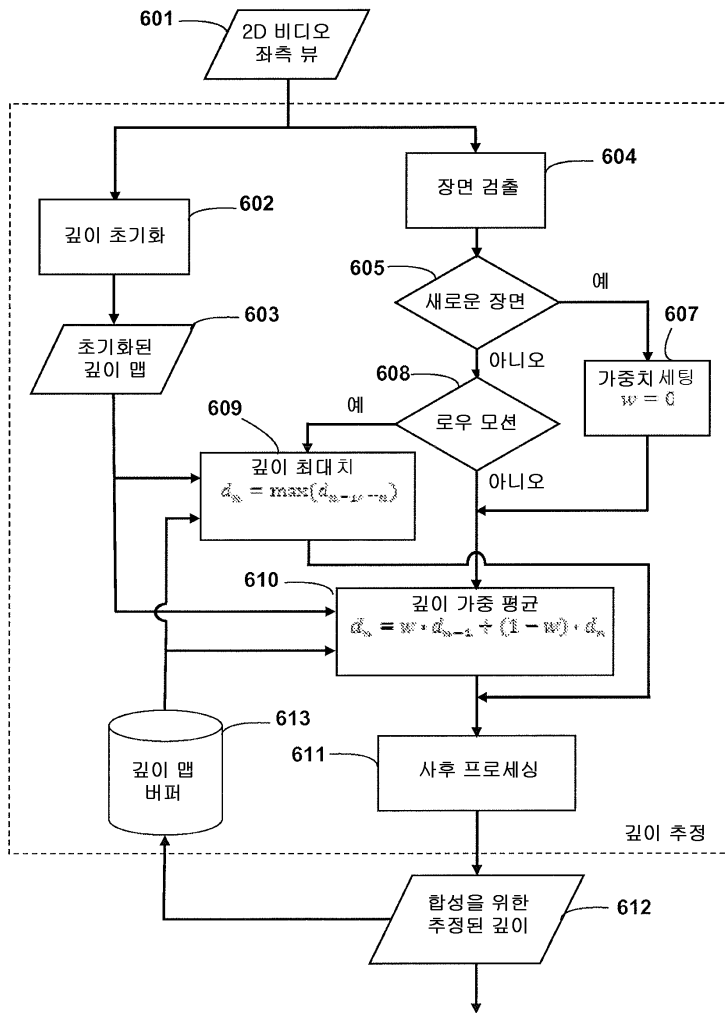
도면4



도면5



도면6



도면7

